

## EDUCATIONAL USE OF CLOUD COMPUTING AND AT-MEGA MICROCONTROLLER – A CASE STUDY OF AN ALARM SYSTEM

Tomasz Cieplak<sup>1</sup>, Dariusz Lecyk<sup>2</sup>, Agnieszka Cieplak<sup>3</sup>

<sup>1</sup> Lublin University of Technology, Management Faculty, Department of Enterprise Organization, Nadbystrzycka 38, 20-618 Lublin, Poland, e-mail: t.cieplak@pollub.pl

<sup>2</sup> PSW of Pope John Paul II, Biała Podlaska, Poland, e-mail: dlczyk@gmail.com

<sup>3</sup> Asseco Business Solutions S.A. Lublin, Poland, e-mail: agnieszka.cieplak@assecobs.pl

Received: 2016.03.01

Accepted: 2016.04.25

Published: 2016.06.01

### ABSTRACT

The article showcases a case study of Cloud Computing model combined with AT-Mega microcontrollers for educational purposes. The presented system takes advantage of many aspects of Internet of Things model, thus conjoining Cloud Management system with measurement-execution module based on Arduino platform. One benefit of this solution is a cost-effective way of showcasing machine and device integration with distinct cloud services. This article is based on practical experience with students' projects and an home alarm system with use of a Cloud Computing services will be described.

**Keywords:** computing cloud, IBM Bluemix, Arduino, internet of things.

## INTRODUCTION

The Internet has become a commonplace tool to such an extent that it has become necessary to introduce students of IT to combined fields of Information Technology and Electronics. In particular, the ever-increasing prominence of Internet of Things whose notion is firmly based on close collaboration of Internet services and electronic devices creates a good reason for preparing such classes [6]. There are many fields where Cloud Computing model has been successfully substituting classic IT solution based on local infrastructure. Moreover, many modern human endeavors requires constant use of electrical devices of varied complexity and functionality [5].

The Curriculum of IT faculty encompasses classes focused on programming classic application, as well as programming for embedded systems [10] and programming for mobile devices [9]. Teaching programming for cloud solution has also become a trend, recently. All of the above contribute a solid foundation for use during prac-

tical workshop classes, such as IT Project class. In the following article Arduino platform was used to showcase an example of programming of a microcontroller. In addition, the students can apply their knowledge in application' development for the Android operating system. The most interesting feature of the present system is the ability to integrate multiple hardware platforms and software platforms using Cloud Computing services of IBM Bluemix.

## BACKGROUND – COMPONENTS USED IN THE PROJECT

Within the project's specifications the students can take advantage of the following technologies and hardware solutions:

- hardware layer: Arduino and other communication modules and peripherals, Android OS based mobile phone [12].
- software layer: C for Arduino, MQTT protocol, IBM Bluemix computing cloud. Android OS software.

## Arduino

Arduino microcontroller is an embedded computer system capable of controlling other daily use devices. It enables the user to build electronic devices that can be controlled remotely thanks to programming code tailored specifically for these purposes. Arduino can also receive signals from input peripherals and send them to output extensions [2]. Its sheer flexibility contributes to great scalability – it can be used to create both simple and impressively complex solutions while maintaining simplicity of software architecture. Arduino is an electronic platform based on open source and open hardware licenses. Arduino Uno microcontroller used in this project, which is distinguished by 12 digital pins and 6 analogue pins, has been build based on Atmel ATmega328P 8-bit microcontroller. The digital pins may be used as digital inputs and outputs. Analogue pins, on the other hand, can function as analogue inputs, via which analogue sensors can be connected to Arduino [1].

## PIR (Passive Infra-Red) Sensors

PIR Sensors are capable of detecting movement. They are small, cost-effective devices characterized by low power-consumption and the ease of use. For this reason, they are common in security system installations. In most cases PIR is made of pyroelectric sensor, which enables it to detect infrared. Living beings radiate substantial amount of infrared radiation, which is why they can be detected by the sensors. The detector in the sensor is composed of two detectors, which identify not only the level of radiation but also movement. Each of those detectors is made of special infrared-sensitive material. In addition special camera lenses are used to improve their detection capabilities (this depends also on their sensitivity). When the sensor is in default state both detectors observe the same amount of infrared. When the observed object such as a human or an animal moves within the sensor's range it creates a positive change in potentials between the two detectors. When the objects moves away from the detection range of the sensor the opposite happens - the sensor generates negative shift in potentiality [11].

## Radio Module

NRF24L01 is a singular two-way radio module with in-built EnhancedShockBurst en-

gine, which works on 2,4 Ghz frequency. This makes an optimal choice for wireless, power-conservative use. The module works within the frequency range of 2.400 – 2.4835GHz. “EnhancedShock it Burst” functionality improve the power efficiency of two-way and one-way systems, without any redundant complexity added on the host controller's side. “EnhancedShockBurst” enables a user to configure a wide range of parameters, such as maximal number of retransmission and delay between transmissions. Its control is automatic, without any input from microcontroller. “Enhanced Shock Burst” guarantees fast data transmission and processing even without access to fast microcontroller. nRF24L01 module configured to be PRX can receive data from 6 different streams. Each data stream may have a unique address but it must fit within the same frequency channel. As a result 6 different nRF24L01 devices configured to be PTX can communicate with one nRF24L01 device configured as PRX. In effect all streams can take advantage of “Enhanced Shock Burst”.

## MQTT Protocol

The data is not only saved on a device but, crucially, it is processed in the cloud. To achieve that, the device is sending all measurements as MQTT protocol packages, published under the topic of the unique device identifier. Because of the fact that every data package (acceleration, time and location) is useful for further processing only as a whole package, the packages are sent as one, rather than being published separately under different MQTT topics.

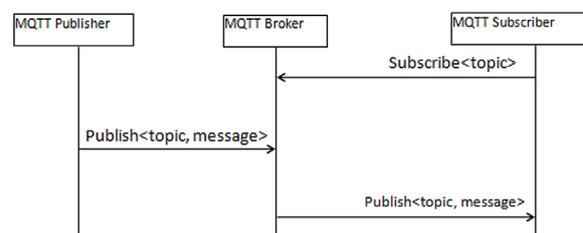


Fig. 1. MQTT publish – subscribe mechanism [3]

The MQTT protocol uses the publish/subscribe principle (see Fig. 1). It is a useful lightweight messaging protocol that proves effective even when working with low power sensors and lossy networks. MQTT is TCP-less and does not require the IP protocol layer [3, 7].

### IBM BlueMix platform

IBM Bluemix is an open cloud platform developed by IBM, which enables programmers of web and mobile applications to access service integration software, security solutions and other important features. Bluemix was developed using CloudFoundry technology under open source license. It offers PaaS (Platform as a Service) model and, additionally MBaaS (Mobile Backend as a Service) solutions. Its primary function is to simplify accessibility of applications over the Internet as services based on it, once properly implemented, are ready for instant use. Bluemix offers the following capabilities:

- a set of services which enable fast development of mobile and web applications,
- goal-oriented models and software services,
- ease of service and application management,
- constant accessibility.

Bluemix can host a wide range of applications whether they are desktop applications created with the most popular scripting languages, Android or iOS based mobile applications or web applications developed with HTML and JavaScript. Bluemix also provides an environment for hosting applications' artifacts running on an application server, such as Liberty. In addition Bluemix takes advantage of SoftLayer services and implements virtual container which can then host applications. This enables a programmer to use ready services in development process. Lastly, the clients, understood here are either developers or mobile applications can take advantage of REST mechanism, which renders access to API

services available, and this in turn permits sending requests via Bluemix between application instances and advanced services.

### Node-RED

Node-RED connects devices, APIs and online services to create Internet of Things. Node-RED features browser-like stream editors which render the whole process of combining these resources together through a wide range of components, straightforward. The in-build library enables users to save particularly handy functions and templates for later frequent use. The executive environment has been developed with Node.js platform as its foundation, fully taking advantage of event-driven model. This makes it a perfect choice for low-grade networks and cost-efficient hardware. The repository that comprises over 225 000 extensions, which may add additional functionality. Node-RED's streaming features are stored with JSON syntax and thus they can be imported and exported easily [8].

### IMPLEMENTATION OF THE CLOUD-COMPUTING SYSTEM

To simplify the IoT model for the present research, a diagram of the measurement system in the Cloud was introduced (see Fig. 2) [4]. The basis of the system is measuring equipment like a PIR sensors connected with the alarm control unit by use of radio. The sensor and the control unit are treated as one object and a source of physical phenomena. Devices of this type form the sensor

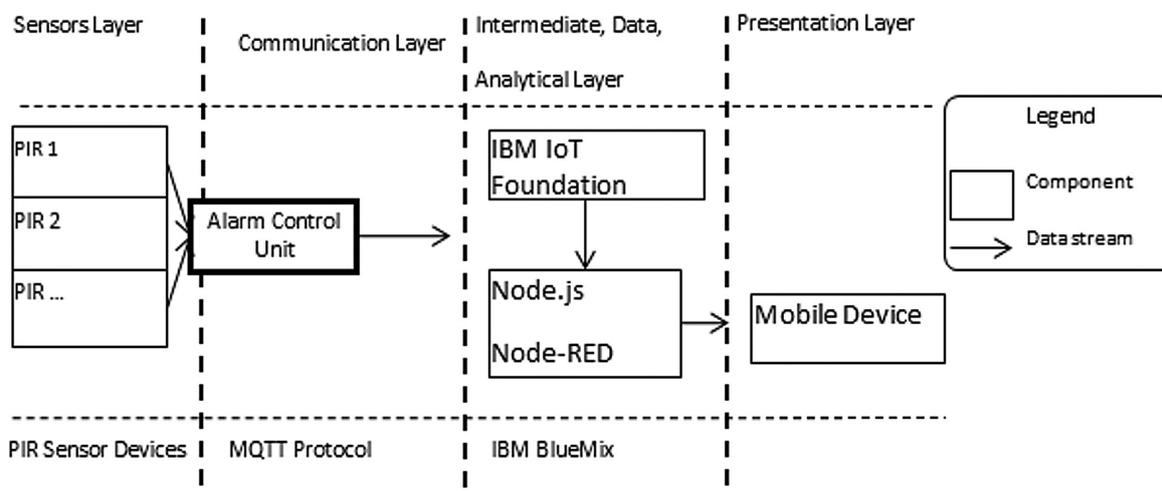


Fig. 2. Diagram of the testbed setup - own elaboration

layer. Next, the communication layer gathers all sorts of devices that enable retrieving basic signals from the meters and detectors. In this case study, the alarm control unit is responsible for communication as well as integration of signals to one data stream. Then, the data goes to an intermediate layer. The aim of this layer is to prepare the data in such a way that they can be transferred to a database system, e.g. one localized in the Cloud service. The intermediate layer registers the measurement system as virtual devices with a set of measured parameters. This layer maintains rules of security and device access to the system. In order to transmit the data an MQTT protocol was used.

The implemented software system was based on the architecture depicted above. The application was built by means of Node-red, an open-source tool for building solutions for the Internet of Things [8].

**Alarm control unit**

All of the elements were connected either on a 830-contact circuit board or directly to the system. The circuit board is a very convenient solution for students since should the components be placed incorrectly, it is easy to relocate them to fix the problem. A 9 V AC power adapter was used to supply power. Network access has been obtained via Arduino Ethernet Shield module, which includes an Ethernet cable port. The display and membrane keyboard positioned at the front panel constitute the communication interface with the device.

In addition, a 10kΩ resistance potentiometer was connected to the display as a voltage divider, which is responsible for the contrast in the display. The display communicates the status of the system as well as date, time and information about the system’s creator.

The membrane keyboard is another important communication device. Since the keyboard uses 8 digital pins, and Arduino Uno has a very limited supply of these, a method was needed to connect this keyboard to a single analogue pin, which could read the value of voltage. In order to connect the keyboard a set of 8 resistors was used, 4 out of which have 1kΩ resistance, three 220Ω resistance and one 10kΩ resistance. These values were carefully chosen so as to easily differentiate between pressed keys. For example, the resistance will be highest for key “1” since only a single 1 kΩ resistor will generate resistance, while for key “9” the total resistance of 3440Ω will be generated by five different resistors [13].

The device also includes Buzzer module, which is responsible for generating sound signal when alarm is triggered, and wireless connectivity module, through which the device connects with another device - operational PIR movement detector.

**PIR sensor device**

All elements of PIR sensor device were connected into small universal circuit board or directly to the system. The miniscule size

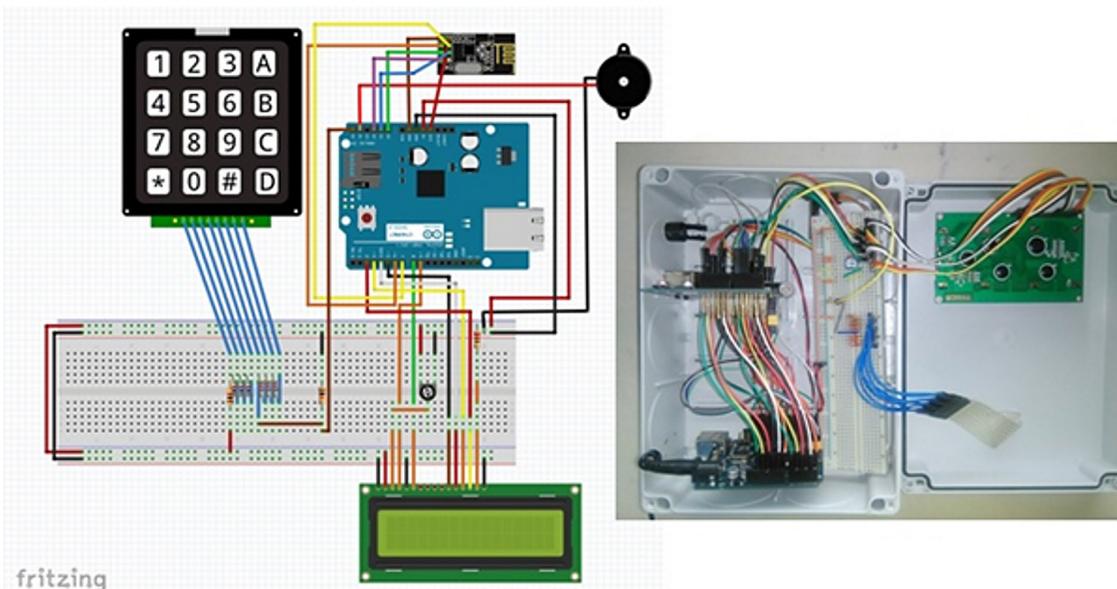


Fig. 3. Diagram of alarm control unit and its implementation – own elaboration.



Fig. 4. Digital display of alarm control unit - own elaboration

of the board was due to the size of the case itself - large circuit board would not fit the case. The system is powered by 9V battery and as a result the entire structure is very compact. A switch has been attached to battery so as to switch off the device when alarm is not needed.



Fig. 5. The alarm control unit - own elaboration



Fig. 7. The sensors unit - own elaboration.

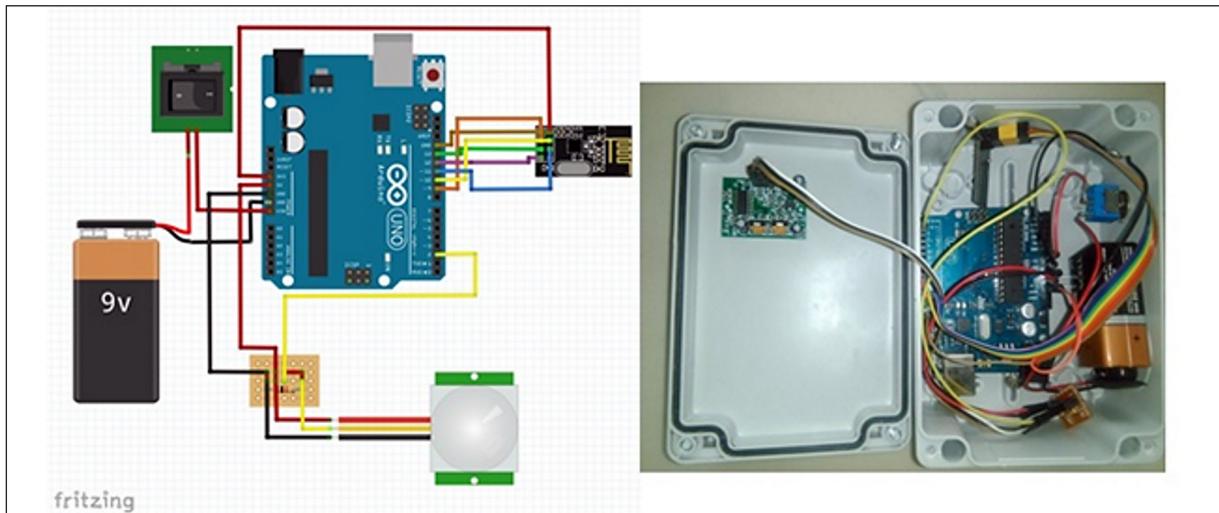


Fig. 6. Diagram of sensors unit and its implementation - own elaboration.

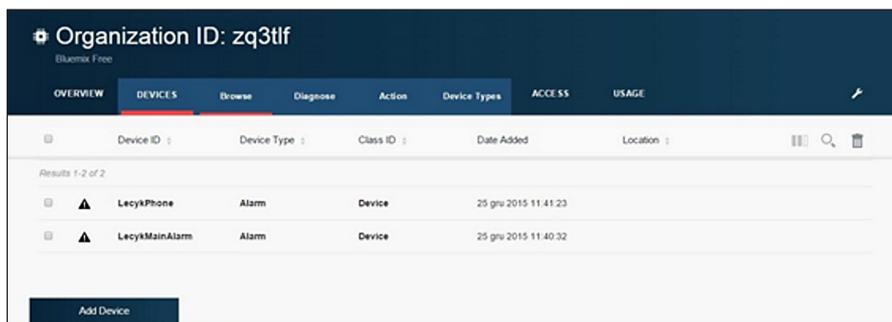


Fig. 8. Screen-shot showcasing the device added into IBM Bluemix platform - own elaboration.

### Integrating device with the Cloud

The cloud integration procedure has been performed using Internet of Things Foundation service available on IBM Bluemix platform.

Two devices were to IBM Bluemix platform. The first of them is Arduino central alarm unit, the second is a mobile Android device. The devices communicated with each other through a Node-RED-based application. Both devices were configured in accordance with the parameters provided by the platform: Device ID and Authentication tokens necessary to establish connection between them.

Figure 8 presents screen-shot showcasing the device added into IBM Bluemix platform. Figure 9 presents IoT application which takes advantage of Internet of Things foundation service and Node.js runtime. This application ensures correct communication between the devices. There is also a link in information pane, clicking which will enable configuration tools of the application.

### Connecting data source with data exchange system

The foundations of the abovesaid Node-RED application are firmly based on three Nodes: Input Node, Output Node and Functional Node.

The purpose of this application is translating the status information of one device into command sent to the other piece of hardware registered in IoT Foundation service. This ensures communication between the alarm and the mobile device.

In order that the application could receive the data from the cloud the Input Node must be first authorized to do so. This can be achieved via configuration tool in Bluemix platform services. The data necessary to configure this Type come from IoT Foundation Service. Since the setup of exchange between the devices is event-based, in the configuration panel option “Device Event” should be chosen.

The code of the Function Node reads the received message and declares it to a new variable following JSON syntax so that it can be sent on as a command.

IBM Bluemix platform enables configuration of Output Node, which must be first authorized in order to send data as commands. All the necessary parameters are provided by IoT Foundation service. The message will be interpreted as a command, and hence “Device Command” option should be chosen in the Device panel. The “Data” field on the other hand should reference the message declared earlier in Functional Node, which in this case is “`msg.payload`”

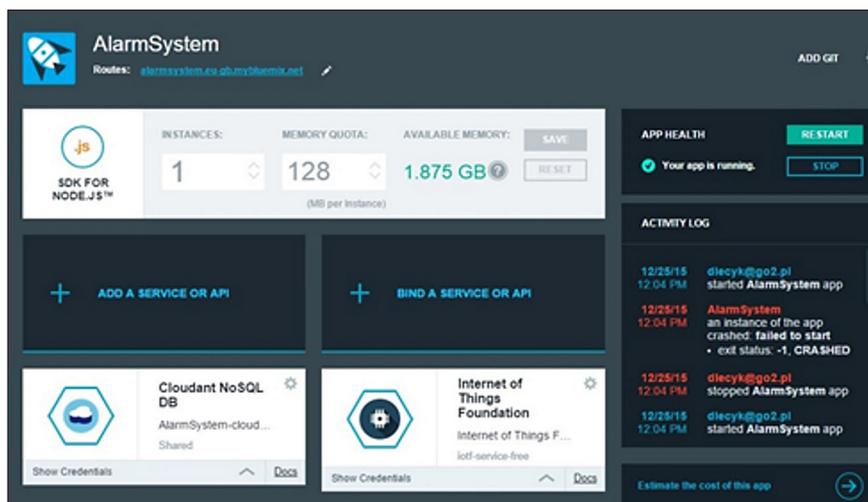


Fig. 9. Screen-shot presents application added to IBM Bluemix platform - own elaboration.

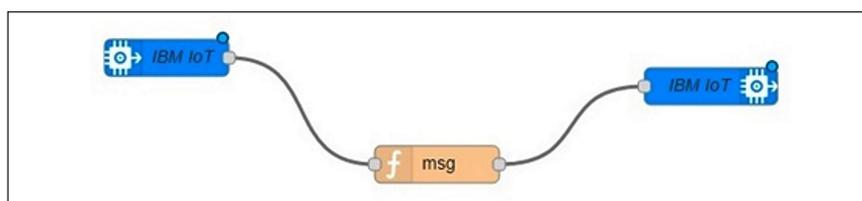


Fig. 10. Screen-shot presents Node-RED connectivity scheme - own elaboration

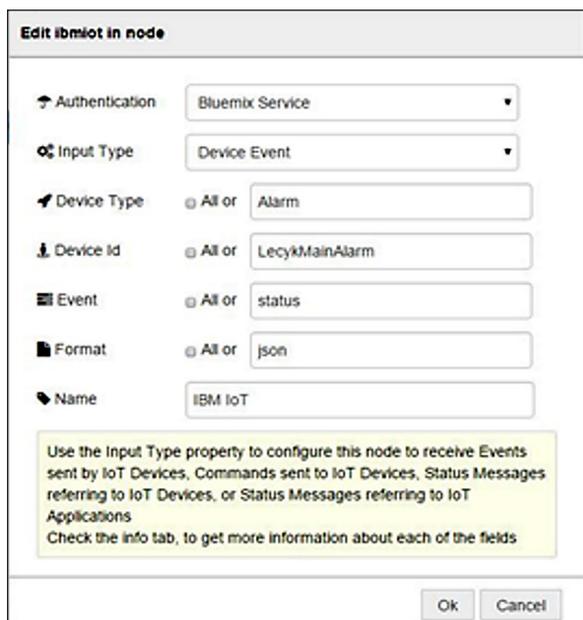


Fig. 11. Screen-shot showcases Input Node properties - own elaboration

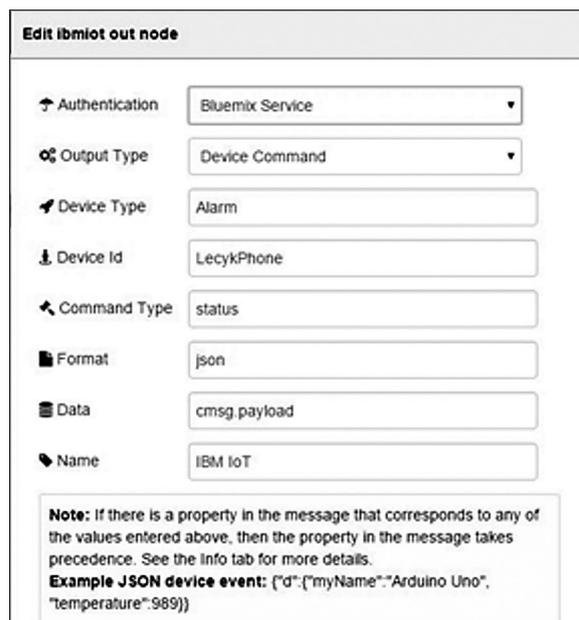


Fig. 13. Screen presents configuration of Output Node - own elaboration

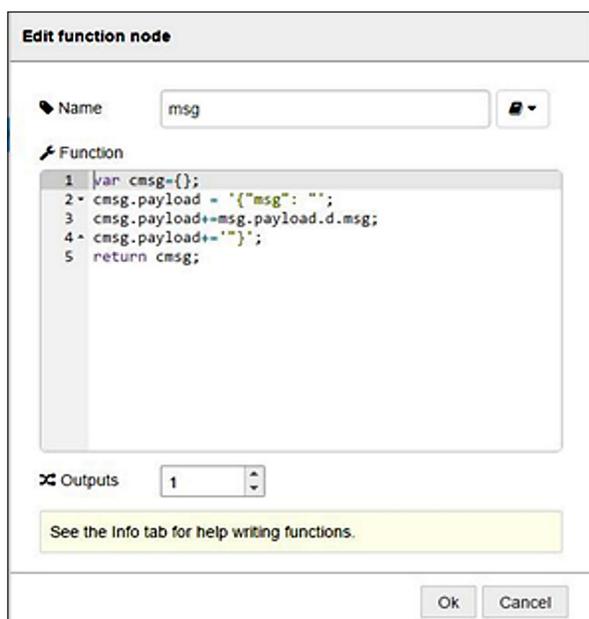


Fig. 12. Screen-shot presents the configuration of Functional Node - own elaboration

This configuration is sufficient for the functionality required in this project. As a result the devices communicate with each other without any issues.

## CONCLUSIONS

The presented project requires a wide range of skill developed by a student in the course of education. The functionality of the system and

its results are clear and easy to verify. Students are familiarized with capabilities of Internet of Things including the capacity for automation of different layers of hardware. In addition they take advantage of a wide selection of programming languages and communication methods within a single coherent system. The languages and hardware platforms used in this project are also common and cost-effective, which renders it accessible to any university regardless of financial considerations. The classes which employ the projects of this kind are considered to be interesting and mind-broadening by the students who participate in them voluntarily. Such projects lay solid foundations for future experiments with more advanced multiplatform systems.

## REFERENCES

1. Durfee W., University of Minnesota, Arduino Microcontroller Guide, 2011, [www.me.umn.edu/courses/me2011/arduino/](http://www.me.umn.edu/courses/me2011/arduino/)
2. Hardianto T., Supeno B., Kurnia Setiawan D., Arifin S., Gunawan, Wahyu E., Design of Real Time Anemometer Based on Wind Speed and Direction, Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics, Palembang, Indonesia, EECSI, 2015.
3. Hunkeler U., Truong H., Stanford-Clark A., MQTT-S: A Publish/Subscribe Protocol for Wireless Sensor Networks, in Proc. Workshop on Information

- Assurance for Middleware Communications IAM-COM, 2008.
4. Internet-of-Things Architecture IoT-A Project Deliverable D1.2 – Initial Architectural Reference Model for IoT,” 2015.
  5. Jamieson P., Herdtner J., More Missing the Boat - Arduino, Raspberry Pi, and Small Prototyping Boards and Engineering Education Needs Them, *Frontiers in Education Conference (FIE)*, IEEE, 2015, 1–6.
  6. Kortuem G., Bandara A., Smith N., Richards M., Petre M., Educating the Internet-of-Things generation. *Computer*, IEEE Computer Society, 46 (2), 2013, 53–61.
  7. Lampkin V., Tat Leong W., Olivera L., Rawat S., Subrahmanyam N., Xiang R., Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry, IBM Redbooks, 2012.
  8. Node-RED, A visual tool for wiring the Internet-of-Things, Available: <http://nodered.org/>. [Accessed 10 01 2016].
  9. Parkhomenko A., Gladkova O., Kurson S., Sokolyanskii A., Ivanov E., Internet-Based Technologies for Design of Embedded Systems, *Journal of Control Science and Engineering* 2, 2015, 55–63.
  10. Purdum J., *Beginning C for Arduino - Learn C Programming for the Arduino*, Second Edition, Apress Berkely, 2015.
  11. Rustige T., *Home Security Projects for Arduino*, TR Computers Limited, 2015.
  12. Schwartz M., Buttigieg S., *Arduino Android Blueprints*, Packt Publishing, 2014.
  13. Wilcher D., *Learn Electronics with Arduino*, Apress Berkely, 2012.