IoT Connectivity Application for Smart Building based on Analysis and Prediction System

Ghenadie COROTINSCHI[†] Cătălin FRÂNCU[†], Ionel ZAGAN[††] and Vasile Gheorghiță GĂITAN[††]
[†]SC Building Technology Group R SRL
[††]Ștefan cel Mare University of Suceava 13, Universitătii Street, RO-720229 Suceava;
Integrated Center for Research, Development and Innovation in Advanced Materials, Nanotechnologies, and Distributed Systems for Fabrication and Control (MANSiD), Stefan cel Mare University, Suceava, Romania

Abstract
The emergence of new technologies and their implementation by different manufacturers of electronic devices are experiencing an ascending trend. Most of the time, these protocols are expected to reach a certain degree of maturity, and electronic equipment manufacturers use simplified communication standards and interfaces that have already reached maturity in terms of their development such as ModBUS, KNX or CAN. This paper proposes an IoT solution of the Smart Home type based on an Analysis and Prediction System. A data acquisition component was implemented and there was defined an algorithm for the analysis and prediction of actions based on the values collected from the data update component and the data logger records.

Key words:
IoT, Smart Building, Data Logger, Analysis and Prediction System

1. Introduction

In recent years, the progress of information technologies in terms of hardware and software has led to an increase in the development of new technological branches. Research on the applicability and implementation of new solutions, using information technologies, has led both to the automation of many economic sectors, such as transport, agriculture or distribution networks, and to the development of new sectors such as building automation or health monitoring devices. The use of technologies in different economic fields has led to the emergence of new business models, such as that of the "smart" branches within each economic field: "smart grid", "smart-agriculture", "smart logistic", "smart home" or "smart medical care". The modernization of the main energy systems and the implementation of the "smart-grid" concept is a priority at the level of central and regional authorities.

The Internet of Things (IoT) field has rapidly diversified, adapting to the main areas of economic activity, and the main IoT categories enjoying major interest from both researchers and authorities [1] are Industrial Internet of Things (IIoT), Smart Homes, Internet of Medical Things (IoMT) and Smart Cities. Smart Homes means the multitude of systems within homes and buildings, including: smart devices, televisions, thermostats, air conditioning systems and security systems. All of these exchange information in order to provide increased comfort to the end-user and to improve the energy efficiency of buildings.

The universe of IoT devices targeting the smart home devices sector has enjoyed an extraordinary rise. According to Gartner research company, the building automation sector will see a major growth rate in the next years [2]. In the case of buildings, automation systems are homogeneous, most often equipment from the same manufacturer and a single communication protocol being used. In the case of home users, this is not possible due to the multitude of devices and their diversity. Some devices use wireless communication media (Wi-Fi, Bluetooth, Z-Wave, LoRa or ZigBee), while others can be connected using wired connections and communication protocols such as Modbus [3] or KNX. With regard to cloud solutions, a diversification of the concept of cloud computing has emerged. This evolution includes software solutions provided via cloud platforms (SaaS - software as a Service), as well as infrastructure as a service (IaaS - Infrastructure as a Service) or even the support of business processes in the cloud (BPaaS - Business Process as a Service). In order to be able to implement a solution that is suitable for smart home systems, it is necessary to carry out an analysis of the main types of cloud platforms and of the advantages and disadvantages of each type of solution, in order to build cloud solutions that are applicable to a large number of automations of houses and buildings. Fog Computing technology solves problems related to congestion and delays in the network. The concept of "Fog Computing" or "fogging" provides computing power, data storage space and network services at the edge of the network. It also provides an intelligent platform for managing the infrastructure of real-time and distributed IoT applications or of applications undergoing development.

Manuscript received September 5, 2021
Manuscript revised September 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.9.13

The first section of the paper contains a concise introduction, and section 2 presents related work within the current IoT, communication protocols, LoRa and Smart Building challenges. Section 3 is dedicated to the IoT concept based on Analysis and Prediction System (APS) for Smart Building. Section 4 contains the conclusions and future directions of research.

2. Related Work

IoT architectures are constantly changing, and for this reason, special attention has been paid to the policy of versioning databases by using migration files. The authors [4] use wireless devices and cloud technologies to expand the concept of "smart-agriculture", by monitoring parameters such as humidity, temperature and even the movement of insects and sending alerts to farmers.

In [5] the authors provide a comprehensive study on the wireless networks used in IoT applications and beyond, including the technical challenges of deploying LoRa networks and recent solutions in the field. Based on the LPWAN (Low Power Wide Area Networking) technologies, by investigating the challenges they face during the deployment of LoRa networks, the recently developed solutions are discussed in detail in [5]. In [6] the authors use LoRa to propose and implement a smart home system for remote monitoring and maintenance of sensors and IoT devices using the concept of artificial intelligence (AI). All sensors and IoT devices are designed to be powered from the intelligent solar energy storage system (e.g. solar panel and battery). An AI-based data server can also be designed to control and monitor household appliances. The paper [7] proposes Industrial LoRa, a new communication environment access strategy for LoRa, which allows the implementation of industrial wireless networks for IIoT and Industry 4.0 applications. The authors describe the proposed MAC scheme as well as the advantages it offers, the following parameters being considered for the simulation: Coding Rate (4/5), Bandwidth (125kHz), Transmission Power (14 dBi), Spreading Factor (SF) (7, 8, 9), Sub-bands (h1.4 - h1.6 - h1.7), Physical Payload (50 bytes), Periodic generation messages period (35s) and Superframe duration (17.5s). A simulative assessment considering a maximum indicative coverage range (125m, 180m and 250m) for messages sent with a certain Spreading Factor (SF) shows the performance of the approach proposed in a realistic scenario that complies with the requirements of the real-time software applications in Industry 4.0.

Although it is a relatively new concept, Fog Computing enjoys a growing popularity among researchers. In the paper [8], the authors expose Cisco's vision of Fog Computing, highlighting the main characteristics of this new concept. At the same time, some examples of applications such as those in the automotive field, Smart Grid, Smart Cities and wireless sensors and sensors networks (WSAN's - Wireless Sensors and Actuators Networks) are also given. At the same time, the papers [9], [10] identify the main security threats at the level of the devices in the Fog network, and some experimental results are also presented. In [11] the authors present a project in which a wireless EEG (electroencephalograph) medical device is used, using IoT infrastructure. An interfacing system was implemented between the human brain and the computer through which the data is processed by the local Fog server, managing to move some objects within a soft application according to their level of concentration.

3. IoT Architecture based on APS for Smart Building

The architecture proposed within the data acquisition system is composed of several interacting sub-systems and servers, for increased system efficiency, and for modular implementation so that the development of subsequent sub-modules is carried out without a major change in the architecture. The main components of the architecture are the analysis and prediction system, the database server, the message management server and the data acquisition system. The architecture in Figure 1 was designed based on the needs of a building automation system called "Home Automation" or "Building Management System". Within these types of systems, the mechanisms for integrating new devices and their efficient management are of great importance.

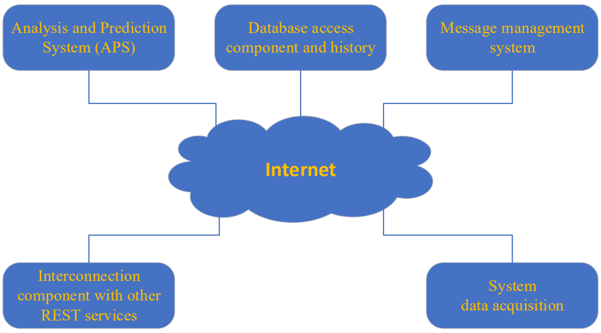

Analysis and Prediction System (APS)
Database access component and history
Message management system
Internet
Interconnection component with other REST services
System data acquisition

Fig. 1 General architecture of the data acquisition platform [12].

The APS is developed to take over some of the operators' tasks, to automate some of the actions required within the application and which can be automated based on previously established rules. This set of rules can be determined according to certain constant or variable parameters. Constant parameters can be data, time constants, or fixed values. The execution of actions on certain dates are actions that run cyclically, in a certain time interval or actions that set parameters within the application with the values of predefined constants.

Variable parameters are those parameters that come from the data acquisition system, whose values vary over time, and the comparison of these values is made using comparison operators (>, <, ==, !=) or logical operators such as (|| or &&). The main components of the APS mode are as follows: Program, Event, and Actions. Within the APS, components are tree-organized so that a program can have multiple events defined, and each event can have a series of actions to be accomplished when the conditions within the event are met.

Home automation applications or any sensor that generates a data flow, will store the data logger through the dedicated server [12]. All clients need a token authentication. This token will be obtained from the Authorization Server. The role of the authorization server is to validate its username and password and generate an authentication token (authentication-token) and a refresh-token. The refresh token has a much longer expiration time than the authentication token and is used in the procedure of regenerating a new authentication token when it expires. All calls for API (Application Programming Interface) functions within the log server will be accompanied by the authentication token. The following is an example of the connection model of the data logger record storage application within the MongoDB database server, graphically illustrated in Figure 2.

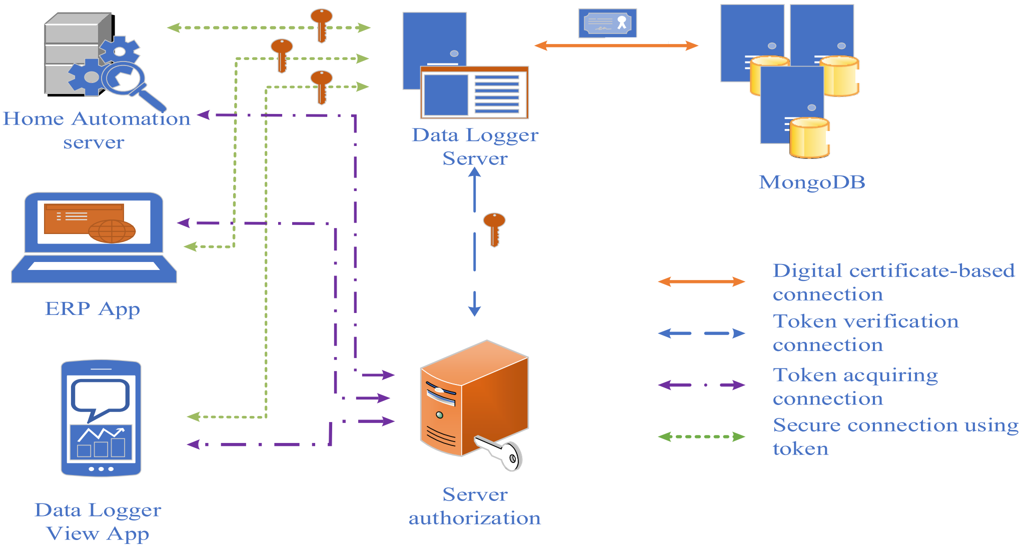
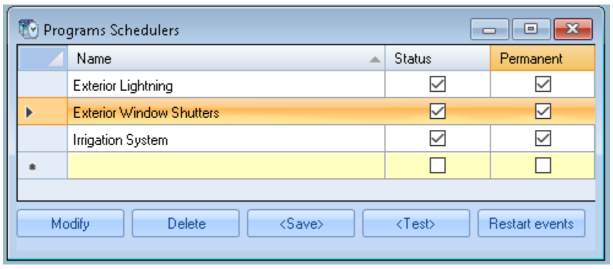

Home Automation server
Data Logger Server
MongoDB
ERP App
Data Logger View App
Server authorization
Digital certificate-based connection
Token verification connection
Token acquiring connection
Secure connection using token

Fig. 2 Overall architecture of the data logger storage application.

In order to have a functional and efficient system, it is necessary to have a well-established mechanism of predefined actions (e.g. turning off all lights and setting temperatures to a minimum threshold for common spaces when spaces become unoccupied). Sometimes, the definition of all scenarios is not possible, for this reason it was chosen to define and implement a module for prediction and analysis. This algorithm is able to identify some patterns and make autonomous decisions. For example, if, usually, the user sets his temperature to 22°C starting at 08:00, and after 22:00 the user sets the temperature to 20°C, if the "self-learning" mode is activated, having as a basis for calculating the data logger records of the respective parameter, the system will suggest to the user the automatic creation of a scenario based on the data provided by the "self-learning" algorithm. This algorithm will define some Time-type events, the Range subtype. The first event is for the interval [08:00-22:00], where as a suggested action will be the setPoint setting (desired temperature) to 22°C, and the second event is for the interval [22:00-08:00], where the suggested reference value (setpoint) will be 20°C. If the user accepts the proposed suggestion, this scenario will come into operation automatically and will be checked within the event scan cycle. Each parameter within the system has a unique identifier called the IDparameter. Within the self-learning algorithm, the variables were classified by using a Resource ID variable. The system performs actions based on probabilities that are calculated based on the algorithm described in [12]. The self-learning algorithm will calculate the probability of an event to trigger based on the data logger from previous days. The results of the prediction system based on the analysis of the data collected over a period of 4 days are presented. Following the analysis, the algorithm will calculate the following probabilities:

i. Interval 1 [10-12] – 50% by interpolation of the data from day 2 and 3;
ii. Interval 2 [14-18] – 75% by interpolating the data from days 1, 2 and 4;
iii. Interval 3 [18-20] - 50% by interpolating the data from day 1 and 4;
iv. Interval 4 [22-00] - 50% by interpolating the data from day 1 and 3.

The following will exemplify the use of the data acquisition and prediction module by using the proposed IoT application. Figure 3 presents a list of standard programs used in the case of a classical automation used to light the outdoor lighting, one for irrigation of the lawn in the yard and another for closing the blinds in case of bad weather conditions. Figure 4 exemplifies how to configure a Time event; the running cycle of the program (Daily) can be seen. It is only active on weekends and is active within the time frame given by the global variable Sunset and until midnight.

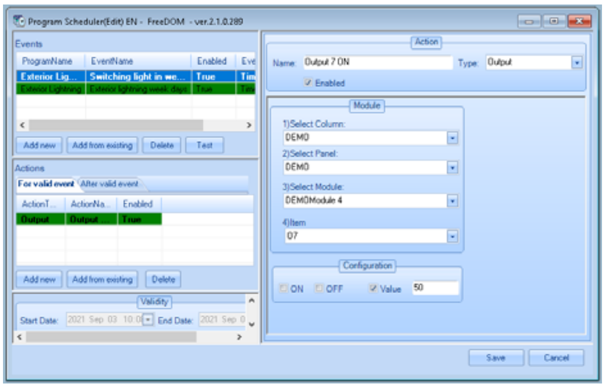

Programs Schedulers

	Name	Status	Permanent
	Exterior Lightning	☑	☑
▶	Exterior Window Shutters	☑	☑
	Irrigation System	☑	☑
*		□	□

Fig. 3 Program scheduler events.

An example of how to configure specific actions for an event when it is active is also provided. The event is of the Output type and consists of setting the 50% value corresponding to a variable output of a lighting circuit.

The implementation of home automation IoT solutions must allow the definition of adaptable and scalable solutions. A solution is required that can be used, both for small installations, such as that of an apartment that includes a relatively small number of monitored or operated parameters, sensors and automation scenarios, as well as for automation solutions for office buildings or residential complexes. In this context, the automation solution must comply with the same system performance requirements regarding the quality of services, low latencies in the operation of the application as well as an economically reliable solution.


Program Scheduler(Edit) EN - FreeDOM - ver.2.1.0.289

Events

ProgramName	EventName	Enabled	Eve
Exterior Lig...	Switching light in we...	True	Tim
Exterior Lightning	Exterior lightning week days	True	Tim

Actions

For valid event | After valid event

ActionT...	ActionNa...	Enabled
Output	Output ...	True

Validity

Start Date: 2021 Sep 03 10:0 End Date: 2021 Sep 0

Action

Name: Output 7 ON Type: Output

☑ Enabled

Module

1)Select Column: DEMO

2)Select Panel: DEMO

3)Select Module: DEMOModule 4

4)Item: 07

Configuration

□ ON □ OFF ☑ Value 50

Fig. 4 Scheduler edit actions if the event is active.

Defining automation solutions that can run both on devices with limited resources, such as RaspberryPI and on servers dedicated to automation applications is difficult to achieve. A multi-platform automation solution that supports both classic Windows/Linux/MacOS operating systems and mobile platforms can increase the implementation of IoT solutions. All applications that need to use the data logger server will have to follow the flow of actions taken. An attempt will be made to obtain an authentication token, which can be stored within the data acquisition application and can be used later without the need to resume the authentication step. Then it will verify the existence of a reference of the current project within the data logger server. If the project reference is missing, an attempt will be made to create a record with the current project ID data that consists of a single type record (ProjectHardwareID, ProjectHardwareSerialNumber) and that must be validated by the data logger server based on an algorithm known only at the level of this server. This way, a clear separation of projects is achieved and the confidentiality of the data within a project is ensured, as other users of other projects cannot access it. The structure of the data logger server API is of REST (Representational State Transfer) type, so calling functions can be done from any type of client, from automation servers, ERP (Enterprise Resource Planning) applications, data visualization applications on mobile devices, to embedded devices with limited resources.

For a quick integration of applications using the data logger server, a user-friendly test interface has been implemented, using Swagger Specifications, which is now called OpenAPI Specifications, and which is a unified way to describe, develop and view REST services. Figure 5 shows the configuration and testing interface of the data logger server using OpenAPI Specifications. An adaptive data selection algorithm has been implemented within the data logger server. This algorithm will allow a limitation of the number of values returned. Depending on the selected time period, the returned parameters will be grouped in certain time intervals. Table 1, Execution times with and without adaptive algorithm, presents the experimental results for testing the data logger server by comparative analysis of execution times in the selection of a single parameter on different time periods.

Fig. 5 Data logger server test and configuration interface.

Figure 6 is a graphical view of the data within the Table 1. Since the differences between the tests on the non-indexed database from the indexed one are of the order 10^2 and in some places 10^3, a logarithmic visualization was carried out in base 10 of the data in the following graph. The interconnection component of the data acquisition system consists of the REST interfacing module that uses a standardized way of defining the methods of access to the system components and uses the Swagger library to easily integrate the proposed system into other IoT systems. The data logger storage system can also be used by other devices and platforms due to the implementation of a REST architecture, with test interfaces using the Swagger library.

Table 1: Execution times with adaptive algorithm and without.

Period	***Unindexed Collection Query (ms)***			***Indexed Collection Query (ms)***		
	Queries on the database	***API REST***	***REST API with adaptive***	***Queries on the database***	***API REST***	***REST API with adaptive algorithm***
> 30 days	11968	114564.1	12069.2	103.7	106774.1	263.6
≤ 30 days	12377.6	72712.5	12112.6	98.3	68821.3	352.3
≤ 24 hours	11848	13987.8	12128.4	91.5	1419.1	367.2
≤ 6 hours	11688.5	11805.9	11951.2	92.2	832.5	247.8
≤ 1 hour	11914.2	11510.9	11813.4	90.8	260	186.2

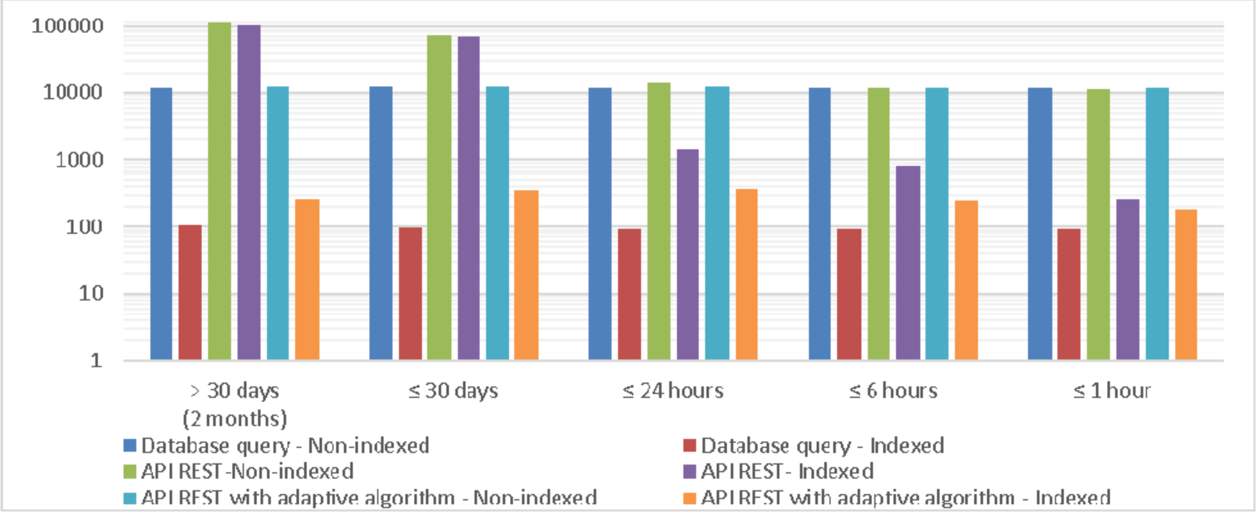


Fig. 6 Data logger server performance.

4. Conclusions and Future Work

The APS provides a unified way of defining program routines called events, and actions when the validation conditions of the event are met. The main algorithms of the event verification procedure as well as a self-learning algorithm based on the identification of certain user actions are also described. Adaptive algorithms for selecting data stored within the data logger server, which make it possible to return a small set of data depending on the selected period, help in using data logger records on devices with low processing power. These devices will have a set of values restricted depending on the selected time period, and the subsystems that must graphically display the parameter values will provide a satisfying experience to the end user. In terms of future work, the use of an architecture based on .NET Core allows the implementation of the ecosystem described in the current work, both on individual servers, cloud and on devices with limited resources such as RaspberryPI.

Acknowledgments

This project benefited from technical support through the „Operational Program for Competitiveness 2014-2020, Axis 1", Contract no. 138, dated 27.09.2016 „BRAIN-IN - Intelligent Automation Systems for Building Management, Production and Industrial Automation".

References

[1] S. Chen, H. Xu, D. Liu, B. Hu and H. Wang, "A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective," IEEE Internet of Things Journal, vol. 1, no. 4, pp. 349-359, 2014.

[2] "Gartner Says 5.8 Billion Enterprise and Automotive IoT Endpoints Will Be in Use in 2020," Aug 2019. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io. [Accessed 15 May 2021].

[3] V.G. Găitan, I. Zagan. "Experimental Implementation and Performance Evaluation of an IoT Access Gateway for the Modbus Extension," Sensors 2021, 21, 246. https://doi.org/10.3390/s21010246.

[4] G. Sushanth and S. Sujatha, "IOT Based Smart Agriculture System," in 2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 2018. .

[5] V. K. Sarker, J. P. Queralta, T. N. Gia, H. Tenhunen and T. Westerlund, "A Survey on LoRa for IoT: Integrating Edge Computing," 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), 2019, pp. 295-300, doi: 10.1109/FMEC.2019.8795313.

[6] M. Shahjalal, M. K. Hasan, M. M. Islam, M. M. Alam, M. F. Ahmed and Y. M. Jang, "An Overview of AI-Enabled Remote Smart- Home Monitoring System Using LoRa," 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2020, pp. 510-513, doi: 10.1109/ICAIIC48513.2020.9065199.

[7] L. Leonardi, F. Battaglia, G. Patti and L. L. Bello, "Industrial LoRa: A Novel Medium Access Strategy for LoRa in Industry 4.0 Applications," IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, 2018, pp. 4141-4146, doi: 10.1109/IECON.2018.8591568.

[8] B. Flavio, R. Milito, J. Zhu and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp. 13-16, 2012.

[9] I. Stojmenovic and S. Wen, "The Fog computing paradigm: Scenarios and security issues," in 2014 Federated Conference on Computer Science and Information Systems (FedCSIS) pp.1,8, 2014.

[10] S. Kulkarni, S. Saha and R. Hockenbury, "Preserving privacy in sensor-fog networks," in 2014 9th International Conference for Internet Technology and Secured Transactions (ICITST), 2014.

[11] J. Zao, T. T. Gan, C. K. You, S. Rodríguez Méndez, C. E. Chung, Y. T. Wang, T. Mullen and T. P. Jun, "Augmented Brain Computer Interaction Based on Fog Computing and Linked Data," 2014 International Conference on Intelligent Environments (IE), pp. 374-377, 2014.

[12] G. Corotinschi, "Contribuții privind arhitecturile Internet of Things," Ph.D. Thesis, University Ștefan cel Mare of Suceava, Suceava, Romania, 2020.

Corotinschi Ghenadie received M.Sc. degrees from the Gheorghe Asachi Technical University of Iasi, Romania in 2009 and Ph.D. degree from the Stefan cel Mare University of Suceava, Suceava, Romania, in 2020. His research interests include IoT systems, fieldbuses, communication protocols and software architectures.
E-mail: corotinschi.ghenadie@gmail.com

Cătălin Frâncu received M.Sc. degrees from the Gheorghe Asachi Technical University of Iasi, Romania in 2004. His research interests include IoT systems, web technologies, database systems and software architectures.
E-mail: catalin.francu@btgromania.ro

Ionel Zagan received the M.Sc. degree in computer science from the Stefan cel Mare University of Suceava, Suceava, Romania, in 2005. He is currently a Lecturer Ph.D. Eng. at the Department of Computers of Stefan cel Mare University of Suceava. His research interests include real-time systems, FPGAs, microcontrollers and pipeline processors with parallel execution of tasks. Mr. Zagan is a member of the IEEE Computer Society. Address: Str. Universitatii nr. 13, 720229, Suceava, Romania.
E-mail: zagan@usm.ro

Vasile Gheorghiță Găitan received the M.Sc. and Ph.D. degree from the Gheorghe Asachi Technical University of Iasi, Romania, in 1984 and 1997, respectively. He is currently a Professor at the Department of Computers of Stefan cel Mare University of Suceava, Romania. His main research interests include real time scheduling, embedded middleware, digital systems design with FPGAs, fieldbuses and embedded system application. He is a member of the IEEE, and a member of the IEEE Computer Society. Address: Str. Universitatii nr. 13, 720229, Suceava, Romania. E-mail: vgaitan@usm.ro