ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 3, No. 2, 2018

SENSORS MONITORING AND ANALYSIS SYSTEM FOR CYBERPHYSICAL SYSTEM "SMART HOME"

Bachynskyy Ruslan, Stefanivskyi Mykhailo

Lviv Polytechnic National University, 12, S. Bandery Str., Lviv, 79013, Ukraine
Authors' e-mail*: bac_ruslan@ukr.net*

Submitted on 01.12.2018

Abstract: **The system for various types of sensors monitoring and analysis has been described in the article. The system architecture and some aspects of implementation have been described.**

Index Terms: **Internet of Things, client-server system, indicators, smart home.**

I. INTRODUCTION

The building sensors values and status processing system is one of the components of remote monitoring and controlling system. More recently, the IoT (Internet of Things) term is used for describing network, which consists of interconnected physical devices with built-in sensors, as well as software, that allows the data transmission and exchanging between the physical world and computer systems, using standard communication protocols. The prospect of development of this concept is enormous, since the system built on the basis of the IoT allows remote monitoring of various indices in the room, which, in turn, allows quick detection of possible problems.

II. THE CONCEPT OF THE INTERNET OF THINGS

Internet of Things is one of the most popular concepts in modern futurology. Moreover, one of those few who are no longer concepts and are widely used in life.

According to the most widespread formulation, Internet of things is the concept of a computer network of physical objects (that is, things, actually) that are equipped with such technologies to interact with each other. The concept suggests that Internet of things can seriously affect society life, since it allows to automate a lot of processes. Internet of Things (IoT) is a global network of physical devices connected to the Internet - "things" that are equipped with sensors, sensors and data communication devices. These devices are connected to information control and processing centers. On the other hand, IoT, apparently, stands out in the middle of zero when the number of devices connected to the World Wide Web exceeded the number of users. IoT combines real things into virtual systems that are capable of solving completely different tasks. The main idea of the concept is to connect all the objects that can be connected to the network [4, 5].

The IoT architecture is shown in Fig. 1.

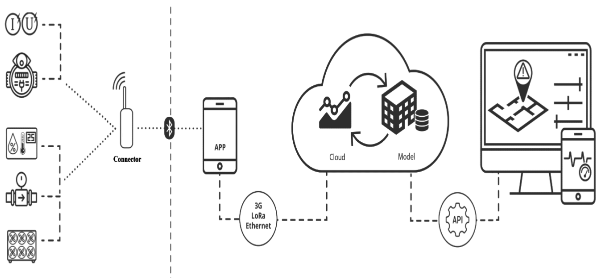


Fig. 1. The architecture of IoT

The main advantage of such system is in possibility to automate some processes, which allow people to have more free time. The connected devices on the Internet will also give people more opportunities for efficient resource management. Already today, they help to optimally spend heat, water, light and save on payment of utilities. It is important to note that not only the lives of individuals, but also entire industries will change.

One of the most susceptible industries to change will probably be a telecom, as mobile operators will gradually change their business models from network providers to smart service providers and applications [1, 2].

III. TECHNOLOGIES USED FOR IMPLEMENTATION

To combine everyday things in the network the following should be taken into account:

- Object identification. Only unique identification system can provide correct collecting and accumulating information about a particular object in the network.
- Data processing. For sensors, data processing and accumulation, a built-in processor (computer) such as Raspberry Pi, Intel Edison should be used.
- Data transmission without wired connection. Wireless technology (Wi-Fi, Bluetooth, ZigBee, 6LoWPAN) can be used to exchange information between devices, which are mobile and autonomous.

First of all, we need some sensor, for example The Eaton VoltageWatch, which is shown in Fig. 2.

We will collect data from this sensor using communications protocol Modbus. It has become a standard communication protocol and is now a commonly available means of connecting industrial electronic devices. Data from the sensor is transmitted to the “Sensor Connect”, which is responsible for sending this data to the server.

Fig. 2. The Eaton VoltageWatch

The server based on Node.js or .NET Core after having received the data must save them. For the system implementation, Node.js server is chosen. The principle of Node.js is shown in Fig. 3.

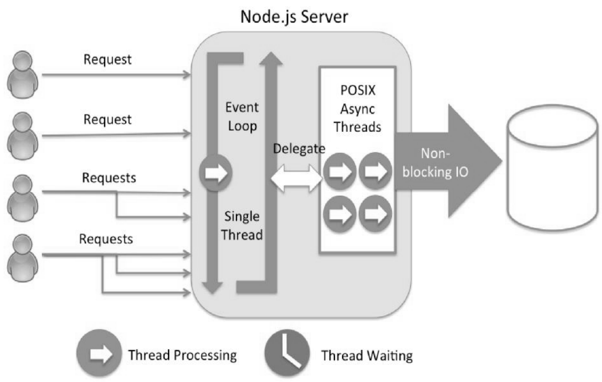


Fig. 3. Node.js server

To store data we need to use a “key-value” database (MySQL, ClickHouse, MongoDB or Firebase). The database view is shown in Fig. 4.

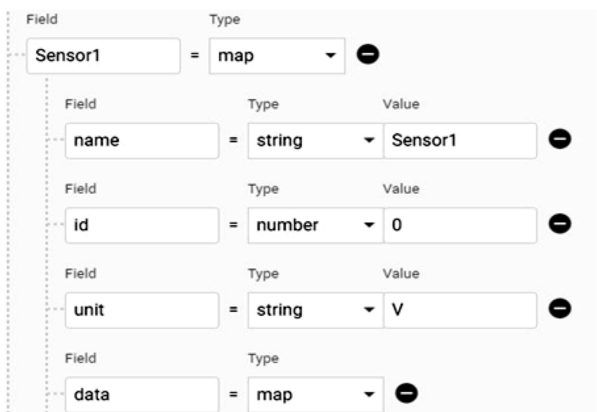


Fig. 4. Database view

After data saving, it could be depicted on the Web Client with the help of JavaScript framework – Angular (architecture is shown in Fig. 5).

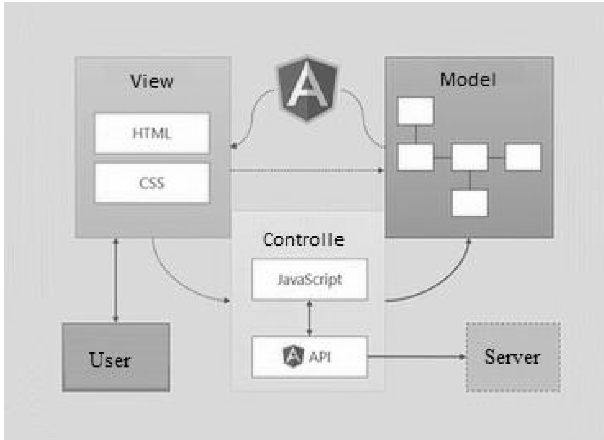


Fig. 5. Architecture of Angular framework

The server will interact with the Angular with the help of REST API. Fig. 6 shows how REST API works

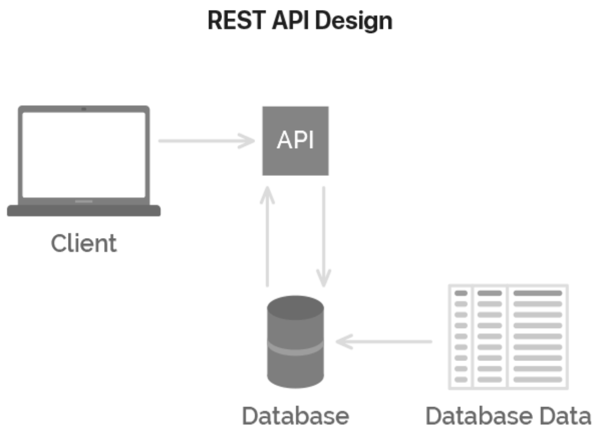


Fig. 6. REST API operation

For the data transfer in real time, WebSocket communications protocol is chosen, what provides full-duplex communication channels over a single TCP connection. It is shown in Fig. 7

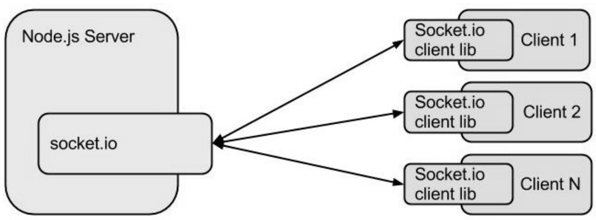


Fig. 7. SOCKET.IO

An important element in building the Internet of things system is also an architectural template. To solve this problem, the best architectural template is Model- View-Controller (MVC). It is an architectural template that is used when designing and developing software.

This template separates the system into three interrelated parts: a data model, a view (user interface), and a control module. It is used to separate data (models) from the user interface (view), so that changes to the user interface minimally affect the operation of data, and changes in the data model could be carried out without changing the user interface. [[3]]

Fig. 8 illustrates MVC pattern.

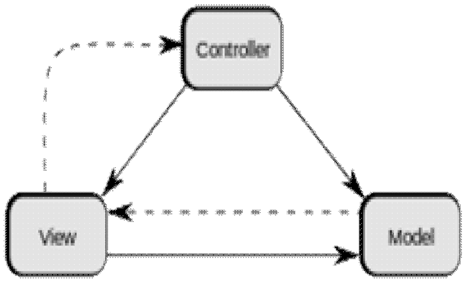


Fig. 8. MVC pattern

IV. THE SYSTEM DESCRIPTION

The system collects data from the sensors of various types. Using Modbus communications protocol, data from the sensors are transmitted to the Sensor Connect, which is responsible for sending them to the server. After receiving a data array from server, the system processes it by generating the input data that is depicted on the graphs. After the graph is shown, it is necessary to specify units of measurement, which also come in the array.

Graph is a graphic display of indicators from different types of sensors. It is also required to implement the possibility to view metrics in online mode with the help of WebSockets. The system is also able to analyze the data obtained. Furthermore, such systems provide reaction on the sensors values changes. For example, if the room temperature has started to drop sharply, the system could react on that in an automatic mode (without human help) by using an additional heater. To make it possible, it is necessary to allow to set the minimum and maximum ranges of the sensor values. When sensors exit a certain range, system must inform the user (using SMS/email/message on user interface) and provide an appropriate reaction on that.

V. THE DESCRIPTION OF RESULTS

First of all, the user must pass the authorization in the system. This process is shown in Fig. 9.

In case of successful validation of the user in the system, it will be redirected to the main page, where the status of all active premises is displayed, as shown in Fig. 10.

After the user selects the premises, which indicators he wants to view, the system will redirect to the monitoring page of this room.

Fig. 11 illustrates the graph view, which is responsible for monitoring two voltage sensors.

For a danger area, a user should specify how often to send notifications. For the warning area, the user should specify how many seconds the data must be in the warning zone, so that the system sends the alert.

Fig. 12 shows the process of setting danger and warning area conditions for the voltage indicators.

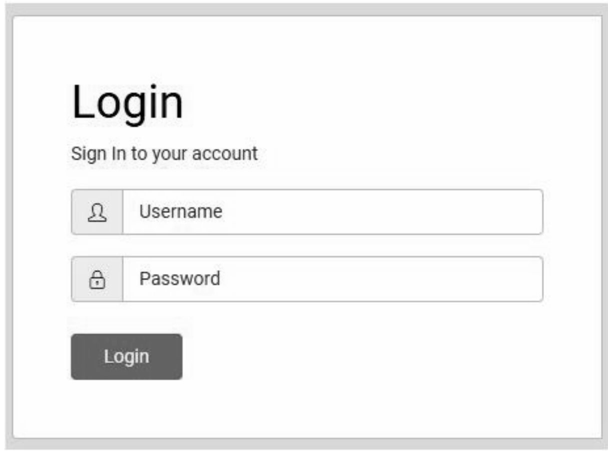


Fig. 9. Login page

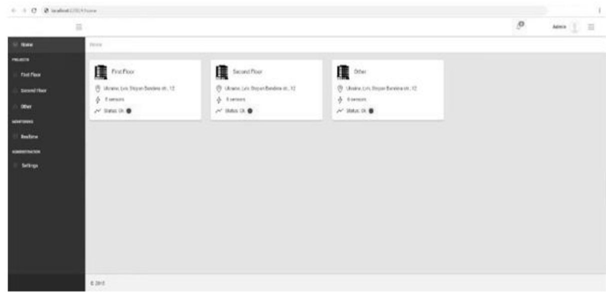


Fig. 10. Main page of the system

As can be seen in Fig. 12, all values, which are greater than 242 and lower than 198, are drawn as a red line, values between 198 and 208, 230 and 242 are drawn as the yellow one. The values in the valid range (from 209 – to 230) are drawn as a blue line.

But what if the user does not sit at the computer at a time when the sensor is out of the allowed limits. In order for the user to find out about the problem, a notification system must be implemented (SMS, email, web-notification). After setting the conditions, the user should determine the method of notification for that case. It is shown in Fig. 14

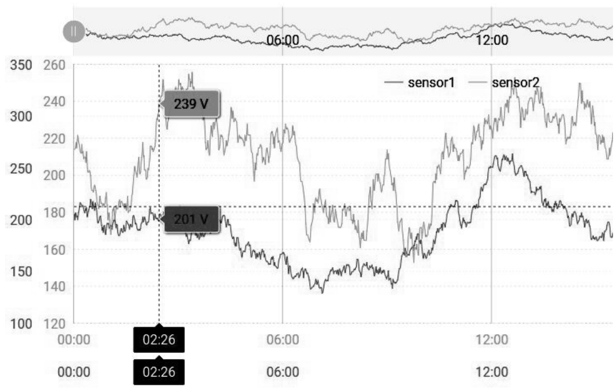


Fig. 11. Indicators of two voltage sensors

Fig. 15 shows the Error web-notification when the sensor output exceeds the allowed range.

Fig. 16 shows the Warning web-notification when the sensor output exceeds the allowed range.

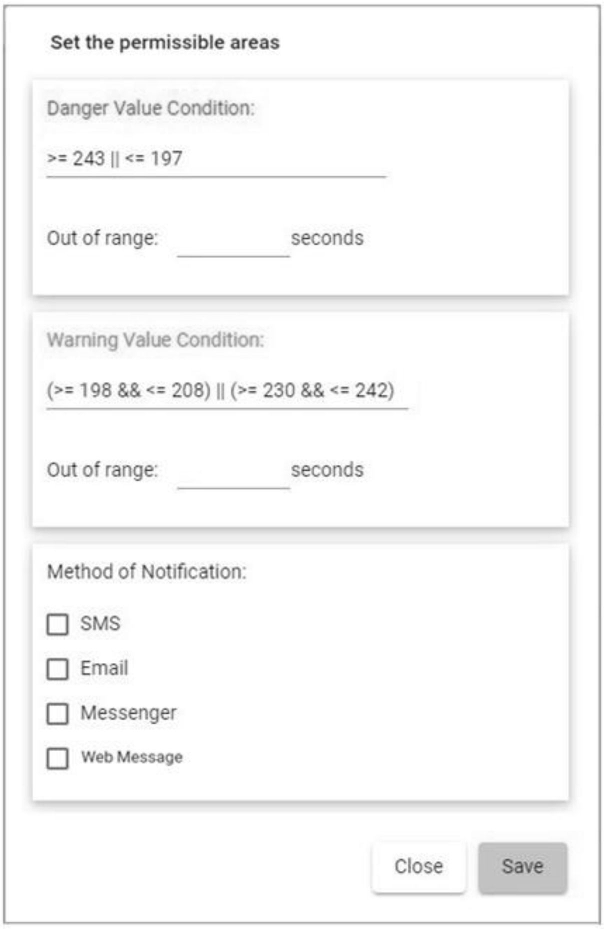


Fig. 12. The process of setting conditions to the sensor

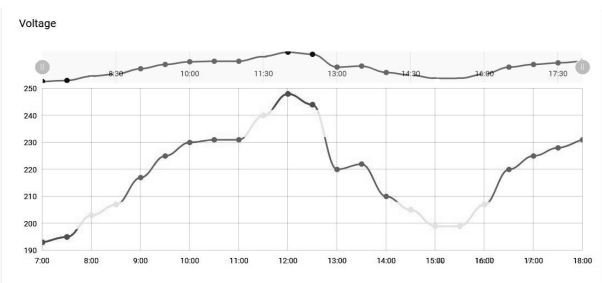


Fig. 13. The result of permissible value

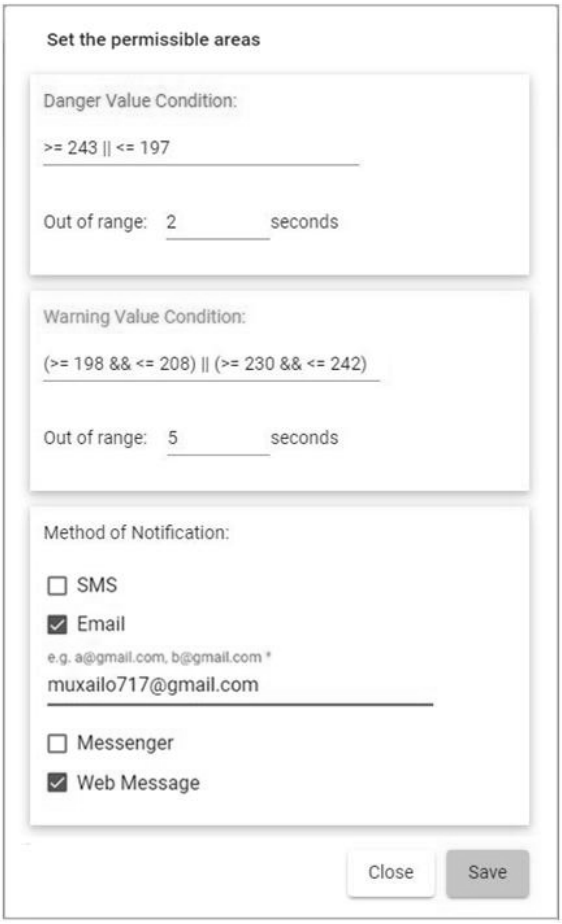


Fig. 14. Methods of notifications

Fig. 15. Web-notification with error

Fig. 16. Web-notification with warning

VI. CONCLUSIONS

The concept of the Internet of things, principles of functioning and technology for its implementation were considered and chosen as the technology for implementing monitoring, processing and displaying system for using in smart house. The client-server architecture was chosen for the proposed system.

REFERENCES

[1] David A. Patterson and John L. Hennessy. Computer Organization and Design. The Hardware / Software Interface, Fourth edition, Morgan Kaufmann Publishers, 2009, 940 p.

[2] The Internet of Things (MIT Press Essential Knowledge series) Paperback – March 20, 2015.

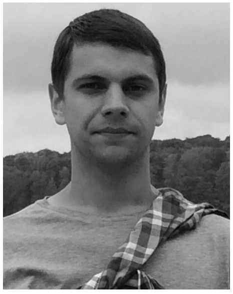
[3] https://www.iotevolutionworld.com/iiot/articles/434018-impact-internet-things-the-telecommunication-industry.htm

[4] Pethuru Raj; Anupama Raman; Harihara Subramanian. "Architectural Patterns". December 2017.

[5] Samuel Greengard. "The Internet of Things". – March 20, 2015.

[6] Michael Stanton Young, Cathy Young. "Smart Home: Digital Assistants, Home Automation, and Internet of Things". 2018. – 314 p.

Ruslan Bachynskyy was born in Lviv, Ukraine. He obtained his Ph.D. degree in Computer Sciences at Lviv Polytechnic National University in 2008.



Stefanivskyi Mykhailo was born in 1995 in Lviv, Ukraine. He received the B.S. degree in Computer Engineering at Lviv Polytechnic National University in 2017. Now he gets a Master's degree.