UDC 681.5:004.77
DOI: 10.25140/2411-5363-2019-3(17)-146-154

Róbert Rákay, Alena Galajdová

COMPARISON OF COMMUNICATION PROTOCOLS FOR SMART DEVICES

The urgency of the research. *Modern trends in the automation focus on the implementation of new communication protocols, wireless data transfer and reduced costs. The communication part of every automation system is crucial, whether it is in the home or industry.*

Target setting. *During the design of the automated systems and the connection of different devices solution, developers have to address different requirements as addressing, data rates, data security, etc. The newest communication protocols and data transfer technologies provide significant data rate and MCU load reduction.*

Actual scientific researches and issues analysis. *To prepare this paper, different free available datasheets and experimental solutions were analyzed as well as conclusions of our previous and other ongoing experiments were used to create the knowledge base about this research topic.*

Uninvestigated parts of general matters defining. *There are many different communication solutions and every manufacturer of communication device provides its best solution. Not all of them can be described in this article.*

The research objective. *The different communication technologies were analyzed for future implementation to a smart devices for home automation, in this article.*

The statement of basic materials. *To propose a future model of home automation system, it is necessary to implement the newest communication technologies. Using the latest communication protocols, such as MQTT, CoAP or Websocket provides a good basis to solve this issue.*

Conclusions. *The proposed paper provides possibilities of the communication for a smart devices of the home automation system. Compared communication protocols have different advantages and disadvantages. The tested protocols meet the communication requirements for home automation devices.*

Keywords: *communication protocols; smart devices; home automation.*

Fig.: 9. Tables: 2. References: 13.

Introduction. Internet of Things is the term which is used to describe any form of applications that connect and make things – devices to interact through the internet. These devices and “their” network can be divided into Consumer Internet of Things (CIoT) and Industrial Internet of Things (IIoT). These classes of IoT share the same architecture composed of:

- Data collection
- Data storing
- Analysis of data
- Share of information

The IoT for customers or simply IoT represents the consumer oriented applications where all the devices are working to fulfil the needs of the consumer. Typical representatives of this class are smart devices of home automation systems, for example: refrigerator, washer, dryer, personal gadgets such as fitness sensors, smartwatches, etc.

The typical data volumes and rates are relatively low for these systems. The gathered data represent temperature, air pressure, number of steps and other information. The majority of applications are not mission or safety critical and the failure of the devices won´t cause any harm, maximally financial or comfort [1].

The Industrial Internet of Things IIoT (Fig.1) represents the industrial applications of interconnected devices which work together [2].

Typically devices operate in industrial grade transport systems, energy production or distribution and also medical environment, etc. The data volumes and transmission rates are relatively higher than in standard IoT. The majority of applications are critical in meanings of goals, aims and safety. Failure or bigger delay can cause great damage, both financial and environmental, e.g. failure of smart grid has a severe impact on the life of people and economy, the errors of intelligent traffic system can threaten drivers and walkers too. The majority of IIoT applications are system centric [3].

Fig. 1. Typical IIoT system

INDUSTRY 4.0 is the term describing the increased integration of information and communication technologies into production systems. According to the three leading German associations of mechanical engineering, information, communication and electrical industry, Industry 4.0 aims for optimization of value chains by implementing an autonomously controlled and dynamic production. With this integration better customization and individualization is carried out [4, 5].

The minimal requirements to determine the suitability of protocols are based on the shown automated monitoring system. (Fig. 2) The main goal of the system is to collect the temperature and humidity in three different room, and in case of emergency alarm the people are warned in the room through a signaling device, smartphone, tablet or notebook.

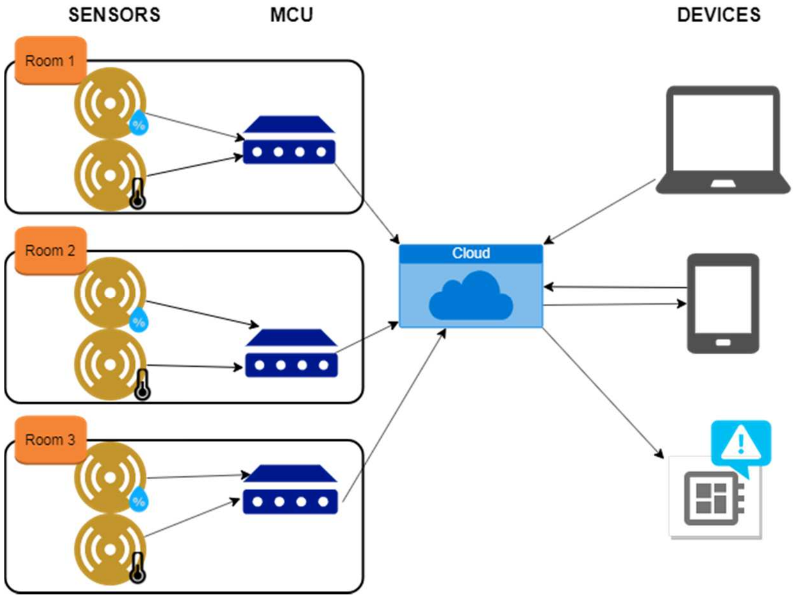

Fig. 2. Automated monitoring system

The collected data is processed from digital values to temperature and humidity in the microcontrollers and after that is sent to the cloud platform. The message body will consist of address and measured value. The values should be collected every 1 hour. The data is represented as numbers with two decimal places. Because of the experimental nature of the monitoring system the data collection is not time critical and the data rates are based on the available wireless connection to a local Wi-Fi router. For this system the measured values are not sensitive or private data therefore the security is not a critical aspect. The data collection will be carried out on one floor of an administrative building where all three rooms are adjacent. The aim of the experiments is not to test the technical characteristics of the standardized communication protocols but to find a suitable, easily applicable protocol for the proposed automation system.

Data processing. The data is collected from the cyber-physical world (CPS). CPS is the instrument to reach the increased automation. The main parts of CPS are microcontroller (MCU), actuators, sensor and communication interface. CPS can work autonomously and co-operate with the production environment.

By integrating CPS to the industrial system a "smart factory" is created. Depending on the type of applications data is being created from sensors or sensor networks for market analysis, web pages statistics, etc. The collected data is stored for on-line and off-line processing.

The stored data is analyzed, statistics are created and short/long term trends are updated. The processed data is shared to the relevant services or applications and subscribers. These systems are supposed to react, display, publish, and store the data. The key requirement in IoT is the efficient and scalable data sharing. The degree of performance depends on the application and sharing platform. Several standards to fulfil the requirements have been proposed to address this key need of the IoT [6].

There are two key aspects in the IoT: the devices and the server side that supports them. In some applications there is a third category, the gateway that supports data aggregation, event processing, bridging. The gateway connects the device to the wider Internet. The connections are based on GPRS connectivity, battery discharging, radio interference, etc, for both types of connections. Typical representatives of devices are embedded controllers of lower and higher classes, such as Arduino, Arduino Yún, and Raspberry Pi.

Some of these devices integrate sensor, some include communication interfaces. The communication between devices, Internet or the gateway is usually carried out by different models (Fig. 2):

- Direct Ethernet or Wi-Fi connectivity, TCP or UDP
- Bluetooth
- Bluetooth Low Energy (BLE)
- Near Field Communication (NFC)
- Zigbee
- SRF and point-to-point radio links
- UART or serial lines
- SPI or I2C wired buses
- ESPNow etc. [7]

There are many protocols to regulate the communication. For example HTTP protocol is very important for many devices, and a simple controller can create request such as GET and POST to read or write data to other device.

Although the HTTP is supported by many devices and is well known protocol, it has some disadvantages such as the size of the overhead, big requirement on memory size and power requirements. In order to fulfil needs of IoT we need simpler, smaller communication protocols.

Other requirements for IoT devices are:

- The ability to disconnect a rogue or stolen device.
- The ability to update the software on a device.
- Updating security credentials.
- Remotely enabling or disabling certain hardware capabilities.
- Locating a lost device.
- Wiping secure data from a stolen device.
- Remotely configuring Wi-Fi, GPRS or network parameters [8].

Testing of communication protocols. For the testing of basic communication properties we used the MCU ESP 32 and ESP 8266 connected to sensors and LEDs.

The main goal was to find the most suitable easily configurable communication protocol for embedded devices of smart homes and consumer IoT applications. The detailed parameters of MCU are described in the next section.

The ESP32 microcontroller. It supports different applications of different complexity, from simple sensing with one MCU to sensor networks. The MCU integrates two microchips with different operation cycles from 80-240 MHz. The main board includes various peripheries such as: capacitive sensors, hall sensor, communication interfaces SDIO, SPI, UART, I2S, and I2C. Bluetooth, Bluetooth Low Energy and Wi-Fi are available to support wireless communication. Technical details of MCU are described in the Table 1 [9, 10].

Table 1

Technical details of ESP8266 and ESP32

Characteristics	ESP8266	ESP32
MCU	Xtensa Single-Core 32-bit L106	Xtensa Dual-Core 32-bit LX6 600 DMIPS
Frequency	80 MHz	80-240 MHz
Wi-Fi	802.11 b/g/n	802.11 b/g/n
Bluetooth	No	BL v4.2, BLE
SRAM	160 kB	512 kB
Flash	SPI Flash, up to 16 MB	SPI Flash, up to 16 MB
GPIO	17	36
HW/ SW PWM	No/8 channels	1/ 16 channels
SPI/I2C/I2S/UART	2/1/2/2	4/2/2/2
ADC	10 bit	12 bit
CAN	No	1
Ethernet Mac interface	No	1
Touch sensor	No	Yes
Temperature sensor	No	Yes

Programming of devices was conducted in Arduino IDE programming environment. The described MCUs created the base of experiments. With their communication interfaces they have many connectivity options. The EP32 was used as a basis for the MQTT Experiment.

The ESP8266 microcontroller has been used with the Wi-Fi, WebSocket, CoAP protocols. Both development boards are part of the monitoring system, where ESP8266 board is used with HTTP and ESP32 is programmed as a MQTT signaling device.

Communication protocols.

CoAP - Constrained Application Protocol. This is an alternative to HTTP. It is currently applied in various IoT solutions, where the client-server structure is preferred and not the broker as with MQTT. It uses a binary representation of data that is more efficient than HTML or XML format. The formatting protocol uses, among other things, message header compression, resource search, automatic configuration, and other additional features. It supports 4 types of messages: No Confirmation, With Confirmation, Reset, and Only Confirmation [10].

MQTT is a communication protocol for M2M (machine-machine) connection. It is designed as a "very light" communication protocol for sending messages between the publisher and the subscriber with the aim to minimize bandwidth and minimize equipment resource requirements, but with reliability and a certain amount of messaging (Fig.3). These goals have made MQTT an ideal case for internet and M2M protocols, and for mobile applications where bandwidth and battery capacity are limited.

The current implementations are in experimental workplaces that are focused on: location-based messaging, home automation, automation in a laboratory environment, as this protocol is still being developed, In addition to experimental solutions, deployment options are explored in everyday applications like Facebook Messenger, Amazon Web Services, and more [11, 12].

The MQTT protocol works on TCP / IP and provides lossless, two-way communication.

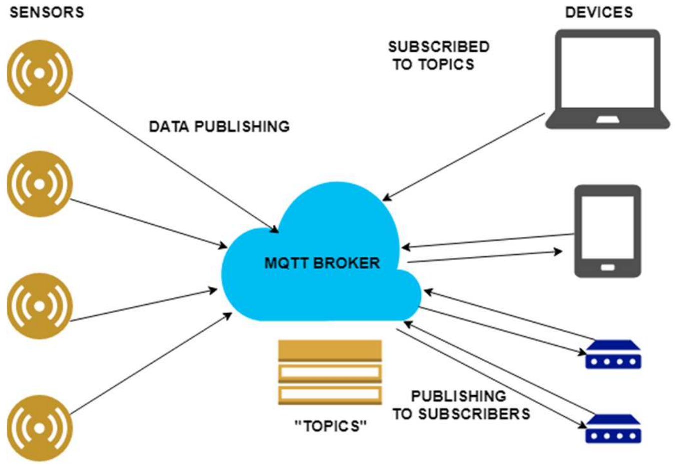

Fig. 3. MQTT communication principle

WebSocket is a protocol that allows communication between the client and the server (endpoint) using the TCP protocol. The advantage of WebSocket communication is a full duplex connection, which allows simultaneous two-way communication over HTTP protocol. Header is smaller than HTTP. The communication cycle is as follows:

1. The client sends the handshake request to the server.
2. The server responds to the request with the next HTTP header, the last in this communication, if the request is received, the server sends an HTTP client message that switches to WebSocket protocol.
3. From now on, communication between the client and the server is open, and they can exchange an unlimited number of messages until the end of the connection [13].

Experiments. The experiments were focused on creating functioning wireless connection with the MCU. Each experiment use a different protocol described in the previous section with aim to find a most suitable, sustainable one for embedded devices in industrial field and home automation systems.

Protocol CoAP. One of the newest protocol for IoT is CoAP- Constrained Application Protocol. CoAP is similar to HTTP (Fig. 4). To test this protocol the ESP8266 board was used with combination of a web tool Cupper. This protocol uses methods PUT, GET, POST, and DELETE to exchange data.

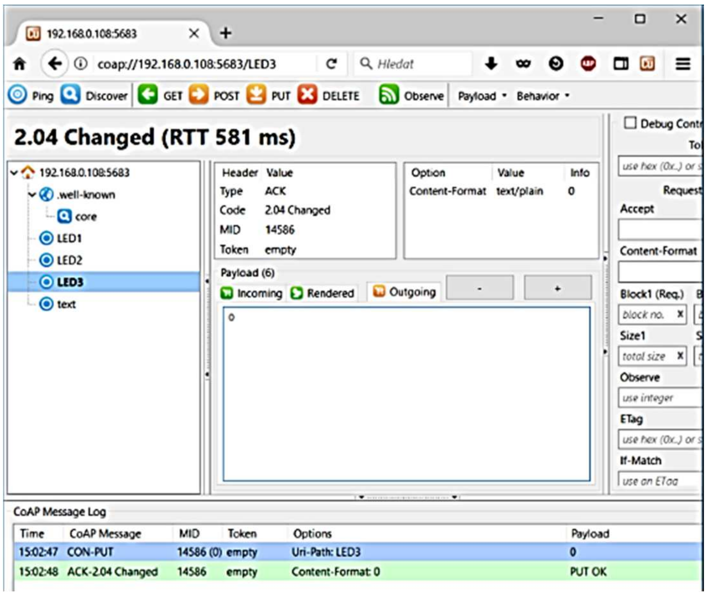

Fig. 4. Preview of Cupper interface during CoAP communication

The communication addressing is based on IP addresses of devices. If the control algorithm process the received data and the callback function includes relevant messages the MCU execute the received orders. In our case acknowledge message and LED light were executed.

The experiment with this protocol showed that "constrained communication" or smaller messages with binary format are suitable for MCU and embedded devices. This protocol is currently under development and not all platforms are supported, which is currently a big disadvantage. The latency during the test was up to 800 ms, which is sufficient for non-time critical applications.

Protocol Websocket. To test this protocol the combination of ESP8266 and RGB LED was used. ESP board creates webserver, an internal web page with 3 sliders to control the colors of LED. Individual sliders represent values for each color of RGB, carried as 3 hexadecimal numbers. This type of communication is suitable for wireless controllers of remotely controlled devices.

Through this continuous connection we are able to control the color of RGB without waiting or interrupting the connection. Preview of control program and received messages are shown on the Figure 5.

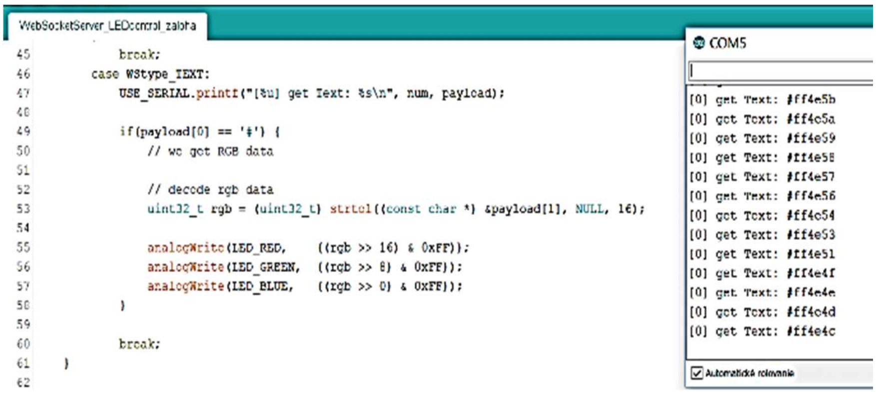

Fig. 5. Preview of WebSocket communication

It is a two-way (full-duplex) communication; the RGB color is continuously controlled with the sliders. This protocol is not supported by the majority of current platforms. A preview of running WebSocket connection is shown on the Figure 6.

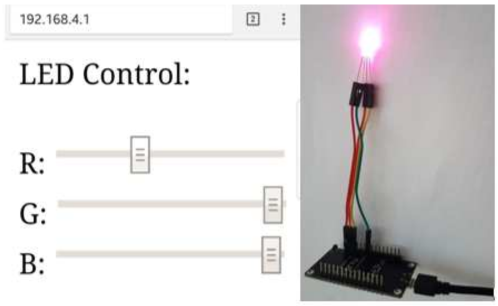

Fig. 6. Preview of Webserver and controlled RGB LED

HTTP and MQTT protocols. The last test included a combination of both EPS boards and sensors. The created system collects data from its environment with photo-resistors, temperature sensors, humidity sensors and accelerometers. Monitoring of the environment was carried out every 15 minutes. The collected data are then transferred to cloud to store and visualize data. Logical structure of the system is shown on the Figure 7.

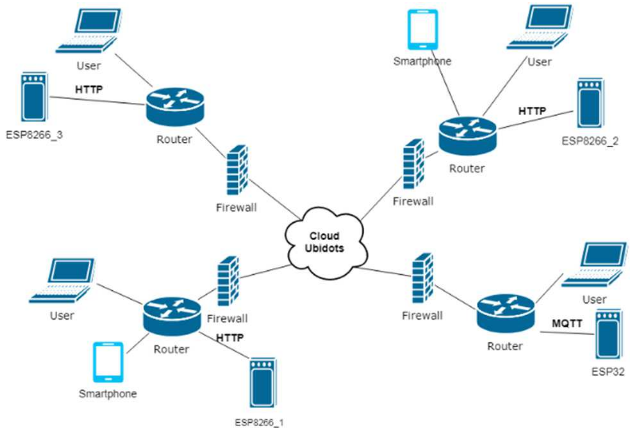

Fig. 7. Logical structure of monitoring system

Preview of the monitoring system with HTTP and MQTT communication is shown below (Fig.8). The collected data are visualized in graphs, tables and numeric widgets.

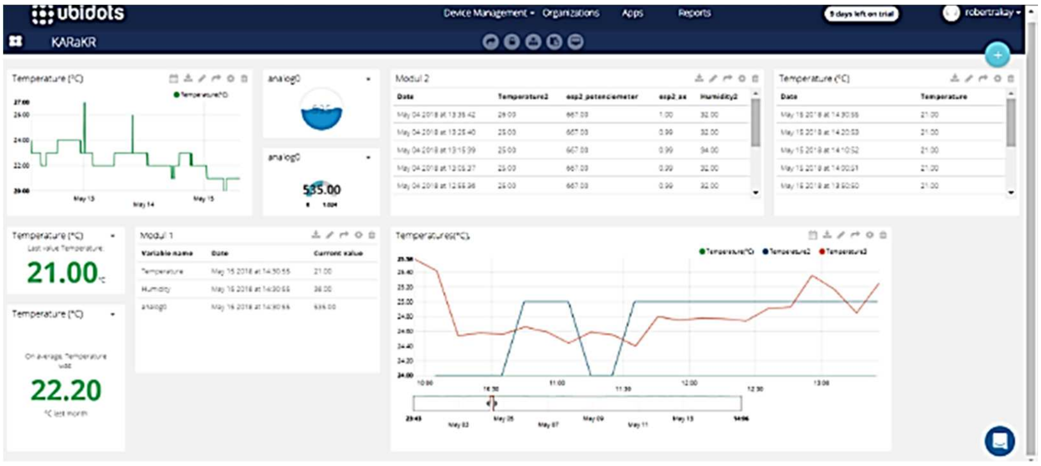

Fig. 8. Preview of monitoring systems interface

The Ubidots cloud provides MQTT broker to control the communication, which exchange messages between the publisher and the subscriber with the relevant topics. Messages represent the last value of variable, to which is subscribed at least one subscriber. Publisher send data with method PUBLISH, and adds values to TOPIC, basically the name of a variable or information of a sensor. The Figure 9 shows the received messages in the EPS board. Messages control a connected outputs, in this case LEDs on/off states.

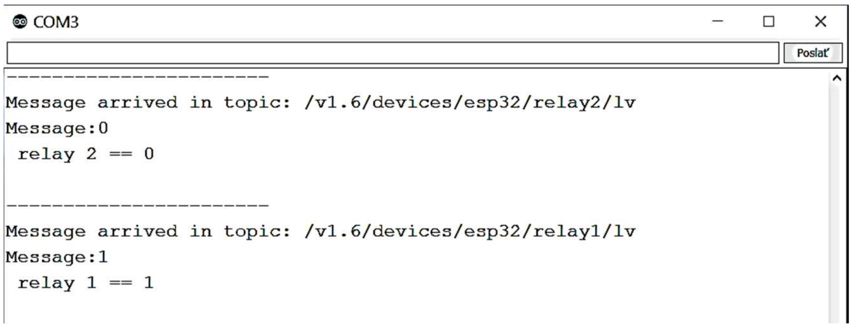

Fig. 9. Preview of messages from the monitoring system

These tests showed that with an integrated MQTT broker, it is well prepared for integration in to IoT systems of nowadays. For more advanced devices like Raspberry Pi, which have advanced technical features, it is easy to use HTTP protocol but for embedded devices with limited computing capacity MQTT is more suitable.

Finally, a comparison of the tested communication protocol is shown in the Table 2.

Table 2

Comparison of tested communication protocol

	HTTP	**MQTT**	**CoAP**	**WebSocket**
Transfer	TCP	TCP	UDP	TCP
Mechanism	Request/Response	Publish/Subscribe	Request/Response	Publish/Subscribe
Designation	Smart Energy, home automation	Remote access	Local networks and communication	Smart Energy, home automation
Quality		QoS 0-1-2	With acknowledgement/ Without	
Architecture	Client/ Server	Publisher/Subscriber	Publisher/Subscriber with restrictions	WebSocket Server/Client
Message size	big	small	small	„infinite“
Format	ASCII	Binary	Binary	
Data distribution	1 to 1	1 to 0/1/N	1 to 1	1 to 1
Security	No, HTTPS for safe comm.	SSL/TLS	Less safe as HTTP (UDP)	SSL/TLS

It summarizes the main characteristics of protocols. Even all protocols are different in some meaning, they are suitable to be used in the IoT area.

Conclusion. In this paper, a comparison of selected IoT communication protocols is described. The communication protocol CoAP is unique in the size of the transferred messages but has its limitations in the currently supported platforms. CoAP will find its enforcement in the embedded devices of future automation system in commercial applications. The alternative protocol WebSocket creates a two-way continuous connection within its special solution. Currently there is no exact field for this protocol, but in systems with continuous control there are many applications which could be controlled with this type of communication. In the last experiment, we tested HTTP and MQTT protocols. The HTTP protocol burdens the hardware and for this reason it´s not suitable for embedded devices of IoT. The MQTT with publish/subscribe and Broker/topic system offers a very suitable communication form for IoT systems. For its latency (time delay) this protocol is currently intended for solutions where time is not critical.

Finally the carried out experiments showed that for integrating MCU to home automation system the most suitable protocol is the MQTT. With its message size and transfer rate provides enough to transfer information like temperature, pressure, presence or other information.

Acknowledgement. This work has been supported by the Slovak Grant VEGA 1/0330/19 - Research and design of algorithms and systems for the fusion of heterogeneous data in multi-sensor architectures and by H2020: Manufacturing Industry Digital Innovation Hubs (MIDIH), reference no. 767498.

References

1. Ray, P. P. (2018). A survey on Internet of Things architectures, Journal of King Saud University - Computer and Information Sciences, Volume 30, Issue 3, pp. 291-319, [online]. ISSN: 1319-1578. Available from: https://www.sciencedirect.com/science/article/pii/S1319157816300799?via%3Dihub.

2. Miao Wu ; Ting-Jie Lu ; Fei-Yang Ling ; Jing Sun ; Hui-Ying Du (2010). Research on the architecture of Internet of Things, 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), Chengdu, 2010, pp. V5-484-V5-487. [online] ISBN: 978-1-4244-6542-2. Available from: https://ieeexplore.ieee.org/abstract/document/5579493.

3. Bednar, S., Modrak, V. (2014). Mass customization and its impact on assembly process' complexity, International Journal for Quality Research, 8(3), pp. 417-430. 20.[online] ISSN 1800-6450 Available from: http://www.ijqr.net/journal/v8-n3/10.pdf.

4. Carruthers, K. (2016). Internet of things and beyond: cyber-physical systems, [online]. Available from: https://katecarruthers.com/2016/04/internet-of-things-and-beyond-cyber-physical-systems/.

5. Šimšík, d., Galajdová, A., Šeminský, J., Rákay, R., Višňovský, M. (2016). Inovácia študijného programu automatizácia a riadenie strojov a procesov v reakcii na požiadavky priemyselnej praxe [Innovation of the study programme of automation and the machine and systems control regarding the demands of industrial practice]. ARaP. Prague: MM publishing, pp. 79-82. ISBN 978-80-906310-1-4.

6. Anawar, M. R. et al. (2018). Fog Computing: An Overview of Big IoT Data Analytics, *Wireless Communications and Mobile Computing,* Volume 2018, Article ID 7157192, Available from: https://www.hindawi.com/journals/wcmc/2018/7157192/.

7. SUMEDHA, R. Communication without and with gateway. Available from: https://www.slideshare.net/wso2.org/wso2con-eu-2017-building-smart-connected-products-with-wso2-iot-platform/.

8. Ray, P. P. (2016). A survey of IoT cloud platforms, Future Computing and Informatics Journal, Volume 1, Issues 1–2, pp. 35-46, [online]. ISSN: 2314-7288. Available from: www.sciencedirect.com/science/article/pii/S2314728816300149/.

9. ESP8266. Datasheet (2019). [online], Available from: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.

10. ESP32 Datasheet [online], Available from: https://www.espressif.com/sites/default/files/documentation/esp_wroom_32_datasheet_en.pdf/.

11. Shelby, Z., Hartke, K., Bormann, C. (2-14). The Constrained Application Protocol (CoAP) 6/2014[online]. ISSN: 2070-1721 Available from: https://tools.ietf.org/html/rfc7252/.

12. MQTT v3.1.1 (2015). OASIS Standard [online]. Available from: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.

13. Srinivasan, l., Scharnagl, J., Schilling, K. (2013). Analysis of WebSockets as the New Age Protocol for Remote Robot Tele-operation, IFAC Proceedings, Volumes, Volume 46, Issue 29, pp. 83-88, [online]. ISBN 9783902823564, Available from: https://www.sciencedirect.com/science/article/pii/S1474667015343688.

УДК 681.5:004.77

Роберт Ракай, Алена Галайдова

ПОРІВНЯННЯ ПРОТОКОЛІВ ЗВ'ЯЗКУ ДЛЯ SMART ПРИСТРОЇВ

Актуальність теми дослідження. *Сучасні тенденції автоматизації зосереджені на впровадженні нових протоколів зв'язку, бездротовій передачі даних та зменшенні витрат. Комунікаційна частина кожної системи автоматизації є вирішальною, будь то в домашніх умовах чи в промисловості.*

Постановка проблеми. *Під час проектування автоматизованих систем та під'єднання різних пристроїв розробникам доводиться задовольняти різні вимоги, такі, як адресація, швидкість передачі даних, безпека даних тощо. Найновіші протоколи зв'язку та технології передачі даних забезпечують значне скорочення як швидкості передачі даних, так і завантаження мікроконтролера (MCU).*

Аналіз останніх досліджень і публікацій. *Під час підготовки даної роботи була проаналізована різноманітна документація у вільному доступі та експериментальні рішення, а також зроблені висновки з наших попередніх та інших проведених експериментів для створення бази знань за темою дослідження.*

Виділення недосліджених частин загальної проблеми. *Існує багато різних комунікаційних рішень, і кожен виробник пристрою зв'язку пропонує найкраще рішення. Не всі вони можуть бути описані в цій статті.*

Постановка завдання. *У даній статті проаналізовані різні комунікаційні технології для подальшої реалізації у смарт-пристроях для домашньої автоматизації.*

Виклад основного матеріалу. *Щоб запропонувати майбутню модель системи домашньої автоматизації, необхідно впровадити новітні комунікаційні технології. Використання останніх протоколів зв'язку, таких як MQTT, CoAP або Websocket, дає хорошу основу для вирішення цієї проблеми.*

Висновки відповідно до статті. *Ця стаття пропонує можливості комунікації для розумних пристроїв системи домашньої автоматизації. Порівняні протоколи зв'язку мають різні переваги та недоліки. Перевірені протоколи відповідають вимогам зв'язку для пристроїв домашньої автоматизації.*

Ключові слова: *протоколи зв'язку; старт-пристрої; домашня автоматизація.*

Рис.: 9. Табл.: 2. Бібл.: 13.

Róbert Rákay – Assistent Professor, Ph.D. of technical sciences , Technical University of Kosice (Letná 9, 04200, Košice, Slovak Republic).
Роберт Ракай – кандидат технічних наук, Технічний університет Кошице (Letna 9, 04200, Košice, Slovak Republic).
E-mail: robert.rakay@tuke.sk
ORCID: https://orcid.org/0000-0002-7151-3749
Scopus Author ID: 56922070700

Alena Galajdová –Head of the Department, associate professor, Ph.D. of technical sciences, Technical University of Kosice (Letna 9, 04200, Košice, Slovak Republic).
Алена Галайдова – кандидат технічних наук, Технічний університет Кошице (Letna 9, 04200, Košice, Slovak Republic).
E-mail: alena.galajdova@tuke.sk
ORCID: https://orcid.org/0000-0003-0128-4191
Scopus Author ID: 6506796741