

# IMPLEMENTATION OF IOT TECHNOLOGY FOR DATA MONITORING VIA CLOUD SERVICES

**Veronika M. Antonova,**

*Moscow Technical University of Communications and Informatics, Moscow, Russia;  
IRE RAS, Moscow, Russia, [xarti@mail.ru](mailto:xarti@mail.ru)*

**Elena E. Malikova,**

*Moscow Technical University of Communications and Informatics, Moscow, Russia, [emalikova@gmail.com](mailto:emalikova@gmail.com)*

**Alexey E. Panov,**

*Moscow Technical University of Communications and Informatics, Moscow, Russia, [pnvalex@mail.ru](mailto:pnvalex@mail.ru)*

**Igor V. Spichek,**

*Moscow Technical University of Communications and Informatics, Moscow, Russia, [spichekig@gmail.com](mailto:spichekig@gmail.com)*

**Andrey Y. Malikov,**

*Moscow Technical University of Communications and Informatics, Moscow, Russia, [amalikov2000@gmail.com](mailto:amalikov2000@gmail.com)*

DOI: 10.36724/2072-8735-2021-15-1-46-53

**Manuscript received** 26 October 2020;

**Accepted** 20 December 2020

**Keywords:** Internet of Things, MQTT protocol, publisher, subscriber, broker, ThingSpeak, Microcontroller ESP8266, cloud service

An operating device has been designed for long term aggregating, storing and visualizing climate records with a view to their further publication in a cloud service. In order to address the given problem a number of technical issues related to the device concept development and its operating algorithms were solved. The device runs using the MQTT protocol and a microcontroller unit based on ESP8266 chip which is designated for the application in Internet of Things (IoT) devices. The designed system is based on open-source software and allows providing access to the received data to all authorized users. The system is expanded easily since the number of attached sensors and peripheral units can change and the program can be transformed so as to solve emerging tasks. The ability to connect the Internet from any access point provides the mobility for the device and permits to make measurements within the range of a Wi-Fi network. In some instances, it is convenient to use smart phones or tablets that have access to the Internet via cellular networks for research and scientific experiments. In this case, mobile devices can act as monitors to control the system operation. This device can be useful for carrying out research work when data collection over a long period of time and long-term storage of information with the possibility of its further processing are essential. The examples are automatic monitoring of the equipment, medical supervision of patients' health or gathering and processing of various climate parameters. Undergraduate students can also make use of the developed device when studying IoT technology.

## Для цитирования:

Антонова В.М., Маликова Е.Е., Панов А.Е., Спичек И.В., Маликов А.Ю. Реализация технологии IoT для мониторинга данных через облачный сервис // Т-Comm: Телекоммуникации и транспорт. 2021. Том 15. №2. С. 46-53.

## For citation:

Antonova V.M., Malikova E.E., Panov A.E., Spichek I.V., Malikov A.Y. (2021) Implementation of IoT technology for data monitoring via cloud services. T-Comm, vol. 15, no.2, pp. 46-53. (in Russian)

## Introduction

Nowadays there is a large number of systems for remote data control and display on the information technology market. Most of them are quite capable of coping with industrial tasks, but cannot always be applied to small tasks because of uneconomical prospects for the installation of cumbersome software and hardware systems. The work presents the developed functioning system for aggregating climatic parameters (humidity, air temperature, atmospheric pressure, luminous intensity). These parameters are sent to the cloud server for storage or further processing. The structural diagram of this device is given as well as its connection to the cloud service and its basic settings are described. The relevance of the development of this device stems from the growing need to use IoT technology and its application in all areas of human activity [1].

## Internet of Things Features

### A. MQTT Protocol Operation in IoT Technology

The development of successful IoT applications [2] implies several tasks: mobility, reliability and scalability. The first task derives from the fact that the change of IoT device location results in the change of its IP address and, therefore, necessitates that the work of routing protocols be organized. While ensuring its high reliability, the system should collect and transmit data from different devices without disrupting its operation.

The requirement for scalability originates from the possibility of increasing the number of IoT devices connected to the network. In the general case, the following model of device interaction in the Internet of Things is adopted. Terminal devices, detectors, sensors interact with each other (the so-called D2D or Device to Device interaction). Information aggregated by the devices is sent to a server for further analysis and processing (D2S or Device to Server interaction). The server may comprise several computing machines or objects which also need to communicate with each other (S2S or Server to Server interaction) [2].

At the moment there is a variety of different types of application layer protocols being used for IoT device interaction. The most common protocol used in IoT technology today is MQTT.

**MQTT (Message Queuing Telemetry Transport)** is a network protocol for messages between devices implementing the Master/Slave model that are transported over low bandwidth networks [3]. It was developed by IBM, which has its own service, IBM Cloud, for building IoT networks. The protocol operates on top of TCP/IP stack that provides a simple and reliable data stream. MQTT is aimed at stable transmitting information aggregated from a large number of devices. It's a credible transport for the IT infrastructure. The protocol is designed for a large network of small devices that need managing from the cloud. It is utilized for "multicast" data transmission to many receivers as well. The MQTT protocol is extremely easy to use; it implies several control options. An example of this protocol functioning is oil pipeline monitoring for leaks or acts of vandalism. The protocol allows transferring the information which is received from a thousand of sensors and is needed to be concentrated in one place to be analyzed. When the monitoring system detects a problem, it is able to take actions to address it.

There are several reasons why the MQTT protocol was chosen for the implementation of a device for climate data collec-

tion. Firstly, it is aimed at efficient handling of multiple small requests, which fully corresponds to the algorithm of the designed device for collecting climatic parameters. Secondly, it is well documented and doesn't require large processor capacities for data transfer organization. Thirdly, a sufficient number of libraries for microcontroller platforms have been created for the protocol.

MQTT represents an application layer protocol of the OSI model. It runs over the TCP/IP protocol stack. MQTT is a protocol for exchanging messages according to the Publisher/Subscriber model [3]. Information exchange via this protocol is illustrated in Fig. 1.

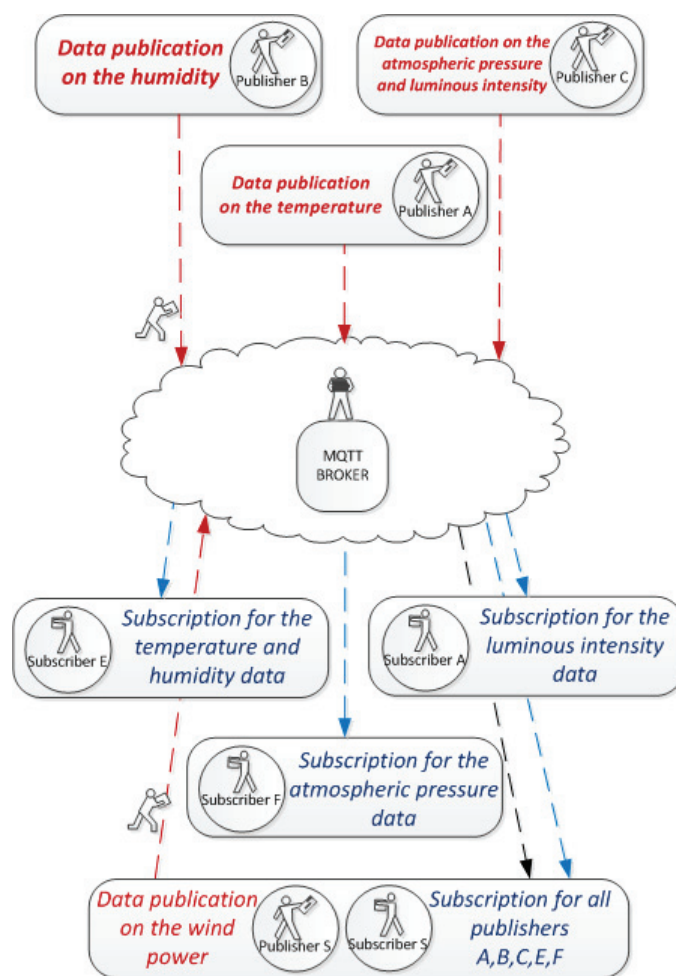


Fig. 1. The scheme of interaction between the participants of data exchange in the MQTT ideology

The communications system based on the MQTT protocol consists of a publisher server, a broker server and some clients. Every client publishing a message for the broker incorporates a subject into the message. The subject represents routing information for the broker. Every client wishing to receive messages subscribes to a specific subject, and the broker delivers all the messages with corresponding subjects to the client. Therefore clients don't need to know each other, they communicate only on the subject. Such kind of architecture allows to expand the solutions not considering the relations between data sources and their consumers.

The MQTT protocol can be used for telemetry and remote monitoring of various objects. This makes it possible to receive data from different devices when some events occur. The exchange of messages in the MQTT protocol is carried out between a client, a subscriber or the broker of messages. The publisher sends information to the broker indicating a certain subject in the message. Subscribers can receive different data from multiple publishers according to the subscription for corresponding subjects. The MQTT protocol features are described in detail in the specification [3].

Thus, the MQTT protocol is one of the most commonly used ones in the IoT sphere with the support for almost all cloud services. The MQTT provides a scalable and efficient Publisher/Subscriber model for data transmission. Also, the protocol does well in networks with unstable level of signal reception. The minimum amount of service information, the availability of classes of service and the hierarchical structure are the advantages of MQTT. This is confirmed by a great variety of both client and server open source software. Moreover, this protocol was specially developed for data transferring from devices of low power consumption. It is aimed at the minimal amount of time needed for data exchange between the devices since the time spent for transmitting information between the devices affects the duration of autonomous device functioning.

### B. Overview of IoT Service for Data Receiving and Processing

**ThingSpeak** [4] service is one of the possible options for organizing a data aggregation device. In addition to the HTTP protocol, it supports transmitting via the MQTT protocol. ThingSpeak provides free access for non-commercial projects with a limited number of channels and some frequency of data updates. Messages cannot be received more often than once every 15 seconds. It is a kind of protection from exceeding the limit of 8,200 messages a day.

The principal features of ThingSpeak include aggregating data issued in real time by a publisher, storing it on a long-term basis and their further in-depth analysis made by the power of MATLAB integrated to the service. The analytics can also be carried out by means of third-party applications. Data export is provided through the ability to store information arrays in the CSV database format.

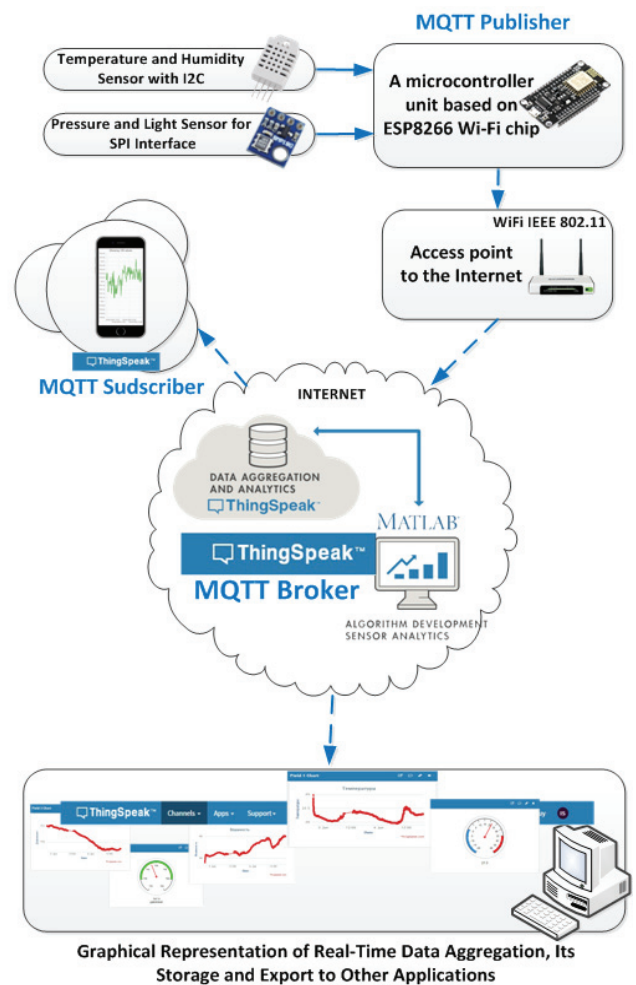
ThingSpeak allows using various statistical methods for data processing as well as programming actions when channel data meets a certain condition. This is implemented by programming algorithms in Python. The channels in which data is sent for storage and visualization are central for the ThingSpeak platform. The name of each channel is unique. It is comprised of eight fields for any type of data.

After registering for an account in ThingSpeak it is possible to create up to four channels, each of which contains eight data fields. Once a channel is created, you can send data to it, process and provide access to it through third-party applications by means of GET or POST requests. Thus, ThingSpeak is the service of choice for the implementation of the given task.

## III. object Implementation

### A. Designing the Block Diagram for the Data Aggregation Device

The MQTT Broker is a central node which is responsible for receiving and sending all messages between publishers and subscribers. A client is any device that connects to the broker and is able to publish its messages or subscribe for subjects to access information. Every subject contains routing data necessary for the broker, i.e. the broker itself is an “address” like a file path in the file structure. Every client (a Publisher) issues its messages in a certain subject, and every client (a Subscriber) wishing to get these messages is to subscribe for them. The broker delivers all the messages of a certain subject to the specified clients. The block diagram of the interoperation of the climatic data aggregation device being developed with the ThingSpeak service and subscribers is illustrated in Fig. 2.



**Fig. 2.** The block diagram of the interaction of the designed device with the ThingSpeak service and the subscribers

Nodemcu Lua unit was chosen to be the basic part of the designed device; the unit has a 32-bit ESP8266 microcontroller [5] on it, which is shown in Fig. 3.



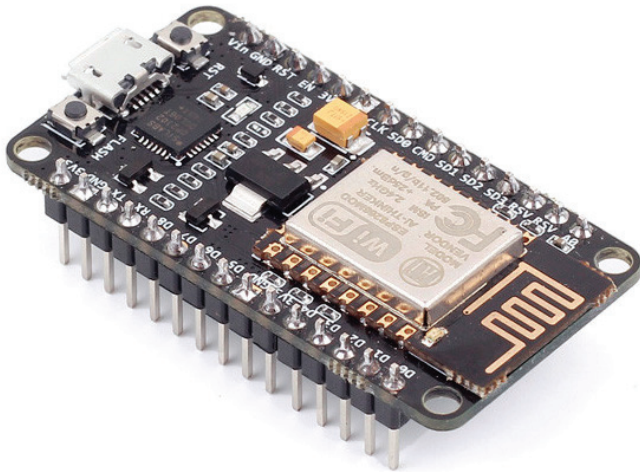


Fig. 3. The physical appearance of the Nodemcu Lua chip

The Nodemcu Lua chip, where the microcontroller is placed, contains all the necessary elements for the power supply and programming of the latter. This platform allows optimizing the process of small scale projects developing in the IoT sphere. According to the manufacturer's specification, this chip combines a microprocessor running at 80MHz and a Wi-Fi unit supporting IEEE 802.11 b/g/n standards. The microprocessor supports the TCP/IP protocols. It has ten available input-output (I/O) ports. The current consumption of this device is up to 215 mA in transmitting mode and 100 mA in receiving mode.

ESP8266 Wi-Fi unit is able to operate in Access Point mode as well as in Transmitting Station mode or in both modes at the same time. The station can be linked to only one access point at a time. Software for the ESP8266 module is written in different programming languages including C++.

Various climatic sensors have the possibility to connect to the module. The block diagram of sensors connecting to the ESP8266 module is illustrated in Fig. 4.

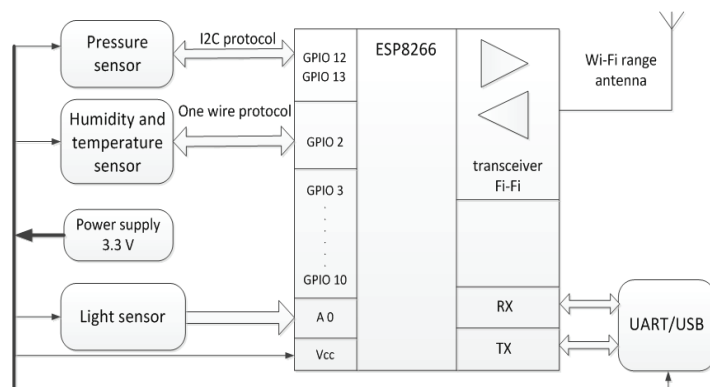


Fig. 4. The block diagram of the data aggregation module

The device is powered by a 3.3V stabilized source. A linear stabilizer located on the chip provides the required voltage to power the ESP8266 microcontroller and sensors. There is also a USB to UART interface converter which is needed for the ability to quickly program the microcontroller. A large number of input-output ports is not employed in the initial stage of the project

implementation. This allows for great potential in increasing the number of external devices and, therefore, addressing additional problems.

### B. Customization of the ThingSpeak Service Application

It is necessary to create information channels and perform their setting on the ThingSpeak service page. This requires the registration process to be completed and an account to be made. After that a new channel and 4 data fields are created in the My Channels tab on the main page of the website. Each channel is assigned an identifier (Channel ID). The latter is necessary to set up a connection of any device to the ThingSpeak database through the Internet. Since four sensors are used, four channels are required for receiving data. Once the ThingSpeak service is configured, it is ready to accept and aggregate data. At the same time, four data arrays will be formed on the cloud service according to the given task. Some information can be added and placed on the page of the service. The information may represent geographic coordinates where the measurements are held, a link to a website page with the project description and some other auxiliary information.

### C. Software Development for the Data Aggregation Unit

The next stage in the system designing is to develop a program algorithm for the microcontroller. It is necessary to survey sensors and to transmit the data received from them to the server in the mqtt.thingspeak.com format set by the MQTT protocol standard. Fig. 5 shows a general flowchart for the program operation. After the power supply is ON and the hardware of the microcontroller has been reset, the process of linking up to a Wi-Fi access point with the SSID parameters and password determined at the stage of program designing starts. When it is detected, the DHCP gateway server assigns an IP address to the device. Upon a successful connection, the system begins to survey sensors cyclically.

Otherwise it waits for setting up a connection. Serial protocol supporting libraries are used to work with sensors. The application of specialized libraries allows to accelerate the designing of a program. Surveying of a luminous intensity sensor is carried out by reading the analogue signal value at the input of an ADC. In case of successful delivery of data to the broker, the information being transmitted is displayed in the current value graphs on the ThingSpeak webpage.

Sensor survey and data publication are made with a given frequency. An interval equal to 20 seconds was set to update the data. When the measured values of such climatic parameters as temperature, atmospheric pressure and humidity are changing slowly, this periodicity is sufficient. Provided the microcontroller with an independent power supply is applied, the implementation of an energy-efficient operation mode is possible. Use of a deep sleep mode for the microcontroller between the sessions of data exchange permits extending the operation of the independent power supply.

### D. Data Visualization in the Cloud Service Application

The ThingSpeak service provides the wide range of options for visualizing the data accepted from sensors. Fig. 6 demonstrates graphs based on the data being accepted by the channels. The four graphs show the dynamics of measured parameters.

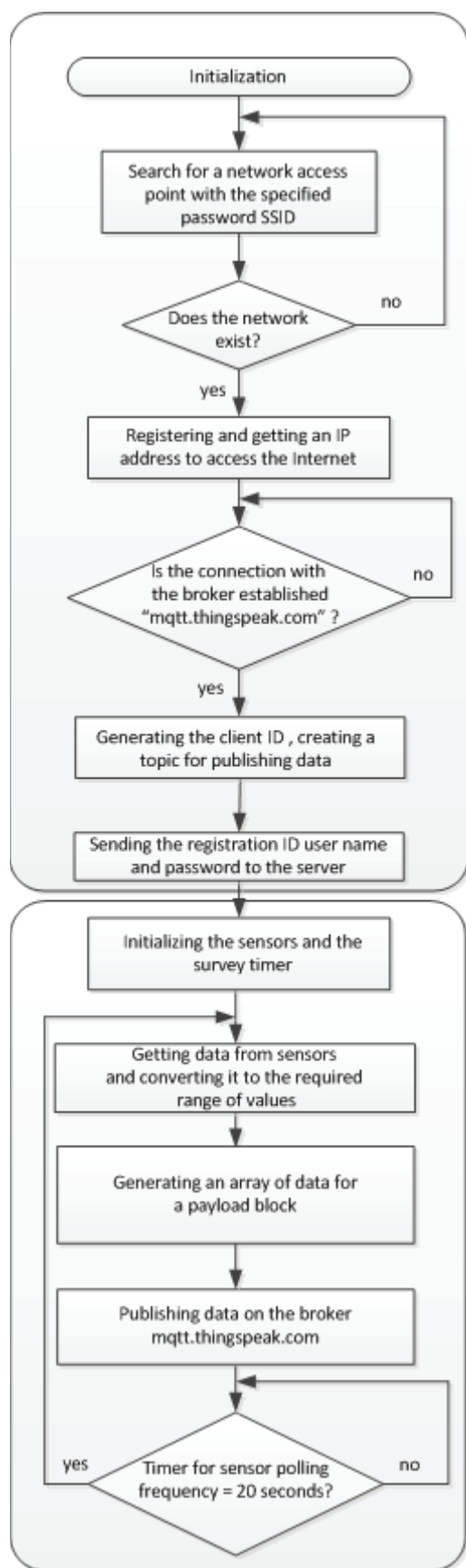


Fig. 5. General flowchart for the program algorithm

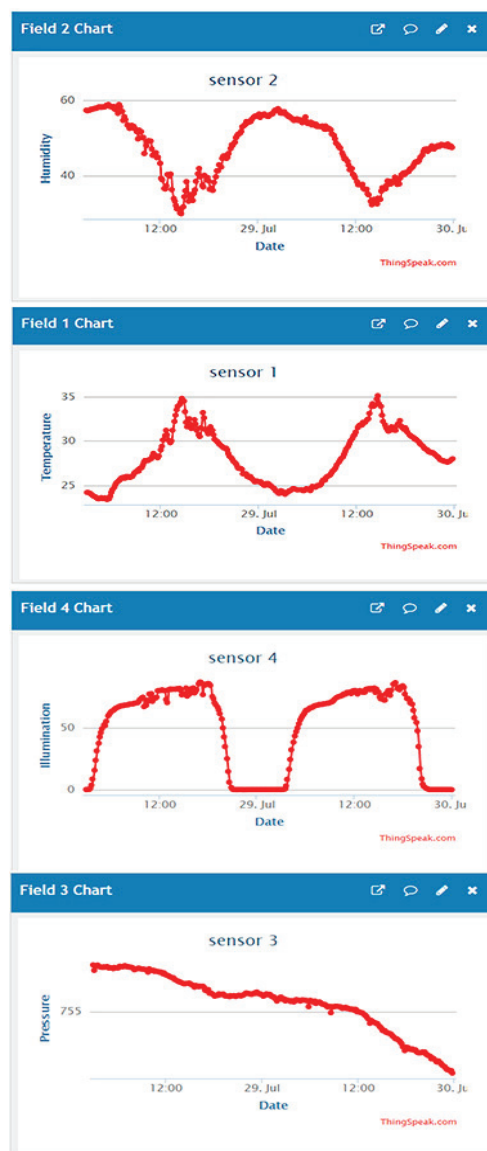


Fig. 6. Data flow visualization in the form of graphs in a 48-hour period

Moreover, the generation of graphs is carried out in real time as data packets arrive. This is a standard form of graphic representation for the results of measurements. In the settings of the graph tab you can select the desired time period and the interval between the values. Close integration of the service with the MATLAB application provides more opportunities for the project and transforms it into a powerful tool for an in-depth analysis of all the data being received.

Despite the novelty of the concept of a device cloud, some applications of this technology go without saying. Devices from “the cloud” will allow real-time reception and interexchange of measurement data accepted from objects and devices located all over the world using widgets and webpages. The program features provide the possibility to add widgets for current value visualization. Variants of displaying current climatic parameter values are presented in Fig. 7. Devices from “the cloud” facilitate standardization of measurements, mutual exchange of tools, information and experimental results.

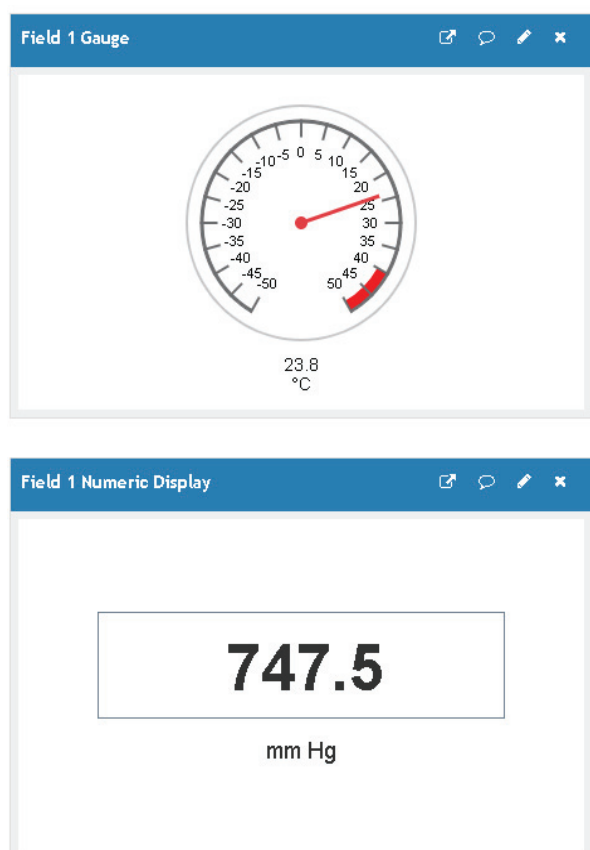


Fig. 7. Variants of displaying current climatic parameter values

The given version of the ThingSpeak service makes it possible to store the accumulated data on the server for a year. Data can be exported to any application for further processing by saving it in the csv format. The data imported can be displayed by third-party analysis and visualization programs [6]. This provides great opportunities for statistical data processing, performing additional calculations and in-depth analysis of aggregated data.

On-line indicator monitoring can be carried out by means of accessing data through mobile applications. For instance, the ThingView mobile application for Android OS allows generating the graphs illustrated in Fig. 8.

The application enables generation of graphs for the selected time period or current value tracking, acting as a subscriber to the data published by the device. To connect the application to the server, it is enough to specify the identifier (ID) of the channel assigned to it while being created.

There are quite a few mobile applications that act as subscribers and allow customizing the interface depending on the type of information displayed. The examples of such applications for Android OS include ThingViewer, MQTT Dash, Mqtt Dashboard, Virtuino MQTT. Each of them maintains the visualization of data in their own graphical interfaces. Moreover, using the ThingSpeak service gives you the possibility to embed graphs into the pages of the site with the ability to scale to build your own monitoring systems.

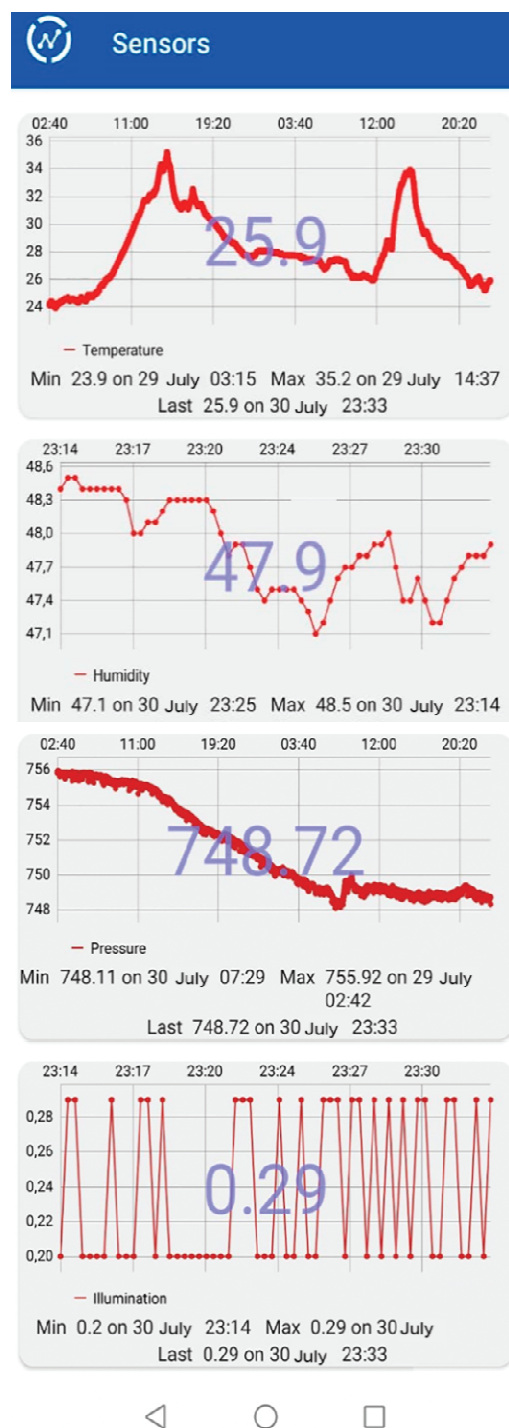


Fig. 8. Graph visualization in the ThingView for Android OS

The ThingSpeak service gathers daily statistics of received data packages. The number of the packets accepted allows assessing the transmission quality and the stability of the data conveyed.



### E. Areas for Further Study

The increase of incoming data flows by connecting additional sensor arrays to the master device will significantly expand the system's capabilities.

The implementation of the given scheme will necessitate the microcontroller to be switched to a mixed Access Point – Client operation mode. It allows connecting up to 9 clients in a Wi-Fi network. The scheme makes it possible to spread sensors over the Wi-Fi network coverage. Such location of sensors is relevant if there are physical obstacles as well as if the measurement point is changed frequently.

GPS module added to every measuring mobile device will maintain the visualization procedure with displaying measurement points in three-dimensional projections. Intermediate calculations and preprocessing of data received from a large number of sensors will optimize the overall traffic from them.

Devices that enable blood pressure, temperature and heart rate of a person to be measured can be utilized as sensors. Then such a device may be employed to control a patient's health at home.

### Conclusion

The operating device was designed to collect data from sensors measuring temperature, humidity, atmospheric pressure, luminous intensity and to send it a cloud service by means of the IoT technology. The system has been proven to be operational in the long-term. A data transfer method with the Publication-Subscription mechanism was tested. Stable transmission of climate data to the cloud server was established, and some graphic relations based on it were modeled.

At present, this device is being used to aggregate data for an in-depth analysis of interdependences of different parameters. The implemented model is a baseline one; it allows performing various functions such as sending text messages, emitting sound signals, and even conveying necessary correcting commands to the sensors. It can act as an information and measurement system which records power consumption at some checkpoints along the spans of power supply lines as well as carries out automatic remote monitoring of the equipment.

The work is partially supported by the Russian Foundation for Basic Research (project №19-07-00525 A - Developing flow-based models of routing problems in telecommunications networks).

### References

1. A. Roslyakov, S. Vanashin, A. Grebeshkov and M. Samsonov (2014), *Internet veshchei* [Internet of things], Roslyakov A. (ed.), PSUTI, OOO AS GARD Publishing House, Samara, Russia. 342 p.
2. V. Tikhvinsky, V. Koval, G. Bochechka, A. Babin (2017), *Seti IoT/M2M: tehnologii, arhitektura y prilozheniya* [IoT/M2M Networks: technologies, architecture and applications]. Moscow: Media Publisher. 338 p.
3. The official site of OASIS (2015), "MQTT Version 3.1.1 Plus Errata 01", available at: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html> (Accessed November 4, 2020).
4. The official site of ThingSpeak (2020), available at: <https://www.thingspeak.com/> (Accessed November 4, 2020).
5. The official site of ESPRESSIF (2020), available at: <https://www.espressif.com/en/products/socs/esp8266> (Accessed November 4, 2020).
6. The official site of Datavisualization.ch (2016), available at: <http://selection.datavisualization.ch/> (Accessed November 4, 2020).

## РЕАЛИЗАЦИЯ ТЕХНОЛОГИИ ИОТ ДЛЯ МОНИТОРИНГА ДАННЫХ ЧЕРЕЗ ОБЛАЧНЫЙ СЕРВИС

**Антонова Вероника Михайловна**, Московский Университет Связи и Информатики, Москва, Россия, [xarti@mail.ru](mailto:xarti@mail.ru)

**Маликова Елена Егоровна**, Московский Университет Связи и Информатики, Москва, Россия, [emalikova@gmail.com](mailto:emalikova@gmail.com)

**Панов Алексей Евгеньевич**, Московский Университет Связи и Информатики, Москва, Россия, [pnvalex@mail.ru](mailto:pnvalex@mail.ru)

**Спичек Игорь Вячеславович**, Московский Университет Связи и Информатики, Москва, Россия, [spichekig@gmail.com](mailto:spichekig@gmail.com)

**Маликов Андрей Юрьевич**, Московский Университет Связи и Информатики, Москва, Россия, [amalikov2000@gmail.com](mailto:amalikov2000@gmail.com)

### Аннотация

Разработано действующее устройство для долговременного сбора, хранения и отображения климатических данных с целью их дальнейшей публикации в облачном сервисе. Для реализации поставленной задачи был решен ряд технических вопросов, связанных с разработкой концепции устройства и алгоритмами его работы. Данное устройство работает с использованием протокола MQTT и микроконтроллера модуля на основе микросхемы ESP8266. Данная микросхема ориентирована для применения в устройствах Internet of Things (IoT). Разработанная система построена на открытом программном обеспечении и позволяет предоставлять доступ к полученным данным всем авторизованным пользователям. Система легко расширяема, поскольку количество подключаемых сенсоров и периферийных модулей может меняться, а программа – модифицироваться для решения новых задач. Подключение к любой точке доступа для выхода в Интернет делает устройство мобильным и позволяет производить измерения в пределах радиуса действия сети Wi-Fi. В некоторых случаях, для проведения исследований и научных экспериментов удобно использовать смартфоны или планшеты, имеющие выход в Интернет через сети сотовой связи. В этом случае мобильные устройства могут выступать в роли мониторов для контроля работы системы. Данное устройство может быть полезным для проведения исследовательских работ, в которых важен долговременный сбор и длительное хранение информации с возможностью ее последующей обработки. Это может быть автоматический мониторинг за состоянием оборудования, наблюдение за состоянием здоровья пациента, сбор и обработка различных климатических параметров. Также разработанное устройство будет полезно для студентов вузов при изучении технологии IoT.

**Ключевые слова:** Интернет вещей, протокол MQTT, датчики сбора информации, облачный сервис, мониторинг, модель передачи данных издатель/подписчик, сервис ThingSpeak, микроконтроллер ESP8266.

### Литература

1. Росляков А.В., Ванышин С.В., Гребешков А.Ю., Самсонов М.Ю. Интернет вещей. Самара: ПГУТИ, ООО "Издательство Ас Гард", 2014. 342 с.
2. Тихвинский В.О., Коваль В.А., Бочечка Г.С., Бабин А.И. Сети IoT/M2M: технологии, архитектура и приложения. М.: Издательский дом Медиа Паблишер, 2017. 320 с.
3. <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>.
4. <http://www.thingspeak.com>.
5. <https://www.espressif.com/en/products/socs/esp8266>.
6. <http://selection.datavisualization.ch>.

### Информация об авторах:

**Антонова Вероника Михайловна**, к.т.н., доцент кафедры "Сети связи и систем коммутации", Московский Университет Связи и Информатики (МТУСИ), Москва, Россия

**Маликова Елена Егоровна**, к.т.н., доц., доцент кафедры сетей связи и систем коммутации, Московский Университет Связи и Информатики (МТУСИ), Москва, Россия

**Панов Алексей Евгеньевич**, к.т.н., доцент кафедры "Сети и системы фиксированной связи", Московский Университет Связи и Информатики (МТУСИ), Москва, Россия

**Спичек Игорь Вячеславович**, Московский Университет Связи и Информатики (МТУСИ), Москва, Россия

**Маликов Андрей Юрьевич**, Московский Университет Связи и Информатики (МТУСИ), Москва, Россия