UDC 621.396
DOI: 10.15587/1729-4061.2020.212301

DEVELOPMENT OF A DISTRIBUTED WIRELESS WI-FI SYSTEM FOR MONITORING THE TECHNICAL CONDITION OF REMOTE OBJECTS

N. Kaliaskarov
Doctoral Student*
E-mail: 90nurbol@mail.ru
V. Ivel
Doctor of Technical Sciences, Professor*
E-mail: ivelvic@mail.ru
Y. Gerasimova
PhD, Associate Professor*
E-mail: julyvic@mail.ru
V. Yugay
PhD, Head of Department
Department of Technology of Communication Systems
Karaganda Technical University
N. Nazarbayev ave., 56, Karaganda, Kazakhstan, 100000
E-mail: slawa_v@mail.ru
S. Moldakhmetov
Senior Lecturer*
E-mail: sayatmoldakhmetov@gmail.com
*Department of Energetic and Radioelectronics
M. Kozybayev North Kazakhstan University
Pushkin str., 86, Petropavlovsk, Kazakhstan, 150000

The research addresses the issues of collection and transmission of measurement data when monitoring various construction objects and bridge structures. The issues were resolved by developing a distributed system using Wi-Fi technologies. The results of measurements of the following parameters and data were obtained: the distance between cracks and joints, magnetometer readings and location and possible inclination of objects in three axes according to accelerometer and gyroscope data. By configuring the server, new channels are created for receiving data, which allow their subsequent processing and complete analysis of the study, for example, to solve the problem of predicting the technical conditions of construction and bridge objects. The completeness of the analysis of the study solves the problem of identifying and detecting possible errors and determining delays in the communication system. In general, the validity of all research results plays an important role in predicting the technical conditions of various objects and in finding solutions to problems arising from the technical difficulties of remote control. In this regard, the issue of the validity of the choice of Wi-Fi modules, which take into account the parameters of power consumption and availability of the boards of these modules for programming in order to obtain results of measurements in a long time has been posed and resolved. Setting power consumption parameters made it possible to increase the research time at remote sites, which in turn increases the durability and life of the control system. The simplicity of programming the module boards and support of various electronic sensors allow varying the scope and research objects, thereby expanding the geography of the subject area of research. Therefore, the developed distributed system is easily adapted to the necessary problem areas of research, the monitoring results of which can be used in many areas, such as agriculture, ecology, power, health care, meteorology and others

Keywords: wireless communication, remote monitoring, gyroscope, accelerometer, magnetometer, distance sensor, server

Received date 27.07.2020
Accepted date 15.09.2020
Published date 22.10.2020

1. Introduction

Improvement of autonomous systems for technical diagnostics of load-bearing structures of construction objects or bridge structures is associated with the need to use new measuring sensors and wireless information transmission devices. These sensors and devices must have increased accuracy, noise immunity and ultra-low power consumption.

An important element of any system of technical diagnostics of construction objects or bridge structures is their continuous monitoring, which is based on measuring the stress-strain parameters of load-bearing structures, as well as determining the level of their main parameters. An integral part of technical diagnostics systems is a data collection subsystem, i. e., a subsystem for transmitting measured data to a single information center. A necessary condition for the effective functioning of distributed technical diagnostics systems with a large number of sensors is the ability of wireless transmission of received data.

The distributed system is designed to receive information from measuring sensors [1]. In the system, a process takes place in which the measured physical quantities are first converted into electrical signals, and for further processing, analysis and storage they are converted into digital codes through an analog-to-digital converter. In this regard, the distributed system can be considered as a system that can not only collect information, but also allows processing and additional actions with the received data. This basic approach is used in many automation systems, at the initial levels of which collection of results, evaluation of states and solution of control problems are performed [2].

In modern automation, control and information transmission systems, the number of applications and software interfaces being developed has increased, to which various restrictions and requirements are set. Restrictions and requirements are related to the fact that any developed application is built with an expectation of instant scaling, which depends on the growing user demand. In this regard, there is a unified requirement that any application must be a distributed system [3]. With the emergence and widespread use of the "Internet of Things" terminology [4, 6], the task of writing applications and physical binding of them to various measuring devices has become easier. Accordingly, the implementation and development of various distributed systems have been simplified. In most cases, the measuring devices in these distributed systems are various sensors that measure parameters in strict accordance with data identification and tracking technologies. After measurement, the data are transmitted by wired and wireless methods [5].

The practical implementation of such a distributed system will allow it to be used in construction, communication systems, bridge structures on railways and highways, "smart" technologies and other areas. It is worth noting that the system allows timely detection of damage and provides ways to eliminate it. The system will increase the efficiency of spending on various measures to correct shortcomings by properly determining the location, angle of inclination, vibration indicators, temperature effect and the type of necessary repair measures. Sending data over a wireless distributed system will allow the user to remotely monitor the object under study. For example, he can find out the condition of a crack or joint in a building. This provides low power consumption, increases the service life of the device and improves the accuracy, information content and quality of forecasting the state of structures and buildings.

2. Literature review and problem statement

The work [7] presents the results of research using a wireless tilt sensor unit designed to monitor the state of structures. It is shown that two systems were used together to solve the problems related to communication distance limitations: a 2.4 GHz system and a code division multiple access (CDMA) system. Today, many countries stop using CDMA due to the transition to newer wireless technologies. Therefore, this development, described in [7], is currently difficult to implement due to the peculiarities of using the CDMA method in a distributed system. In addition, this technology does not use frequencies and does not transmit data if the receiving site is outside the country. Therefore, when using a remote monitoring system, it is desirable to choose unique wireless technologies that operate everywhere and on all devices that are compatible with other network nodes. An option to overcome these difficulties can be the use of Wi-Fi technology. Using this technology, sensors and modules are connected to an access point that provides access to the global Internet. The receiving site can be located anywhere, since all measured data will be received on a server in the global network, and only authorized researchers can access the server. For programming and managing a distributed data acquisition system, modules and boards programmed in the Arduino environment were chosen. This is because the chosen Wi-Fi technology is compatible with boards, modules and sensors working together with the Arduino platform. The features of these boards and their connectivity to sensors are described in [8, 9]. A wide range of board capabilities programmed in the Arduino environment when developing remote monitoring, data collection and transmission systems are described in [10, 11]. The works [10, 11] confirm the usefulness of this platform, since it allows building a wireless system for taking various sensor parameters. For example, in the system described in [10], measurements of temperature, humidity, soil moisture, rain and water level were made. But this paper does not solve an important issue of power consumption. In the work, a wired power supply is connected, which makes the system not completely wireless. The work [11] proposes a cost-effective system that monitors sensor data such as soil moisture, temperature, relative humidity and light intensity, but a prototype of the system has not been created. The paper is considered a simulation to determine the life of the network and has no practical reason that the issue of power consumption is completely resolved. All this suggests that it is expedient to conduct research and experimental work on not only reading and transmitting various parameters from sensors, but also to solving, first of all, the power consumption issue.

The work [12] provides an overview of environmental conditions of smart cities in order to develop an indoor air quality monitoring system. In [12], an analysis of more than 100 sources is given, but methods and techniques for measuring the readings of the magnetic field, which affects the seismological state of buildings and cities, are not indicated. Magnetic field readings are taken using a magnetometer, the introduction of which into the system described in [12] would cover this problem area more extensively.

In [13], a system for monitoring and analyzing the vibration acceleration spectrum based on the Raspberry Pi 3 microcomputer and ADXL345 three-axis digital accelerometer for real-time operation is developed. The system is up-to-date and easy to use when wired. When additional sensors are connected to the circuit to measure other research parameters, it is necessary to reprogram not only the Arduino board, but also the microcomputer board, which complicates system installing and configuring. Also, in the case of monitoring via wireless technologies, sending data to a single server or viewing results from devices such as a cell phone, laptop or tablet, additional difficulties will arise. Therefore, this development is not quite suitable for remote monitoring based on wireless technologies.

The work [14] presents latest solutions for monitoring the state of structures. The system includes various accelerometers, load cells, strain gauges and other devices. The sensors included in the system have been experimentally tested, but the issue of reliability and constancy of long-term monitoring has not been fully resolved. The authors of [14] considered these problems, so, when developing a monitoring system, it is necessary to take into account the issue of monitoring time and system reliability.

It is worth noting that there should be no structural and functional differences in systems that measure and transmit the results of some parameters (described in [10, 11]) to the server from similar systems that measure other parameters (described in [12, 14]).

As a result, all systems that take characteristics and parameters remotely must be autonomous, easy to configure, and low power-consuming. In addition, the system must reliably collect and transmit data wirelessly with high security and reliability of communication channels. All results should be available to researchers from different devices during long-term or periodic monitoring.

Therefore, it would be advisable to conduct a study using a single distributed system with these features and providing the maximum reliability of the measured data for monitoring, for example, the technical condition of structures and facilities. To solve problems in these research objects, it is necessary to configure a communication channel using a distributed system and measure the corresponding sensor readings. These include the distance between cracks and joints, magnetometer readings, the location and possible inclination of objects in three axes according to the accelerometer and gyroscope data.

3. The aim and objectives of the study

The aim of this study is to develop a distributed Wi-Fi system for collecting and transmitting data with the provision of power saving and information protection for monitoring the technical condition of bridge structures and construction sites.

This will allow conducting research to obtain measurement results and improve the quality of practical remote monitoring of technical conditions of various objects in any weather conditions and time. The low cost of connected devices and their availability for programming do not impair the reliability of the entire system, which increases the advantages of the developed wireless system. Proper server configuration simplifies operation from a practical point of view. This is due to the fact that an authorized user who has access to all measured data only needs to have access to the Internet from anywhere in the world and work on the created server channels to process the research results.

To achieve the aim, the following objectives were set:

– to select, program and configure the devices included in the distributed wireless system for collecting and transmitting measurement results using Wi-Fi technology;

– to get the results of experiments on the server and export the data for further processing and analysis of the distributed wireless system for full monitoring of the technical condition of bridge structures and construction sites;

– to use power-saving methods when configuring Wi-Fi modules included in the distributed system;

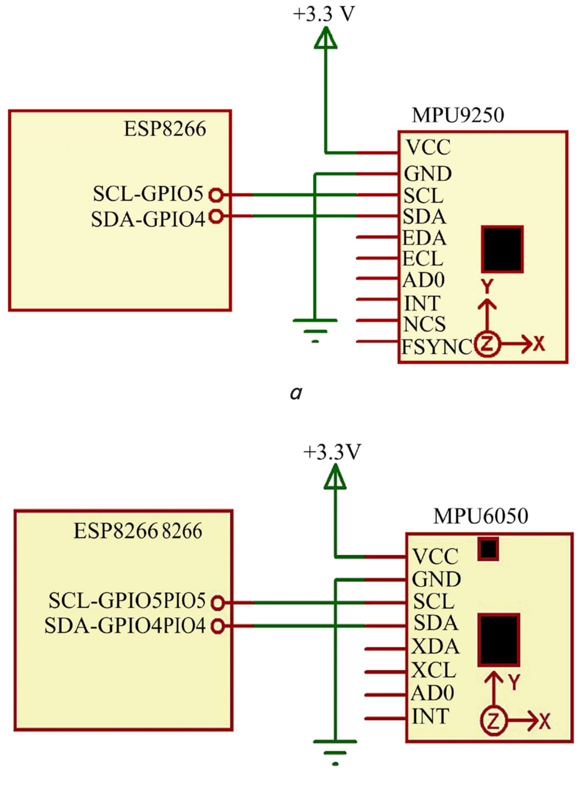
– to ensure the security of the transmitted data.

4. Materials and methods for studying the parameters of the distributed Wi-Fi system

The MPU-6050 and MPU-9250 sensors are miniature complexes that include gyroscopes, thermometers and accelerometers. Each of the sensors measures data along the X, Y, Z axes; in addition, a magnetometer is also built into the MPU-9250 sensor, which measures the corresponding data along the same three axes.

In the distributed wireless system, the NodeMCU V3 Wi-Fi module is used to send measured data to the server channels. The https://thingspeak.com/ site was used as a server, providing users with extensive settings for research work. The module is a kind of esp8266 board and is easily programmed to achieve the tasks and consumes low voltage, depending on the configured power-saving mode.

The characteristics of the sensors and the Wi-Fi module made it possible to choose them as the main devices for determining the position in space and the angle of inclination and magnetic field indicators of the objects under study. All measurement results received from the sensors are transmitted to the created server channels. Fig. 1, *a* shows the connection diagram of the MPU-6050 sensor and the esp8266 Wi-Fi module. The VCC pin is connected to a 3.3 V power supply, the GND pin is connected to the ground port (the Wi-Fi module has 3 ground ports), the SCL pin is a data line, and SDA is a synchronization line. The SCL and SDA pins are ports for connecting I2C devices, facilitating the connection and performing all the necessary functions for data transmission and management. The INT pin is a configurable interrupt, the AD0 pin – I2C address. When connecting the sensor, keep in mind that if this pin is pulled to the ground, the device address will be 0x68; if you connect AD0 to the power pin, the address will change to 0x69. Fig. 1, *b* shows the connection diagram of the MPU-9250 sensor and the esp8266 Wi-Fi module.



a

b

Fig. 1. Connection diagram of sensor pins and Esp8266 Wi-Fi module: *a* – MPU-9250 sensor; *b* – MPU-6050 sensor

The difference between the MPU-6050 and MPU-9250 sensors is the XCL, XDA pins, which are considered additional I2C interfaces for connecting an external magnetometer, all other pins are identical in their functional properties with the MPU-9250 sensor. Both types of sensors are used in the distributed wireless system and operate as follows. Several sensors are installed on the investigated object or structures in different places depending on the height and width of the object. Each sensor is connected to its own module that supports Wi-Fi technology. The module is set to the power-saving mode because measurement data is not transmitted every second, so there is no need to keep all devices active if no measurement is made at that time. The sensors measure data along three axes at the points of the object where they are installed. The sensors allow determining gyroscope, accelerometer and magnetometer parameters separately along the *X*-, *Y*-, *Z*-axes. Due to the fact that the distributed system has a scalability feature, the number and location of sensors in the system can be changed. The measurement time is set to 15 seconds to carry out more experiments in order to determine inaccuracies and obtain a large amount of data for processing. Therefore, the sensors send measurement results to the server on the created channels every 15 seconds. At the receiving site, the users responsible for data processing enter their accounts on respective channels and receive values from a computer or other receiving device that supports Internet access.

In addition to determining the angle of inclination and orientation of construction objects or bridge structures, an important parameter for remote monitoring is the ability to determine the distance of cracks or joints. The elements used for distance measurement in the wireless distributed system are described as follows: Fig. 2 shows the block diagram of the device for measuring the distance of cracks or joints. The device works as follows. With the expansion of the controlled crack, the distance between the reflecting panel and the receiver increases, since the receiver and emitter included in the optical crack measurement sensor are located on one side of the crack, and the reflecting panel – on the other. As a result, the signal of the receiver proportional to the width of the controlled crack is recorded. The optical sensor transmits an analog signal proportional to the crack size to the amplifier, the output of which is connected to the proportional-derivative element. The signal from the PD element is transmitted to the ADC. Then the digitized ADC signal is transmitted to the microcontroller. In the microcontroller, the signal is processed in accordance with a given algorithm and then via the Wi-Fi interface, which includes a transmitter and a receiver, is transmitted to the computer, where the final processing of the information signal takes place. In addition to the information signal, the microcontroller generates a signal that turns on the timer, the function of which is to periodically disconnect the power supply from all active units of the monitoring device in order to save energy [15].

Fig. 3 shows the connection diagram of the distance sensor and the wireless module.

This device is part of a distributed wireless system for collecting and transmitting analog data using dual-processor Wi-Fi transmitters. In the wireless distributed system, the HC-SR04 ultrasonic sensor serves as the distance sensor. NodeMCU V3 Lua Wi-Fi serves as a Wi-Fi module. The system must be connected to a 5 V power supply. Due to the fact that the Wi-Fi module does not have direct outputs to this voltage, an additional 4.7 kOhm resistor is connected in the circuit. The resistor converts the voltage and allows reading the distance data without any error. The sensor is equipped with four pins (standard 2.54 mm): positive power pin – +5 V; Trig (T) – input signal output; Echo (R) – output signal output; GND – “Ground” pin.

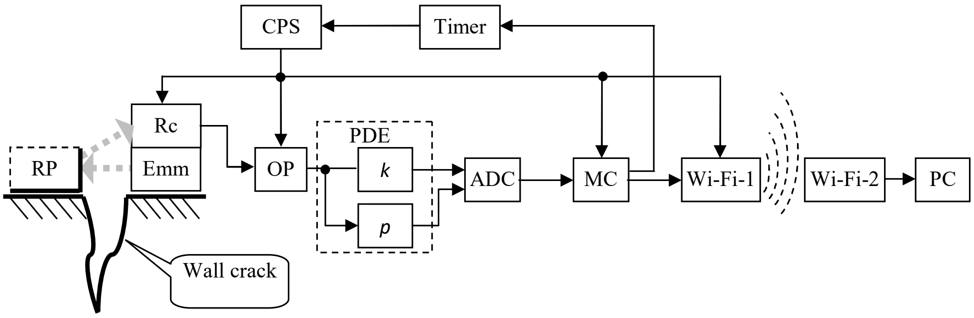


Fig. 2. Block diagram of the wireless device for monitoring the condition of cracks and joints of buildings and structures: optical sensor, consisting of a reflective panel (RP), emitter (Emm) and receiver (Rc), operational amplifier (OA), analog-to-digital converter (ADC), proportional-derivative element (PDE), microcontroller (MC), Wi-Fi transmitter (Wi-Fi-1), Wi-Fi receiver (Wi-Fi-2), personal computer (PC), controlled power supply (CPS) and timer

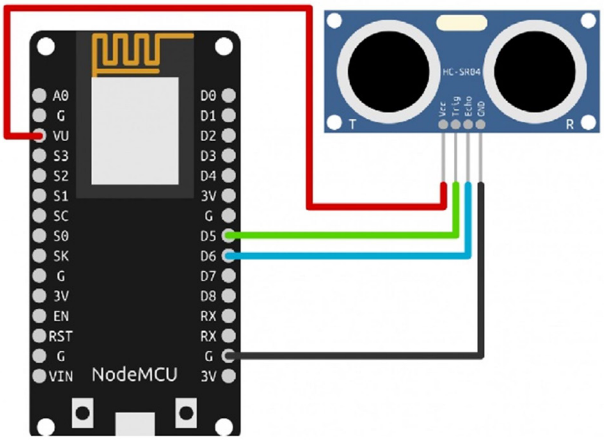


Fig. 3. Schematic connection diagram of the distance sensor and the Wi-Fi module

The solution to the problem of improving quality characteristics and reducing the power consumption of the monitoring device is based on the use of various methods. These methods include: the construction of autonomous wireless systems with minimum power consumption, the triangulation method for measuring the distance to the object, and the use of circuitry methods for constructing integrated electronic modules. The triangulation method is based on changing the angle of incidence of the laser beam depending on the distance to the object.

The wireless module was chosen because the NodeMCU Wi-Fi module supports various methods to minimize power consumption. Hence, it has many small size features with different modes including sleep mode and these modes can be accessed using some hardware and software modifications. The NodeMCU V3 ESP8266 module operates in the following modes.

Active mode: in this mode, the entire chip is on and the chip can receive, transmit data. Obviously, this is the most power-consuming mode.

Modem standby mode: in this mode, the CPU is running and Wi-Fi radio is disabled. This mode can be used in applications where processor operation is required, such as pulse width modulation. This causes the Wi-Fi modem to turn off when connected to a Wi-Fi access point (AP) without data transmission to optimize power consumption.

Standby mode: in this mode, the processor and all peripherals are suspended. Any wake-up like external interrupts will wake up the chip. Without data transmission, the Wi-Fi modem circuit can be disabled and the CPU is suspended to save power.

Deep sleep mode: in this mode, only the RTC works, and all other components of the chip are disabled. This mode is useful when data is transmitted at certain intervals. Table 1 shows the characteristics of the power-saving mode.

Table 1

Characteristics of the NodeMCU Wi-Fi power-saving mode

Name		Modem standby mode	Standby mode	Deep sleep mode
Wi-Fi		Off	Off	Off
System clock		On	Off	Off
RTC		On	On	On
CPU		On	Standby	Off
Substrate current		15 mA	0.4 mA	20 μA
Average current	DTIM=1	16.2 mA	1.8 mA	–
	DTIM=3	15.4 mA	0.9 mA	–
	DTIM=10	15.2 mA	0.55 mA	–

5. Results of studying the parameters of the distributed Wi-Fi system

5. 1. Connecting and configuring devices included in the wireless distributed system

After these devices are connected to the wireless distributed data collection and transmission system, the process of preparing the Wi-Fi module for recording and programming begins; for this, a unique software sketch is developed (Fig. 4–6). At the beginning of the sketch, the data of the Wi-Fi channel on the server is indicated, the data transfer rate is specified, the conditions of correct connection to the Wi-Fi network or no connection are prescribed. Next, the sketch takes into account the conditions necessary for binding to individual channel elements in order to receive gyroscope, accelerometer or magnetometer data for all parameters and axes. The sketch also configures the text displayed by the port monitor when the receiver is connected to the sensor using a USB cable. This setting is necessary to check the measurement accuracy of the sensors and monitor their performance before sending over a wireless connection. The rest of the sketch contains other software settings to perform all the necessary functions of this part of the distributed system, designed to determine the angle of inclination and orientation in space.

To connect the Wi-Fi module and the distance sensor, a new unique code is similarly written, which includes many strings in the programming language in the Arduino Ide software environment. The unique code ensures that all results are correctly read, transmitted and received on the server; it was also decided to display the measurement data on the port monitor. It is necessary to take into account the fact that after configuration and upgrading firmware of the board, all data can be transmitted to the server without a direct connection to a computer, the main thing is to have a power source for this device.

This sketch allows you to send data to the port monitor of the Arduino Ide software environment (Fig. 7) to check the performance of the sensors and to the server's personal channel over a wireless connection (Fig. 8, 9).

The operation of the distance sensor is based on the following process: the distance sensor is installed at a remote object, the operation of which is based on the triangulation method of measuring the distance to the object. Note that the distance sensor is included in the block diagram described in [15]; according to these diagrams, the sensor is connected to the Wi-Fi module. This module receives measured distance values with a certain period, which the user can change for his application. After that, the received distance values are sent to the server via the Wi-Fi module using a router. Based on the technical description of the distance sensor and the Wi-Fi module, these devices are compatible with each other and can work in the same system. The procedure for collecting and transmitting distance data is similar to the above procedures for transmitting measurement data of MPU-6050 and MPU-9250 sensors, the only differences are unique software sketches.

```
    if (client.connect(server,80))
                         {
                         String postStr = apiKey;
                                postStr +="&field1=";
                                postStr += String(IMU.getAccelX_mss(),6);
                                postStr +="&field2=";
                                postStr += String(IMU.getAccelY_mss(),6);
                                postStr +="&field3=";
                                postStr += String(IMU.getAccelZ_mss(),6);
                                postStr +="&field4=";
                                postStr += String(IMU.getGyroX_rads(),6);
                                postStr +="&field5=";
                                postStr += String(IMU.getGyroY_rads(),6);
                                postStr +="&field6=";
                                postStr += String(IMU.getGyroZ_rads(),6);
                                postStr += "\r\n\r\n";
                   client.print("POST /update HTTP/1.1\n");
                                client.print("Host: api.thingspeak.com\n");
                                client.print("Connection: close\n");
                                client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
                                client.print("Content-Type: application/x-www-form-urlencoded\n");
```

Fig. 4. Part of the software sketch designed to read measured gyroscope and accelerometer data and transmit them to the server via Wi-Fi

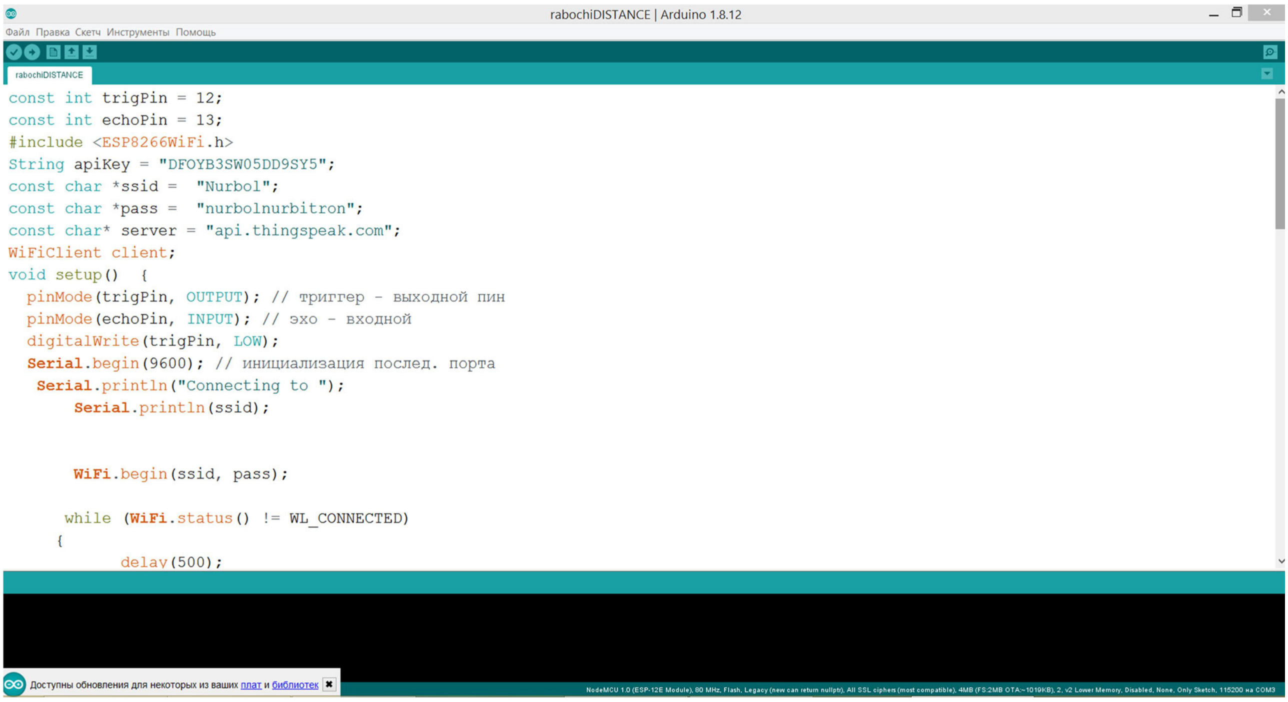
```
                             client.print("Connection: close\n");
                             client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
                             client.print("Content-Type: application/x-www-form-urlencoded\n");
                             client.print("Content-Length: ");
                             client.print(postStr.length());
                             client.print("\n\n");
                             client.print(postStr);

                             Serial.print("AccelX: ");
                             Serial.print(IMU.getMagX_uT(),6);
                             Serial.print("AccelY: ");
                             Serial.print(IMU.getMagY_uT(),6);
                             Serial.print("AccelZ: ");
                             Serial.print(IMU.getMagZ_uT(),6);

 Serial.println("%. Send to Server.");
                        }
          client.stop();
           Serial.println("Waiting...");

  delay(15000);
}
```

Fig. 5. Part of the software sketch designed to read measured magnetometer data and transmit them to the server via Wi-Fi

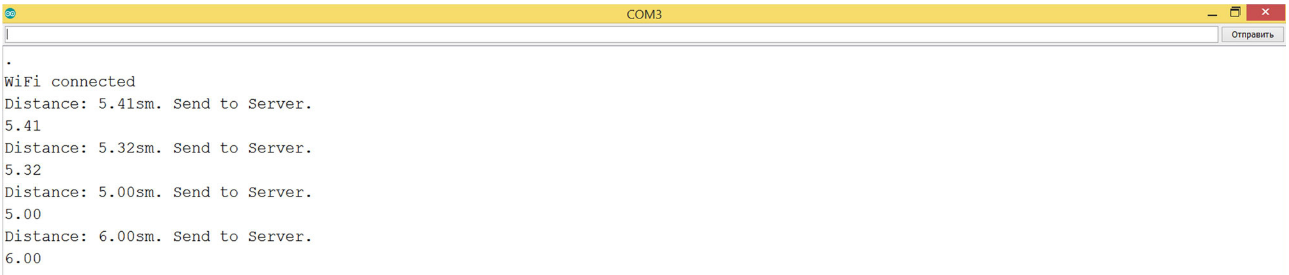


```
const int trigPin = 12;
const int echoPin = 13;
#include <ESP8266WiFi.h>
String apiKey = "DFOYB3SW05DD9SY5";
const char *ssid =  "Nurbol";
const char *pass =  "nurbolnurbitron";
const char* server = "api.thingspeak.com";
WiFiClient client;
void setup()  {
  pinMode(trigPin, OUTPUT); // триггер - выходной пин
  pinMode(echoPin, INPUT); // эхо - входной
  digitalWrite(trigPin, LOW);
  Serial.begin(9600); // инициализация послед. порта
   Serial.println("Connecting to ");
       Serial.println(ssid);


       WiFi.begin(ssid, pass);

      while (WiFi.status() != WL_CONNECTED)
     {
            delay(500);
```

Fig. 6. Part of the software sketch for transmitting data from the distance sensor via the Wi-Fi module and setting up the board for connecting to the server



```
.
WiFi connected
Distance: 5.41sm. Send to Server.
5.41
Distance: 5.32sm. Send to Server.
5.32
Distance: 5.00sm. Send to Server.
5.00
Distance: 6.00sm. Send to Server.
6.00
```

Fig. 7. Output of the measured distance data to the port monitor of the Arduino Ide environment

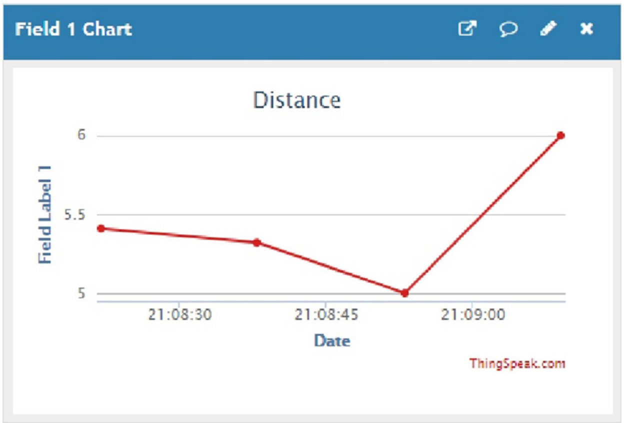


Fig. 8. Graph of measured distance data on the server

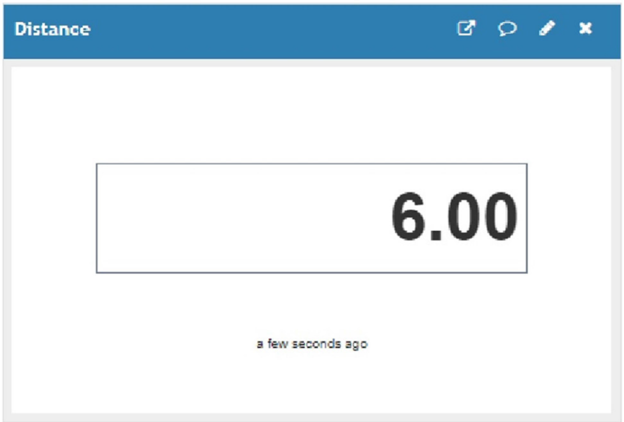


Fig. 9. Displaying the last distance measurement on the server

Fig. 10 shows the example of connecting the distance sensor to the Wi-Fi module.

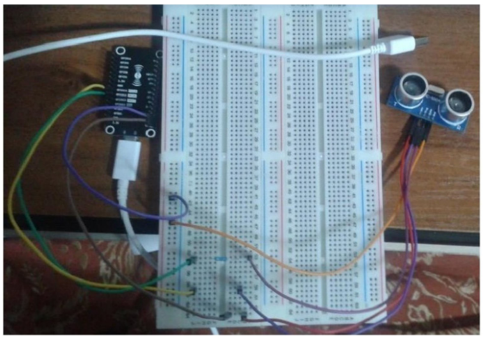


Fig. 10. Connecting the distance sensor to the Wi-Fi module in the developed distributed wireless system

5. 2. Getting research results at the receiving site of the server

All data (Fig. 11–17) is sent to the server, the convenience and advantage of this server are that you can find out the time and date of receiving a particular value. Based on the data obtained, the user can draw conclusions about how the values change during long-term monitoring of a remote object.

Fig. 11, 12 show the dates and times of the minimum and maximum distance measurements.

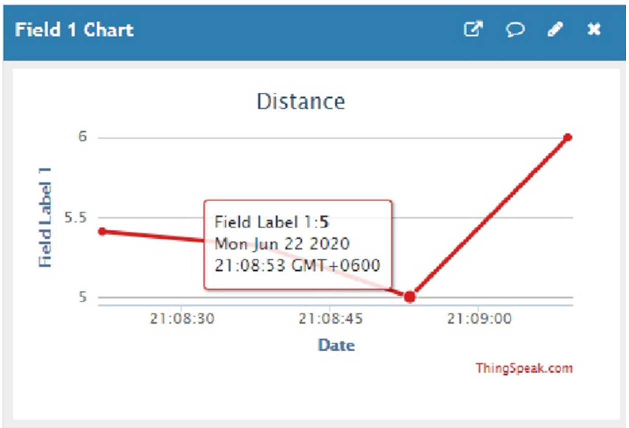


Fig. 11. Minimum value of measured distance with detailed description of date and time

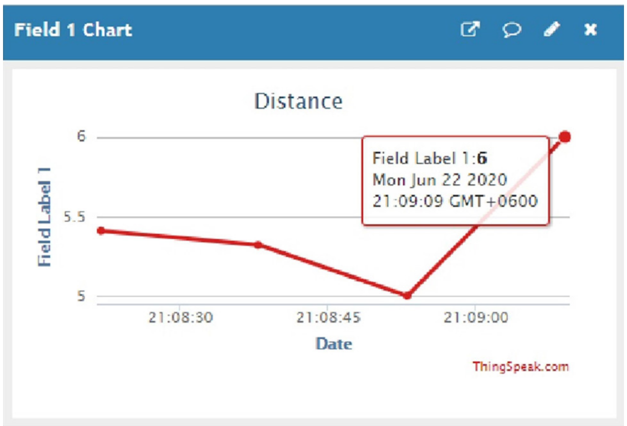
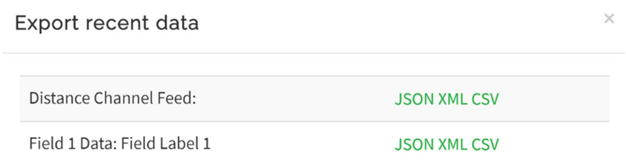


Fig. 12. Maximum value of measured distance with detailed description of date and time

Fig. 13 shows the available formats for exporting distance measurement results, and Fig. 14 shows the measurement data from research results in the Xml format. This format allows the user to process data for future embedding of these values in a separate HTML file. The HTML file is widely used when creating local pages using the MQTT server, which allows expanding the methods of monitoring and collecting the received data at the receiving site.

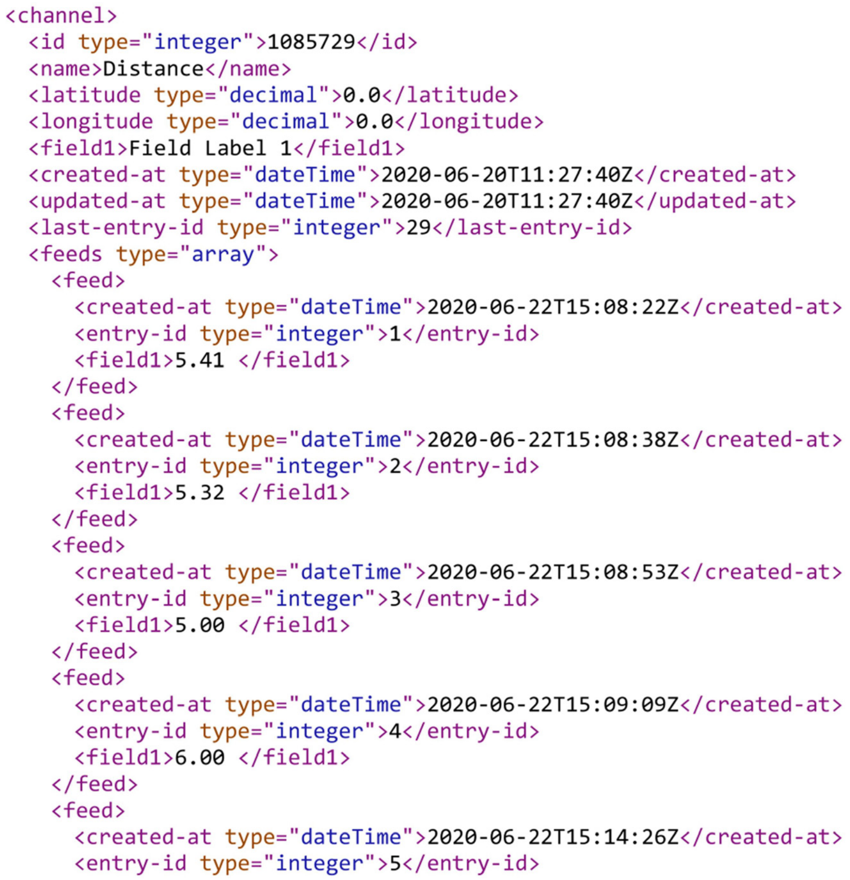


Export recent data

Distance Channel Feed:	JSON XML CSV
Field 1 Data: Field Label 1	JSON XML CSV

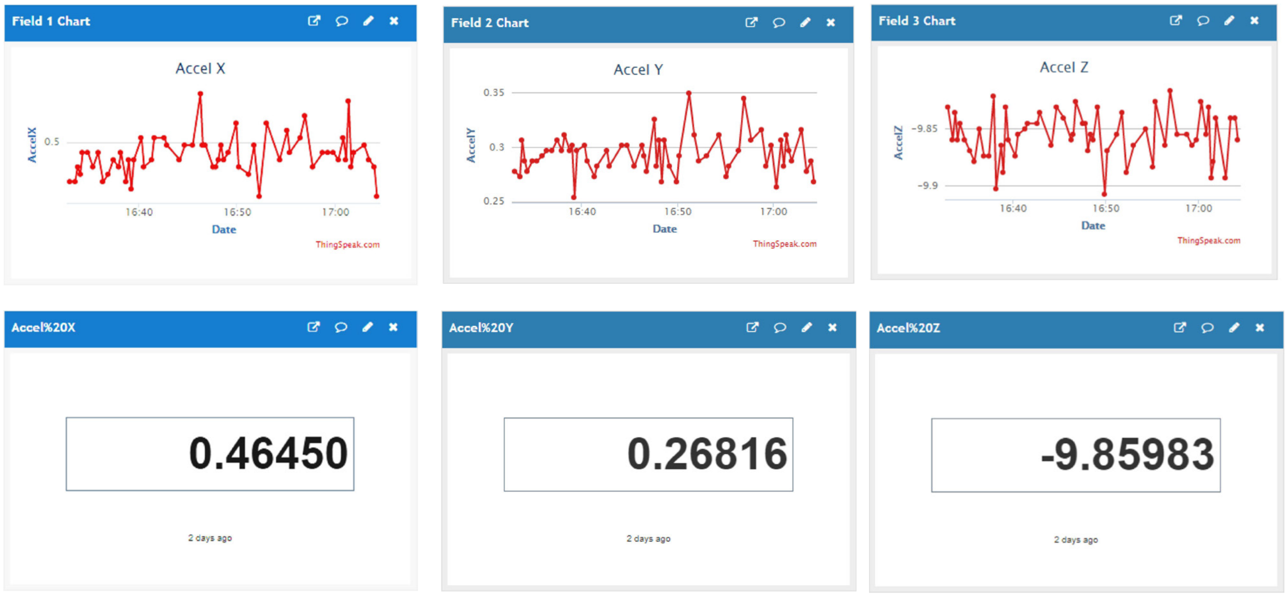
Fig. 13. Selecting the format for exporting measured distance readings

Fig. 15–17 show the results of experiments with initial measurements of the MPU-9250 sensors. In the course of the experimental work, the position of the building layout was changed every 15 seconds. Changes in the inclination angle and orientation in space were measured along three axes and the measurement results were sent using a Wi-Fi network from the sensors to the server.



```
<channel>
  <id type="integer">1085729</id>
  <name>Distance</name>
  <latitude type="decimal">0.0</latitude>
  <longitude type="decimal">0.0</longitude>
  <field1>Field Label 1</field1>
  <created-at type="dateTime">2020-06-20T11:27:40Z</created-at>
  <updated-at type="dateTime">2020-06-20T11:27:40Z</updated-at>
  <last-entry-id type="integer">29</last-entry-id>
  <feeds type="array">
    <feed>
      <created-at type="dateTime">2020-06-22T15:08:22Z</created-at>
      <entry-id type="integer">1</entry-id>
      <field1>5.41 </field1>
    </feed>
    <feed>
      <created-at type="dateTime">2020-06-22T15:08:38Z</created-at>
      <entry-id type="integer">2</entry-id>
      <field1>5.32 </field1>
    </feed>
    <feed>
      <created-at type="dateTime">2020-06-22T15:08:53Z</created-at>
      <entry-id type="integer">3</entry-id>
      <field1>5.00 </field1>
    </feed>
    <feed>
      <created-at type="dateTime">2020-06-22T15:09:09Z</created-at>
      <entry-id type="integer">4</entry-id>
      <field1>6.00 </field1>
    </feed>
    <feed>
      <created-at type="dateTime">2020-06-22T15:14:26Z</created-at>
      <entry-id type="integer">5</entry-id>
```

Fig. 14. Output of all measured data in the Xml format



0.46450

2 days ago

0.26816

2 days ago

-9.85983

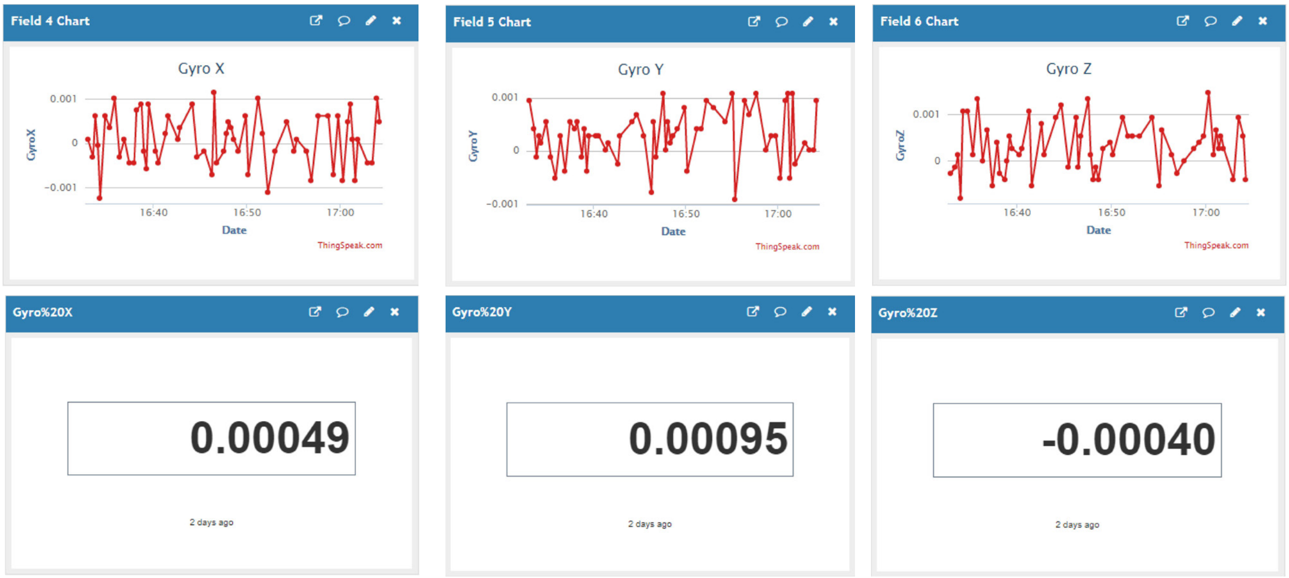
2 days ago

a *b* *c*

Fig. 15. Accelerometer readings during initial measurement: *a* — along the *X*-axis; *b* — along the *Y*-axis; *c* — along the *Z*-axis

The obtained measurements allow the researcher at the receiving site to see all deviations, jumps and possible errors in the network in the form of graphs. Processing the initial measurement data made it possible to correct all inaccuracies and errors occurring in the network. The configuration and successful testing of sensors and devices included in the wireless distributed network, as well as the construction of a wireless network, allows installing these sensors on any construction site or bridge structure.

Fig. 18, 19 show the available formats for exporting gyroscope, accelerometer and magnetometer measurements along the available axes of the research work.



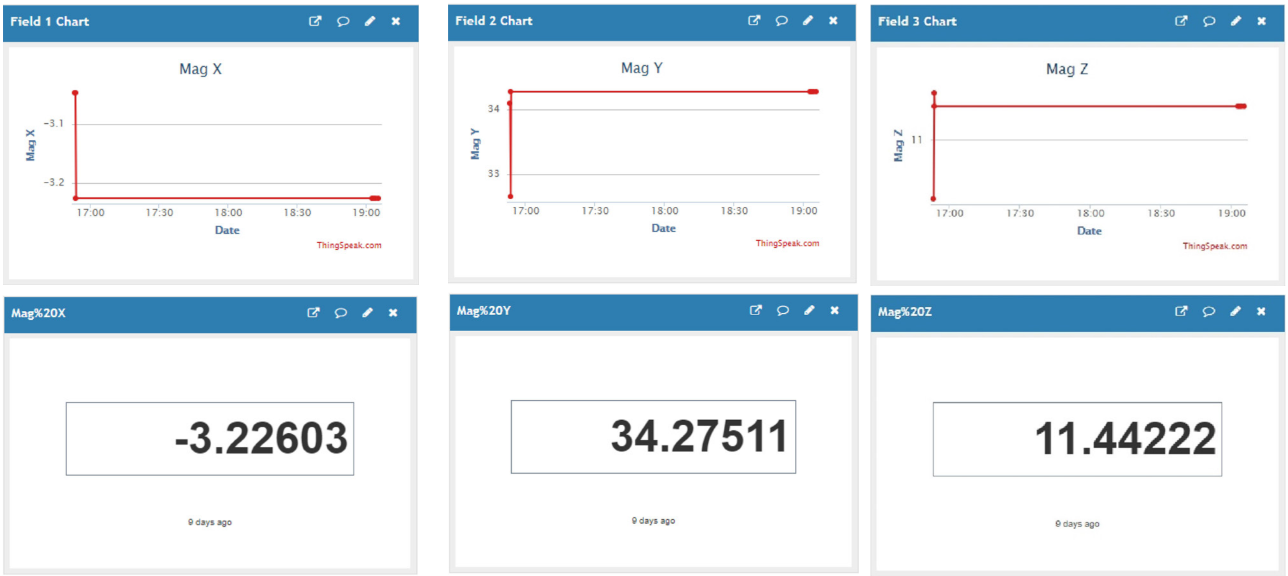
0.00049

0.00095

-0.00040

a *b* *c*

Fig. 16. Gyroscope readings during initial measurement: *a* – along the *X*-axis; *b* – along the *Y*-axis; *c* – along the *Z*-axis



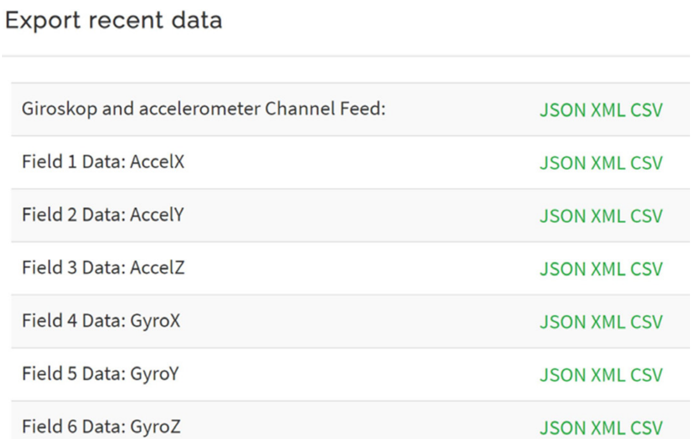
-3.22603

34.27511

11.44222

a *b* *c*

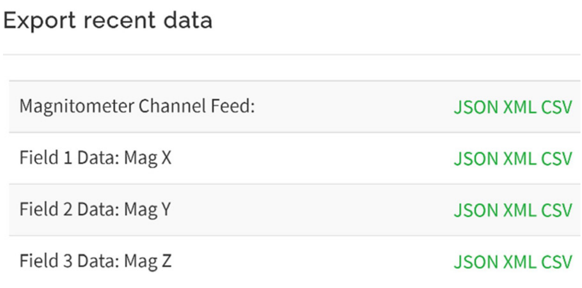
Fig. 17. Magnetometer readings during initial measurement: *a* – along the *X*-axis; *b* – along the *Y*-axis; *c* – along the *Z*-axis



Export recent data

Giroskop and accelerometer Channel Feed:	JSON XML CSV
Field 1 Data: AccelX	JSON XML CSV
Field 2 Data: AccelY	JSON XML CSV
Field 3 Data: AccelZ	JSON XML CSV
Field 4 Data: GyroX	JSON XML CSV
Field 5 Data: GyroY	JSON XML CSV
Field 6 Data: GyroZ	JSON XML CSV

Fig. 18. Export of gyroscope and accelerometer data in 3 formats



Export recent data

Magnitometer Channel Feed:	JSON XML CSV
Field 1 Data: Mag X	JSON XML CSV
Field 2 Data: Mag Y	JSON XML CSV
Field 3 Data: Mag Z	JSON XML CSV

Fig. 19. Export of magnetometer data in 3 formats

5. 3. Features of modes, operating methods and ways to ensure the reliability of the distributed system

Measurements in the distributed system allow highlighting the following features:

– all measured analog data of the sensors are automatically digitized using built-in ADC and displayed as results on the created server channel. To create a channel, it is necessary to specify its name and set the number of windows with graphs (indicated in the settings as “field”) to receive sensor data or other additional data. As shown in Fig. 20, *a*, the maximum allowed number of graphs for obtaining data is 8, and in Fig. 20, *b*, the created channels are displayed, the maximum number of which is 4. In this regard, one account will allow working with 32 parameters on the server at the same time, and when working in the system with several accounts, the number of parameters will increase several times.

– the possibility to select any point on the graph and view the date and time of the measurement at any point;

– the possibility to select power mode;

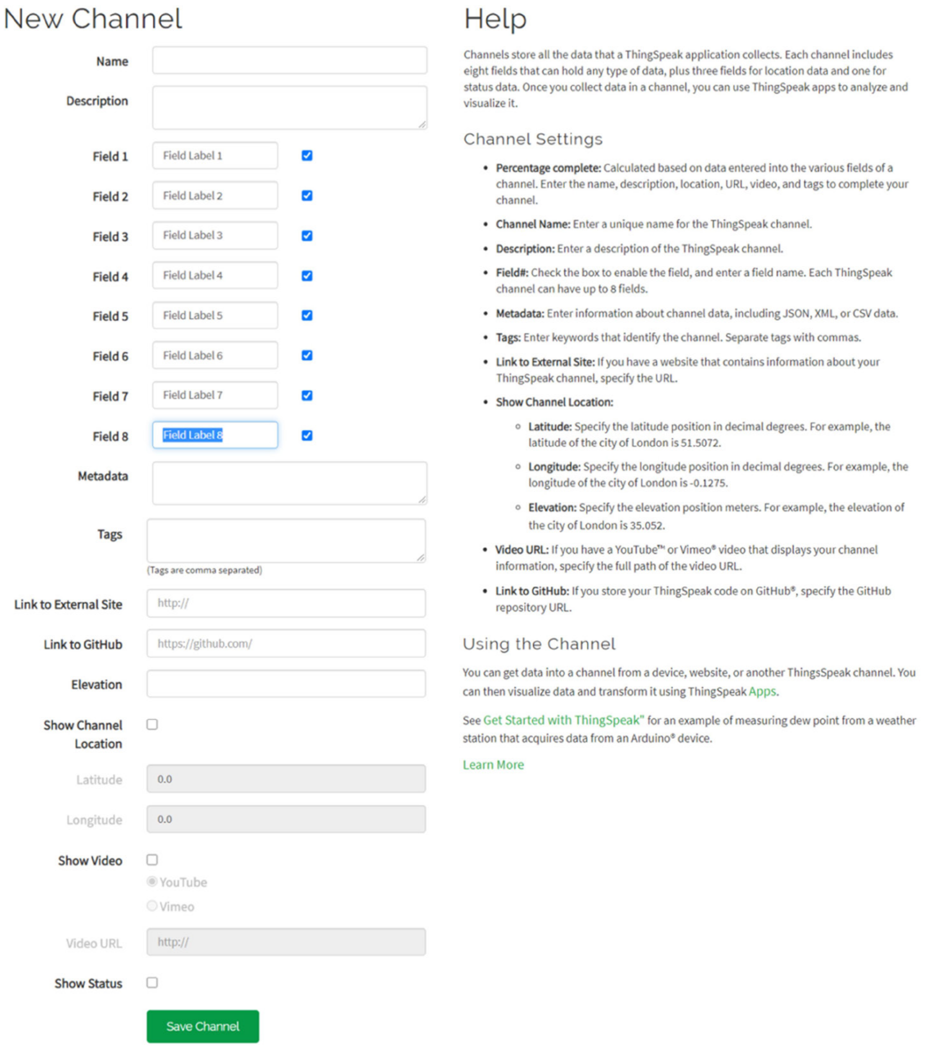
– the server automatically creates a coordinate system for displaying the results in the form of graphs;

– the possibility to export the received data in 3 formats (json, xml, csv);

– displaying the last result as a measuring device, a display with a number or a lamp indicator;

– the possibility to process measurement results in the Matlab software environment, without leaving the site and without installing this program on a computer (Fig. 21, 22).

It should be borne in mind that data transmission over any network should be as secure as possible. In the experimental part of the work, the first step in data protection is to create a login and password in the server’s personal account (Fig. 23, *a*). The second step of protection: each created channel has its own encryption keys (Fig. 23, *b*), without which the Wi-Fi module will not be able to create a connection, so, this key is written in the Arduino Ide software sketch (Fig. 23, *c*). The third step of protection is access to the data of the Wi-Fi network itself, the login and password of which are also specified in the Arduino Ide software sketch (Fig. 23, *c*). In this regard, it can be argued that protection against someone else’s access to data, and even more to changing the programming sketch, is at a good and reliable level.



New Channel

Name:
Description:
Field 1: Field Label 1 ☑
Field 2: Field Label 2 ☑
Field 3: Field Label 3 ☑
Field 4: Field Label 4 ☑
Field 5: Field Label 5 ☑
Field 6: Field Label 6 ☑
Field 7: Field Label 7 ☑
Field 8: Field Label 8 ☑
Metadata:
Tags: (Tags are comma separated)
Link to External Site: http://
Link to GitHub: https://github.com/
Elevation:
Show Channel Location ☐
Latitude: 0.0
Longitude: 0.0
Show Video ☐
● YouTube
○ Vimeo
Video URL: http://
Show Status ☐

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

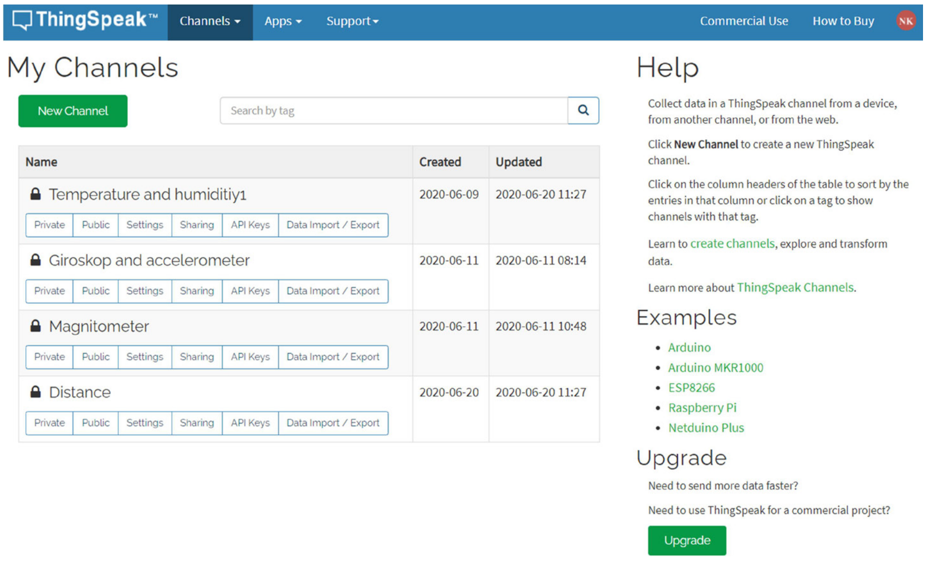
- **Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Show Channel Location:**
  - **Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
  - **Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
  - **Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
- **Video URL:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.
- **Link to GitHub:** If you store your ThingSpeak code on GitHub®, specify the GitHub repository URL.

Using the Channel

You can get data into a channel from a device, website, or another ThingsSpeak channel. You can then visualize data and transform it using ThingSpeak Apps.

See Get Started with ThingSpeak" for an example of measuring dew point from a weather station that acquires data from an Arduino® device.

a



My Channels

Name	Created	Updated
Temperature and humiditiy1	2020-06-09	2020-06-20 11:27
Giroskop and accelerometer	2020-06-11	2020-06-11 08:14
Magnitometer	2020-06-11	2020-06-11 10:48
Distance	2020-06-20	2020-06-20 11:27

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

Learn to create channels, explore and transform data.

Learn more about ThingSpeak Channels.

Examples

- Arduino
- Arduino MKR1000
- ESP8266
- Raspberry Pi
- Netduino Plus

Upgrade

Need to send more data faster?

Need to use ThingSpeak for a commercial project?

b

Fig. 20. Working with channels on the server to receive data:
a — configuring a new channel; *b* — list of available channels on the server

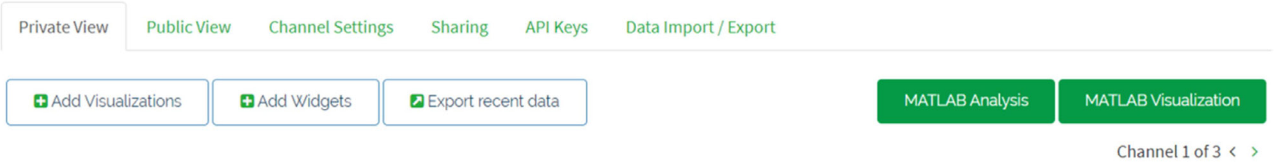
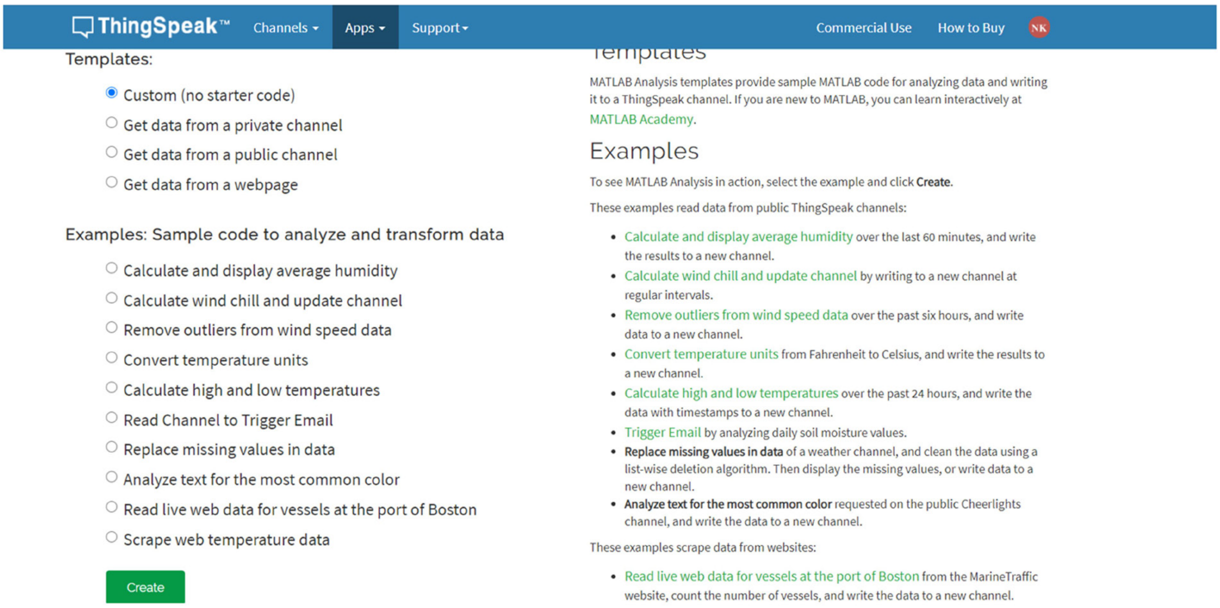


Fig. 21. Buttons for processing received data using Matlab on the server



Templates:

● Custom (no starter code)
○ Get data from a private channel
○ Get data from a public channel
○ Get data from a webpage

Examples: Sample code to analyze and transform data

○ Calculate and display average humidity
○ Calculate wind chill and update channel
○ Remove outliers from wind speed data
○ Convert temperature units
○ Calculate high and low temperatures
○ Read Channel to Trigger Email
○ Replace missing values in data
○ Analyze text for the most common color
○ Read live web data for vessels at the port of Boston
○ Scrape web temperature data

Create

Templates

MATLAB Analysis templates provide sample MATLAB code for analyzing data and writing it to a ThingSpeak channel. If you are new to MATLAB, you can learn interactively at MATLAB Academy.

Examples

To see MATLAB Analysis in action, select the example and click **Create**.

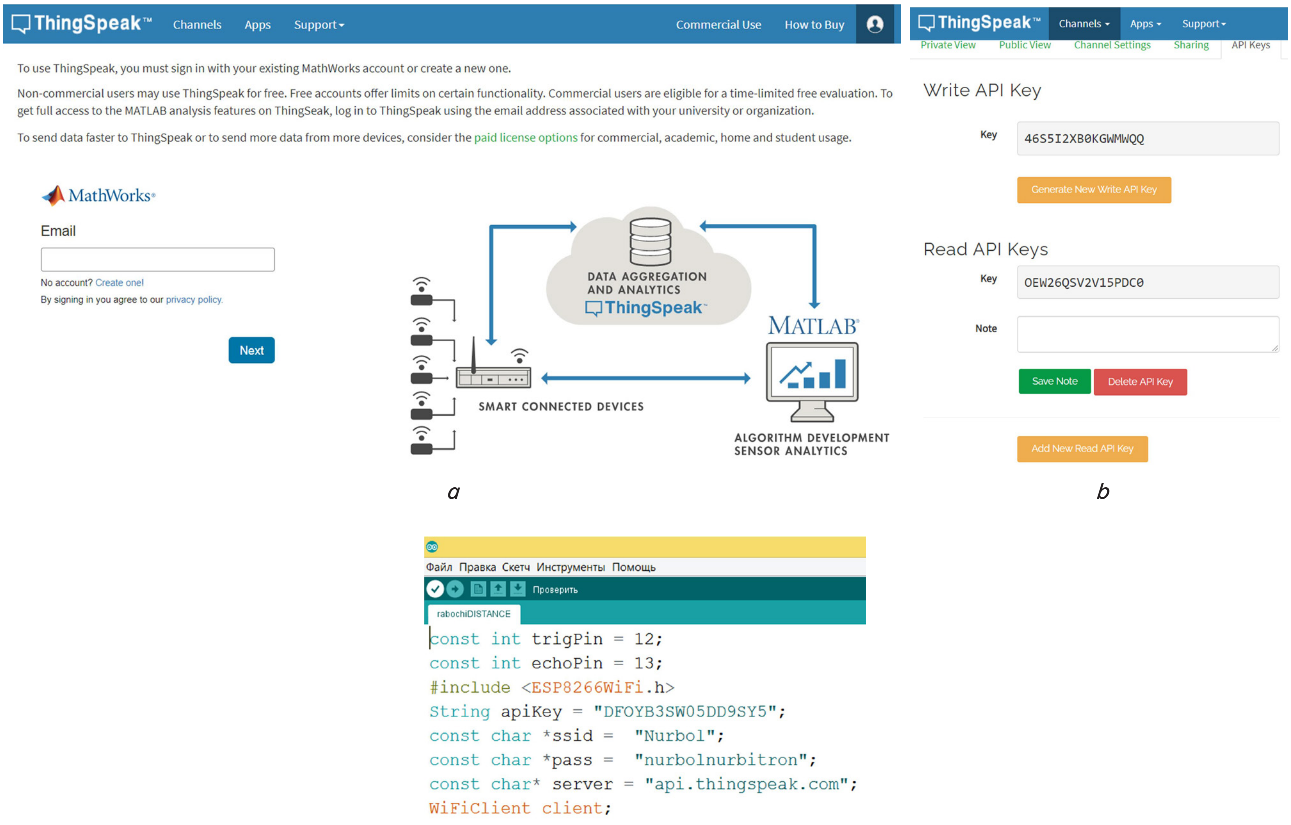
These examples read data from public ThingSpeak channels:

- Calculate and display average humidity over the last 60 minutes, and write the results to a new channel.
- Calculate wind chill and update channel by writing to a new channel at regular intervals.
- Remove outliers from wind speed data over the past six hours, and write data to a new channel.
- Convert temperature units from Fahrenheit to Celsius, and write the results to a new channel.
- Calculate high and low temperatures over the past 24 hours, and write the data with timestamps to a new channel.
- Trigger Email by analyzing daily soil moisture values.
- **Replace missing values in data** of a weather channel, and clean the data using a list-wise deletion algorithm. Then display the missing values, or write data to a new channel.
- **Analyze text for the most common color** requested on the public Cheerlights channel, and write the data to a new channel.

These examples scrape data from websites:

- Read live web data for vessels at the port of Boston from the MarineTraffic website, count the number of vessels, and write the data to a new channel.

Fig. 22. Server window for processing data in Matlab



To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the paid license options for commercial, academic, home and student usage.

Email

No account? Create one!

By signing in you agree to our privacy policy.

Next

Write API Key

Key: 46S5I2XB0KGWMWQQ

Generate New Write API Key

Read API Keys

Key: OEW26QSV2V15PDC0

Note:

Save Note

Delete API Key

Add New Read API Key

```
const int trigPin = 12;
const int echoPin = 13;
#include <ESP8266WiFi.h>
String apiKey = "DFOYB3SW05DD9SY5";
const char *ssid =  "Nurbol";
const char *pass =  "nurbolnurbitron";
const char* server = "api.thingspeak.com";
WiFiClient client;
```

Fig. 23. Configuring data security: *a* – authentication on the server; *b* – window for setting channel encryption keys; *c* – setting the encryption key and Wi-Fi parameters in the Arduino Ide software environment

6. Discussion of the results of studying the parameters of the distributed Wi-Fi system

Due to the proper Wi-Fi module that supports 4 power options, power consumption is reduced. Depending on the type and frequency of the study, each user chooses his own power mode. To reduce power consumption, the deep sleep mode was selected and configured in the developed distributed system, during which the following work was performed:

1) connecting the module with a Wi-Fi access point;

2) performing a task such as reading the sensor value, posting an MQTT message, and others;

3) sleep for a certain number of microseconds. Sleep time is measured in microseconds. According to ESP8266 SDK, the device can operate in this mode only for 4.294.967.295 µs, which is about 71 minutes;

4) these processes can be repeated several times.

The dimensions of the devices (Wi-Fi module and sensors) selected in the system are small, the entire circuit occupies a small area, which allows the system to work at remote sites in hard-to-reach places, thereby solving the issue of autonomy. The choice of devices used in the system is made so that the programmable part does not complicate the work of researchers in the field under study and can be adjusted to new conditions without the help of professional programmers. Despite this, the written sketches provide high reliability of the transmitted information due to the fact that a universal programming environment has been chosen and the system is configured so that the authentication and encryption processes take place in several stages (Fig. 23).

The obtained measurement results and the possibility of their processing make it possible to conduct remote online monitoring. Research results can be viewed using any device with access to the global Internet. As you know, today Internet access is carried out using a variety of devices (smartphones, tablets, laptops, computers, minicomputers, etc.). Thus, the distributed wireless system solves the problem of configuring and compatibility of equipment at the receiving site with the transmitting device.

To resolve the issue of conducting a full analysis and processing of the results, a server was selected that provides the ability to export data and process results in the embedded Matlab software environment (Fig. 22) without leaving a personal account. The measured output data and results in the system can be processed in various ways. For example, all research results can be recorded manually and, filling in data sets, dependencies of parameter changes can be built – in this case, the researcher can see all possible errors. But the process of manually filling in all the results takes much time and human factors may arise when entering data, which will lead to errors. Therefore, the created system has the ability to export all data in a convenient format and process all the results in the Matlab software environment.

The transmission of measurement results takes at least 15 seconds, due to the fact that the collection and reception of data on the server take a certain time. In this regard, the server will not have time to receive results every second due to the specifics of configuration. But this minimum time does not affect the overall quality and system's performance. This is because the measurement of distances between cracks and joints, magnetometer readings, location and possible inclination of objects in three axes according to the accelerometer and gyroscope data is periodic and does not require updating the results every second.

In addition to the minimum time limit, there are also limitations associated with installing additional libraries in the Arduino Ide software environment. When replacing sensors or selecting updated versions of Wi-Fi modules in the future, it will be necessary to reprogram the boards taking into account the new versions of libraries. If the configuration is performed from one computer and there is a need to reconfigure the system elements from another computer, all the libraries and modules should be added on the new computer being configured. But this task is not difficult, all libraries and boards are available in all versions of the Arduino Ide software environment, the licensed version of which is available in free space.

The research based on the use of the wireless distributed system described in the paper should be developed with implementation in other areas, using and replacing the corresponding control elements in problem areas. In addition to the parameters of the distances of cracks and joints, magnetometer, gyroscope and accelerometer readings, it is necessary to develop a study of problematic issues in such areas as agriculture, ecology, power, health care, meteorology and others. For example, for agriculture, the possibility of remote monitoring of soil temperature and moisture is being expanded. In the field of ecology, it will be possible to remotely observe the dynamics of water pollution or air quality. In the power industry, it will be possible to remotely control the power modes of remote objects, depending on the load of the sensors used. In health care, it will be possible to implement a system for remote measurement of heart rate, body temperature or pressure. In meteorology, the system will be able to monitor climate changes. In other words, by conducting a similar study in these areas using the developed system, it is possible to achieve reliable monitoring results in a simple, remote and reliable way.

7. Conclusions

1. After comparing all available devices, suitable sensors: MPU-6050 (China), MPU-9250 (China) and HC-SR04 (China) and Wi-Fi modules: Node MCU belonging to the esp8266 family are selected. The modules selected in the distributed system operate using Wi-Fi technology, the advantages of this technology and the validity of choosing it were described in [16]. When choosing sensors that perform the functions of a gyroscope, accelerometer, magnetometer and distance measurement, the possibility of installing them in hard-to-reach places was taken into account, and the choice of Wi-Fi module was based on the possibility of configuring an energy-saving mode. A unique software sketch has been developed in the Arduino Ide environment, which provides reliable communication between the transmitting and receiving sites of the wireless distributed system. The proposed concept of building an automated wireless system, which ensures minimum power consumption, will allow creating efficient and low-cost systems for autonomous round-the-clock monitoring of the state of load-bearing structures. The elements used in the system contribute to the monitoring of the technical condition and will prevent accidents, increase the service life and improve the accuracy, information content and quality of forecasting the state of bridge structures and buildings.

2. Data reception is carried out using individual channels created on the server, each channel is configured individually, based on the researcher's requirements. Each created channel is able to receive up to 8 parameters, and the server used allows one user to create up to 4 channels to choose from, thereby

ensuring the scalability of the research volume as needed. The server used in the wireless distributed system is configured to store all measured data by date and time. All research results are saved and display numerical values on the server in the created personal channels. The ability to process research results helps the researcher assess the quality of the wireless communication system and detect errors for timely correction.

3. Of the 4 power modes, the module deep sleep mode is selected. This is because the modem standby and standby modes are useful when you want the ESP8266 module to work while some functions are disabled. But if serious power control is needed, it is necessary to switch to the deep sleep mode. The total average current is less than 1 mA. At 2.5 V, the current requirement is only 20 µA.

4. In the distributed system, the security of transmitted data was ensured in several stages. To gain access to the data receiving channel, a server was selected that requires authentication only of those people who have access to the study. Encryption keys were generated for updating the firmware of boards and binding them to sensors and the server. To build a wireless communication line, Wi-Fi technology was chosen that protects transmitted data using the WPA2 PSK technology, which only researchers who have access to the work have access to.

Acknowledgments

The study was funded by the rector of the NKSU named after M. Kozybaev following the results of the competition for the rector's grant for young scientists of the NKSU named after M. Kozybaev for 2020, order number OD-356 dated 06.29.2020.

References

1. Park, Dzh., Makkey, S. (2006). Sbor dannyh v sistemah kontrolya i upravleniya. Prakticheskoe rukovodstvo. Moscow: OOO "Gruppa IDT", 504. Available at: https://www.studmed.ru/park-dzh-makkey-s-sbor-dannyh-v-sistemah-kontrolya-i-upravleniya-prakticheskoe-rukovodstvo_faab5c7fbdc.html
2. Pyavchenko, O. N. (2011). The distributed systems of collection and information processing of sensors of dynamic objects. Izvestiya Yuzhnogo federal'nogo universiteta. Tehnicheskie nauki, 5 (118), 8–14. Available at: https://cyberleninka.ru/article/n/raspredelennye-sistemy-sbora-i-obrabotki-informatsii-datchikov-dinamicheskih-obektov
3. Burns, B. (2018). Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services. O'Reilly Media, 166. Available at: https://www.amazon.com/Designing-Distributed-Systems-Patterns-Paradigms/dp/1491983647
4. Wortmann, F., Flüchter, K. (2015). Internet of Things. Business & Information Systems Engineering, 57 (3), 221–224. doi: https://doi.org/10.1007/s12599-015-0383-3
5. Atzori, L., Iera, A., Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, 54 (15), 2787–2805. doi: https://doi.org/10.1016/j.comnet.2010.05.010
6. Xia, F., Yang, L. T., Wang, L., Vinel, A. (2012). Internet of Things. International Journal of Communication Systems, 25 (9), 1101–1102. doi: https://doi.org/10.1002/dac.2417
7. Ha, D., Park, H., Choi, S., Kim, Y. (2013). A Wireless MEMS-Based Inclinometer Sensor Node for Structural Health Monitoring. Sensors, 13 (12), 16090–16104. doi: https://doi.org/10.3390/s131216090
8. Ocak, M. A. (2018). Where does Arduino's power come from?: An extended literature review. Journal of Learning and Teaching in Digital Age, 3 (1), 21–34. Available at: https://dergipark.org.tr/en/download/article-file/1175624
9. Faugel, H., Bobkov, V. (2013). Open source hard- and software: Using Arduino boards to keep old hardware running. Fusion Engineering and Design, 88 (6-8), 1276–1279. doi: https://doi.org/10.1016/j.fusengdes.2012.12.005
10. Khoa, T. A., Man, M. M., Nguyen, T.-Y., Nguyen, V., Nam, N. H. (2019). Smart Agriculture Using IoT Multi-Sensors: A Novel Watering Management System. Journal of Sensor and Actuator Networks, 8 (3), 45. doi: https://doi.org/10.3390/jsan8030045
11. Kumar, S., Ch, N. (2020). IoT Based Multi-sensor Data Acquisition System for the Application of Smart Agriculture. Communications in Computer and Information Science, 329–342. doi: https://doi.org/10.1007/978-981-15-3666-3_27
12. Marques, G., Saini, J., Dutta, M., Singh, P. K., Hong, W.-C. (2020). Indoor Air Quality Monitoring Systems for Enhanced Living Environments: A Review toward Sustainable Smart Cities. Sustainability, 12 (10), 4024. doi: https://doi.org/10.3390/su12104024
13. Holovatyy, A., Teslyuk, V., Iwaniec, M., Mashevska, M. (2017). Development of a system for monitoring vibration accelerations based on the raspberry pi microcomputer and the ADXL345 accelerometer. Eastern-European Journal of Enterprise Technologies, 6 (9 (90)), 52–62. doi: https://doi.org/10.15587/1729-4061.2017.116082
14. Abruzzese, D., Micheletti, A., Tiero, A., Cosentino, M., Forconi, D., Grizzi, G. et. al. (2020). IoT sensors for modern structural health monitoring. A new frontier. Procedia Structural Integrity, 25, 378–385. doi: https://doi.org/10.1016/j.prostr.2020.04.043
15. Ivel', V. P., Razinkin, V. P., Kaliaskarov, N. B. (2019). Development of wireless device monitoring of the condition of cracks and joints of buildings and structures and its advantages. The Bulletin of KazATC, 2 (109), 10–17. Available at: https://www.kazatk.kz/material/nauka/11.04.2019/%D0%B2%D0%B5%D1%80%D1%81%D1%82%D0%BA%D0%B0%20%D0%92%D0%B5%D1%81%D1%82%D0%BD%D0%B8%D0%BA%20%E2%84%96%202%20(109),%202019%20%D1%81%20%D0%BE%D0%B1%D0%BB%D0%BE%D0%B6%D0%BA%D0%BE%D0%B9%20%D0%B8%20%D0%B2%D1%8B%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%BC%D0%B8%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.pdf
16. Ivel', V. P., Gerasimova, Yu. V., Kaliaskarov, N. B. (2018). Obzor raspredelennoy besprovodnoy sistemy sbora i peredachi analogovyh dannyh. «Kozybaevskie chteniya-2018: Evraziyskiy potentsial i novye vozmozhnosti razvitiya v usloviyah global'nyh vyzovov»: materialy mezhdunarodnoy nauchno-prakticheskoy konferentsii. Vol. 2. Petropavlovsk: SKGU im. M. Kozybaeva, 243–247. Available at: http://repository.nkzu.kz/7813/