UDC 004.42: 620.9
DOI: 10.15587/1729-4061.2022.267892

SOFTWARE PROTOTYPE DEVELOPMENT FOR NON-CENTRALIZED OBJECTS OF WIND FLOW AMPLIFICATION

Gaukhar Alina
Corresponding author
Doctoral Candidate, Master of Technical Science*
E-mail: alinagaukhar@gmail.com
Nurlan Tashatov
PhD, Candidate of Physical and Mathematical Science
Department of Computer Engineering
L. N. Gumilyov Eurasian National University
Satpayev str., 2, Nur-Sultan, Republic of Kazakhstan, 010000
Galina Tatkeyeva
Doctor of Technical Sciences, Head of Department **
Madi Bauyrzhanuly
Master of Technical Sciences**
Dinara Kaibassova
PhD*
Margulan Nurtay
Master of Technical Sciences*
*Department of Information Computing Systems
Abylkas Saginov Karaganda Technical University
N. Nazarbayev ave., 56, Karaganda,
Republic of Kazakhstan, 100027
**Department of Power Supply
S. Seifullin Kazakh Agro Technical University
Zhenis ave., 62, Nur-Sultan, Republic of Kazakhstan, 010000

This research is devoted to the development of software to increase the efficiency of autonomous wind-generating substations using panel structures, which will allow the use of wind energy to generate electricity with minimal losses and for the life support of buildings and structures. In the course of the work, a software and hardware system with a functional diagram for experimental measurements was developed. The paper also describes the process of modeling wind generation, collecting and transmitting real-time data to a web server via the HTTPS protocol. Due to the intensive development of wind energy in Republic of Kazakhstan, there is a need to apply methods to improve the energy generation process. In particular, the use of hardware and software to monitor and make decisions on optimizing the power generation process will help solve the problem of limited economic and labor resources. The results of the experiments revealed that the automatic control of the shield structures allows specialists to increase the effectiveness of the energy generation process by 25 % and, thus, a non-linear relationship between the power of the generated energy, the speed and direction of wind has been revealed. It should also be noted that the results obtained in the course of this research make it possible to solve the problem of saving electricity in the cities of Republic of Kazakhstan, since so far there are only large-scale wind farms, which is not always available in simple urban conditions. Moreover, the software developed during the study will allow autonomous control and analysis of the behavior of the wind farm, taking into account various weather conditions. In the future, the methods of data analysis will be applied to the data obtained via the process of modeling.

A script for receiving and transmitting real-time data with wind speed and direction sensors has been developed

Keywords: wind energy, Internet of Things, software, real time, Django, process monitoring

Received date 02.09.2022
Accepted date 15.11.2022
Published date 30.12.2022

***How to Cite:** Alina, G., Tashatov, N., Tatkeyeva, G., Bauyrzhanuly, M., Kaibassova, D., Nurtay, M. (2022). Software prototype development for non-centralized objects of wind flow amplification. Eastern-European Journal of Enterprise Technologies, 6 (9 (120)), 80–88. doi: https://doi.org/10.15587/1729-4061.2022.267892*

1. Introduction

The use of software to improve the process of generating electricity at wind farms concerned mainly its individual elements, for example, turbines. Recently, however, scientists have begun to focus on the use of software to analyze the performance of the entire power plant. In connection with the new opportunity to use wind energy to generate energy in urban areas, it became necessary to improve existing wind farms. This area remains unexplored, since initially, it is necessary to have appropriate weather conditions, such as in the city of Astana, where a constant wind flow is observed based on the wind rose and other observations. Therefore, the problem of the need to develop software for reading information, analyzing data, and improving the energy generation process remains relevant.

Nowadays, the world is undergoing modernization of the energy sector and the transition to a new technological model, which includes updating the technological base, increasing production efficiency, as well as ensuring an improvement in the quality of life and the living environment. The key direction of energy modernization is the formation of “green” energy. In Kazakhstan, interest in wind generation is high, but mainly focused on large-scale wind farms. The rapid development of real-time data processing systems has also affected this industry. Thus, today there is a need to introduce innovative approaches to solving the problem of increasing the efficiency of electric power generation using an approach based on IT technologies.

Moreover, the need and relevance of such a study are due to the fact that there is a worldwide trend towards the use of renewable energy sources, since modern energy sources are exhaustible, and besides, unsafe. In other words, the whole world is moving towards decarbonization. Also, the success of the introduction of autonomous energy systems will allow each participant in energy relations to be independent of each other. Therefore, studies that are devoted to finding improvements in green energy have scientific relevance.

2. Literature review and problem statement

In [1], the analysis of wind data is carried out in order to determine the potential location for the application of wind energy. The study was accompanied by the introduction of a new method for assessing the classification of sides based on factors such as wind power and turbulence. However, this study did not have a practical component in the form of software to obtain constant up-to-date data. Another great example [2] discusses the development of an IoT system for monitoring weather parameters and gaseous pollutants in the air using a web application. The paper presented a technique for sending all parameter data. A significant drawback of this work is the lack of application of the results obtained for a specific industry, in this case, energy.

In [3], methods for developing algorithms for monitoring parameters in real time were proposed. It should be noted that in this paper, the platform on which the software part was developed is not cross-platform, so there is a problem of implementation and maintenance.

The paper [4] discusses the development of a system for remote control of agricultural work using a hardware platform based on the Raspberry Pi microcontroller. The result of this study was the invention of AGRIPI, however, it solves a narrow range of problems regarding the problems of agriculture. The paper [5] considers the design of a hydroponic planting process monitoring system based on the Internet of Things, where the ESP32 microcontroller board is used as the main controller. In this case, the parameters that were monitored and collected were the conditions of the hydroponic growing medium. Advances in technology have made IoT adoption easier, such as how the authors of [6] developed a security system with which residential buildings can be monitored. The authors of [7] solved the problem of air conditioning control by proposing a system that can make the fan work on the principle of automatic on/off and motor speed control in accordance with room temperature requirements. In [8], a system for transmitting data from sensors to a microcontroller via a wireless network is proposed.

Assessment of weather conditions is also useful in determining the potential for wind energy. To develop the theory of wind energy generation [9, 10], namely base station energy storage planning, evaluation of blade waste in Germany. The efficiency of renewable energy sources was considered in [11]. On the issue of fully renewable energy in the United States, the work [12] was studied.

In [12], a project for a system for the production of small wind energy based on Arduino is proposed. The data was transmitted to the server with GPRS via mobile data, which could then be accessed in a web application or an Android application. However, the disadvantage of this approach is the insecurity of the GPRS channel without using any encryption. However, due to the long observation period during the simulation, the authors were able to collect a large amount of data, which is subsequently analyzed. In [13], the use of hardware is justified by the need to control the charging voltage of the turbine generator. Moreover, the modified controller is able to determine the charging time of the wind turbine battery. The authors of [14] show by example the use of the websocket protocol for real-time data transfer. The papers highlight the benefits of using websocket technology as a good case for using this method to solve the problem of the authors of this study.

All this allows us to assert that it is expedient to conduct a study on the automation of the wind energy generation process.

3. The aim and objectives of the study

The aim of the study is to develop software for collecting and storing real-time data transmitted from wind speed and direction sensors, as well as prototyping a wind generation system. Thus, it is planned to prepare a digital platform for further modeling the direction of wind flows from the received data to enhance the wind flow itself and design a multi-rotor wind generator.

To achieve this aim, the following objectives are accomplished:

– to simulate the direction of the wind flow and thereby increase the wind flow for a multi-rotor wind generator with horizontal axes in a simulated environment;

– to develop a script for receiving and processing data obtained in real time from information sensors;

– to develop a web application for monitoring wind power data.

4. Materials and methods

There is a model of a custom wind farm, which includes a wind power plant and shield structures. A wind power plant is a canvas with a group of blades on which transmission mechanisms are located. The energy generated from these blades is transmitted through a gearbox and stored in a special battery. Shield structures are designed to increase the wind flow and can significantly increase the efficiency of wind turbines without changing anything in their design.

When developing the software and hardware complex, the main emphasis was placed on ease of development and scalability. The software part was supposed to have the ability to automatically regulate the shield structures of the wind generator, as well as to increase the wind flow in the area of its location in order to obtain maximum current power.

To develop the software and hardware complex, the Arduino Mega 2560 microcontroller was used as a platform for the rapid development of electronic devices. The hardware connection diagram is demonstrated in Fig. 1.

The functional component of the scheme of this complex is described below. The hardware part consists of an Arduino Mega 2560 microcontroller (1), an Ethernet Shield (2), a power supply with a 12 V output (3), a wind direction sensor (4), a wind speed sensor or anemometer (5), 4 TS90M servo-motors (6–9), breadboard (10), electronic display (11).

Initially, the wind speed and direction sensors and servo motors are initialized. After that, a connection was established between the Arduino and the web server via the HTTPS protocol. During initialization, the Ethernet shield was assigned the specified MAC address of the microcontroller and an IP address on the local network.

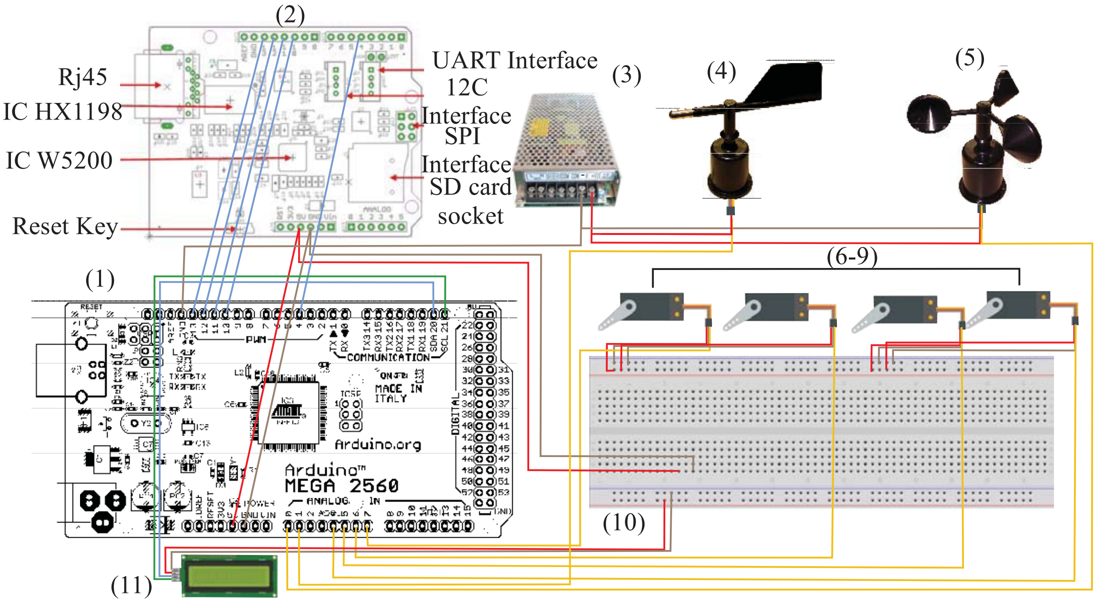


Fig. 1. Hardware connection diagram: (1) – Arduino Mega 2560 microcontroller; (2) – Ethernet Shield; (3) – power supply with a 12 V output; (4) – wind direction sensor; (5) – wind speed sensor or anemometer; (6–9) – 4 TS90M servo motors; (10) – breadboard; (11) – electronic display; UART – Universal Asynchronous Receiver-Transmitter; SPI – Serial Peripheral Interface; SD – Secure Digital Memory Card

Data from the sensors begin to arrive and be read as soon as the connection to the web server is established. For the correct acquisition and processing of data from the sensors, the time interval for taking readings was set equal to 3 minutes for the wind speed sensor [13], and 2 minutes for the wind direction sensor. The sensors are connected to the Arduino microcontroller via analog inputs and are powered by a power supply unit with an output voltage of 12 V [14, 15].

Arduino microcontroller through the analog input A0 receives readings in the form of voltage values on the wind speed sensor in the range from 0 to 5 V. The input data is converted into the actual wind speed according to the following formula:

$$\vartheta_5 = \frac{V_{in} - V_{\min}}{V_{\max} - V_{mm}} * \vartheta_{\max}, \tag{1}$$

where V_{in} – input voltage at the wind speed sensor, $V_{\min}$ – minimum voltage from the anemometer, $V_{\max}$ – maximum voltage, $\vartheta_{\max}$ – maximum wind speed, corresponding to $V_{\max}$.

Through the analog input A1, the microcontroller receives the readings of the wind direction sensor in a similar form. Transformation of input voltage values on this sensor is performed using the C++ map() function, which proportionally transfers the value from the current range of values to a new range specified by the corresponding parameters. An example of calling the map() function is illustrated below:

map(VaneValue, 0, 1023, 0, 360),

where VaneValue – a variable that stores the read value from the wind direction sensor from the analog input A1 of the Arduino microcontroller.

Servo motors were installed on 4 sides of the wind turbine layout and were controlled in a standard way using a software script to rotate to the desired angle. The servo motor control logic is embedded in a special script written in C++ for the Arduino sketch. The layout of the servo drives for the panel structures of the wind turbine is shown in Fig. 2.

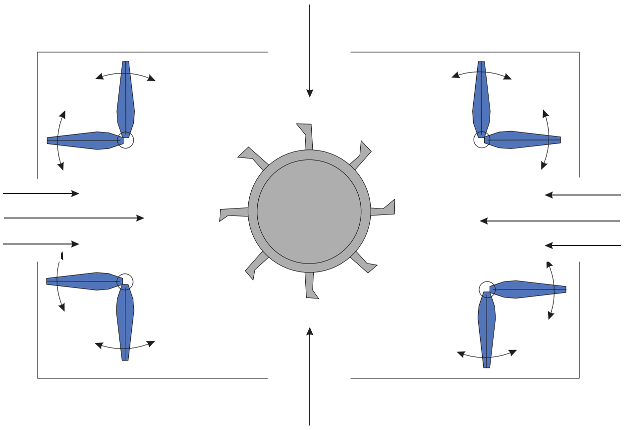


Fig. 2. Layout of the wind turbine

Wind speed and direction data were simultaneously transmitted to the web server via the HTTPS GET requests as string parameters, after which the web server starts processing them in real time [16, 17]. The data transfer scheme is shown in Fig. 3.

The dataflow diagram is shown in Fig. 4.

The process diagram is shown in Fig. 5.

To store the incoming readings, a relational database was developed with the PostgreSQL database management system, consisting of two tables: indications, rotations [18]. The structure of the database tables is shown below in Tables 1, 2.

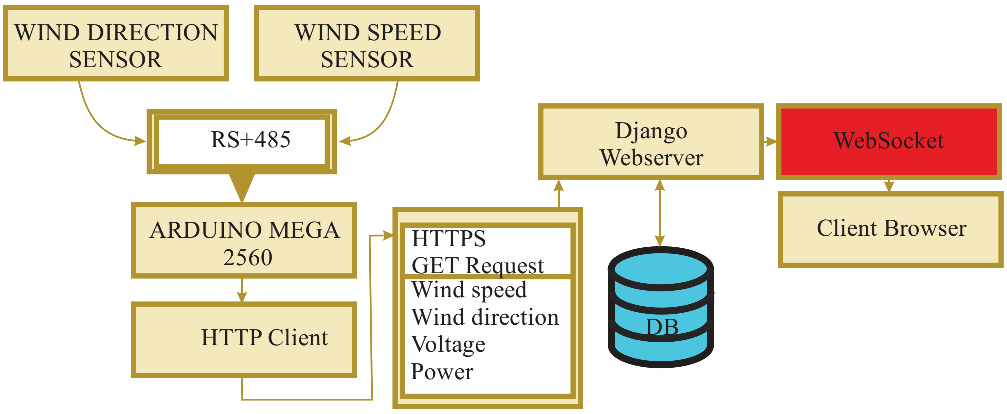


Fig. 3. Data transfer scheme

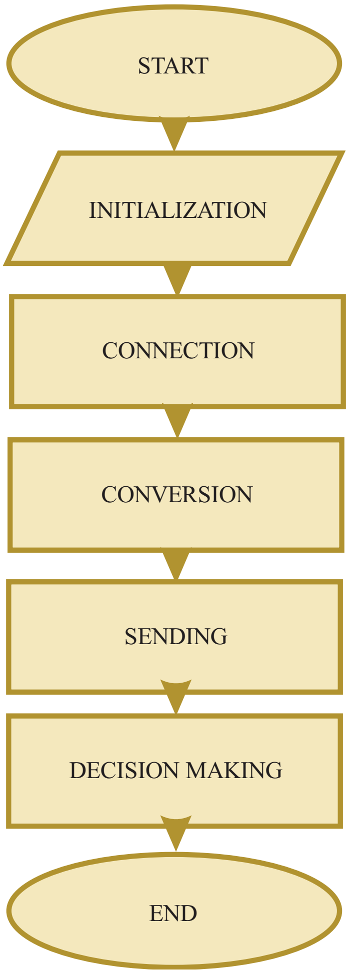


Fig. 4. Dataflow diagram

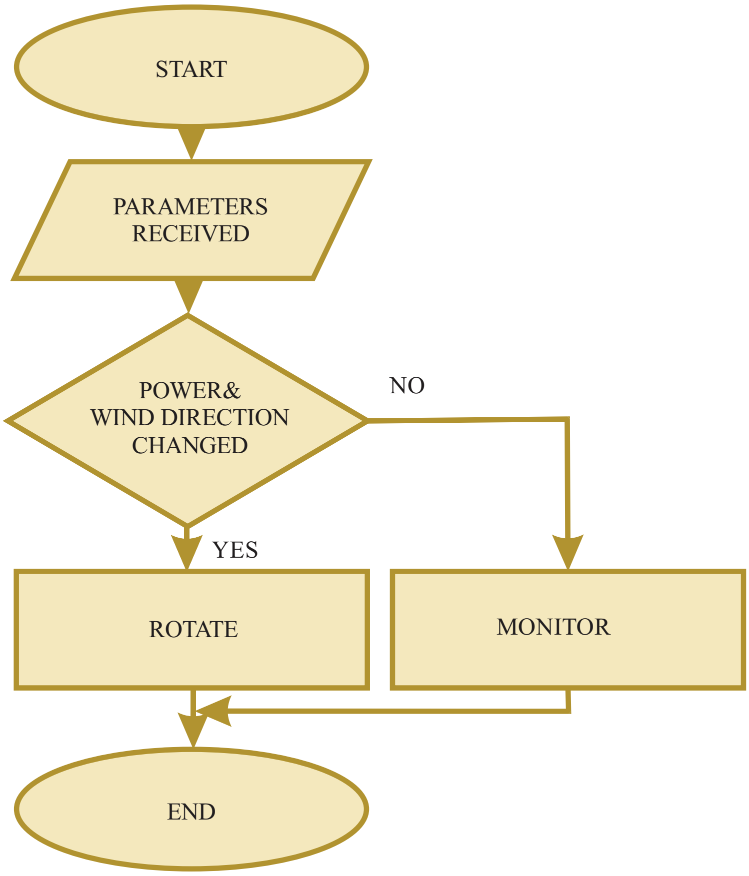


Fig. 5. Process diagram

Table 1

Structure of the indications table

Field	Datatype
Id	Integer, Primary key
wind_speed	Float
wind_direction	Integer
Power	Float
Voltage	Float
date_time	Datetime

Table 2

Structure of the rotations table

Field	Datatype
Id	Integer, Primary key
servo_motor_1	Integer
servo_motor_2	Integer
servo_motor_3	Integer
servo_motor_4	Integer
indication_id	Integer, Foreign key

According to the direction and speed of the wind, the positions of the shields were changed by rotating the servo motors at a certain angle. It is important to note that in this study, it was data collection that was carried out, so the angle of rotation of each servo was experimental. The decision to rotate the desired servo came from the direction of the wind. During the rotation process, the amount of power generated by the wind turbine was also taken into account. The rotation was gradual with a step of 10 degrees and a time interval of 10 seconds. If within 10 seconds the generated power increased by the corresponding amount, then the rotation continued, and vice versa, with a decrease in power, the servo returned to its original position.

The web application backend was developed on the Django Web Framework of the Python programming language [19, 20]. During the development, the Model-Template-View (MTV) pattern was used, which implies the division of the project into layers of business logic, data access, and the client part.

Data monitoring was carried out in real time using the Websockets protocol. The monitoring function logic is developed using the Python asyncio library. The web application implements the ability to visualize parameters in the form of tables and graphs received from sensors.

5. Results of modeling and software development

5. 1. Simulation results of hardware implementation

The shield structures of the wind generator were attached to the Arduino microcontroller and controlled with it. The angle of rotation of each shield structure was determined by the indicators of wind speed and direction, which were given by the corresponding sensors. Below are the test results for each device.

The first test was done to find out if the main controller circuit, in this case Arduino Uno with Ethernet Shield, can function properly or not. The test included a test of connecting the Arduino to the computer's serial port and the added Ethernet Shield on the Arduino board. The goal is to add the Arduino's ability to connect to computer networks.

To test the Arduino connection, a USB cable is connected from a computer to the Arduino series, then the Arduino IDE software is launched. If the Arduino is connected to a computer, the Arduino IDE provides a serial port connection on the computer that is using the Arduino. For the Ethernet Shield test, the module is connected by placing the Ethernet Shield PCB on the Arduino board. An RJ-45 UTP cable, crossed over, connected the local host computers to the Ethernet Shield. The Arduino IDE was loaded with example programs from the Ethernet Library, the IP address of the computer with the Ethernet Shield set to one network. If the Arduino is working properly, the Arduino IDE monitor will show the connection, and the monitor's serial width will show the port used by the Arduino, as shown in Fig. 6.

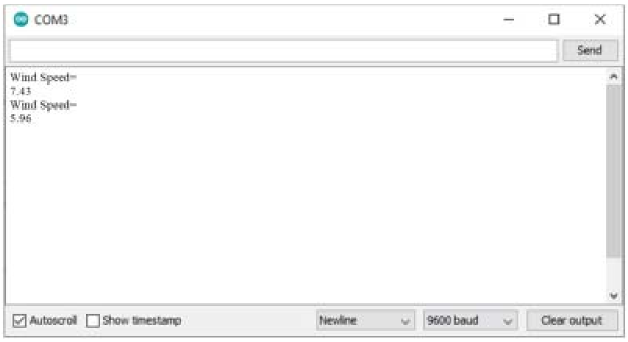
Fig. 6 shows that DHCP was successfully initialized and the microcontroller automatically received its IP address on the local network. The next step is to connect to the test server.

To test the performance of the wind speed sensor, monitoring of the COM port was enabled and the converted wind speed value was output from the analog input in the console. The result is illustrated in Fig. 7.

After a successful check of the wind speed sensor, the device was calibrated.

```
Initialize Ethernet with DHCP:
DHCP assigned IP 172.16.62.63
connecting to 172.16.62.56
connected to 172.16.62.56
0.00
0
```

Fig. 6. Connection display



```
Wind Speed=
7.43
Wind Speed=
5.96
```

Fig. 7. Wind speed sensor test

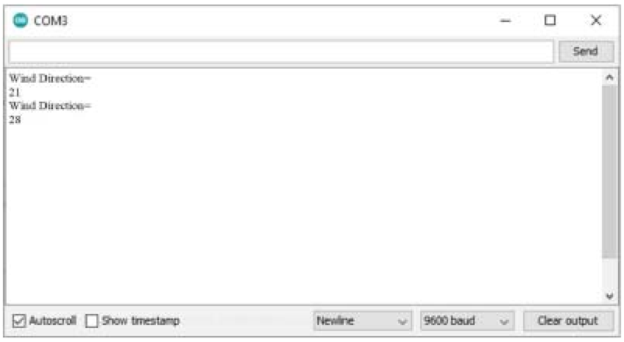
Wind direction was tested by connecting a wind direction sensor to the analog input A1 of the Arduino microcontroller. The script for receiving data from this sensor is shown in the program code below:

```
int VaneValue=analogRead(A1);

int Direction=map(VaneValue, 0, 1,023, 0, 360);

Serial.println(Direction).
```

The result of testing the wind direction sensor is shown in Fig. 8.



```
Wind Direction=
21
Wind Direction=
28
```

Fig. 8. Wind direction sensor test

As shown in the code, the range of values from 0–1,023 scales proportionally to the range from 0 to 360, which corresponds to the direction of the wind, where 0 is set to north.

5. 2. Script test results

In the process of work, a C++ script for receiving and processing data has been written. To test the results of the script, below is a fragment of the program code of the script with comments.

```
    // initially, create test suite object
    TestSuite suite;
    connectionByDHCP() {
        if (Ethernet.begin(mac) == 0) {
                Serial.println(“Failed to configure Ethernet using DHCP”);
                // check for Ethernet hardware present
                if (Ethernet.hardwareStatus() == EthernetNoHardware) {
                    Serial.println(“Ethernet  shield was not found.  Sorry, can’t run without hardware. :(“);
                    while (true) {
                        delay(1); // do nothing, no point running without Ethernet hardware
                    }
                }
                if (Ethernet.linkStatus() == LinkOFF) {
                    Serial.println(“Ethernet cable is not connected.”);
                }
                // try to congifure using IP address instead of DHCP:
                Ethernet.begin(mac, ip, myDns);
                return 0;
        } else {
                Serial.print(“ DHCP assigned IP”);
                Serial.println(Ethernet.localIP());
                return 1;
                }
    }
    // then create a test called ‘connectionByDHCP’ in the test suite

    test(connectionByDHCP) {
        assertEqual(connectionByDHCP(), 0);
    }

    void loop() {
        // run test suite, printing results to the serial port
        suite.run();
    }
```

As shown in the code, unit testing was performed to assign an IP address through DHCP. The assertEqual() function provides a check of the device’s performance and the correctness of the program code. Moreover, all the expected results were obtained for the remaining functions, the code works successfully.

5. 3. Web application test and visualization of simulation results

The web application was tested using the Django Web Framework testing tool. Unit tests and integration tests were used for testing. The form of the user interface of the web application is shown in Fig. 9. This software was developed as a prototype, the HTTPS protocol with a generated SSL certificate was used to ensure the security of data transfer.

The reading data from the wind speed and direction sensors are successfully stored in the database and displayed on the user interface of the web application. Moreover, the web application reveals the angle of rotation of each switchboard installation, which allows you to visualize and monitor the process of wind generation.

The results of the experiment carried out during the study are given in the form of graphs of readings taken from the sensors.

Fig. 10 shows how the wind speed changes with the use of shields.

The graph shows the dynamics of data transmission through the anemometer. The red lines on the graph show the values of this parameter. Obviously, the indicator shows an increase in wind speed. Consequently, the use of shield structures has made it possible to significantly increase the wind speed, which is a key factor in increasing power.

Fig. 11 reveals the difference in output voltage.

Fig. 12 shows a power graph calculated using the formula:

$$P = U * I, \tag{2}$$

where P – power; U – voltage; I – current strength (amperage).

The current strength was measured with an ammeter.

Fig. 10–12 demonstrate the statistics of the voltmeter readings without the use of wind amplification shields (blue line) and with the use of shields (red line).

In total, approximately 350 thousand records were collected in the database during the simulation. A fragment of the collected data is shown in Tables 3, 4.

Table 3

A fragment of the data obtained without the use of shield structures

Time, hh:mm	Wind speed, m/s	Wind direction	Power, W	Voltage, V
10:00	0.6	SW	5.8	2.0
10:30	0.3	SW	1.1	0.6
11:00	0.8	S	5.3	1.7
11:30	1.0	S	11.7	3.0
12:00	2.1	SW	2.6	0.9
12:30	2.5	S	3.8	1.5
13:00	1.1	SW	7.6	2.1

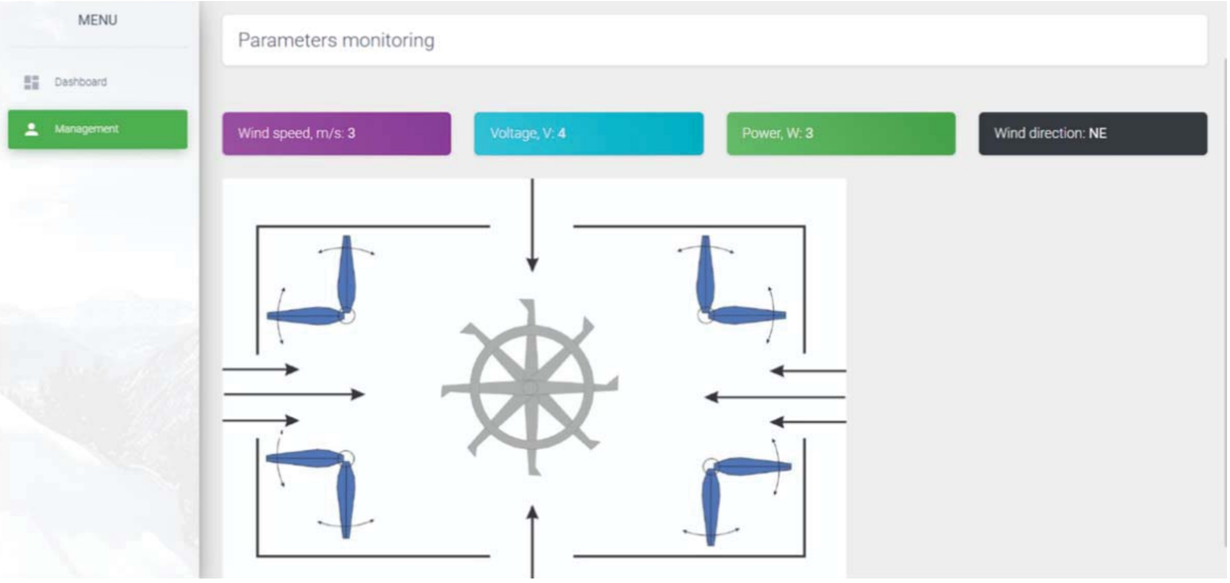
Table 4

A fragment of the data obtained with the use of shield structures

Time, hh:mm	Wind speed, m/s	Wind direction	Power, W	Voltage, V
10:00	1.3	SE	6.7	2.3
10:30	0.6	S	3.0	1.6
11:00	1.5	SE	6.2	2.0
11:30	2.9	SE	12.1	3.1
12:00	1.4	S	6.7	2.3
12:30	1.5	S	3.8	1.5
13:00	2.4	SE	8.6	2.4

The following is an interpretation of the results based on the data obtained.

According to the data obtained, it can be said that the generated power increases by about a quarter of that measured without the use of side shield structures. This, first of all, allows us to say that there is a statement about the effectiveness of the use of panel structures for generating electrical energy using wind.



Parameters monitoring

Wind speed, m/s: 3

Voltage, V: 4

Power, W: 3

Wind direction: NE

Fig. 9. Web application test

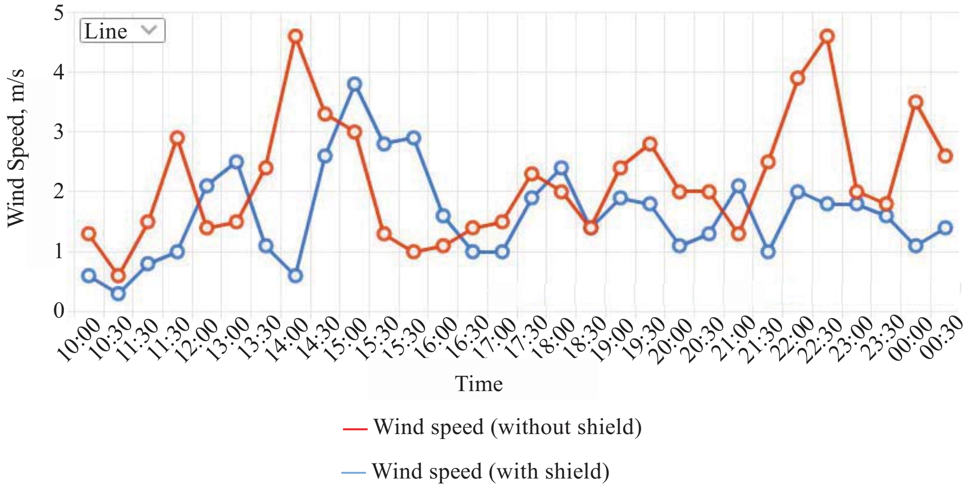


Fig. 10. Efficiency of using wind amplification shields

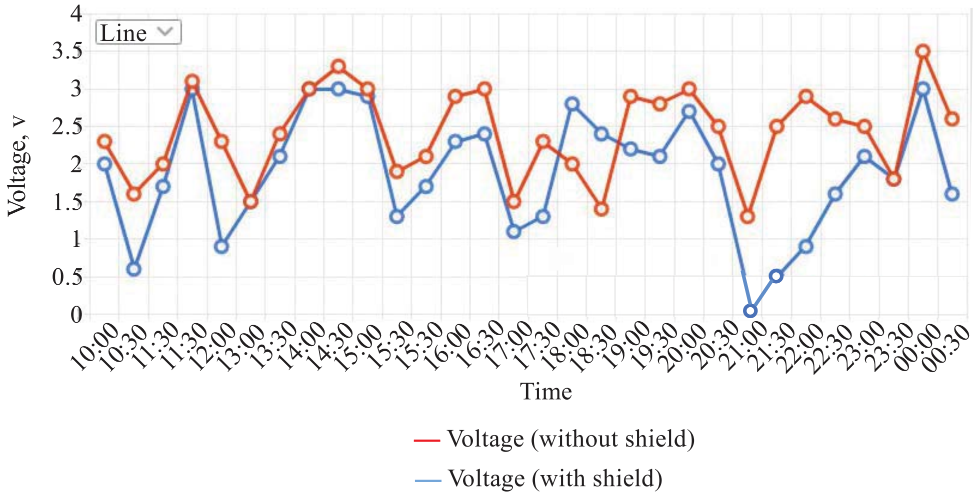


Fig. 11. Output voltage difference

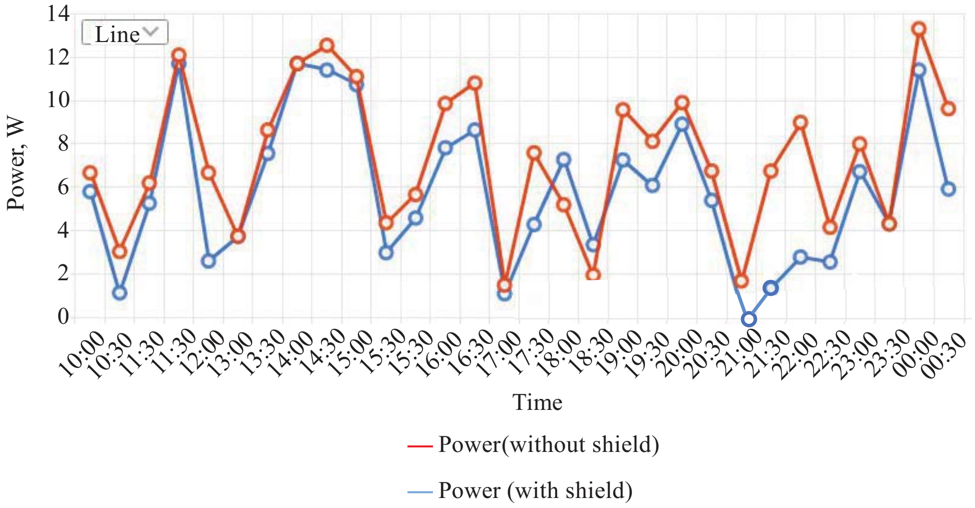


Fig. 12. Power calculation

6. Discussion of experimental results of software development

The prerequisite for the implementation of this study was the creation of a wind turbine based on panel plants to increase the efficiency of electricity generation. An undoubted advantage over existing developments is the ability of the installation to work autonomously. For instance, these systems will provide lighting for city buildings, and thereby reduce the cost of urban lighting, as well as energy supply for industrial enterprises for their own needs. There are no guidelines, regulations or specific information about wind generation in urban environments until these days. The study will solve the problem of identifying the presence of patterns and relationships between the values of wind speed and direction and the power of the generated energy.

Comparison of the results of the study was carried out in several parameters: the current power generated by the energy generation process, the voltage on the anemometer to assess the increase in wind speed and the wind speed itself, respectively.

As shown in the graph of fluctuations in the power of the effective part, it can be noted that the use of switchboard installations has given a significant increase in the power generated during energy generation (Fig. 12). In view of the sharp fluctuations in wind speed, this indicator cannot be kept at a constant value or range of values. Such a jump in power values can be regulated by improving the design of the shields and the wind generator itself. For comparison, let’s take all three graphs (Fig. 10–12) of changes in values during the study with and without shields.

At a wind speed of 0.6 m/s, the power was 5.8 W, while when using a shield installation, the wind speed more than doubled, reaching a value of 1.3 m/s. At the same time, the generated power was 6.7 W (Table 4). However, even when the wind speed dropped to 0.3 m/s, the generated power was only 1.1 W. Similarly, for the case with panel installations, the values of wind speed and power were 0.6 m/s and 3 W. In some cases, shield installations had the opposite effect, reducing the wind speed required for generation (approximately 4.8 m/s versus 1.9 m/s at 22:30) (Fig. 10). Thus, the wind speed in the area of the wind generator could be increased slightly. This example shows the instability, first of all, not only of the wind speed, but also of the operation of the wind generator itself. This is mainly due to the imperfection of the design of the wind generator, since the error in the readings of the measuring instruments used in the study is minimal. This shortcoming can be eliminated by improving the design of both the switchboard installations and the wind generator in accordance with the laws of aerodynamics.

It should be noted that the experiments were carried out under conditions as close to natural as possible. The wind flow model was simulated in such a way that wind flows were generated according to the normal distribution law. This fact can be interpreted as a shortcoming of the study, which can be eliminated by setting the layout in real environmental conditions.

Moreover, to control panel structures, it is necessary to develop an algorithm based on the accumulated data during the simulation process. It is assumed that to solve such a problem it is advisable to use elements of fuzzy logic.

7. Conclusions

1. First of all, in order to test the effectiveness of the use of shield structures, two test models of a wind turbine were modeled. In both layouts, hardware based on the Arduino microcontroller and wind speed and direction sensors were introduced. Some cases with obtaining values of wind speed require additional consideration of the issue of improving the forms of elements of the wind generator. However, in this context, it is the acquisition and processing of simulation results that are important. Regarding this, it can be said that the comparison of the results of the two cases (with and without shield structures) is quite fair and is in favor of the first one, respectively. The developed prototype of the software, first of all, is intended to reveal and prove just that.

2. The script was developed for receiving and processing data received in real time. Unit testing was carried out and intervals for polling sensors were configured. During the script execution, the parameters of wind speed and direction were monitored. Moreover, the parameters of the generated

power were observed, according to which the efficiency of electricity generation was evaluated.

3. A database was designed to store the obtained readings, and a web application was developed that makes it convenient to visualize and analyze the data. Although the creation of such system is not something new, the innovation in this case is still the integration of hardware and software with elements such as shield structures for improving the process of generating energy by a wind generator. Thus, data monitoring has shown a significant improvement in energy generation indicators such as wind power and speed. The results of the study show that with the correct regulation of the wind flow, the efficiency of the energy generation process will increase. Thanks to the data obtained during the modeling process, it will be possible to carry out data analysis in the future to identify patterns and dependencies between the values of generated power, wind speed and direction.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

Data cannot be made available for reasons disclosed in the data availability statement.

References

1. Megantoro, P., Pramudita, B. A., Vigneshwaran, P., Yurianta, A., Winarno, H. A. (2021). Real-time monitoring system for weather and air pollutant measurement with HTML-based UI application. Bulletin of Electrical Engineering and Informatics, 10 (3), 1669–1677. doi: https://doi.org/10.11591/eei.v10i3.3030
2. Tomilova, N., Tomilov, A., Kaibassova, D., Kalinin, A., Amirov, A., Nurtay, M. (2022). Digital models of stabilizing the hydraulic mode of heat supply systems. Journal of Theoretical and Applied Information Technology, 100 (2), 322–335. Available at: http://www.jatit.org/volumes/Vol100No2/3Vol100No2.pdf
3. Navulur, S., S. C. S. Sastry, A., N. Giri Prasad, M. (2017). Agricultural Management through Wireless Sensors and Internet of Things. International Journal of Electrical and Computer Engineering (IJECE), 7 (6), 3492. doi: https://doi.org/10.11591/ijece.v7i6.pp3492-3499
4. Megantoro, P., Prastio, R. P., Kusuma, H. F. A., Abror, A., Vigneshwaran, P., Priambodo, D. F., Alif, D. S. (2022). Instrumentation system for data acquisition and monitoring of hydroponic farming using ESP32 via Google Firebase. Indonesian Journal of Electrical Engineering and Computer Science, 27 (1), 52. doi: https://doi.org/10.11591/ijeecs.v27.i1.pp52-61
5. Chaudhuri, T., Nyamati, V., Jayavel, K. (2018). Design and implementation of IoT framework for Home Automation and Monitoring. 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference On. doi: https://doi.org/10.1109/i-smac.2018.8653724
6. Robson, W., Ernawati, I., Nugrahaeni, C. (2021). Design of Multisensor Automatic Fan Control System Using Sugeno Fuzzy Method. Journal of Robotics and Control (JRC), 2 (4). doi: https://doi.org/10.18196/jrc.2496
7. Das, B., Jain, P. C. (2017). Real-time water quality monitoring system using Internet of Things. 2017 International Conference on Computer, Communications and Electronics (Comptelix). doi: https://doi.org/10.1109/comptelix.2017.8003942
8. Peng, Y., Shi, Y., Li, J., Hu, Y. (2022). Optimal Scheduling of 5G Base Station Energy Storage Considering Wind and Solar Complementation. 2022 4th Asia Energy and Electrical Engineering Symposium (AEEES). doi: https://doi.org/10.1109/aeees54426.2022.9759744
9. Volk, R., Stallkamp, C., Herbst, M., Schultmann, F. (2021). Regional rotor blade waste quantification in Germany until 2040. Resources, Conservation and Recycling, 172, 105667. doi: https://doi.org/10.1016/j.resconrec.2021.105667
10. Bezrukikh, P. P., Bezrukikh, P. P. Jr., Karabanov, S. M. (2020). On the Electricity Generation Balances Worldwide and in Russia. Vestnik MEI, 4 (4), 21–28. doi: https://doi.org/10.24160/1993-6982-2020-4-21-28
11. Butterfield, B. L., Bullen, D. B. (2022). Questioning a Green New Deal–Bulk materials requirements for an all-renewable economy in the United States. Energy Research & Social Science, 86, 102424. doi: https://doi.org/10.1016/j.erss.2021.102424
12. Al Mubarok, A. G., Djatmiko, W., Yusro, M. (2018). Design of Arduino-based small wind power generation system. E3S Web of Conferences, 67, 01006. doi: https://doi.org/10.1051/e3sconf/20186701006
13. Mahmuddin, F., Yusran, A. M., Klara, S. (2017). On the use of an Arduino-based controller to control the charging process of a wind turbine. AIP Conference Proceedings. doi: https://doi.org/10.1063/1.4976284
14. Liu, Q., Sun, X. (2012). Research of Web Real-Time Communication Based on Web Socket. International Journal of Communications, Network and System Sciences, 05 (12), 797–801. doi: https://doi.org/10.4236/ijcns.2012.512083
15. Skvorc, D., Horvat, M., Srbljic, S. (2014). Performance evaluation of Websocket protocol for implementation of full-duplex web streams. 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). doi: https://doi.org/10.1109/mipro.2014.6859715
16. Ferdiansyah, N., Rahayu, D. A., Permala, R. (2019). Comparation of postgresql, mariadb and mongodb capabilities in processing lapan satellite ais data. Pengembangan Iptek Litbangyasa Pesawat Terbang, Roket Dan Satelit. doi: https://doi.org/10.30536/p.siptekgan.2019.v23.23
17. Rubio, D. (2017). Introduction to the Django Framework. Beginning Django, 1–29. doi: https://doi.org/10.1007/978-1-4842-2787-9_1
18. Shyam, A., Mukesh, N. (2020). A Django Based Educational Resource Sharing Website: Shreic. Journal of Scientific Research, 64 (01), 138–152. doi: https://doi.org/10.37398/jsr.2020.640134