

LIGHTING CONTROL MODULE SOFTWARE DEVELOPMENT

Yurii Vizir¹, Olena Chala¹, Svitlana Maksymova¹, Ahmad Alkhalailah²

¹Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Ukraine

²Senior Developer Electronic Health Solution, Amman, Jordan

Abstract: This article describes the development software for a lighting control module based on Arduino Mega. At first main requirements for such software development are considered. These requirements include the following: lighting monitoring; determination of the normative level of illumination; deviation warning; lighting control. A client-server model was used. The client part is implemented in the form of a web interface, and the server part provides data storage and processing.

Key words: Lighting control, Illumination control, Lighting module, Intelligent production, Software, Web-application, Arduino.

Introduction Modern production is rapidly being rebuilt taking into account cyber-physical systems and Industry 4.0 [1]-[12]. Due to the fact that in the modern world the use of energy-saving technologies [13], [14] both in production and in everyday life is gaining momentum, the relevance of lighting control is also increasing.

It should be noted that the use of the principles of the Industry 4.0 concept involves the use of a wide variety of sensors [15]-[17]. Therefore, many sensors are also used to implement such systems. Various methods and algorithms can also be used here [18]-[29].

Light control refers to the regulation of the level of light in a certain space in order to create optimal conditions for work, leisure or other activities. There are several ways to control lighting in different contexts. Among them the following types can be distinguished.

Switches and dimmers. Switches with dimmers – allow you to gradually adjust the brightness of the lighting. Dimmers are special devices for precise control of light brightness.

Automated systems. Motion sensors – turn on the lights when motion is detected and turn them off when people leave the room. Light sensors – adjust the brightness of the light depending on the level of natural light in the room.

Smart home systems. Smart lamps and light bulbs – can be controlled using a smartphone or voice commands.

Lighting schedule programming – systems that automatically turn lights on and off at specific times.

Energy saving technologies. LED lighting – consumes less energy and provides greater flexibility in dimming. Energy saving light bulbs.

Centralized control systems. Smart lighting systems – allow you to integrate lighting control with other smart home systems. Building Management Systems – Large buildings and offices use systems that automatically control all aspects, including lighting.

Solar curtains and blinds. Automatic adjustment – can adjust the interior lighting according to the angle of the sun's rays.

Light control can be adapted to different needs and scenarios, ensuring comfortable conditions while optimizing energy consumption.

This work will consider an automated lighting control system using appropriate sensors and software development for a device developed in [30].

Related works

Due to the extreme relevance of the introduction of energy-saving technologies, many scientists are engaged in research in this area, in particular the problem of lighting control

Let us begin our literature analysis from the paper [31]. In this work authors consider optimal control of environmental conditions that can be used to facilitate the improvement of lettuce plant growth and light use efficiency. This article reviews and discusses the effects of light, air velocity, temperature, relative air humidity, and CO₂ concentration on the growth and development of lettuce plants.

Researchers in [32] propose a low-cost new system and successfully evaluated it by developing three devices, namely the measure and control device for street lights, lighting level measurement device and gateway LoRa network, based on the Arduino open-source electronic platform.

Article [33] proposes a smart lighting control method for indoor environments with both dimmable (controllable) and uncontrollable external light sources.

In [34] the authors present recent and significant state-of-the-art interior lighting system control techniques in commercial buildings. The review focuses on several key research, including sensing technologies, objective functions and constraints, techniques, tools, and



energy performances. A survey trend analysis is presented graphically and the findings are discussed extensively.

Research [35] provide information about the differences, advantages and disadvantages of Web, Thingspeak and Blynk applications for Smart Home devices to turn on and off house lights via the internet using laptops and smartphones.

A novel data-driven model predictive control (MPC) method that is based on the temporal sequential-based artificial neural network is proposed to overcome this challenge using an updated historical occupancy status [36].

Carli, R., & Dotoli, M. in [37] propose a decision-making procedure that supports the city energy manager in determining the optimal energy retrofit plan of an existing public street lighting system throughout a wide urban area.

Paper [38] considers the system that is capable to autonomously adjust street lamps' brightness on the basis of the presence of vehicles (busses/trucks, cars, motorcycles and bikes) and/or pedestrians in specific areas or segments of the streets/roads of interest to reduce the energy consumption.

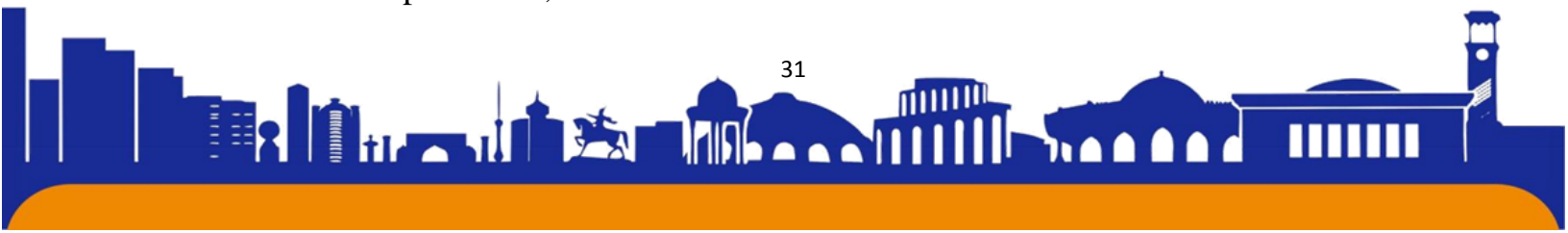
So we see a lot of work in different scientific fields aimed at controlling lighting. Later in this article we will look at the software development for the lighting control module proposed in [30].

Web application development

Proposed software was developed for the developed device presented in [30]. Since the ESP32 microcontroller is used as the controller of the module nodes, a development environment called Arduino IDE version 1.8.1 was chosen for the software development.

An Arduino program was written to control indoor lighting using an Arduino Mega board and an LM393 sensor:

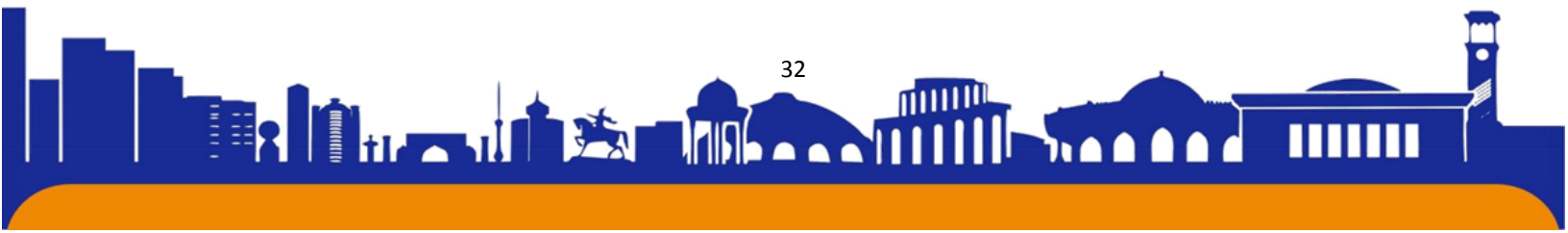
```
// Connecting libraries
#include <BH1750.h>
// Connecting the light sensor
BH1750 lightSensor;
// Connecting pins for controlling lamps
const int lamp1Pin = 2;
const int lamp2Pin = 3;
const int lamp3Pin = 4;
```





```
const int lamp4Pin = 5;
// Illuminance thresholds for turning on/off lamps
const int thresholdOn = 500; //The illumination value for turning on the lamps
const int thresholdOff = 300; //The illumination value for turning off the lamps
void setup() {
  // Light sensor initialization
  lightSensor.begin();
  //Setting the lamp control pins
  pinMode(lamp1Pin, OUTPUT);
  pinMode(lamp2Pin, OUTPUT);
  pinMode(lamp3Pin, OUTPUT);
  pinMode(lamp4Pin, OUTPUT);
void loop() {
  // Getting the illuminance value
  int lightLevel = lightSensor.readLightLevel();
  // Turning on/off the lamps depending on the light level
  if (lightLevel > thresholdOn) {
    digitalWrite(lamp1Pin, HIGH);
    digitalWrite(lamp2Pin, HIGH);
    digitalWrite(lamp3Pin, HIGH);
    digitalWrite(lamp4Pin, HIGH);
  } else if (lightLevel < thresholdOff) {
    digitalWrite(lamp1Pin, LOW);
    digitalWrite(lamp2Pin, LOW);
    digitalWrite(lamp3Pin, LOW);
    digitalWrite(lamp4Pin, LOW);
  }
  delay(100); // Delay before next iteration
```

}) In this program, the light sensor is connected to the Arduino Mega board. Depending on the light value received from the sensor, the lamps will be turned on or off. The brightness threshold values thresholdOn and thresholdOff can be adjusted according to the conditions of use.



This example program uses buttons and lights connected to the corresponding pins on the Arduino Mega. Each button turns the corresponding lamp on or off, as well as the ability to turn off the light completely. The interface of the application is shown in Figure 1.

Lighting Control Lamp 4

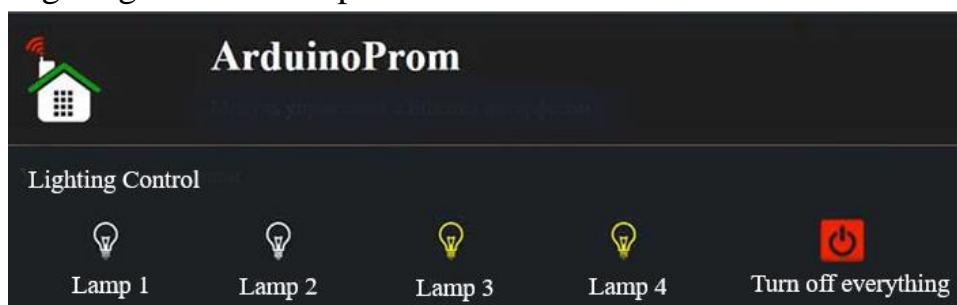


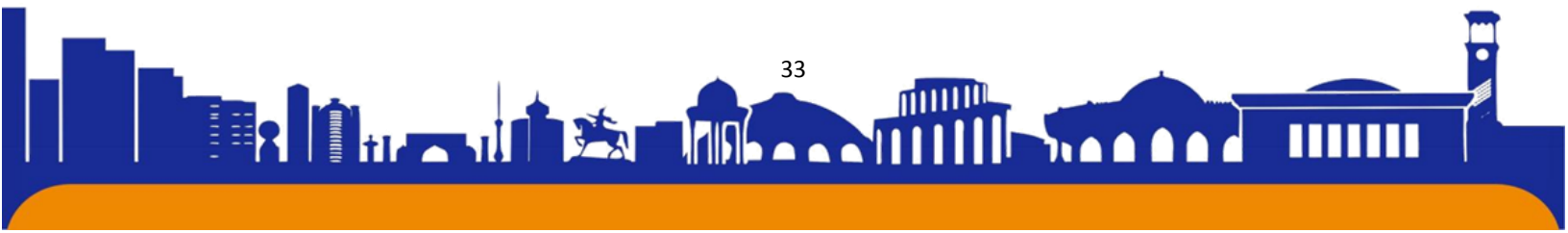
Figure 1: System control application

In order to control and ensure the necessary level of illumination, a web application was developed, shown in Figure 2. It allows illumination monitoring and control in production premises.

The first step in developing a web application was to conduct a requirements analysis. The customer defined the following main functional requirements:

- lighting monitoring: The web application must receive data on the level of illumination from sensors placed in production premises and display this data on a graph or in a table;
- determination of the illumination normative level: the Application should allow setting the illumination normative level in accordance with the requirements of normative documents;
- warning about deviations: the user should receive a warning or a notification if the level of illumination exceeds or falls below the set standard;
- lighting control: The application should allow remote lighting control, turn on or off lights, and adjust their brightness.

After the requirements analysis, the next step was to develop the web application architecture. It was decided to use a client-server model, where the client part will be implemented in the form of a web interface, and the server part will provide data storage and processing.

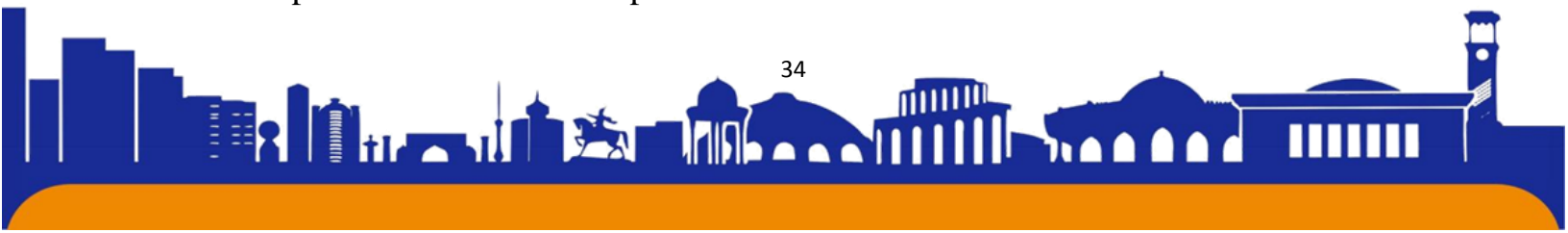




At the stage of web application implementation, modern web technologies and programming tools were used. For the client side, HTML, CSS and JavaScript were used to develop the user interface. For the server part, Node.js was used to create an API for processing requests from the client and interacting with the database.

Example of html code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Web App</title>
</head>
<body>
  <div class="container">
    <h1>Web App</h1>
    <div class="button-container">
      <button class="button">On</button>
      <button class="button">Off</button>
    </div>
    <div class="slider">
      <p class="slider-label"> Brightness level </p>
      <input type="range" min="1" max="5" value="1" step="1" class="slider"
id="slider">
      <div class="slider-markers">
        <div class="marker"></div>
        <div class="marker"></div>
        <div class="marker"></div>
        <div class="marker"></div>
        <div class="marker"></div>
      </div>
      <p id="sliderValue">1</p>
```





```

</div>
</div>
<script src="script.js"></script>
</body>
</html>

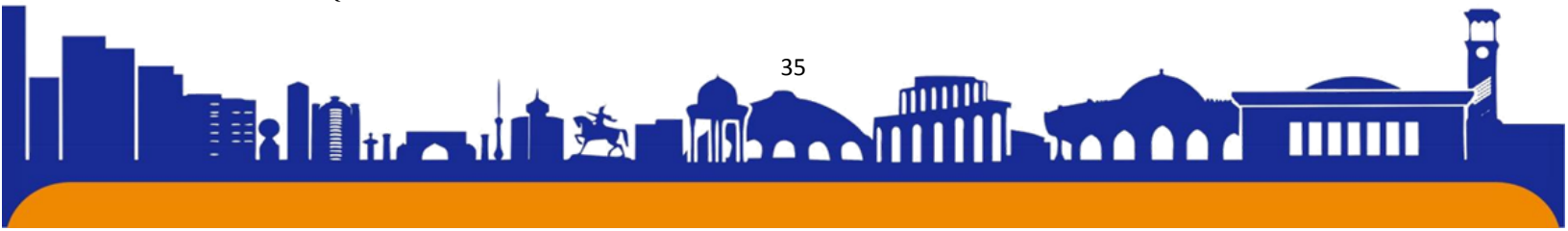
```

An example of css code that is responsible for the on and off buttons and their characteristics:

```

body {
  margin: 0;
  padding: 0;
  font-family: Arial, sans-serif;
  background-color: #f2f2f2;
}
.container {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  height: 100vh;
}
h1 {
  color: #333333;
  margin-bottom: 10px;
  margin-top: 20px;
}
.button-container {
  display: flex;
  justify-content: center;
  margin-bottom: 10px;
  margin-top: 10px;
}
.button {

```

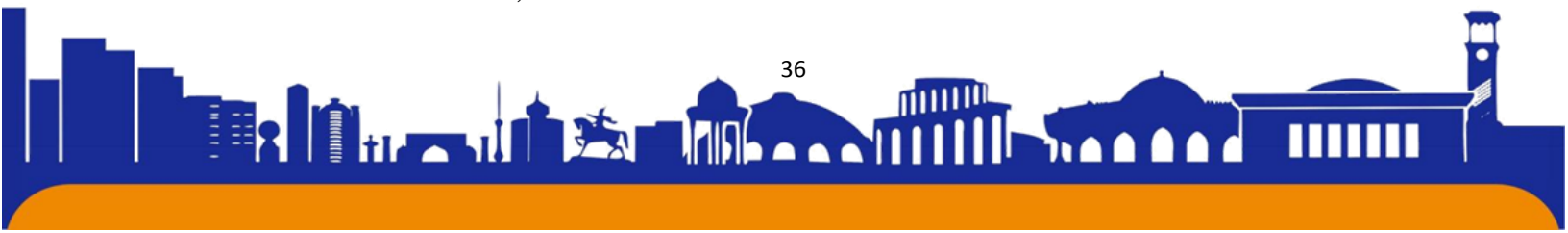




```
background-color: #4CAF50;
color: #ffffff;
border: none;
border-radius: 4px;
padding: 10px 20px;
font-size: 16px;
cursor: pointer;
margin: 0 10px;
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);
transition: background-color 0.3s ease;
}
.button:hover {
background-color: #45a049;
}
```

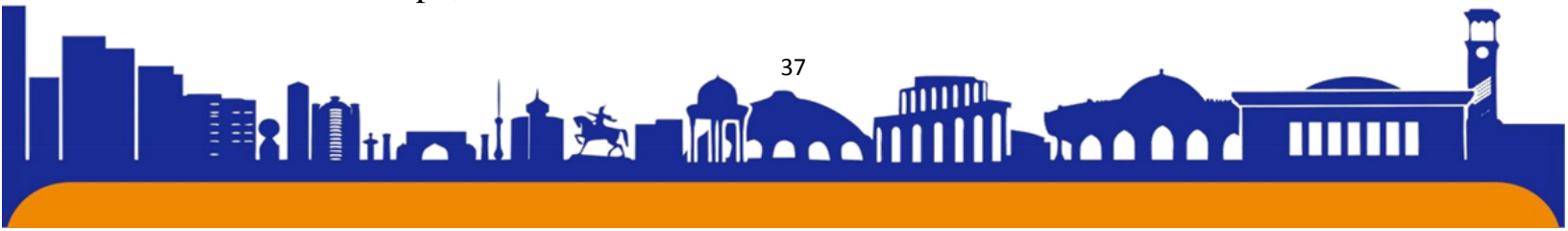
This block of code is responsible for the brightness slider in the app:

```
.slider {
width: 300px;
background-color: #dddddd;
-webkit-appearance: none;
height: 10px;
border-radius: 5px;
outline: none;
margin-top: 10px;
position: relative;
}
.slider::-webkit-slider-thumb {
-webkit-appearance: none;
appearance: none;
width: 20px;
height: 20px;
background-color: #4CAF50;
border-radius: 50%;
```





```
cursor: pointer;
margin-top: -5px;
position: relative;
z-index: 2;
}
.slider::-moz-range-thumb {
width: 20px;
height: 20px;
background-color: #4CAF50;
border-radius: 50%;
cursor: pointer;
margin-top: -5px;
position: relative;
z-index: 2;
}
.slider-markers {
display: flex;
justify-content: space-between;
position: absolute;
width: 100%;
top: 50%;
transform: translateY(-50%);
z-index: 1;
}
.marker {
width: 5px;
height: 5px;
background-color: #888888;
border-radius: 50%;
}
.slider-label {
font-size: 16px;
```



```

color: #333333;
margin-bottom: 10px;
}
    
```

An example of a java script script:

```

const slider = document.getElementById('slider');
const sliderValue = document.getElementById('sliderValue');
slider.addEventListener('input', function() {
    sliderValue.textContent = this.value;
});
    
```

Off

The result of the system management web application is shown in Figure 2.

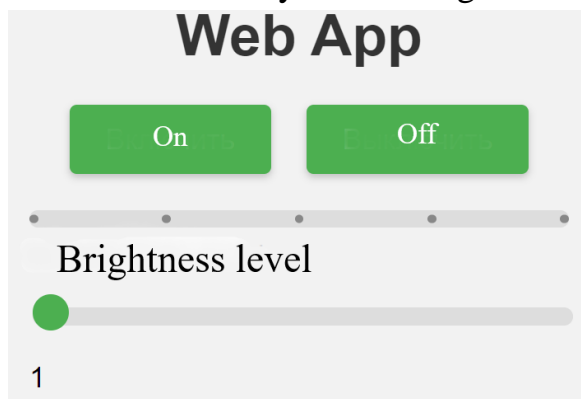


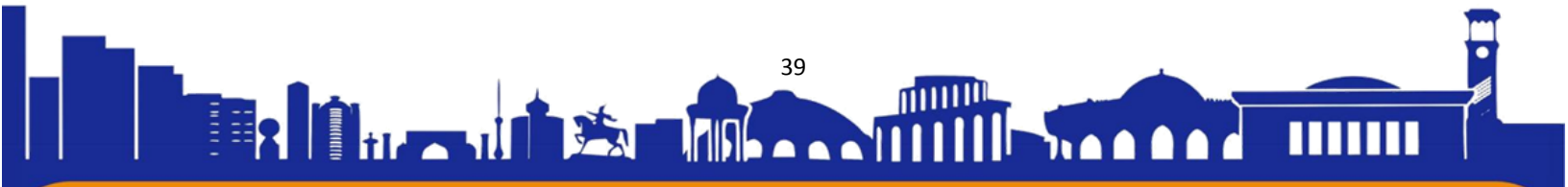
Figure 2: System control web-application

Conclusion

The article discusses the developed software for the device, the development of which is described in the previous article [30]. The main stages of creating such software are given. The requirements for such tasks are considered. The interface of the lighting control system is presented. It was decided to adopt a client-server model. The client part was implemented in a web interface, and the server part ensured the saving and processing of data. For the client part, HTML, CSS and JavaScript were used to develop the client interface. For the server side, Node.js was used.

**REFERENCES:**

1. Lyashenko, V., & et al. (2023). Automated Monitoring and Visualization System in Production Int. Res. J. Multidiscip. Technovation, 5(6), 09-18.
2. Bondariev, A., & et al. (2023). Automated Monitoring System Development for Equipment Modernization. Journal of Universal Science Research, 1(11), 6-16.
3. Maksymova, S., & et al. (2024). The Monitoring System Architecture Development. Journal of Universal Science Research, 2(1), 69-79.
4. Невлюдов, І. Ш., та ін. (2023) Моделі та методи кіберфізичних виробничих систем в концепції Industry 4.0 : монографія. Oktan Print, Prague, 321 с.
5. Nevliudov, I., & et al. (2020). Development of an Architectural Logical Model to Automate the Management of the Process of Creating Complex Cyberphysical Industrial Systems. Eastern-European Journal of Enterprise Technologies, 4 (3-106), 44-52.
6. Attar, H., Abu-Jassar, A. T., Amer, A., Lyashenko, V., Yevsieiev, V., & Khosravi, M. R. (2022). Control System Development and Implementation of a CNC Laser Engraver for Environmental Use with Remote Imaging. Computational intelligence and neuroscience, 2022, 9140156.
7. Abu-Jassar, A. T., Attar, H., Yevsieiev, V., Amer, A., Demska, N., Luhach, A. K., & Lyashenko, V. (2022). Electronic User Authentication Key for Access to HMI/SCADA via Unsecured Internet Networks. Computational Intelligence and Neuroscience, 2022, 5866922.
8. Abu-Jassar, A. T., Al-Sharo, Y. M., Lyashenko, V., & Sotnik, S. (2021). Some Features of Classifiers Implementation for Object Recognition in Specialized Computer systems. TEM Journal: Technology, Education, Management, Informatics, 10(4), 1645-1654.
9. Nevliudov, I., & et al.. (2020). Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis. International Journal of Emerging Trends in Engineering Research, 8(10), 7465-7473.
10. Lyashenko, V., Deineko, Z., & Ahmad, A. (2015). Properties of Wavelet Coefficients of Self-Similar Time Series. International Journal of Scientific and Engineering Research, 6, 1492-1499.
11. Nevliudov, I., Yevsieiev, V., Lyashenko, V., & Ahmad, M. A. (2021). GUI Elements and Windows Form Formalization Parameters and Events Method to Automate the





Process of Additive Cyber-Design CPPS Development. *Advances in Dynamical Systems and Applications*, 16(2), 441-455.

12. Mustafa, S. K., Yevsieiev, V., Nevliudov, I., & Lyashenko, V. (2022). HMI Development Automation with GUI Elements for Object-Oriented Programming Languages Implementation. *SSRG International Journal of Engineering Trends and Technology*, 70(1), 139-145.

13. Zharikova, I., & et al. (2023). Automatic Machine of Plastic Bottles and Aluminum Cans Collection for Recycling. *Journal of Universal Science Research*, 1(11), 169-178.

14. Stetsenko, K., & et al. (2023). Exploring BEAM Robotics for Adaptive and Energy-Efficient Solutions. *Multidisciplinary Journal of Science and Technology*, 3(4), 193-199.

15. Bortnikova, V., & et al. (2019). Mathematical model of equivalent stress value dependence from displacement of RF MEMS membrane. In 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), IEEE, 83-86.

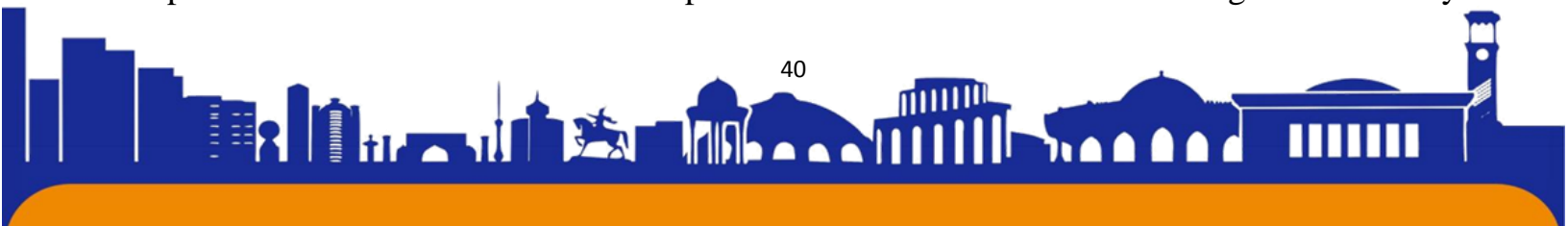
16. Funkendorf, A., & et al. (2019). 79 Mathematical Model of Adapted Ultrasonic Bonding Process for MEMS Packaging. In 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), IEEE, 79-82.

17. Nevliudov, I., & et al. (2018). Modeling MEMS membranes characteristics. In САІР у проектуванні машин. Задачі впровадження та навчання: матеріали XXVI Міжнародної українсько-польської науково-технічної конференції, Видавництво Львівської політехніки, 61-68.

18. Tvoroshenko, I., Ahmad, M. A., Mustafa, S. K., Lyashenko, V., & Alharbi, A. R. (2020). Modification of models intensive development ontologies by fuzzy logic. *International Journal of Emerging Trends in Engineering Research*, 8(3), 939-944.

19. Lyashenko, V., Ahmad, M. A., Sotnik, S., Deineko, Z., & Khan, A. (2018). Defects of communication pipes from plastic in modern civil engineering. *International Journal of Mechanical and Production Engineering Research and Development*, 8(1), 253-262.

20. Lyashenko, V. V., Matarneh, R., Baranova, V., & Deineko, Z. V. (2016). Hurst Exponent as a Part of Wavelet Decomposition Coefficients to Measure Long-term Memory





Time Series Based on Multiresolution Analysis. American Journal of Systems and Software, 4(2), 51-56.

21. Kobylin, O., & Lyashenko, V. (2014). Comparison of standard image edge detection techniques and of method based on wavelet transform. International Journal, 2(8), 572-580.

22. Dadkhah, M., Lyashenko, V. V., Deineko, Z. V., Shamshirband, S., & Jazi, M. D. (2019). Methodology of wavelet analysis in research of dynamics of phishing attacks. International Journal of Advanced Intelligence Paradigms, 12(3-4), 220-238.

23. Sotnik, S., Mustafa, S. K., Ahmad, M. A., Lyashenko, V., & Zeleniy, O. (2020). Some features of route planning as the basis in a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(5), 2074-2079.

24. Lyubchenko, V., Matarneh, R., Kobylin, O., & Lyashenko, V. (2016). Digital image processing techniques for detection and diagnosis of fish diseases. International Journal of Advanced Research in Computer Science and Software Engineering, 6(7), 79-83.

25. Sotnik, S., & Lyashenko, V. (2022). Prospects for Introduction of Robotics in Service. Prospects, 6(5), 4-9.

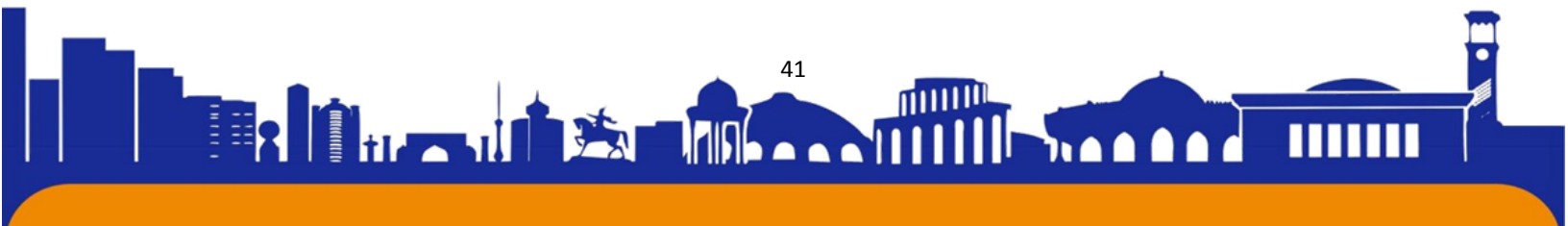
26. Kuzomin, O., Lyashenko, V., Tkachenko, M., Ahmad, M. A., & Kots, H. (2016). Preventing of technogenic risks in the functioning of an industrial enterprise. International Journal of Civil Engineering and Technology, 7(3), 262-270.

27. Boboyorov Sardor Uchqun o'g'li, Lyubchenko Valentin, & Lyashenko Vyacheslav. (2023). Image Processing Techniques as a Tool for the Analysis of Liver Diseases. Journal of Universal Science Research, 1(8), 223-233.

28. Lyashenko, V., Zeleniy, O., Mustafa, S. K., & Ahmad, M. A. (2019). An advanced methodology for visualization of changes in the properties of a dye. International Journal of Engineering and Advanced Technology, 9(1), 711-7114.

29. Sotnik, S., & et al.. (2022). Analysis of Existing Influences in Formation of Mobile Robots Trajectory. International Journal of Academic Information Systems Research, 6(1), 13-20.

30. Vizir, Y., & et al. (2023). Lighting Control Module Development. Journal of Universal Science Research, 1(12), 645-657.





31. Ahmed, H. A., & et al. (2020). Optimal control of environmental conditions affecting lettuce plant growth in a controlled environment with artificial lighting: A review. *South African Journal of Botany*, 130, 75-89.
32. Sánchez Sutil, F., & Cano-Ortega, A. (2020). Smart public lighting control and measurement system using LoRa network. *Electronics*, 9(1), 124.
33. Seyedolhosseini, A., & et al. (2020). Daylight adaptive smart indoor lighting control method using artificial neural networks. *Journal of Building Engineering*, 29, 101141.
34. Wagiman, K. R., & et al. (2020). Lighting system control techniques in commercial buildings: Current trends and future directions. *Journal of Building Engineering*, 31, 101342.
35. Artiyasa, M., & et al. (2020). Comparative Study of Internet of Things (IoT) Platform for Smart Home Lighting Control Using NodeMCU with Thingspeak and Blynk Web Applications. *FIDELITY: Jurnal Teknik Elektro*, 2(1), 1-6.
36. Jin, Y., & et al. (2021). A data-driven model predictive control for lighting system based on historical occupancy in an office building: Methodology development. In *Building Simulation*, Tsinghua University Press, 14, 219-235.
37. Carli, R., & Dotoli, M. (2020). A dynamic programming approach for the decentralized control of energy retrofit in large-scale street lighting systems. *IEEE Transactions on Automation Science and Engineering*, 17(3), 1140-1157.
38. Gagliardi, G., & et al. (2020). Advanced adaptive street lighting systems for smart cities. *Smart Cities*, 3(4), 1495-1512.

