Electronics and Information Technologies, 2026, 35, 185–204
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.35:004.5:004.62:681.5

OPTIMIZATION OF AN AUTOMATED ARDUINO-BASED SYSTEM FOR CHARACTERIZATION OF DISCRETE OPTOELECTRONIC DEVICES

Mykhailo Rusyniak , ***Yuriy Korchak**** , ***Bohdan Kuzhel***
Department of Optoelectronics and Information Technology
Ivan Franko National University of Lviv
107 Gen. Tarnavskoho Street, Lviv, 79017, UKRAINE

Rusyniak, M., Korchak, Yu., Kuzhel, B. (2026). Optimization of an Automated Arduino-based System for Characterization of Discrete Optoelectronic Devices. *Electronics and Information Technologies*, 35, 185–204. https://doi.org/10.30970/eli.35.14

ABSTRACT

Background. Qualitative investigation of the electrical and optical characteristics of optoelectronic components is an essential part of designing, developing, and testing optoelectronic systems. For this purpose, an automated system for measuring the characteristics of optoelectronic devices was developed based on an Arduino UNO microcontroller, featuring programmable voltage control via digital potentiometers and data transfer to a computer for visualization. Additionally, the system was optimized to enhance its performance.

Materials and Methods. A fully functional automated system for measuring the voltage–current and optical characteristics of optoelectronic devices has been developed and implemented based on the Arduino UNO microcontroller. The hardware part of the system is built using MCP4131 digital potentiometers and an integrated GY-302 light sensor. The desktop application was developed in Python, utilizing the Tkinter, pySerial, Matplotlib, and Threading libraries. Adaptive system optimization was carried out using the finite difference method to compute the corresponding local derivatives.

Results and Discussion. Experimental investigation of the optimized system's performance demonstrated a significant increase in efficiency while maintaining high-quality characteristics. The developed adaptive optimization system showed a stable acceleration of 2.4-4.3 times, depending on the type of component. During the experiments, no cases of missing critical regions or significant distortion of characteristic curves were observed. This confirms the reliability of the developed zone-classification system and the correctness of the selected adaptive algorithm parameters.

Conclusion. The results demonstrate the feasibility of creating an efficient and accessible system for studying the characteristics of optoelectronic devices based on modern microcontroller platforms. The developed adaptive optimization system proved effective under real operating conditions, making it a valuable tool for both educational and practical applications in the research and development of optoelectronic devices.

Keywords: current–voltage characteristic, finite difference method, Arduino microcontroller, LED, photodiode, digital potentiometer

INTRODUCTION

Over the past decades, optoelectronics has evolved into one of the most promising branches of modern electronics. It encompasses a wide range of applications in lighting engineering, displays, solar energy systems, fiber-optic communication channels, sensor technologies, and the like. This popularity is driven by the ability of optoelectronic devices

© 2026 Mykhailo Rusyniak et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

to efficiently convert light and electrical energy into one another, as well as by the high information capacity and energy density of the optical channel, which opens up wide opportunities for technical solutions across various fields [1, 2].

Optoelectronic systems often include discrete emitting and photodetector devices. The electrical and optical characteristics of these elements significantly affect the efficiency and stability of the entire electronic device. Therefore, precise investigation of their parameters is a necessary component during the design, development, and testing of optoelectronic systems. However, measuring such characteristics by traditional methods typically requires fairly expensive laboratory equipment. In addition, the manual data collection process is labor‑intensive, inefficient, and not always accurate. Even with process automation, traditional approaches with a fixed scanning step lead to an excessive number of measurements in predictable regions of the characteristic, making the process time‑inefficient.

Given the accessibility, ease of use, and functionality of modern microcontrollers, an automated experimental setup has been developed for investigating the characteristics of discrete optoelectronic components using the Arduino UNO [3, 4]. Arduino enables the creation of simple yet efficient data acquisition and processing systems, making it an ideal tool for both educational purposes and basic laboratory research. The system provides programmable control of voltage and current applied to the device under test via digital potentiometers, allows measurement of electrical parameters using the microcontroller's built-in capabilities, and facilitates data transfer to a computer for visualization and analysis. The approximate cost of this system (excluding the PC) is up to $50.

To address the problem of redundant measurements in linear and predictable regions of a device's characteristic under a fixed voltage step; an adaptive measurement optimization system was developed. This system is based on numerical analysis techniques and dynamic classification of characteristic regions. It utilizes the finite difference method to calculate local derivatives [5], which are then used to classify the current segment of the characteristic and dynamically adjust the scanning step size.

MATERIALS AND METHODS

The development of a compact, software-integrated system for investigating the electrical and optical characteristics of optoelectronic components in laboratory conditions involves both hardware and software design [6].

The hardware design includes the electrical circuit connecting optoelectronic elements to the Arduino Uno, ensuring device safety and measurement reliability. The cloud-based platform EasyEDA was used for circuit and printed circuit board (PCB) design, offering schematic editing, PCB layout, and component simulation. EasyEDA facilitates component placement, connection verification, signal and power trace design, and ergonomic layout assessment, while enabling seamless export to formats suitable for PCB fabrication.

The prototype also incorporates a physical structure with optimally arranged components and adjustable lighting to enhance measurement accuracy and minimize external influences. Two controllable voltage sources allow applying both positive and negative voltages from −7 to +7 V, while current is measured via Arduino's analog-to-digital converter (ADC) using a shunt resistor. Illumination monitoring is provided by an integrated GY-302 lux meter module.

Arduino Uno serves as the central controller, acquiring sensor data, controlling voltage, and transmitting measurements to a computer via a serial interface. A Python-based program automates experiments, performs data processing and calibration, and visualizes results. It also coordinates voltage settings, lighting control, and measurement sequencing, enabling fully automated experimental workflows.

This integrated hardware-software approach provides precise control and analysis of electrical and optical parameters, offering a versatile platform for educational, laboratory, and research applications. The system allows rapid reconfiguration for different experimental scenarios, combining measurement accuracy with user-friendly operation and automation.

The software component of the prototype is implemented as a single Python file that integrates all functional elements of the system. Despite this unified code structure, the program is logically divided into two main blocks: the graphical user interface (GUI) and the hardware interaction module. This approach facilitates organized code, improves readability, and separates logic from data presentation [7].

The GUI block is responsible for displaying control elements and experimental results. It includes buttons for connecting the Arduino, calibrating the system, selecting the optoelectronic device, and starting the automated measurement process. Additionally, the GUI presents results in the form of graphs and tables. The visual interface not only provides informative output but also enables interactivity: users can adjust experiment parameters in real time, monitor measurements, and, if necessary, restart individual stages.

The second block manages hardware interaction, specifically with the Arduino microcontroller. This module implements communication via the serial port, allowing commands to control voltage sources, acquire data from ADCs and sensors, and perform prototype calibration. All operations are automated: once the experiment begins, the Python program controls the voltage applied to the device under test, reads measured current and illumination values, and transfers the data for processing and visualization.

To enhance GUI responsiveness and efficiency, multithreading is employed: one thread handles data processing and Arduino communication, while a second thread updates the interface and plots graphs in real time. This approach prevents program lag and ensures immediate visibility of experimental results, even during computations.

Several key Python libraries are utilized to handle data acquisition, prototype control, and result visualization (Tkinter, pySerial, Matplotlib, Threading), each performing a specific function and simplifying software implementation.

Tkinter is the standard Python library for creating graphical user interfaces (GUI) [8], allowing the development of windowed applications with buttons, input fields, sliders, and graphical widgets. Tkinter facilitates rapid integration of control elements without relying on additional graphical frameworks.

pySerial is a library for working with serial interfaces (COM ports) [9], enabling data exchange between the computer and the Arduino microcontroller via USB.

Matplotlib is a library for plotting and visualizing data [10], used for visually presenting experimental results. It supports line, scatter, histogram, and other types of plots, making data analysis more intuitive [11, 12].

Threading is a Python library for implementing multithreading [13], allowing simultaneous execution of multiple tasks without blocking the user interface. In this system, threading enables parallel data acquisition from the Arduino, signal processing, and real-time graph updates.

The combination of these libraries provides a fully functional software component, covering experiment control, data acquisition, processing, and visualization. Using Python and its libraries ensures system flexibility, scalability, and suitability for further expansion or integration with other sensor systems. By combining modularity, automation, and an intuitive GUI, the software component becomes an effective tool for educational and scientific experiments with optoelectronic devices.

RESULTS AND DISCUSSION

Development and testing of a desktop application

Using the computer-aided design (CAD) tool EasyEDA [14], the electrical schematic of the experimental prototype was created (**Fig. 1**). The circuit is based on two adjustable voltage regulators, LD1117-ADJ. The input voltage is supplied from the *Vin* pin of the Arduino Uno, while the output voltage is controlled by voltage dividers consisting of resistor R1 (1 kΩ) and a digitally controlled potentiometer P1 (MCP4131-502 [15]) for voltage regulator U1, and resistor R2 (1 kΩ) with a digitally controlled potentiometer P2 (MCP4131-502 [15]) for voltage regulator U2. The output voltage is determined by the following relationship:

$$V_{out} = V_{ref} \cdot \left(1 + \frac{R_{2a}}{R_{1a}}\right), \quad (1)$$

where V_{ref} = 1.25 V, R_{2a} represents the resistance of the digitally controlled potentiometer (P1 or P2 in Circuit Diagram), and R_{1a} corresponds to the resistance of the fixed resistor R1 or R2, depending on the regulator.

Voltage adjustment is performed using digitally controlled potentiometers MCP-4131-502. Their maximum resistance is 5 kΩ (±20%), and the register controlling the resistance has a 7-bit resolution, providing 128 possible resistance values. The wiper resistance, approximately 75 Ω, must also be taken into account. In practice, with 9 V applied to the *Vin* pin, output voltages of 2.15-7 V were obtained. This voltage range from both sides allows applying up to 7 V in either forward or reverse direction.

To measure high currents, the circuit includes resistor R3, which also acts as a current limiter. For low-current measurements, a digitally controlled potentiometer MCP4131-503 (50 kΩ) and a 10 kΩ resistor were added to the circuit, enabling measurement of currents on the order of microamperes for investigating the current–voltage characteristics (*I-V* characteristic) of the photodiode. Illumination measurements are performed using a GY-302 lux meter.

Based on this electrical schematic, a custom PCB was manufactured. The board's form factor replicates the dimensions of the Arduino Uno, serving as a shield (**Fig. 2a**).

To ensure accurate measurements, the system requires light shielding to eliminate any influence of external illumination. Additionally, user convenience and protection of system components from physical impact are necessary. For this purpose, a case was

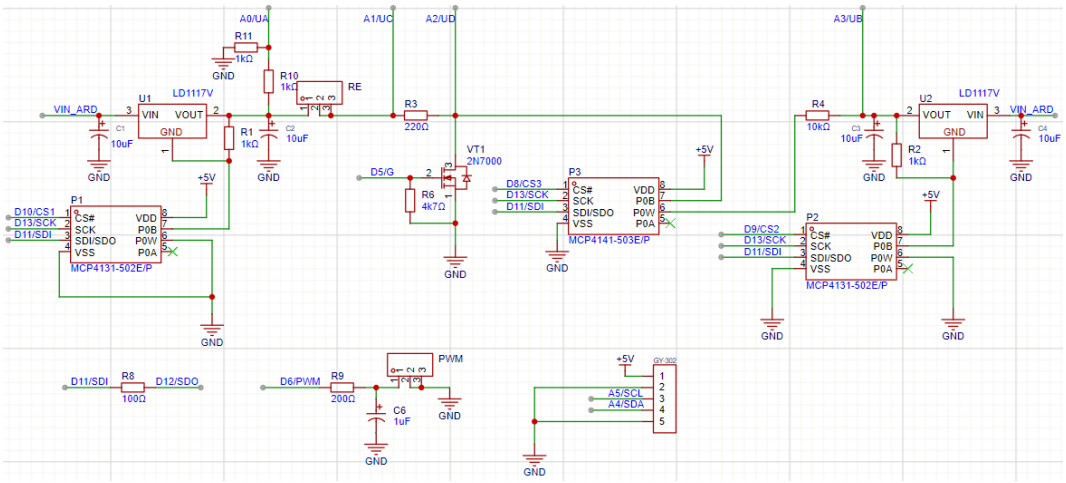
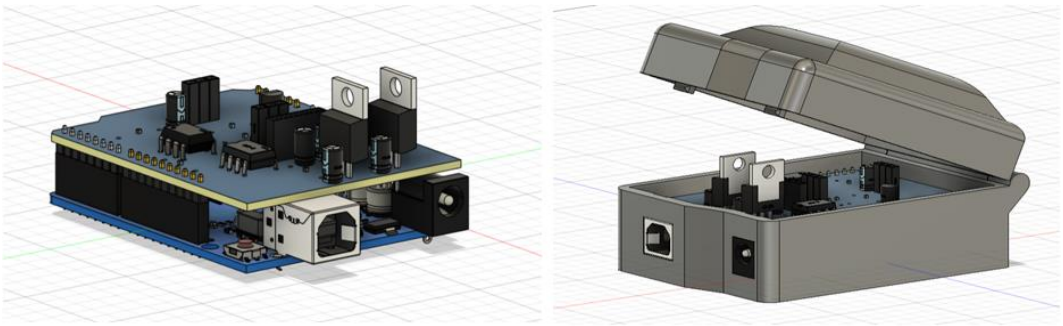


Fig. 1. Circuit Diagram of the Prototype



a) b)

Fig. 2. Experimental prototype: board integrated with Arduino Uno (a), protective enclosure (b)

designed, shown in **Fig. 2b**, optimized for 3D printing and taking into account all system requirements.

As mentioned above, the cross-platform library Tkinter was chosen for implementing the GUI. Tkinter is a standard Python library that does not require additional installation and is well-suited for developing desktop applications due to its sufficient set of widgets for creating a functional interface.

The application interface, shown in **Fig. 3**, is built on a modular principle and consists of three main sections: a control panel and two graph display areas. This structure allows users to conveniently manage the measurement process, monitor results in real time, and receive information about the system status.

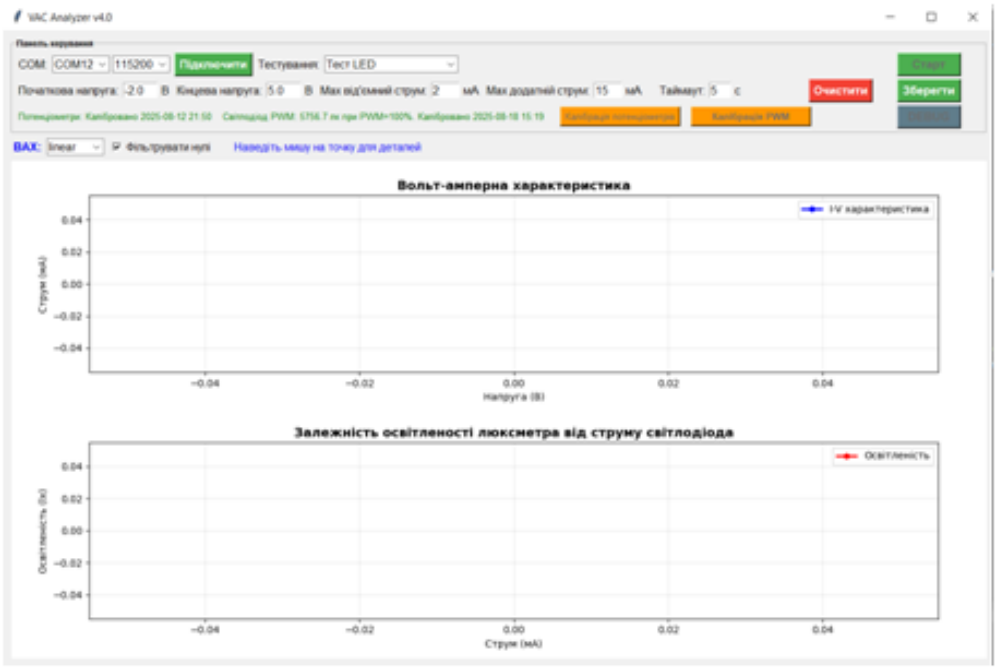
The control panel is located at the top of the application window and is organized into three horizontal rows of control elements. The first row contains controls for connecting to the microcontroller and selecting the type of testing. The application supports two operation modes: light-emitting diode (LED) testing (**Fig. 3a**) and photodetector testing (**Fig. 3b**). When the photodetector testing mode is selected, an additional field appears for setting the pulse-width modulation (PWM) value of the illuminating LED (see **Fig. 3b**). The second row of the control panel contains parameters for configuring the measurement process, while the third row allows system calibration. This row displays the status of potentiometer and LED PWM calibration.

The graph display area occupies the main part of the application window and is implemented using the Matplotlib library. Integration of Matplotlib with Tkinter is achieved via the *FigureCanvasTkAgg* module, which enables embedding graphs directly into the application window. The interface includes two vertically stacked graphs: the upper graph displays the *I-V* characteristic of the device under test, while the lower graph shows the dependence of illumination on current for LEDs. When working with photodetectors, the lower graph is automatically hidden, as it provides no relevant information for this type of device. Instead, a text field appears at the top, displaying the current illumination value in lux.

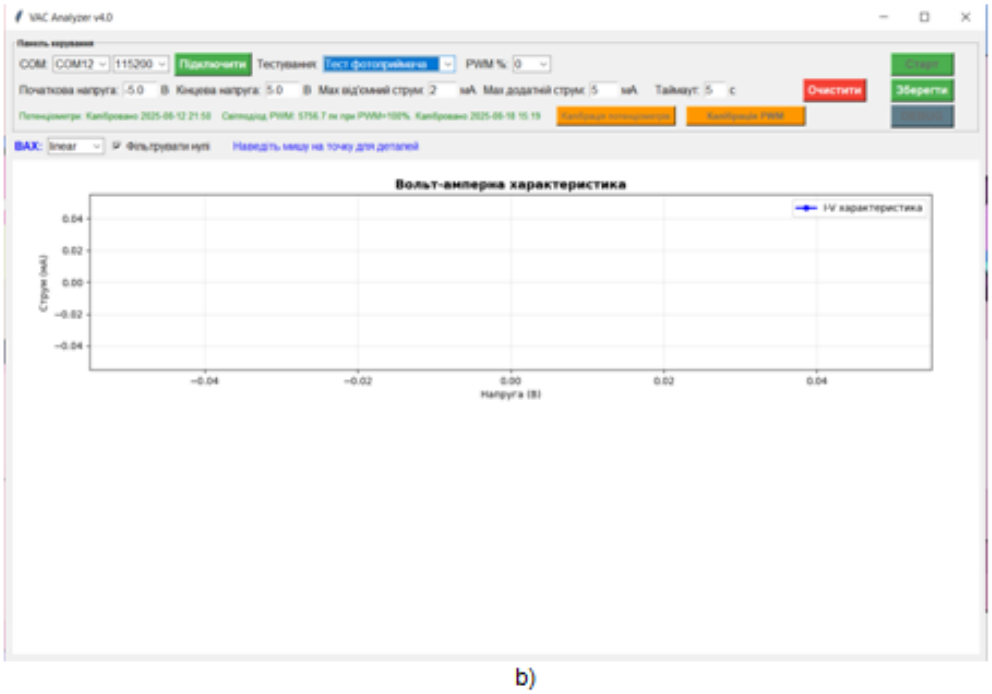
Above the graphs, an additional settings panel is located, allowing users to change the display mode. The user can switch between linear and logarithmic scales. A zero-value filtering option is also provided, which is enabled by default. This prevents points with zero current from being displayed on the graph, improving the readability of the characteristics.

A key feature of the graphs is their interactivity. When the cursor hovers over any point on the graph, a tooltip automatically appears, providing detailed information about that point.

During the interface development, special attention was paid to the logic of control element states. Many buttons and input fields are automatically enabled or disabled depending on the system state. For example, the "Start" button is only available when a connection to the microcontroller is established, and the system is properly calibrated. During measurements, input fields for test parameters are locked to prevent accidental changes.



a)



b)

Fig. 3. Application interface in different testing modes: LED testing (a), photodetector testing (b)

Of particular note is the DEBUG modal window, which can be opened via a dedicated button. This window is designed for low-level hardware control and allows manual adjustment of digital potentiometer values, control of the transistor and illuminating LED, as well as real-time monitoring of all measured voltages.

Overall, the developed graphical interface combines the functionality of professional measurement equipment with ease of use. The modular code structure and clear separation of logic and presentation make it easy to add new features and adapt the application for specific needs [7].

After the development and debugging of the basic version of the measurement system, a detailed analysis of its performance was conducted. For this purpose, clear evaluation criteria were defined, and appropriate test components were selected.

The main criteria for assessing the system's performance were: the total time required to complete a full cycle of *I-V* characteristic measurements, the number of measured points and their distribution across the voltage range, the resolution in critical regions of the characteristic, and the quality of the obtained data, including smoothness of the graph, absence of artifacts and jumps, and monotonic voltage increase.

To objectively evaluate the system, three different types of components with fundamentally different characteristics were used: a 1 kΩ resistor, a general-purpose white LED, and a BPW21R photodiode. All measurements were performed under identical parameters to ensure a valid comparison. For the resistor and the LED, the voltage range was −2 V to +5 V with a maximum allowable current of 15 mA. For the photodiode, a wider range (−5 V to +5 V) was used due to its operation under reverse bias, with a maximum current of 5 mA and illumination set to 100% (maximum brightness of the white LED illuminator). All measurements were carried out on the same hardware platform without changes in circuitry between tests.

Based on the test results, several shortcomings of the basic system version were identified. These can be grouped into three main categories: slow performance, inefficient distribution of measurement points, and insufficient data quality.

The most obvious problem was the low operating speed of the system. A detailed analysis of the timing characteristics showed that each measurement point takes approximately two seconds, including:

- 0.5 seconds – delay after sending the potentiometer position change command (DELAY_COMMAND);
- 0.5 seconds – delay before and after the measurement for system stabilization (DELAY_MEASURE);
- 1 second – time required for data transmission via the serial port and processing.

It is important to note that these delays are not excessive – they were determined experimentally as the minimum values at which the system operates stably without failures. Shorter delays led to unstable measurements due to insufficient time for voltage stabilization after changing the potentiometer positions or to missed commands due to the speed limitations of the serial interface. In contrast, the real source of slowness is the excessive number of measurements. The system performs a measurement at every potentiometer step, even when it is clearly unnecessary.

The second issue is related to the distribution of points across the voltage range. The basic version of the system uses a fixed step size for potentiometer changes, which results in an excessively dense placement of points in linear regions. This problem is especially pronounced during adjustments of pot2, which regulates resistance in the current circuit. This potentiometer has a range from 0 to 127, and the system sequentially passes through all these values with a step of 1. However, most of this range corresponds to the dead zone or the pre-bend region of the LED and the linear element, where detailed scanning is unnecessary.

The third group of problems directly concerns the quality of the obtained measurements. Analysis of the graphs revealed several systematic artifacts:

- *Points with reverse voltage movement*: This may be caused by ADC reference voltage instability, measurement noise due to coupling from potentiometers or other sources, or inaccuracies in setting the potentiometers due to their discrete nature. Such points create small "spikes" on the graph;

- *Jumps during range switching*: When the system switches the measurement range (turns on transistor VT1), a small current jump inevitably occurs. Although the algorithm attempts to select a post-switch voltage higher than the previous one, due to the discreteness of the potentiometers and ADC, it is impossible to perfectly smooth this transition. As a result, a small gap in the current is visible on the graph;
- *Insufficient noise filtering*: When working with low currents, the relative measurement noise can reach 5–10%. To filter such noise, the basic version averages 30 ADC measurements per point.

All of these identified shortcomings became the basis for defining the requirements for an optimized version of the system, which was expected to provide:

- a significantly higher measurement speed by reducing the number of redundant measurements;
- a rational distribution of points, with detailed scanning of critical regions and rapid traversal of linear sections;
- improved data quality through artifact filtering and ensuring the monotonic change of voltage.

Strategy of adaptive optimization of measurements

The analysis of the basic system's limitations clearly indicated that the primary issue is the excessive number of measurements in regions that provide little or no useful information. To address this, an adaptive algorithm was implemented, which automatically adjusts the measurement density according to the segment of the *I-V* characteristic under investigation.

The core idea of the adaptive approach is to perform a local analysis of the *I-V* characteristic in real time during the measurement process [16]. Rather than proceeding with a fixed step across the entire range, the system evaluates several of the most recently measured points and, based on this analysis, determines whether the next measurement should be taken with a small step or whether several potentiometer values can be skipped.

To implement this approach, the concept of the derivative – the rate of change of one quantity relative to another – was employed. For *I-V* characteristic, two options are possible: *dI/dV*, representing the differential conductance, and *dV/dI*, representing the differential resistance. The choice depends on the parameter being controlled at a given stage of the algorithm: when the voltage across the element is varied (*pot1*), *dI/dV* is analyzed; when the circuit resistance is varied (*pot2*), *dV/dI* is more informative.

The key concept is to classify the current region of the characteristic based on the local derivative, estimated by a first-order finite difference, i.e., the slope of the secant line between adjacent points. Small derivatives (slow changes in current or voltage) indicate "uninteresting" regions, where large steps can be taken, whereas large or rapidly changing derivatives mark critical regions requiring fine-grained measurements.

To calculate the derivative, the system stores a history of the last N measurements (typically 3–10). Local derivatives are computed for each pair of adjacent points, with mean and maximum values derived across the history. These metrics assess the rate of parameter change and identify sharp transitions indicative of key points.

Based on this analysis, the system determines the appropriate step size (number of potentiometer units to the next measurement) and whether measurements can be skipped. Importantly, the adaptive algorithm does not alter the core logic of the basic system; it optimizes the scanning trajectory while preserving all safety checks, current and voltage limits, and the overall measurement sequence.

Let us examine these two modes in more detail. The first mode of adaptive analysis (*dI/dV* mode) is applied at stages of the algorithm where the system varies the voltage across the element under study. This occurs when adjusting *pot1* (anode voltage) or *pot3*

(cathode voltage). In these cases, it is logical to analyze how the current responds to changes in voltage. The derivative of current with respect to voltage characterizes the differential conductance of the element, which, for a resistor, is inversely proportional to its resistance and remains constant across the entire characteristic. For semiconductor elements, this quantity depends strongly on the operating point and can vary by several orders of magnitude.

For LEDs, characteristic regions can be distinguished:

- *dead zone* (voltage below the threshold, approximately 0-2.3 V for a white LED): the current is nearly zero regardless of voltage, and *dI/dV* is close to zero. This region provides no useful information about the LED properties, so large steps can be taken here;
- *I-V characteristic bend* (threshold voltage region, approximately 2.4-2.8 V): the current begins to increase sharply and exponentially. *dI/dV* is very high and changes rapidly. This is the most critical region of the characteristic, requiring maximum measurement detail;
- *linear region* (after the bend, approximately 2.9-3.5 V): the current–voltage relationship approaches linearity, *dI/dV* is large but roughly constant. Moderate point density is sufficient in this region.

The derivative was calculated using the finite difference method. If two adjacent points in the history have coordinates (V_1, I_1) and (V_2, I_2), the local derivative is calculated as:

$$\frac{dI}{dV} = \frac{|I_2 - I_1|}{|V_2 - V_1|}, \tag{2}$$

where V_1 is the voltage of the previous measurement, I_1 is the current of the previous measurement, V_2 is the voltage of the current measurement, and I_2 is the current of the current measurement. Since the magnitude of the change, rather than its sign, is important, absolute differences are used.

This method was chosen for several reasons. First, it is extremely simple to implement. Since the calculations are performed in real time during measurements, more complex methods could slow down the system's operation.

Second, the finite difference method works well under our conditions. The distance between adjacent points is small enough for the local approximation to be valid. For example, when *pot1* is adjusted by one unit, the voltage changes by approximately 0.05-0.1 V, which is only 1-2 % of the full range. Within such a small range, most characteristics can be considered nearly linear.

Third, the method does not require complex assumptions about the nature of the function under study. Some alternative methods (e.g., polynomial or spline interpolation [17]) require knowledge of the function's smoothness or the availability of a certain number of derivatives. The algorithm, however, operates on real measurements, which may contain noise and discontinuities or non-smooth regions.

To improve the reliability of the estimation, the derivative is calculated for all pairs of adjacent points in the history. The mean derivative (*avg_derivative*) is then computed, characterizing the typical rate of current change in the current region, and the maximum derivative (*max_derivative*) is determined, which may indicate sharp transitions. Based on these computed derivatives, the system classifies the current region according to the logic presented in **Table 1**.

The second mode of adaptive analysis is used at stages where the system changes the resistance in the current path through the element. This occurs when adjusting *pot2*, which is connected in series with the element under study. Changing the position of the potentiometer causes a change in the current through the element, and consequently, a change in the voltage across it.

Table 1. **Description of the system's logic for classifying the currently studied regions**

Logic for zone determination	Explanation
Algorithm stage where the voltage across the element is changed (dI/dV mode)	
IF avg_derivative <0.001 A/V: Zone: "dead_zone" Step: 5 (large) Skipping allowed: yes	The threshold of 0.001 A/V was selected based on the consideration that, for typical voltage changes of 0.1 V in the LED's dead zone, the current changes by less than 0.1 mA, and *dI/dV* < 0.001. A large step allows the system to quickly traverse this "uninteresting" region.
IF max_derivative > avg_derivative × 2: Zone: "knee" Step: 1 (minimum) Skipping allowed: not	This condition detects sharp jumps in the derivative. If the maximum value is twice the average, the history contains points with very different rates of current change – a characteristic sign of a bend. The experimentally chosen factor of 2 represents a compromise between sensitivity and stability of the classification.
IF avg_derivative < 0.05 A/V: Zone: "linear" Step: 3 (middle) Skipping allowed: yes	The threshold of 0.05 A/V separates moderately increasing current from rapidly increasing current. In the linear region, the current rises predictably, so an average step can be used, and measurements can occasionally be skipped.
ELSE: Zone: "active" Step: 5 (large) Skipping allowed: not	If none of the previous conditions are met, the current increases rapidly (*dI/dV* > 0.05) but without sharp jumps. This is the operating region of the LED, which does not require very detailed scanning.
Algorithm stage where the resistance in the current circuit through the element is changed (dV/dI mode)	
IF avg_resistance > 100 Ω: Zone: "high_resistance" Step: 5 (large) Skipping allowed: yes	The threshold of 100 Ω corresponds to the LED's dead zone or the region of very low photodiode currents. At such a high resistance, changing the position of *pot2* has virtually no effect on the current through the element, so this region can be traversed quickly.
IF resistance_spread > avg_resistance × 3: Zone: "resistance_transition" Step: 1 (minimum) Skipping allowed: not	This condition detects resistance instability. If the range of values is three times greater than the average, it indicates a sharp change in the behavior of the dependence – a sign of a transition between the operating modes of the element.
IF avg_resistance > 10 Ω: Zone: "medium_resistance" Step: 3 (middle) Skipping allowed: yes	The range of 10-100 Ω corresponds to the region where the LED is partially turned on but not yet in full active mode. Moderate measurement detail is required in this region.
ELSE: Zone: "low_resistance" Step: 5 (large) Skipping allowed: not	For resistances below 10 Ω, the element operates in the active mode with high currents. This region does not require very detailed investigation.

The derivative of voltage with respect to current (dV/dI) represents the differential resistance of the element at a given point on the characteristic. For a resistor, this value is constant and equal to its nominal resistance. For semiconductor elements, it strongly depends on the operating point.

For LEDs, characteristic regions can be distinguished in terms of dynamic resistance:

- *high resistance* (thousands to tens of thousands of Ω – dead zone): current is nearly absent, and a large voltage change causes only a very small change in current;
- *transition region* (*I*–*V* curve bend): resistance changes sharply from high to low. This is the region of greatest instability in *dV/dI*;
- *moderate resistance* (tens to hundreds of Ω – linear region): resistance is relatively constant;
- *low resistance* (single Ω – saturation region): at high currents, the resistance decreases due to the conduction onset.

The calculation of dynamic resistance is performed similarly to the calculation of conductance, but with the order of quantities reversed:

$$\frac{dV}{dI} = \frac{|V_2 - V_1|}{|I_2 - I_1|}, \tag{2}$$

where V_1 is the voltage of the previous measurement, I_1 is the current of the previous measurement, V_2 is the voltage of the current measurement, and I_2 is the current of the current measurement.

An important feature is the need to check for division by zero or very small values. If the current changes by less than 1 μA (corresponding to the measurement accuracy), the derivative calculation is skipped for that pair of points.

As in the voltage mode, statistical characteristics are calculated: the average dynamic resistance (*avg_resistance*), the maximum and minimum resistance values in the history (*max_resistance* and *min_resistance*), and the spread of values (*resistance_spread* = *max_resistance* −*min_resistance*), which indicates stability. Classification in the current mode is based on the absolute resistance value and its variability (see **Table 1**).

The key issue in implementing the adaptive algorithm is determining the mode of analysis (voltage or current) to use at each measurement stage. This decision is based on the physical understanding of what is being controlled and what is being measured. If, at a certain stage of the algorithm, the position of *pot1* or *pot3* changes (voltage control) (at a certain stage of the algorithm), the *dI/dV* mode (voltage-based analysis) is used. In this case, changing *pot1* or *pot3* directly alters the potential at the anode or cathode of the studied element, meaning the voltage is controlled and the current response is measured.

If, however, the position of *pot2* changes (circuit resistance control), the *dV/dI* mode (current-based analysis) is used. Changing *pot2* alters the total circuit resistance, which causes a change in current, and the corresponding voltage change is measured.

It is important to understand that in both cases, the same *I*-*V* characteristic is being analyzed, just from different “perspectives.” The voltage mode corresponds to movement along the characteristic from left to right (*V* increases while observing *I*), while the current mode corresponds to movement from bottom to top (*I* increases due to a decrease in total resistance, while observing *V*).

Using different analysis modes for different stages provides several important advantages:

- *first*, each mode is optimally suited for its specific task. The voltage mode is ideal for detecting the bending of a LED’s *I*-*V* characteristic, where the current increases exponentially with increasing voltage, whereas the current mode is suitable for high-resistance regions, where small changes in current cause large changes in voltage;

- *second*, different stages of the algorithm have different “problematic zones”. For example, the *pot2* increase mode is particularly slow for photodiodes due to a long high-resistance region. The current mode with a 100 Ω threshold is perfectly suited for quickly passing through this region;
- *third*, a differentiated approach makes the system more versatile. Automatic mode selection allows effective operation with all optoelectronic components without changing settings.

However, the adaptive approach also has a number of limitations:

- *first*, a minimum measurement history is required for correct operation – at least 3 points. This means that the first few measurements at each stage are always performed with a small fixed step, regardless of the nature of the dependence;
- *second*, the classification of zones is based on thresholds (0.001; 0.01; 100 Ω, etc.). These values work well for typical LEDs and photodiodes but may be suboptimal for specific components. Therefore, it is necessary to conduct an investigation to determine the required step size and measurement accuracy of the system;
- *third*, the adaptive algorithm adds some complexity to the program logic and requires additional computations. Although these calculations are performed on the computer side and do not affect the Arduino’s operating speed, they still make the code more complicated to understand and configure.

The main method is skipping measurements in regions where the characteristic changes predictably. It is implemented through the *skip_allowed* parameter, which is returned by the analysis function together with *suggested_step*. When *skip_allowed* = *True*, the system can skip several consecutive measurements by increasing the potentiometer position without communicating with the Arduino. The number of skips is limited by the MAX_CONSECUTIVE_SKIPS parameter (for example, 5) to avoid missing important changes in the characteristic. Skipping is allowed only in the *dead_zone*, *linear*, *high_resistance*, and *medium_resistance* regions, where the characteristic changes smoothly. In the *knee*, *active*, *resistance_transition*, and *low_resistance* regions, skipping is prohibited because abrupt changes may occur. This strategy is especially effective for photodiodes, where the long horizontal section of the characteristic allows up to 5 consecutive measurements to be skipped, resulting in a time reduction of up to 50-60 % during reverse bias scanning.

In addition to speed optimization, several mechanisms for improving the quality of the acquired data were implemented. When adding a new point to the graph, the monotonic increase of voltage is verified. Statistics of the filtered points show that typically 2-5 % of points are discarded, which noticeably improves the smoothness of the graphs.

To objectively evaluate the effectiveness of the implemented optimizations, a series of comparative experiments was conducted in which the characteristics of the same test elements as in the baseline version of the system were measured. To ensure a fair comparison, all measurements were carried out on the same hardware without any changes to the circuitry or hardware configuration. The system was powered by a 9 V power supply, ensuring stable voltage conditions. The room temperature was maintained at 20-22 °C to minimize temperature drift in the characteristics of the semiconductor elements.

Each of the components was tested ten times in both the baseline and the optimized versions of the system. Such a sample size makes it possible to obtain statistically significant results and evaluate not only the average values but also the parameter variation. A pause of 1-2 minutes was maintained between tests to allow the system to stabilize.

For each test, the following parameters were recorded: the total test execution time from start to finish, measured with one-second precision; the number of points added to the *I–V* characteristic graph; the distribution of points across different regions of the characteristic (qualitative assessment); and the visual quality of the resulting graph (curve smoothness and presence of artifacts). From the ten measurements for each component, the mean value and standard deviation were calculated for both time and number of points. This made it possible to evaluate not only the typical performance but also the stability of the system operation.

As in the baseline version, the optimized version of the system used the same three components: a 1 kΩ resistor, a general-purpose white LED with a threshold voltage of approximately 2.5 V, and a BPW21R photodiode, as well as the same testing parameters described above.

Before testing began, optimal parameters for the *AdaptiveController* class were selected based on a preliminary analysis of the component characteristics. The tuning process involved several iterations of testing with different threshold values, followed by analysis of the point distribution and zone classification.

In particular, the final parameters for the voltage mode (*dI/dV* analysis) were selected as follows:

- the dead-zone threshold was set very low (0.001 A/V);
- the linear-zone threshold was set to 0.05 A/V. This value works well for 1 kΩ resistors, where $dI/dV \approx 1/R = 1$ A/V, which is higher than the defined threshold. Therefore, resistors are classified as being in the active region, resulting in a step size of 5 instead of a smaller step, which proved to be an optimal compromise;
- the bend detection multiplier was reduced to 2, making the system more sensitive to abrupt changes in the derivative. This is critically important for photodiodes, where the transition from reverse to forward bias may exhibit only a relatively moderate increase in the derivative;
- the step size for the dead zone was increased to 5, allowing the system to traverse “uninteresting” regions very quickly. For the linear and active regions, the step sizes were set to 3 and 5 respectively, representing a compromise between speed and detail.

In turn, the final parameters for the current mode (*dV/dI* analysis) were selected as follows:

- the high-resistance threshold was set to 100 Ω, which effectively separates the dead zone of photodiodes (where resistance may reach the kΩ range) from active regions;
- the transition detection multiplier was set to 3, which proved sufficient for detecting abrupt resistance changes without excessive sensitivity to noise;
- the step sizes for the different resistance regions were set to 5, 3, and 5 for high, moderate, and low resistance respectively, ensuring a balance between speed and safety.

The history size for the analysis was set to MIN_HISTORY = 3 and MAX_HISTORY = 5, meaning that the system analyzes the last 3-5 measurements before making a decision. A smaller value would make the system overly sensitive to noise, while a larger value would make it too inert to characteristic changes. The maximum number of consecutive skipped measurements (MAX_CONSECUTIVE_SKIPS) was set to 5. Thus, even in the most stable regions, the system is forced to perform a контрольне measurement after every five skipped points. The reverse-voltage movement filtering tolerance (VOLTAGE_TOLERANCE) was set to 5 mV, corresponding to approximately one ADC quantization step and effectively filtering noise without losing useful data.

These parameters demonstrated the best balance between speed, quality, and universality during testing across all three types of components.

A *1 kΩ resistor* is an ideal component for evaluating the efficiency of the system when working with linear elements. **Fig. 4** shows its I–V characteristic in the baseline (**Fig. 4a**) and optimized (**Fig. 4b**) versions of the system.

The analysis of the point distribution revealed non-uniformity: in the region from -5 V to +0.5 V, a clustering of points was observed, whereas in the range from +3 V to +5 V, the number of points was slightly lower (**Fig. 4a**). This non-uniformity is related to the operating characteristics of digital potentiometers and the nonlinear dependence of voltage on their position.

Despite the significant reduction in detail in the optimized version, the quality of the characteristic remained high – the straight-line behavior is reproduced with sufficient accuracy for practical applications (see **Fig. 4b**). The slight increase in time per point (1.53 sec in the baseline version versus 1.76 sec in the optimized version) is explained by the additional time required for derivative calculations and classification. However, this increase in algorithmic complexity is fully compensated by the reduction in the number of measurement points.

The analysis of the adaptive controller operation showed that the resistor is consistently classified as an “active region” with $dI/dV \approx 1.0$ A/V. When changing *pot2*, the system switches to current-analysis mode and classifies the characteristic as a “low-resistance” region.

The point distribution (**Fig. 4b**) did not become more uniform – the system displays approximately the same spread of points as in the baseline version, with clustering in the range from -0.5 V to +0.5 V. Visually, the graph of the optimized version does not differ from the baseline one – both versions produce nearly ideal straight lines.

The *white LED* represents the most interesting case for the adaptive algorithm due to the clearly distinguishable regions of its characteristic: the dead zone, the knee region, and the active region. **Fig. 5** presents its *I–V* characteristic in the baseline (**Fig. 5a**) and optimized (**Fig. 5b**) versions of the system.

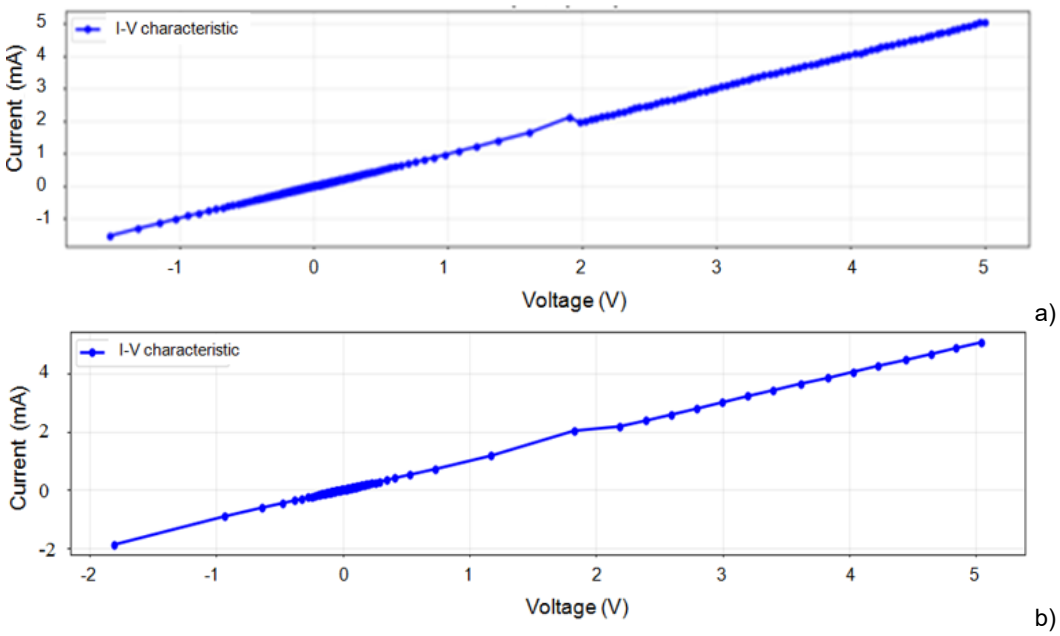


Fig. 4. Current–voltage characteristic of a 1 kΩ resistor in the baseline (a) and optimized (b) versions of the system

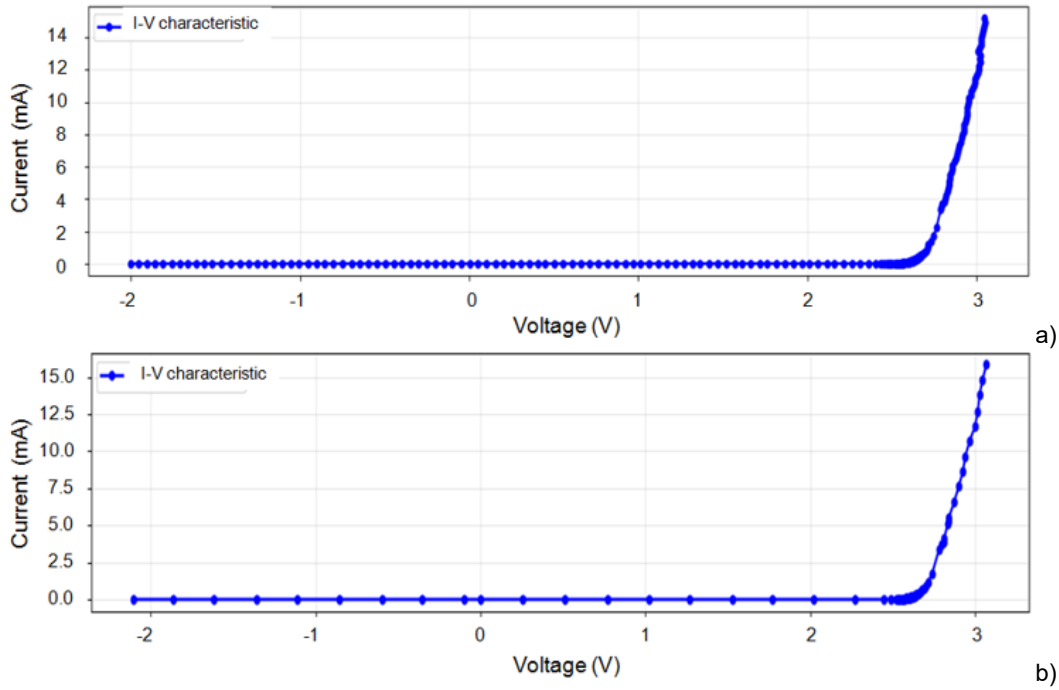


Fig. 5. Current–voltage characteristic of a white LED in the baseline (a) and optimized (b) versions of the system

The point distribution in the baseline version of the system was approximately as follows: 90 points in the dead zone (0-2.3 V), 200 points in the knee region of the *I–V* characteristic (2.3-2.7 V), and 65 points in the active region (2.7-3.1 V). The main issue was the excessive number of points in the dead zone, where the current is practically zero and the characteristic contains no useful information, as well as the excessive number of points in the knee region of the *I–V* curve (**Fig. 5a**).

The point distribution in the optimized version of the system, shown in **Fig. 5b**, was approximately 20 points in the dead zone, 70 points in the knee region, and 15 points in the active region. Critically, the number of points in the knee region remained sufficient for detailed reproduction of the exponential dependence of current on voltage.

Analysis of the obtained results revealed an almost threefold acceleration of the test execution while reducing the number of points to 29.5 % of the baseline version. The key achievement is the preservation of high characteristic quality in critical regions.

The analysis of region classification demonstrated the correct operation of the algorithm:

- in the dead zone (V < 2.3 V), the system stably detects dI/dV < 0.001 A/V and uses a step size of 5 with permission to skip points;
- in the knee region of the *I–V* characteristic (V ≈ 2.3-2.8 V), the system detects a sharp increase in the derivative (max_derivative > avg_derivative × 2) and switches to a no-skipping mode. This ensures detailed scanning of the most important region of the characteristic;
- in the active region (V > 2.9 V), the current increases rapidly but more or less linearly, and the system switches to the high-current measurement mode.

Another positive effect is the reduction in the number of filtered points associated with reverse voltage movement. In the baseline version, an average of 12-15 points (3-5 % of the total number) were filtered out, whereas in the optimized version only 1-3

points (1-3 %) were removed. This indicates that the adaptive algorithm is less prone to creating situations involving accidental voltage decreases.

The *BPW21R photodiode* is the most problematic component for measurement due to the very long horizontal region of its characteristic under reverse bias. **Fig. 6** presents its *I–V* characteristic in the baseline (**Fig. 6a**) and optimized (**Fig. 6b**) versions of the system.

The main problem of the baseline version is the enormous number of redundant points on the horizontal region, where the current is practically constant at approximately 0.9 mA (see **Fig. 6a**). These points provide no additional information.

The observed reduction in both absolute measurement time and point dispersion indicates more predictable operation of the optimized system. Analysis of the results revealed more than a twofold acceleration of the test execution while reducing the number of points to 42.0 % of the baseline version (**Fig. 6b**).

The analysis of the algorithm operation demonstrated effective region classification:

- on the horizontal region under reverse bias, the system detects a very small derivative *dI/dV* (practically zero) and a high dynamic resistance (> 100 Ω) in current mode, and therefore enables point skipping;
- in the transition region (+0.1...+0.5 V), the system correctly detects an increase in the derivative and reduces the step size to 1-3, ensuring detailed scanning.

An important result is that even with the number of points reduced by more than half, the quality of the characteristic did not deteriorate (see **Fig. 6b**). Visual comparison of the graphs shows practically identical curves with the same level of detail in the critical regions.

A comparison of the testing results for the electronic components investigated is summarized in **Table 2**.

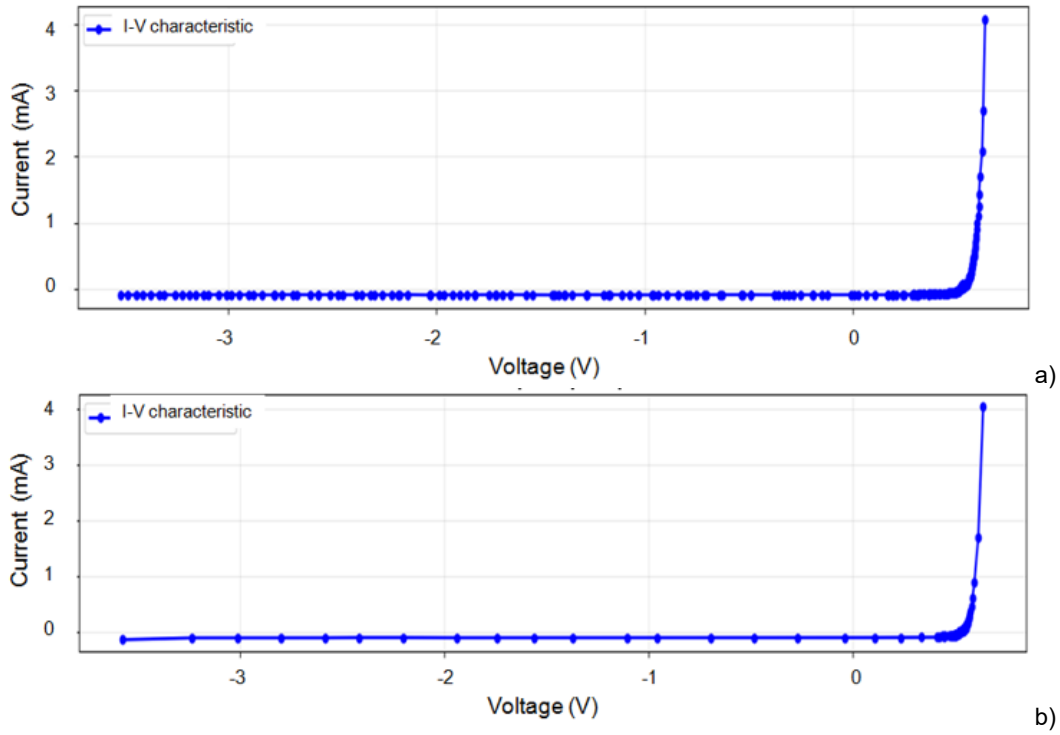


Fig. 6. Current–voltage characteristic of the BPW21R photodiode in the baseline (a) and optimized (b) versions of the system

Table 2. **Results of testing electronic components in the baseline and optimized versions of the system**

Component	The baseline version				The optimized version				Speedup, times	Point savings, %
	Test duration, sec	Point number	Duration standard deviation	Number point standard deviation	Test duration, sec	Point number	Duration standard deviation	Number point standard deviation		
1 kΩ resistor	735	480	4.3	1.8	172	98	3.1	2.1	4.27	79.6
White LED	458	356	4.8	1.9	155	105	3.2	2.3	2.95	70.5
BPW21R photodiode	675	355	4.9	2	285	149	3.3	2.8	2.37	58

The developed adaptive optimization system demonstrates a stable speedup of 2.4-4.3 times depending on the type of component, while maintaining high-quality characteristics. The greatest performance gain is observed for components with predictable behavior.

Importantly, the optimization does not degrade data quality – critical regions are scanned with high resolution, while redundant measurements in predictable regions are eliminated. Visual comparison of the graphs shows that the optimized version does not worsen the *I–V* characteristics compared to the baseline version, which is explained by better filtering and the absence of certain artifacts present in the baseline version. The reverse-voltage movement filter effectively removes “jagged edges” from the graph.

For the LED, the *I–V* curve knee in the optimized version is reproduced somewhat better than in the baseline due to the concentration of points in this region. The exponential increase in current from 0.1 mA to 15 mA with a voltage change from 2.5 V to 3.1 V is clearly visible in both graphs, but the optimized version has a higher density of points specifically in the initial part of the knee (2.3-2.7 V), where the rate of change is maximal. For the photodiode, the transition region from reverse to forward bias (+0.1 V to +0.5 V) in the optimized version has sufficient detail for correctly reproducing the nature of the dependence.

Overall, the qualitative assessment shows that the optimized version is not only faster but also provides even better measurement quality compared to the baseline version. This confirms the success of the chosen optimization approach.

CONCLUSION

A fully functional system for automated measurement of current-voltage and optical characteristics of optoelectronic devices has been developed and implemented based on the Arduino UNO microcontroller with adaptive optimization of the scanning process. The system combines hardware built from readily available electronic components with software for both the microcontroller and a personal computer.

The baseline version of the system with a fixed scanning step demonstrated reliable operation for all types of investigated components; however, performance analysis revealed significant drawbacks that led to excessively long test durations. To address the issue of inefficient time usage, an adaptive optimization system based on numerical analysis methods was developed. The system was integrated into the existing code with minimal

changes to the core logic, preserving all safety and calibration functions. Additionally, a reverse-voltage movement filter was implemented, which removes artifact points caused by measurement noise, improving the visual quality of the obtained characteristics.

An experimental study of the optimized system's efficiency was carried out on three types of electronic components, with ten independent measurements performed for each type. The developed adaptive optimization system demonstrates a stable speedup of 2.4-4.3 times depending on the component type while maintaining high-quality characteristics. Importantly, during all thirty experiments, no cases were recorded in which critical regions were missed or the characteristic shape was significantly distorted. This confirms the reliability of the developed region classification system and the correctness of the chosen adaptive algorithm parameters. It should also be noted that the measurement time per point increases by 10-15 % due to additional derivative and classification computations; however, this is fully compensated by the reduction in the total number of points. The region classification parameters were selected empirically, so specific components may require adjustment of threshold values.

Overall, the presented work demonstrates the possibility of creating an efficient and accessible system for studying the characteristics of optoelectronic devices based on modern microcontroller platforms. The developed adaptive optimization system has proven effective under real operating conditions, providing a significant increase in measurement speed while maintaining high result quality.

Comparison with simpler adaptive scanning strategies was not conducted, and this remains a subject for future research.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [M.R., Yu.K]; methodology, [M.R., Yu.K., B.K.]; investigation, [M.R., B.K.]; writing – original draft preparation, [M.R., B.K.]; writing – review and editing, [M.R., Yu.K., B.K.]; visualization, [M.R., Yu.K., B.K.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Piprek J. (2018). Handbook of Optoelectronic Device Modeling and Simulation. Vol. II. Lasers, Modulators, Photodetectors, Solar Cells, and Numerical Methods. *CRC Press.* 233 p. http://lab.semi.ac.cn/library/upload/files/2020/8/59160607.pdf

[2] Barua P. (2023). Optoelectronics: Basics and Applications. *Kindle Edition.*104 p. https://www.amazon.com/Optoelectronics-Basics-Applications-Prasun-Barua/dp/B0CD15SBN7

[3] Monk S. (2017). Hacking Electronics: Learning Electronics with Arduino and Raspberry Pi. *McGraw Hill TAB*. 276 p. https://rpitst.com/img/ebook/1710996146_7958c4023e1d82a99bd4.pdf

[4] Arsath Natheem S. (2021). Handbook of Arduino: 100+ Arduino Projects learn by doing practical guides for beginners and inventors. *Kindle Edition*. https://www.scribd.com/document/721172292/Handbook-of-Arduino-100-Arduino-Projects

[5] Burden R. L., Faires J. D. (2010). Numerical Analysis (9th ed.) *Brooks/Cole*, https://faculty.ksu.edu.sa/sites/default/files/numerical_analysis_9th.pdf
[6] Cejer M. A., Tucker J. L., Weng L. (2013). Choosing the Optimal Source Measurement Unit Instrument for Your Test and Measurement Application. *Keithley*. p. 1-18. https://download.tek.com/document/HowToCHooseSMU.pdf
[7] Wilson G, Aruliah DA, Brown CT, Chue Hong NP, Davis M, Guy RT, et al. (2014) Best Practices for Scientific Computing. *PLoS Biol* 12(1): e1001745. https://doi.org/10.1371/journal.pbio.1001745
[8] Python Software Foundation. (2026). *tkinter — Python interface to Tcl/Tk*. Python 3 documentation. https://docs.python.org/3/library/tkinter.html
[9] PySerial Documentation. (2020). *PySerial*. https://www.pyserial.com/docs
[10] Hunter J., Dale D., Firing E., Droettboom M., the *Matplotlib* Development Team (2017) Matplotlib. https://matplotlib.org/2.1.0/Matplotlib.pdf
[11] Matplotlib 3.10.8 documentation. (2025). *Mtplotlib*. https://matplotlib.org/stable/
[12] Tosi S. (2009). Matplotlib for Python Developers. *Packt Publishing*, https://theswissbay.ch/pdf/Gentoomen%20Library/Programming/Python/Matplotlib%20for%20Python%20Developers%20%282009%29.pdf
[13] Introduction to Threading in Python. (2025). *ManageEngine Site24x7*. https://www.site24x7.com/learn/threading-in-python.html
[14] EasyEDA Tutorial. V.6.4.32. (2022). https://image.easyeda.com/files/EasyEDA-Tutorial_v6.4.32.pdf
[15] MCP413X/415X/423X/425X: Single/Dual Digital Potentiometer with SPI Interface. Datasheet. (2008). *Microchip Technology Inc.* https://www.lcsc.com/datasheet/C640354.pdf
[16] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P. (2007). Numerical Recipes: The Art of Scientific Computing (3rd ed). *Cambridge University Press.* 1235 p. https://assets.cambridge.org/97805218/80688/frontmatter/9780521880688_frontmatter.pdf
[17] Karpfinger, C. (2022). Polynomial and Spline Interpolation. In: Calculus and Linear Algebra in Recipes. *Springer, Berlin, Heidelberg*. p. 311–320. https://doi.org/10.1007/978-3-662-65458-3_29

ОПТИМІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДОСЛІДЖЕННЯ ДИСКРЕТНИХ ОПТОЕЛЕКТРОННИХ ПРИСТРОЇВ З ВИКОРИСТАННЯМ МІКРОКОНТРОЛЕРА ARDUINO

Михайло Русиняк, Юрій Корчак*, Богдан Кужель
Кафедра оптоелектроніки та інформаційних технологій
Львівський національний університет імені Івана Франка
вул. Ген. Тарнавського 107, м. Львів, 79017, УКРАЇНА

АНОТАЦІЯ

Вступ. Якісне дослідження електричних і світлових характеристик оптоелектронних компонентів є необхідною складовою під час проєктування,

розробки та тестування оптоелектронних систем. З цією метою розроблено систему автоматизованого вимірювання характеристик оптоелектронних пристроїв на базі мікроконтролера Arduino UNO з програмованим керуванням напругою через цифрові потенціометри та передачею даних на комп'ютер для візуалізації, а також проведено її оптимізацію.

Матеріали та методи. Повнофункціональна система автоматизованого вимірювання вольт-амперних та світлових характеристик оптоелектронних пристроїв розроблена та реалізована на базі мікроконтролера Arduino UNO. Апаратна частина системи побудована на базі цифрових потенціометрів MCP4131 з використанням інтегрованого люксметра GY-302, настільний додаток створено на мові програмування Python з використанням бібліотек Tkinter, pySerial, Matplotlib, Threading. Адаптивна оптимізація системи проведена на основі методу скінченних границь для обчислення відповідних локальних похідних.

Результати. Експериментальне дослідження ефективності оптимізованої системи показало суттєве підвищення продуктивності за збереження високої якості характеристик. Розроблена система адаптивної оптимізації демонструє стабільне пришвидшення в 2,4-4,3 рази залежно від типу елемента. Жодного разу під час проведених експериментів не було зафіксовано випадків пропускання критичних зон або суттєвого спотворення форми характеристики. Це підтверджує надійність розробленої системи класифікації зон та коректність вибору параметрів адаптивного алгоритму.

Висновки. Результати роботи демонструють можливість створення ефективної та доступної системи для дослідження характеристик оптоелектронних пристроїв на базі сучасних мікроконтролерних платформ. Розроблена система адаптивної оптимізації показала свою ефективність у реальних умовах експлуатації.. Це робить систему цінним інструментом як для навчальних, так і для практичних застосувань у галузі дослідження та розробки оптоелектронних пристроїв.

Ключові слова: вольт-амперна характеристика, метод скінченних різниць, мікроконтролер Arduino, світлодіод, фотодіод, цифровий потенціометр.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
30 June, 2026	15 September, 2026	21 September, 2026	28 September, 2026