A JavaFX-Based Interface for Monitoring a Laser Security System Integrated with Arduino

**Popescu Maria Cristina**

Faculty of Automation, Computers and Electronics, Craiova, Romania,*,*

*E-mail Popescu.maria.z3n@student.ucv.ro*

**Abstract**

This paper presents the design and implementation of a monitoring interface for a laser-based alarm system using JavaFX and Arduino. The interface allows users to arm or disarm the alarm, monitor its current status, and receive real-time notifications when the alarm is triggered. The physical system includes a laser, photoresistor, buzzer, and an Arduino microcontroller. Communication between the application and Arduino is achieved through serial communication. This integration provides a practical and responsive approach to security monitoring, enabling an interactive, user-friendly interface alongside reliable hardware functionality.

**Keywords**

JavaFX; Arduino; alarm system; laser sensor; photoresistor; buzzer; real-time monitoring; serial communication; user interface; IoT

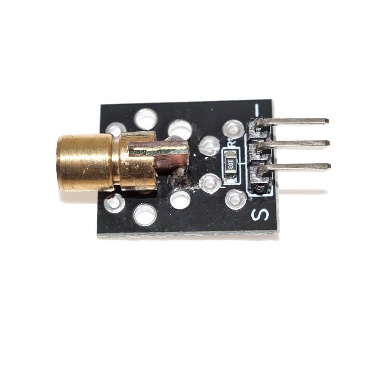
1. **Introduction**

Security systems play a vital role in protecting property and assets. With the rise of IoT and embedded systems, integrating hardware and software platforms provides both flexibility and cost-effectiveness. This project presents a laser-based security system monitored and controlled via a custom-built JavaFX interface.

**2. System Architecture**

**2.1 Hardware Components**

• **Laser Diode** – emits a constant beam toward a photoresistor



**Figure 1. Laser**

* **Photoresistor (LDR**) – detects interruptions in the laser beam



**Figure 2. Photoresistor**

* **Buzzer** – emits an audible alert when the system is triggered



**Figure 3. Buzzer**

* **Arduino UNO** – processes input from the sensor and communicates with the JavaFX application



**Figure 4. Arduino UNO**

**2.2 Software Components**

The software consists of a JavaFX application with the following capabilities:

• Display alarm status (armed/disarmed)

• Show timestamp of latest activation

• Notify user when the alarm is triggered

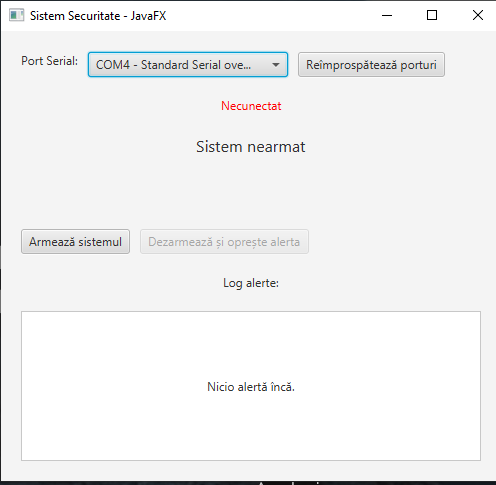
• Send commands to Arduino (e.g., arm/disarm)

**3. Implementation**

**3.1 Communication Protocol**

Serial communication (via USB) was used for data exchange. The application sends characters (e.g., 'A' for arm, 'D' for disarm) and listens for specific responses when the alarm is triggered.

**3.2 Interface Overview**



**Figure 4 JavaFX User Interface for Alarm Monitoring**

**3.3 JavaFX Application Code Overview**

The monitoring application is implemented using JavaFX and communicates with an Arduino via serial communication. The core functionalities include arming/disarming the system, handling alerts, and managing the interface.

Below is a highlight of the structure of the code:

public class AlarmSystemApp2 extends Application {

private boolean isArmed = false;

private boolean alertActive = false;

private Circle ledIndicator;

private Label statusLabel;

private Timeline blinkTimeline;

private SerialPort serialPort;

private ListView<String> alertLog;

private final String correctPassword = "1234";

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage primaryStage) {

Button armButton = new Button("Armează sistemul");

Button disarmButton = new Button("Dezarmează și oprește alerta");

disarmButton.setDisable(true);

statusLabel = new Label("Sistem nearmat");

statusLabel.setStyle("-fx-font-size: 16px;");

ledIndicator = new Circle(15, Color.DARKRED);

ledIndicator.setVisible(false);

**4. Results of testing**

The system was tested under various conditions. When the laser beam was interrupted (e.g., by a hand or object), the Arduino sent a signal to the JavaFX app, triggering an alert and updating the interface in under one second. The system operated reliably in multiple test runs.

**5. Conclusions**

This project demonstrates a functional security system using low-cost components and modern software practices. The integration of JavaFX with Arduino provides a responsive and intuitive monitoring platform. Future improvements may include adding remote monitoring features or integrating SMS/email alerts.

**Acknowledgment**

We would like to thank our professors and colleagues who provided feedback and support throughout the development of this project.

**References:**

1. Monk, S., Programming Arduino: Getting Started with Sketches, McGraw-Hill Education, New York, 2016.

2. Oracle. JavaFX Documentation. Available online: https://openjfx.io/ [accessed on 5 June 2025]

3. Margolis, M., Arduino Cookbook, 3rd ed., O'Reilly Media, Sebastopol, 2020