

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет
«Харківський авіаційний інститут»

Кафедра мехатроніки та електротехніки

Лабораторна робота 3

з дисципліни **«Мехатронні системи»**
(назва дисципліни)

На тему: Розробка системи керування зумом оптичної мехатронної системи на базі Arduino Leonardo та графічного інтерфейсу користувача для керування і візуального контролю по відеопотоку

Здобувача освіти 359 групи

Ліпницька Д.В.

(прізвище та ініціали студента)

Прийняв: доцент, к.т.н., доцент

Кочук С.Б.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

м. Харків – 2025

Мета роботи

1. Налаштuvати керування платою-драйвером системи лінз за допомогою Arduino Leonardo через USB (Serial).
2. Реалізувати керування «наближення/віддалення/покій» відповідно до заданої логіки рівнів сигналів на D4/D5 та таймера 500 мс.
3. Розробити GUI (Python + PySide6) для захоплення відео з USB-камери та надсилання команд керування зумом.
4. Перевірити працездатність системи за зміною зображення з камери під час керування зумом.

Завдання роботи

1. Ознайомитись із механікою типової оптичної zoom системи [1].
2. Виконати аналіз сучасних типів крокових двигунів та принципів їх керування, охарактеризувати їх відмінності [2, 3].
3. Надати принципову електричну схему керування біполярним кроковим двигуном із використанням транзисторної логіки та описати принцип її роботи [4].
4. Ознайомитися з платою-драйвером системи лінз та її інтерфейсами.
5. Ознайомитися з цифровою камерою Hawkeye 4 [5], її характеристиками, форматом відеосигналу та способами підключення.
6. Розробити програмне забезпечення для керування платою-драйвером системи керування лінз з використанням плати розробника Arduino Leonardo. Необхідно отримувати по USB команди керування підвісом та відправляти. Керування має відбуватись за наступною логікою: по USB з ПК буде приходити зовнішня команда про наближення чи віддалення.

Відповідно до цього на двох цифрових виходах плати Arduino мають змінюватись значення рівня сигналу:

покій: *HIGH, HIGH*

приближення: *LOW, HIGH*

віддалення: *HIGH, LOW*

Логіка наближення та віддалення мають бути встановлені на 500 мсек, після чого автоматично переходить в режим покоя. Для керування використовувати цифрові виходи D4 та D5 (Arduino Leonardo).

7. Розробити графічний інтерфейс користувача (GUI), який:

- здійснює захоплення відеосигналу з USB-камери;
- відображає вимірюну відстань до об'єкта;
- містить елементи керування платою-драйвером системи лінз.

8. Описати підключення електроніки до плати розробника Arduino Leonardo на стенді у звіті.

Теоретичні відомості

Оптична zoom-система — це механізм зміни положення лінз/груп лінз відносно сенсора, що змінює фокусну відстань та масштаб зображення. У мехатронних системах цей механізм зазвичай приводиться у рух електроприводом із редуктором/гвинтовою парою та датчиками положення (кінцевики, енкодери або оптичні мітки). [1]

Кроковий двигун — електромеханічний перетворювач, що переміщує вал дискретними кроками. Переваги для оптики: висока повторюваність позиціювання, просте керування імпульсами, можливість утримання позиції. Основні типи: уніполярні та біполярні (біполярні мають кращий момент при тих самих габаритах, але потребують відповідного драйвера). [2], [3]

Керування кроковим двигуном зазвичай реалізують через драйвер (Н-мости, мікросхемні драйвери), який приймає логічні сигнали (STEP/DIR або фазні сигнали) та комутує струм у обмотках. Для біполярних двигунів типова силова частина — Н-міст на транзисторах, що забезпечує зміну напрямку струму в обмотках. [2], [3]

Інтерфейс USB Serial (UART over USB) — стандартний спосіб обміну «ПК ↔ Arduino»; ПК надсилає текстові команди, Arduino інтерпретує їх і формує керуючі рівні на виходах D4/D5. Таймер 500 мс потрібен для «імпульсного» керування: команда активна короткий час, після чого система повертається у безпечний стан (покій).

GUI керування поєднує: (1) відеозахоплення з USB-камери (OpenCV), (2) відображення кадру в Qt-віджеті, (3) керуючі кнопки, що надсилають команди у Serial-порт.

Опис підключення електроніки

Підключення виконано так, щоб Arduino Leonardo формувала два керуючі сигнали для плати-драйвера системи лінз:

Сигнал	Arduino Leonardo	Вхід плати драйвера/стенда	Призначення
PIN_A	D4	IN_A (керуючий вхід 1)	Формування режимів покій/zoom
PIN_B	D5	IN_B (керуючий вхід 2)	Формування напрямку (in/out)
GND	GND	GND	Спільна «земля» для коректних рівнів
USB	USB	ПК	Живлення + Serial-зв'язок

Логіка керування рівнями (за умовою):

- **покій:** HIGH, HIGH
- **приближення (zoom in):** LOW, HIGH (500 мс → потім покій)
- **віддалення (zoom out):** HIGH, LOW (500 мс → потім покій)

ХІД РОБОТИ

1. Підготовка середовища Arduino

- Запущено Arduino IDE.
- Перевірено підключення Arduino Leonardo до ПК та до плати керування системою лінз.
- Встановлено швидкість Serial: **115200**.
- Завантажено скетч керування D4/D5 відповідно до логіки «zoom_in / zoom_out / idle».

2. Реалізація прошивки Arduino (керування D4/D5)

- Оголошено виходи **PIN_A = 4, PIN_B = 5**.
- Реалізовано стани: IDLE, ZOOM_IN, ZOOM_OUT.
- Після прийому команди з USB запускається відповідний режим на **500 мс**, далі автоматичне повернення в IDLE.

3. Підготовка Python-проекту GUI

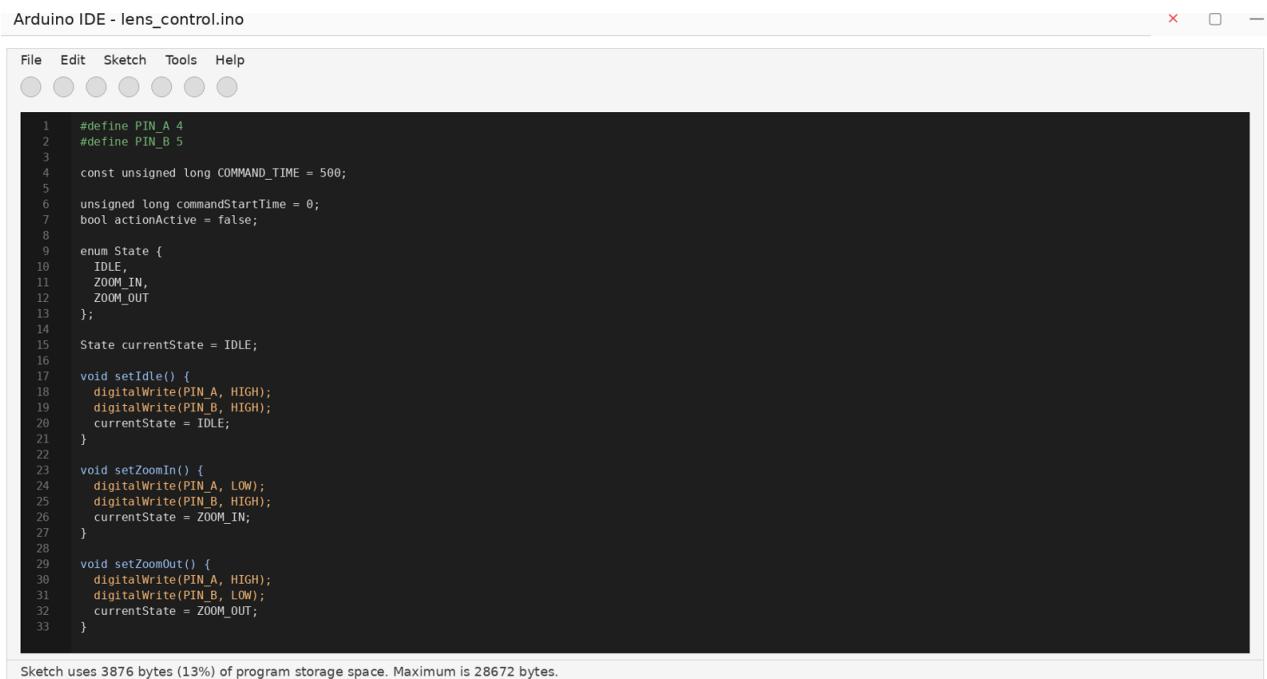
- Запущено Sublime Text.
- Створено структуру проекту:
 - project/
 - main.py
 - video_thread.py
 - serial_manager.py
- Реалізовано модуль роботи з Serial (serial_manager.py), модуль потоку відеозахоплення (video_thread.py), та головне вікно GUI (main.py).

4. Запуск та перевірка працездатності

- Запущено main.py від імені адміністратора.

- Перевірено, що з USB-камери отримується відеопотік і кадр оновлюється в UI.
- Натискання **ZOOM IN** надсилає рядок `zoom_in\n` у Serial; Arduino активує LOW/HIGH на D4/D5 на 500 мс і повертається в HIGH/HIGH.
- Натискання **ZOOM OUT** надсилає `zoom_out\n`; Arduino активує HIGH/LOW на 500 мс і повертається в HIGH/HIGH.
- Під час команд зуму візуально фіксується зміна зображення з камери (ознака зміни положення лінз).

РЕЗУЛЬТАТИ



```

1 #define PIN_A 4
2 #define PIN_B 5
3
4 const unsigned long COMMAND_TIME = 500;
5
6 unsigned long commandStartTime = 0;
7 bool actionActive = false;
8
9 enum State {
10     IDLE,
11     ZOOM_IN,
12     ZOOM_OUT
13 };
14
15 State currentState = IDLE;
16
17 void setIdle() {
18     digitalWrite(PIN_A, HIGH);
19     digitalWrite(PIN_B, HIGH);
20     currentState = IDLE;
21 }
22
23 void setZoomIn() {
24     digitalWrite(PIN_A, LOW);
25     digitalWrite(PIN_B, HIGH);
26     currentState = ZOOM_IN;
27 }
28
29 void setZoomOut() {
30     digitalWrite(PIN_A, HIGH);
31     digitalWrite(PIN_B, LOW);
32     currentState = ZOOM_OUT;
33 }

```

Sketch uses 3876 bytes (13%) of program storage space. Maximum is 28672 bytes.

Рисунок 1 – Arduino IDE із завантаженим скетчем керування D4/D5 та логікою `zoom_in/zoom_out`.

Arduino IDE - Serial Monitor

```
COM5 115200 baud
zoom_in
OK: ZOOM_IN active (500 ms)
OK: IDLE
zoom_out
OK: ZOOM_OUT active (500 ms)
OK: IDLE
```

Рисунок 2 – Serial Monitor: тест надсилання команд та перевірка реакції (активація режиму на 500 мс і перехід у IDLE).

Sublime Text - project

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- project
- main.py
- serial_manager.py
- video_thread.py

main.py

```
1 import sys
2 import cv2
3 from PySide6.QtWidgets import (
4     QApplication, QWidget, QLabel, QPushButton,
5     QVBoxLayout, QHBoxLayout
6 )
7 from PySide6.QtGui import QImage, QPixmap
8 from video_thread import VideoThread
9 from serial_manager import SerialManager
10
11 class MainWindow(QWidget):
12     def __init__(self):
13         super().__init__()
14         self.setWindowTitle("Optical Mechatronic System")
15         self.video_label = QLabel("Camera")
16         self.video_label.setFixedSize(640, 480)
17         self.btn_zoom_in = QPushButton("ZOOM IN")
18         self.btn_zoom_out = QPushButton("ZOOM OUT")
19         self.btn_zoom_in.clicked.connect(self.zoom_in)
20         self.btn_zoom_out.clicked.connect(self.zoom_out)
21         layout = QVBoxLayout()
22         layout.addWidget(self.video_label)
23         buttons = QHBoxLayout()
24         buttons.addWidget(self.btn_zoom_in)
25         buttons.addWidget(self.btn_zoom_out)
26         layout.addLayout(buttons)
27         self.setLayout(layout)
```

Рисунок 3 – Структура Python-проєкту та фрагмент коду GUI (архітектура main.py, video_thread.py, serial_manager.py).

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\student\Desktop\project> python -m pip install pyside6 opencv-python pyserial
Requirement already satisfied: pyside6 in c:\python\lib\site-packages
Requirement already satisfied: opencv-python in c:\python\lib\site-packages
Requirement already satisfied: pyserial in c:\python\lib\site-packages

PS C:\Users\student\Desktop\project> python main.py
[INFO] Camera: index=0 opened
[INFO] Serial: COM5 @ 115200
[INFO] UI started

PS C:\Users\student\Desktop\project>_
```

Рисунок 4 – Запуск застосунку з консолі та перевірка залежностей/старту



Рисунок 5 – робота GUI: відображення відео та кнопки керування ZOOM IN/OUT.

Таблиця перевірки логіки

Це невелика таблиця котра показує, що було перевірено умову ($D4/D5 + 500$ мс):

Команда з ПК	D4 (PIN_A)	D5 (PIN_B)	Тривалість	Після 500 мс
(нема) IDLE	HIGH	HIGH	постійно	HIGH/HIGH
zoom_in	LOW	HIGH	500 мс	HIGH/HIGH
zoom_out	HIGH	LOW	500 мс	HIGH/HIGH

ВИСНОВОК

Під час виконання лабораторної роботи було розроблено та перевірено систему керування зумом оптичної мехатронної системи на базі Arduino Leonardo та програмного GUI на Python. Реалізовано прийом команд керування по USB (Serial) та формування керуючих рівнів на виходах D4 і D5 відповідно до заданої логіки: **покій (HIGH/HIGH), наближення (LOW/HIGH), віддалення (HIGH/LOW)** з автоматичним поверненням у режим покою через **500 мс.**

Також створено графічний інтерфейс, який здійснює захоплення відеосигналу з USB-камери, відображає зображення у вікні програми та забезпечує керування зумом за допомогою кнопок **ZOOM IN** і **ZOOM OUT**. Працездатність системи підтверджено коректним виконанням команд та візуальною зміною зображення з камери під час керування положенням лінз.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kingslake R. *Lens Design Fundamentals* (2010).
2. FAULHABER. *Stepper Motors / Technical Information* (AN001_EN).
3. OME Technology. *Stepper Motor Handbook* (ome_stepper-handbook_en.pdf).
4. Відеоматеріал з принципової схеми/керування (YouTube): apO-n8kkWQo.
5. Hawkeye 4 (опис/характеристики камери): rcdone.top — сторінка продукту.
6. Arduino.ua — датчик відстані VL53L0X (опис та характеристики).