

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**МІКРОПРОЦЕСОРНА СИСТЕМА ТОЧНОГО ВИМІРЮВАННЯ І
ПОЗИЦІОНУВАННЯ В ПРОСТОРІ ДЕТАЛЕЙ ДЛЯ МЕТАЛООБРОБКИ**

Виконав студент 2 курсу, групи 2КІ-24м
спеціальності 123 — Комп'ютерна інженерія


Добровольський Д.С.

Керівник к.т.н., доц. каф. ОТ


Муращенко О.Г.

" "

2025 р.

Опонент к.т.н., доц. каф. МБІС


Грицак А. В.

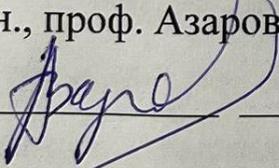
" "

2025 р.

Допущено до захисту

Завідувач кафедри ОТ

д.т.н., проф. Азаров О.Д.



" "

2025 р.

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

Освітньо-кваліфікаційний рівень магістр

Галузь знань — 12 — Інформаційні технології

Спеціальність — 123 — Комп'ютерна інженерія

Освітньо-професійна програма — Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної техніки

д.т.н., проф. Азаров О.Д.

" 25 " 09 2025 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студенту **Добровольському Данілу Сергійовичу**

1 Тема роботи «Мікропроцесорна система точного вимірювання і позиціонування в просторі деталей для металообробки» керівник роботи Муращенко Олександр Геннадійович к.т.н., доц. каф. ОТ, затверджено наказом вищого навчального закладу від 24.09.2025 року № 313

2 Строк подання студентом роботи 04.12.2025.

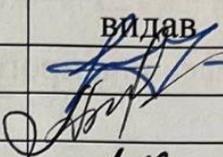
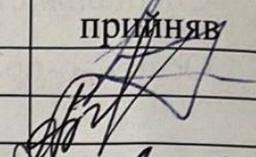
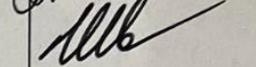
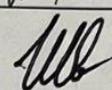
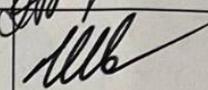
3 Вихідні дані до роботи: Вихідні дані до роботи: пристрій для точного вимірювання та позиціонування на верстатах із підключенням валового оптичного енкодера, наявністю зовнішніх органів керування, програмно реалізованою процедурою калібрування та відображення інформації на вбудованому дисплеї.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз існуючих систем вимірювання та позиціонування, дослідження методів вимірювання та підбір компонентів для створення пристрою, проектування та розробка, економічна частина.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): принципова електрична схема пристрою, блок-схема програмного забезпечення пристрою.

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 — Календарний план

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
	к.т.н., доц. каф. ОТ Муращенко О.Г.		
	к.т.н., доц. каф. ЕПВМ Адлер О.О.		
Нормо контроль	асист. Каф ОТ Швець С.І.		

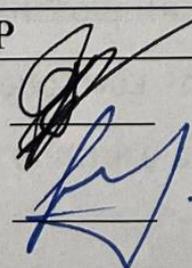
7 Дата видачі завдання 25.09.2025 р.

8 Календарний план виконання МКР наведено в таблиці 2.

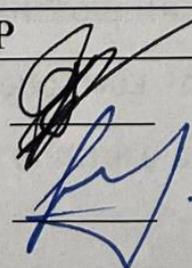
Таблиця 2 — Календарний план

№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	2	3	4	5
1	Вибір та затвердження теми МКР	25.09.2025	30.09.2025	виконано
2	Огляд існуючих рішень	01.10.2025	04.10.2025	виконано
3	Дослідження методів вимірювання	05.10.2025	11.10.2025	виконано
4	Підбір необхідних компонентів для виготовлення прототипу	12.10.2025	17.10.2025	виконано
5	Виготовлення та налагодження	18.10.2025	22.10.2025	виконано
6	Усунення недоліків та остаточна перевірка	23.10.2025	28.10.2025	виконано
7	Оформлення роботи у вигляді пояснювальної записки	29.10.2025	04.11.2025	виконано
8	Перевірка виконаної роботи	05.11.2025	07.11.2025	виконано
9	Остаточні виправлення, збирання підписів та зшивання роботи	08.11.2025	10.11.2025	виконано
10	Попередній захист	10.11.2025	11.11.2025	виконано
11	Захист МКР	16.12.2025	22.12.2025	виконано

Студент

 Добровольський Д.С.

Керівник роботи

 - к.т.н., доц. каф. ОТ Муращенко О.Г.

АНОТАЦІЯ

УДК 004.056.53

Добровольський Д.С. Мікропроцесорна система точного вимірювання і позиціонування в просторі деталей для металообробки. Магістерська кваліфікаційна робота зі спеціальності 123 — Комп'ютерна Інженерія, Вінниця: ВНТУ, 2025 — 125 с.

На укр. мові. Бібліогр.: 20 назв; рис.: 39; табл 10;

У роботі було розроблено функціонуючий прототип пристрою для точного вимірювання та позиціонування деталей на верстатах. Розглянуто підходи до перетворення обертального руху валу в кутові та лінійні координати, оцінено джерела похибок і принципи калібрування. Розроблено електричну схему з підключенням оптичного енкодера, індикатора та органів керування, а також програмне забезпечення з підтримкою збереження користувацьких одиниць в енергонезалежній пам'яті. Спроектовано та виготовлено корпус засобами 3D-друку, обґрунтовано компоновку елементів і зручність експлуатації. Отримані результати підтверджують практичну придатність пристрою для модернізації верстатів у навчальних і виробничих умовах.

Ключові слова: позиціонування, оптичний енкодер, калібрування, індикація, мікрометр.

ABSTRACT

УДК 004.056.53

Dobrovolskyi D.S. Microprocessor system for precision measurement and Spatial Positioning of Metalworking Parts. Master's qualification thesis in specialty 123 — Computer Engineering, Vinnytsia: VNTU, 2025 — 125 p.

In Ukrainian. Bibliography: 20 titles; figures:39; tables: 10.

The work developed a functioning prototype of a device for precise measurement and positioning of parts on machine tools. Approaches to converting the rotational motion of a shaft into angular and linear coordinates were considered, sources of errors and calibration principles were assessed. An electrical circuit was developed with the connection of an optical encoder, indicator and controls, as well as software with support for saving user units in non-volatile memory. The housing was designed and manufactured using 3D printing, the layout of elements and ease of operation were substantiated. The results obtained confirm the practical suitability of the device for the modernization of machine tools in educational and production conditions.

Keywords: positioning, optical encoder, calibration, indication, micrometer.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ВИМІРЮВАННЯ ТА ПОЗИЦІОНУВАННЯ	12
1.1 Роль і значення систем вимірювання у процесах металообробки.....	12
1.2 Класифікація систем вимірювання у верстатах: контактні, безконтактні, комбіновані .	13
1.3 Принципи роботи механічних систем вимірювання.....	16
1.4 Оптичні системи позиціонування та вимірювання: енкодери, лінійки, датчики.....	19
1.5 Індуктивні та ємнісні методи визначення положення	22
1.6 Цифрові системи індикації та позиціонування (DRO-системи)	24
1.7 Мікропроцесорні засоби контролю положення і руху у верстатах	27
1.8 Тенденції розвитку та автоматизації вимірювальних систем у промисловості	30
1.9 Системи калібрування та налаштування точності вимірювальних пристроїв	32
1.10 Порівняльний аналіз сучасних систем вимірювання	34
2 ДОСЛІДЖЕННЯ МЕТОДІВ ВИМІРЮВАННЯ ТА ПІДБІР КОМПОНЕНТІВ	36
2.1 Методика дослідження та критерії ефективності.....	36
2.2 Математична модель дискретного підрахунку подій	38
2.3 Перетворення обертального руху в кутові й лінійні координати.....	41
2.4 Обґрунтування вибору електронних компонентів та модулів	43
2.4.1 Вибір платформи.....	43
2.4.2 Вибір індикатора.....	47
2.4.3 Вибір елементів керування	50
2.4.4 Вибір компонентів для вимірювального тракту	53
3 РОЗРОБКА КОРПУСУ ТА ЕЛЕКТРИЧНОЇ СХЕМИ	56
3.1 Формування вимог до корпусу пристрою	56
3.2 Створення 3D-моделі корпусу у середовищі AutoCAD	58
3.3 Розробка електричної схеми.....	64
4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	67
4.1 Обмеження операцій по часу та пропускну здатність переривань	67
4.2 Обґрунтування застосування ООП та структура головного класу.....	70
4.3 Опис функцій обробки вхідних даних.....	74
4.4 Опис функцій відображення інтерфейсу користувача.....	78
5 ТЕСТУВАННЯ	84
6 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ	87
6.1 Проведення комерційного та технологічного аудиту розробки	87
6.2 Розрахунок витрат на здійснення розробки	89

6.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	96
ВИСНОВКИ	102
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	103
ДОДАТОК А Технічне завдання	106
ДОДАТОК Б Протокол перевірки кваліфікаційної роботи	111
ДОДАТОК В Алгоритм програми	112
ДОДАТОК Г Схема електрична принципова	113
ДОДАТОК Д Лістинг коду програми	114

ВСТУП

Розвиток промислових технологій завжди тісно пов'язаний із підвищенням точності обробки матеріалів і вдосконаленням засобів контролю. У металообробці, де навіть незначне відхилення може вплинути на якість готового виробу, питання вимірювання та позиціонування деталей має особливе значення. Висока точність стає базовою передумовою ефективної роботи обладнання, тому конструктори постійно шукають нові рішення, що дозволяють поєднати надійність механічних систем із гнучкістю електронного керування.

Одним із перспективних напрямів розвитку є впровадження мікропроцесорних систем, які здатні забезпечити високу точність вимірювань і одночасно значно розширити функціональні можливості обладнання. Такі пристрої дають змогу виконувати не лише вимірювання, а й обробку даних, калібрування, збереження результатів та автоматичне позиціонування інструмента або деталі. Це створює передумови для модернізації верстатів без необхідності повної заміни їх механічної частини.

Технологічні процеси дедалі частіше поєднують механічні й електронні системи в єдину інтелектуальну структуру. Це дозволяє не лише підвищити точність і стабільність роботи обладнання, а й гнучко адаптувати його під конкретні виробничі завдання. Саме мікропроцесорні системи забезпечують цей перехід, даючи можливість з високою точністю і швидкістю аналізувати показники, реагувати на зміни умов і забезпечити швидке позиціонування інструмента чи деталі. Завдяки широкій доступності платформ для розробки, таких як Arduino, стало можливим створювати високоточні вимірювальні пристрої навіть у невеликих майстернях, не вдаючись до дорогих промислових рішень.

Крім того, розвиток адитивних технологій, зокрема 3D-друку, відкрив нові можливості для швидкого прототипування корпусів, тримачів, кріплень та інших елементів вимірювальної системи. Це дозволяє розробникам створювати індивідуальні рішення під специфіку конкретного верстата або робочого

процесу. В умовах, коли виробництво потребує постійної модернізації та підвищення ефективності, можливість швидко виготовити і протестувати власний вимірювальний пристрій є важливою конкурентною перевагою. Саме тому тема розробки мікропроцесорної системи точного вимірювання та позиціонування набуває особливої актуальності для сучасної металообробки.

Більшість верстатів, які використовуються у невеликих майстернях, навчальних лабораторіях або на малих підприємствах, мають застарілі механічні системи контролю переміщень, зокрема мікрометри та індикатори годинникового типу. Вони не забезпечують належної точності, не мають можливості цифрового зчитування та не мають можливості автоматизації. Водночас промислові системи цифрової індикації позицій (так звані DRO-системи) є доволі дорогими, складними у встановленні та вимагають спеціалізованого обладнання для інтеграції. У такій ситуації актуальним є пошук недорогих, універсальних і гнучких технічних рішень, які могли б забезпечити високу точність вимірювання без суттєвих фінансових витрат і складного налаштування.

Розроблена у межах даної магістерської роботи мікропроцесорна система точного вимірювання і позиціонування в просторі для верстатів призначена для забезпечення точності контролю положення елементів обладнання на базі доступних компонентів. Основу системи становить мікроконтролер Arduino Nano, який обробляє сигнали оптичного енкодера з роздільною здатністю 1600 імпульсів на оберт. Результати вимірювань виводяться на компактний OLED-дисплей, а взаємодія користувача з системою відбувається за допомогою трьох тактових кнопок. Корпус пристрою спроектовано в середовищі AutoCAD і виготовлено методом 3D-друку, що забезпечує компактність, ергономічність та можливість швидкої модернізації під конкретні потреби.

Розроблений пристрій є доволі універсальним, оскільки може застосовуватись на різних типах верстатів — від токарних до фрезерних. Його можна встановлювати замість штатного механічного мікрометра або інтегрувати у будь-який вузол, де необхідно контролювати переміщення валу, осі або

поступальний рух елемента. Залежно від конкретної конструкції обладнання енкодер може працювати як кутовий або лінійний датчик. Пристрій підтримує можливість програмного калібрування, збереження індивідуальних одиниць вимірювання в енергонезалежній пам'яті, встановлення міток у процесі вимірювання та тимчасового призупинення оновлення показників, що зручно при роботі з деталями, які потребують складних налаштувань.

Актуальність роботи визначається потребою у створенні доступних і точних вимірювальних систем для модернізації існуючих верстатів, особливо тих, які не обладнані сучасними засобами цифрової індикації. Розроблений пристрій дозволяє досягти високої точності без суттєвого ускладнення конструкції верстата, забезпечує цифрову індикацію показників і може бути використаний як окремий вимірювальний модуль або як частина більших автоматизованих комплексів.

Метою дослідження є створення, виготовлення та випробування мікропроцесорної системи, здатної забезпечити точне вимірювання та позиціонування деталей. При цьому особливу увагу приділено розробці не лише апаратної, а й програмної частини, що дозволяє користувачу задавати, редагувати і зберігати одиниці вимірювання, проводити калібрування, ставити мітки і контролювати рух у реальному часі.

Поставлені **задачі**: розробити програмне забезпечення яке буде реалізувати надійний підрахунок подій і зручне меню керування; спроектувати та зібрати електричну схему; реалізувати алгоритм калібрування та збереження користувацьких одиниць у пам'яті; спроектувати 3D-корпус із зручним доступом до роз'ємів і надійною фіксацією внутрішніх елементів.

Об'єктом дослідження є процес вимірювання і позиціонування рухомих елементів верстатного обладнання за допомогою мікропроцесорних засобів.

Предметом дослідження виступає розроблений пристрій, його конструкція, програмна реалізація, а також принципи роботи з оптичними енкодерами для досягнення максимальної точності.

Під час виконання роботи застосовувалися аналітичні та експериментальні **методи**, що дозволили дослідити існуючі системи вимірювання, визначити їх недоліки, обґрунтувати вибір компонентів та провести випробування створеного прототипу. Було здійснено моделювання конструкції у середовищі AutoCAD, розроблено електричну схему підключення компонентів та проведено експериментальні вимірювання на реальному верстаті для визначення точності і стабільності показників.

Наукова новизна полягає у підвищенні універсальності та розширенні функціональних можливостей вимірювальної системи верстатів без зміни апаратної частини, методами зміни коефіцієнтів перетворення між обертальним і поступальним рухом, можливості збереження користувацьких налаштувань у енергонезалежній пам'яті та реалізації функцій, які раніше не використовувалися у подібних компактних пристроях.

Практичне значення одержаних результатів полягає у можливості використання розробленого пристрою для модернізації існуючих верстатів без великих витрат. Пристрій уже виготовлений і успішно протестований у реальних умовах. Його можна встановлювати на токарних, фрезерних та інших верстатах, де потрібен контроль точного положення елементів.

Апробація результатів роботи була проведена на конференції «Молодь в науці: дослідження, проблеми, перспективи». За результатами досліджень **опубліковано** тези доповіді на науково–технічній конференції [1].

1 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ВИМІРЮВАННЯ ТА ПОЗИЦІОНУВАННЯ

1.1 Роль і значення систем вимірювання у процесах металообробки

У металообробці точність є однією з ключових характеристик, яка визначає якість і придатність готової деталі до подальшої експлуатації. Будь-який процес різання, фрезерування чи токарної обробки передбачає суворе дотримання геометричних параметрів, заданих у кресленні. Відхилення навіть на кілька сотих міліметра можуть призвести до порушення функціональності вузла, швидкого зношування чи відмови усієї конструкції. Тому системи вимірювання та контролю положення відіграють критичну роль у забезпеченні стабільності та повторюваності результатів обробки.

Основне завдання вимірювальної системи забезпечити зворотний зв'язок між фактичним положенням робочого інструмента чи заготовки та заданими параметрами керування. Це дозволяє компенсувати похибки механіки, знос елементів, температурні деформації та інші фактори, що впливають на точність. У найпростіших випадках вимірювання здійснюється оператором вручну за допомогою мікрометрів, індикаторів, штангенциркулів чи механічних лімбів. Проте такі методи мають обмежену швидкість і залежать від людського чинника, що не відповідає вимогам високопродуктивного виробництва.

Зі збільшенням складності деталей та зменшенням допусків виникла потреба у впровадженні автоматизованих вимірювальних систем, здатних у реальному часі визначати положення елементів з високою роздільною здатністю. Вони забезпечують не лише вимірювання, а й інтеграцію з системами керування верстатом, що дозволяє виконувати прецизійні операції з мінімальним втручанням людини. Сучасні системи позиціонування у верстатах стали невід'ємною частиною технологічного процесу, забезпечуючи якість, повторюваність і надійність обробки.

Ще одним важливим аспектом є економічний ефект. Використання точних вимірювальних систем знижує кількість браку, скорочує час налаштування і дає можливість обробляти складніші деталі без додаткових контрольних операцій.

Це підвищує ефективність виробництва та конкурентоспроможність підприємства. Крім того, автоматизовані вимірювальні системи спрощують роботу оператора, зменшуючи потребу в ручному контролі і тим самим підвищуючи безпеку праці.

У загальному розумінні системи вимірювання в металорізальних верстатах можна розглядати як ключовий елемент точного машинобудування. Вони не лише фіксують положення, а й формують інформаційне середовище для управління процесом. Саме через них здійснюється взаємодія між фізичними параметрами реального об'єкта та цифровою моделлю у керуючій системі. Таким чином, розвиток технологій вимірювання напряду визначає можливості подальшого вдосконалення верстатів і підвищення точності металообробки.

1.2 Класифікація систем вимірювання у верстатах: контактні, безконтактні, комбіновані

Системи вимірювання, які застосовуються у металорізальних верстатах, мають велике різноманіття конструкцій, принципів роботи та рівнів автоматизації. Їх класифікація базується на способі взаємодії з об'єктом вимірювання, типі датчиків, фізичних принципах перетворення сигналів, а також на рівні інтеграції з керуючими системами. Найпоширенішим критерієм поділу є принцип контакту — за цим параметром розрізняють контактні, безконтактні та комбіновані вимірювальні системи.

Контактні системи вимірювання (рис. 1.1) є найдавнішими й найпростішими за конструкцією. Вони базуються на механічному контакті між вимірювальним елементом і поверхнею деталі. Прикладами таких систем є індикатори годинникового типу, штангенциркулі, мікрометри, механічні лімби з нониусами, а також системи з механічним передаванням руху до стрілочного або цифрового індикатора. Головною перевагою контактних систем є їхня простота, надійність і невисока вартість. Вони не потребують складної електроніки, не чутливі до пилу, мастил чи перепадів температури. Проте головним недоліком є залежність точності від стану поверхні, зносу вимірювального наконечника та

зусилля притискання. Крім того, такі системи не підходять для автоматизованих процесів, оскільки потребують ручної взаємодії оператора.



Рисунок 1.1 — Приклад систем контактного вимірювання

Безконтактні системи (рис 1.2) вимірювання є більш прогресивними, оскільки усувають недоліки, пов'язані з механічним контактом. Вони використовують різні фізичні принципи — оптичний, індуктивний, ємнісний, лазерний, ультразвуковий тощо.

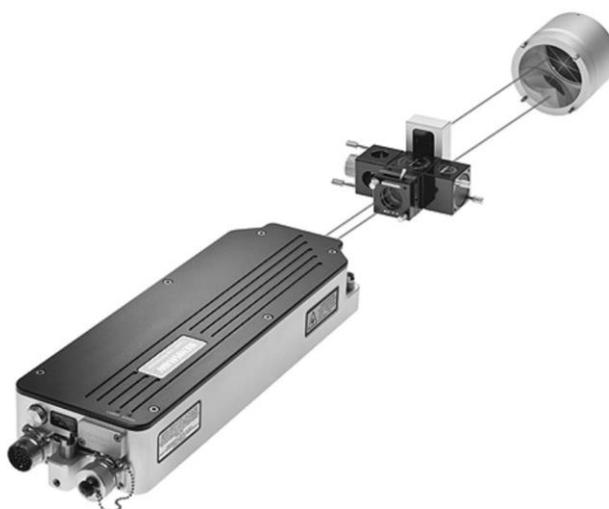


Рисунок 1.2 — Система безконтактного вимірювання

Оптичні системи, зокрема енкодери або лінійки (рис.1.3), перетворюють переміщення у світлові імпульси, які далі інтерпретуються електронною схемою. Індуктивні та ємнісні датчики вимірюють зміну електромагнітного поля або ємності між елементами, що змінюється при русі. Такі системи мають високу швидкодію, стабільність та можуть працювати у складі числового програмного керування (ЧПК). Їхнім недоліком є чутливість до забруднень, вібрацій і необхідність точного монтажу, а також вища вартість порівняно з механічними засобами.



Рисунок 1.3 — Промисловий оптичний енкодер

Комбіновані системи вимірювання поєднують елементи обох підходів. У таких рішеннях механічна частина забезпечує базовий контакт або утримання деталі, а безконтактні сенсори відповідають за точне визначення положення або переміщення. Прикладом можуть бути прецизійні вимірювальні головки, де механічний щуп з'єднаний із оптичним або індуктивним сенсором, що передає сигнал у цифрову систему контролю. Такий підхід дозволяє зберегти надійність і простоту механіки, але водночас підвищити точність, стабільність і можливість автоматичного зчитування даних.

Окрім класифікації за принципом дії, вимірювальні системи поділяються за типом переміщення: лінійні та кутові. Лінійні використовуються для вимірювання поступальних переміщень — наприклад, у напрямках подачі фрезерного верстата або переміщення супорта токарного. Кутові вимірювальні

системи застосовуються для визначення обертів або поворотів елементів, таких як шпинделі, осі інструментів чи регулювальні ручки. Оптичні енкодери, що вимірюють кут повороту валу, є типовим прикладом кутових систем, тоді як оптичні лінійки чи магнітні стрічки — прикладом лінійних.

Ще один підхід до класифікації передбачає поділ за рівнем автоматизації. Пасивні системи лише інформують оператора про положення, тоді як активні інтегровані у систему керування верстатом і беруть участь у зворотному зв'язку. Активні системи дають змогу створювати адаптивні технологічні процеси, де керування рухами інструмента відбувається в реальному часі залежно від показників датчиків. Такий підхід лежить в основі сучасних систем числового програмного керування.

Загалом класифікація вимірювальних систем дозволяє зрозуміти еволюцію від простих механічних інструментів до складних електронних комплексів. Кожен тип має своє призначення і сферу застосування. Контактні пристрої залишаються актуальними для базових операцій і налаштування, тоді як безконтактні та комбіновані рішення поступово стають стандартом для високоточної металообробки. Саме поєднання цих підходів — механічної надійності й електронної точності — визначає ефективність і гнучкість сучасних верстатних систем.

1.3 Принципи роботи механічних систем вимірювання

Механічні системи вимірювання є основою точного машинобудування та залишаються важливим інструментом навіть за наявності складних електронних рішень. Вони забезпечують вимірювання геометричних параметрів, переміщень та відхилень у процесі обробки деталей. Основний принцип їхньої дії полягає у безпосередньому або опосередкованому перетворенні лінійного чи кутового переміщення вимірювального елемента в показник, який може бути сприйнятий оператором або переданий іншим системам контролю.

Механічні вимірювальні системи найчастіше використовують принцип важеля, гвинтової пари або передачі руху через систему зубчастих коліс.

Наприклад, у мікрометрі переміщення вимірювального гвинта перетворюється у показ на шкалі за допомогою точно виготовленої різьби. В індикаторі годинникового типу невелике лінійне переміщення штока перетворюється в обертання стрілки через систему важелів і зубчастих передач (рис. 1.4). Такий механічний підхід забезпечує достатню точність і повторюваність при невисокій вартості виготовлення.

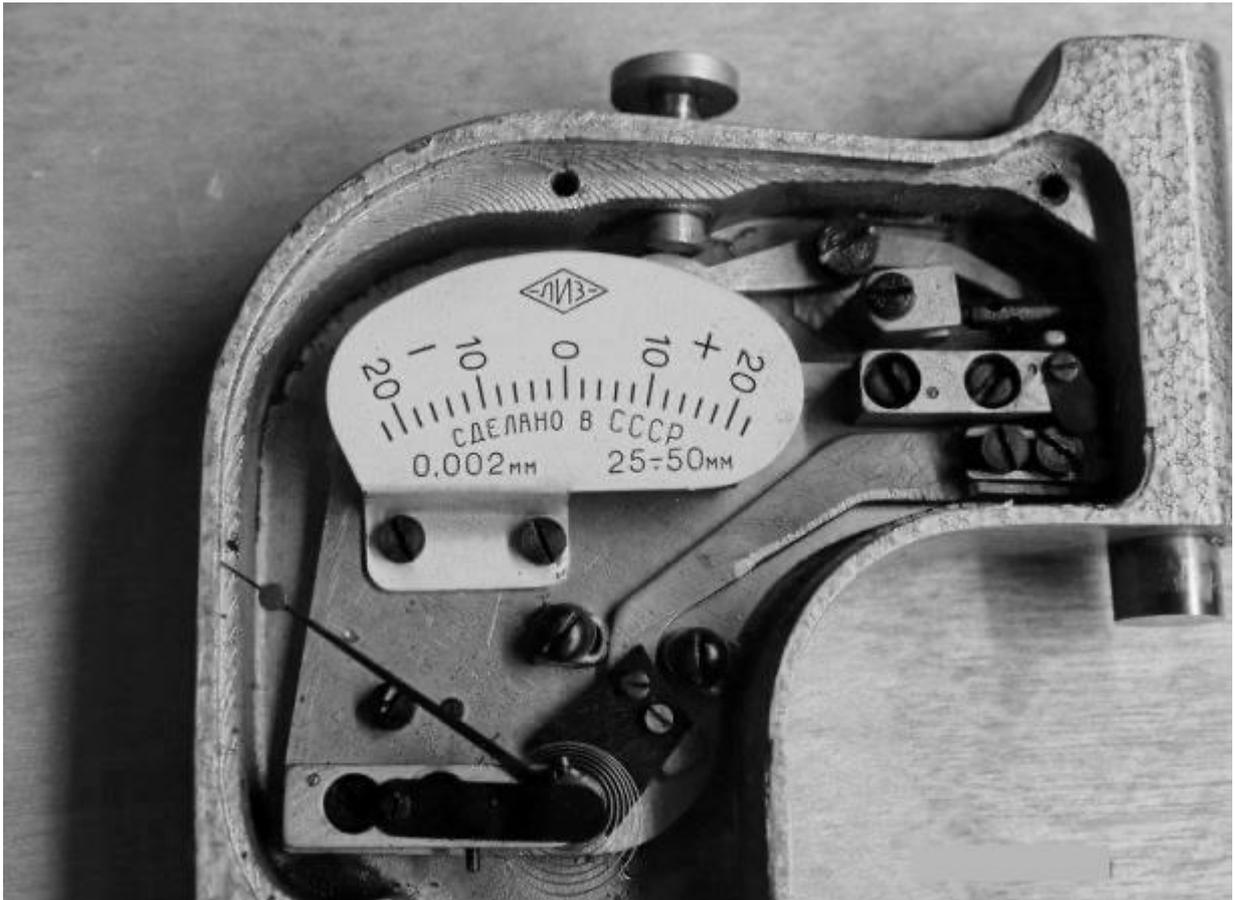


Рисунок 1.4 — Внутрішня будова годинникового мікрометра

Простота конструкції є головною перевагою механічних вимірювальних систем. Вони не потребують живлення, стійкі до впливу електромагнітних полів, мають довгий термін служби і не потребують складного технічного обслуговування. Завдяки цим якостям механічні індикатори і шкали досі широко використовуються для налаштування та контролю у верстатах, зокрема в токарних, фрезерних, розточувальних і шліфувальних. У ручному режимі роботи верстатів механічні шкали на лімбах дозволяють оператору відстежувати

переміщення подачі та глибину різання.

Водночас механічні системи мають низку обмежень, що зумовили їх поступову заміну на електронні аналоги. Головною проблемою є залежність точності від стану механічних з'єднань. Будь-який люфт у передачі або знос деталей призводить до похибок показань. Крім того, механічні шкали мають обмежену роздільну здатність та обмежений діапазон вимірювань, що ускладнює виконання надточних операцій. Також оператор змушений візуально зчитувати покази, що збільшує час контролю та створює ризик людських помилок.

Попри ці недоліки, механічні системи залишаються важливим елементом верстатів з ручним або напівавтоматичним керуванням. У таких випадках вони виконують роль основного засобу вимірювання або резервного індикатора у разі відмови електронних систем. Механічні шкали також часто застосовують для калібрування або перевірки точності електронних датчиків, що свідчить про їхню надійність і довіру до результатів вимірювань.

Окрему групу становлять механічні системи з оптичним відліком, де стрілковий індикатор або шкала з'єднані з оптичною системою збільшення (лупою або мікроскопом). Такі рішення дозволяють досягти підвищеної точності без використання електроніки. Наприклад, оптичні мікрометри або спеціальні вимірювальні головки здатні фіксувати переміщення з точністю до кількох мікрон.

Сьогодні механічні вимірювальні прилади часто використовуються у поєднанні з цифровими індикаторами, які лише зчитують переміщення, але сама передача залишається механічною. Така інтеграція дозволяє зберегти простоту і надійність механічних систем, одночасно розширивши їх функціональність — наприклад, забезпечивши цифрову індикацію, калібрування або обмін даними з іншими пристроями.

Отже, принципи роботи механічних вимірювальних систем базуються на перетворенні механічного переміщення в показ на шкалі, при цьому ключовими факторами точності є якість виготовлення, жорсткість деталей і мінімізація люфтів. Хоча такі системи поступово витісняються безконтактними й

мікропроцесорними рішеннями, вони залишаються незамінними у простих, надійних та мало бюджетних верстатних конструкціях, де стабільність і довговічність важливіші за автоматизацію.

1.4 Оптичні системи позиціонування та вимірювання: енкодери, лінійки, датчики

Оптичні системи вимірювання та позиціонування відіграють важливу роль у металорізальних верстатах, особливо коли йдеться про підвищену точність і автоматизацію процесів. Основна перевага таких систем полягає у безконтактному способі визначення переміщення, що дозволяє уникнути проблем, характерних для механічних пристроїв, таких як знос деталей, люфти та нестабільність показань. Оптичні методи забезпечують високу роздільну здатність, швидке реагування та можливість інтеграції з цифровими системами керування.

Одним із найпоширеніших рішень є використання оптичних енкодерів. Вони бувають інкрементними (рис 1.5) та абсолютними (рис 1.6). Інкрементні енкодери в свою чергу генерують серію імпульсів під час обертання вала або переміщення лінійки. Кількість імпульсів на оберт визначає роздільну здатність, і, знаючи їхню суму, можна обчислити зміщення або кут повороту.



Рисунок 1.5 — Інкрементний оптичний енкодер

Абсолютні енкодери відрізняються тим, що кожне положення валу або шляху відповідає унікальному цифровому коду завдяки особливому розміщенню прорізів на тілі диска (рис. 1.7). Це дозволяє відновлювати точне положення навіть після відключення живлення або перерваної роботи системи. Така характеристика робить їх незамінними для верстатів із числовим програмним керуванням та автоматизованих виробничих ліній.



Рисунок 1.6 — Абсолютний оптичний енкодер

Оптичні лінійки й шкали (рис.1.8) також широко застосовуються для визначення лінійного переміщення. Вони представляють собою прозору або відбивну шкалу з нанесеними поділками, яку зчитує оптичний датчик. Світловий промінь, що проходить або відбивається від шкали, перетворюється на електричний сигнал, який фіксує переміщення з високою точністю. Важливим аспектом таких систем є необхідність чистоти робочої поверхні та точного монтажу, оскільки забруднення або невірна установка можуть знизити точність вимірювань. Оптичні датчики поділяються на різні типи залежно від фізичного принципу роботи. Наприклад, фотодіодні та фототранзисторні системи визначають зміни інтенсивності світла, що проходить крізь поділки шкали.

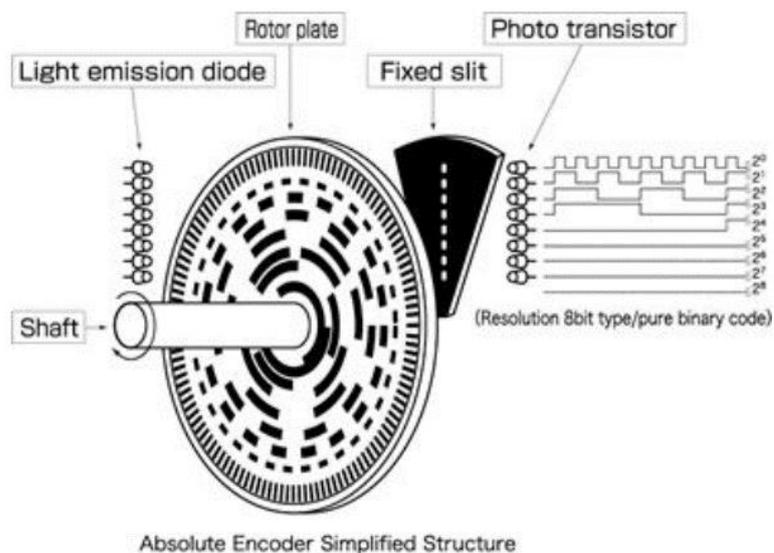


Рисунок 1.7 — Внутрішня будова абсолютного енкодера

Оптичні датчики поділяються на різні типи залежно від фізичного принципу роботи. Наприклад, фотодіодні та фототранзисторні системи визначають зміни інтенсивності світла, що проходить крізь поділки шкали. Інтерферометричні системи використовують принцип інтерференції світлових хвиль, що дозволяє досягати мікронної та навіть субмікронної точності. Лазерні датчики позиціонування забезпечують високу точність на великих відстанях і швидкодію, що особливо важливо для сучасних високошвидкісних верстатів.

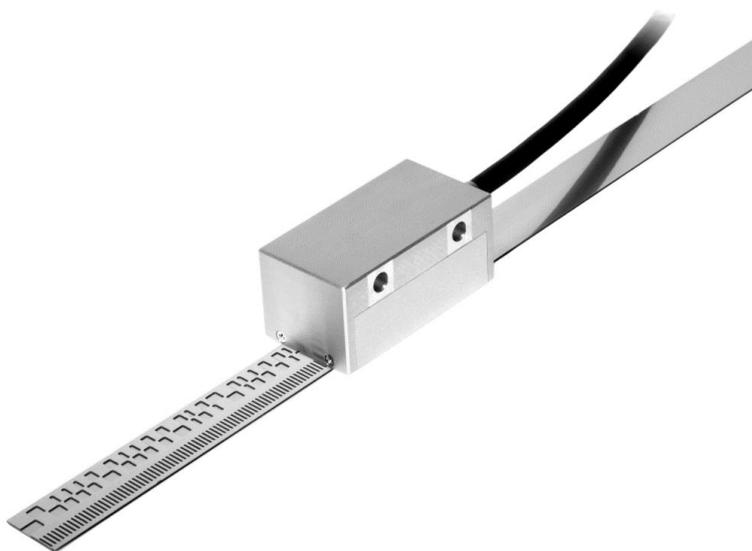


Рисунок 1.8 — Вигляд лінійного енкодера

Переваги оптичних систем очевидні: висока точність, можливість безконтактного вимірювання, інтеграція з електронними системами та сумісність із числовим програмним керуванням. Водночас вони мають і певні обмеження, серед яких чутливість до забруднення, вібрацій та температурних коливань. Це зумовлює необхідність дотримання умов експлуатації, періодичного технічного обслуговування та, у деяких випадках, додаткових захисних корпусів або покриттів.

Особливе значення оптичні системи набувають у тих випадках, коли потрібна інтеграція з мікропроцесорними системами. Сигнали від енкодера або оптичної лінійки можуть безпосередньо надходити на контролер, який обробляє дані, виконує калібрування, зберігає одиниці вимірювання та забезпечує відображення на дисплеї. Це дозволяє створювати універсальні вимірювальні пристрої, які легко адаптуються до різних типів верстатів та конфігурацій, забезпечуючи одночасно високоточний контроль і зручність користування.

Таким чином, оптичні системи позиціонування та вимірювання є важливим кроком у розвитку точних, швидких і гнучких технологічних процесів. Вони дозволяють поєднати механічну надійність та точність з цифровою обробкою даних, що робить їх незамінними у високоточній металообробці та модернізації існуючих верстатів. Розуміння принципів роботи, переваг і обмежень цих систем є необхідним для проектування власних мікропроцесорних вимірювальних пристроїв та їх інтеграції у виробничі процеси.

1.5 Індуктивні та ємнісні методи визначення положення

Серед безконтактних методів вимірювання індуктивні та ємнісні системи займають важливе місце завдяки своїй простоті інтеграції, надійності та високій швидкодії (рис. 1.9). Ці методи базуються на електромагнітних явищах і дозволяють визначати положення деталей або рухливих елементів без прямого контакту, що знижує знос і збільшує довговічність системи.

Індуктивні датчики визначають переміщення об'єкта на основі зміни електромагнітного поля. Основний принцип їх роботи полягає у взаємодії

катушки з металевим елементом, який рухається. При наближенні металевої деталі до катушки змінюється індуктивність, що перетворюється на електричний сигнал.

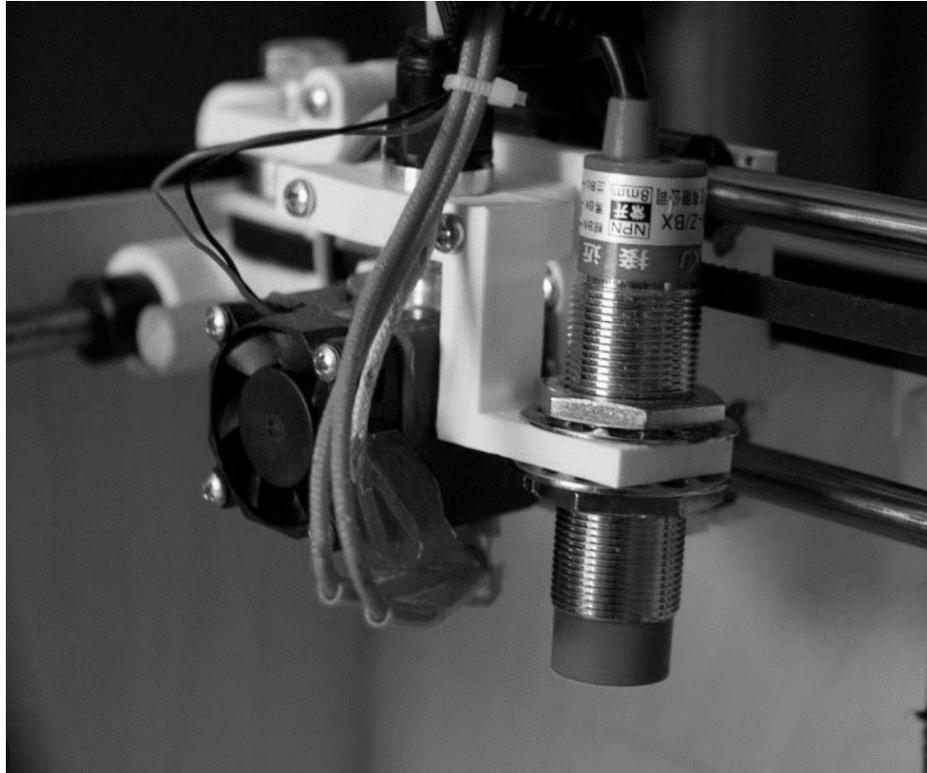


Рисунок 1.9 — Індуктивний датчик встановлений на друкуючій голові 3D-принтера

Такі системи широко застосовуються для контролю кутового або лінійного переміщення у верстатах, у яких потрібна висока надійність роботи в умовах вібрацій, мастила та пилу. Індуктивні датчики здатні визначати переміщення з точністю до десятків мікрон, що робить їх конкурентоздатними порівняно з оптичними рішеннями в багатьох промислових умовах.

Ємнісні датчики вимірюють зміни ємності між електродами, спричинені переміщенням об'єкта. При зміні відстані між рухомим і нерухомим електродом ємність змінюється, що фіксується електронним блоком і перетворюється на сигнал, пропорційний переміщенню. Ці системи відрізняються високою точністю та швидкістю, але чутливі до електричних перешкод і зміни вологості або температури. Ємнісні методи зазвичай застосовуються для вимірювання

малих переміщень у прецизійних верстатах, де необхідна точність до кількох мікрон.

Важливою перевагою індуктивних та ємнісних методів є можливість безконтактного контролю, що усуває проблеми з механічним зносом і люфтами, характерні для традиційних контактних пристроїв. Також ці системи забезпечують високу повторюваність результатів і дозволяють легко інтегруватися з цифровими системами керування, такими як мікропроцесорні контролери або числове програмне керування (ЧПК). Це робить їх незамінними у сучасних верстатах, де одночасно потрібна швидкість і точність.

Недоліком цих систем є обмеження робочого діапазону та чутливість до зовнішніх факторів. Індуктивні датчики ефективні лише для металевих об'єктів, а ємнісні - для провідних або діелектричних елементів певного розміру. Обидва типи датчиків вимагають точного монтажу і стабільного живлення для забезпечення високої точності вимірювань.

Сучасні індустріальні рішення часто комбінують індуктивні та ємнісні датчики з іншими методами вимірювання. Наприклад, у верстатах високої точності ці системи доповнюють оптичні енкодери або цифрові індикатори, створюючи гібридні комплекси, які дозволяють підвищити точність і надійність контролю положення. Комбінація методів забезпечує баланс між швидкодією, точністю та стійкістю до зовнішніх факторів.

Таким чином, індуктивні та ємнісні методи визначення положення є ефективними інструментами для безконтактного вимірювання у верстатах. Їх застосування дозволяє підвищити точність, надійність і довговічність роботи обладнання, що робить ці методи актуальними для модернізації існуючих верстатів та розробки прецизійних мікропроцесорних систем вимірювання.

1.6 Цифрові системи індикації та позиціонування (DRO-системи)

Цифрові системи індикації та позиціонування, або DRO (рис. 1.10) є важливим етапом розвитку вимірювальних систем для металорізальних верстатів. Вони поєднують механічні або безконтактні датчики з електронною

обробкою сигналу, що дозволяє в реальному часі відображати положення робочого інструмента або заготовки на цифровому дисплеї. Завдяки цьому оператор може точно контролювати хід подачі, глибину різання та положення деталі без необхідності додаткового обчислення або ручного зчитування даних зі шкали.

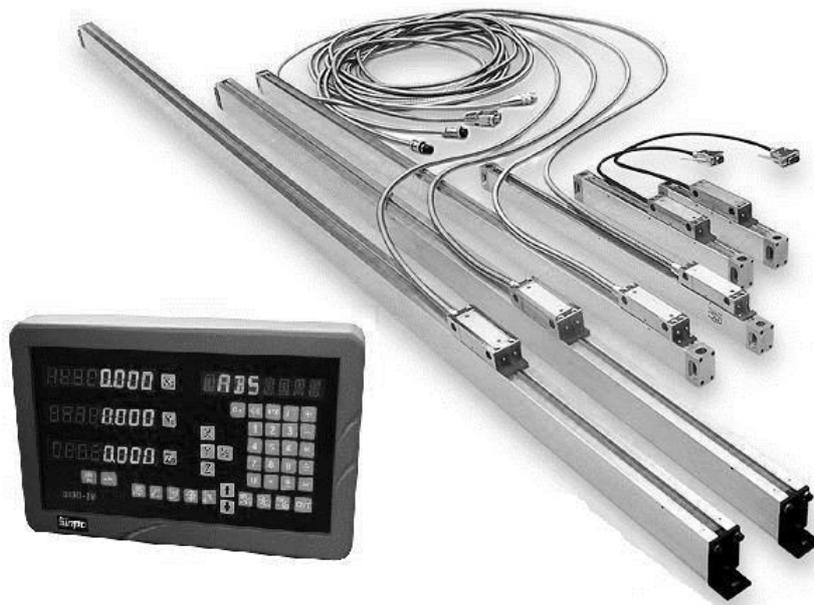


Рисунок 1.10 — Вигляд DRO-системи Dro Digital Readout System Milling PKG

Основними компонентами DRO-систем є датчики переміщення, електронний блок обробки сигналу та дисплей індикації. Датчики можуть бути різного типу — оптичні, індуктивні, ємнісні або механічні — залежно від вимог до точності та умов експлуатації. Сигнали від датчиків надходять на мікропроцесорний контролер, який здійснює перетворення імпульсів або аналогових змін у цифрові показники, відображає їх на дисплеї і, у деяких випадках, передає до системи керування верстатом.

Перевагою DRO-систем є значне підвищення точності роботи, можливість підключення різної кількості сенсорів (рис 1.11) та зниження впливу людського фактору. У ручних верстатах оператор може легко відстежувати переміщення супорта або шпинделя з точністю до сотих або навіть тисячних часток міліметра. Це особливо важливо при виконанні складних операцій, де необхідна висока

повторюваність та мінімізація похибок. Крім того, цифрові системи дозволяють зберігати нульові точки, виконувати обчислення різниці між позиціями та використовувати додаткові функції, наприклад, вимірювання відстані до контрольної точки або калібрування під конкретні одиниці вимірювання.



Рисунок 1.11 — Вигляд панелі DRO з роз’ємами для підключення датчиків

Цифрові системи індикації можуть бути інтегровані з ЧПК, що відкриває можливості для автоматизації обробки. Вони виконують роль зворотного зв’язку у системі управління, передаючи точну інформацію про положення елементів у реальному часі. Це дозволяє програмному забезпеченню коригувати подачу інструмента, уникати перевищення допусків та забезпечувати стабільну якість продукції. Відповідно, DRO-системи стають критично важливими для модернізації старих верстатів і впровадження сучасних технологічних процесів.

Сучасні DRO-системи надають широкі можливості налаштування та розширення функціоналу. Наприклад, вони можуть працювати з різними одиницями вимірювання, зберігати налаштування для різних операцій, здійснювати калібрування під специфіку конкретного верстата та навіть інтегруватися з персональним комп’ютером або мобільним пристроєм для моніторингу та аналізу. Завдяки цифровій індикації можна реалізувати функцію «заморожування» показів — коли обертання або переміщення виконується, а на

дисплеї значення залишається стабільним. Це особливо зручно для проведення складних вимірювань і контролю точності позиціонування.

Недоліком цифрових систем є потреба в стабільному живленні та захист від електричних перешкод. Також точність системи безпосередньо залежить від якості встановлених датчиків та точності їх монтажу. Проте при правильній експлуатації і регулярному технічному обслуговуванні DRO-системи забезпечують високу надійність, простоту використання і суттєве підвищення ефективності металорізальних операцій.

Таким чином, цифрові системи індикації та позиціонування стали стандартом для підвищення точності, автоматизації та зручності роботи на верстатах. Вони поєднують механічну надійність датчиків з можливостями обробки і відображення даних, що робить їх ключовим елементом у модернізації обладнання та створенні нових прецизійних мікропроцесорних вимірювальних систем.

1.7 Мікропроцесорні засоби контролю положення і руху у верстатах

З розвитком цифрових технологій та електроніки виникла потреба у впровадженні мікропроцесорних засобів контролю положення у верстатах. Такі системи дозволяють не лише відображати позицію інструмента або деталі, а й виконувати складні обчислення, інтегруватися з системою керування верстатом та забезпечувати автоматичний зворотний зв'язок. Мікропроцесорні рішення створюють основу для сучасних високоточних і автоматизованих технологічних процесів.

Основою таких систем є контролер або мікропроцесор, який приймає сигнали від різних датчиків — оптичних, індуктивних, ємнісних або механічних. На основі цих сигналів процесор визначає точне положення рухомого елемента і виконує необхідні обчислення для відображення на дисплеї або передачі даних у систему керування. Завдяки достатній обчислювальній потужності мікропроцесор може одночасно обробляти кілька потоків інформації,

виконувати калібрування та забезпечувати адаптивну реакцію системи на зміну умов обробки.

Переваги мікропроцесорного контролю очевидні. По-перше, він дозволяє значно підвищити точність і повторюваність позиціонування, що критично для операцій з малими допусками. По-друге, з'являється можливість зберігати налаштування та одиниці вимірювання в енергонезалежній пам'яті, а також здійснювати їхнє редагування без ручного перерахунку. По-третє, мікропроцесорні системи дозволяють реалізовувати функцію «заморожування» показів або встановлення контрольних точок, що значно спрощує виконання прецизійних операцій. Навіть найпростіша сучасна мікропроцесорна система для вимірювання надає в рази ширший функціонал порівняно з механічними аналогами (рис. 1.12).



Рисунок 1.12 — Набір базових функцій сучасної вимірювальної системи

Сучасні мікропроцесорні системи контролю положення можуть працювати з різними типами енкoderів і датчиків, що дозволяє адаптувати їх під конкретний верстат або технологічний процес. Наприклад, інкрементні оптичні енкoderи із високою роздільною здатністю забезпечують точність до сотих або тисячних часток градуса. Крім того, можливе використання редукторів,

черв'ячних передач або інших механічних трансмісій (рис. 1.13), які змінюють точність і масштаб одиниць вимірювання. Мікропроцесор враховує ці параметри і коригує відображувані дані, що дозволяє створювати універсальні пристрої для різних застосувань.



Рисунок 1.13 — Енкодер з редуктором

Інтеграція мікропроцесорних систем з електронними дисплеями та додатковими інтерфейсами(рис.1.14) відкриває нові можливості для оператора. Наприклад, на дисплеї, який легко інтегрується в систему на базі мікропроцесора, можна відображати відстань до контрольної точки, напрямок переміщення, значення у вибраних одиницях вимірювання та результати калібрування. Це дозволяє швидко орієнтуватися у процесі обробки, зменшує вплив людського фактора та підвищує ефективність роботи.



Рисунок 1.14 — Приклад застосування РК дисплея у вимірювальній системі

Мікропроцесорні системи контролю положення також є основою для модернізації застарілих верстатів. Встановлення такого пристрою на токарний або фрезерний верстат замість механічного мікрометра або ручного індикатора дозволяє значно підвищити точність обробки та автоматизувати процес контролю. Крім того, мікропроцесорні системи можуть інтегруватися з програмними засобами для аналізу даних, збереження історії вимірювань та підготовки звітів, що робить їх універсальним інструментом для сучасного виробництва.

Таким чином, мікропроцесорні засоби контролю положення і руху відіграють ключову роль у розвитку високоточних та автоматизованих технологічних процесів. Вони забезпечують точність, повторюваність, гнучкість та інтеграцію з іншими системами, що робить їх незамінними для модернізації верстатів і впровадження нових методів прецизійної металообробки.

1.8 Тенденції розвитку та автоматизації вимірювальних систем у промисловості

Вимірювальні системи металорізальних верстатів продовжують активно розвиватися, що зумовлено зростанням вимог до точності, швидкодії та гнучкості виробничих процесів. Сучасні тенденції спрямовані на інтеграцію механічних, оптичних, індуктивних та мікропроцесорних технологій у єдині комплексні системи, здатні виконувати автоматизований контроль положення та руху деталей і інструментів. Такий підхід дозволяє підвищувати продуктивність, знижувати похибки та мінімізувати вплив людського фактора.

Однією з ключових тенденцій є широке впровадження цифрових і безконтактних технологій. Оптичні енкодери, лазерні лінійки та індуктивні датчики стають стандартом у високоточних верстатах. Завдяки цифровим системам обробки сигналів, дані з датчиків перетворюються у зрозумілі для оператора або керуючого програмного забезпечення величини, що дозволяє реалізовувати автоматичний контроль та зберігання налаштувань. Безконтактні технології також забезпечують високу довговічність обладнання, оскільки

зменшують знос механічних елементів і потребу у частому технічному обслуговуванні.

Другим важливим напрямком розвитку є інтеграція вимірювальних систем із ЧПК та програмним забезпеченням промислових підприємств (рис 1.15). Це дозволяє не лише контролювати положення та переміщення інструмента, а й автоматично коригувати процес обробки на основі отриманих даних. Системи можуть зберігати історію вимірювань, проводити аналіз точності та оптимізувати операції, що значно підвищує ефективність виробництва та зменшує брак.



Рисунок 1.15 — Промислова система керування ЧПК

Активно впроваджуються гібридні системи, які поєднують переваги різних методів вимірювання. Наприклад, комбінація оптичного енкадера з індуктивним датчиком або механічною передачею дозволяє одночасно забезпечити високу точність, надійність і стабільність показань у складних умовах. Гібридні підходи дозволяють створювати універсальні вимірювальні пристрої, які можуть застосовуватися на різних верстатах і під різні технологічні завдання.

Ще однією тенденцією є зростання ролі мікропроцесорних систем у вимірювальних пристроях. Сучасні контролери дозволяють обробляти великі обсяги даних у реальному часі, виконувати калібрування та перерахунок одиниць вимірювання, а також забезпечувати збереження налаштувань у енергонезалежній пам'яті. Це дозволяє створювати пристрої, які легко адаптуються до різних конфігурацій верстатів і виконують як функцію вимірювання, так і допоміжну роль у контролі та оптимізації процесу.

Автоматизація вимірювальних систем також включає розширені функції для зручності оператора, такі як «заморожування» показів, установка контрольних точок, індикація напрямку руху, відстані до нульової або контрольної точки, а також інтеграція з дисплеями, планшетами або комп'ютерами для дистанційного моніторингу. Ці функції дозволяють підвищити точність, скоротити час виконання операцій і знизити ризик людських помилок.

Важливим напрямком розвитку є також універсалізація вимірювальних пристроїв. Сучасні системи проектуються так, щоб їх можна було легко адаптувати до різних типів верстатів, замінювати датчики, підключати додаткові модулі та змінювати одиниці вимірювання залежно від специфіки обробки. Такий підхід забезпечує економічну ефективність модернізації обладнання та дозволяє використовувати один універсальний пристрій для різних завдань.

Отже, тенденції розвитку вимірювальних систем у промисловості спрямовані на підвищення точності, надійності та гнучкості технологічних процесів за рахунок інтеграції безконтактних методів, цифрових технологій та мікропроцесорного контролю. Автоматизація і універсалізація таких систем дозволяє модернізувати існуюче обладнання, підвищити ефективність виробництва та забезпечити стабільну якість продукції.

1.9 Системи калібрування та налаштування точності вимірювальних пристроїв

Точність вимірювальних систем у металорізальних верстатах

безпосередньо залежить від правильної калібровки та налаштування пристроїв. Навіть найсучасніші оптичні, індуктивні або мікропроцесорні системи не можуть забезпечити заявлену точність без регулярної перевірки та корекції показів. Калібрування дозволяє врахувати вплив зносу механічних елементів, похибок монтажу, зміни температури та інших факторів, які можуть впливати на результати вимірювань.

Процес калібрування включає визначення нульової точки, перевірку точності вимірювальних лінійок або енкoderів та при необхідності внесення коригувань у контролер. Наприклад, у цифрових системах індикації DRO або мікропроцесорних пристроях оператор може встановлювати контрольні точки, вводити поправки для врахування механічних редукторів або змінювати коефіцієнт перерахунку одиниць вимірювання. Це дозволяє забезпечити правильну інтерпретацію сигналів від датчиків і точність позиціонування у межах заданих допусків.

Особливе значення мають автоматизовані системи калібрування, які дозволяють виконувати процедуру безпосередньо через програмне забезпечення контролера. Такі системи можуть зберігати профілі налаштувань для різних верстатів або типів обробки, виконувати періодичне тестування точності та самостійно коригувати покази у разі виявлення відхилень. Автоматизація калібрування знижує залежність від кваліфікації оператора, прискорює процес налаштування і забезпечує стабільну якість роботи.

Калібрування також включає перевірку повторюваності показів, тобто здатності системи давати однакові результати при повторному переміщенні в ту ж саму точку. Для цього використовуються контрольні еталони, калібровані лінійки або спеціальні мікрометри, що дозволяють визначити точність вимірювань і внести необхідні поправки. Такий підхід забезпечує надійність і передбачуваність роботи пристрою у різних умовах експлуатації.

Системи налаштування точності зазвичай включають можливість регулювання нуля та масштабування одиниць вимірювання. Наприклад, при зміні редуктора або застосуванні іншого типу датчика оператор може під

коригувати коефіцієнт перерахунку, щоб покази дисплея відповідали реальному переміщенню деталі або обертів вала. Це особливо важливо для універсальних пристроїв, які використовуються на різних верстатах і для різних технологічних операцій.

Крім того, сучасні мікропроцесорні пристрої дозволяють зберігати налаштування у енергонезалежній пам'яті, що забезпечує відновлення точності після вимкнення живлення або переміщення пристрою на інший верстат. Це робить такі системи зручними для модернізації обладнання та повторного використання у різних умовах виробництва.

Таким чином, системи калібрування та налаштування точності є критично важливими для забезпечення високої точності та надійності роботи вимірювальних пристроїв. Вони дозволяють адаптувати прилади до специфіки конкретного верстата, компенсувати похибки механічних та електронних компонентів, забезпечити повторюваність і стабільність показів, що є необхідною умовою для модернізації верстатів і впровадження мікропроцесорних систем вимірювання.

1.10 Порівняльний аналіз сучасних систем вимірювання

Після розгляду основних типів вимірювальних систем у металорізальних верстатах стає очевидним, що кожен метод має свої переваги та обмеження. Оптичні системи забезпечують високу точність і можливість інтеграції з цифровою обробкою даних, але чутливі до забруднення та вібрацій. Індуктивні та ємнісні датчики надійні, прості в інтеграції та стійкі до механічного зносу, але мають обмеження по робочому діапазону та специфіку матеріалів, що вимірюються.

Порівняльний аналіз показує, що для реалізації універсального прецизійного пристрою оптимальним є комбінований підхід. Оптичні енкодери забезпечують високу базову точність, індуктивні або ємнісні датчики можуть використовуватися для резервного контролю або у випадках, коли умови експлуатації перешкоджають роботі оптики. Мікропроцесорна обробка сигналів

дозволяє реалізувати додаткові функції, такі як збереження контрольних точок, калібрування під різні одиниці вимірювання та «заморожування» показів. Такий підхід дозволяє поєднувати переваги різних методів, мінімізуючи їхні обмеження і забезпечуючи гнучкість використання пристрою на різних верстатах.

Вибір методів для мікропроцесорного пристрою ґрунтується на таких критеріях: точність вимірювань, повторюваність результатів, простота інтеграції з контролером, стабільність показів у різних умовах експлуатації та можливість калібрування. Оптичні енкодери, обрані як основний датчик, дозволяють досягти високої роздільної здатності та легко передавати дані на мікропроцесор. При необхідності використання інших типів датчиків, їх сигнали також можуть бути оброблені контролером, що забезпечує універсальність і адаптивність системи.

Ключовим фактором при проектуванні пристрою є комбінування точності, зручності та надійності. Оптична система забезпечує прецизійність, мікропроцесорна обробка — автоматизацію та облік різних умов експлуатації, а резервні датчики дозволяють підтримувати стабільність показів у складних виробничих середовищах. Такий підхід забезпечує не лише високоточне вимірювання та позиціонування, а й зручність використання, що особливо важливо при модернізації старих верстатів та впровадженні універсальних мікропроцесорних пристроїв.

Враховуючи зазначені фактори, обрана комбінація методів для розробки пристрою забезпечує оптимальний баланс між точністю, надійністю та універсальністю. Це дозволяє створити мікропроцесорну систему вимірювання та позиціонування, яка може бути адаптована для різних верстатів, типів обробки та умов експлуатації, забезпечуючи високу ефективність і практичну цінність у промисловому застосуванні.

2 ДОСЛІДЖЕННЯ МЕТОДІВ ВИМІРЮВАННЯ ТА ПІДБІР КОМПОНЕНТІВ

2.1 Методика дослідження та критерії ефективності

Основна задача будь-якої вимірювальної системи перетворити фізичний рух або стан у число, яке можна стабільно відтворити. Щоб оцінити, наскільки добре це відбувається, потрібно дивитися не на окремі деталі, а на весь ланцюг:

- як формується сигнал на вході;
- як він обробляється;
- як масштабується в потрібні одиниці;
- як результат показується користувачу.

Такий підхід дозволяє порівнювати різні методи між собою не за марками компонентів, а за тим, як вони впливають на точність, швидкодію і надійність у реальних умовах.

Методика спирається на два види роботи — аналітичний і експериментальний. В аналітичній частині робимо припущення про джерела похибок і затримок, описуємо їх простими математичними моделями та оцінюємо, як ці фактори проходять крізь перетворення від сигналу до індикатора. Зручно розкласти похибку на складові: дискретизація (мінімальний крок відліку), механічні ефекти (люфт, нерівномірність руху), електричні завади (помилкові спрацювання, пропуски подій). Усі ці внески розглядаються як сума систематичних і випадкових складових (рис 2.1).

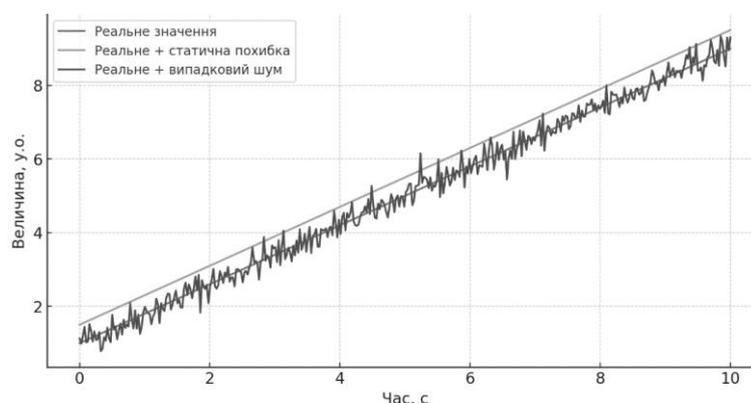


Рисунок 2.1 — Вигляд статичної і випадкової похибки на графіку

Значення сталої похибки можна обчислити за формулою:

$$\bar{\varepsilon} = \overline{\hat{X}} - X = \frac{1}{m} \sum_{i=1}^m (\hat{X}_i - X_i),$$

де X_i — показ приладу;

X^{\wedge} — еталон (істинне значення);

m — кількість пар.

Значення випадкової похибки у свою чергу можна розрахувати скориставшись іншою формулою:

$$s = \sqrt{\frac{1}{m-1} \sum_{i=1}^m (\hat{X}_i - \overline{\hat{X}})^2},$$

де X_i — показ приладу;

X^{\wedge} — еталон (істинне значення);

m — кількість пар.

Щоб порівнювати методи об'єктивно, потрібно домовитися про метрики. Основні три — це точність (наскільки близько показ до еталона), повторюваність (наскільки стабільний результат при багаторазовому підході до однієї точки) та латентність (скільки часу минає від події на вході до оновлення показу). Додатково оцінюємо стійкість до завад і надійність збереження налаштувань (чи не втрачаються параметри після вимкнення живлення, чи не пошкоджуються записи). Усі методи тестуємо одним набором сценаріїв і в однакових умовах, щоб різниця в результатах йшла від самих методів, а не від різних процедур.

Важливо розділяти вимірювальне ядро і показ результату. Ядро відповідає за формування відліку зі сигналу, перевірку послідовності подій, обробку нестандартних ситуацій (розриви, переповнення, реверс). Показ результату — це оновлення екрана і форматування чисел. Якщо їх не розділити, важкі операції

інтерфейсу можуть заважати відстеженню подій і давати пропуски. Тому аналізуємо часові бюджети окремо: скільки займає обробка події, скільки — оновлення відображення, чи не накладаються ці інтервали так, що втрачаються імпульси.

Калібрування розглядаємо як частину методики, а не як разову дію. Нульова прив'язка визначає відлік у зручній опорній точці. Масштаб краще отримувати за двома еталонними положеннями і перевіряти на третьому — так помітно систематичний зсув. Після калібрування оцінюємо залишкову похибку на характерних режимах: повільні підходи, швидкі переміщення, зупинки з реверсом. Параметри зберігаємо безпечно: записуємо лише при зміні та перевіряємо валідність, щоб уникнути корупції даних.

Узагальнюючи, методика будується як послідовність з'єднаних кроків: спочатку описуємо ланцюг перетворення, далі задаємо прості моделі похибок і часу, потім перевіряємо їх на стенді єдиним набором сценаріїв, і в кінці порівнюємо методи за узгодженими метриками. Перевагу отримує рішення, яке тримає прийнятний баланс між точністю, повторюваністю та латентністю, показує стійкість до типових завод і зберігає передбачувану поведінку протягом довгого циклу роботи. Такий підхід достатньо абстрактний, щоб не «прив'язуватись» до конкретних деталей, і водночас достатньо конкретний, щоб напряду підготувати матеріал для наступних розділів із моделями, алгоритмами та вибором компонентів.

2.2 Математична модель дискретного підрахунку подій

Дискретний підрахунок подій — це універсальний спосіб перетворити рух у послідовність кроків, придатних для цифрової обробки. Джерелом кроків може бути оптичний енкодер, магнітна або індуктивна лінійка, ємнісний датчик, інкрементальний диск з оптопарою. Незалежно від фізичного принципу усі ці перетворювачі видають події, які можна інтерпретувати як переходи між станами або як рахунок імпульсів уперед та назад. Саме ця абстракція дозволяє будувати єдину логіку для різних типів сенсорів.

Класичний приклад — квадратурний сигнал з двох каналів, зсунених у часі (рис. 2.2). Стан такої пари завжди належить до невеликої множини комбінацій, та правильний рух утворює цикл переходів, де кожен новий стан є сусідом попереднього. Якщо напрям змінюється, цикл проходиться у зворотному порядку. Модель перевіряє лише «сусідність» і заздалегідь відкидає нестандартні стрибки, наприклад перескок через один стан або миготіння на місці. Це просте правило виявляється дуже надійним у цехових умовах, де можливі вібрації та імпульсні завади.

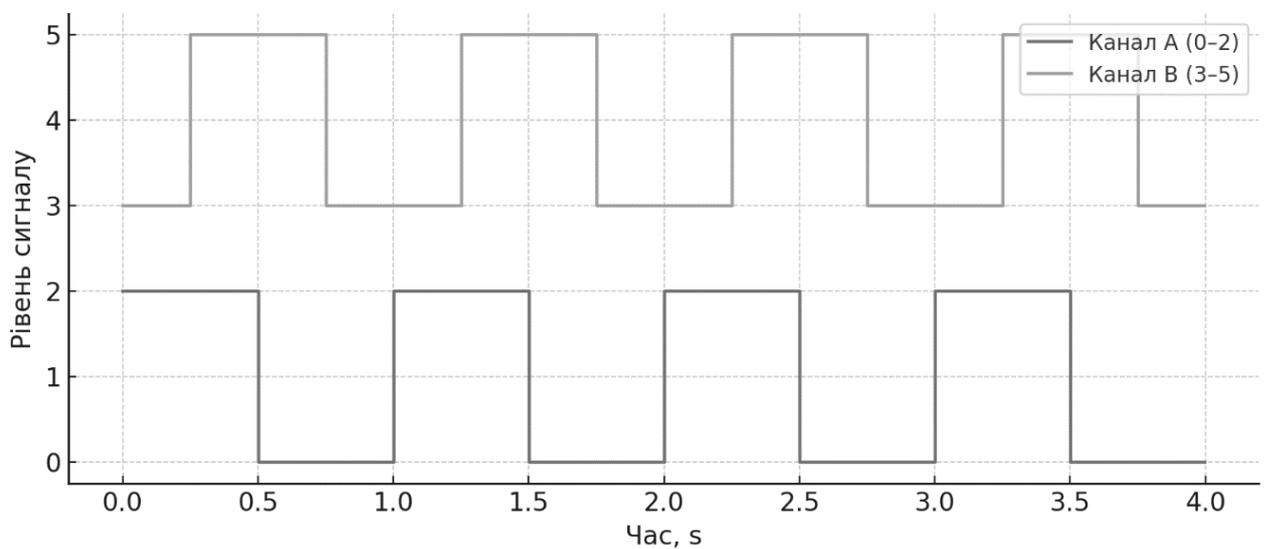


Рисунок 2.2 — Графік квадратурного сигналу і зміщенням фази

Лічильник подій повинен зростати або зменшуватись на одиницю щоразу, коли фіксується коректний перехід. Окремо можна підтримувати останній напрям руху для індикації або додаткової фільтрації коротких реверсів. Захист від нелегітимних стрибків реалізується табличною валідацією: перехід вважається чинним лише тоді, коли новий стан є сусіднім до попереднього; усі інші зміни рівнів інтерпретуються як дребезг чи помилка та не впливають на рахунок. Так само поведуться й альтернативні сенсори: індуктивні й ємнісні лінійки дають послідовність подій, яку можна перевіряти за аналогічною логікою, навіть якщо фізична реалізація відрізняється.

Отриманий відлік можна інтерпретувати як універсальну координату. Для

кутових задач зручно трактувати його як кут на колі, тобто інтерпретувати позицію в межах одного оберту й акуратно працювати з переходом через нуль, щоб на індикаторі не виникали різкі стрибки. Для перерахунку значення лічильника у градуси можна скористатися формулою:

$$\theta_{\text{deg}} = 360 \cdot \frac{N}{C},$$

де N — значення лічильника подій;

C — кількість подій на оберт.

Для лінійних задач той самий відлік масштабується у довжину за обраними одиницями з використанням коефіцієнта, отриманого з геометрії механізму або калібруванням, завдяки такому підходу одна й та сама серцевина виміральної логіки однаково добре працює та на обертовому валу, і на лінійній направляючій. Для обчислення лінійних переміщень можна використовувати формулу:

$$x = k \cdot N,$$

де N — значення лічильника подій;

k — вага одного кроку.

Будь-який дискретний рахунок має квантування, тобто мінімальний помітний крок. Цей крок визначається механічною будовою сенсора та режимом зчитування і встановлює нижню межу деталізації показів. На практиці це означає, що індикатор не повинен намагатися показати дрібнішу «дробову» частину, ніж реально допускає сенсор, інакше цифри будуть змінюватися випадково та вводити в оману. Також діє обмеження розрядності: внутрішній лічильник має зберігатися у достатньому діапазоні, щоб не відбувалося переповнення під час довгої роботи; якщо відлік все ж таки обмежений, варто

чітко визначити політику насичення або обгортання і уникати двозначності під час обчислення різниць.

Особлива увага приділяється надійності при реальних перешкодах. У довгих лініях зручніше працювати з диференційними приймачами та екрануванням, але навіть тоді похибки фронтів трапляються. Модель дискретного підрахунку має вміти ігнорувати одиночні збиті переходи, не накопичуючи похибку, та швидко відновлювати коректне положення при поверненні в легітимний стан. Для цього застосовують порогові умови на частоту подій, обмеження на тривалі операції відображення і чітке розділення вимірювального ядра від користувачького інтерфейсу, щоб довгі графічні операції не заважали реєстрації подій.

У підсумку дискретний підрахунок подій дає прозору й переносну модель для багатьох типів датчиків. Вона спирається на кінцеву кількість станів, суворі правила переходів, чисельний лічильник і продуману інтерпретацію кутових та лінійних координат, а також враховує реальні обмеження: квантування, розрядність, завади та затримки. Саме така рамка потрібна, щоб у наступних розділах пов'язати вимоги до точності з параметрами сенсорів із паспортів, задати правила калібрування і обґрунтувати вибір компонентів для прототипу.

2.3 Перетворення обертального руху в кутові й лінійні координати

Дискретний лічильник подій дає універсальну сиру координату, яку далі потрібно подати в зручних одиницях — кутових або лінійних. Спочатку визначають масштаб: для кутової інтерпретації доречно працювати або з розгорнутим кутом, що накопичується з кожним обертом, або з нормованим відображенням у межах одного кола. Обидва режими корисні з різних причин: розгорнутий кут зручний під час багато оберткових переміщень і аналізу траєкторій, тоді як нормований кут прибирає стрибки на індикаторі й полегшує точне встановлення положення в межах $0...360^\circ$. Для лінійної інтерпретації застосовують коефіцієнт масштабу, що пов'язує один крок лічильника з довжиною в вибраних одиницях; цей коефіцієнт можна отримати аналітично з

геометрії механізму або через двоточкову процедуру калібрування. Важливо, щоб числовий формат індикації не створював ілюзії надмірної точності: кількість десяткових розрядів має відповідати реальній роздільній здатності вимірювання.

На практиці між датчиком та рухом виконавчої ланки часто стоять передавальні механізми. Редуктори змінюють співвідношення обертів, черв'ячні пари задають великі передатні числа з зручним самогальмуванням, а гвинтові пари трансформують кут у поступальний хід каретки. Усі ці варіанти зводяться до введення узгоджувального коефіцієнта, який враховує, кількість обертів первинного валу відповідає одному оберту або лінійному кроку робочого вузла. Доцільно фіксувати місце застосування цієї поправки — у профілі налаштувань, у прошивці за замовчуванням чи у сервісному меню — аби уникнути подвійного масштабування. Для гвинтових пар додатково перевіряють полярність: правий та лівий гвинт дають протилежний напрям руху при однаковому напрямі обертання датчика, тому знак зростання відліку повинен відповідати реальному напрямку подачі.

Якщо використовується лінійна шкала, відлік уже безпосередньо відповідає переміщенню вздовж осі. Тут головне — узгодити нуль і напрям так, щоб збільшення чисел інтуїтивно відповідало руху вперед, а збережені «мітки» давали зрозумілу відстань та сторону до цільової точки. початкові позначки на самій шкалі можна використати для періодичної перевірки консистентної: при проходженні мітки система порівнює очікуване значення з фактичним і, за дозволом користувача, виконує м'яку корекцію накопиченої похибки. Для лінійних трактів доцільно коротко описати процедуру калібрування: встановлення двох опорних позицій, автоматичний розрахунок масштабу, перевірка на третій позиції та підтвердження збереження параметрів.

На довгих траєкторіях важливо, щоб внутрішній лічильник не втрачав однозначності. Тому його зберігають у «довгому» цілому типі з великим запасом, а для індикації застосовують зручне представлення. У кутовому режимі це зазвичай модуль по колу: користувач бачить значення в межах одного оберту, тоді як усередині продовжує накопичуватися повний відлік; такий поділ

дозволяє безпомилково обчислювати різниці після тисяч обертів. У лінійному режимі показ завжди наростаючий, однак інтерфейс може відображати «вікно видимості» навколо нуля чи робочої мітки, щоб не перевантажувати екран великими числами. Обробка знаку має бути однаковою для усіх підсистем: від підрахунку подій до відображення стрілок та текстових підказок, щоб рух «до мітки» завжди інтерпретувався правильно. Особливої уваги потребує різниця кутів біля переходу через нуль; надійний підхід — завжди обирати найкоротший шлях по колу, уникнувши штучних стрибків, які ускладнюють підведення інструмента. Таким чином, перетворення обертального руху у кут чи довжину спирається на узгоджений масштаб, коректне врахування передавальних ланок та продуману стратегію індикації, що задовольняє вимоги до однозначності, передбачуваності і зручності роботи оператора. Це створює цілісну рамку, у якій одна й сама дискретна «серцевина» надійно працює з різними типами механік та датчиків, а також легко переноситься між кутовими та лінійними задачами без втрати логіки обчислень.

2.4 Обґрунтування вибору електронних компонентів та модулів

2.4.1 Вибір платформи

Вибір апаратних засобів підпорядковано вимогам до обраних методів вимірювання та практичним вимогам для користувача: вимірювальний тракт має стабільно рахувати події з квадратурного перетворювача без втрат; індикація повинна давати контрастний, читабельний результат у різних умовах; блок живлення залишатися прохолодним і надійним у компактному закритому корпусі. Додатково враховано наявність компонентів на ринку, сумісність рівнів сигналів і простоту монтажу та обслуговування. Нижче узагальнено аргументи, чому обрано саме цю конфігурацію.

У нашому випадку вибір платформи починається не з переліку технічних параметрів, а з відповідності реальним умовам: обмежений простір у корпусі, потреба у детермінованій обробці переривань та мінімальному тепловиділенні. У такому контексті сімейство Arduino на AVR виглядає практичним

компромiсом: воно забезпечує передбачувану часову поведінку переривань на 16 МГц, просту обв'язку, 5-вольтову логіку з добрим запасом за рівнями, велику базу бібліотек і документовані вимоги по живленню та заземленню. Саме тому в роботі обрано Arduino Pro Mini(рис. 2.3): компактна плата без зайвих периферійних доповнень, що дає мінімальну апаратну латентність і не гріє корпус через зайву перетворювальну електроніку.

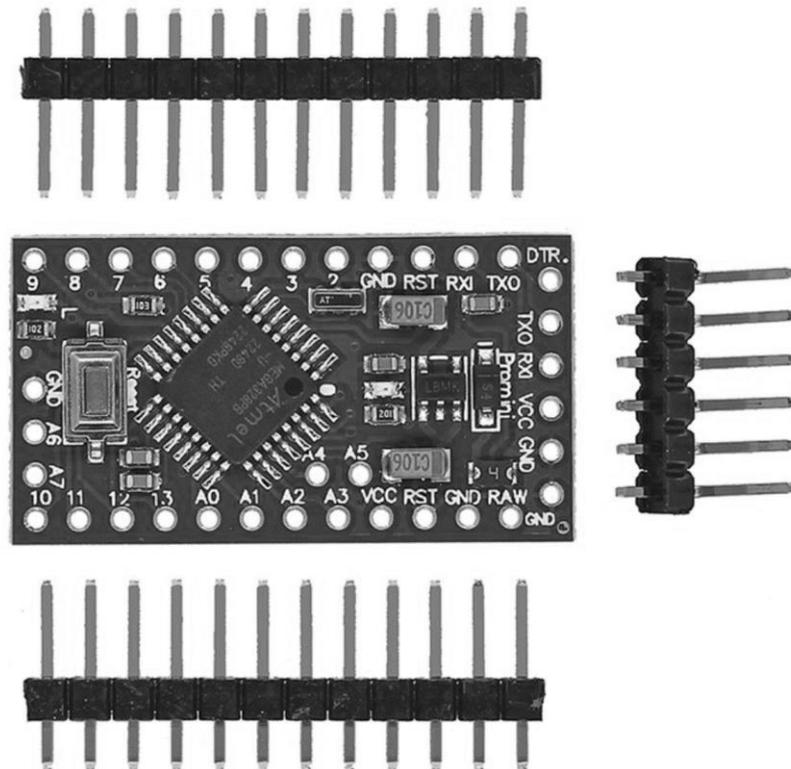


Рисунок 2.3 — Плата Arduino Pro Mini

Як альтернатива у межах тієї ж платформи розглядаються Arduino Nano та Uno. Вони технологічно близькі (той самий ATmega328P у 5 В/16 МГц), але мають більші габарити та додаткові вузли (USB-UART, стабілізатори), які полегшують відлагодження, проте збільшують теплове та електромагнітне навантаження всередині невеликого корпусу. Arduino Mega дає суттєвий запас пінів і таймерів, однак його форм-фактор надмірний для приладу індикації біля робочої зони. Плати на ATmega32U4 (Leonardo/Pro Micro) зручні в USB-HID задачах, але це не дає переваг для квадратурного підрахунку; навпаки, 32U4 має

інші нюанси з таймерами і живлення, що тут не додають користі. Таким чином, всередині «класичного» Arduino-ряду є найкращими кандидатами, де Pro Mini виграє саме компактністю і базовою апаратною конфігурацією.

Поза AVR варто згадати продуктивні 32-бітні варіанти. ESP32 (рис. 2.4) та ESP8266 (рис. 2.5) приваблюють інтегрованим Wi-Fi/BLE та високою частотою, але у нашій задачі бездротові інтерфейси не є обов'язковими, а наявність радіотракту лише ускладнює EMC; додатково, 3,3 В логіка вимагатиме ширшого узгодження рівнів з периферією. RP2040 (Raspberry Pi Pico) має сильну сторону — PIO-блоки, які можуть апаратно обробляти квадратуру з мінімальним навантаженням на ядра; проте це додає етап налаштування PIO та переходу на 3,3 В, а виграш у продуктивності не є критичним для ISR-навантаження. STM32 (лінійка Nucleo/Blue Pill) пропонує апаратні лічильники з режимом «encoder interface» і багаті таймери — сильна перевага для високошвидкісних систем. Однак знову постає питання 3,3 В логіки, більший вхідний бар'єр по інструментах і складніша плата розв'язки; для індикатора на одному енкодері це того не вартує.

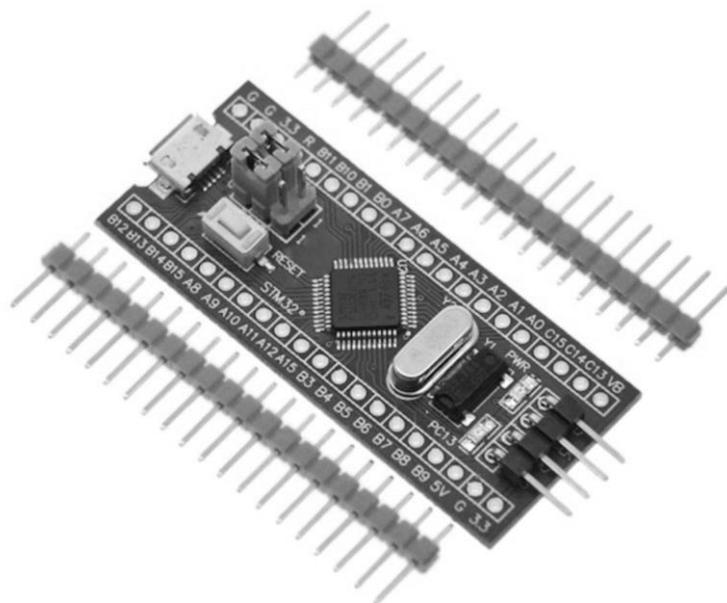


Рисунок 2.4 — Плата STM32F103C8T6

Ще один варіант — Teensy 3.x/4.x, він забезпечує величезний запас по швидкодії, прецизійні таймери та добрі бібліотеки, але ціною зростання енергоспоживання, вимог до живлення і вартістю. У задачі з помірною частотою подій і локальним відображенням цей запас не буде конвертовано у відчутне підвищення якості вимірювання: вузьким місцем швидше стане механіка та фронти сигналу на вході, а не обчислювальні ресурси ядра.

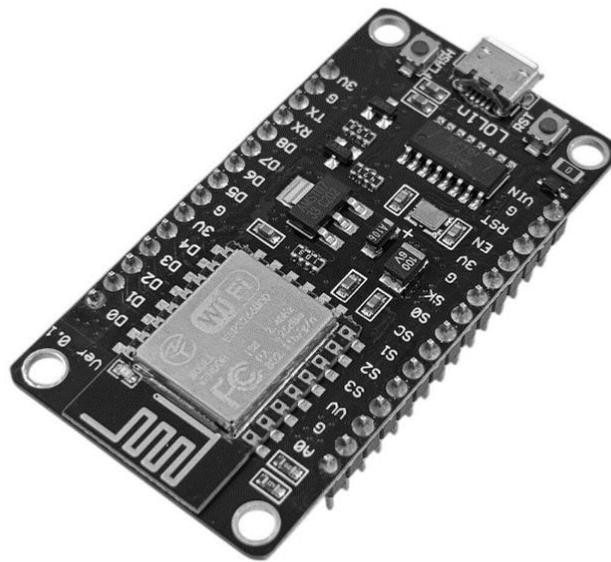


Рисунок 2.5 — Плата ESP8266 Node Mcu V3 Lua Wi-Fi

Підсумовуючи, Arduino Pro Mini обрано як платформу, що задовольняє ключовим критеріям: 5 В логіка з добрим запасом по шумам, достатня кількість переривань і таймерів для надійної квадратури, детермінований ISR-профіль на 16 МГц, мінімальна апаратна «надлишковість» у компактному корпусі, стабільна екосистема бібліотек для дисплея та інших модулів. Ближчі альтернативи всередині Arduino (Nano/Uno) програють габаритами й тепловим бюджетом, тоді як потужніші 32-бітні платформи додають непотрібної складності по живленню, ЕМС та інструментах без практичного виграшу в точності та надійності саме для цього класу вимірювальних пристроїв.

2.4.2 Вибір індикатора

Вузол індикації вимагав поєднання високої читабельності, компактності та низького енергоспоживання за обмеженого теплового відводу корпусу. За цими критеріями матриця OLED на контролері SSD1306 (рис. 2.6) виявилася оптимальною. Висока статична контрастність і широкий кут огляду забезпечують стабільне зчитування показів на відстані витягнутої руки й під довільним кутом — те, чого бракує традиційним символьним LCD із TN-матрицями. Відсутність підсвічування знімає теплове навантаження з вузла індикації, зменшує кількість провідників і джерел завад, а також спрощує розміщення дисплея в передній панелі.



Рисунок 2.6 — OLED дисплей SSD1306

Для приладу, який має показувати велике число з дробовою частиною, статусні мітки та короткі службові підписи, графічна матриця 128×64 забезпечує достатню деталізацію під великі шрифти й прості шкали. Контролер SSD1306

апаратно підтримує розбиття екрану на сторінки, що дозволяє організувати часткове оновлення без мерехтіння і з мінімальним навантаженням на вимірювальний цикл. Завдяки невисокому струму споживання і стабільній роботі при зниженій яскравості зменшується ризик локального перегріву й збільшується ресурс панелі — важливо для закритого пластикового корпусу з обмеженою вентиляцією.

Порівняно з альтернативами вибір виглядає прагматично. Символьні LCD (рис. 2.7) економніші за ціною, але мають обмежений кут огляду, нижчий контраст та жорстку сітку символів, що ускладнює відображення великих «цифрових» чисел і десяткової крапки на потрібних позиціях. Кольорові TFT (рис. 2.8) надають гнучке графічне оформлення, проте вимагають суттєвіших струмів, мають складніший електромагнітний профіль і часто потребують ширших шин даних — усе це підвищує ризики для ЕМС поруч із силовим устаткуванням. Електронне чорнило добре читається при зовнішньому світлі й практично не споживає енергії у статичному режимі, однак має повільне оновлення та помітні артефакти при оновленні кадру, що не припустимо для динамічної індикації відстані до мітки та швидкого зворотного зв'язку.



Рисунок 2.7 — Символьний lcd1602

З огляду на завадостійкість і розводку, графічний OLED не вимагає широких паралельних шин, мінімізуючи кількість активних ліній на передню панель і зменшуючи інжекцію перешкод у спільну землю. Невеликі токи й відсутність підсвітки сприяють кращій електромагнітній сумісності з трактом приймання квадратури. Додатковим аргументом є зрілість бібліотек для шрифтів для цього контролера й стабільна доступність сумісних модулів на ринку з однаковою механікою кріплення. Практичні ризики — деградація яскравості при тривалому статичному зображенні та температурні обмеження — компенсуються програмними прийомами (невеликі зсуви, періодичне оновлення розташування маркерів, контроль яскравості) та вибором панелей із заявленим промисловим діапазоном. У підсумку модуль SSD1306 забезпечує необхідний баланс: високу читабельність і швидкість оновлення при помірному споживанні, простій електричній обв'язці та передбачуваній поведінці.



Рисунок 2.8 — Графічний TFT дисплей

2.4.3 Вибір елементів керування

Вибір тактових кнопок (рис. 2.9) обґрунтований поєднанням передбачуваного «механічного» кліку, простоти інтеграції та прозорості логіки натискання—відпускання. На відміну від ємнісних сенсорів на кшталт TTP223 (рис. 2.10), де спрацювання залежить від чутливості, калібрування й умов торкання (площа пальця, наявність накладки), тактовик дає чіткий бінарний сигнал у момент фізичного замикання контактів. Це спрощує налаштування порогів і виключає «напівнатискання». Крім того, тактові кнопки легко масштабуються за кількістю: три окремі дії реалізуються трьома незалежними входами без потреби у матричному опитуванні або налаштуванні порогів для кожного каналу.

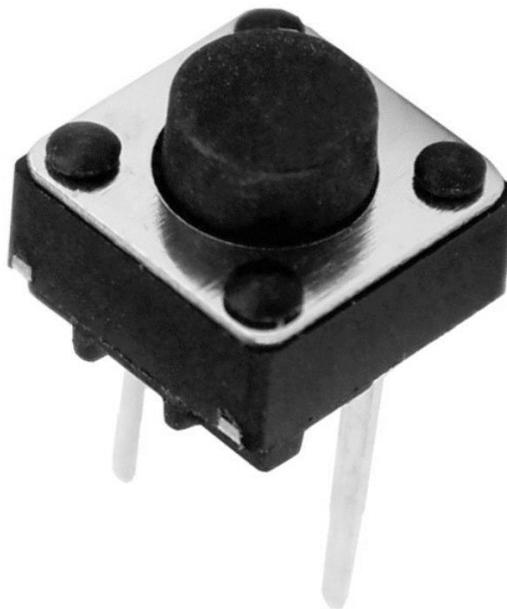


Рисунок 2.9 — Тактова кнопка

Якщо порівнювати з сенсорними панелями чи невеликими сенсорними екранами, тактові кнопки виграють апаратною простотою: не потрібні контролери сенсорної матриці, калібрування координат, графічні бібліотеки для віртуальних кнопок. Фізичний зворотний зв'язок — відчутний хід і клацання — природно сигналізує користувачу про спрацювання, тоді як на сенсорі

підтвердження доводиться відтворювати візуально або звуком. Мембранні клавіатури забезпечують низьку висоту профілю й готові «плівкові» інтерфейси, але поступаються за тактильністю й ремонтпридатністю: заміна окремої кнопки неможлива, доводиться міняти всю клавіатуру.

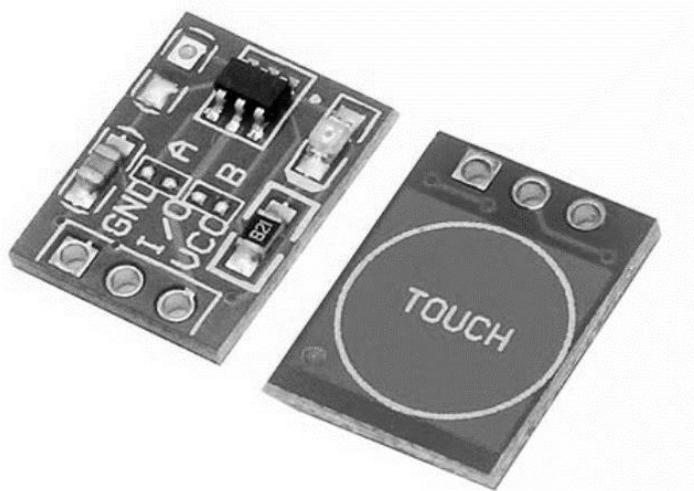


Рисунок 2.10 — Сенсорна ємкісна кнопка

Порівняно з поворотним енкадером (рис 2.11) як універсальним органом керування, тактові кнопки краще підходять для дискретних команд — «скинути», «пауза/продовжити», «підтвердити» — коли не потрібно обертання або швидкого прокручування меню. Енкадер зручно використовувати для безперервного налаштування (параметри, скрол), але він додає рухому механіку, потребує осьового розміщення і створює іншу модель взаємодії: випадковий дотик може зсунути значення на кілька кроків. У нашому випадку простий і передбачуваний «клік» краще відповідає семантиці команд і зменшує ймовірність помилкових дій.

У плані ергономіки тактові кнопки легко диференціювати формою і тактильними ковпачками: важливі функції отримують різний розмір або фактуру, щоб зменшити ймовірність переплутати дії. Це складніше реалізувати

на ємнісних сенсорах, де усі «зони торкання» відчуються однаково, і на енкодерах, де кілька функцій часто поєднують в одній ручці (обертання + натискання). Окремо варто відзначити, що для простого меню з фіксованою кількістю пунктів мінімальний комплект із трьох тактовиків (лівий/правий або «—/+» і «ОК») забезпечує швидке навчання користувача та однозначну карту жестів без додаткових підказок.

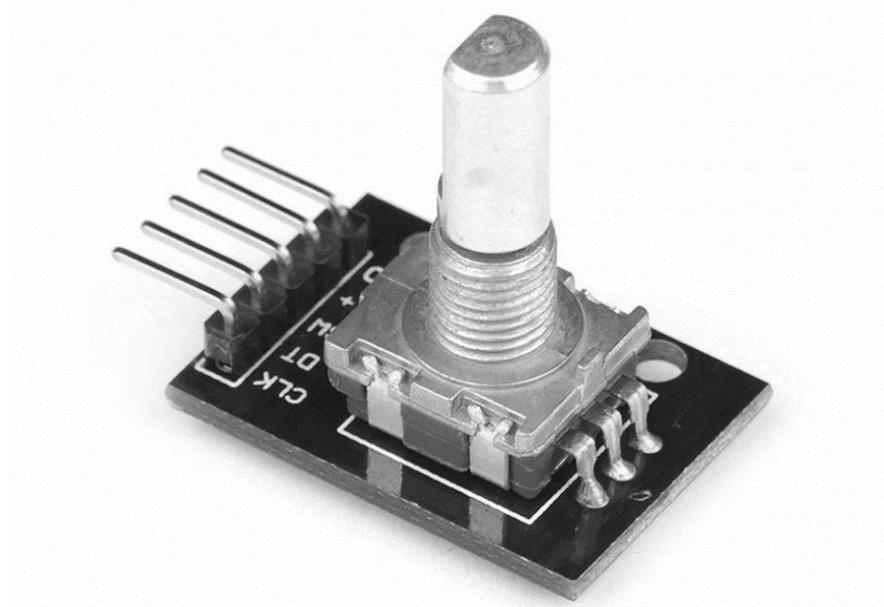


Рисунок 2.11 — Модуль енкодера

З погляду інженерної реалізації також спрощується схема: тактові кнопки безпосередньо під'єднуються до входів мікроконтролера з внутрішніми підтяжками, антидребезг реалізується простим RC і програмною фільтрацією, а логіка обробки подій — у вигляді «натискання/утримання/повтору» — не вимагає окремих драйверів чи складних станів. На відміну від TTP223, де для кожного каналу бажано забезпечити стабільну ємність оточення і коректний шар покриття, тактовики менш чутливі до конструктивних дрібниць та легше переносять варіації товщини панелі. У підсумку, у задачі з невеликою кількістю чітко визначених дій та вимогою до надійного, відчутного підтвердження — фізичні тактові кнопки є найпростішим і найпередбачуванішим вибором порівняно з ємнісними клавішами, енкодерами та сенсорними інтерфейсами.

2.4.4 Вибір компонентів для вимірювального тракту

В основі вимірювального тракту обрано оптичний інкрементальний енкодер (рис 2.12) із квадратурними каналами А/В, оскільки саме цей тип дає передбачувану щільність подій на оберт і однозначне визначення напрямку без складних перетворень. Квадратура природно узгоджується з дискретним лічильником: кожен коректний перехід — це крок, а індекс — канал Z (за наявності) дає можливість для швидкого відновлення нульової позиції в межах одного оберту. На відміну від потенціометрів (рис. 2.13), де точність обмежується зносом доріжки й нелінійністю, оптичний енкодер працює безконтактно й забезпечує відтворюваність кроку на всьому ресурсі. Порівняно з абсолютними енкодерами оптичний інкрементальний спрощує електричну частину і зменшує вартість, залишаючи можливість задати потрібну «ефективну роздільну здатність» через вибір ліній/оберт та режим підрахунку ($\times 1/\times 2/\times 4$) у поєднанні з передавальними механізмами.



Рисунок 2.12 — Інкрементний оптичний енкодер

Як альтернатива розглядалися магнітні та індуктивні рішення. Магнітні інкрементальні датчики мають добру стійкість до відхилень зазору і можуть працювати з простішою механікою, проте їхня миттєва фазова стабільність і

рівномірність кроку зазвичай гірші за оптичні, що важливо для повторюваного позиціонування на малих переміщеннях. Індуктивні/резольверні системи забезпечують високий ресурс та плавність, але вимагають синусно-косинусної обробки та подальшої цифро-аналогової обв'язки, що у нашому класі приладу є надмірним. Лінійні шкали зчитують переміщення безпосередньо, однак потребують іншої механічної інтеграції та дорожчі при однакових вимогах до кроку; до того ж валовий енкадер легко вбудовується у вже наявний кінематичний ланцюг (вал, редуктор, гвинт), зберігаючи універсальність пристрою.



Рисунок 2.13 — Потенціометр

Механічно вибір зроблено на користь валового виконання з еластичною компенсувальною муфтою, що зменшує вплив невеликої несоосності і биття на фазову картину. Важливими параметрами добору були допустимі радіальні/осьові навантаження на підшипники енкадера, довжина та діаметр вала, спосіб кріплення фланця. Правильно підібрана муфта із низькою інерцією зменшує накопичення мікропружних деформацій при реверсах та покращує повторюваність нульового проходу. Додатково враховано необхідність простої

заміни: стандартна посадка та типовий розмір корпусу спрощують сервіс без переробки кріплень.

Електрично передбачено використання стандартних рівнів A/B (TTL) з можливістю переходу на лінійний драйвер типу RS-422 для довших з'єднань; це дає гнучкість у виборі кабельної інфраструктури без зміни логіки приймання. Пікова частота подій визначалась виходячи з максимальної швидкості обертання та числа ліній на оберт, що дозволило узгодити роздільну здатність із пропускнуою здатністю лічильника та часовим бюджетом обробника переривань. Підсумково оптичний інкрементальний енкодер забезпечує потрібний баланс: високу повторюваність кроку, прозору інтеграцію у дискретну модель підрахунку, передбачувану механічну компоновку та достатній запас швидкодії для коректної роботи у всьому робочому діапазоні пристрою.

3 РОЗРОБРА КОРПУСУ ТА ЕЛЕКТРИЧНОЇ СХЕМИ

3.1 Формування вимог до корпусу пристрою

Корпус у нашому випадку повинен забезпечувати зручне щоденне користування на верстаті, надійне кріплення та захист внутрішніх вузлів. Першою вимогою є читаність індикації: вікно під дисплей потрібно розташувати так, щоб цифри було добре видно під типовими кутами огляду оператора та на відстані витягнутої руки. Лицьова панель повинна мати легкий нахил або геометрію козирка, щоб зменшити відблиски та зробити числа контрастнішими. Кнопки розміщуються на одній площині, з достатнім кроком між ними, щоб їх можна було натискати у рукавичках і не промахуватись; бажано зробити невелике заглиблення або бортик навколо, аби уникати випадкових натискань під час маніпуляцій біля верстата. Для дисплея варто передбачити захисне прозоре вікно, яке не спотворює картинку та легко миється. Місце для цього вікна — в одній площині з лицьовою панеллю, без гострих кромek.

Друга група вимог стосується універсальності монтажу. Пристрій має ставитись як на горизонтальні, так і на вертикальні поверхні; тому бажано передбачити кілька варіантів кріплення: пази під стандартні, місця під стяжки, а за потреби — посадку під магнітну пластину або перехідну скобу. Кабельні вводи (до енкодера, живлення, додаткового датчика/стрічки) повинні виходити униз або назад під кутом. У місцях входу кабелів слід передбачити елементи розвантаження натягу (фіксатори/вушка), щоб випадкове смикання не передавало сили на роз'єми чи плату. Задня частина корпусу має мати доступний роз'єм для підключення енкодерної стрічки та датчика до неї, із захистом від переплутування та місцем під маркування. В середині — місця для укладання запасу проводу, щоб кришка зачинялась без перетирання.

Третє — обслуговуваність і збирання. Конструкція має розкриватися послідовно: зняття задньої кришки відкриває доступ до плати, дисплея, роз'ємів і кріплень, не знімаючи пристрій із верстата. Рекомендовано стандартизувати кріплення і передбачити внутрішні направляючі під плату та під дисплей, щоб

під час складання компоненти ставали на місце без підгонки. Товщина стінок — помірна орієнтовно 2мм, з радіусами на внутрішніх кутах, щоб деталі впевнено друкувались і не деформувалися; посадкові місця під гвинти — з розширеннями під гайки або тепловставки, аби кріплення витримувало багаторазове відкривання. Передня рамка має бути жорсткою: ребра або локальні підсилення навколо вікна дисплея і зони кнопок зменшують прогин та «гуляння» перемикачів під натиском. Внутрішній простір організовується так, щоб провідники не торкалися рухомих частин (вала енкодера, якщо він інтегрований), а плата не тиснула на кришку.

Четверта частина вимог — захист і довговічність. Лицьова панель повинна тримати бризки та пил: для цього доречно зробити виступаючу рамку навколо прозорого вікна і мінімальні зазори між панеллю та кнопками. Вентиляція пасивна, через непомітні щілини знизу або ззаду; від прямих отворів на лицьовій частині бажано відмовитись, щоб не накопичувався бруд. Усі гострі кромки — скошені або заокруглені, щоб пристрій було приємно брати в руки, а кабелі не перетирались. Зовнішні поверхні — з невеликою фактурою (після 3D-друку це виходить природно), яка приховує дрібні подряпини. Колір — контрастний для цифр на екрані. Якщо енкодер встановлюється всередині корпусу, потрібна точна соосність посадки валу і опора з боку кришки (наприклад, втулка-опора), щоб на датчик не передавалися поперечні навантаження.

Окремо фіксуються розміри й обмеження. Корпус не має перекривати огляд робочої зони верстата або заважати рукояткам; тому закладається компактний фронт (висота під дисплей і кнопки) і невелика глибина. Загальна вага — помірна, щоб кріплення не розхитувалося під вібраціями. Простір усередині планується з запасом під майбутні зміни: можливість заміни плати керування на близьку за габаритом, встановлення іншого дисплея з такою ж видимою областю, додавання другого роз'єму під альтернативний датчик. Вся маркувальна інформація — місця для підписів кнопок, позначення роз'ємів, стрілка напряму обертання — передбачаються на стадії моделі, щоб після друку не доводилось імпровізувати.

Підсумовуючи, корпус повинен: тримати читаність і зручність керування; дозволяти простий і надійний монтаж на різних поверхнях верстата; забезпечувати порядок із кабелями та легкий сервісний доступ; захищати дисплей і кнопки від бризків і пилу; мати запас міцності у місцях натискань і кріплень; та залишати «простір для росту» — на випадок, якщо згодом зміниться датчик або плата. Усі ці вимоги варто закласти безпосередньо в CAD-модель, щоб отримати корпус, який одразу добре збирається й не потребує переробок після першого друку.

3.2 Створення 3D-моделі корпусу у середовищі AutoCAD

AutoCAD — це професійне середовище для 2D-та 3D-проектування з точними інструментами креслення, параметричними обмеженнями, шарами, блоками та повною системою розмірів і приміток, що дозволяє швидко переходити від ескізу до робочої моделі й супровідної документації (рис. 3.1). У ньому зручно будувати корпусні деталі із заданими габаритами, радіусами й допусками, перевіряти стикування елементів, вирівнювання отворів і товщини стінок, а також готувати файли stl для 3D-друку чи фрезерування .

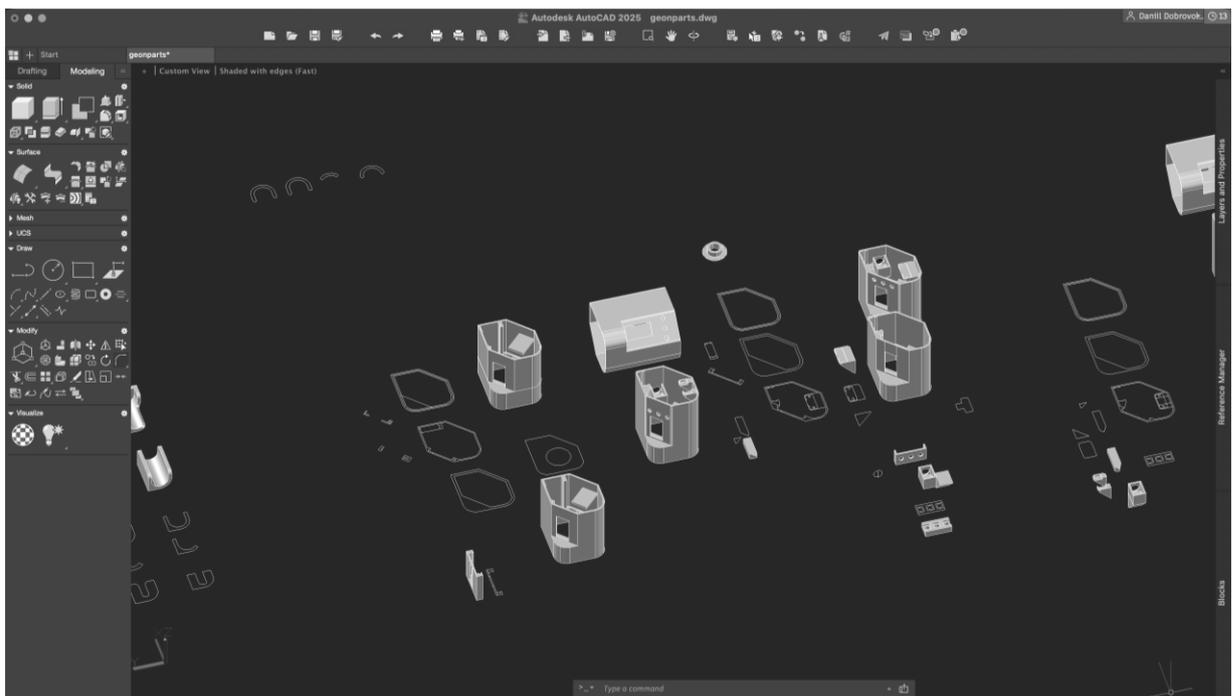


Рисунок 3.1 — Інтерфейс програми AutoCAD

Переваги — висока точність, сумісність зі DWG, велика бібліотека команд і знайомий інтерфейс, що скорочує час на ітерації. Саме тому AutoCAD було обрано для моделювання корпусу: він забезпечує контроль геометрії на рівні десятих міліметра, швидке внесення змін і простий експорт моделей у формати, які напряду сприймає слайсер 3D-принтера та інші CAD/CAM-системи.

Базова ідея полягає у створенні корпусу з нахиленим вікном індикації, групою отворів під кнопки (рис. 3.2) та знімну кришку з центровим отвором під вал. Лицьова панель має невеликий кут нахилу, завдяки чому цифри на екрані читаються зверху й збоку без зайвих відблисків. Саме вікно під дисплей утоплене в рамку, яка одночасно працює як жорстке ребро та козирок. Радіуси на ребрах зменшують концентратори напружень і роблять деталь приємною на дотик. Таке компонування було вибрано після попередніх ескізів у AutoCAD, де перевірялась посадка дисплея, кнопок і простір під плату керування.

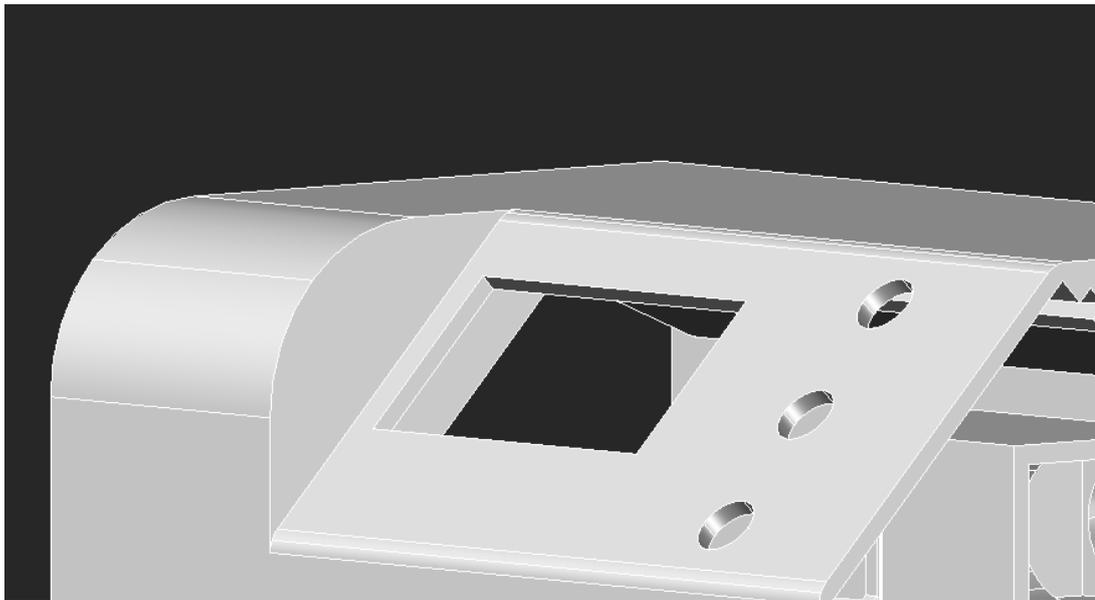


Рисунок 3.2 — Вигляд лицьової панелі корпусу

Внутрішня компоновка вирішує два завдання: фіксація електроніки без підгонки та прямолінійний шлях зборки. У тілі корпусу змодельовані напрямні під плату, роз'єми та кнопки — плата встановлюється в пази й зафіксувати без люфтів. Під дисплей передбачена полиця-опора, яка задає глибину посадки і кут нахилу, щоб його видима область чітко співпадала з вікном. За лицьовою

панеллю зроблено технологічний карман під шлейф і трохи запасу проводу, щоб при закриванні кришки нічого не затискалося. На внутрішній поверхні присутні локальні підсилення — короткі ребра, які прибирають прогин у зоні кнопок та навколо вікна дисплея. Таке рішення просте для друку і водночас суттєво підвищує жорсткість (рис 3.3).

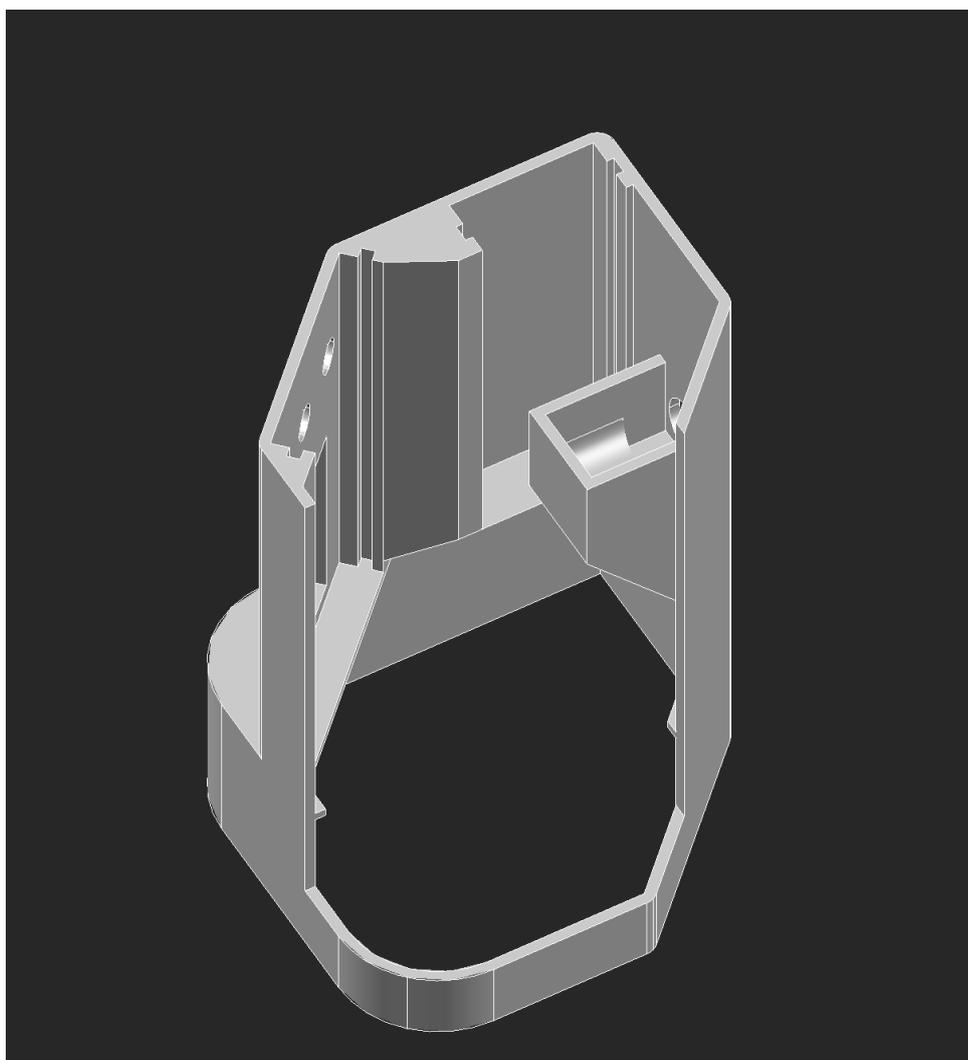


Рисунок 3.3 — Вигляд корпусу в розрізі

Бокова кришка — окрема деталь, що встановлюється у корпус завдяки спеціальним виступам. Це спрощує центрування при складанні і дає акуратний шов. Посередині кришки — круглий отвір: він працює як прохід для вала енкодера. На внутрішньому боці кришки передбачено місце під ущільнювальне кільце або тонку прокладку — це не герметизація в повному розумінні, але такий м'який бар'єр добре відсікає пил і бризки від індикатора та плати. Фіксація

кришки вирішена простими гвинтами в різьбові стійки, розташовані симетрично, — доступ до них відкривається відразу після зняття корпусу з кронштейна, тож сервіс не потребує повного розбирання.

На задній частині корпусу виведено два основні роз'єми: живлення та підключення енкодера, розміщені на плоскій панелі з невеликим заглибленням, щоб штекери сідали врівень і не чіплялися під час роботи (рис. 3.4). Живлення виконано у вигляді стандартного панельного гнізда, а інтерфейс енкодера — у вигляді ключованого роз'єму з фіксацією, що виключає переплутування полярності та каналів А/В. Між роз'ємами залишено технологічний зазор для зручності під'єднання пальцями та щоб штекери не заважали один одному; з внутрішнього боку передбачене вушко для розвантаження натягу кабелів і короткий прямий шлях до плати без різких перегинів.

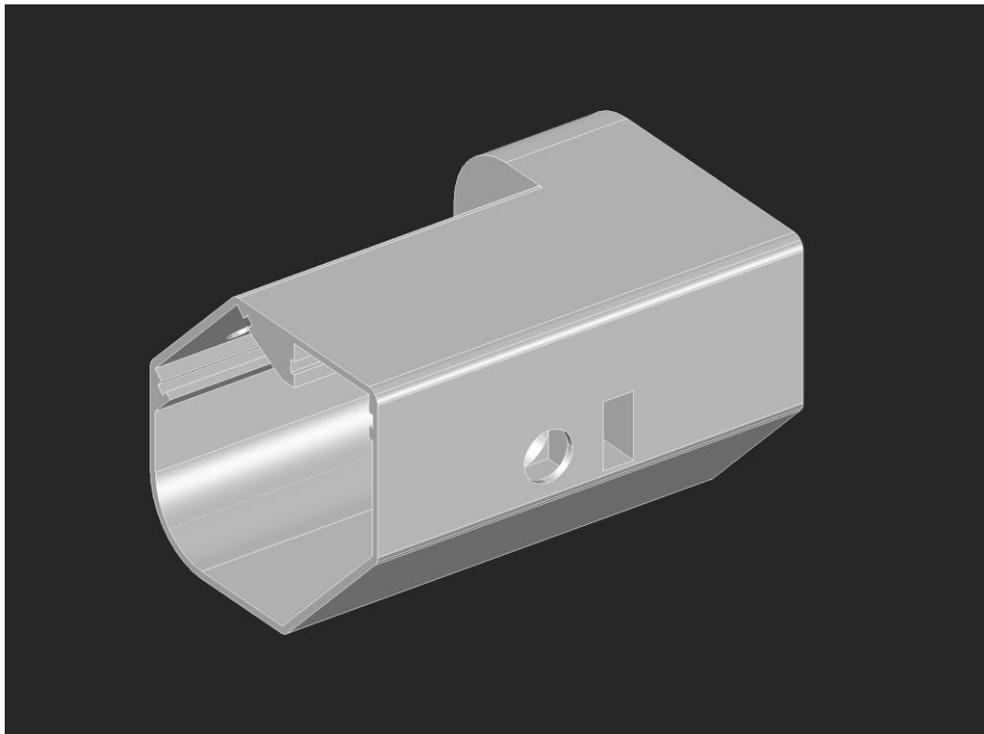


Рисунок 3.4 — Задня частина корпусу з роз'ємами

Посадочні отвори під роз'єми мають фаски для чистого входу штекера, а полиця всередині корпусу підтримує роз'єм по товщині стінки, що підвищує жорсткість і витримує багаторазові підключення. Розпінування підписується на панелі (живлення, А, В, GND), а вільне місце поряд дозволяє в майбутньому

встановити альтернативний роз’єм для енкодерної стрічки без зміни основної геометрії корпусу.

З погляду виробництва 3D-друком корпус спроектований під друк на спинці (рис 3.5), так уся лицьова геометрія виходить без підтримок, а внутрішні полиці формуються чисто. Товщина стінок вибрана рівномірною по периметру, щоб уникнути локальної усадки; усі вертикальні кути мають внутрішні радіуси для надійного з’єднання шарів. Кришка друкується окремо плазом — її посадочна чверть тоді виходить рівною, і кришка щільно сідає без доопрацювання. Щоб різьбові вузли витримували багаторазове відкривання, у стійки передбачені гнізда під закладні гайки— це проста операція після друку, яка різко підвищує ресурс.

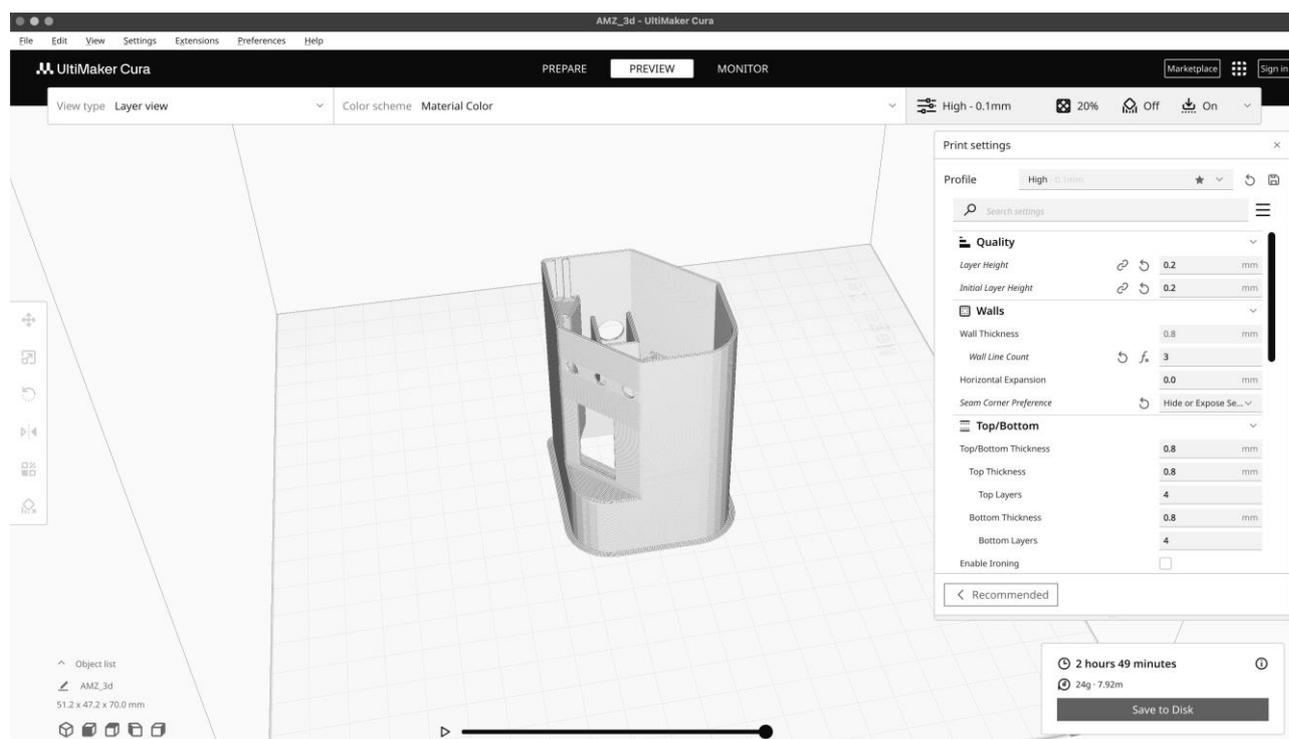


Рисунок 3.5 — Розміщення корпусу для 3D-друку

Маршрут проводів продумано ще на етапі моделі. Вхідний шлейф від датчика заходить через задню частину і відразу фіксується розвантаження натягу — навіть якщо кабель смикнуть, сила не передається на роз’єм і не зігне плату. Під шлейф дисплея залишено окремий канал, щоб він не перетинався з силовими проводами і не затискався кришкою. Всередині достатньо місця для невеликого

запасу проводу — це полегшує збирання і дає можливість безпечно відкрити корпус при обслуговуванні, не від'єднуючи роз'єми.

Ергономіка закладена в геометрію зовнішнього профілю. Корпус має заокруглені боки — його зручно тримати під час монтажу, а плавний перехід у скошений фронт природно орієнтує дисплей до очей оператора. Нахил панелі одночасно зменшує глибину тіні від рамки дисплея, тож цифри не «провалюються» в нішу. Габарити підібрані так, щоб корпус не перекривав огляд робочої зони і не заважав штатним рукояткам. Для кріплення передбачені плоскі площадки — їх легко використовувати з кутниками, магнітними підставками або простими гвинтовими кронштейнами.

У підсумку отримано корпус (рис. 3.6), який поєднує просту збірку, жорсткість у критичних зонах і зручність щоденного користування. Під час проектування якого були враховані можливі похибки друку, який добре центрує вузли під час складання і забезпечує стабільну посадку дисплея та кнопок без підгонки — саме цього і вимагає прототип, який планується використовувати як основу для подальшої модернізації верстатів.



Рисунок 3.6 — Вигляд корпусу надрукованого на 3D-принтері

3.3 Розробка електричної схеми

Насамперед електрична схема була розбита на кілька простих вузлів: живлення, мікроконтролер, інтерфейс енкодера, блок кнопок і індикатор. Такий поділ відразу задає логіку з'єднань і мінімізує кількість перехресть доріжок. Мікроконтролер обрано з апаратними перериваннями на лініях D2/D3 — саме сюди заведені канали А і В оптичного енкодера, що дозволяє рахувати події по обох фронтах без втрат. Дисплей під'єднано по апаратному SPI (SCK/MOSI спільні, керуючі — DC та RES), чип-select у модулі зафіксовано апаратно, щоб зекономити один пін. Блок керування — три тактові кнопки, які підключені до А1—А3 в режимі INPUT_PULLUP; спільну землю для них подаємо з А0, налаштованого як вихід у нуль — це спрощує джгут і зменшує кількість проводів до загальної шини. Така прив'язка виводів відповідає коду прошивки й дає однозначну трасу для плати та шлейфів (рис. 3.7).

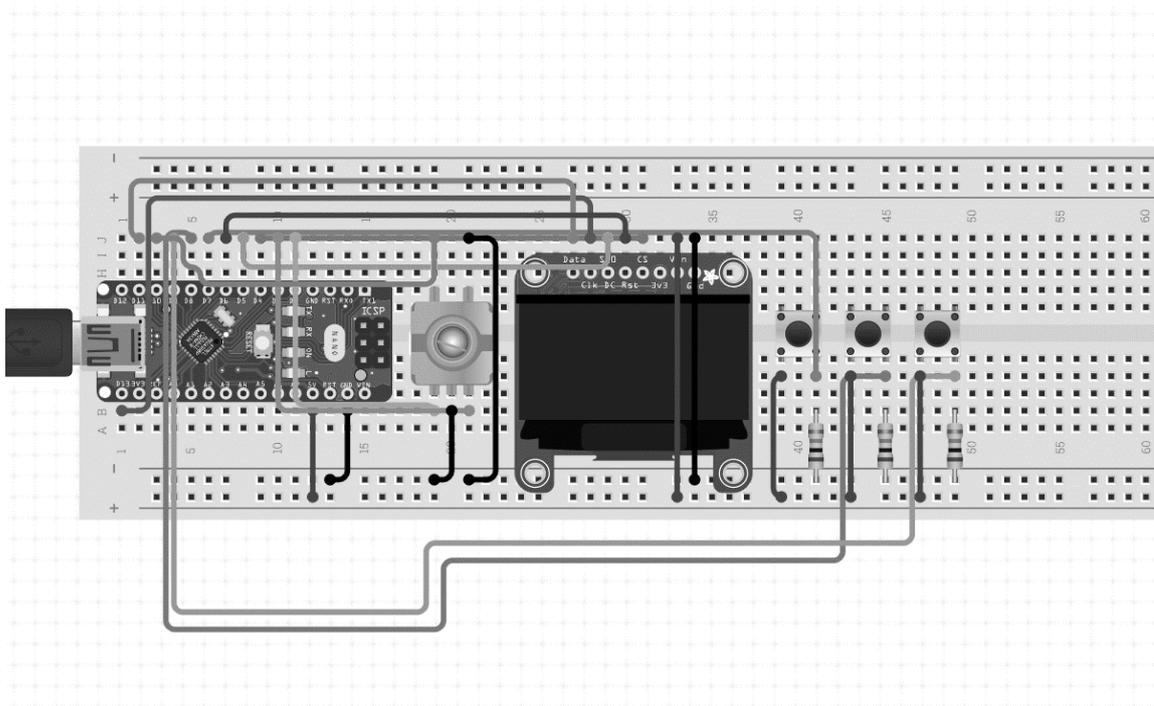


Рисунок 3.7 — Макет електричної схеми пристрою

Підсистема живлення побудована на стабілізаторі типу RD7805 із запасом по струму для мікроконтролера та дисплея. Передбачені три рівні розв'язки:

вхідний електроліт (гасить пульсації після джерела), керамічний конденсатор біля стабілізатора та по 100 нФ біля мікроконтролера й дисплея — короткими доріжками впритул до виводів. На вході встановлюється захист від переполюсування і допускається встановлення запобіжника малої інерційності.

Гніздо живлення винесене на задню панель, а всередині кабель одразу фіксується у вушку розвантаження натягу, щоб випадкове смикання не передавалося на плату.

Інтерфейс енкодера спроектовано з урахуванням швидких фронтів і необхідності надійно визначати напрям. Канали А/В заведені на D2/D3 з внутрішнім підтягуванням до +5 В; при використанні довших кабелів рекомендується додати невелике RC-фільтрування на вході або використовувати екранований провід. Бажано дотримати симетрію довжин ліній і прокладати їх парою до роз'єму — це зменшує різницю затримок між каналами. На панель виведено ключований роз'єм із підписаними контактами А, В, +5V, GND; у схемі залишене місце під TVS/ESD-захист, аби багаторазові підключення датчика не виводили з ладу входи. Така обв'язка повністю збігається з логікою прошивки: обробник переривань читає стан обох каналів і коректно визначає знак інкремента.

Блок кнопок реалізований максимально просто й надійно. Кожна кнопка працює на замикаючий контакт «на землю (до А0), а вхід мікроконтролера тримається у «1» внутрішньою підтяжкою — жодних зовнішніх резисторів не потрібно. Програмна антидребезг-фільтрація виконується бібліотечним класом, тому апаратні RC-ланцюжки не обов'язкові; тим не менш на платі залишено місця під SMD-резистор і конденсатор у розрив, якщо буде потреба підвищити стійкість до механічних коливань. Кнопки розміщені компактно, доріжки короткі, повернення на «землю» зроблено окремою доріжкою до вузла кнопок, а не петлею через інші модулі — це знімає паразитні перешкоди на входах під час перемальовування дисплея по SPI.

Підключення дисплея спроектоване з пріоритетом до стабільності кадру. Лінії SCK та MOSI йдуть паралельно коротким шляхом, поруч із ними —

локальна «земля»; керуючі лінії DC та RES виведені на окремі піни (аналогові A4/A5, використані як цифрові), що відповідає виклику конструктора в коді. Якщо використовується модуль OLED із власним стабілізатором/узгодженням рівнів, до нього подається 5 В без додаткових конвертерів; якщо передбачено альтернативне живлення дисплея або плата-адаптер. Усі з'єднання згруповані в один шлейф до фронтальної полиці — це спрощує складання та виключає натяг при закритті кришки. На задній панелі поруч із гніздом живлення розміщено роз'єм енкодера; між ними залишено технологічний зазор, а посадочні місця під роз'єми підсилені стійками — отвори не розбиваються від багаторазових підключень.

У результаті електрична схема вийшла однозначною для трасування і зручною в налагодженні є зрозуміла прив'язка пінів до функцій, тест-пойнти на 5V і GND для швидкої перевірки, короткі шляхи для швидких сигналів і запасні місця під фільтри/захисти. Така структура дозволила швидко зібрати макет на проводах (рис. 3.8), перевірити правильність рахунку енкодера, реакцію кнопок і стабільність відрисовки, після чого перенести схему на плату без змін логіки. Рішення залишають простір для подальшої модернізації: можна замінити модуль дисплея, додати альтернативний роз'єм під лінійну шкалу чи винести живлення на інший тип гнізда — базова топологія від цього не змінюється.

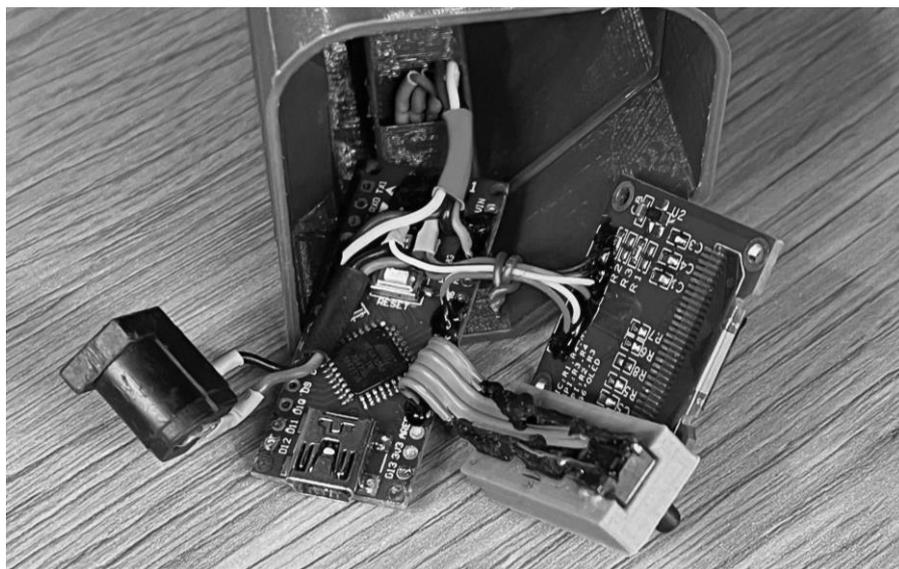


Рисунок 3.8 — Вигляд електронної частини пристрою

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Обмеження операцій по часу та пропускна здатність переривань

Одним із ключових параметрів є те як часто система здатна обробляти події від датчика, скільки часу витрачається на їх обробку та яким має бути режим роботи головного циклу, щоб імпульси не губилися у будь-яких режимах роботи. Відлік подій приходить нерівномірно: у реальній механіці бувають відрізки повільного руху, ділянки розгону, короткі реверси, а ще — пакети хибних фронтів через завади або некоректні переходи фронтів. Тому модель навантаження будується не за середнім значенням, а за верхньою оцінкою частоти подій, яку здатен породити сенсор у крайньому випадку. Для квадратури ця оцінка прив'язується до максимальної механічної швидкості вала або каретки та до способу підрахунку подій. Далі ця верхня частота переноситься в часову шкалу: між будь-якими двома подіями може бути дуже малий інтервал, і саме в цей інтервал система повинна встигнути виконати весь критичний код. Для обрахунку кількості подій які повинна обробляти система за одиницю часу можна скористатися цією формулою:

$$f_{\text{events}} = C \cdot M \cdot \frac{\omega}{2\pi},$$

де C — кількість імпульсів на оберт;

M — множник;

ω — кутова швидкість.

Так формується перше правило: ми орієнтуємося на пікову частоту, а не на типовий режим роботи.

Для того щоб короткий детермінований фрагмент коду, який виконується негайно у відповідь на апаратну подію і тимчасово перехоплює керування у «головного» циклу. Її головний обов'язок — мінімальною ціною за час оновити критичний стан: зчитати прапор чи входи, обчислити найпростішу дію,

зафіксувати результат у спільних змінних і якнайшвидше повернутися. Через підвищений пріоритет ISR заборонено робити все, що може дати непередбачувану або довгу затримку: виклики виводу на дисплей та консоль, динамічні виділення пам'яті, блокувальні очікування, тривалі цикли, доступ до повільних шин. Дані, якими діляться ISR і основний код, слід позначати як `volatile`, а операції над багатобайтними змінними виконувати атомарно (або з коротким вимкненням переривань), щоб уникнути гонок і «розірваних» читань. Тривалість ISR треба тримати сталою та вимірюваною, інакше з'являється джиттер, пріоритетні збої та ризик втрати подій за пікових частот. Практично ISR має бути «тонкою»: для енкодера — прочитати біти A/B, зіставити перехід із таблицею легітимних станів, інкрементувати або декрементувати лічильник, оновити напрям і негайно вийти; будь-яку подальшу роботу (масштабування, фільтрацію високого рівня, оновлення інтерфейсу, запис в пам'ять) варто відкласти у головний цикл через прості прапорці або кільцевий буфер. Вкладені переривання та тимчасові заборони слід застосовувати обережно й на відомий, дуже малий час, щоб не «закрити» прийом нових подій. Варто контролювати використання стеку й уникати рекурсії в ISR, слідкувати за пріоритетами, щоб не виникало інверсії пріоритетів, і мати план «самовідновлення»: якщо прийшов нелегітимний перехід, таблиця просто не змінює лічильник, а нормальна послідовність одразу повертає систему у валідний стан. Такий підхід забезпечує пропускну здатність без втрат та передбачувану часову поведінку усієї вимірювальної підсистеми.

Щоб витримати пікові інтервали, обробник переривань (ISR) роблять максимально коротким і детермінованим. До нього входить лише те, що безпосередньо пов'язане з валідністю переходу і оновленням лічильника: читання двох бітів стану, перетворення їх у код стану, перевірка коректності та інкремент або декремент ціло чисельного відліку. Усе інше — форматування числа, оновлення екрану, робота з кнопками, збереження у пам'ять — виноситься з ISR у основний цикл програми. Така структура гарантує стабільний час обслуговування кожного переривання незалежно від того, що робить

користувацький інтерфейс. Можна зафіксувати цю логіку у вигляді короткого псевдокоду:

- прочитати стан каналів А та В;
- порівняти попередній та новий стан;
- якщо перехід коректний, відкоригувати лічильник;
- повернутися до виконання основного циклу.

Критично важливо не викликати у перериванні довгі та непередбачувані операції, зокрема роботу з дисплеєм, динамічні виділення пам'яті, друк в консоль або сторонні бібліотечні виклики.

Поза перериваннями працює головний цикл, який повинен м'яко «поглинати» завдання із різною тривалістю, не створюючи заторів. Планувальна політика тут проста: спочатку — обробка подій, що впливають на вимірювання (зчитування «сирого» лічильника, перерахунок у кут або довжину, розрахунок різниці до мітки), потім — легкі оновлення інтерфейсу, та лише після цього — рідкі операції на кшталт збереження параметрів у пам'ять. Щоб інтерфейс не блокував вимірювання, використовують два прийоми. Перший — часткове оновлення екрану: перерисовується тільки те, що змінилося, а великі шрифти та графічні елементи не «мигають». Другий — дозування частоти оновлень: навіть якщо події приходять дуже швидко, цифри не обов'язково малювати у кожному циклі, достатньо тримати останнє коректне значення та оновлювати його з відомою періодичністю.

Ще один елемент політики — асинхронні події від кнопок. Їхній дребезг і обробка не повинні потрапляти в критичний шлях вимірювання. Практика підказує: опитування кнопок виконується у головному циклі з м'яким антидребезгом, а зміна режимів (наприклад, увімкнення паузи або входу в меню) має лише виставляти прапорець, без важких дій у момент натискання. Якщо потрібне збереження параметрів, краще ставити «заявку на запис» і виконувати її, коли система бачить, що темп подій низький. Так ми не ризикуємо втратити імпульси в мить, коли користувач взаємодіє з інтерфейсом.

Умови відсутності втрати імпульсів формулюються як поєднання трьох

вимог. По-перше, час обслуговування переривання має бути малим та стабільним на фоні мінімального інтервалу між подіями. По-друге, тривалі операції головного циклу не повинні масово забороняти переривання або створювати довгі критичні секції; якщо у окремих місцях заборона необхідна, її тривалість має бути відомою і значно меншою за середній міжімпульсний інтервал. По-третє, у системі має бути механізм «самоочищення» після збурень: якщо через завади стався нелегітимний стрибок станів, таблиця валідації просто не змінить лічильник і при наступному валідному переході відлік продовжитья без накопичення похибки. Додатково для довгих ліній сигналу варто передбачити диференційні вхідні каскади та базові апаратні фільтри, щоб на вхід ПЗ потрапляло менше хибних фронтів; але навіть без них таблична перевірка добре тримає логіку підрахунку.

На завершення доцільно оцінити пропускну здатність у цілому: від максимальної механічної швидкості та вибраного режиму підрахунку ми отримуємо пікову частоту подій; знаючи, скільки часу забирає обробник переривань, і яку частоту оновлень дисплея ми допускаємо, легко перевірити запас за часом. Якщо запас недостатній, є три типові шляхи підсилення: спростити ISR до мінімуму, обмежити частоту оновлень інтерфейсу або зменшити спосіб підрахунку (наприклад, замість « $\times 4$ » перейти на « $\times 2$ » у режимах високих швидкостей). Усі ці рішення прозоро відображаються на часовій діаграмі і дають контрольований компроміс між чутливістю, плавністю індикації та гарантією безвтратного підрахунку. Такий підхід формує зрозумілий часовий бюджет і забезпечує передбачувану роботу вимірювальної частини незалежно від того, як завантажений інтерфейс або які короткочасні збурення виникають у лінії сигналу.

4.2 Обґрунтування застосування ООП та структура головного класу

Архітектура прошивки побудована навколо єдиного класу, у якому зосереджені стан, інваріанти апаратної частини та вся прикладна поведінка (лістинг 4.1). Такий підхід свідомо протиставляється розрізненим глобальним

змінним і набором функцій без єдиного центру. Клас дає чітку точку входу й визначає життєвий цикл: спочатку одноразова ініціалізація, далі короткі ітерації логіки, за потреби — відрисовка кадру, а підрахунок подій відбувається асинхронно в обробнику переривань. Завдяки цьому часові властивості системи стають передбачуваними: швидкі події не змішуються із відносно повільною графікою, а інтерфейс ніколи не впливає на вимірювальний тракт.

Лістинг 4.1 — Структура класу Program

```
class Program {

public:
    void init();
    void update();
    void draw();

private:
    static void isr();
    enum class MENU_PAGE;
    MENU_PAGE menuPage;
    int32_t counter;
    bool pause;
    bool blink;
    Save *saves;
    Save tempSave;
    Save quickSave;
    int menuSel;
    int menuBound;
    int currSave;
    byte savesCnt;
    byte newNameIndex;
};
```

Інкапсуляція обмежує видимість службових прапорців, лічильників, буферів імен та індикаторів режимів — вони належать класу і змінюються лише методами цього класу. Це прибирає цілі класи помилок, що виникають у процедурному коді, коли різні функції випадково змінюють спільні глобальні змінні. Поліморфізм у даному проєкті не є ключовим, натомість критичною є структурованість станів: перелік екранів оформлено як строгий enum, переходи

між ними локалізовано в одному місці, а дані профілів подано окремою структурою з мінімально достатніми полями. Така організація робить код прозорим для рев'ю і стабільним при розширенні: додавання нового екрана чи параметра не тягне каскаду правок по всьому проєкту.

Логіка класу упорядкована довкола чотирьох базових методів, кожен із чіткою контрактною відповідальністю. `init()` виконує одноразову конфігурацію апаратури та встановлює інваріанти: налаштовує GPIO, дисплей, таймери, прив'язує переривання та готує структури даних. Після його завершення решта коду може покладатися на те, що периферія в коректному стані, а пам'ять — заповнена валідними стартовими значеннями. Саме завдяки цьому відбувається розрив між «конфігураційним» і «робочим» часом: усі дорогі або ризикові операції робляться один раз, і в робочому циклі не виникає спонтанних повторних ініціалізацій, що здатні спричиняти мерехтіння, зависання чи неконсистентні стани.

У методі `update()` послідовно опитуються елементи керування, безпечно зчитується підрахований у перериваннях лічильник, застосовується активний режим, синхронізуються робочі змінні, ведеться навігація сторінками, приймаються рішення про паузу, скидання, зміну профілю. Важливо, що `update()` не займається графікою: він лише встановлює прапорець потреби перерисовки, коли змінюється зміст кадру. Завдяки цьому час виконання `update()` короткий і стабільний у кожній ітерації, а система зберігає запас пропускну здатності для реакції на «піки» подій.

Метод `draw()` виконує чистий рендеринг інтерфейсу на основі вже підготовлених у `update()` даних. Він не змінює логіки та не проводить обчислень, від яких залежить вимірювання; його завдання — синхронно зібрати кадр і вивести його на дисплей сторінковим циклом без мерехтіння. Це жорстке розділення обов'язків «логіка → кадр» робить систему структурованою: будь-який екран можна відтворити за фіксованого стану класу, а поведінку легко перевіряти модульно. Типові проблеми на цьому рівні — спроби частково перерисовувати окремі ділянки й одночасно оновлювати стани; у

запропонованій моделі вони унеможливлені, тому що `draw()` лише зчитує стани.

Метод `isr()`— мінімальний і детермінований обробник переривань, що займається виключно підрахунком квадратурних переходів. Він не форматує числа, не працює з пам'яттю дисплея, не викликає блокувальних API. Усе несуттєве за часом винесено в «повільний» контекст `update()`. Таким чином утримуються верхні межі часу обслуговування переривань, і система масштабовано витримує пікові швидкості обертання без втрати подій. У процедурному підході саме тут часто трапляються фатальні помилки: спроби «обробити все відразу» в ISR, доступ до змінних, які змінюються з двох контекстів, або взаємодія з периферією, що може затримувати виконання на непередбачуваний час.

Сукупно ці чотири ролі формують стійкий конвеєр: апаратні події накопичуються найкоротшим шляхом, прикладна логіка приймає рішення незалежно від графіки, а графіка запускається лише за фактом зміни змісту. Класова форма коду фіксує цей конвеєр у структурі: публічний інтерфейс мінімальний і однозначний, внутрішні дані захищені від несанкціонованого доступу, переходи між екранами відбуваються в одному місці, а профілі вимірювання мають чіткий тип і життєвий цикл. Це знижує когнітивне навантаження на розробника, спрощує рев'ю і розширення, мінімізує ризик помилок виконання та гарантує, що часово критичний вимірювальний тракт завжди має пріоритет перед усім іншим.

Нарешті, такий дизайн добре масштабується. Додавання нової сторінки, додаткової службової ознаки чи ще одного режиму відображення не потребує змін базової логіки програми : достатньо розширити локалізовані гілки `update()` і `draw()`. Це саме те, чого очікують від інженерного ПЗ для вимірювальних приладів: стабільний, передбачуваний каркас із чіткими межами відповідальності, який дозволяє розвивати функціональність без ризику зламати базові часові гарантії системи.

4.3 Опис функцій обробки вхідних даних

Метод `update()` це частина у якій описана вся базова логіка програми, у якій обробляються усі вхідні данні з датчиків та від користувача. Він працює в єдиному послідовному контексті головного циклу, де опитує введення, приймає рішення залежно від активної сторінки меню та встановлює прапорець потреби перерисовки. Критично, що тут не виконується важка графіка й не торкаються апаратні налаштування - лише чиста логіка

На початку методу обпитуються кнопки та енкодер, окремий таймер задає ритм миготіння курсора/маркерів, не пов'язаний із частотою подій(лістинг 4.2). Це розриває залежність між частотою руху вала й «живістю» інтерфейсу: навіть якщо енкодер стоїть, курсор лишається «живим», а якщо енкодер крутять дуже швидко — миготіння не прискорюється і не тягне зайвих перерисовок кадру. Типова помилка на цьому місці, прив'язати миготіння до змін значення або до проходів головного циклу без таймера: тоді візуальний ритм стає випадковим.

Лістинг 4.2 — Оновлення таймера та станів кнопок

```
btn1.tick();
btn2.tick();
btn3.tick();
enc.tick();

if (blinkTm.isReady()) {
    blink = !blink;
    redraw = false;
}
```

В стані MAIN три кнопки виконують три незалежні дії: скидання вимірювання, перехід до меню збережень, пауза/продовження. Важливий нюанс — при знятті паузи лічильник енкодера синхронізується з робочим відліком, щоб показ одразу відповідав реальній позиції. Це усуває поширену логічну яму, коли після паузи індикатор стрибає або починає рахувати «від старого значення». Останній блок плавно підтягує відображуване значення до апаратного лічильника лише коли не активна пауза (лістинг 4.3). Навігація списком

побудована так, щоб поєднати крокові натискання й аналогове прискорення від енкодера. За великого відхилення лічильника застосовано масштаб 1 до 10 — це дозволяє швидко прокручувати довгі списки, не втрачаючи контролю на коротких рухах. Після зміни індексу курсора код нормалізує межі та зсуває вікно видимості, щоб активний рядок залишався в межах 5-рядкового екрану.

Лістинг 4.3 — Логіка обробки натискань кнопок у стані MAIN

```

case MENU_PAGE::MAIN:

    if (btn1.click()) {
        counter = 0;
        enc.counter = 0;
        pause = false;
        redraw = false;
        enc.counter = counter;
    }

    if (btn2.click()) {
        menuPage = MENU_PAGE::SAVES;
        pause = false;
        counter = 0;
        enc.counter = 0;
        menuSel = 0;
        menuBound = 0;
        redraw = false;
    }

    if (btn3.click()) {
        pause = !pause;
        if (!pause)
            enc.counter = counter;
        redraw = false;
        enc.counter = counter;
    }

    if (!pause && counter != enc.counter) {
        counter = enc.counter;
        redraw = false;
    }

    break;

```

Це прибирає «стрибаючу» поведінку й робить навігацію передбачуваною. Далі — єдиний обробник підтвердження вибору, який або повертає на головний екран, або запускає потрібний майстер, або застосовує збережений профіль (лістинг 4.4).

Лістинг 4.4 — Логіка обробки натискань кнопок у стані MAIN

```

case MENU_PAGE::SAVES:
    if (btn1.click()) {
        menuSel--;
        enc.counter = 0;
        redraw = false;
    }
    if (btn3.click()) {
        menuSel++;
        enc.counter = 0;
        redraw = false;
    }
    if (menuSel > menuBound + 4)
        menuBound = menuSel - 4;

    if (btn2.click()) {
        switch (menuSel) {
            case 0:
                menuPage = MENU_PAGE::MAIN;
                break;
            case 1:
                menuPage = MENU_PAGE::NEW_SAVE_NAME;
                counter = 0;
                enc.counter = 0;
                break;
            case 2:
                menuPage = MENU_PAGE::QUICK_SAVE_DIST;
                counter = 0;
                enc.counter = 0;
                break;
            default:
                menuPage = MENU_PAGE::MAIN;
                counter = 0;
                break;
        }
        redraw = false;
    }
}

```

Функція Quicksave складається з трьох послідовних сторінок, які мінімізують ризик переплутати порядок операцій при налаштуванні масштабу. На першому кроці задається базова «відстань» у кроках; на другому — кількість кроків (через «пакетний» приріст 10 за обертом, що зручно набирати), після чого масштаб перетворюється діленням; на третьому — кількість десяткових знаків обмежується діапазоном і фіксується в тимчасовому профілі. Важливі дрібниці: на кожному кроці скидається `enc.counter`, щоб жести не переносилися між екранами; усі переходи ставлять `redraw=false`, дозволяючи `draw()` самостійно відпрацювати повний кадр (лістинг 4.5).

Лістинг 4.4 — Логіка додавання нової метрики у пам'ять

```
case MENU_PAGE::QUICK_SAVE_DIST:

    if (btn1.click()) {
        counter = 0;
        redraw = false;
    }

    if (btn2.click() && abs(counter) > 0) {
        menuPage = MENU_PAGE::QUICK_SAVE_COUNT;
        quickSave.koef = abs(counter);
        pause = false;
        counter = 0;
        enc.counter = 0;
        redraw = false;
    }
    if (btn3.click()) {
        pause = !pause;
        if (!pause)
            enc.counter = counter;
        redraw = false;
    }

    if (!pause && counter != enc.counter) {
        counter = enc.counter;
        redraw = false;
        quickSave.koef = abs(counter);
        pause = false;
        counter = 0;
    }
}
```

4.4 Опис функцій відображення інтерфейсу користувача

Метод `draw()` відповідає лише за відображення кадру, він не приймає рішень та не змінює станів, а лише формує цілісний кадр на основі вже підготовлених у `update()` даних. Така дисципліна робить поведінку передбачуваною: швидкі події квадратури обслуговуються незалежно, а інтерфейс малюється тільки тоді, коли змінився зміст кадру (сам цикл `firstPage()/nextPage()` викликається поза межами `draw()`, в головному циклі). Внутрішньо рендерер побудований так що кожен екран має свою компактну гілку із обмеженою відповідальністю. Всі спільні елементи оформлені окремою функцією `drawValue()`, що забезпечує однаковий формат чисел, стабільну геометрію та контроль за дробовою частиною. Головна сторінка вирішує дві задачі: показати числове значення в обраній одиниці і дати мінімальний контекст (рис 4.1).



Рисунок 4.1 — Вигляд головної сторінки інтерфейсу

Ключовий нюанс — керування перерисовкою показів через умову `!pause || blink`. Коли вимір заморожено, відмальовується лише миготіння (позначка «дані

затримано»), а не беззмістовне перерисовування того самого числа (лістинг 4.5). Це розвантажує шину даних дисплея, утримує рівний ритм кадрів і одночасно наочно сигналізує користувачу режим роботи. Числове поле малюється через `drawValue()`: спершу ціла частина великим шрифтом, потім крапка й дробова частина — це прибирає ефект «стрибаючих» цифр при зміні розрядів.

Тут усі координати підбрані так, щоб утворити «праву колонку» для числа і «нижню лінійку» для підпису. В `init()` попередньо встановлено вирівнювання шрифту по верхній межі (`setFontPosTop()`), тож обчислення Y-координат просте й повторюване. Обрані шрифти виконують різні ролі: великий «цифровий» (`u8g2_font_fur30_tn`) для значення і компактний (`u8g2_font_fur11_tf`) для підписів. Саме розділення «типографічних каналів» прибирає візуальний шум і зберігає читабельність на 128×64. Сторінка зі списком профілів відмальовується в «вікні» з п'яти рядків (лістинг 4.6). Це рішення уникає резких перелітань екрана: коли курсор рухається, зміщується лише нижня межа «вікна», а координати рядків лишаються фіксованими.

Лістинг 4.5 — Логіка відрисовки головної сторінки інтерфейсу

```
case MENU_PAGE::MAIN: {
    if (!pause || blink) {
        if (currSave == -1) {
            drawValue(0, counter / quickSave.koef, quickSave.digs);
        } else {
            drawValue(0, counter / saves[currSave].koef, saves[currSave].digs);
        }
    }
}

u8g2.setFont(u8g2_font_fur11_tf);
u8g2.setCursor(0, 47);
if (currSave == -1) {
    u8g2.setCursor(0, 47);
    u8g2.print("");
} else {
    u8g2.setCursor(0, 47);
    u8g2.print(saves[currSave].name);
}
break;
```

Маркер активного рядка реалізовано як миготіння між порожнім колом та залитим диском — такий контраст помітний навіть на монохромі, не зачіпаючи положення тексту. Комбінація «фіксовані Y + миготливий маркер» дозволяє зберегти стабільну геометрію та мінімізує артефакти.

Лістинг 4.6 — Логіка відрисовки списку профілів

```
case MENU_PAGE::SAVES: {
    u8g2.setFont(u8g2_font_fur11_tf);
    for (int i = menuBound; i <= menuBound + 4; i++) {
        if (i == menuSel) {
            if (blink) u8g2.drawCircle(4, 7 + (i - menuBound) * 13, 2);
            else      u8g2.drawDisc (4, 7 + (i - menuBound) * 13, 2);
        }
        u8g2.setCursor(8, (i - menuBound) * 13);
    }
    break;
}
```

Окрема увага приділена індексації службових пунктів. Перші три рядки — це керуючі дії «Back», «New save», «Quick save», та тільки далі йдуть імена користувацьких профілів (`saves[i - 3].name`). Такий розподіл робить код читабельним та позбавляє умов із «магічними числами», а отже, виключає ризик переплутати візуальний порядок зі структурою даних (рис. 4.2).



Рисунок 4.2 — Вигляд сторінки зі списком профілів

Для набору назви профілю використаний простий, але дуже наочний прийом: каретка ставиться не в «клітинку», а на позицію, що відповідає фактичній ширині вже набраних символів (рис. 4.3). Це досягається викликом `getStrWidth()` для підрядка до `newNameIndex`. Далі у цю точку виводяться «сусідні» символи алфавіту — попередній і наступний. Користувач одразу бачить, що саме зміниться за оберту енкодера вліво/вправо. Буфер `tempSave.name` оновлюється синхронно, тому в кадрі немає «розриву» між тим, що збережено, і тим, що видно (лістинг 4.7).

Лістинг 4.7 — Код відрисовки рядка вводу імені профілю

```
case MENU_PAGE::NEW_SAVE_NAME: {
    u8g2.setFont(u8g2_font_fur11_tf);
    u8g2.drawStr(0, 0, "Set save name:");

    int f = 0;
    if (newNameIndex > 0)
        f = u8g2.getStrWidth(tempSave.name.substring(0, newNameIndex).c_str());

    tempSave.name[newNameIndex] = s[counter];

    u8g2.setCursor(f, 20);
    u8g2.print(counter == 0 ? s[37] : s[counter - 1]);

    u8g2.setCursor(0, 35);
    u8g2.print(tempSave.name);

    u8g2.setCursor(f, 50);
    u8g2.print(counter == 37 ? s[0] : s[counter + 1]);
    break;
}
```

Завдяки фіксованому алфавіту та кільцевій навігації немає проблем із кодуваннями та відсутніми символами у шрифті; користувач не може ввести невидимий символ, а компоновка рядка не руйнується від випадкових знаків. Це значно спрощує перевірку і зменшує кількість кутових сценаріїв.

Уніфікований рендер чисел (лістинг 4.8) — головний фактор стабільного інтерфейсу. Щоб уникнути нестабільного міжсимвольного інтервалу та різної

ширини цифр, ціла частина друкується великою цифрою в одному місці, а крапка та дробова частина — окремо, з точним контролем їхніх координат. Від’ємність позначається графічним маркером-боксом, що завжди має однакову ширину і не зсуває розрядну сітку.



Рисунок 4.3 — Вигляд сторінки вводу назви профілю

Дробова частина виводиться тільки якщо це дозволяє розрядність і задана точність — це запобігає накладанню на назви елементів інтерфейсу чи обрізанню тексту.

Лістинг 4.6 — Логіка відрисовки списку профілів

```
void drawValue(byte pos, double val, byte digits) {
    if (val < 0) {
        u8g2.drawBox(3, 20 + pos, 10, 3);
        val *= -1;
    }
    u8g2.setFont(u8g2_font_fur30_tn);
    u8g2.setCursor(13, 6 + pos);
    u8g2.print((int32_t)val);
    if ((int32_t)val < 10000 && digits > 0) {
        int l = String((int32_t)val).length();
        u8g2.drawDisc(15 + 22 * l, 35 + pos, 2);
        u8g2.setCursor(17 + 22 * l, 6 + pos);
        u8g2.println(String(val, digits).substring(l + 1, l + 1 + digits));
    }
}
```

Координата крапки обчислюється від довжини цілої частини: 1 — кількість її цифр, а далі використовується сталий крок по X (≈ 22 пікселі на розряд для вибраної гарнітури). Це запобігає «плаванню» дробових знаків і тримає композицію рівною для 1—4-розрядних чисел. Обмеження на $((\text{int32}_t)\text{val} < 10000)$ — свідомий компроміс: при п'ятизначній цілій частині на 128×64 дробова частина втратить читабельність, тож її коректно відключити повністю.

Екрани «Set distance», «Set count», «Set accuracy» мають однаковий композиційний принцип: заголовок зверху малим шрифтом та велике числове поле нижче. На етапі точності демонструється прев'ю із «демо-значенням» (0.1234) та поточною кількістю десяткових знаків — це показує, як виглядатиме формат без зайвої логіки масштабування під час введення. Числове поле зсувається на $\text{pos}=16$, звільняючи простір під заголовок, та використовує той самий підхід `drawValue()`, що й основний екран. Результат — повна візуальна консистентність між сторінками без дублювання верстки.

Метод `draw()` не викликає `firstPage()/nextPage()` — цим займається головний цикл, який робить повний рендер лише за фактом зміни стану. Це принципово важливо для продуктивності: складні екрани (списки, редагування) не відмальовуються дарма й не конкурують із підрахунком квадратури. Миготіння контролюється окремим таймером, не залежить від частоти подій та жодним чином не впливає на час ISR. У межах `draw()` відсутні блокуючі операції з пам'яттю чи взаємодії з периферією, які могли б створити непередбачувану латентність.

Геометрія підібрана так, щоб витримувати типові крайні сценарії: від'ємні значення, 4 розряди в цілій частині, нульова точність, довгі імена профілів в межах ширини дисплея. Там, де можливі конфлікти, наприклад, п'ятизначна ціла частина, дробова частина вимикається цілком — не обрізається, щоб не створювати помилкових асоціацій. Локалізація логіки відрисовки за екранами знімає перехресні ефекти: зміни у меню не можуть вплинути на головну сторінку, а модифікації редактора імен не зсуватимуть координати числового поля.

5 ТЕСТУВАННЯ

Перевірку працездатності було проведено безпосередньо на верстаті, оптичний енкодер був закріплений на валу через перехідник (рис.5.1), самий пристрій встановлений безпосередньо на корпусі верстату (рис. 5.2), живлення від блока живлення напругою 5 вольт та силою струму 1А. Після скидання нуля оператор обертав рукоятку та переміщував вузол у робочому діапазоні. Покази на екрані змінювалися плавно й передбачувано, без ривків та зависань. Під час зміни напрямку відлік коректно йшов у зворотний бік. Для контрольного порівняння орієнтувалися на штатну шкалу верстата та вимірну лінійку — розходжень, помітних на практиці, не було зафіксовано.



Рисунок 5.1 — Розміщення енкодера на верстаті

Також були перевірені основні функції, режим паузи зупиняє індикацію: при прокручуванні валу цифри не змінюються, а після повернення в роботу покази одразу підтягується до фактичного положення. Функція мітка працює за

очікуванням: у вибраній точці ставиться позначка, на головному екрані відображається відстань до неї та напрям, що зручно при підводі до розміру.

Окремо було перевірено збереження користувацьких одиниць (рис. 5.3): створений профіль, задана назва, коефіцієнт та точність — після перезапуску вони збереглися й застосувалися без додаткових дій. Швидке збереження дозволило оперативно завести ще один профіль без збереження в пам'ять та не заходячи у повне меню.



Рисунок 5.2 — Розміщення пристрою на верстаті

Кнопки відпрацьовують чітко, короткі натискання переведуть між пунктами, центральна підтверджує вибір, скидання нуля відбувається одразу. Дисплей читається з робочої позиції — великий шрифт основних цифр, підписи не перекриваються. Під час тривалої роботи із періодичними переміщеннями пристрій не перезавантажувався, випадкових збоїв не було зафіксовано; від'єднання та під'єднання живлення проходить штатно, налаштування із пам'яті

підхоплюються. Роз'єми на задній панелі тримають штекери впевнено, кабель не заважає переміщенням. Узагальнюючи результати, прототип у повному обсязі підтвердив працездатність в реальних умовах, монтаж на верстат займає кілька хвилин, інтерфейс освоюється швидко, покази стабільні та достатні для підведення до розміру та повторюваних переходів. Основні функції працюють без збоїв та справді скорочують час налаштувань. Корпус із роз'ємами зручний в щоденному користуванні, кабелі не заважають роботі, після відключення живлення параметри зберігаються в постійній пам'яті.



Рисунок 5.3 — Меню зі списком функцій

6 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ

6.1 Проведення комерційного та технологічного аудиту розробки

У сучасних умовах цифровізації виробництва та переходу до високоточної металообробки зростає потреба у впровадженні інтелектуальних мікропроцесорних систем контролю та позиціонування. Висока конкуренція на ринку вимагає від підприємств підвищення точності, надійності та швидкодії вимірювального обладнання. Тому розробка системи точного вимірювання й позиціонування деталей є актуальною та відповідає загальним тенденціям розвитку “розумного” машинобудування.

Проведення комерційного аудиту такої розробки дозволяє оцінити її ринкові перспективи, економічну доцільність та конкурентоспроможність. Це важливо для визначення потенційних споживачів, обсягів інвестицій та строків окупності, а також для формування стратегії просування інноваційного продукту. Комерційний аудит забезпечує обґрунтованість прийняття управлінських рішень щодо подальших етапів проекту.

Технологічний аудит, своєю чергою, дає змогу комплексно оцінити технічний рівень, інноваційність, надійність та відповідність розробки світовим стандартам точного машинобудування. Він допомагає виявити потенційні технічні ризики, оптимізувати конструктивні рішення та забезпечити інтеграцію системи у виробничі процеси. У сукупності комерційний і технологічний аудит підвищують шанси успішної реалізації та впровадження мікропроцесорної системи на ринку.

Для проведення комерційного та технологічного аудиту залучаємо 3-х незалежних експертів, якими є провідні викладачі випускової або спорідненої кафедри.

Оцінювання науково-технічного рівня мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки та її комерційного потенціалу здійснюємо із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, а результати зводимо до таблиці 6.1.

Таблиця 6.1 — Результати оцінювання науково-технічного рівня і комерційного потенціалу

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Технічна здійсненність концепції			
Ринкові переваги (наявність аналогів)			
Ринкові переваги (ціна продукту)			
Ринкові переваги (технічні властивості)			
Ринкові переваги (експлуатаційні витрати)			
Ринкові перспективи (розмір ринку)			
Ринкові перспективи (конкуренція)			
Практична здійсненність (наявність фахівців)			
Практична здійсненність (наявність фінансів)			
Практична здійсненність (необхідність нових матеріалів)			
Практична здійсненність (термін реалізації)			
Практична здійсненність (розробка документів)			
Сума балів			
Середньоарифметична сума балів, СБ			

За результатами розрахунків, наведених в таблиці 6.1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки — вищий середнього.

6.2 Розрахунок витрат на здійснення розробки

Належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

Витрати на основну заробітну плату дослідників (Z_o) розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p},$$

де k — кількість посад дослідників, залучених до процесу дослідження;

M_{ni} — місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

T_p — число робочих днів в місяці; приблизно $T_p = (21 \dots 23)$ дні, приймаємо 22 дні;

t_i — число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 6.2.

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт розраховують за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i,$$

де C_i — погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год; t_i — час роботи робітника на виконання певної роботи, год.

Таблиця 6.2 — Витрати на заробітну плату дослідників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	21 725	988	4	3950
Розробник	8 700	395	80	31636
Консультанти	16 600	755	4	3018
Всього:	38605			

Погодинну тарифну ставку робітника відповідного розряду C_i і можна визначити за формулою:

$$C_i = \frac{M_m \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}$$

де M_m — розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), у 2025 році $M_m=8000$ грн;

K_i — коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду;

K_c — мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати, складає 1,1;

T_p — середня кількість робочих днів в місяці, приблизно $T_p = 21...23$ дні, приймаємо 22 дні;

$t_{зм}$ — тривалість зміни, год., приймаємо 8 год.

Таблиця 6.3 — Витрати на заробітну плату робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
--------------------	------------------------	---------------	--------------------------	--------------	----------------

Розробка пз	20	3	59	1,18	1180
Моделювання корпусу	40	3	59	1,18	2360
Пайка електронної схеми	20	3	59	1,18	1180
Всього					4720

Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = 0,1 \cdot (Z_o + Z_p) = 0,1 \cdot (38605 + 4720) = 4332 \text{ грн.}$$

Нарахування на заробітну плату $H_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$\begin{aligned} H_{зп} &= \beta \cdot (Z_o + Z_p + Z_d) = \\ &= 0,22 \cdot (38605 + 4720 + 4332) = 10485 \text{ грн.} \end{aligned}$$

де Z_o — основна заробітна плата розробників, грн.;

Z_p — основна заробітна плата робітників, грн.;

Z_d — додаткова заробітна плата всіх розробників та робітників, грн.;

β — ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

Витрати на матеріали M , що були використані під час виконання даного етапу роботи, розраховуються за формулою:

$$M = \sum_1^n H_i \cdot C_i \cdot K_i - \sum_1^n B_i \cdot C_i,$$

де N_i — кількість матеріалів i -го виду, шт.;

C_i — ціна матеріалів i -го виду, грн.;

K_i — коефіцієнт транспортних витрат,

$K_i = (1, 1 \dots 1, 15)$; n — кількість видів матеріалів.

Таблиця 6.4 — Матеріали, що використані на розробку

Найменування матеріалів	Ціна за одиницю, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Філамент для 3d принтера	500	1	500
Всього, з врахуванням коефіцієнта транспортних витрат			550

Витрати на комплектуючі K , що були використані під час виконання даного етапу роботи, розраховуються за формулою:

$$K = \sum_{i=1}^n N_i \cdot C_i \cdot K_i,$$

де N_i — кількість комплектуючих i -го виду, шт.;

C_i — ціна комплектуючих i -го виду, грн.;

K_i — коефіцієнт транспортних витрат, $K_i = (1, 1 \dots 1, 15)$;

n — кількість видів комплектуючих.

Таблиця 6.5 — Комплектуючі, що використані на розробку

Найменування комплектуючих	Ціна за одиницю, грн.	Витрачено	Вартість витрачених комплектуючих, грн.
Кнопка тактова	5	3	15
Дисплей oled ssd1306	100	1	100
Плата Arduino nano	300	1	300
Оптичний енкодер	900	1	900
Всього, з врахуванням коефіцієнта транспортних витрат			1459,65

До балансової вартості програмного забезпечення входять витрати на його інсталяцію, тому ці витрати беруться додатково в розмірі 10...12% від вартості

програмного забезпечення. Балансову вартість програмного забезпечення розраховують за формулою:

$$V_{\text{прг}} = \sum_1^k C_{\text{іпрг}} \cdot C_{\text{прг.і}} \cdot K_i,$$

де $C_{\text{іпрг}}$ — ціна придбання програмного забезпечення і-го виду, грн.;

$C_{\text{прг.і}}$ — кількість одиниць програмного забезпечення відповідного виду, шт.;

K_i — коефіцієнт, що враховує інсталяцію, налагодження програмного забезпечення,

$K_i = (1, 1 \dots 1, 12)$; k — кількість видів програмного забезпечення.

Таблиця 6.6 — Витрати на придбання програмного забезпечення

Найменування програмного забезпечення	Ціна за одиницю, грн.	Витрачено	Вартість програмного забезпечення, грн.
AutoCAD	25 000	1	2083,3
Всього, з врахуванням коефіцієнта інсталяції та налагодження			2312,5

Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час (чи для) виконання даного етапу роботи.

У спрощеному вигляді амортизаційні відрахування A в цілому бути розраховані за формулою:

$$A = \frac{C_6}{T_B} \cdot \frac{t}{12},$$

де C_6 — загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.;

t — термін використання основного фонду, місяці;

T_B — термін корисного використання основного фонду, роки.

Таблиця 6.7 — Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
3d Принтер	20000	5	1	333,3
Персональний комп'ютер	20000	3	2	1111,1
Паяльна станція	4000	3	1	111,1
Всього	1555,6			

Витрати на силову електроенергію V_e , що мають суттєве значення для виконання даного етапу роботи, приведені у таблиці 6.8:

Таблиця 6.8 — Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
3d Принтер	0,1	20
Персональний комп'ютер	0,35	150
Паяльна станція	0,2	20

Витрати на силову електроенергію V_e , якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

$$\begin{aligned}
 V_e &= \sum \frac{W_i \cdot t_i \cdot C_e \cdot K_{\text{впі}}}{\text{ККД}} \\
 &= \frac{0,1 \cdot 20 \cdot 4,32 \cdot 0,75}{0,98} + \frac{0,35 \cdot 150 \cdot 4,32 \cdot 0,75}{0,98} \\
 &\quad + \frac{0,2 \cdot 20 \cdot 4,32 \cdot 0,75}{0,98} = 193,4 \text{ грн.},
 \end{aligned}$$

де W_i — встановлена потужність обладнання, кВт;

t_i — тривалість роботи обладнання на етапі дослідження, год.;

C_e — вартість 1 кВт електроенергії, 4,32 грн.;

$K_{\text{впі}}$ — коефіцієнт використання потужності;

ККД — коефіцієнт корисної дії обладнання.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (Z_o + Z_p) \cdot \frac{N_{\text{ів}}}{100\%} = (38605 + 4720) \cdot \frac{85}{100} = 36825,86 \text{ грн.},$$

де $N_{\text{ів}}$ — норма нарахування за статтею «Інші витрати».

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...200% від суми основної заробітної плати дослідників та робітників за формулою:

$$N_{\text{нзв}} = (Z_o + Z_p) \cdot \frac{N_{\text{нзв}}}{100\%} = (38605 + 4720) \cdot \frac{140}{100} = 60654,36 \text{ грн.},$$

де $N_{\text{нзв}}$ — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Витрати на проведення розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки. Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = 38605 + 4720 + 4332 + 10485 + 550 + 1459,65 + 2312,5 + 1555,6 + 193,4 + 36825,86 + 60654,36 = 161692,9 \text{ грн.}$$

Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи з розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\eta} = \frac{161692,9}{0,9} = 179658,8 \text{ грн.},$$

де η — коефіцієнт, що характеризує етап виконання науково-дослідної роботи.

Оскільки, якщо науково-технічна розробка знаходиться на стадії впровадження, то $\eta=0,9$.

6.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки, є збільшення у потенційного інвестора величини чистого прибутку.

В даному випадку відбувається розробка засобу, тому основу майбутнього економічного ефекту буде формувати: ΔN — збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу; N — кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки; Π_0 — вартість послуги у році до впровадження інформаційної системи; $\pm \Delta \Pi_0$ — зміна вартості послуги (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta\Pi$ для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta\Pi = (\pm\Delta C_0 \cdot N + C_0 \cdot \Delta N_i)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right),$$

де $\pm\Delta C_0$ — зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки), 8000 грн.;

N — основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки, 50 шт., 150 шт., 200 шт.;

C_0 — основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році, 4000 грн.;

C_6 — основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів 12000 грн.;

ΔN — зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки), 0 шт., 100 шт., 150 шт.;

λ — коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2025 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda = 0,8333$;

ρ — коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати $\rho = 0,2 \dots 0,5$; ϑ — ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2025 році $\vartheta = 18\%$.

Очікуваний термін життєвого циклу розробки 3 роки, тому: 1-й рік: $\Delta\Pi_1=81967$ грн.; 2-й рік: $\Delta\Pi_2=163934$ грн.; 3-й рік: $\Delta\Pi_3=204918$ грн.

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t} = \frac{81967}{(1 + 0,1)^1} + \frac{163934}{(1 + 0,1)^2} + \frac{204918}{(1 + 0,1)^3} = 363956,5 \text{ грн.},$$

де $\Delta\Pi_i$ — збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.;

T — період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо $T=3$ роки);

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t — період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot 3B = 2 \cdot 179658,8 = 359318 \text{ грн.}$$

де $k_{\text{інв}}$ — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання

персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}}=1...5$, але може бути і більшим;

ЗВ — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV = 363956,5 - 359318 = 4639 \text{ грн.},$$

де ПП — приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки, грн.;

PV — теперішня вартість початкових інвестицій, грн.

Оскільки $E_{\text{абс}} > 0$, то можемо припустити про потенційну зацікавленість у розробці мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність $E_{\text{в}}$ або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій $E_{\text{в}}$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної

розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки, розраховується за формулою:

$$E_B = \sqrt[T_{ж}] \left(1 + \frac{E_{абс}}{PV} \right) = \sqrt[3] \left(1 + \frac{4639}{359318} \right) = 0,34,$$

де $T_{ж}$ — життєвий цикл розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки, роки.

Далі розраховуємо період окупності інвестицій T_o , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки:

$$T_o = \frac{1}{E_B} = \frac{1}{0,34} = 2,96 \text{ роки.}$$

Оскільки $T_o < 1 \dots 3$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки мікропроцесорної системи точного вимірювання і позиціонування в просторі деталей для металообробки та виведення її на ринок.

Проведений комерційний та технологічний аудит підтверджує, що розробка мікропроцесорної системи точного вимірювання і позиціонування деталей для металообробки є актуальною та відповідає запитам сучасного високоточного машинобудування. За результатами експертного оцінювання науково-технічний рівень та комерційний потенціал розробки визначено як вищий за середній, що свідчить про наявність перспектив для подальшої реалізації та впровадження.

Розрахунок витрат показав, що загальні витрати на виконання науково-технічної роботи становлять 179,66 тис. грн, що є обґрунтованими з огляду на складність та інноваційність проєкту. Проведена оцінка економічної ефективності демонструє позитивний інтегральний результат: чистий приведений дохід становить 4639 грн, а період окупності інвестицій — 2,96 року, що відповідає прийнятним інвестиційним нормам для подібних розробок.

Таким чином, проєкт може бути розглянутий як економічно доцільний та комерційно привабливий для потенційного інвестора, а його впровадження здатне забезпечити приріст чистого прибутку та підвищення конкурентоспроможності підприємства.

ВИСНОВКИ

У результаті виконання роботи було досягнуто поставленої мети, а саме створено функціонуючий прототип пристрою для точного вимірювання та позиціонування деталей на верстатах. Всі етапи розробки виконано послідовно: від обґрунтування доцільності створення системи до практичної реалізації, налагодження та перевірки працездатності.

Було розроблено електронну схему пристрою на базі мікропроцесорної платформи Arduino із підключенням оптичного енкодера, дисплея та елементів керування. Забезпечено стабільну роботу вузлів, коректне зчитування імпульсів енкодера і відображення результатів вимірювання. Розроблено програмне забезпечення мікропроцесорної системи, яке реалізує основні режими роботи: відображення поточних значень, керування через кнопки та енкодер, збереження користувацьких параметрів одиниць вимірювання, редагування/калібрування коефіцієнтів перерахунку, функцію заморожування показів.

Виконано моделювання корпусу в середовищі AutoCAD та виготовлено корпус методом 3D-друку. Конструкція передбачає розміщення електронних компонентів, органи керування і роз'єм для підключення енкодера, що забезпечує зручність експлуатації та можливість швидкого монтажу.

Наукова новизна роботи полягає в запропонованому підході до побудови мікропроцесорної системи вимірювання на основі оптичного енкодера з реалізацією гнучкого програмного перерахунку у користувацькі одиниці, збереженням налаштувань в енергонезалежній пам'яті та підтримкою сервісних функцій.

Практична цінність роботи полягає у створенні реального працездатного прототипу, що може бути використаний для модернізації верстатів в навчальних та виробничих умовах. Розроблені апаратна схема, програмне забезпечення та конструкція корпусу можуть слугувати основою для подальшого вдосконалення, масштабування під різні типи верстатів і впровадження у практику як доступне рішення для підвищення точності та зручності позиціонування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 «Комп'ютерна інженерія / Азаров О. Д., Дудник О. В., Швець С. І. — Вінниця: ВНТУ, 2023 — 60с. [Електронний ресурс]. — Режим доступу: https://iq.vntu.edu.ua/method/getfile.php?fname=185158.pdf&x=1&card_id=93876&id=185158 (дата звернення: 1.10.2025)
2. Контактні датчики для вимірювання/налаштування на технологічну операцію [Електронний ресурс]. — Режим доступу: <https://www.koda.ua/ukr/products/group.html?id=607> (дата звернення: 4.10.2025)
3. Принцип роботи абсолютного енкодера [Електронний ресурс]. — Режим доступу: <https://megasensor.com/products/princip-raboty-absolyutnogo-enkodera/> (дата звернення: 6.10.2025)
4. Абсолютний поворотний енкодер [Електронний ресурс]. — Режим доступу: <https://www.omchsmmps.com/absolute-rotary-encoder/> (дата звернення: 8.10.2025)
5. Arduino Documentation [Електронний ресурс]. — Режим доступу: <https://docs.arduino.cc/> (дата звернення: 29.10.2025)
6. Контактні датчики для вимірювання/налаштування на технологічну операцію [Електронний ресурс]. — Режим доступу: <https://www.koda.ua/ukr/products/group.html?id=607> (дата звернення: 7.10.2025)
7. Jeremy Blum. Arduino: tools and methods / Jeremy Blum [S. 1.]. — Л. : Свічадо, 2006. — 307 с. — (Wiley.; № 14) (дата звернення: 7.10.2025).
8. Gudekar S. K., Shinge N., Satpute S., Madankar S. B. Object Detection in Video Surveillance System [Електронний ресурс] / S. K. Gudekar, N. Shinge, S. Satpute, S. B. Madnkar // International Journal of Computer Science & Information Technology Research, 2015, vol. 3, № 1, С. 322-327. — Режим доступу: <https://www.researchpublish.com/upload/book/Object%20Detection%20in%20Video%20Surveillance%20System-1250.pdf> (дата звернення: 29.10.2025).

9. Payghode V. Object Detection and Activity Recognition in Video Surveillance [Электронный ресурс] / V. Payghode // International Journal of WIS, 2023, vol. 19, № 3-4, С. 123-160. — Режим доступа: <https://www.emerald.com/ijwis/article/19/3-4/123/160463/Object-detection-and-activity-recognition-in-video> (дата звернения: 29.10.2025).
10. Eckert M. Object Detection Featuring 3D Audio Localization for Blind Assistance [Электронный ресурс] / М. Eckert. — 2018. — С. (...). — Режим доступа: <https://www.scitepress.org/papers/2018/66556/66556.pdf> (дата звернения: 12.10.2025)
11. Pieropan A., Salvi G., Pauwels K., Kjellström H. Audio-Visual Classification and Detection of Human Manipulation Actions [Электронный ресурс] / A. Pieropan, G. Salvi, K. Pauwels, H. Kjellström. — 2014. — С. (...). — Режим доступа: https://people.kth.se/~hedvig/publications/iros_14.pdf (дата звернения: 3.10.2025).
12. Microsoft Learn System.Drawing [Электронный ресурс] — Режим доступа до ресурсу: <https://learn.microsoft.com/ruru/dotnet/api/system.drawing.graphics?view=dotnet-plat-ext-8.0>
13. 7805 Datasheet (PDF) - КЕС(Korea Electronics [Электронный ресурс]. — Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/69437/КЕС/7805.html> (дата звернения: 9.10.2025)
14. 8-bit AVR Microcontroller with 32К Bytes In-System Programmable Flash [Электронный ресурс] /. — 2014. — 298 с. — Режим доступа: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf (дата звернения: 15.10.2025).
15. 128 x 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller [Электронный ресурс] /. — 2008. — 59 с. — Режим доступа: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf (дата звернения: 15.10.2025).

16. Anycubic Kobra S1 User Manual-EN-V1.2 [Электронный ресурс] /. — 2020. — 20 с. — Режим доступа: <https://acop-prod-private.s3-accelerate.amazonaws.com/asop/2025-02/11/pdf/173923633714729900-67aaa3f123f6.pdf?response-content-type=application%2Fpdf&response-content-disposition=inline&X-Amz-Content-Sha256> (дата звернения: 15.10.2025).
17. Serial Peripheral Interface 2 [Электронный ресурс] / Режим доступа: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface (дата звернения: 25.10.2025).
18. TFT LCD Display Technology Practical Handbook From Principles to Commercial Impleme [Электронный ресурс] / Режим доступа: <https://www.stoneitech.com/sharing/tft-lcd-display-technology-practical-handbook-from-principles-to-commercial-implementation/> (дата звернения: 5.10.2025).
19. Pulse-width modulation Impleme [Электронный ресурс] / Режим доступа: https://en.wikipedia.org/wiki/Pulse-width_modulation (дата звернения: 5.10.2025).
20. 1 key Touch Pad Detector IC [Электронный ресурс] /. — 2016. — 8 с. — Режим доступа: <https://radiolux.com.ua/files/pdf/TTP223.pdf> (дата звернения: 15.10.2025).

ДОДАТОК А

Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної техніки

д.т.н., проф. Азаров О.Д.

" 3 " 10 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи
“Комп'ютерна система моніторингу регіонального радіомовлення”
08-23.МКР.006.00.003.ТЗ

Науковий керівник: к.т.н., доц. каф. ОТ._____ Муращенко О.Г.

Студент групи 2КІ-24м

_____ Добровольський Д.С.

1 Підстава для виконання магістерської кваліфікаційної роботи (МКР)

1.1 Актуальність полягає у потребі модернізації металорізальних верстатів простими й недорогими засобами точного вимірювання та позиціонування. Механічні індикатори зношуються й потребують налаштувань, тоді як оптичні датчики та мікропроцесорні плати дають стабільні покази, зручну індикацію та можливість швидкого калібрування під конкретний станок. Розробка переносного пристрою з індикатором, збереженням користувацьких одиниць і функцією мітки скорочує час наладки та підвищує повторюваність операцій;

1.2 Наказ про затвердження теми МКР.

2 Мета МКР і призначення розробки

2.1 Мета роботи — дослідити методи перетворення обертального руху у кутові та лінійні координати, обґрунтувати вибір компонентів, розробити електричну схему й програмне забезпечення, спроектувати корпус, зібрати працюючий прототип та перевірити його на верстаті;

2.2 Призначення розробки — мікропроцесорний пристрій індикації та позиціонування для встановлення на металорізальні верстати; забезпечує відлік переміщення, збереження користувацьких одиниць, калібрування, тимчасову паузу індикації, постановку мітки з показом відстані й напряму до неї, відображення показників на OLED-дисплеї; підтримує підключення валового оптичного енкодера та можливість підключення енкодерної стрічки через окремий роз'єм.

3 Вихідні дані для виконання МКР

3.1 Аналіз існуючих систем вимірювання на верстатах та вибір підходу на основі оптичного датчика;

3.2 Теоретичне обґрунтування підрахунку подій, перетворення обертів у кутові/лінійні величини, базові моделі похибок і вимоги до часу обробки;

3. Розробка принципової схеми та схеми підключень та виготовлення

макета й перенесення на проводовий монтаж;

3.4 Проектування корпусу в CAD, підготовка файлів для 3D-друку, збірка з електронікою;

3.5 Розробка програмного забезпечення пристрою;

3.6 Економічне обґрунтування доцільності.

4 Вимоги до виконання МКР

Пристрій має коректно відображати переміщення з плавною індикацією та правильним знаком при зміні напрямку. Підтримка підключення валового оптичного енкодера та можливість під'єднання датчика з енкодерною стрічкою. Корпус має забезпечити зручний доступ до роз'ємів і кріплень, просте збирання та надійну фіксацію елементів

5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в Таблиці А.1.

Таблиця А.1 — Етапи МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз існуючих систем вимірювання та позиціонування	25.09.2025	30.09.2025	Аналітичний огляд джерел, задачі досліджень, розділ 1
2	Дослідження методів вимірювання	01.10.2025	04.10.2025	Розділ 2
3	Підбір електронних компонентів, розробка електричної схеми	05.10.2025	10.10.2025	Розділ 4
4	Розробка корпусу та пз	11.10.2025	16.10.2025	Розділ 5
5	Підготовка економічної частини	17.10.2025	20.10.2025	Розділ 6

Продовження таблиці А.1

6	Апробація та впровадження результатів дослідження	21.10.2025	24.10.2025	Тези доповідей
7	Опублікування результатів	25.10.2025	26.10.2025	Стаття
8	Оформлення пояснювальної записки	11.11.2025	25.11.2025	ПЗ, графічний матеріал
9	Захист МКР	15.12.2025	19.12.2025	

6 Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами.

7 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

8 Вимоги до оформлювання та порядок виконання МКР

8.1 При оформлюванні МКР використовуються:

- ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;
- ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;
- ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

— методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія»;

— документи на які посилаються у вище вказаних.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

ДОДАТОК Б

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Мікропроцесорна система точного вимірювання і позиціонування в просторі деталей для металообробки

Тип роботи: магістерська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ Кафедра ОТ, факультет ІТКІ, група 2КІ-24м
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КПІ) 1 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак академічного плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту.
- У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- У роботі виявлено ознаки академічного плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. Робота до захисту не приймається.

Експертна комісія:

д.т.н., проф. Азаров О.Д., завідувач кафедри ОТ _____
(прізвище, ініціали, посада) (підпис)

д.т.н., проф. Мартинюк Т.Б., гарант освітньої програми _____
(прізвище, ініціали, посада) (підпис)

Особа, відповідальна за перевірку _____ к.т.н., проф. Захарченко С.М.
(підпис) (прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник _____
(підпис)

к.т.н., доц. каф. ОТ Муращенко О.Г.
(прізвище, ініціали, посада)

Здобувач _____
(підпис)

Добровольський Д.С.
(прізвище, ініціали)

ДОДАТОК В

Алгоритм програми

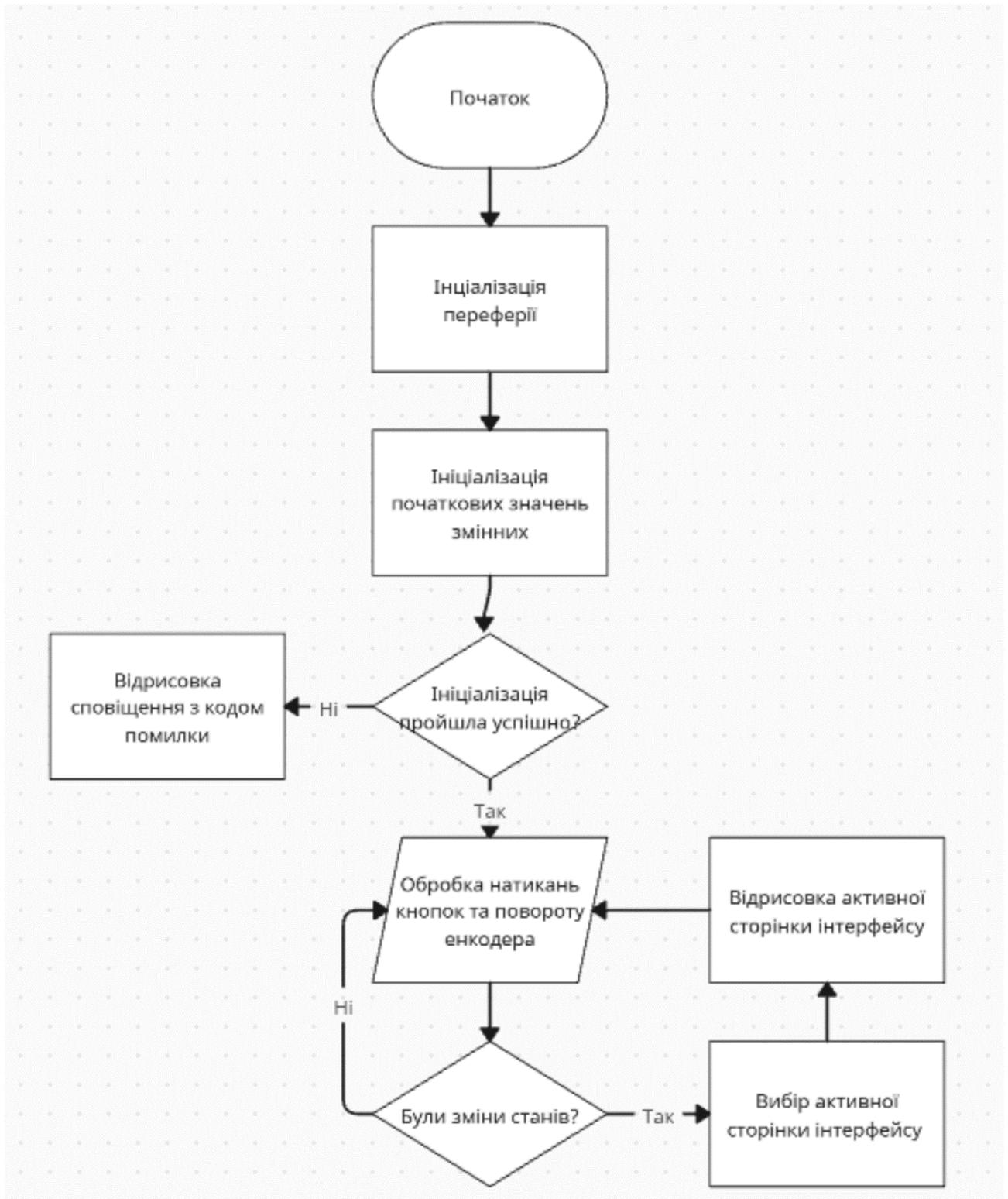


Рисунок В.1 — Блок-схема алгоритму програми

ДОДАТОК Г

Схема електрична принципова

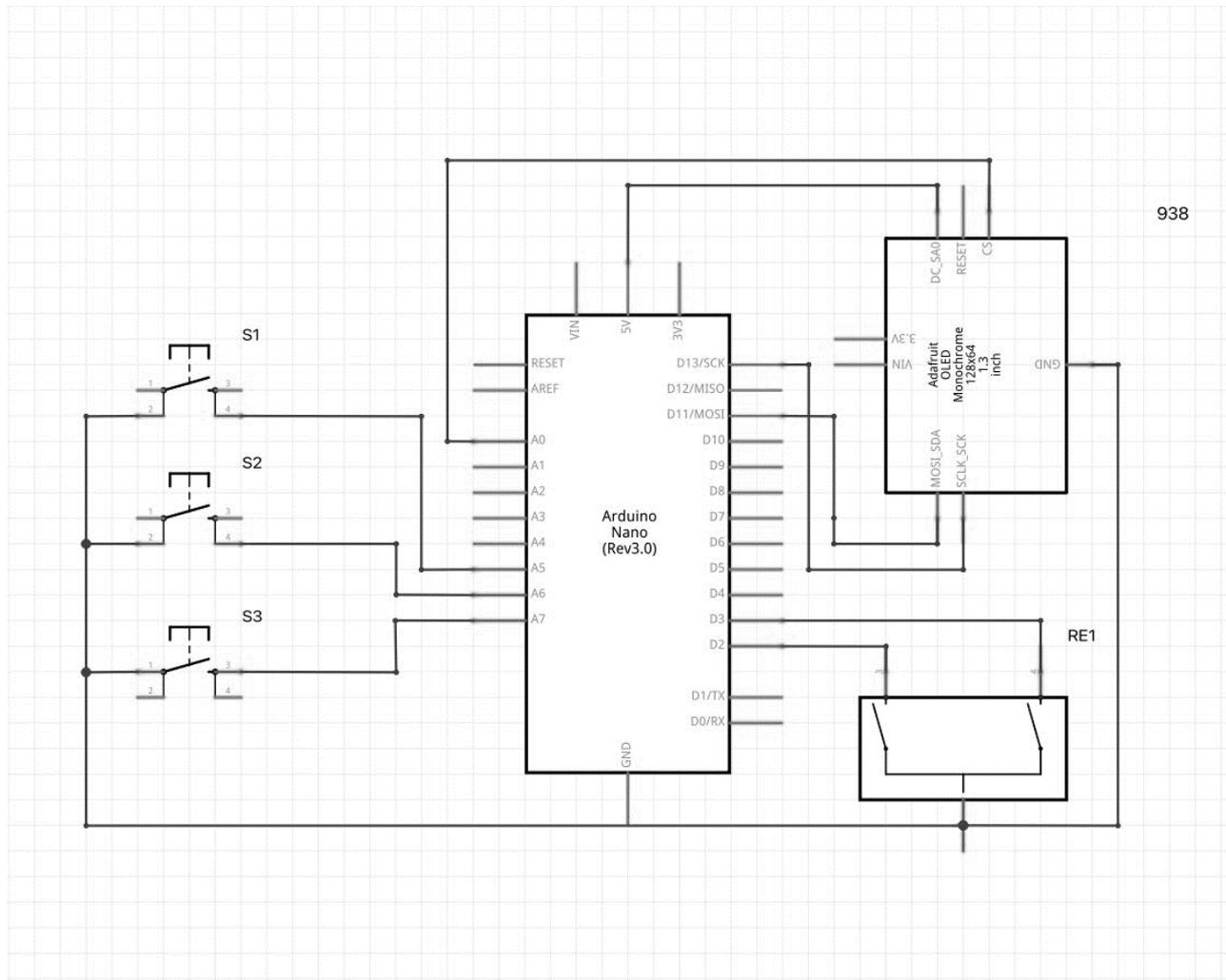


Рисунок Г.1 — Схема електрична принципова

ДОДАТОК Д

Лістинг коду програми

```

#include <SPI.h>
#include <U8g2lib.h>
#include <string.h>
#include <EEPROM.h>
#define BTN_GND_PIN A0
#define BTN1_PIN A3
#define BTN2_PIN A2
#define BTN3_PIN A1
#define LCD_DC_PIN A4
#define LCD_RES_PIN A5
const char s[38] = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't',
'u', 'v', 'w', 'x', 'y', 'z', '_', ' ', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' };
static const unsigned char logo[] U8X8_PROGMEM = {
  0xe0, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x03, 0x00, 0xf8, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff,
  0x0f, 0x00, 0xfc, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x1f, 0x00, 0xfe, 0xff, 0xff, 0xff,
  0xff,
  0xff, 0xff, 0x3f, 0x00, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3e, 0x00, 0x1f,
  0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x7c, 0x00, 0x0f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x78, 0x00, 0x0f,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0x00, 0x0f, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x78,
  0x00, 0x0f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0x00, 0x0f, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x78, 0x00, 0x0f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0x00, 0x0f,
  0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x78, 0x00, 0x0f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78,
  0x00, 0x0f, 0x00,
  0xfe, 0xff, 0xff, 0x7f, 0x00, 0x78, 0x00, 0x0f, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00,
  0x78, 0x00,
  0x0f, 0x80, 0xff, 0xff, 0xff, 0xff, 0x01, 0x78, 0x00, 0x0f, 0x80, 0xff, 0xff, 0xff,
  0xff, 0x01,
  0x78, 0x00, 0x0f, 0x80, 0xff, 0xff, 0xff, 0xff, 0x01, 0x00, 0x00, 0x0f, 0x80, 0xff,
  0xff, 0xff,
  0xff, 0x01, 0x00, 0x00, 0x0f, 0x80, 0xff, 0xff, 0xff, 0xff, 0x01, 0x00, 0x00, 0x0f,
  0x80, 0xff,
  0xff, 0xff, 0xff, 0x01, 0x00, 0x00, 0x0f, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
  0x00, 0x0f,
  0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x78, 0x00, 0x0f, 0x00, 0x00, 0xf8, 0x1f, 0x00,
  0x00, 0x78,

```



```

};
ButtonT<BTN1_PIN> btn1(INPUT_PULLUP, LOW);
ButtonT<BTN2_PIN> btn2(INPUT_PULLUP, LOW);
ButtonT<BTN3_PIN> btn3(INPUT_PULLUP, LOW);
EncoderT<2, 3> enc(INPUT_PULLUP);
GTimer blinkTm(MS, 400);
U8G2_SSD1306_128X64_VCOMH0_1_4W_HW_SPI u8g2(U8G2_R0,
U8X8_PIN_NONE, LCD_DC_PIN, LCD_RES_PIN);
struct Save {
  byte num;
  byte digs;
  float koef;
  String name;
  bool isValid;
};
class Program {
public:
  void init() {
    pinMode(BTN_GND_PIN, OUTPUT);
    digitalWrite(BTN_GND_PIN, LOW);
    u8g2.begin();
    u8g2.setContrast(255);
    u8g2.setFontPosTop();
    while (millis() < 3500) {
      u8g2.firstPage();
      do {
        u8g2.drawXBMP(29, 0, 70, 64, logo);
      } while (u8g2.nextPage());
      delay(100);
    }
    attachInterrupt(0, isr, CHANGE);
    attachInterrupt(1, isr, CHANGE);
    enc.setEncType(EB_STEP1);
    enc.setEncISR(true);
    menuPage = MENU_PAGE::MAIN;
    redraw = false;
    counter = 0;
    pause = false;
    blink = false;
    currSave = 0;
    menuSel = 0;
    menuBound = 0;
    savesCnt = 15;
    saves = new Save[savesCnt];
  }
};

```

```

tempSave.name = "          ";
quickSave.name = "Quick save";
newNameIndex = 0;
for (int i = 0; i < savesCnt; i++) {
    saves[i].name = "step/";
    saves[i].name += i + 1;
    saves[i].koef = i + 1;
    saves[i].digs = 1;
}
saves[0].digs = 0;
//EEPROM.put(0, Save{"sm_s24u_ln"});
//EEPROM.get(0,ys);
}
void update() {
    btn1.tick();
    btn2.tick();
    btn3.tick();
    enc.tick();
    if (blinkTm.isReady()) {
        blink = !blink;
        redraw = false;
    }
    switch (menuPage) {
    case MENU_PAGE::MAIN:
        if (btn1.click()) {
            counter = 0;
            enc.counter = 0;
            pause = false;
            redraw = false;
        }
        if (btn2.click()) {
            menuPage = MENU_PAGE::SAVES;
            pause = false;
            counter = 0;
            enc.counter = 0;
            menuSel = 0;
            menuBound = 0;
            redraw = false;
        }
        if (btn3.click()) {
            pause = !pause;
            if (!pause)
                enc.counter = counter;
            redraw = false;
        }
    }
}

```

```

}
if (!pause && counter != enc.counter) {
    counter = enc.counter;
    redraw = false;
}
break;
case MENU_PAGE::SAVES:
if (btn1.click()) {
    menuSel--;
    enc.counter = 0;
    redraw = false;
}
if (btn3.click()) {
    menuSel++;
    enc.counter = 0;
    redraw = false;
}
if (abs(enc.counter) > 10) {
    menuSel += enc.counter / 10;
    enc.counter = (enc.counter % 10);
    redraw = false;
}
menuSel = constrain(menuSel, 0, savesCnt + 2);
if (menuSel < menuBound)
    menuBound = menuSel;
if (menuSel > menuBound + 4)
    menuBound = menuSel - 4;
if (btn2.click()) {
    switch (menuSel) {
        case 0:
            menuPage = MENU_PAGE::MAIN;
            counter = 0;
            enc.counter = 0;
            break;
        case 1:
            menuPage = MENU_PAGE::NEW_SAVE_NAME;
            counter = 0;
            enc.counter = 0;
            break;
        case 2:
            menuPage = MENU_PAGE::QUICK_SAVE_DIST;
            counter = 0;
            enc.counter = 0;
            break;

```

```

    default:
        menuPage = MENU_PAGE::MAIN;
        counter = 0;
        enc.counter = 0;
        currSave = menuSel - 3;
        break;
    }
    redraw = false;
}
break;
case MENU_PAGE::NEW_SAVE_NAME:
    if (btn1.click()) {
        counter--;
        redraw = false;
    }
    if (btn2.click()) {
        newNameIndex++;
        redraw = false;
    }
    if (btn3.click()) {
        counter++;
        redraw = false;
    }
    if (abs(enc.counter) > 10) {
        counter -= enc.counter / 10;
        enc.counter = (enc.counter % 10);
        redraw = false;
    }
    if (counter > 37)
        counter -= 38;
    if (counter < 0)
        counter += 38;
    break;
case MENU_PAGE::QUICK_SAVE_DIST:
    if (btn1.click()) {
        counter = 0;
        enc.counter = 0;
        pause = false;
        redraw = false;
    }
    if (btn2.click() && abs(counter) > 0) {
        menuPage = MENU_PAGE::QUICK_SAVE_COUNT;
        quickSave.koef = abs(counter);
        pause = false;

```

```

    counter = 0;
    enc.counter = 0;
    redraw = false;
}

if (btn3.click()) {
    pause = !pause;
    if (!pause)
        enc.counter = counter;
    redraw = false;
}
if (!pause && counter != enc.counter) {
    counter = enc.counter;
    redraw = false;
}
break;
case MENU_PAGE::QUICK_SAVE_COUNT:
if (btn1.click()) {
    counter--;
    enc.counter = 0;
    redraw = false;
}
if (btn3.click()) {
    counter++;
    enc.counter = 0;
    redraw = false;
}
/*
if (abs(enc.counter) > 10) {
    counter += enc.counter / 10;
    enc.counter = (enc.counter % 10);
    redraw = false;
}
*/
/*
if (enc.counter != 0) {
    counter += enc.counter * 10;
    enc.counter = 0;
    redraw = false;
}
*/
if (abs(enc.counter) > 3) {
    counter += (enc.counter / 3) * 10;
    enc.counter = enc.counter % 3;
}

```

```

    redraw = false;
}
if (btn2.click() && abs(counter) > 0) {
    menuPage = MENU_PAGE::QUICK_SAVE_ACCR;
    quickSave.koef /= abs(counter);
    //quickSave.koef = 1.f / quickSave.koef;
    counter = 2;
    enc.counter = 0;
    redraw = false;
}
break;
case MENU_PAGE::QUICK_SAVE_ACCR:
    if (btn1.click()) {
        counter++;
        enc.counter = 0;
        redraw = false;
    }
    if (btn3.click()) {
        counter--;
        enc.counter = 0;
        redraw = false;
    }
    if (abs(enc.counter) > 20) {
        counter += enc.counter / 20;
        enc.counter = (enc.counter % 20);
        redraw = false;
    }
    counter = constrain(counter, 0, 4);
    if (btn2.click()) {
        menuPage = MENU_PAGE::MAIN;
        quickSave.digs = counter;
        currSave = -1;
        counter = 0;
        enc.counter = 0;
        redraw = false;
    }
    break;
}
}
void draw() {
    //u8g2.drawFrame(0, 0, 128, 64);
    switch (menuPage) {
    case MENU_PAGE::MAIN:
        {

```

```

if (!pause || blink) {
    if (currSave == -1) {
        drawValue(0, counter / quickSave.koef, quickSave.digs);
    } else
        drawValue(0, counter / saves[currSave].koef, saves[currSave].digs);
}
u8g2.setFont(u8g2_font_fur11_tf);
u8g2.setCursor(0, 47);
if (currSave == -1) {
    u8g2.print(quickSave.name + "(");
    u8g2.print(quickSave.koef);
    u8g2.print(")");
} else
    u8g2.print(saves[currSave].name);
break;
}
case MENU_PAGE::SAVES:
{
    u8g2.setFont(u8g2_font_fur11_tf);
    for (int i = menuBound; i <= menuBound + 4; i++) {
        if (i == menuSel) {
            if (blink)
                u8g2.drawCircle(4, 7 + (i - menuBound) * 13, 2);
            else
                u8g2.drawDisc(4, 7 + (i - menuBound) * 13, 2);
        }
        u8g2.setCursor(8, (i - menuBound) * 13);
    }
    switch (i) {
        case 0:
            u8g2.print("Back");
            break;
        case 1:
            u8g2.print("New save");
            break;
        case 2:
            u8g2.print("Quick save");
            break;
        default:
            u8g2.print(saves[i - 3].name);
            break;
    }
}
break;
}

```

```

case MENU_PAGE::NEW_SAVE_NAME:
{
    //u8g2.drawFrame(0, 18, 128, 42);
    u8g2.setFont(u8g2_font_fur11_tf);
    u8g2.drawStr(0, 0, "Set save name:");
    int f = 0;
    if (newNameIndex > 0)
        f = u8g2.getStrWidth(tempSave.name.substring(0, newNameIndex).c_str());
    tempSave.name[newNameIndex] = s[counter];
    u8g2.setCursor(f, 20);
    u8g2.print(counter == 0 ? s[37] : s[counter - 1]);
    u8g2.setCursor(0, 35);
    u8g2.print(tempSave.name);
    u8g2.setCursor(f, 50);
    u8g2.print(counter == 37 ? s[0] : s[counter + 1]);
    break;
}
case MENU_PAGE::QUICK_SAVE_DIST:
{
    u8g2.setFont(u8g2_font_fur11_tf);
    u8g2.drawStr(0, 0, "Set distance:");
    if (!pause || blink)
        drawValue(16, abs(counter), 0);
    break;
}
case MENU_PAGE::QUICK_SAVE_COUNT:
{
    u8g2.setFont(u8g2_font_fur11_tf);
    u8g2.drawStr(0, 0, "Set count:");
    drawValue(16, abs(counter), 0);
    break;
}
case MENU_PAGE::QUICK_SAVE_ACCR:
{
    u8g2.setFont(u8g2_font_fur11_tf);
    u8g2.drawStr(0, 0, "Set accuracy:");
    drawValue(16, 0.1234, counter);
    break;
}
}
}
bool redraw;
private:
void drawValue(byte pos, double val, byte digits) {

```

```

if (val < 0) {
  u8g2.drawBox(3, 20 + pos, 10, 3);
  val *= -1;
}
u8g2.setFont(u8g2_font_fur30_tn);
u8g2.setCursor(13, 6 + pos);
u8g2.print((int32_t)val);
if ((int32_t)val < 10000 && digits > 0) {
  int l = String((int32_t)val).length();
  u8g2.drawDisc(15 + 22 * l, 35 + pos, 2);
  u8g2.setCursor(17 + 22 * l, 6 + pos);
  u8g2.println(String(val, digits).substring(l + 1, l + 1 + digits));
}
}
static void isr() {
  enc.tickISR();
}
enum class MENU_PAGE {
  MAIN,
  SAVES,
  NEW_SAVE_NAME,
  QUICK_SAVE_DIST,
  QUICK_SAVE_COUNT,
  QUICK_SAVE_ACCR
};
MENU_PAGE menuPage;
int32_t counter;
bool pause;
bool blink;
Save *saves;
Save tempSave;
Save quickSave;
int menuSel;
int menuBound;
int currSave;
byte savesCnt;
byte newNameIndex;
};
Program prog;
void setup() {
  //Serial.begin(9600);
  prog.init();
}
void loop() {

```

```
prog.update();
if (!prog.redraw) {
  u8g2.firstPage();
  do {
    prog.draw();
  } while (u8g2.nextPage());
  prog.redraw = true;
}
}
```