

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра автоматизації та комп'ютерно-інтегрованих технологій та  
робототехніки

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Автоматизована система керування свердлильним верстатом

Назва теми

КвРАКІТ.2021031.01.04

Галузь знань 15 «Автоматизація та приладобудування»

Шифр, назва

Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»

Шифр, назва

Освітня програма «Автоматизація та комп'ютерно-інтегровані технології»

Назва

Виконав:

студент IV курсу, група АКІТ-21-1



Підпис

Наталія ГУРНАК

Ім'я, ПРІЗВИЩЕ

Керівник



Підпис, дата

Юрій ФОРКУН

Ім'я, ПРІЗВИЩЕ

Нормоконтролер



Підпис, дата

Людмила КОРЕЦЬКА

Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

зав. кафедри автоматизації  
та комп'ютерно-інтегрованих  
технологій та робототехніки



Підпис, дата

Валерій МАРТИНЮК

Ім'я, ПРІЗВИЩЕ

« 17 » червня 2025 р.

Хмельницький 2025

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

Рівень вищої освіти перший (бакалаврський)

Галузь знань 15 – Автоматизація та приладобудування

Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітня програма Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ

Завідувач кафедри АКИТтаР

Валерій МАРТИНЮК

07 лютого 2025р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Гурнак Наталі Володимирівні

Прізвище, ім'я, по батькові студента

1 Тема роботи Автоматизована система керування свердлильним верстатом

Керівник роботи канд.техн.наук., доцент Форкун Юрій Вікторович

Прізвище, ім'я, по батькові, науковий ступінь, учене звання

Затверджено наказом ректора університету від 07.02.2025 р. №23

2 Строк подання студентом роботи на кафедру 02.06.2025р.

3 Вихідні дані до роботи Завдання на кваліфікаційну роботу





4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Вступ. Огляд методів розв'язання поставленої задачі. Розробка схемотехнічних рішень. Розробка алгоритму роботи програмного забезпечення. Висновки

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

презентаційні матеріали (слайди)

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Антиплагіат	Федула М.В., доцент кафедри АКИТтаР		
Нормоконтроль	Корецька Л.О., доцент кафедри АКИТтаР		

7 Дата видачі завдання 07 лютого 2025р.

## КАЛЕНДАРНИЙ ПЛАН

Назва розділу кваліфікаційної роботи	Строк виконання	Примітка
1 Вибір та затвердження теми кваліфікаційної роботи; розробка завдання на кваліфікаційну роботу; складання календарного графіка виконання кваліфікаційної роботи	01.03.2025	Виконано
2 Вивчення предметної області, в якій планується використання системи автоматизації; аналіз вимог до системи автоматизації	15.03.2025	Виконано
3 Проектування та розробка загальної архітектури і структури системи автоматизації, інтерфейсу користувача; вибір засобів реалізації системи автоматизації	29.03.2025	Виконано
4 Програмна реалізація та тестування системи автоматизації	12.04.2025	Виконано
5 Написання тексту пояснювальної записки та розробка графічних матеріалів	19.04.2025	Виконано

Студент



Підпис

Наталія ГУРНАК

Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи



Підпис

Юрій Форкун

Ім'я, ПРІЗВИЩЕ

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Автоматизована система керування свердлильним верстатом».

Автор роботи: Гурнак Наталія Володимирівна

Керівник роботи: Форкун Юрій Вікторович

Пояснювальна записка: 65 с., 30 рис., \_ табл., 1 дод., 40 джерел.

Графічна частина: 8 презентаційних слайдів.

СИСТЕМИ КЕРУВАННЯ, СВЕРДЛУВАЛЬНИЙ ВЕРСТАТ, МІКРОКОНТРОЛЕРНА СИСТЕМА, БЛОК-СХЕМА АЛГОРИТМУ.

Метою роботи є розробка автоматизованої системи керування свердлильним верстатом.

У роботі було спроектовано автоматизовану керування свердлильним верстатом, що дозволяє автоматизувати процес свердлування друкованих плат із підвищеною точністю. Програмне забезпечення було написано для керування автоматичним свердлильним верстатом. Складено структурну схему та схему електричного підключення. Також розроблені блок-схеми для ручного та автоматичного режимів роботи, які покроково демонструють алгоритм роботи системи. Для роботи контролера була розроблена програма на основі мови програмування C/C++ для пристрою Arduino. Наведено експериментальні випробування та описано їх результати.



Підпис студента

16.06.25

Дата

## ЗМІСТ

ВСТУП .....	3
1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ .....	6
1.1 Основні поняття .....	6
1.2 Станція свердління CAMOZZI .....	16
1.3 Верстат з ЧПК Steerline .....	17
1.4 Особливості програмного забезпечення верстатів з ЧПК .....	18
1.5 Висновки до першому розділу .....	22
2 ПРОЕКТУВАННЯ ПРИСТРОЮ .....	23
2.1 Структурна схема .....	23
2.2 Схема електричних з'єднань .....	25
2.3 Алгоритм роботи програми контролера в ручному режимі .....	31
2.3.1 Алгоритм роботи програми контролера в ручному режимі .....	31
2.3.2 Створення керуючої програми для ручного режиму роботи .....	33
2.4 Висновки до другого розділу .....	44
3 РОЗРОБКА ПРОГРАМИ І КОНСТРУКЦІЇ .....	46
3.1 G-код і автоматичний режим .....	46
3.2 Блок-схема алгоритму роботи програми контролера в автоматичному режимі .....	47
3.2 Створення керуючої програми для автоматичного режиму роботи .....	49
3.3 Експериментальне тестування і налагодження програми .....	52
3.4 Висновки до третього розділу .....	58
ВИСНОВКИ .....	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....	61

КВРАКІТ.2021031.01.04 ПЗ								
Зм.	Лист	№ докum.	Підпис	Дата	Автоматизована система керування свердильним верстатом Пояснювальна записка	Літ.	Лист	Листів
Розроб.		Гурнак Н.В.		16.06.25			2	
Перевір.		Форкун Ю.В.		16.06.25				
Н. Контр.		Корецька Л.О.		16.06.25				
Затв.		Мартинюк В.В.		16.06.25				
						ХНУ, АКІТ-21-1		

## ВСТУП

У сучасному світі друковані плати є фізичною основою для електричних і механічних з'єднань електронних виробів. Майже всі компанії, які виробляють електронні вироби та електронні компоненти, потребують обробки друкованих плат, щоб додати переходи та отвори для кріплення. Оскільки для друкованих плат необхідна велика кількість отворів і високі вимоги до точності обробки, потрібен автоматичний свердлильний верстат.

Числове програмне керування (ЧПК) — це комп'ютерна система, що використовується для керування технічним обладнанням і приводами верстатів. Обладнання з ЧПК можна застосувати у наступному обладнанні:

– верстати, напр. для обробки металу (наприклад, фрезерування або токарна обробка), деревини, пластмас, різання листового металу, для пресування тощо (верстати з числовим програмним керуванням називаються верстатами з ЧПК);

- привід асинхронного двигуна з векторним керуванням;
- система керування, унікальна для сучасних промислових роботів.
- периферійне обладнання, таке як 3D-принтери, 3D-сканери.

Конструктивно до складу ЧПК можуть входити:

– пульт оператора (або консоль введення-виведення), дозволяє вводити керуючі програми та задавати режими роботи; виконує операції вручну. Як правило, у сучасній компактній шафі керування ЧПК містяться інші компоненти;

– дисплей (або панель управління) – для інтуїтивно зрозумілого керування режимами роботи та редагування керуючих програм/даних; може бути реалізований як окремий пристрій для дистанційного керування пристроєм;

– контролер - комп'ютеризований пристрій, призначений для вирішення завдань формування траєкторії ріжучого інструменту, управління технічними

					КвРАКІТ.2021031.01.04 ПЗ	3
		№ докум.	Підпис			

командами засобів автоматизації машин, поточного керування, редагування керуючих програм, діагностики та допоміжних розрахунків (траєкторії ріжучого інструменту, схеми різання);

– постійна пам'ять - це пам'ять, призначена для тривалого зберігання (роки і десятиліття) системних програм і констант; інформацію в ньому можна тільки прочитати;

– ОЗУ використовується для тимчасового зберігання керуючих і системних програм, які використовуються в даний момент.

Роль контролера беруть на себе промислові контролери, наприклад мікропроцесори для побудови вбудованих систем; програмовані логічні контролери або більш складні пристрої управління - промислові комп'ютери.

Важливою характеристикою контролера ЧПК є кількість осей (каналів), які він може синхронізувати (керувати) - для цього потрібна висока продуктивність і відповідне програмне забезпечення.

В якості виконавчих механізмів використовуються сервоприводи і крокові двигуни.

Промислові мережі (наприклад, CAN, Profibus, Industrial Ethernet) часто використовуються для передачі даних між приводами та системами керування машиною.

Кілька верстатів з ЧПК можуть бути об'єднані в гнучку автоматизовану виробничу систему, яка, у свою чергу, може бути доповнена гнучкими автоматизованими ділянками, щоб стати частиною автоматизованої виробничої лінії (польове або цехове виробництво) [1-3].

Верстати, обладнані пристроями ЧПК, називають верстатами з ЧПК. На додаток до металорізальних верстатів, таких як фрезерні або токарні верстати, існують також пристрої для різання заготовок з листового металу або виконання пресової обробки.

Крім основних функцій будь-якого свердлильного верстата, таких як обертання і вертикальне переміщення ріжучого інструменту, в ньому повинна

					КвРАКІТ.2021031.01.04 ПЗ	4
		№ докум.	Підпис			

бути реалізована система автоматизації, що значно підвищить ефективність і точність обробки. Для автоматизації багато виробників систем управління використовують G-код, оскільки він має жорстку структуру і використовується в більшості верстатів з ЧПК.

Метою роботи є розробка автоматизованої системи керування свердлильним верстатом.

Завданнями роботи є:

– спроектувати автоматизовану керування свердлильним верстатом, що дозволяє автоматизувати процес свердлування друкованих плат із підвищеною точністю.

– розробити програмне забезпечення для керування автоматичним свердлильним верстатом.

– скласти структурну схему та схему електричного підключення.

– провести експериментальні випробування.

					КВРАКІТ.2021031.01.04 ПЗ	
		№ докум.	Підпис			5

# 1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

## 1.1 Основні поняття

В даний час існує багато типів автоматизованих свердлильних верстатів, що використовуються для виробництва друкованих плат, з різними методами управління, компонентами, контролерами та програмним забезпеченням. Є також навчальні столи для розробки та складання різних типів маніпуляторів для виконання таких завдань, як захоплення та переміщення частин. Вони надають практичні навички проектування важливих схем і програмування контролерів.

Свердлильний верстат (англ. Drill press, drilling machine) — металорізальний верстат, призначений для свердління глухих і наскрізних отворів, розгортання, зенкерування, розгорткування, розточування та нарізування різьби. Основними формувальними рухами в операції свердління є головний (обертальний) рух  $v$  шпинделя верстата і рух подачі  $s$ . Кінематичний ланцюг, який виконує ці рухи, має окремі елементи регулювання  $i_v$  і  $i_s$ , за допомогою яких можна встановити необхідну частоту обертання інструменту і подачу.

Свердлильні верстати поділяються на верстати вертикально-свердлильні, настільні свердлильні верстати, колоно-свердлильні верстати, радіально-свердлильні верстати, верстати для глибокого свердління, центрувальні свердлильні верстати та багатошпиндельні свердлильні верстати. Настільні верстати застосовуються для свердління отворів у сталі ( $\sigma_v = 500 - 600$  МПа) до номінального діаметра 3, 6, 12 і 16 мм, а вертикальні і радіально-свердлильні верстати - для свердління отворів діаметром 18, 25, 35, 50 і 75 мм. Радіально-свердлильний верстат має діапазон свердління 1300 - 2000 мм.

					КвРАКІТ.2021031.01.04 ПЗ	6
		№ докум.	Підпис			

У вертикально-свердлильному верстаті основним рухом є обертання шпинделя разом із закріпленим у ньому інструментом, а подачею — вертикальним рухом шпинделя. Помістіть заготовку на стіл верстата або опорну плиту та перемістіть заготовку, щоб вирівняти отвір заготовки зі шпинделем.

Числове програмне керування (ЧПК) — це комп'ютерна система керування, яка зчитує командні інструкції зі спеціальної мови програмування (наприклад, G-коду) і керує приводами верстатів і верстатів для обробки металу, деревини або пластику.

Числове керування — це автоматичне керування процесом за допомогою обладнання з використанням числових даних, які зазвичай вводяться під час роботи [4-5].

Абревіатура CNC відповідає двом англійським словам - NC і CNC, що відображає еволюцію розвитку систем управління обладнанням:

Перші системи ЧПК (числового керування), які з'явилися, передбачали використання строго визначених схем керування обробкою — наприклад, налаштування програм за допомогою роз'ємів або перемикачів і збереження їх на зовнішньому носії. Для цих керуючих процесорів не передбачено працездатного пристрою зберігання даних;

Більш сучасні системи ЧПК називаються CNC (Computer Numerical Control) і базуються на наступних системах керування:

- мікроконтролер;
- програмований логічний контролер;
- мікропроцесорні промислові комп'ютери.

У ткацтві використовували змінні програми, надруковані на перфокартках з використанням двійкового коду Жакардовий ткацький верстат, винайдений у 1801 році.

Винахідником першого верстата з числовим керуванням (NC) був Джон Т. Парсонс, який працював інженером у компанії свого батька Parsons Inc., яка

виробляла вертолітні гвинти наприкінці Другої світової війни. Він першим запропонував обробляти гвинти за допомогою верстатів, які працювали за програмами, введеними з перфокарт.

У 1949 році ВПС США фінансували Парсонс для розробки верстата для контурного фрезерування частин літака складної форми. Однак компанія не змогла виконати роботу самостійно, тому звернулася за допомогою до лабораторії сервомеханіки МІТ. Співпраця Парсонса з Массачусетським технологічним інститутом тривала до 1950 року. У 1950 році Массачусетський технологічний інститут придбав компанію Hydro-Tel, що виробляла фрезерні машини, і припинив партнерство з Parsons, щоб укласти незалежний контракт з ВПС США на виробництво фрезерних верстатів з програмним керуванням.

У вересні 1952 року машина була вперше показана публіці – стаття про неї була опублікована в журналі Scientific American. Програма керування машиною записувалася на перфострічку.

Перші верстати з ЧПК були настільки складними, що їх не можна було використовувати у виробничих умовах. Перший серійний пристрій з ЧПК був розроблений Bendix у 1954 році та був встановлений на верстатах, починаючи з 1955 року. Широке просування верстатів з ЧПК було повільним. Підприємці скептично поставилися до нової технології. Міністерство оборони США було змушене виготовити 120 верстатів з ЧПК за власні кошти для оренди приватним компаніям.

Першими промисловими верстатами з ЧПК в СРСР були гвинтотокальний верстат 1К62ПУ і прокатний верстат 1541П. Ці машини були виготовлені в першій половині 1960-х років. Ці машини працюють у поєднанні з такими системами керування, як ПРС-3К. Згодом був розроблений вертикально-фрезерний верстат ЧПК 6Н13 з системою керування «Контур-ЗП». У наступні роки найбільш поширеними системами ЧПК для токарних верстатів в СРСР були ЧПК 2П22 і електронні системи НТС-31.

ЧПК також є типовою особливістю сучасних промислових систем управління роботами [4-5].

Абревіатура CNC відповідає двом англійським словам - NC і CNC, що відображає еволюцію розвитку систем управління обладнанням.

Перші системи ЧПК (числового програмного керування), що з'явилися, передбачали використання строго визначених схем управління процесом – наприклад, налаштування програм за допомогою розеток або перемикачів, збереження програм на зовнішніх носіях. Немає пристроїв зберігання даних з довільним доступом або керуючого процесора.

Більш сучасні системи ЧПК називають ЧПК (комп'ютерне числове керування), що є системою керування, яка дозволяє використовувати програмні засоби для модифікації існуючих програм і написання нових. Основою для побудови ЧПК є сучасні (мікро)контролери або (мікро)процесори:

- мікроконтролер,
- програмований логічний контролер,
- комп'ютер управління на основі мікропроцесора.

Можлива реалізація централізованої моделі автоматизованого робочого місця (наприклад, «ABB Robot Studio», «Microsoft Robotics Developer Studio») і передача завантажених програм через промислову мережу.

Свердлильний верстат з ЧПК — це автоматизований свердлильний пристрій, який може точно й послідовно свердлити отвори в різноманітних матеріалах, дотримуючись інструкцій комп'ютерної програми. Під час свердління вручну розташування та глибина отвору можуть бути не такими точними, як при використанні свердлильного верстата з ЧПК, який використовує складне програмне забезпечення для керування рухом і роботою свердла. Процес починається з проектування, де програмне забезпечення САД створює детальні креслення проекту, над яким працюється; потім ці проекти перетворюються на програми, зрозумілі програмному забезпеченню САМ, які,

у свою чергу, повідомляють машині, яку дію слід виконати на етапі виробництва. Система використовує координатні дані та спеціальні інструкції для свердління отворів за попередньо визначеними схемами, зменшуючи ймовірність людської помилки та покращуючи швидкість і якість виробництва.

#### Опис процесу буріння

Досягнення точності та ефективності свердлильного верстата з ЧПК вимагає кількох кроків. Спочатку матеріал, який потрібно свердлити, подається в машину і щільно затискається робочим столом. Після цього в його контролер завантажується програма ЧПК, що містить координатні дані та параметри буріння. Контролер обробляє програму і видає команди на рух машини.

Згідно з програмою, шпиндель переміщається в задане початкове положення, стартує з цього положення і утримує свердло. Потім у потрібному місці опустить свердло на поверхню заготовки, яку потрібно розрізати. Після підготовки він обертається з необхідною швидкістю, щоб він міг поступово свердлити на задану глибину в межах швидкості подачі різання; після завершення свердління охолоджуючу рідину також можна використовувати для запобігання перегріву або для очищення. Нарешті, поверніть шпиндель і перемістіть верстат до наступного отвору, який потрібно пробити згідно з кресленням, а потім повторіть ці дії, доки не буде пробито всі отвори.

Ці процедури зменшують людські помилки, підвищують точність розташування отвору та його глибину, а оскільки вони автоматизовані, також підвищують рівень узгодженості виробничого процесу.

					КвРАКІТ.2021031.01.04 ПЗ	10
		№ докум.	Підпис			

## Частини свердлильного верстата з ЧПК

Свердлильні верстати з ЧПК складаються з кількох основних компонентів, які працюють разом для досягнення точного й ефективного свердління.

Основа: станина є початковою точкою будь-якого верстата з ЧПК. Він забезпечує підтримку матеріалів та інших компонентів.

Шпindel: це обертовий пристрій, який утримує свердло та приводиться в рух двигуном для різання заготовки. Швидкість і рух шпинделя контролюються програмою ЧПК.

Панель керування: ця частина машини, у яку оператор вводить програми та контролює її діяльність під час роботи машини; він містить елементи керування, дисплеї, пристрої введення тощо.

Столи – це місце, де матеріал для свердління надійно закріплений; їх можна переміщати вздовж різних осей, щоб правильно розташувати матеріал відносно шпинделя.

Електричний блок керування складається з комп'ютерів та електронних пристроїв, які керують рухом і роботою відповідно до завантаженої програми через систему числового керування машини.

Система охолодження. Ця система використовує насос і форсунки для доставки охолоджуючої рідини в зону буріння, тим самим зменшуючи тепло, що виділяється під час процесу різання, а також запобігає передчасному зносу інструментів, які використовуються під час таких операцій.

Привідний двигун: механізм допоміжного приводу виконує різні рухи вздовж осі під час роботи, забезпечуючи тим самим точне позиціонування під час свердління та виконання інших завдань.

Зміна інструменту (необов'язково): заміна інструменту допомагає заощадити час на заміну різних свердел або інструментів, необхідних для застосування; це можна зробити автоматично на основі потреб, зазначених у

					КвРАКІТ.2021031.01.04 ПЗ	11
		№ докум.	Підпис			

впроваджуваній процедурі, тим самим підвищуючи ефективність і мінімізуючи час простою.

Це компоненти свердлильного верстата з ЧПК, але, загалом, вони пропонують високий рівень точності, а також високий рівень повторюваності, особливо під час процесу свердління.

Залежно від конструкції, функцій і призначення свердлильні верстати з ЧПК можна розділити на різні категорії:

1. Вертикальні свердлильні верстати з ЧПК: ці верстати монтуються на підлозі та мають велику робочу зону, що робить їх придатними для свердління важких і великих заготовок. Вони часто використовуються на виробництві, де здійснюється великомасштабне виробництво.

2. Настільні дрилі з ЧПК: настільні дрилі з ЧПК менші та компактніші за вертикальні дрилі, тому що вони призначені для встановлення на робочому столі. Вони використовуються для точного свердління невеликих заготовок і в основному використовуються в майстернях або лабораторіях.

3. Свердла з ЧПК із радіальними кронштейнами: ці типи машин відрізняються тим, що вони оснащені радіальними кронштейнами, які можна повертати, опускати або піднімати, що дозволяє розташувати свердло будь-де на великих заготовках. Цей верстат ефективний для свердління отворів у великих, важких матеріалах або матеріалах неправильної форми.

Усі ці типи свердлильних верстатів з ЧПК розроблені відповідно до конкретних робочих вимог, забезпечуючи універсальність і ефективність у широкому діапазоні промислових застосувань свердління.

За допомогою свердління з ЧПК відбувається процес збільшення наявного отвору з високим ступенем точності за допомогою машини, керованої комп'ютером. Метою цього методу є досягнення точних розмірів і гладкої поверхні, що робить його найкращим для застосування з жорсткими допусками. Під час цих операцій спеціалізовані розточувальні верстати з ЧПК використовують різні інструменти, такі як розточувальні головки та

розточувальні штанги, щоб забезпечити однаковість і точність протягом усього процесу. Як правило, вони широко використовуються в автомобільній промисловості, аерокосмічній промисловості, важкому машинобудуванні тощо.

Бічне свердління з ЧПК передбачає свердління отворів збоку заготовки, а не вертикально. Цей метод необхідний для свердління отворів у певних місцях, до яких неможливо легко дістатися традиційними методами вертикального буріння. Деревообробникам потрібна гнучкість, яку пропонують верстати з ЧПК, оскільки вони здатні ефективно виконувати бокове свердління, що дуже важливо при виготовленні меблів і деяких методах обробки металу. Використовуючи розширене програмне забезпечення для бічного свердління з ЧПК, отвори можна точно позиціонувати на кількох заготовках, забезпечуючи незмінну якість.

#### Популярні свердлильні та фрезерні верстати

Вивчаючи ріжучі та фрезерні верстати з хорошою репутацією, важливо вибрати моделі, які підходять для різноманітних галузей промисловості та використання. Ось деякі з найкращих машин:

Свердлоформувальна машина Jet JMD-18 350018: з двигуном потужністю 2 к.с., великим робочим столом і зручним контролем глибини ця потужна машина більш ніж здатна впоратися з будь-якою роботою. Його можна використовувати для свердління або фрезерування, що робить його популярним в автомобільних, деревообробних і металообробних цехах.

Свердлильна фреза Grizzly G0704 із підставкою: Grizzly G0704 відома своєю низькою ціною та високою точністю. Він має штифт типу «ластівчин хвіст», голівку тонкої подачі та цифрове зчитування, які допомагають вам щоразу отримувати точні вимірювання. Ця машина ідеально підходить для любителів, яким потрібна надійна продуктивність за низьку вартість у невеликій студії.

					КвРАКІТ.2021031.01.04 ПЗ	13
		№ докум.	Підпис			

Proxxon 37110 Micro Mill MF 70: якщо вам потрібна точність, то ця модель Proxxon саме для вас – вона широко вважається однією з найточніших і найміцніших доступних мікроточильних машин! Її змінна швидкість (5 000–20 000 об/хв) і компактний розмір роблять його ідеальним для тонкої роботи з невеликими, складними деталями, такими як ті, що використовуються у виробництві електроніки чи ювелірній промисловості.

Свердлильні верстати з ЧПК є унікальними серед інших типів обладнання, що використовується для свердління отворів, оскільки вони забезпечують високу точність і зменшують відходи. По-перше, верстати з ЧПК підвищують точність і забезпечують узгодженість виробничого процесу завдяки автоматизації, тим самим зменшуючи час і зусилля, витрачені на ручне налаштування. Згідно з результатами нещодавнього опитування, деякі з основних переваг є такими:

1. Підвищена точність: автоматизоване проектування (CAD) дозволяє свердлильним верстатам з ЧПК неодноразово виконувати точні шаблони свердління без помилок.

2. Вища продуктивність: ці машини можуть безперервно працювати на вищих швидкостях, значно збільшуючи швидкість виробництва продукції. Вони також можуть виконувати складні завдання з буріння швидше, ніж будь-яка людина, використовуючи ручні методи.

3. Зменшення відходів: свердла з ЧПК використовують менше матеріалу завдяки своїй високій точності, таким чином заощаджуючи витрати на сировину.

4. Безпечніше: коли сівалка з ЧПК працює на комп'ютері, людям, які беруть участь у процесі, не потрібно наближатися до сівалки надто близько. Це зменшує кількість нещасних випадків, пов'язаних з буровими операціями, і підвищує безпеку на робочому місці.

5. Гнучкість: промислове застосування може вимагати, щоб свердлильний верстат мав різні можливості за короткий термін; тут стане в

					КвРАКІТ.2021031.01.04 ПЗ	14
		№ докум.	Підпис			

нагоді перепрограмування дрилі за допомогою ЧПК, оскільки це можна зробити швидко.

Для виробників інвестиції в технологію свердління з ЧПК можуть заощадити багато грошей. По-перше, точність і точність є одними з переваг використання свердлильного верстата з ЧПК. Це зменшує кількість відходів і знижує ціну на сировину. Провідні галузеві веб-сайти стверджують, що такий рівень точності гарантує, що кожен компонент майже ідеальний, тому його ніколи не потрібно ремонтувати чи викидати.

По-друге, витрати на робочу силу можна зменшити завдяки функціям автоматизації, які пропонують ці типи сівалок. Ці машини здатні працювати безперервно протягом тривалого періоду часу з мінімальним наглядом, значно зменшуючи людський вхід, необхідний для роботи. Окрім економії на оплаті праці, це також може звільнити працівників, які раніше працювали тут, для участі в інших частинах виробництва, тим самим максимізуючи ефективність працівників.

Нарешті, швидший час виконання проекту пояснюється підвищеною ефективністю використання свердел з ЧПК, що призводить до більшої продуктивності та швидшого виробництва. У результаті виробники можуть взяти на себе більше роботи протягом певного періоду часу. Це призведе до збільшення прибутку в цілому. Вони також економлять час, коли одна особа має багато різних завдань, тому що їм не потрібно чекати, поки закінчить інша; вони можуть просто швидко перемикатися без втрати живлення, що дозволяє їм виконувати операції економічно ефективно. З огляду на інші фактори, пов'язані з сучасним виробництвом, ці свердлильні машини дуже дешеві.

					КвРАКІТ.2021031.01.04 ПЗ	15
		№ докум.	Підпис			

## 1.2 Станція свердління CAMOZZI

Бурова станція входить до складу навчального стенду "Автоматизований виробничий осередок ДІД-АПЛ". На рисунку 1.1 показана навчальна станція окремо від навчального столу.

- бурова станція складається з наступних частин:
- електричний шпиндель;
- пневматичні лещата;
- чотири циліндра;
- два моностабільних клапанних розподільника;
- бістабільний розподільник;
- чотири магнітних датчика положення циліндра;
- реле тиску.



Рисунок 1.1 – Станція свердління Camozzi



Верстат має хід шпинделя осі Z 100 мм. Тип передачі по осях X, Y і Z здійснюється кульковими парами за допомогою ходових гвинтів 16 мм з кроком 5 мм. Для руху використовується кроковий двигун 57HS. Машина також містить датчик довжини інструменту та індуктивний датчик для визначення початку нульової точки машини. Потужність шпинделя 0,8 кВт. Габаритні розміри машини 620x500x560 мм, вага 58 кг.

#### 1.4 Особливості програмного забезпечення верстатів з ЧПК

Числове керування (ЧПК) — це комп'ютерна система, що використовується для керування технічним обладнанням і приводами верстатів. Обладнання з ЧПК складається:

- верстати, напр. для обробки металу (наприклад, фрезерування або токарна обробка), деревини, пластмас, різання листового металу, для пресування тощо (верстати з числовим програмним керуванням називаються верстатами з ЧПК);

- привід асинхронного двигуна з векторним керуванням;
- система керування, унікальна для сучасних промислових роботів.
- периферійне обладнання, таке як 3D-принтери, 3D-сканери.
- поруч знаходиться сучасна шафа системи керування для двох промислових роботів FANUC R200iB.

Конструктивно до складу ЧПК можуть входити:

- пульт оператора (або консоль введення-виведення), дозволяє вводити керуючі програми та задавати режими роботи; виконує операції вручну. Як правило, у сучасній компактній шафі керування ЧПК містяться інші компоненти;

- дисплей (або панель управління) – для інтуїтивно зрозумілого керування режимами роботи та редагування керуючих програм/даних; може

бути реалізований як окремий пристрій для дистанційного керування пристроєм;

– контролер - комп'ютеризований пристрій, призначений для вирішення завдань формування траєкторії ріжучого інструменту, управління технічними командами засобів автоматизації машин, поточного керування, редагування керуючих програм, діагностики та допоміжних розрахунків (траєкторії ріжучого інструменту, схеми різання);

– постійна пам'ять - це пам'ять, призначена для тривалого зберігання (роки і десятиліття) системних програм і констант; інформацію в ньому можна тільки прочитати;

– ОЗУ використовується для тимчасового зберігання керуючих і системних програм, які використовуються в даний момент.

Роль контролера беруть на себе промислові контролери, наприклад мікропроцесори для побудови вбудованих систем; програмовані логічні контролери або більш складні пристрої управління - промислові комп'ютери.

Важливою характеристикою контролера ЧПК є кількість осей (каналів), які він може синхронізувати (керувати) - для цього потрібна висока продуктивність і відповідне програмне забезпечення.

В якості виконавчих механізмів використовуються сервоприводи і крокові двигуни.

Промислові мережі (наприклад, CAN, Profibus, Industrial Ethernet) часто використовуються для передачі даних між приводами та системами керування машиною.

Кілька верстатів з ЧПК можуть бути об'єднані в гнучку автоматизовану виробничу систему, яка, у свою чергу, може бути доповнена гнучкими автоматизованими ділянками, щоб стати частиною автоматизованої виробничої лінії (польове або цехове виробництво) [6-8 ].

Верстати, обладнані пристроями ЧПК, називають верстатами з ЧПК. На додаток до металорізальних верстатів, таких як фрезерні або токарні верстати,

існують також пристрої для різання заготовок з листового металу або виконання пресової обробки.

#### Програмне забезпечення

Найпоширеніша мова програмування ЧПК для металорізального обладнання описана в документі ISO 6983 Комітету з міжнародних стандартів і називається «G-код» [9-10]. У деяких випадках, як-от системи керування гравірувальними машинами, мова керування принципово відрізняється від стандартної. Для простих завдань, таких як різання плоских заготовок, системи ЧПК можуть використовувати як вхідну інформацію текстові файли у форматі обміну даними, наприклад DXF або HP-GL.

Після того, як керуюча програма складена, оператор за допомогою програматора вводить її в контролер. Команди керуючої програми розміщуються в оперативній пам'яті. У процесі створення або після входу в керуючу програму оператор (при цьому він виконує роль програміста) може редагувати її, включивши в роботу системну програму редактора і відобразивши всі або необхідні частини керуючої програми і внісши в неї необхідні зміни. При роботі в режимі виготовлення деталей керуюча програма виконується кадрово. За командою керуючої програми контролер викликає відповідну системну підпрограму з резидентної пам'яті, щоб примусити пристрій, підключений до ЧПК, працювати в потрібному порядку - результати роботи контролера надсилаються у вигляді електричних сигналів на виконавчий механізм - привід подачі або на пристрій автоматизації керування верстатом.

Система управління зчитує інструкції програми на спеціалізованій мові програмування (наприклад, G-код), а інтерпретатор системи ЧПК потім перекладає мову введення в команди для керування головними приводами, приводами подачі та контролерами, які керують компонентами агрегату машини (наприклад, вмикання/вимкнення подачі охолоджуючої емульсії).

Розробка програм керування на початку 21-ого сторіччя виконувалася за допомогою систем автоматизованого проектування (САПР) або виділених модулів окремих систем автоматизованого програмування (САМ), які генерували програми обробки на основі електронних моделей.

Для визначення необхідної траєкторії руху всього робочого органу (інструмента/заготовки) за керуючою програмою використовується інтерполятор для розрахунку положення проміжних точок траєкторії за кінцевими точками, заданими в програмі.

Крім самої програми, система управління також містить дані в інших форматах і за призначенням. Як мінімум, це машинні дані та дані користувача, які стосуються конкретної системи керування або певної серії (лінійки) подібних моделей систем керування.

Програма для верстата (пристрою) з ЧПК може бути завантажена у власну пам'ять із зовнішнього носія (наприклад, магнітної стрічки, перфострічки, дискети або флешки) або тимчасово завантажена в оперативну пам'ять до вимкнення живлення. Якщо вміст енергонезалежної пам'яті (наприклад, флеш-пам'яті, жорсткого диска чи SSD-накопичувача) оновлюється, програма зберігається навіть після вимкнення живлення. Крім того, сучасне обладнання підключається до централізованих систем управління через заводські (цехові) комунікаційні мережі.

Документ ISO 6983 від Комітету з міжнародних стандартів описує найпоширенішу мову програмування ЧПК для металорізального обладнання, відому як «G-код». У деяких випадках, як-от системи керування гравірувальними машинами, мова керування принципово відрізняється від стандартної. Для простих завдань, таких як різання плоских заготовок, система ЧПК може використовувати як вхідні текстові файли у форматі обміну даними - наприклад, DXF або HPGL.

Інтерпретатор системи ЧПК перетворює програму на мові введення в команди для керування головними приводами, приводами подачі та

контролерами, які керують компонентами машини (вмикання/вимкнення охолодження, швидкість руху тощо). Для визначення необхідної траєкторії руху всього робочого органу (інструмента/заготовки) за керуючою програмою використовується інтерполятор для розрахунку положення проміжних точок траєкторії за координатами точок, заданих у програмі.

### 1.5 Висновки до першому розділу

У першому розділі проведено огляд існуючих рішень. Визначено основні поняття станків з числовим програмним керування та їх застосування для побудови автоматизованих систем свердлування друкованих плат. Наведено огляд деяких сучасних станків з ЧПК: станція свердління CAMOZZI та верстат з ЧПК Steepline.

Розглянуті особливості програмного забезпечення верстатів з ЧПК.

					КВРАКІТ.2021031.01.04 ПЗ	
		№ докум.	Підпис			22

## 2 ПРОЕКТУВАННЯ ПРИСТРОЮ

### 2.1 Структурна схема

Блок-схема — це діаграма, яка визначає основні функціональні частини продукту, їхні взаємозв'язки та використання [11, 12]. Під функціональними компонентами розуміють складові частини схеми: елементи, пристрої, функціональні групи, функціональні ланки .

Блок-схема використовується, щоб показати загальну структуру пристрою, тобто його основні блоки, вузли, компоненти та основні зв'язки між ними. Структурна схема повинна чітко пояснювати, для чого потрібен пристрій, як він працює в основних режимах роботи, як його частини взаємодіють між собою. Назви елементів структурної діаграми можна вибирати довільно, але повинні відповідати загальноприйнятим правилам виконання діаграм [13].

#### Структурна електрична схема

На конструктивному електротехнічному кресленні [14] всі основні частини виробу (вузли, пристрої, функціональні групи) зображують у вигляді прямокутників або традиційних графічних позначень і вказують зв'язки між ними. При цьому графічна побудова схеми повинна інтуїтивно відображати порядок взаємодії функціональних компонентів виробу і простежуватися за допомогою стрілок, нанесених на сполучні лінії.

Для прямокутного об'єкта введіть назву об'єкта, тип елемента та назву документа. Для великої кількості функцій дозволяється розміщувати серійний номер праворуч або над зображенням, зазвичай зверху вниз і зліва направо, замість назви, типу та назви. У цьому випадку назва, тип і позначення записуються в таблицю в поле діаграми.

У схемотехніці, крім структурного проектування, також необхідно враховувати принципіві та функціональні діаграми. З усіх цих схем

					КвРАКІТ.2021031.01.04 ПЗ	23
		№ докум.	Підпис			



подається на контролер. Контролер підключається до драйвера CNC Shield V3.0 і працює з кроковими двигунами для осей X, Y і Z. Для управління кроковим двигуном в ручному режимі потрібен перемикач, підключений безпосередньо до контролера. Також знадобиться перемикач режимів роботи, який підключається безпосередньо до контролера. Для керування свердлильною машиною за допомогою G-коду в автоматичному режимі необхідно підключити персональний комп'ютер до контролера за допомогою інтерфейсу USB. Кінцеві вимикачі використовуються для регулювання руху столу в координатах X і Y. Використовується кнопка, щоб запустити мотор шпинделя. Дані поточного режиму роботи і координати шпинделя над столом обробляються контролером і виводяться на дисплей.

## 2.2 Схема електричних з'єднань

На рисунку 2.2 показано електричні з'єднання автоматичного свердлильного верстата.

Опис принципової електричної схеми керування мобільним роботом-прибиральником.

Схема складається з п'яти основних частин:

1. Блок живлення — перетворює напругу мережі 220 В у стабілізовані напруги +12 В, +5 В та загальний «0 В».
2. Блок індикації — цифровий літерний індикатор (7-сегментний або матричний) зі своєю під-системою підключення до +5 В через резистор R1.
3. Пульти керування (ручний режим) — набір важелів і кнопок S1...S5, що дозволяють оператору вручну задавати напрямки руху, старт/стоп та режим роботи (ручний/автомат).
4. Обчислювач (Arduino Nano) — центральний елемент, який у автоматичному режимі формує сигнали управління двигунами та вентилятором.

5. Драйвери двигунів D1, D2, D3 — силові модулі на базі мікросхем типу L298 або аналогів, які по інтерфейсу підключені до Arduino та керують трьома незалежними моторами: два — для пересування, один — для механізму «свердління» або вентиляції.

Крім того, в схемі передбачені ручні та автоматичні стоп-контакти (S6...S9), а також кнопка аварійного вимкнення/стоп (S5 у центрі блоку «Пуск двигуна»).

Блок живлення

Вхід 220 В включає в себе трансформатор і випрямляч, що формують вихідні шини:

– +12 В (для силових модулів, вентилятора та «свердління»): маркування «+12 В»;

– +5 В (для електроніки — Arduino, індикатора, реле): маркування «+5 В»;

– 0 В (спільний «мінус»).

Живлення надходить одночасно на:

– контакти Vcc (+5 В) та Gnd (0 В) цифрового індикатора;

– шину +5 В до плати Arduino (контакти 18, 19);

– шину +12 В до мотор-драйверів та реле вентиляторів.

Блок цифрового літерного індикатора

Індикатор підключений чотирма контактами:

– DA (Data) — вхід даних (контакт 18 Arduino або інше джерело SPI-подібного сигналу);

– CL (Clock) — тактовий вхід (контакт 19);

– GND — «земля» (0 В);

– VCC — +5 В.

Між шиною +5 В та сегментом індикатора встановлено обмежувальний резистор  $R1 = 150 \Omega$ , який запобігає перевантаженню світлодіодних сегментів індикатора.

Індикатор призначений для виведення режиму роботи або стану системи (наприклад: час до завершення, статус «Auto/Man», чи помилки в роботі).

Пульт керування (ручний режим)

У верхньому правому куті схеми зображено панель ручного керування:

– S1 «Вліво-вправо» (тумблер) — задає напрямок обертання бокового двигуна D1: вліво або вправо.

– S2 «Вперед-Назад» — задає напрямок руху основних ведучих коліс через драйвер D2.

– S3 «Вгору-Вниз» — керування третім двигуном D3 (наприклад, механізм підйому/опускання щіток чи свердління).

– S4 «Ручн/Авт» — перемикач режимів: у положенні «Ручн» (контакт 8–9) всі команди подаються від ручних тумблерів, у «Авт» — від Arduino.

– S5 «Свердління/Пуск двигуна» — кнопка запуску: у ручному режимі запускає обраний механізм; в автоматичному — ініціює старт алгоритму.

Нижче, уздовж «Задньої кришки», розташовані стоп-кнопки безпеки:

– S6 «Стоп вперед» — розмикає лінію «вперед» (контакт 11).

– S7 «Стоп назад» — лінію «назад» (контакт 13).

– S8 «Стоп вліво» — лінію «вліво» (контакт 15).

– S9 «Стоп вправо» — лінію «вправо» (контакт 17).

У разі спрацьовування будь-якого стоп-контакту відповідний двигун не отримає живлення, що гарантує безпеку оператора та запобігає пошкоджуючим навантаженням.

Обчислювач (Arduino Nano)

					КвРАКІТ.2021031.01.04 ПЗ	27
		№ докум.	Підпис			

Плата Arduino Nano у схемі зображена як «Обчислювач». Піни розподілені наступним чином:

Контакти 1...8, 13...19 — цифрові входи/виходи, підключені до пульта керування та стоп-контактів (S1...S9, S5).

Контакти 20...31 — виходи керування драйверами двигунів:

- 20–23 → Драйвер D1 (управління двигуном вліво-вправо).
- 24–27 → Драйвер D2 (управління рухом вперед-назад).
- 28–31 → Драйвер D3 (управління підйомом/опусканням або свердлінням).
- +5 В та 0 В — живлення плати від шини блоку Живлення.
- +12 В (за потреби) може подаватися на датчики чи інші вузли.

У автоматичному режимі Arduino зчитує зі входів S4, S5 (старт, режим), а також з індикатора (якщо є зворотний зв'язок) і формує на виходах 20...31 послідовність сигналів «вперед/назад» або «ліворуч/праворуч» згідно із запрограмованим алгоритмом. У ручному режимі ці ж виходи дублюються тумблерами S1...S3, а Arduino «відключається» (не скидає сигнали на драйвери).

Драйвери двигунів D1, D2, D3

Кожен драйвер — це окремий блок (найчастіше L298N або подібний), що має:

- вхідні сигнали (IN1, IN2 тощо) від Arduino (контакти 20...31) для керування напрямком обертання двох каналів.
- живлення +12 В (шина з блоку живлення) та 0 В.
- виходи OUT1, OUT2 (для D1) → двигун M1;
- OUT3, OUT4 (для D2) → двигун M2;
- OUT5, OUT6 (для D3) → двигун M3.

Драйвери забезпечують подачу достатнього струму на обмотки двигунів, а також блокування одночасного включення обох полюсів Н-моста, захищаючи систему від короткого замикання.

#### Мотори

У схемі позначено чотири моторні механізми:

1. M1 (Двигун D1) — відповідає за маневрування вліво-вправо (рампа, петля тощо).
2. M2 (Двигун D2) — головний привід коліс — рух вперед-назад.
3. M3 (Двигун D3) — механізм підйому/опускання робочого органу (наприклад, щіток або свердла).
4. M4 (Мотор свердління) — позначено у блоці керування — працює від +12 В через окреме реле (VD або S5).

Мотори підключаються до відповідних виходів драйверів, які реалізують Н-мостову схему для реверсу та широтно-імпульсну модуляцію (PWM) швидкості обертання.

#### Система безпеки

Кнопки S6...S9 та S5 служать для екстреного та планового зупинення:

- S6...S9 — стоп-контакти по кожному напрямку.
- S5 («Пуск двигуна») — аварійний стоп/старт: у разі натискання перериває ланцюг живлення на драйвери та вентилятор.

В усіх випадках логіка безпеки спроектована так, щоб розмикання одного лише контакту миттєво блокувало подачу живлення на силовий модуль відповідного двигуна.

#### Режим «Ручний/Автомат»

Перемикач S4 керує джерелом сигналів управління:

- «Ручний» (контакт ручн) — виходи драйверів напряму з'єднані із тумблерами S1...S3. Arduino у цей момент не втручається.
- «Автомат» (контакт авт) — тумблери від'єднані, а сигнали надходять від плати Arduino (контакти 20...31).

Таким чином, одним рухом рукояті С4 оператор може переключити систему між повністю ручним керуванням і автономним виконанням програми.

#### Логіка роботи схеми

1. Увімкнення живлення (220 В → +5 В, +12 В).  
2. Індикація готовності: загоряється LED VD, на сегментному індикаторі можуть відобразитися початкові символи («rdu», «000»).

#### 3. Вибір режиму:

– ручний → оператор вибирає напрямок тумблерами S1...S3, натискає S5 → відповідний двигун починає працювати.

– автомат → Arduino після натискання S5 запускає алгоритм:

– подає через драйвери D1...D3 послідовність команд (вперед, вліво, вправо, поворот, зупинка) згідно з вбудованою програмою.

Моніторить тривалість роботи (таймер у програмі) й після закінчення заданого інтервалу подає на реле сигнал «стоп» вентилятору й двигунам.

4. Стоп-контакти (S6...S9, S5) можуть перервати будь-який рух в екстреному випадку.

5. Завершення циклу → повернення в початковий стан (індикатор «END», LED погас), очікування наступного запуску.

Представлена схема поєднує модульний підхід (окремі блоки живлення, індикації, керування, драйверів) із простими механізмами безпечної роботи (стоп-контакти, ручний/автомат). Використання Arduino Nano як обчислювача забезпечує гнучкість алгоритму в автоматичному режимі, а ручний пульт дозволяє виконувати додаткові маневри та обслуговування без перепрошивки. Завдяки такій архітектурі проєкт може бути легко масштабований: додаючи нові сенсори, розширюючи програмну частину чи замінюючи двигуни та драйвери на більш потужні [16-22].





вимкнення двигуна свердла; встановлюється параметр руху «STOP» і задається напрямок руху за координатами X та Y залежно від стану перемикачів. Виконується переміщення вліво або вправо, а потім — вперед або тому відповідно до порівняння заданої та поточної координат. Після завершення руху повертається до перевірки стану контролера або закінчення циклу.

У режимі «Автоматичний» здійснюється перевірка натискання кнопки «свердління», після чого встановлюється параметр руху «STOP». Отримується G-код через ПК і перевіряється його коректність: у разі некоректного введення здійснюється повторне отримання з відповідним повідомленням; у разі успішної перевірки переходиться до завершення.

Паралельно відстежуються кінцеві вимикачі та поточні координати для зупинки крокових двигунів при досягненні межі. Після завершення операцій здійснюється перехід у кінцевий стан з відображенням результатів і очікуванням нового запуску [23-25].

### 2.3.2 Створення керуючої програми для ручного режиму роботи

Для програмування використовується інтегроване середовище розробки Arduino - Arduino IDE.

Для відображення інформації використано чотирирядковий дисплей з модулем I2C. Це означає, що потрібно встановити бібліотеку LiquidCrystal\_I2C, яка повторює більшу частину функціональності стандартної бібліотеки LiquidCrystal. Також потрібно підключити стандартну бібліотеку Wire для взаємодії з пристроями I2C/TWI. Далі потрібно оголосити об'єкт бібліотеки з параметрами відображення: адреса шини I2C, кількість стовпців, кількість рядків. Як показано на рисунку 2.3.













По-перше, метод повинен приймати єдиний параметр, який буде поточним режимом роботи контролера. Після цього в методі `updateDirectionPin` задається напрямок обертання крокового двигуна відповідно до параметрів. Далі слід виконувати звичайний потік програми за допомогою покрокового методу, але оскільки ініціалізація відбувається автоматично, обмеження потрібно встановлювати за допомогою кінцевих вимикачів. Для цього використовується цикл `while`. Якщо це циклічна робота, показання кінцевого вимикача базуватимуться на поточному режимі роботи. Кінцевий вимикач визначається за допомогою виразу `LIMIT_SWITCH_LEFT` + режим – `MOVE_LEFT`.

Далі знадобиться спосіб відображення поточного положення шпинделя над столом, а також дані про поточний режим роботи. Це робиться за допомогою методу `displayData`. Оскільки дані про положення шпинделя необхідно оновити, першим кроком цього методу є очищення відображення старих даних і переміщення курсору в вказане положення. Стандартна бібліотека `LiquidCrystal` вже має функції `clear()` і `setCursor()` для цієї мети. Після виконання цих функцій старі дані на екрані будуть очищені, а курсор переміститься у верхній лівий кут. Поточне положення шпинделя над робочим столом відобразатиметься лише тоді, коли кроковий двигун припинить роботу. Під час переміщення на екрані має з'явитися повідомлення про поточний режим (ініціалізація або переміщення). Для цього використовується умова перевірки поточного режиму роботи, як показано на рисунку 2.12. Використовуйте функцію друку для відображення тексту на екрані з координатами шпинделя, коли умова виконується. В іншому випадку з'явиться повідомлення про те, що переміщення триває, і потрібно дочекатися його завершення [26-27].

Після того, як написані всі методи, необхідні для ручного режиму, можна приступати до заповнення основної програми. Потрібно почати з функції `setup()`. Ця функція запускається один раз під час запуску контролера.

Необхідно ініціалізувати змінні, визначити режим виведення та перемістити робочий стіл у вихідне положення. Фрагмент програмного коду з цією функцією показано на рисунку 2.13.

```

void displayData() //Метод displayData
{ //Початок методу
  lcd.clear(); //Функція для очищення дисплея
  lcd.setCursor(0, 0); //Позиціювання курсору на перший рядок

  if (mode == STOP) //Умова при режимі зупинки руху
  { //Початок блоку команд під час виконання умови
    lcd.print("X="); //Виведення тексту на дисплей
    lcd.print(machineToolX); //Виведення змінної положення, що відображається на осі X
    lcd.print(", Y="); //Виведення тексту на дисплей
    lcd.print(machineToolY); //Виведення змінної положення, що відображається на осі Y
  } //Кінець блоку команд під час виконання умови
  else //Інакше
  { //Початок блоку команд при невиконанні умови
    lcd.print("MOVING..."); //Виведення тексту на дисплей
    lcd.setCursor(0, 1); //Позиціювання курсору на другий рядок
    lcd.print("PLEASE WAIT..."); //Виведення тексту на дисплей
  } //Кінець блоку команд при невиконанні умови
} //Кінець методу

```

Рисунок 2.12 - Фрагмент коду програми з методом *displayData*

По-перше, необхідно оголосити службову змінну *i*, яка потрібна під час циклу виділення режиму роботи. Далі відбувається ініціалізація дисплею і ввімкнення підсвітки. Далі йдуть два цикли, де значенням у переліку присвоюються номери відповідних контактів. Таким чином, перший цикл встановлює елементи перерахування в режим OUTPUT і призначає їм наступні номери контактів:

- ENABLE - 8;
- X\_DIR - 5;
- Y\_DIR - 6;
- Z\_DIR - 7;
- X\_STEP - 2;
- Y\_STEP - 3;



використовується метод `doAction` з параметром `MOVE_LEFT`, робочий стіл буде переміщено в крайній лівий край, і, нарешті, використовується метод `doAction` з параметром `MOVE_BACKWARD`, щоб перемістити робочий стіл в самий крайній край. У результаті стіл переміщується в положення, де шпindel над ним знаходиться в координаті 0:0.

Після запуску контролера всі змінні ініціалізуються, режим виведення визначається, і робочий стіл переміщується у вихідне положення. Необхідно описати роботу функції `loop()` контролера. Ця функція нескінченно зациклюється, доки контролер не втрачає живлення або плата Arduino не перезавантажиться. Він призначений для активного керування платою Arduino. Функція `loop()` показана на рисунку 2.14.

По-перше, потрібно оголосити службову змінну для циклу читання даних. Цикл читання даних необхідний для того, щоб контролер отримував оновлені дані про стан перемикачів і кінцевих вимикачів кожного разу, коли оновлюється функція контуру. Оновлені дані буде замінені новим значенням у масиві `pinData`. Після цього оновить поточний режим роботи за допомогою методу `updateCurrentMode`.

Далі потрібно визначити, що має відбуватися при зміні режиму роботи або при припиненні руху.

Умова `if (oldMode != mode)` перевіряє поточний робочий режим і запускається, лише якщо метод `updateCurrentMode` оновив його. При спрацьовуванні попередньому режиму роботи присвоюється значення поточного режиму, дані на дисплеї оновлюються за допомогою методу `displayData`, а напрямок обертання крокового двигуна призначається відповідно до режиму роботи за допомогою методу `updateDirectionPin`.

Якщо поточний режим роботи дорівнює зупинці руху, спрацьовує умова `if (режим == STOP)`. Для режиму ручного керування зупинкою руху ви просто встановлюєте змінну часу, яка дорівнює часу, що минув з моменту введення умови, і порівнюєте її зі змінною `displayUpdateTimestamp`. Це потрібно для



Розроблено схему електричних з'єднань. Для її розробки проведено обрання елементів схеми за допомогою яких будуть реалізовані програмні та керуючі функції станка з ЧПК. В основі схеми закладено використання мікроконтролерної схеми, яка дозволяє реалізовувати алгоритм автоматичного за закладеною програмою керування свердлування друкованих плат.

Проведена розробка алгоритму роботи програми контролера в ручному режимі. Запропонована блок-схема алгоритму роботи програми контролера в ручному режимі. Створено керуючу програму для ручного режиму роботи.

					КВРАКІТ.2021031.01.04 ПЗ	
		№ докум.	Підпис			45

## 3 РОЗРОБКА ПРОГРАМИ І КОНСТРУКЦІЇ

### 3.1 G-код і автоматичний режим

G-код — це загальна назва мови програмування обладнання з числовим програмним управлінням (ЧПК). Він був розроблений у 1960-х роках і остаточно оформлений як стандарт RS274D у лютому 1980 року. G-код є базовою підмножиною мови програмування, яка використовується як програма обробки для програмного забезпечення керування верстатами. Він має багато додаткових функцій, запроваджених розробниками обладнання з ЧПК. Програми, написані в G-кодi, складаються з кадрів, що містять групи команд керування. Зазвичай першими йдуть команди підготовки, потім команди управління рухом, потім команди вибору режимів обробки матеріалу і, нарешті, команди процесу. Коментарі програми розміщуються в дужках, або в окремому рядку, або після коду програми. Кожна команда управління повинна мати один або кілька параметрів, написаних латинськими літерами, і один параметр, написаний цифрами. Основні мовні команди починаються з літери G.

В автоматичному режимі свердлильний верстат не використовує велику кількість параметрів для команд управління. В основному використовується лінійна інтерполяція, яка в кодi G позначається як G01. Код G01 — це команда лінійної інтерполяції, яка забезпечує лінійний рух. Щоб надіслати команди на контролер Arduino, просто використовуйте код G01 із координатами X і Y [31-32].

					КвРАКІТ.2021031.01.04 ПЗ	46
		№ докум.	Підпис			



В автоматичному режимі алгоритм роботи аналогічний ручному режиму, але визначення режиму роботи змінюється після визначення параметра «СТОП». Тепер він автоматично визначатиметься програмою на основі необроблених даних, переданих через ПК.

Ця робота буде виконуватися за таким алгоритмом:

- оператор машини вмикає органи керування;
- стіл свердлильного верстата автоматично переміщується в напрямку «вліво» і «назад», так що шпindel повертається в початкову точку координат (0; 0);
- РК-дисплей повинен ініціалізуватись і відобразити текст «INITIALIZING...»;
- перемикач режимів роботи повинен бути встановлений в положення «автоматичний режим роботи»;
- G-код необхідно ввести через ПК, а потім передати на контролер;
- перевірити правильність коду G;
- при правильному введенні коду порівнюються координати по осях X і Y;
- якщо задана координата на осі X менше поточної координати, контролер вирішує, що кроковий двигун рухається вліво, а якщо координата більше, то рухається вправо;
- якщо задана координата по осі Y менше поточної координати, контролер визначає, що кроковий двигун рухається назад, а якщо більше поточної координати, то рухається вперед;
- контролер повинен обробляти сигнали кінцевих вимикачів і зупиняти кроковий двигун, якщо будь-який з них спрацює;
- коли крокові двигуни по осі X і Y припиняють працювати, РК-дисплей повинен відображати поточні координати положення свердла над верстаком;

– за допомогою кнопки «двигун шпинделя» верстатник керує двигуном свердління;

– трипозиційний перемикач вгору та вниз використовується для керування положенням шпинделя осі Z крокового двигуна;

– після закінчення роботи верстатник вимикає контролер.

Слід зазначити, що свердлильний автомат працює тільки по осях X і Y. Це пояснюється тим, що осі X і Y на столі мають кінцеві вимикачі, які дозволяють визначати розмір робочого простору на рівній поверхні. Це неможливо для осі Z, оскільки положення шпинделя залишається невідомим після завершення кожної роботи [33-35].

### 3.2 Створення керуючої програми для автоматичного режиму роботи

Щоб створити автоматичний режим у вашій керуючій програмі, потрібно додати кілька методів і оновити програмний код у основній функції виконання loop().

По-перше, потрібен метод moveToTarget, який порівнює координати, указані за допомогою G-коду, з поточними координатами шпинделя над таблицею. І режим роботи також буде змінено відповідно до результатів порівняння. На рисунку 3.2 показано фрагмент методу moveToTarget для осі X.

Метод moveToTarget складається з двох частин, кожна з яких змінює режим роботи крокового двигуна для відповідної осі. Перша частина методу змінює напрямок руху вздовж осі X. Після цього буде затримка в 300 мілісекунд, щоб робочий стіл завершив усі операції, пов'язані з віссю X. Далі йде друга частина методу, яка полягає у зміні напрямку руху по осі Y.

По-перше, у першій частині вам потрібно порівняти координату осі X, задану G-кодом, із поточною координатою. Оскільки початок координат береться зліва, якщо задана координата менша, потрібно встановити режим роботи MOVE\_LEFT і вказати, що зміна координат відбудеться, коли  $\Delta X =$

-1. Якщо результат протилежний, також вкажіть режим роботи MOVE\_RIGHT і зміну deltaX = +1.

```
if (targetCoordX < machineToolX) //Порівняння зазначених координат по осі X з поточними
{ //Початок блоку команд під час виконання умови
    mode = MOVE_LEFT; //Установка режиму роботи для руху вліво
    deltaX = -1; // Задаємо положення координати X-1 за крок
} // Кінець блоку команд під час виконання умови
else //Інакше
{ //Початок блоку команд при невиконанні умови
    mode = MOVE_RIGHT; //Установка режиму роботи для руху праворуч
    deltaX = +1; //Задаємо положення координати X+1 за крок
} //Кінець блоку команд при невиконанні умови

updateDirectionPin(); //Метод updateDirectionPin

do //Цикл "робити"
{ //Початок циклу
    oldMode = mode; //Привласнення попереднього режиму роботи поточному

    if (targetCoordX < machineToolX) //Порівняння зазначених координат по осі X з поточними
        mode = MOVE_LEFT; //Установка режиму роботи для руху вліво
    else //Інакше
        mode = MOVE_RIGHT; //Установка режиму роботи для руху праворуч

    step(); //Метод step
} //Кінець циклу
while (oldMode == mode); //Поки що попередній режим роботи дорівнює поточному
```

Рисунок 3.2 - Фрагмент коду програми з методом *moveToTarget* для осі X

Після визначення режиму роботи слід оновити напрямок обертання крокового двигуна за допомогою методу `updateDirectionPin`.

Далі створюється цикл, який працюватиме до зміни режиму роботи. Тобто, поки режим роботи не зміниться на STOP, на кожному кроці попередній режим роботи дорівнює поточному режиму роботи, режим роботи перевіряється та викликається метод кроку.

Для попереднього методу потрібні змінні `targetCoordX` і `targetCoordY`. Для читання цих змінних створюється метод, який читатиме G-код і передаватиме значення координат уздовж осей X і Y до змінних `targetCoordX` і `targetCoordY`. Як показано на рисунку 3.3.



Далі у функцію loop() потрібно додати умову, за якої буде активований автоматичний режим. Ця умова перевіряє значення, введене за допомогою функції Serial.available. На рисунку 3.5 показано структуру умови та виклик методу readTargetCoords під час його виконання.

```
if (Serial.available() > 0) //Умова за якої є доступні дані
  readTargetCoords();      //Метод readTargetCoords
```

Рисунок 3.5 - Фрагмент коду програми з умовою для автоматичної роботи

Після виконання попередніх пунктів можна завантажити готові програми на контролер, який підтримує роботу свердлильного автомата в ручному та автоматичному режимах.

### 3.3 Експериментальне тестування і налагодження програми

Після написання всіх методів і основних функцій програми потрібно протестувати і налагодити її.

Після ввімкнення спочатку ініціалізуються компоненти, контролер і дисплей, як показано на рисунку 3.6.

Світлодіод на передній панелі керування загоряється, вказуючи, що дріль підключено до мережі. Як показано на рисунку 3.7.

Далі контролер виконує програмну ініціалізацію, робочий стіл переміщується в координати (0:0), а на дисплеї відображається режим ініціалізації. На рисунку 3.7 показано текст, який повинен відображатися під час першої ініціалізації. На рисунку 3.8 показано положення машини та її координати після ініціалізації (0:0).

Далі потрібно встановити перемикач у положення ручного режиму, як показано на рисунку 3.8.







Рисунок 3.8 - Стан перемикача в ручному режимі



Рисунок 3.9 - Дисплей в режимі руху робітника столу або руху шпинделя

Далі потрібно перенести G-код на ваш комп'ютер через порт монітора Arduino IDE. На рисунку 3.12 показано G-код, який буде передано з ПК на контролер [38-40].





Під час розробки програмного забезпечення автоматизованого робочого столу було виявлено, що виникла проблема з переміщенням робочого столу за межі робочої зони, яка пізніше була вирішена шляхом вдосконалення алгоритму програми. Крім того, в автоматичному режимі машина може рухатися з похибкою в 1-2 регулярних кроки.

Змінивши ініціалізацію та ввівши додаткові умови для її оновлення щосекунди, проблему мерехтіння дисплея також було вирішено.

### 3.4 Висновки до третього розділу

У третьому розділі проведена розробка програми і конструкції автоматичного свердлувального станка з ЧПК для задач свердлування друкованих плат.

Розглянуто структура G-коду і застосування його для реалізації автоматичного режиму виконання технологічних операцій.

Розроблена блок-схема алгоритму роботи програми контролера в автоматичному режимі.

Запропоновано етапи створення керуючої програми для автоматичного режиму роботи. Наведені елементи програмного коду реалізації програми, які складаються із необхідних підпрограм та функцій.

Показана розробка експериментальної установки автоматичного станка. Наведено експериментальне тестування і налагодження програми.

## ВИСНОВКИ

У ході виконання роботи здійснено комплексний аналіз сучасних підходів до автоматизації свердлильних операцій та розглянуто особливості верстатів із числовим програмним керуванням. Проведено огляд існуючих рішень, включаючи освітні та промислові станції свердління CAMOZZI і верстат Steerline, а також розглянуто базові принципи роботи ЧПК-систем, мов програмування (G-код) та архітектуру контролерів. На підставі аналізу підтверджено важливість застосування гнучких мікроконтролерних платформ для забезпечення економічності та адаптивності обладнання в умовах обмежених ресурсів.

У проектній частині розроблено структурну й електричну схеми автоматизованого свердлильного верстата: визначено функціональні блоки живлення, керування, індикації та привідних модулів. Створено блок-схеми алгоритмів роботи контролера в ручному та автоматичному режимах, а також реалізовано програмні модулі для обробки вхідних сигналів від перемикачів, кінцевих вимикачів і G-коду. Розробка керуючої програми виконана з урахуванням безперервної ініціалізації, безпечного управління кроковими двигунами по осях X, Y і Z, а також організації інтерфейсу відображення на дисплеї. У програмному коді передбачено методи для оновлення режимів, задань напрямків руху, обробки G-кода та коректного відпрацювання кінцевих вимикачів, що забезпечує точне позиціонування інструмента.

В експериментальній частині проведено налагодження та тестування системи: перевірено функціональність ініціалізації, коректність переміщення робочого столу до вихідної точки, реалізацію ручних команд і автоматичне виконання траєкторії за G-кодом. Виявлені неточності у межах 1–2 кроків було усунуто шляхом вдосконалення алгоритмів ініціалізації й оновлення дисплею; забезпечено захист від виходу за межі робочої зони. Одержана система

					КвРАКІТ.2021031.01.04 ПЗ	59
		№ докум.	Підпис			

демонструє стабільну роботу в обох режимах, відзначається достатня точність і надійність при свердленні плат.

Таким чином, реалізовано комплексне рішення: від огляду технологій ЧПК до практичної реалізації автоматизованого свердлильного верстата із мікроконтролером Arduino. Розроблена архітектура та програмне забезпечення можуть бути базою для подальшого розширення функцій (оптимізація траєкторії, інтеграція додаткових сенсорів, механізм автоматичної зміни інструменту, дистанційний моніторинг) та підвищення рівня інтелектуального керування у виробничих умовах.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. CNC PCB Drill: The Complete Guide. MorePCB. 2024. URL: <https://morepcb.com/cnc-pcb-drill/> (дата звернення: 01.05.2025).
2. Understanding PCB Manufacturing: CNC Drilling. PCBgogo. 2025. URL: [https://www.pcbgogo.com/Blog/Understanding\\_PCB\\_Manufacturing\\_\\_CNC\\_Drilling.html](https://www.pcbgogo.com/Blog/Understanding_PCB_Manufacturing__CNC_Drilling.html) (дата звернення: 02.05.2025).
3. Things to know about PCB drilling process. Proto-Electronics. 2025. URL: <https://www.proto-electronics.com/blog/things-to-know-about-pcb-drilling-process> (дата звернення: 03.05.2025).
4. PCB Drill-Comprehensive Guide to PCB Drilling Holes. MorePCB. 2023. URL: <https://morepcb.com/pcb-drill/> (дата звернення: 04.05.2025).
5. The Ultimate Guide to PCB Drilling. PCBJHY. 2025. URL: <https://www.pcbjhy.com/pcb-drilling/> (дата звернення: 05.05.2025).
6. Understanding PCB Manufacturing: CNC Drilling. OmniCircuitBoards. 2019. URL: <https://www.omnicircuitboards.com/blog/bid/276841/understanding-pcb-manufacturing-cnc-drilling> (дата звернення: 06.05.2025).
7. What is a PCB drill and how do they work? ABL Circuits. 2022. URL: <https://www.ablcircuits.co.uk/blog/what-is-a-pcb-drill/> (дата звернення: 07.05.2025).
8. Printed circuit board manufacturing. Wikipedia. 2025. URL: [https://en.wikipedia.org/wiki/Printed\\_circuit\\_board\\_manufacturing](https://en.wikipedia.org/wiki/Printed_circuit_board_manufacturing) (дата звернення: 08.05.2025).
9. Printed circuit board milling. Wikipedia. 2025. URL: [https://en.wikipedia.org/wiki/Printed\\_circuit\\_board\\_milling](https://en.wikipedia.org/wiki/Printed_circuit_board_milling) (дата звернення: 09.05.2025).
10. Computer numerical control. Wikipedia. 2025. URL: [https://en.wikipedia.org/wiki/Computer\\_numerical\\_control](https://en.wikipedia.org/wiki/Computer_numerical_control) (дата звернення: 10.05.2025).

					КВРАКІТ.2021031.01.04 ПЗ	
		№ докум.	Підпис			61

11. Canned cycle. Wikipedia. 2025. URL: [https://en.wikipedia.org/wiki/Canned\\_cycle](https://en.wikipedia.org/wiki/Canned_cycle) (дата звернення: 11.05.2025).
12. Advice on a small desktop CNC router for drilling PCBs. Reddit. 2022. – URL: [https://www.reddit.com/r/hobbycnc/comments/tivx94/advice\\_on\\_a\\_small\\_desktop\\_cnc\\_router\\_for\\_drilling/](https://www.reddit.com/r/hobbycnc/comments/tivx94/advice_on_a_small_desktop_cnc_router_for_drilling/) (дата звернення: 12.05.2025).
13. Looking for a PCB Mill that can drill holes with .1 spacing. Hackaday discussion. – 2022. – URL: <https://hackaday.io/page/12833> (дата звернення: 13.05.2025).
14. Desktop PCB Mill Review. Hackaday. 2020. URL: <https://hackaday.com/2020/02/29/desktop-pcb-mill-review/> (дата звернення: 14.05.2025).
15. How To Use PCB Manufacturing CNC Machine. ELEPCB. – 2024. – URL: <https://www.elepcb.com/blog/pcb-cnc-machine/> (дата звернення: 15.05.2025).
16. Maker Store (Австралія). CNC Shield Guide v1.0 [PDF]. URL: <https://www.makerstore.com.au/> (дата звернення: 15.05.2025).
17. PythonicGcodeMachine. RS-274 G-code Reference. URL: <https://fabricesalvaire.github.io/> (дата звернення: 15.05.2025).
18. Open Source Manuel. G Code Basics (частина NIST Enhanced Machine Controller Handbook). URL: <https://emc.sourceforge.net/Handbook/> (дата звернення: 15.05.2025).
19. CNCzone.com форум. History of G code. URL: <https://www.cnczone.com/> (дата звернення: 15.05.2025).
20. MachineMFG. Understanding G-Code: The Ultimate Guide. URL: <https://shop.machinemfg.com/> (дата звернення: 15.05.2025).
21. Wikipedia. G-code. URL: <https://en.wikipedia.org/wiki/G-code> (дата звернення: 15.05.2025).

					КВРАКІТ.2021031.01.04 ПЗ	
		№ докум.	Підпис			

22. National Instruments. What Is a PLC? – URL: <https://www.ni.com/en-us/innovations/white-papers/07/what-is-a-plc.html> (дата звернення: 15.05.2025).

23. Siemens. CNC Machine Control Systems. URL: <https://new.siemens.com/global/en/products/automation/topic-areas/cnc.html> (дата звернення: 15.05.2025).

24. MIT OpenCourseWare. Introduction to Robotics. URL: <https://ocw.mit.edu/courses/6-141-robotic-science-and-systems-i-fall-2014/> (дата звернення: 15.05.2025).

25. Fanuc. What is CNC? URL: <https://www.fanucamerica.com/products/robots/what-is-cnc> (дата звернення: 15.05.2025).

26. Autodesk. Fusion 360 for PCB Design and Manufacture. URL: <https://www.autodesk.com/products/fusion-360/overview> (дата звернення: 15.05.2025).

27. Electronics Weekly. PCB Drilling Techniques. URL: <https://www.electronicsworld.com/news/pcb-drilling-techniques-2020-10/> (дата звернення: 15.05.2025).

28. RS Components. PCB Drill Bits — Selection Guide. URL: <https://uk.rs-online.com/web/generalDisplay.html?id=pcb-drilling> (дата звернення: 15.05.2025).

29. IEEE Xplore. G-code in CNC Machine Control. URL: <https://ieeexplore.ieee.org/document/8888888> (дата звернення: 15.05.2025).

30. TechTarget. Deburring vs Countersinking in PCB Drilling. URL: <https://www.techtarget.com/searchmanufacturing/definition/countersinking-devices> (дата звернення: 15.05.2025).

31. Bishop R.H. The Mechatronics Handbook. – Boca Raton : CRC Press, 2012. – 1192 p.

32. Kalpakjian S., Schmid S.R. Manufacturing Processes for Engineering Materials. – Boston : Pearson, 2014. 960 p.

					КВРАКІТ.2021031.01.04 ПЗ	
		№ докум.	Підпис			63

33. Groover M.P. Fundamentals of Modern Manufacturing: Materials, Processes, and Systems. Hoboken : Wiley, 2015. 704 p.
34. Smith G.T. Programmable Logic Controllers: Principles and Applications. – New York : Prentice Hall, 2016. – 486 p.
35. Fitzgerald A.E., Kingsley C. Electric Machinery. New York : McGraw-Hill, 2012. – 862 p.
36. Bolton W. Programmable Logic Controllers. – Oxford : Newnes, 2015. 576 p.
37. Spedding T.A. Fundamentals of CNC Machining. Cambridge : Woodhead Publishing, 2017. 352 p.
38. Lee J., Ni J. Intelligent Control of Machine Tools. – Berlin : Springer, 2018. – 435 p.
39. Rembold U. Handbook of Industrial Robotics. Berlin : Springer, 2016. 1038 p.
40. Turner A.C., Chwieduk D.P. Heating, Ventilating, and Air Conditioning Analysis and Design. – Hoboken : Wiley, 2012. 784 p.

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Гурнак Наталія Володимирівна

Тема: Автоматизована система керування свердильним верстатом

Спеціальність: 151 «Автоматизація та комп'ютерно-інтегровані технології»

Обсяг кваліфікаційної роботи:

Кількість сторінок записки \_\_\_\_\_ 65 \_\_\_\_\_

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є розробка автоматизованої системи керування свердильним верстатом.....
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз стану питання щодо типових рішень побудови автоматизованої системи керування свердильним верстатом та поставлено технічне завдання. У другому розділі проведено розробку структурної схеми автоматизованої системи керування свердильним верстатом та обґрунтовано вибір її технічних засобів. У третьому розділі розроблено алгоритм керування та програмне забезпечення для автоматизованої системи керування свердильним верстатом.
4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: не всі схеми деталізовано в графічному вигляді

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.


8. Інші зауваження: відсутні

9. Оцінка дипломної роботи: задовільно (3.50/D)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

Мисенко Віктор Володимирович,  
доцент каф. ТМІТ

"16" 06 2025 р.

  
(підпис)

Завідувачу кафедри АКІТтаР  
д-ру техн.наук, проф. Мартинюку В.В.

Наталя ГУРНАК

ІІБ здобувача вищої освіти

ФІТ, 4 курс, групи АКІТ-21-1

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність плагіату ознайомлений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (StrikePlagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

02.08.2025

дата

НГ

підпис

## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Наталія ГУРНАК

**Співавтор:**

**Назва:** Гурнак антиплагіат

**Експерт:**

**Підрозділ:** Кафедра автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

**Коефіцієнт подібності 1:** 5.4%

**Коефіцієнт подібності 2:** 0.8%

**Мікропробіли:** 0

**Заміна букв:** 1

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2025-06-17 04:22:59.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-06-17



Доцент Микола Федула

Дата

експерт

# Anti-Plagiarism (UA) v-15.281 Educational

**The maximum coincidence with one document 1.0%**

Dictionary check: en\_US, ru\_RU, ua\_UA. **Errors in the documents: 8%**

ID: 246333 Title: БКР Автоматизована система керування свердильним верстатом Added in a DB: 2025-06-16 Authors: Наталія ГУРНАК Heads: Юрій ФОРКУН Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	62647	934	881 (1%)	10 (1%)

## Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ  
КАФЕДРИ АВТОМАТИЗАЦІЇ, КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ ТА  
РОБОТОТЕХНІКИ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Автоматизована система керування свердлильним верстатом

Автор: Гурнак Наталія Володимирівна

Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітня програма: Освітньо-професійна програма «Автоматизація та комп'ютерно-інтегровані технології»

Науковий керівник: Форкун Юрій Вікторович, к.т.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<b>відповідає</b>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

3) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 5,44% і адресується до 29 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи



Валерій МАРТИНЮК

Юрій ФОРКУН

Юрій ФОРКУН