

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки  
Факультет радіоелектроніки та телекомунікацій  
(повне найменування інституту, факультету)

Кафедра радіотехніки та телекомунікацій  
(повне найменування кафедри)

## Пояснювальна записка

до дипломного проєкту (роботи)  
магістра  
(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРАКТИЧНА  
РЕАЛІЗАЦІЯ ПРОТОТИПУ КІБЕРПРОТЕЗУ

Виконав: студент(ка) 2 курсу, групи РТ-229м  
Спеціальності 172 «Телекомунікації та радіотехніка»  
(код і найменування спеціальності)

Освітня програма (спеціалізація) «Телемедичні та біомедичні системи»

Попович Василь Вікторович  
(прізвище та ініціали)

Керівник Гладкова О.М.  
(прізвище та ініціали)

Рецензент Пархоменко А.В.  
(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»  
(повне найменування закладу вищої освіти)

Інститут, факультет Інститут інформатики та радіoeлектроніки, ФРЕТ  
Кафедра Радіотехніка та телекомунікації  
Ступінь вищої освіти Магістр  
Спеціальність 172 «Телекомунікації та радіотехніка»  
(код і найменування)  
Освітня програма (спеціалізація) «Телемедичні та біомедичні системи»  
(назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри РТТ  
к.т.н, доц. Морицавка С.В.   
«  » грудня 2020 року

**З А В Д А Н Н Я**

**НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА**

Поповича Василя Вікторовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та практична реалізація прототипу кіберпротезу

керівник проєкту (роботи) Гладкова Ольга Миколаївна, к.т.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «6» листопада 2020 року № 315

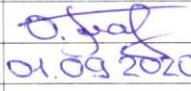
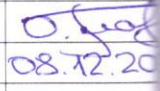
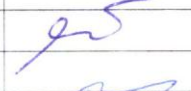
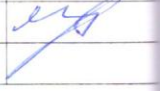
2. Строк подання студентом проєкту (роботи) 11 грудня 2020 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз існуючих медичних кіберфізичних систем. Дослідження існуючих технологій прототипування. Розробка прототипу кіберпротезу верхньої кінцівки людини. Економіко-організаційна частина. Охорона праці та безпека в надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Презентація роботи в Microsoft PowerPoint

## 6. Консультанти розділів проєкту (роботи)

| Розділ         | Прізвище, ініціали та посада консультанта | Підпис, дата                                                                                      |                                                                                                 |
|----------------|-------------------------------------------|---------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
|                |                                           | завдання видав                                                                                    | прийняв виконав завдання                                                                        |
| 1-3            | Гладкова О.М., доцент                     | <br>01.09.2020 | <br>08.12.20 |
| 4              | Левченко Н.М., професор                   |                                                                                                   |                                                                                                 |
| 5              | Якімцов Ю.В., доцент                      |                |              |
| нормо-контроль | Мороз Г.В., ст.викладач                   |                |              |

7. Дата видачі завдання « 01 » вересня 2020 року.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту (роботи)                                          | Строк виконання етапів проєкту ( роботи ) | Примітка |
|-------|-----------------------------------------------------------------------------------|-------------------------------------------|----------|
| 1     | Постановка завдання роботи                                                        | 1-й тиждень                               | Завдання |
| 2     | Аналіз предметної області                                                         | 2-3 тижні                                 | Розділ 1 |
| 3     | Дослідження та удосконалення методів, моделей й алгоритмів вирішення задачі       | 4-5 тижні                                 | Розділ 2 |
| 4     | Розробка архітектури системи                                                      | 6-й тиждень                               | Розділ 2 |
| 5     | Розробка апаратного та програмного забезпечення                                   | 7-9 тижні                                 | Розділ 2 |
| 6     | Тестування та експериментальне дослідження апаратного та програмного забезпечення | 10-11 тижні                               | Розділ 2 |
| 7     | Робота над економіко-організаційною частиною                                      | 12-й тиждень                              | Розділ 2 |
| 8     | Розробка рекомендацій з охорони праці та безпеки в надзвичайних ситуаціях         | 13-й тиждень                              | Розділ 2 |
| 9     | Оформлення пояснювальної записки та презентації                                   | 14-й тиждень                              | Додаток  |
| 10    | Нормоконтроль та рецензування, перевірка на плагіат, оформлення дублінського ядра | 15-й тиждень                              |          |

Студент(ка)

  
( підпис )

Попович В.В.  
(прізвище та ініціали)

Керівник проєкту (роботи)

  
( підпис )

Гладкова О.М.  
(прізвище та ініціали)

## РЕФЕРАТ

ПЗ: 135 сторінок, 56 рисунків, 19 таблиць, 86 джерел, 1 додаток

Об'єкт дослідження – процес прототипування схем та конструкцій при автоматизованому проєктуванні біомедичних виробів.

Метою роботи є розробка та дослідження віртуального та фізичного прототипів кіберпротезу верхньої кінцівки людини для підвищення ефективності процесу автоматизованого проєктування.

Матеріали та методи: плата Arduino Nano, плата розширення Arduino Sensor Shield, серводвигуни SG90, ABS пластик, метод прототипування, метод аналізу, експериментальне дослідження.

В першому розділі виконано огляд кіберфізичних систем; дослідження структурних та функціональних особливостей існуючих кіберпротезів руки. Виявлено основні критерії та правила розробки кіберпротезів, а також сформульовано мету та задачі магістерської роботи. В другому розділі досліджено особливості технології прототипування, а також методів та засобів її реалізації при розробці та дослідженні прототипу кіберпротезу. Третій розділ присвячено практичній реалізації прототипу кіберпротезу на основі розробленого методу комплексного віртуального та фізичного прототипування, а також інформаційної технології створення прототипу механічної частини кіберпротезу. Виконано аналіз та вибір компонентів для електронної частини прототипу, їх інтеграція з механічною частиною. Розроблено програмне забезпечення для функціонування кіберпротезу та сценарії тестування досяжних захоплень. В четвертому розділі виконано економічне обґрунтування проєкту, а в п'ятому розроблено заходи з охорони праці та безпеки в надзвичайних ситуаціях.

КІБЕРФІЗИЧНІ СИСТЕМИ, ВІРТУАЛЬНИЙ ПРОТОТИП, ЦИФРОВИЙ ДВІЙНИК, ФІЗИЧНИЙ ПРОТОТИП, 3D-ДРУКУВАННЯ, 3D-СКАНУВАННЯ, ARDUINO, СЕРВОПРИВІД

## ЗМІСТ

|                                                                                             |    |
|---------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК СКОРОЧЕНЬ.....                                                                      | 7  |
| ВСТУП.....                                                                                  | 8  |
| 1 АНАЛІЗ ІСНУЮЧИХ МЕДИЧНИХ КІБЕРФІЗИЧНИХ СИСТЕМ .....                                       | 9  |
| 1.1 Огляд кіберфізичних систем.....                                                         | 9  |
| 1.2 Дослідження структурних та функціональних особливостей кіберфізичних протезів руки..... | 12 |
| 1.3 Аналіз критеріїв та правил розробки кіберпротезів.....                                  | 18 |
| 1.4 Постановка задач магістерської роботи.....                                              | 23 |
| 2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ТЕХНОЛОГІЙ ПРОТОТИПУВАННЯ...                                         | 25 |
| 2.1 Огляд особливостей прототипування.....                                                  | 25 |
| 2.2 Технології розробки віртуального прототипу .....                                        | 28 |
| 2.2.1 Прототипування електронної частини кіберпротезу.....                                  | 28 |
| 2.2.2 Прототипування механічної частини кіберпротезу.....                                   | 33 |
| 2.2.3 Побудова візуальної моделі цифрового двійника .....                                   | 40 |
| 2.3 Технології розробки фізичного прототипу.....                                            | 48 |
| 2.4 Висновки до розділу 2.....                                                              | 54 |
| 3 РОЗРОБКА ПРОТОТИПУ КІБЕРПРОТЕЗУ ВЕРХНЬОЇ КІНЦІВКИ ЛЮДИНИ.....                             | 55 |
| 3.1 Метод комплексного віртуального та фізичного прототипування.....                        | 55 |
| 3.2 Розробка механічної частини прототипу кіберпротезу.....                                 | 56 |
| 3.2.1 Розробка віртуального прототипу конструктиву кіберпротезу .....                       | 57 |
| 3.2.2 Виготовлення фізичного прототипу конструктиву кіберпротезу.....                       | 62 |
| 3.3 Розробка електронної частини прототипу кіберпротезу.....                                | 64 |
| 3.3.1 Аналіз та вибір плати Arduino.....                                                    | 64 |
| 3.3.2 Аналіз та вибір плати розширення для Arduino.....                                     | 73 |
| 3.3.3 Аналіз та вибір серводвигунів.....                                                    | 77 |

|                                                                                         |     |
|-----------------------------------------------------------------------------------------|-----|
| 3.4 Інтеграція компонентів електронної та механічної частин прототипу кіберпротезу..... | 84  |
| 3.5 Розробка програмного забезпечення для прототипу.....                                | 87  |
| 3.6 Висновки до розділу 3.....                                                          | 91  |
| 4 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА.....                                                  | 92  |
| 4.1 Планування розробки програмного виробу .....                                        | 92  |
| 4.2 Визначення витрат на розробку програми.....                                         | 95  |
| 4.2.1 Розрахунок основної заробітної плати.....                                         | 95  |
| 4.2.2 Розрахунок додаткової заробітної плати.....                                       | 96  |
| 4.2.3 Відрахування на єдиний соціальний внесок.....                                     | 96  |
| 4.2.4 Визначення витрат на матеріали.....                                               | 97  |
| 4.2.5 Витрати на спеціальне устаткування.....                                           | 98  |
| 4.2.6 Загальновиробничі витрати.....                                                    | 101 |
| 4.3 Оцінка економічної ефективності.....                                                | 102 |
| 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ...                                  | 103 |
| 5.1 Аналіз потенційних небезпек.....                                                    | 103 |
| 5.2 Заходи по забезпеченню техніки безпеки.....                                         | 104 |
| 5.3 Заходи з виробничої санітарії та гігієни праці.....                                 | 106 |
| 5.4 Заходи безпеки у надзвичайних ситуаціях.....                                        | 115 |
| 5.4.1 Заходи з пожежної безпеки.....                                                    | 115 |
| 5.4.2 Заходи з цивільного захисту.....                                                  | 116 |
| ВИСНОВКИ.....                                                                           | 120 |
| ПЕРЕЛІК ПОСИЛАНЬ.....                                                                   | 122 |
| ДОДАТОК А.....                                                                          | 131 |

## ПЕРЕЛІК СКОРОЧЕНЬ

|       |                                               |
|-------|-----------------------------------------------|
| API   | – Application Programming Interface           |
| ARM   | – Advanced RISC Machine                       |
| BIM   | – Building Information Model                  |
| CAD   | – Computer Aided Design                       |
| CAM   | – Computer Aided Manufacturing                |
| CPS   | – Cyber-Physical Systems                      |
| CPU   | – Central Processing Unit                     |
| GCC   | – GNU Compiler Collection                     |
| IDE   | – Integrated Development Environment          |
| IoT   | – Internet of Things,                         |
| M2M   | – Machine-to-Machine                          |
| PDM   | – Product Data Management                     |
| PLM   | – Product Lifecycle Management                |
| RTOS  | – Real-Time Operating System                  |
| USART | – Universal Asynchronous Receiver-Transmitter |
| КФС   | – Кіберфізичні системи                        |
| САПР  | – Система автоматизованого проєктування       |
| ЧПУ   | – Числове програмне управління                |

## ВСТУП

Розвиток сучасних технологій в галузі медицини надає нові можливості, починаючи від розумних імплантів і протезів, закінчуючи 3D-принтіngом тканин і навіть органів.

Якщо раніше людина, втративши кінцівку, відчувала значний дискомфорт і неприродність при носінні її замінника, то сучасні протези стали більш наближеними як функціонально, так і візуально. Зокрема, в роботі [1] представлено цілу лінійку кіберпротезів рук та проведено єдину процедуру тестування для всіх. Показано, що важливими параметрами, що впливають на показники якості протезу є вага руки [2-3], розміри та анатомічно правильна кінематика пальців, тип приводу та приводний механізм, сила та швидкість захоплення, довговічність та зовнішній вигляд [4].

В той же час, існує ціла низка проблем, пов'язаних з проєктуванням, виготовленням та використанням протезів. Основна з них полягає в складності розв'язку задачі керування, оскільки контролери протезу мають виконувати різні функції згідно з сигналами, які користувач надсилає за допомогою м'язів, головного або навіть спинного мозку, тобто фактично на рівні рефлексів. Інші проблеми пов'язані зі складністю паралельного проєктування механічної та електронної частин протезу. В результаті, вартість інтелектуальних протезів залишається досить високою, що не дозволяє радикально вирішити проблеми людей з особливими потребами. Тому, актуальною задачею є розробка більш дешевих та доступних пацієнтам роботизованих протезів кінцівок, що надають базовий набір рухів та можливість простого управління [5].

Параметри та характеристики реального виробу можна покращити, а його вартість знизити на основі використання сучасних технологій прототипування [6-7].

Метою роботи є розробка та дослідження віртуального та фізичного прототипів кіберпротезу верхньої кінцівки людини для підвищення ефективності процесу автоматизованого проєктування.

## 1 АНАЛІЗ ІСНУЮЧИХ МЕДИЧНИХ КІБЕРФІЗИЧНИХ СИСТЕМ

### 1.1 Огляд кіберфізичних систем

Кіберфізичні системи (Cyber-Physical Systems, CPS) – це системи, що складаються з різних природних об'єктів, штучних підсистем і керуючих контролерів, що дозволяють уявити таке утворення як єдине ціле. У кіберфізичних системах (КФС) забезпечується тісний зв'язок і координація між обчислювальними і фізичними ресурсами. Комп'ютери здійснюють моніторинг і управління фізичними процесами з використанням такої петлі зворотнього зв'язку, де те, що відбувається у фізичних системах, впливає на обчислення і навпаки [8].

КФС – це гармонійне співіснування двох типів моделей. З одного боку – це традиційні інженерні моделі (механічні, будівельні, електричні, біологічні, хімічні, медичні, економічні та інші), а з іншого – моделі комп'ютерні. Попередниками КФС можна вважати вбудовані системи реального часу, розподілені обчислювальні системи, автоматизовані системи управління технічними процесами і об'єктами, бездротові сенсорні мережі [8].

КФС – це системи, що співпрацюють з обчислювальними структурами, які перебувають у інтенсивному зв'язку з навколишнім фізичним світом і його процесами, що забезпечують і використовують, водночас, доступ до даних і обробки даних послуг, доступних в Інтернеті [9].

КФС є інтеграцією обчислювальних, мережевих та фізичних процесів. Вбудовані комп'ютери та мережі контролюють фізичні процеси, з петлями зворотного зв'язку, коли фізичні процеси впливають на обчислення і навпаки. Економічний та суспільний потенціал таких систем набагато більший, ніж реалізований, і у всьому світі вкладаються великі інвестиції для розвитку даної технології. Ця технологія ґрунтується на більш старій (але ще дуже молодій) дисципліні вбудованих систем, комп'ютерів та програмного забезпечення,

вбудованих у пристрої, основна місія яких – не обчислення, такі як машини, іграшки, медичні прилади та наукові інструменти. CPS інтегрує динаміку фізичних процесів з технологіями програмного забезпечення та мереж, надаючи абстракції та методи моделювання, проєктування та аналізу для інтегрованого цілого [10].

Головним принципом роботи кіберфізичних систем можна назвати глибокий взаємозв'язок між фізичними та обчислювальними елементами. «Мозок» системи у вигляді штучного інтелекту та інших технологій отримує дані від сенсорів в реальному світі, аналізує ці дані і використовує їх для подальшого управління фізичними елементами. Завдяки такій взаємодії кіберфізична система здатна ефективно працювати в умовах, що змінюються, як аналог людського організму або сучасна компанія, яка аналізує ситуацію на ринку, щоб розробити саме той продукт, який зараз потрібен [11].

З технічної точки зору КФС мають багато спільного зі структурами типу грид, реалізованими за допомогою Інтернету речей (Internet of Things, IoT), Індустрії 4.0 (Четверта промислова революція), промислового Інтернету речей (Industrial Internet), межмашинної взаємодії (Machine-to-Machine, M2M), туманного і хмарного комп'ютингу (fog і cloud computing). Але лише цими технічними засобами не можна обмежувати уявлення КФС. Для цих складних систем потрібні нові кібернетичні підходи до моделювання, оскільки саме моделі є центральним моментом в науці та інженерії.

Німецька академія Acatech вказує на перспективи національних кіберфізичних платформ, які складаються з трьох типів мереж [8]:

- інтернет людей;
- інтернет речей;
- інтернет сервісів.

Кіберфізичні системи виникають на стику Інтернету людей, речей і сервісів (рис 1.1). На думку німецьких академіків, перспективи появи кіберфізичних систем і формування на їх основі Індустрії 4.0 зачіпають

інтереси суспільства в цілому, тому повинні розглядатися не тільки в технічному, а в більш широкому соціокультурному аспекті, з урахуванням демографічних та інших змін, що відбуваються [8].

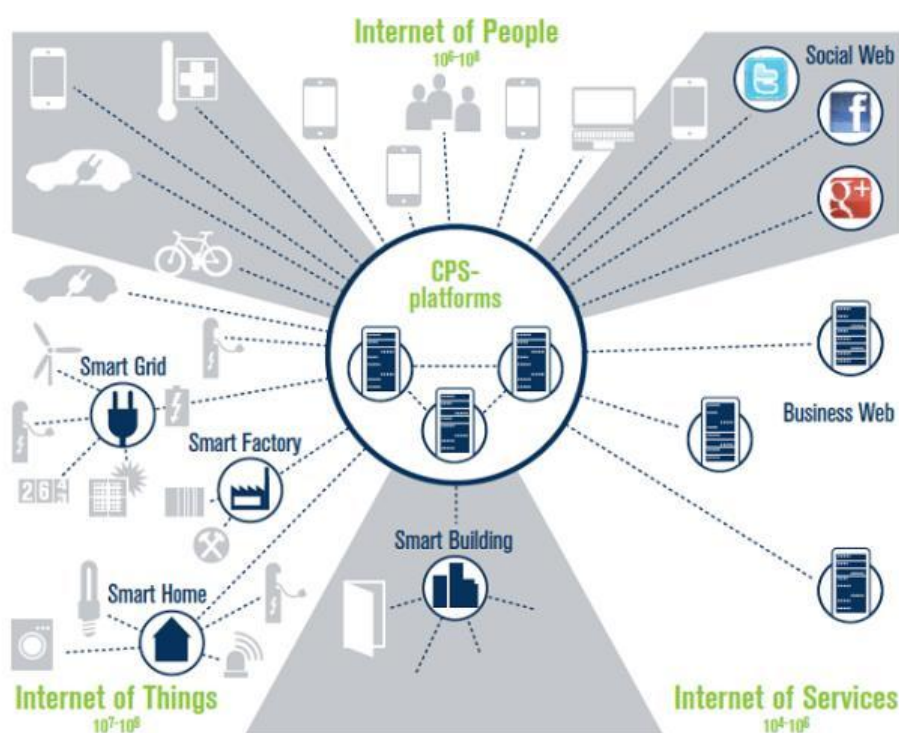


Рисунок 1.1 – Структура кіберфізичної платформи [8]

Область застосування КФС поширюється практично на всі види людської діяльності, включаючи все різноманіття промислових систем, транспортні, енергетичні та військові системи, всі види систем життєзабезпечення від медицини до розумних будинків і міст, а також багато економічних систем [8].

Кіберфізичні систем використовуються в системах управління автономним транспортом, які в реальному часі отримують інформацію від учасників дорожнього руху та дорожньої інфраструктури, щоб уникнути аварії і підібрати оптимальний маршрут до місця призначення з урахуванням ситуації на дорозі; медичне обладнання, яке дистанційно відстежує стан хворих і застосовується для дослідження людського організму; «Розумні» будинки з

ефективним споживанням електроенергії; засоби автоматизації процесів в сільському господарстві, тощо [11].

## **1.2 Дослідження структурних та функціональних особливостей кіберфізичних протезів руки**

Кіберпротези – це продукт кількох дисциплін: медицини, нейрофізіології, інженерії та програмування. Це протези нового покоління, в які вбудовані механізми автоматичного управління, адже, коли людина переставляє ногу при ходьбі, вона не думає про те, якою є сукупна робота по перенесенню ступні, зміни її нахилу в циклі кроку, синхронної зміни кута згинання в коліні і так далі. Все перераховане - це автоматичні механізми [12].

На сьогоднішній день найкраще, що може дозволити собі більшість інвалідів, які втратили, наприклад, ногу від коліна, - це прикріплений до культі стегна вертикальний стрижень з приєднаною внизу на шарнірі горизонтальною планкою, що замінює ступню. Проблема таких механічних протезів полягає в тому, що людина замість ноги отримує механічний упор, який зусиллям м'язів переносить кінцівку в циклі кроку. Кібер-протез, оснащений рушіями і акумулятором, – це фактично самостійно крокуюча нога. Щільно приєднаний до культі протез не тільки вимагатиме менше зусиль при ходьбі, але зробить рух людини більш природним і зручним. При цьому темп, розмір кроку і інші параметри руху можуть бути налаштовані для конкретного користувача [12].

Однак навіть точно налаштований протез, що дозволяє людині без палички і милиць з легкістю ходити по рівній поверхні, не впорається, якщо буде потрібно йти сходами або повернути в ту чи іншу сторону. Для кожного нового режиму руху свої налаштування. Перемикання режимів руху протеза вже не може бути автоматичним, оскільки намір змінити рух народжується в голові у людини, кібер-протезу про це не може бути відомо [12].

Таким чином, досконалі кіберпротези при всіх своїх автоматичних механізмах повинні ще мати систему управління безпосередньо від мозку, хоча б для перемикання режимів. На сьогоднішній день розроблено спеціальну технологію – інтерфейс «мозок-комп'ютер», що дозволяє на основі реєстрації

біопотенціалів мозку зчитувати найпростіші команди людини (наліво-направо, вперед-назад і так далі). Перетворені в сигнали для механізмів протеза, вони дозволяють перемикаати режими його активності. Таким чином, людина може прямо по ходу руху природним чином, тобто уявними командами, регулювати режими протеза в залежності від характеру місцевості і мети свого руху, не використовуючи для цього якісь додаткові пульти управління [12].

Попередньо людині пропонується багато разів подумки відтворювати потрібні рухи, при цьому алгоритмічні системи розпізнавання образів знаходять відповідність між цими намірами і певними ознаками в електричній активності його мозку. Натреновані таким чином алгоритми надалі, вже при вільному прояві намірів людини по ходу руху, з хорошою надійністю детектують ознаки того чи іншого наміру в корекції руху, які тут же транслуються до протезу як команди для перемикаання його автоматики в потрібний режим: рух вгору по сходах, поворот направо і так далі. Такі ж принципи управління закладаються й в кіберпротези верхніх кінцівок. Наприклад, в протези кисті вбудовуються 5-6 моторчиків, які добре керують пальцями. На ринку вже є кіберпротези кисті з керуванням їх режимами, хоча поки не від мозку, а від самих м'язів на рештках кінцівок [12].

В роботі [1] представлено результати дослідження цілої лінійки кіберпротезів руки (рис.1.2). Шість кіберфізичних рук, що показані без косметичної рукавички, являють собою останні розробки в області комерційних міоелектричних рук. У той час як руки iLimb (Touch Bionics; Livingston, Великобританія) і Bebionic (RSL Steeper; Leeds, Великобританія) отримали велику увагу засобів масової інформації, руки Vincent (Vincent Systems; Weingarten, Німеччина) і Michelangelo (Otto Bock; Duderstadt, Німеччина) тільки стають доступними для громадськості. Тому опублікована інформація про руки Вінсента та Мікеланджело обмежена. У статті наведено властивості та характеристики для кожної руки, як заявлено виробником або зібрані з відео та вторинних джерел. SensorHand, розроблений Отто Боком, також вказаний у

статті [1] як порівняння можливостей руки одного ступеня свободи з сучасними багатофункціональними ручними конструкціями.



а – рука Вінсента від Vincent Systems; б – рука iLimb від Touch Bionics;

в – iLimb Pulse від Touch Bionics; г – бевіонічна рука від RSL Steeper;

д – бевіонічна рука v2 від RSL Steeper; е – рука Мікеланджело від Отто Бока

Рисунок 1.2 – Останні розробки в області комерційних міоелектричних рук [1]

Дані, які представлені в статті [1] були зібрані з численних джерел, але експлуатаційні характеристики кожної руки було протестовано за допомогою єдиної процедури тестування, щоб краще порівняти й обговорити деталі, що стосуються конструкції рук.

Руки iLimb Pulse, Bebionic, Bebionic v2 і Michelangelo були протестовані в повністю зібраній конфігурації рук. Робочі характеристики iLimb Prodigits (ті ж пальці й система управління, що й у стандартного iLimb) і Vincent Hand finger були визначені шляхом тестування набору з чотирьох пальців, підключених до нестандартного кріплення долоні, з використанням того ж контролера і батареї, що і вся оригінальна система руки [1].

Кіберпротез i-Limb від компанії Touch Bionics [13] (рис. 1.3) повертає людині можливість використовувати руку: носити тяжкості та хапати пальцями дрібні предмети. Для встановлення i-Limb не потрібна хірургічна операція: кібер-протез управляється за допомогою датчиків, приєднаних до м'язів передпліччя. Відповідно, щоб поворушити кіберрукою достатньо напружити м'язи так, якби це була справжня рука. Власники i-Limb стверджують, що часом виникає відчуття, що кіберпротез - це їх справжня рука. Звичайно ж, це відчуття оманливе і насправді його формує мозок на основі колишніх спогадів.



Рисунок 1.3 – Touch Bionics i-Limb [13]

Кібер-протез i-Limb є одним з найдоступніших, але далеко не самим високотехнологічним. Американське оборонне агентство DARPA фінансує відразу два проекти з біомехатроніки, яким вдалося просунутися в порівнянні з i-Limb мало не на десятиріччя вперед. Великі надії DARPA покладає на кіберруку власної розробки, яка керується не м'язами передпліччя, а мозком. Для цього в мозок імплантується мікročіп, який реєструє сигнали нейронів і передає їх протезу. Головна перевага проекту DARPA - висока точність рухів, що дозволяє маніпулювати чим завгодно, навіть музичними інструментами.

Фінансує DARPA і альтернативний проєкт по створенню кібер руки Deka Luke Arm (рис. 1.4), автором якого є американський винахідник Дін Кеймен. В його основі модульна конструкція, яка дозволяє легко підлаштовувати кіберпротез під потреби конкретного користувача.

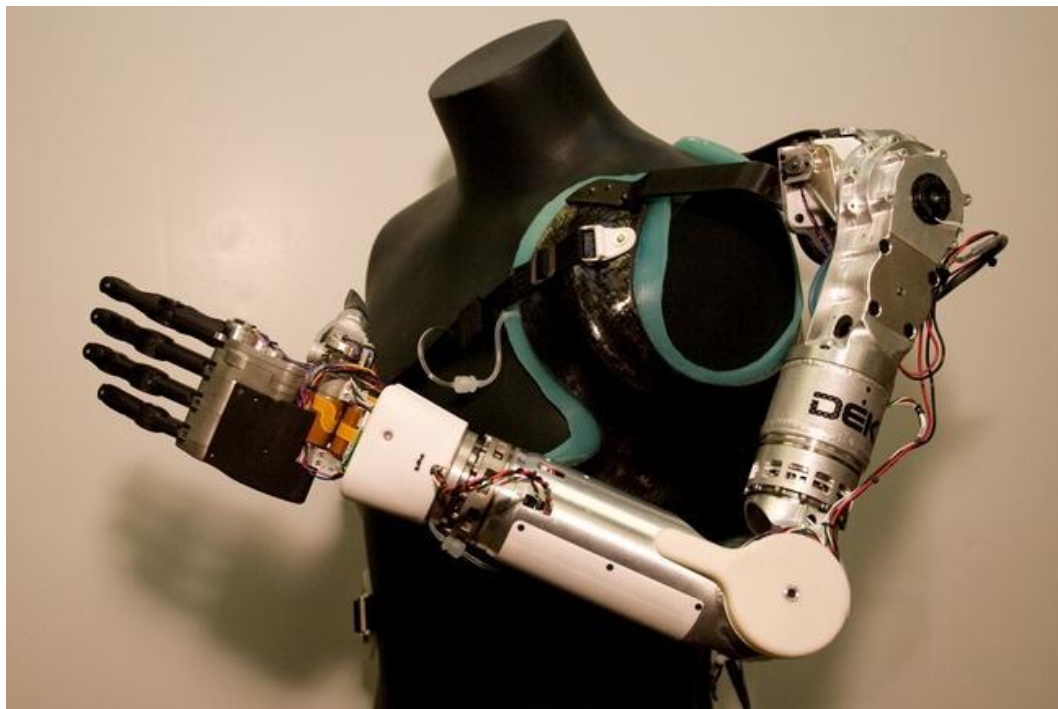


Рисунок 1.4 – Deka Luke Arm [13]

Управлятися Luke Arm може тією частиною тіла, якою забажає власник, наприклад ступнею, тобто щоб рухати кіберрукою, досить притупувати ногою [13]. Шведсько-италійський проєкт Smart Hand активно працює над зворотною віддачею кіберруки (рис. 1.5). Протез зможе не лише отримувати сигнали від мозку, але й передавати йому тактильні відчуття [13].

У випадках, коли руку не втрачено, але через проблеми з нервовою системою втратила можливість рухатися може використовуватися Possessed Hand. Як показано на рисунку 1.6 пристрій одягається на руку і стимулює м'язи згідно заданої програми, а тому може допомогти і здоровим людям. Наприклад, бажаючим навчитися грати на скрипці, ліпити з глини або жонглювати кулями.

Це дозволяє не жертвувати здоровою кінцівкою заради кібернетичної, нехай навіть більш досконалої, але проєкт Possessed Hand ще досить далекий від фінальної стадії розробки [13].

В роботі [14] також представлено результати дослідження цілої лінійки біонічних протезів верхніх кінцівок, зокрема i-Limb, Bebionic 3, Handiii, ProDigits, SensorHand Speed, SmartHand Deus Ex, MyoFacil Luke Arm.

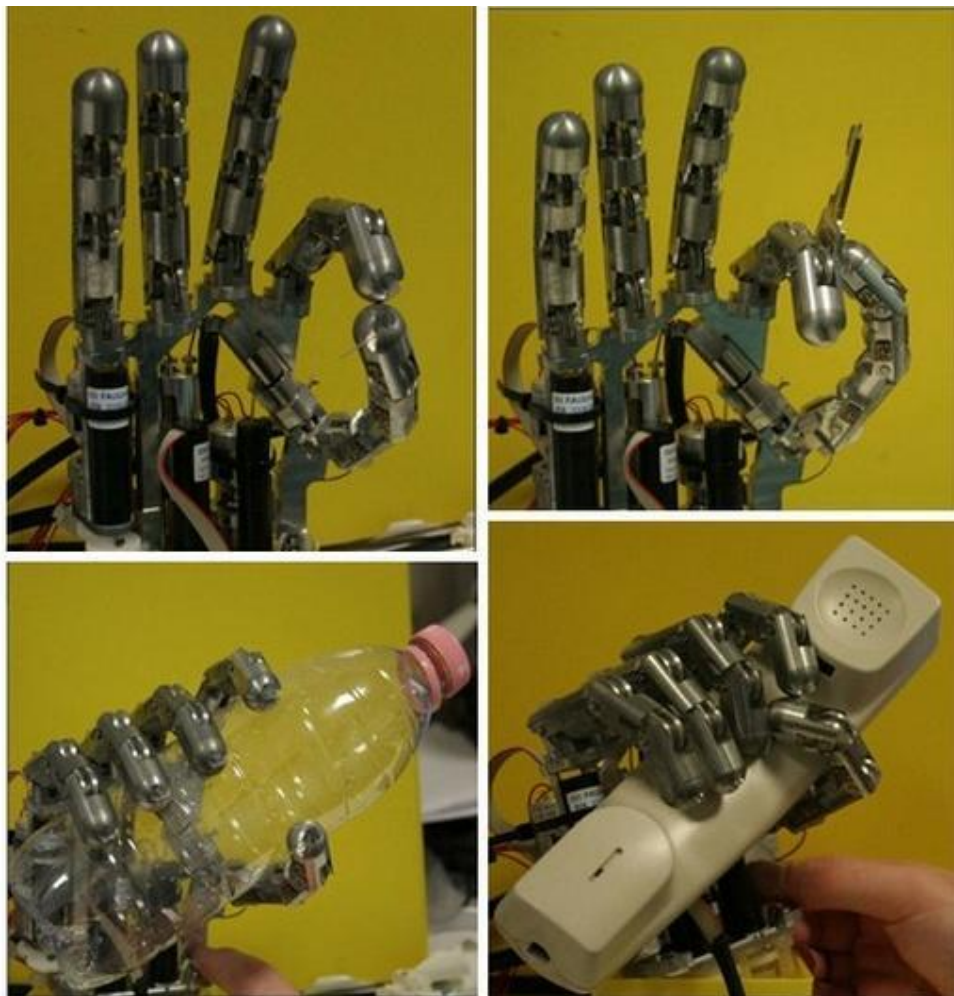


Рисунок 1.5 - Smart Hand [13]

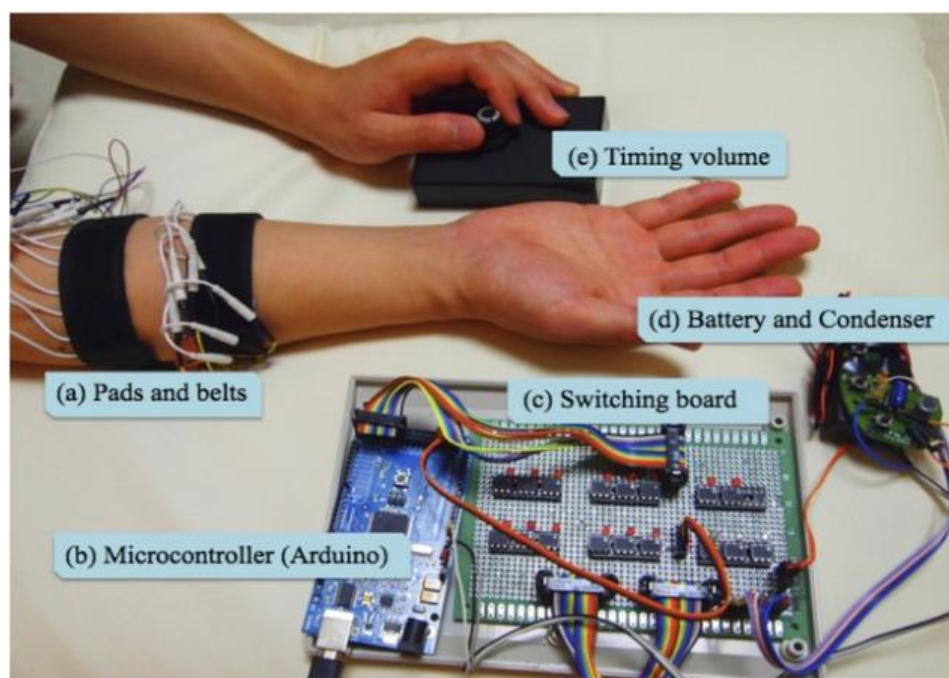


Рисунок 1.6 – Possessed Hand [13]

Показано, що фахівцями лабораторії прикладної фізики Університету Джона Хопкінса вже створено біонічний протез руки – один із найдорожчих протезів світу (6 млн дол). Власник цього протеза при його використанні може обертати ним на  $360^\circ$ , повертати кисть, а також відчувати дотики, кінчиками пальців розрізняти структуру поверхні й навіть температуру об'єкта. Усе це досягається за допомогою нейроінтерфейса – цей метод дає значно більшу повноту відчуттів і можливостей управління, порівняно з використанням датчиків у міоелектричних протезах. Фінансування здійснюється агенством DARPA – підрозділом Пентагону.

Як показали проведені дослідження, існує багато викликів в проектуванні біонічних або міоелектричних протезів [15-17], тому розробка більш дешевих та доступних роботизованих протезів залишається актуальною.

### 1.3 Аналіз критеріїв та правил розробки кіберпротезів

В роботі [17] вказано, що для проектування та виготовлення якісного кіберпротезу конструкторові необхідно забезпечити виконання цілого ряду вимог, таких як: точність руху, наявність достатньої кількості ступенів свободи, інтуїтивно зрозумілий інтерфейс, зовнішня схожість виробу зі здоровою кінцівкою, відповідні вага та габаритні розміри, максимальна енергонезалежність та ін.

В роботі [1] розглядається наступний набір критеріїв (рис. 1.7): вага руки [2-3], розміри та анатомічно правильна кінематика пальців, тип приводу та приводний механізм, сила та швидкість захоплення, довговічність та зовнішній вигляд [4].

Людська рука має середню вагу близько 400 г, що в свою чергу становить 0,5-0,6% від загальної ваги тіла. Однак кіберфізичні протези такої ваги вважаються носіям занадто важкими. Це відбувається через те, що сама структура пристрою прикріплюється до м'яких тканин, а не на скелет, що в свою чергу впливає на сприйняття ваги.

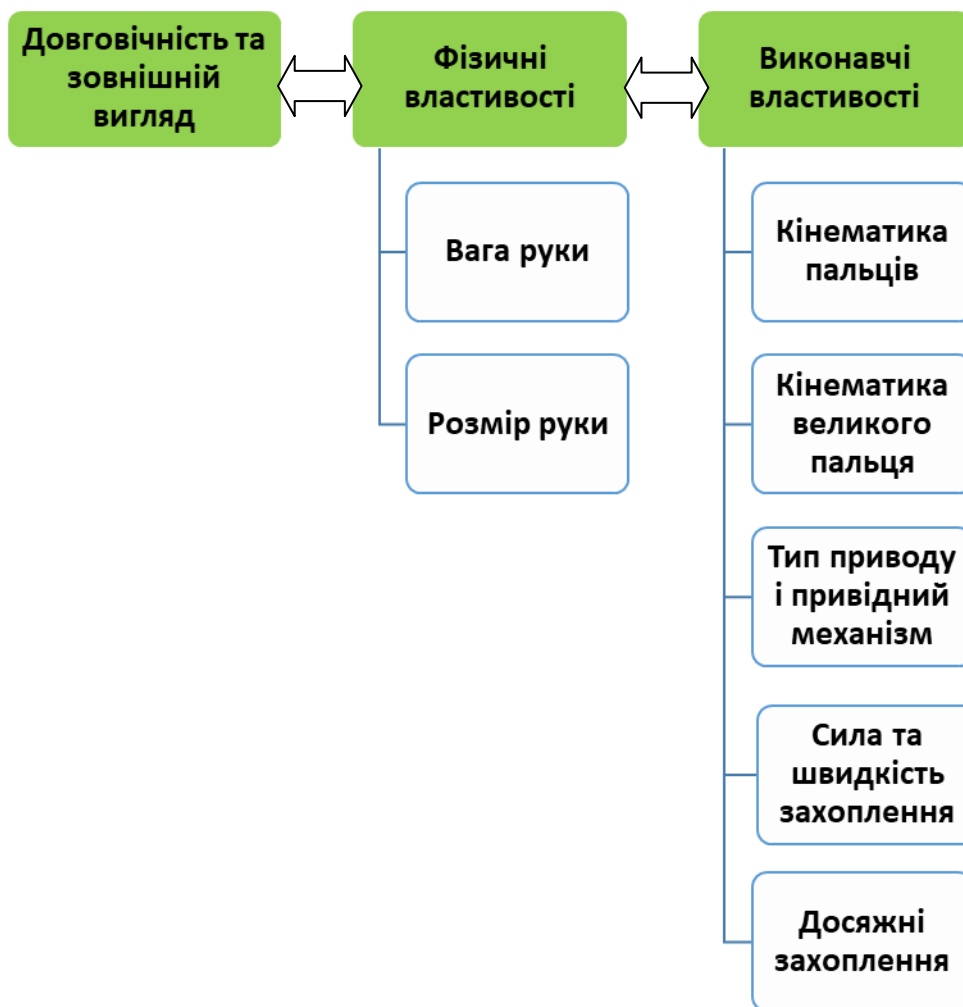


Рисунок 1.7 – Критерії розробки кіберпротезу

Більшість дослідників на даний момент працюють над використанням полегшених матеріалів за умови забезпечення міцності, а також над конструкцією місця прикріплення до кінцівки людини. Важливу роль відіграє й розподіл самих компонентів, а саме виконавчих механізмів і батареї всередині протеза, адже чим ближче до основи, тим легшою відчувається вага [1].

Природним для користувача протезу є те, що оболонка руки повторює природні розміри і форму. Однак для косметичного виду протез покривається спеціальною силіконовою рукавичкою. Тому розміри самої механічної конструкції повинні бути скореговані з урахуванням цього фактору, оскільки структура протезів рук, включаючи косметичне покриття, повинна мати

довжину від 180 до 198 мм і ширину від 75 до 90 мм, що відповідає нормальному розміру руки людини [1].

Метою механічного дизайну протеза руки є анатомічно правильна кінематика пальців. Однак, треба знайти компроміс між анатомічною правильністю та надійністю, вагою, складністю та вартістю. Окремий набір рухів, які можуть бути описані одним параметром, вважається прикладом жорстко пов'язаної руки. Іншим способом з'єднання є адаптивне відключення, при якому один привід управляє декількома незалежними ступенями свободи. Один параметр виконавчого механізму не може використовуватися для опису положення суглобів, оскільки вони залежать від стану контакту кожного зв'язку пальця з об'єктом. Ще один спосіб з'єднання полягає в адаптивній недорозвиненості, при якій один виконавчий механізм управляє рядом незалежних ступенів свободи. У цьому сенсі один параметр приводу не може бути використаний для опису положення суглобів, оскільки вони залежать від стану контакту кожної ланки пальця з об'єктом. Приклади адаптивних конструкцій пальців в протезуванні включають одне сухожилля, спрямоване через кілька суглобів або сумісні пружинні з'єднання [1].

Конструкція великого пальця в протезі руки має вирішальне значення, оскільки на великий палець припадає приблизно 40% всієї функціональності руки людини. У більшості протезних рук великий палець приводиться в рух при згинанні/розгинанні (просте закривання або відкривання) і уздовж осі кондукції. Кругове обертання великого пальця – це рух, необхідний для чергування між бічним захопленням і потужним або точним захопленням. Проведений у статті-огляді Т. Белтера [1] аналіз кінематики людської руки показує, що середній рух при круговому русі становить  $90,2^\circ$ , що досягається за допомогою комбінації трьох суглобів біля основи великого пальця.

Найпоширенішим приводом, що використовується в даний час в протезуванні, за винятком ременів з приводом від тіла, є двигун постійного струму. Ці мотори маленькі, легкі та можуть бути упаковані в кібер-руці.

Мотори постійного струму з щіткою частіше використовуються в протезах через простоту управління. Безщіткові двигуни постійного струму забезпечують більш високі характеристики крутного моменту і ваги, але вимагають більш складних схем управління двигуном. Безщіткові двигуни зазвичай включають датчики, які можуть забезпечувати додатковий зворотний зв'язок по положенню. Більш того, оскільки електроніка управління продовжує зменшуватися в розмірах, безщіткові двигуни постійного струму, ймовірно, стануть домінуючим вибором двигунів. Всі двигуни постійного струму створюють надмірну швидкість і недостатній крутний момент для використання в протезних пристроях. Отже, зменшення приводу необхідно для зменшення швидкості і збільшення крутного моменту, що забезпечується приводом [1].

Через екстремальні обмеження у пакуванні, накладених розміром руки, маленькі двигуни, які включають високі передачі, поміщаються або в проксимальну фалангу (як в руках iLimb, iLimb Pulse, Vincent) або, якщо є доступ, в долоню (як в Bebionic, Bebionic v2 і руках Michelangelo) [1].

У повсякденному житті висока сила захоплення особливо не затребувана, проте бувають моменти, коли потрібно докласти достатньо зусиль для виконання дій (наприклад, відкручування кришки банки, повертання дверної ручки). Сила захоплення, яку можна прикладати рукою до об'єкта, багато в чому залежить від положення руки, геометрії об'єкта та методу передачі. Необхідну силу захоплення для утримання об'єкта в межах конкретного захоплення також складно передбачити, оскільки вона в значній мірі залежить від тертя між пальцями руки і об'єктом, кількості точок контакту, відносно місця контакту та об'єкту, а також геометрії та маси. Людська рука може прикладати в середньому 95,6 Н сили [1].

Хоча людська рука може демонструвати швидкість згинання пальця в 2290 °/с, типові швидкості для повсякденних завдань зі збору та розміщення складають від 172 до 200 °/с [1].

Типові дії, що проводяться людиною з ампутованою кінцівкою, можуть бути виконані з використанням кінцевого набору попередньо визначених захоплень. Частина дослідників вважають, що деякі жести важливі (наприклад, підрахунок пальців), а також здатність кожної руки формувати схеми схоплювання без урахування сил контакту з об'єктом [1].

Середній користувач міоелектричного протеза руки буде носити свій пристрій більше 8 годин на день. Тому будь-який пристрій повинен бути достатньо міцним, щоб витримувати тривале використання, і досить зручним для користувача, щоб носити його протягом цього періоду часу. Механічна конструкція руки повинна враховувати компроміс між міцністю й надійністю з вагою, витратою і розміром. Такі сумісні компоненти, як відповідні накладки для пальців/долонь, сумісні конструкції приводів, розбірні системи підключення додають надійності та функціонування, не додаючи при цьому складності. В цілому, підвищення надійності зазвичай вимагає зменшення складності (кількість супинів свободи, кількість компонентів) та збільшує розмір і вагу [1].

Таким чином, компроміс між характеристиками дизайну в основному вирішує проєктувальник через суворі обмеження за розміром і вагою для протезів рук. Йому періодично належить порівнювати вагу кожної руки та кількість приводів, що були використані, порівнювати вагу руки й точність захоплення всіх рук, знаходити вихід з проблеми множинного введення, множинного виведення сигналів, порівнювати особливості і характеристики з практичною функціональністю, а також робити постійні поліпшення і ітерації дизайну [1].

Як показали проведені дослідження, можливо виділити наступні емпіричні правила для проєктування кіберпротезів рук:

- загальна вага протеза (включаючи механізм, рукавичку, електроніку тощо) повинна бути нижче 500 г; менша вага особливо бажана для людей з високою ампутацією через обмеження сили та ваги всієї протезної руки;

- наразі прості та надійні кінематичні конструкції пальців віддають перевагу анатомічно правильним конструкціям пальців;
- сильна адукція великого пальця вкрай бажана, оскільки дозволяє здійснювати активний контроль постави, наприклад, перехід від бічного передлежання до долонного попереднього;
- використання безщіткових двигунів замість щіткових двигунів є кращим, зважаючи на продуктивність та вагу;
- рекомендується максимальна сила прищипування на кінчику пальця 65 Н, мінімально прийнятна швидкість згинання пальців 115 °/с, а бажана 230 °/с;
- відповідність механічної конструкції протезної руки може бути досягнута різними способами (відповідні накладки для пальців/долонь, відповідні конструкції приводів, розбірні системи зчеплення, сумісні з'єднання, тощо);
- високофункціональні протези повинні бути сконструйовані з малою кількістю виконавчих механізмів з передачами, що дозволяють приймати всі функціональні позиції захоплення.

#### **1.4 Постановка задач магістерської роботи**

Дослідження показали, що кіберфізичні протез вже сьогодні використовуються в якості функціональних замінників втрачених кінцівок людини. Протези нового покоління покликані не лише відновлювати природний зовнішній вигляд пацієнта, але й заповнювати найважливіші втрачені функції. З кожним роком ці протези наближають пацієнтів до повноцінної заміни втраченої кінцівки. Вже відомі певні рішення для кіберпротезів: біонічні кисті, пальці, міоелектричні руки-протези та ін.

Але залишається багато проблем, пов'язаних з забезпеченням вимог до проєктованих виробів через їх складність та необхідність врахування

індивідуальних особливостей пацієнтів. Тому, удосконалення процесу автоматизованого проєктування кіберпротезів на основі сучасних технологій прототипування є актуальною задачею.

Метою роботи є розробка та дослідження віртуального та фізичного прототипів кіберпротезу верхньої кінцівки людини для підвищення ефективності процесу автоматизованого проєктування.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- виконати аналіз критеріїв розробки кіберпротезу;
- дослідити особливості існуючих технологій прототипування;
- розглянути доступні апаратні та програмні засоби для створення прототипу кіберпротезу;
- розробити метод комплексного віртуального та фізичного прототипування кіберпротезу;
- розробити інформаційну технологію створення механічної частини прототипу кіберпротезу;
- розробити фізичну та електронну частини прототипу кіберпротезу;
- розробити та протестувати програмне забезпечення для реалізації базових захоплень на основі кіберпротезу.

## 2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ТЕХНОЛОГІЙ ПРОТОТИПУВАННЯ

### 1.1 Огляд особливостей прототипування

Прототипування – це створення схеми або готового макету призначеного для користувача інтерфейсу, швидка розробка «чорнової» реалізації базового функціоналу системи для її аналізу в цілому [18]. Прототип дозволяє зберегти час та гроші, які були б витрачені на розробку неперевірених рішень. На цьому етапі малими зусиллями створюється працююча система, яка може бути з помилками, з розкритим не в повній мірі функціоналом. Головне представити в найбільш повному вигляді майбутню систему та виявити найважливіші елементи для подальшої розробки. Після етапу прототипування обов'язково йдуть етапи перегляду архітектури системи, розробки, реалізації та тестування кінцевого продукту.

Прототипування не обов'язково виконується в рамках тих самих технологій, що й розроблювана система. Як правило, прототип стає додатком до технічного завдання [18]. Ефективний прототип повинен володіти такими якостями:

- етап створення прототипу не повинен бути затяжним;
- ефективні прототипи є одноразовими. Вони призначені для того щоб донести ідею до зацікавленої особи; після того, як ідея була донесена, прототип може бути відкинутий;
- ефективні прототипи є сфокусованими, це означає що слід звертати увагу на складні частини при створенні прототипів; необхідно перевірити ергономічність системи (юзабіліті-тестування), тобто виконати дослідження з метою визначення зручності деякого штучного об'єкту (веб-сторінка, користувацький інтерфейс або пристрій) для його подальшого застосування [19];

– необхідно звертати увагу на елементи взаємодії, які принесуть користь проєктованому продукту.

Процес створення прототипу складається з чотирьох кроків. Першим кроком є визначення вимог. Другим кроком – розробка першого (ескізного) варіанту прототипу. Третій крок – демонстрація та критична оцінка (вивчення прототипу замовником і кінцевим користувачем, отримання зворотнього зв'язку про необхідні зміни та доповнення). Четвертий крок – прототипування (переробка прототипу з урахуванням отриманих зауважень і пропозицій), після чого кроки 3 та 4 можуть повторюватись [18,20]. Головною метою є швидкість створення прототипу. Скорочення тривалості виконання перших трьох кроків дозволить підвищити продуктивність та ефективність прототипування. На рисунку 2.1 схематично зображено процес протипування [20].

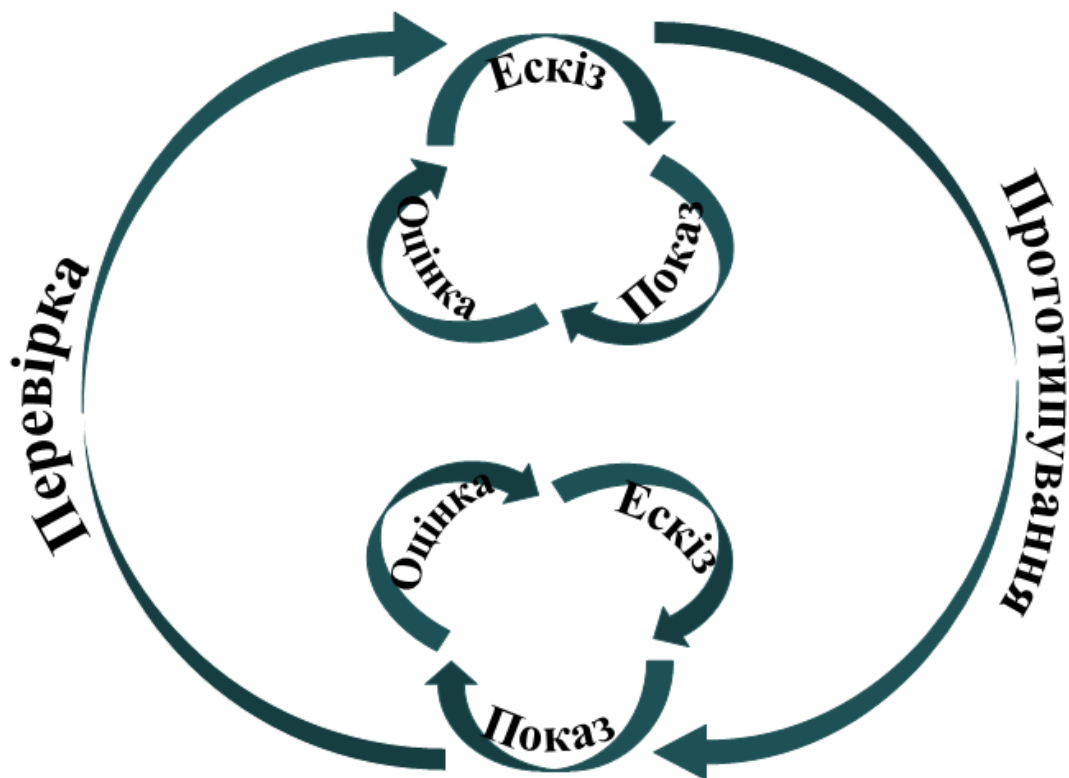


Рисунок 2.1 – Ітераційний процес прототипування

На етапі прототипування виявляються важливі архітектурні помилки, вносяться поправки в інтерфейси модулів системи і перерозподіляється функціональність між модулями системи. Після затвердження прототипу переходять до інших етапів створення системи, при чому на тестування і налагодження системи витрачають порівняно більшу кількість зусиль. Такий підхід використовується в машино- і приладобудуванні, програмуванні і в багатьох інших областях техніки.

В умовах скорочення термінів та фінансування виробництва, при вимогах дотримання якості кінцевої продукції прототипування стало невід'ємним етапом сучасного процесу проєктування [21].

Найчастіше термін «прототипування» пов'язують з термінами «швидке прототипування» та «3D-принтинг». Але, як свідчать проведені дослідження, на сьогоднішній день прототипування - це назва декількох суміжних технологій, які активно використовуються при проєктуванні об'єктів та процесів [6-7].

Наприклад, автори [7] описують декілька типів прототипування:

- прототипування вбудованих пристроїв;
- прототипування фізичного дизайну;
- прототипування Інтернет-компонентів.

При розробці кіберпротезів мають застосовуватись як віртуальне так і фізичне прототипування, відповідно до особливостей проєктування взаємопов'язаних електронної, механічної та програмної частин проєкту.

Для створення й дослідження віртуальних прототипів схем та конструкцій кіберпротезів можуть використовуватися сучасні системи автоматизованого проєктування та моделювання (Altium Designer, AutoCAD, EAGLE, PROTEUS, Tinkercad Circuits Arduino, PTC Creo, Solid Edge, Autodesk Inventor та ін.).

Але, у той же час, більшість дизайнерів погоджуються, що створення фізичного прототипу має часто вирішальне значення для реалізації концепцій дизайну. Чим швидше дизайнери експериментують з новими продуктами, тим

швидше вони отримують потрібні дані для подальших змін у проекті. Крім того, створення фізичного прототипу – це зручний спосіб отримання зворотнього зв'язку з замовником [5].

## **2.2 Технології розробки віртуального прототипу**

### **2.2.1 Прототипування електронної частини кіберпротезу**

Для прототипування електронної частини вбудованих та кіберфізичних систем використовуються програми схемотехнічного моделювання та програми-емулятори. Наприклад, при роботі з платами Arduino зручно використовувати Tinkercad Arduino, який є онлайн емулятором. Можна скористатись пакетом програм для автоматизованого проєктування електронних схем – Proteus Design Suite, який працює з платами Arduino та Raspberry Pi, або використати Atmel Studio – інтегроване середовище розробки від компанії Atmel для розробки застосунків під мікроконтролери AVR та ARM Cortex-M.

Tinkercad Circuits Arduino – безкоштовний, простий і одночасно потужний емулятор Arduino, розроблений у вигляді web-сервісу, який дозволяє працювати з ним онлайн та є його головною перевагою перед іншими системами зі схожим функціоналом. Він надає зручне середовище для написання своїх проєктів [22]. Основним функціоналом Tinkercad Arduino Circuits є наявність (рис. 2.2):

- візуальних редакторів для схем та коду;
- набору моделей більшості популярних електронних компонентів, що згруповані за типами цих компонентів;
- режими налагодження та симуляції схем;
- готові для розгортання проєкти Arduino з схемами та кодом;
- можливість експорту отриманих скетчів і електричних схем в реальні проєкти;
- можливість інтеграції з рештою функціональністю Tinkercad і швидкого створення для розроблюваного пристрою корпусу та інших конструктивних елементів – намальовані модель можна відразу відправити на 3D-принтер;

– вбудовані підручники й величезне співтовариство з колекцією готових проєктів [22-23].

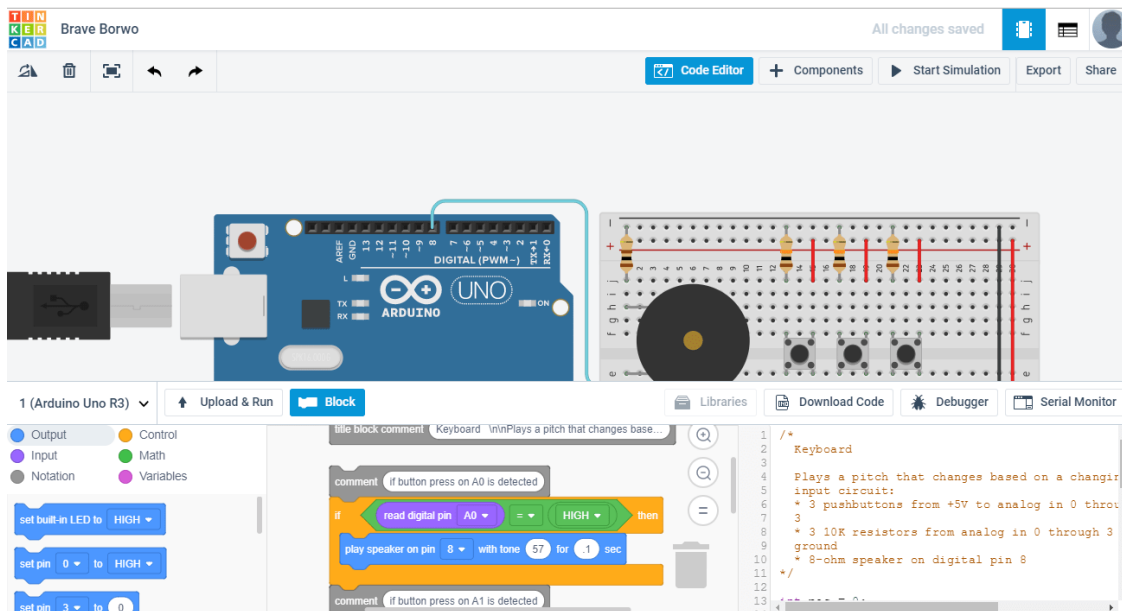


Рисунок 2.2 – Робоча область Tinkercad Circuits Arduino [22]

Tinkercad є зручною онлайн платформою, особливо для новачків, оскільки не потрібно завантажувати Arduino IDE (Integrated Development Environment), шукати і завантажувати популярні бібліотеки і скетчі, збирати схему і підключати плату – все, що потрібно, знаходиться відразу на одній сторінці, а для коректної роботи необхідний тільки браузер та якісний Інтернет.

Proteus – середовище для проєктування, моделювання віртуальної системи та схеми для налагодження електронних пристроїв, в тому числі і виконаних на основі мікроконтролерів різних сімейств (рис.2.3). Надає можливості введення схеми в графічному редакторі, моделювання її роботи і розробки друкованої плати, включаючи тривимірну візуалізацію її складання [24].

Комплекс поєднує моделювання схеми SPICE в змішаному режимі, анімовані компоненти та мікропроцесорні моделі для полегшення спільного моделювання повних конструкцій на основі мікроконтролерів. Proteus також

має можливість імітувати взаємодію між програмним забезпеченням, що працює на мікроконтролері, та будь-якою аналоговою чи цифровою електронікою, підключеною до нього. Він імітує порти вводу / виводу, переривання, таймери, USARTs та всі інші периферійні пристрої, наявні на кожному підтримуваному процесорі [25].

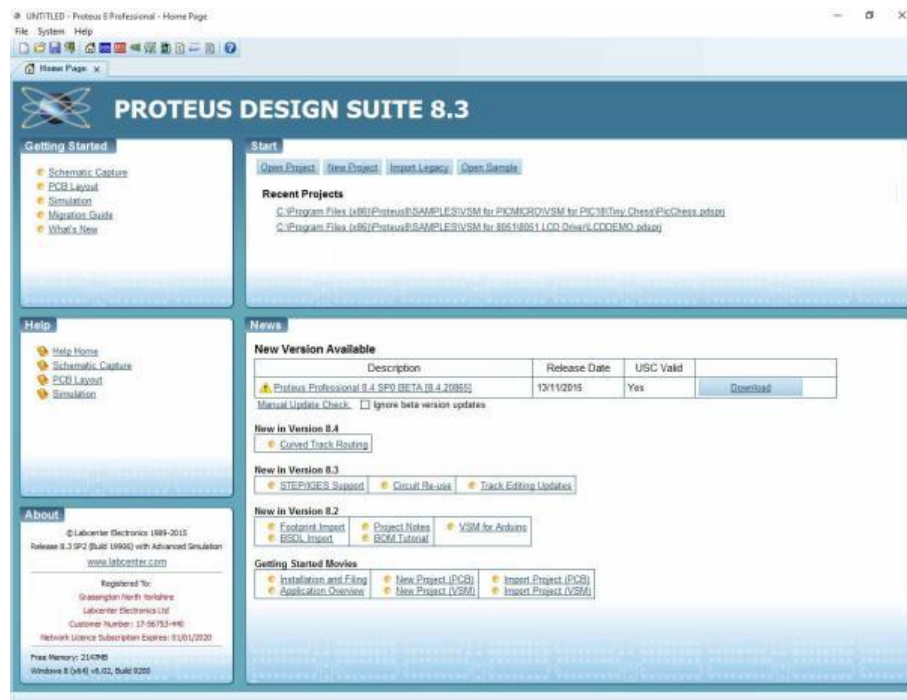


Рисунок 2.3 – Головне вікно Proteus Design Suite [24]

Також Proteus має ряд переваг:

- виконання всіх етапів розробки електронного пристрою на основі мікроконтролера в єдиному середовищі;
- можливість написання, налагодження і тестування вбудованого програмного забезпечення ще до фізичного виготовлення дослідного зразка системи;
- генеруються діагностичні повідомлення (наприклад, при виконанні непередбачуваної інструкції) як з боку CPU, так і з боку моделей пристроїв введення-виведення, дозволяють виявити складні в пошуку помилки програмування;

- прискорює процес розробки електронного пристрою;
- підтримка спільної роботи з апаратними пристроями, підключеними через порт комп'ютера.

Atmel Studio (раніше AVR Studio) – безкоштовне IDE від компанії Atmel для розробки застосунків під мікроконтролери ARM Cortex-M та AVR, що засноване на Visual Studio.

Програма містить в собі майстер проєктів, віртуальний симулятор, редактор вихідного коду, модуль внутрішньо схемного налагодження і інтерфейс командного рядка, компілятор GNU C/C++ і емулятор, що дозволяє налагодити виконання програми без завантаження в мікроконтролер, дозволяє працювати з мовами програмування Асемблер та C/C++. Підтримує компілятор GCC і плагін AVR RTOS (операційної системи реального часу). Користувачі можуть вибрати найбільш оптимальні для їх проєкту способи кодування. Візуальні інструменти дозволяють прискорити написання програми, на рисунку 2.4 зображено домашню сторінку програми [26].

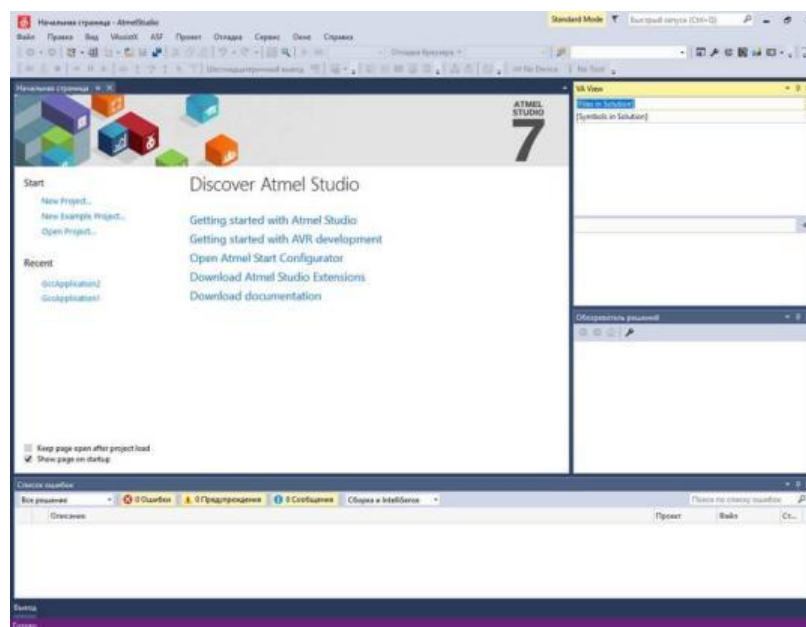


Рисунок 2.4 – Домашня сторінка Atmel Studio [26]

Обираючи Atmel Studio для розробки власного проєкту користувач отримує ряд ключових особливостей системи:

- підтримка 500 + AVR і SAM пристроїв;
- велика бібліотека вихідного коду, включаючи драйвери, зв'язок стеки, 1600 + прикладів проєктів з вихідним кодом, графічні послуги і функціональність Advanced Software Framework (ASF);
- майстер проєктів, що дозволяє створювати проєкти з нуля або з великої бібліотеки прикладів дизайну;
- розширення IDE через галерею Atmel, Інтернет-магазин застосунків, засоби розробки та вбудоване програмне забезпечення від Microchip;
- можливість налаштувати та протестувати продуктивність бездротових конструкцій за допомогою бездротового композитора;
- запис і налагодження C/C++ і складання коду з вбудованим компілятором;
- додаткові функції налагодження включають складні точки переривання даних, підтримка трасування (пристрої SAM3 та SAM4), статистичне профілювання коду, відстеження переривання / моніторинг, відстеження даних опитуваних даних (пристрої Cortex-M0 +), відстеження змінної в режимі реального часу з додатковою міткою часу;
- можливість при підтримці плагіни підключення компілятора GCC у вигляді WinAVR;
- створення прозорих подань налагодження в CPU та периферійних пристроїв для легкого розвитку коду та налагодження;
- повний чіп моделювання для точної моделі CPU процесора, переривання, периферійних пристроїв і зовнішніх подразників [27].

Завдяки зв'язці програмних пакетів Atmel Studio і Proteus від фірми Labcenter Electronics можливе програмування мікроконтролерів без наявності будь-якої матеріальної бази. Atmel Studio по праву вважається найкращим

середовищем створення застосунків для контролерів AVR. А у випадку, якщо не має можливості встановлювати програмне забезпечення, то можна використати Tinkercad Circuits Arduino та створювати проєкти онлайн.

### 2.2.2 Прототипування механічної частини кіберпротезу

Побудова віртуального прототипу механічної частини кіберпротезу зазвичай відбувається на основі твердотільної моделі з CAD-систем (Computer Aided Design) або моделі з замкнутими поверхневими контурами [28].

В даний час на ринку присутні найрізноманітніші сучасні CAD-системи, які відрізняються між собою як за функціональністю, так і за вартістю. Вибрати відповідну систему автоматизованого проєктування (САПР) - непросте завдання. При ухваленні рішення необхідно орієнтуватися на потреби підприємства, завдання, які стоять перед користувачами, вартість придбання й підтримки системи та багато інших чинників.

Деякі середовищі проєктування є спеціалізованими для використання в області медицини, але більшість є універсальними.

Легкі системи САПР призначені для 2D-проєктування та креслення, а також для створення окремих тривимірних моделей без можливості роботи зі складальними одиницями.

Безумовний лідер серед базових САПР - AutoCAD. AutoCAD - це базова САПР, що розробляється та поставляється компанією Autodesk. AutoCAD є найпоширенішою CAD-системою в світі, що дозволяє проєктувати як в двовимірному, так і в тривимірному середовищі. За допомогою AutoCAD можна будувати 3D-моделі, створювати і оформляти креслення і багато іншого. AutoCAD є універсальною САПР, тобто ця система не має чіткої орієнтації на певну проєктну область, в ній можна виконувати різні проєкти [29].

Система автоматизованого проєктування AutoCAD має наступні особливості:

- стандарт "де факто" в світі САПР;
- широкі можливості налаштування й адаптації;
- засоби створення застосунків на вбудованих мовах (AutoLISP та ін.) і з застосуванням API (Application Programming Interface);

- велика кількість сумісних програм від інших виробників.

В даний час на ринку з'явився цілий ряд систем, які позиціонуються, як альтернатива AutoCAD. Серед них можна окремо відзначити Bricscad від компанії Bricsys, яка дуже активно розвивається, підтримує безпосередньо формат DWG і має цілий ряд відмінностей, включаючи інструменти прямого варіаційного моделювання, а також підтримку BIM-технологій [29].

Середні системи САПР - це програми для 3D-моделювання виробів, проведення розрахунків, автоматизації проєктування електричних, гідравлічних та інших допоміжних систем. Дані в таких системах можуть зберігатися як у звичайній файловій системі, так і в єдиному середовищі електронного документообігу та управління даними (PDM- і PLM-системах). Часто в системах середнього класу присутні програми для підготовки керуючих програм для верстатів з ЧПУ (CAM-системи) і інші програми для технологічного проєктування.

САПР середнього рівня – найпопулярніші системи на ринку. Вони вдало поєднують в собі співвідношення "ціна / функціональність", здатні вирішити переважну кількість проєктних завдань і задовільнити потреби більшої частини клієнтів.

Autodesk Inventor. Професійний комплекс для тривимірного проєктування промислових та медичних виробів і випуску документації. Розробник - компанія Autodesk [29].

Серед особливостей Inventor варто відзначити:

- просунуті інструменти тривимірного моделювання, включаючи роботу з вільними формами і технологію прямого редагування;
- підтримку прямого імпорту геометрії з інших САПР зі збереженням асоціативного зв'язку (технологія AnyCAD);
- тісну інтеграцію з програмами Autodesk - AutoCAD, 3ds Max, Alias, Revit, Navisworks та іншими, що дозволяє використовувати Inventor для

вирішення завдань в різних областях, включаючи дизайн, анатомічно точне проєктування та інші;

- підтримку вітчизняних стандартів при проведенні розрахунків, моделюванні та оформленні документації;
- великі бібліотеки стандартних і часто використовуваних елементів;
- широкі можливості параметризації деталей і зборок, в тому числі управління складом виробу;
- вбудовану середу створення правил проєктування iLogic.

Для ефективного управління процесом розробки виробів, управління інженерними даними та організації колективної роботи над проєктами, Autodesk Inventor може бути інтегрований з PLM-системою Autodesk Vault і схожими системами інших розробників [29].

Команда компанії Internal [30] розробила Autodesk Within Medical, програмне рішення, налаштоване на потреби ортопедичної галузі. Це програмне забезпечення дозволяє дизайнерам імплантатів створювати пористі покриття для своїх імплантатів, щоб сприяти осеоінтеграції, тобто зрощенню між кісткою та імплантатом.

Ключові риси:

- Autodesk Within Medical дозволяє дизайнерам розробляти такі частини: пористі (для сприяння осеоінтеграції з різними конфігураціями розміру пор та поверхнями решітки) та тверді (для сприяння фіксації пористого імплантату з кісткою, використовуючи шорсткі поверхні решітки);
- конструкції з безпечним виходом є надійними і можуть виготовлятися тільки певними виробниками;
- точні для адитивного виробництва - конструкції оптимізовані для конкретних виробничих процесів, таких як DMLS та EBM, дозволяючи отримати точний вихід.

Autodesk Within Medical може бути використаний для проєктування пористих імплантатів, де пористість може бути налаштована в рамках

програмного забезпечення для забезпечення оптимальної осеоінтеграції. Решітки топології, що постачаються разом з цим програмним забезпеченням, були розроблені з урахуванням росту клітин і найкраще підходять для цієї функції. Вихід програмного забезпечення Autodesk Within Medical також оптимізований для виробництва добавок [30].

SolidWorks – це програмний комплекс для автоматизації конструкторських робіт промислового підприємства від компанії Dassault Systemes. Риси системи, що вигідно відрізняють її від інших CAD-систем:

- продуманий інтерфейс користувача, який став зразком для наслідування;
- велика кількість надбудов для вирішення вузькоспеціалізованих завдань;
- орієнтація як на конструкторську, так і на технологічну підготовку виробництва;
- бібліотеки стандартних елементів;
- розпізнавання і параметризація імпортованої геометрії;
- інтеграція з системою SolidWorks PDM [29].

Solid Edge задовільняє потребу у скороченні терміну розробки інноваційних, високоякісних виробів, які відповідають вимогам, безпечних і зручних у використанні як для медичних працівників, так і для пацієнтів, пропонуючи функціональні і зручні рішення для проєктування медичних виробів. Solid Edge допомагає як невеликим, так і крупним виробникам покращити процеси проєктування і розробки і швидко випускати якісну продукцію, збільшуючи оборот коштів і підсумковий прибуток [31].

Серед переваг системи можна виділити:

- проведення дигіталізації інженерних процесів;
- підвищення продуктивності, використовуючи інструменти проєктування нового покоління;

- прискорення проєктування механічної частини, використовуючи 3D-CAD;
- візуалізація конструкторських задумів (створення фотореалістичних зображень і анімації протезів чи органів, використовуючи 3D-рендеринг);
- інтегрування електромеханічного проєктування;
- підготовка проєктів до 3D-друку;
- легке створювання моделі з листового металу;
- забезпечення відповідності державним вимогам і вимогам замовників;
- використання безпаперової документації 3D;
- тісну інтеграцію з Microsoft SharePoint і PLM-системою Teamcenter для спільної роботи і управління даними;
- спільно використовувати дані про продукт, використовуючи хмарні сховища [31].

Компас-3D - це система параметричного моделювання деталей і складань, що використовується в областях машинобудування, приладобудування та будівництва. Розробник - компанія Аскон (Росія).

Переваги системи Компас-3D:

- простий і зрозумілий інтерфейс;
- використання тривимірного ядра власної розробки (С3D);
- великий набір надбудов для проєктування окремих розділів проєкту;
- гнучкий підхід до оснащення робочих місць проєктувальників, що дозволяє заощадити при покупці;
- можливість інтеграції з системою автоматизованого проєктування технологічних процесів ВЕРТИКАЛЬ і іншими системами єдиного комплексу [29].

T-FLEX - російська САПР середнього рівня, побудована на основі ліцензійного тривимірного ядра Parasolid. Розробник системи - компанія ТопСистеми.

Відмінні риси системи:

- найпотужніші інструменти параметризації деталей і складань;
- просунуті засоби моделювання;
- простий механізм створення застосунків без використання програмування;
- інтеграція з іншими програмами комплексу T-FLEX PLM;
- інструменти розрахунку і оптимізації конструкцій [29].

Важкі САПР призначені для роботи зі складними виробами (великі збірки в авіабудуванні, суднобудуванні та ін.) Функціонально вони роблять все те ж саме, що і середні системи, але в них закладена зовсім інша архітектура і алгоритми роботи.

PTC Creo - система 2D і 3D параметричного проєктування складних виробів від компанії PTC. САПР PTC Creo широко використовується в самих різних областях проєктування систем [29].

Вигідні відмінності системи від конкуруючих рішень:

- ефективна робота з великими і дуже великими збірками;
- моделювання на основі історії та інструменти прямого моделювання;
- робота зі складними поверхнями;
- можливість масштабування функціональності системи в залежності від потреб користувача;
- різні уявлення єдиної, централізованої моделі, що розробляється в системі;
- тісна інтеграція з PLM-системою PTC Windchill.

NX. NX - флагманська система САПР виробництва компанії Siemens PLM Software, яка використовується для розробки складних виробів, що

включають елементи зі складною формою і щільною компоновкою великої кількості складових частин систем [29].

Ключові особливості NX:

- підтримка різних операційних систем, включаючи UNIX, Linux, Mac OS X і Windows;
- одночасна робота великої кількості користувачів в рамках одного проекту;
- повнофункціональним рішенням для моделювання;
- просунуті інструменти промислового дизайну (вільні форми, параметричні поверхні, динамічний рендеринг);
- інструменти моделювання поведінки мехатронних систем;
- глибока інтеграція з PLM-системою Teamcenter.

CATIA. Система автоматизованого проєктування від компанії Dassault Systemes, орієнтована на проєктування складних комплексних виробів, в першу чергу, в галузі авіабудування і суднобудування.

Відмінні особливості:

- стандарт «де факто» в авіабудуванні;
- орієнтація на роботу з моделями складних форм;
- глибока інтеграція з розрахунковими та технологічними системами;
- можливості для колективної роботи тисяч користувачів над одним проектом;
- підтримка міждисциплінарної розробки систем [29].

Останнім часом активно почали розвиватися "хмарні" САПР, які працюють у віртуальній обчислювальній середовищі, а не на локальному комп'ютері. Доступ до цих САПР здійснюється або через спеціальний застосунок, або через звичайний браузер. Незаперечна перевага таких систем - можливість їх використання на слабких комп'ютерах, так як вся робота відбувається в "хмарі".

Хмарні САПР активно розвиваються, і якщо кілька років тому їх можна було віднести до легких САПР, то тепер вони міцно влаштувалися в категорії середніх САПР.

САПР Fusion 360 орієнтована на вирішення широкого кола завдань, починаючи від простого моделювання і закінчуючи проведенням складних розрахунків. Розробник системи - компанія Autodesk. Особливості Fusion 360:

- високий рівень інтерфейс користувача;
- поєднання різних методів моделювання;
- просунуті інструменти роботи зі складками;
- можливість роботи в онлайн і оффлайн режимах (при наявності і відсутності постійного підключення до мережі Інтернет);
- доступна вартість придбання та утримання;
- розрахунки, оптимізація, візуалізація моделей;
- вбудована САМ-система;
- можливості прямого виведення моделей на 3D-друк.

Повністю "хмарна" САПР Onshape розробляється компанією Onshape. На що варто звернути увагу при виборі Onshape:

- доступ до програми через браузер або мобільні застосунки;
- робота тільки в режимі онлайн;
- вузька спрямованість на машинобудівне проєктування;
- повний набір функцій для моделювання виробів машинобудування;
- контроль версій створюваних проєктів;
- підтримка мови FeatureScript для створення власних застосунків на основі Onshape [29].

Як показали проведені дослідження, найбільш популярними є середовища AutoCAD та PTC Creo. В таблиці 2.1 наведено порівняння основних функціональних можливостей цих систем [29, 32].

### **2.2.3 Побудова візуальної моделі цифрового двійника**

Цифровий двійник складається з візуальної моделі досліджуваного об'єкта та поведінкової моделі. Він є реалістичною віртуальною моделлю досліджуваного об'єкта – фізичного двійника. Цифровий двійник-екземпляр створюється для фізичного об'єкта, що вже існує, наприклад, здорова кінцівка людини.

Таблиця 2.1 – Опис основних можливостей систем САПР

| Назва САПР | Вид САПР    | Можливості                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Особливості                                                                                                                                                                      |
|------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AutoCAD    | Легкої ваги | <p>Дозволяє проєктувати в двовимірної, та тривимірному середовищі.</p> <p>Можна будувати 3D-моделі. Створювати і оформляти креслення.</p> <p>Широкі можливості налаштування і адаптації.</p> <p>Засоби створення застосунків за допомогою вбудованих мов (AutoLISP та ін.) з застосуванням API</p> <p>Велика кількість програм інших виробників.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <p>Система не має чіткої орієнтації на певну проєктну область, в ній можна виконувати будь-які будівельні проєкти, працювати з дослідженнями, електрикою і багато чим іншим.</p> |
| PTC Creo   | Тяжкої ваги | <p>Ефективна робота з великими і дуже великими збірками</p> <p>Моделювання на основі історії та інструменти прямого моделювання</p> <p>Робота зі складними поверхнями та легке виконання вимог необхідних для роботи з ними.</p> <p>Можливість масштабування функціональності системи в залежності від потреб користувача</p> <p>Тісна інтеграція з PLM-системою PTC Windchill.</p> <p>Прискорення концептуального проєктування за допомогою функцій конструювання у вільному стилі FreeStyle.</p> <p>Підвищення продуктивності, завдяки більш ефективним і гнучким засобам робочого 3D-проєктування.</p> <p>Підвищення якості моделі, багаторазове використання власних моделей і моделей частин різних CAD-систем, скорочення помилок моделей.</p> <p>Миттєве підключення до інформації та ресурсів в мережі Інтернет для підтримки високоефективного процесу розробки.</p> | <p>Система 2D і 3D параметричного проєктування складних виробів в різних сферах проєктування</p>                                                                                 |

3D-сканування дає можливість створити візуальну модель цифрового двійника досліджуваного предмета в цифровому форматі, тобто його 3D-модель, що дозволяє спростити та прискорити процес її подальшої реалізації.

В основі роботи 3D-сканера лежить принцип стерео відображення. Сканер, як і людське око, здатний визначити відстань до об'єкта і його розміри. Як у людини два ока, так і у 3D-сканера – дві камери. Після отримання необхідної інформації 3D-сканер будує 3D-модель об'єкта [33]. Для збільшення точності вимірювання під час роботи, 3D-сканер обладнаний елементами підсвітки для кожної з камер, в якості яких може застосовуватись лазер або особлива лампа (рис. 2.5) [34].

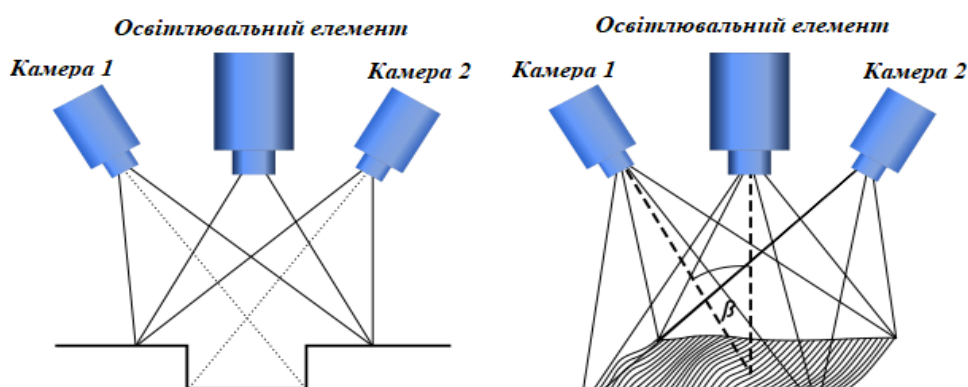


Рисунок 2.5 – Схематичне зображення камер та елементів світла на 3D-сканері [34]

Принцип роботи 3D-сканера визначається технологією сканування, в загальному сенсі – це здатність приладу визначати відстань до об'єкта, а отримані дані перетворює в цифрове зображення (тривимірну модель) та передає його на комп'ютер.

Сканер вимірює відстань до об'єкта з різних ракурсів, тобто визначає координати точок в просторі на поверхні оброблюваного об'єкта, за допомогою пари інтегрованих камер і виставленого освітлення. Потім отримані дані з камер зіставляються, та ретельно аналізуються, після чого на екрані відображається готова цифрова тривимірна модель [35, 36].

Виділяють два базових типи сканерів: контактний та безконтактний.

**Контактний 3D-сканер.** Для сканування пристрою необхідно знаходитися в безпосередньому контакті з об'єктом сканування. Пристрої з контактним принципом роботи вважаються більш точними, проте, в процесі сканування таким приладом тендітні предмети можуть бути пошкоджені.

**Безконтактний 3D-сканер.** Для сканування застосовуються випромінювання або особливий світ (ультразвук, рентгенівські промені), за допомогою яких отримують дані про зображення. Тобто сканеру необов'язково контактувати з об'єктом, що дозволяє проводити 3D-сканування важкодоступних об'єктів, пам'ятників культури і архітектури, а також ювелірних виробів [33-35].

Серед безконтактних 3D-сканерів виділяють дві найпопулярніші технології сканування, які схематично представлено на рисунку 2.6 разом з двома основними типами сканерів.

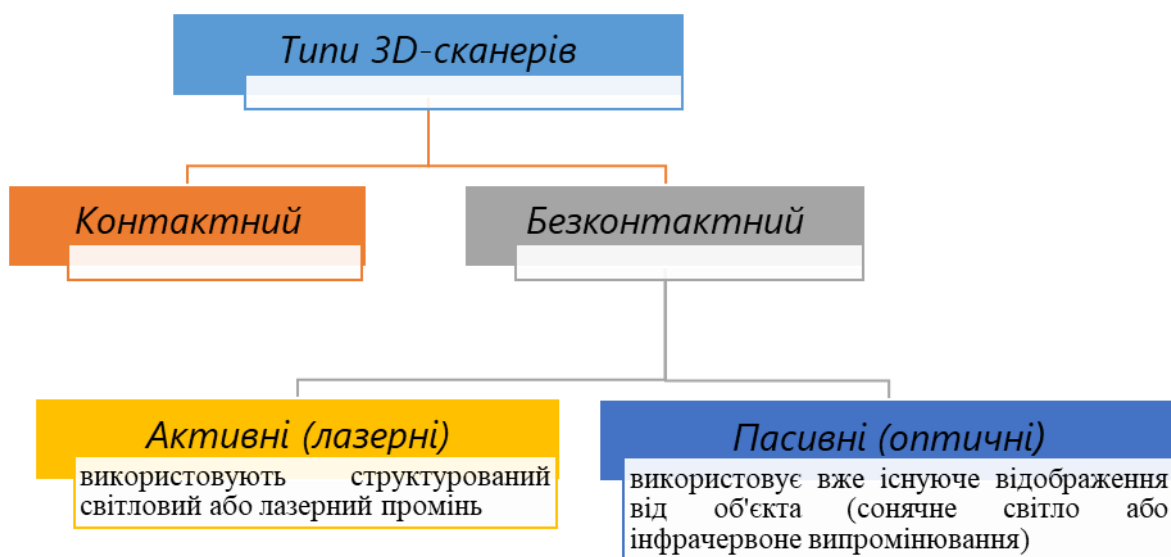


Рисунок 2.6 – Типи 3D-сканерів

**Активні безконтактні 3D-сканери (лазерні).** Для вивчення об'єкта використовують структурований світловий або лазерний промінь, який потрапляючи на об'єкт, відбивається і на основі цього відображення 3D-сканер

будує 3D-модель. Лазерні сканери характеризуються точністю одержуваної тривимірної моделі. Однак, використовувати їх для сканування об'єкта в русі є не найкращим варіантом. Ці сканери краще використовувати для статичних об'єктів, а от сканування людини сканером лазерного типу практично неможливе.

Пасивні безконтактні 3D-сканери (оптичні). Даний вид пристрою не випромінює нічого, а використовує вже існуюче відображення від об'єкта, в основному – сонячне світло або інфрачервоне (теплове) випромінювання предмета. Такі сканери працюють на зразок людського ока та відрізняються великою швидкістю сканування. Його використання виключає будь-яке спотворення, навіть якщо об'єкт буде рухатися. Пасивні сканери не підходять для дослідження дзеркальних, прозорих або блискучих виробів, однак є відмінним варіантом для сканування людини [33, 34].

До загальних переваг всіх 3D-сканерів відносять:

- максимально висока точність моделі – 3D-сканер відтворює навіть найнезначніші, дрібні, деталі фізичного об'єкта;
- висока швидкість роботи – об'ємне сканування займає всього кілька хвилин, а то й секунд, після чого необхідна доробка побудованої сканером 3D-моделі в професійних програмах для роботи з 3D-графікою;
- сканер можна розмістити під різними кутами, в залежності від складності об'єкта, при цьому сам об'єкт можна не чіпати, що особливо важливо при скануванні великих і величезних об'єктів (наприклад, будинки, пам'ятники і ландшафти).

До переваг контактних 3D-сканерів можна віднести:

- простоту у використанні;
- не залежність від рівня освітлення;
- створення моделі високої точності;
- файл 3D-моделі невеликий за обсягом.

До переваг безконтактних 3D-сканерів можна віднести:

- енерго-економічність;
- не вимагають безпосереднього контакту з об'єктом;
- застосування технології структурованого світла;
- не завдає шкоди фізичному об'єкту [33].

В медицині сканери дозволяють легко розпізнають геометрію тіла, при цьому вони не опромінюють організм, завдяки використанню структурованого світлового випромінювання. З їх допомогою розробляють ортопедичні корсети і протези, проєктуються шоломи для правильного розвитку кісток, використовуючи індивідуальні мірки, зняті з конкретного пацієнта.

Зокрема, 3D-сканування стало проривом в області вушної пластичної хірургії. Безконтактний сканер використовується для створення тривимірної моделі здорової вушної раковини. По ній створюється фотополімерна копія здорового органу – саме вона виступає в ролі шаблону в процесі хірургічного втручання [37].

Сучасні 3D-сканери – це електронні пристрої, здатні аналізувати параметри фізичного об'єкта і на основі отриманих даних створювати точну 3D-модель на екрані комп'ютера, в якості прикладу можна розглянути 3D-сканер Gotcha, виробництва 4D Dynamics, Бельгія. На рисунку 2.7 представлено його зовнішній вигляд.



Рисунок 2.7 – Зовнішній вигляд 3D-сканер Gotcha

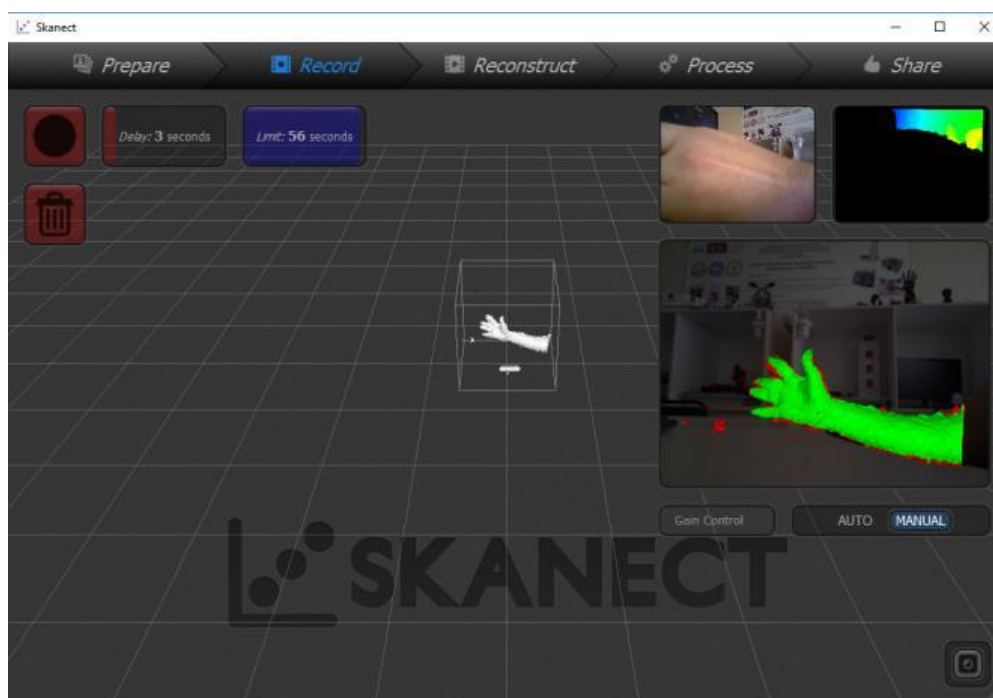
3D-сканер Gotcha є зручним та досить простим у використанні та призначений для безпечного сканування людей і їхніх облич. В ньому передбачена передача кольорової структури. Сканер має точність 1 мм та власне програмне забезпечення, що дозволяє відносно недорого здійснювати 3D-сканування з хорошим дозволом. Він є хорошим вибором як для тих, хто шукає професійний ручний 3D-сканер так для тих, хто має намір використовувати його просто в своєму хобі. В таблиці 2.3 наведено технічні характеристики сканера [32, 33].

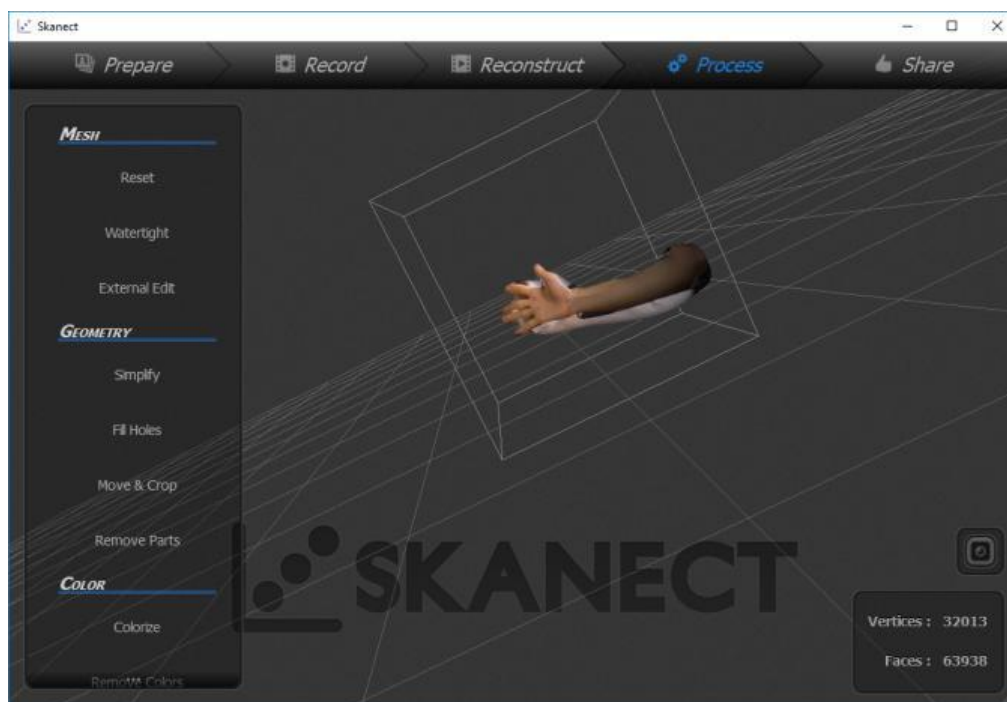
Таблиця 2.3 – Технічні характеристики 3D-сканер Gotcha

| Параметр                 | Значення      |
|--------------------------|---------------|
| Країна-виробник          | Бельгія       |
| Точність сканування (мм) | 1             |
| Метод сканування         | Безконтактна  |
| Технологія сканування    | Оптична       |
| Тип проєктора            | Інфрачервоний |

|                               |                                                |
|-------------------------------|------------------------------------------------|
| Сканування текстури           | Корольова текстура                             |
| Макс. Відстань до об'єкту (м) | 0.5                                            |
| Мин. Відстань до об'єкту (м)  | 3                                              |
| Фокусна відстань              | 3-6 мм                                         |
| Фокусування в межах           | 0,5м – 3,0м                                    |
| Частота кадрів                | 30 кадрів/сек                                  |
| Геометрія камери              | Asus Xtion Pro Live or Primesense Carmine 1.09 |
| Кількість камер               | 1                                              |
| Інтерфейс камери              | USB 2.0                                        |
| Формат даних на виході        | STL                                            |
| Програмне забезпечення        | В комплекті                                    |

На рисунку 2.8 наведено приклад сканування за допомогою 3D-сканер Gotcha в середовищі Skanect.





б

а – вікно сканування, б – вікно обробки сканованого зображення

Рисунок 2.8 – Приклади роботи зі сканером Gotcha в середовищі Skanect

### 2.3 Технології розробки фізичного прототипу

Прототипування є обов'язковим етапом розробки практично будь-якого нового виробу. Створення якісного прототипу, який би максимально був схожим на майбутній виріб є досить складною задачею, оскільки доводиться вирішувати проблеми, що виникають під час розробки, такі як точне повторення геометричних форм, зовнішнього вигляду, пошук матеріалів, які б були максимально схожими з заданими [28].

Сучасний прототип дозволяє не тільки оцінити зовнішній вигляд деталі, але і перевірити елементи конструкції, провести необхідні випробування, виготовити майстер-модель для подальшого лиття. Останнім часом популярними стали технології швидкого прототипування (RP - Rapid Prototyping), тобто пошарового синтезу макета з комп'ютерної моделі виробу.

RP-технологія – це технологія швидкого «макетування», тобто швидкого створення зразків одного типу, або працюючу модель системи для її демонстрації замовнику чи для перевірки можливості реалізації. Пізніше прототип доповнюється, редагується для отримання готового продукту [40]. Такий вид прототипування використовується для:

- оцінки ергономічності, візуалізації, дизайну виробу;
- функціональної оцінки виробів (перевірка якості складальних виробів, аеродинамічних характеристик, практичності);
- використання в якості моделі для подальшого застосування у виробництві (в якості ливарної форми, електроерозійного інструменту та ін).

Використання RP-технологій в прототипування здатне на 50 - 80% скоротити терміни підготовки виробництва, практично повністю виключити тривалий і трудомісткий етап виготовлення дослідних зразків вручну, або на верстатах з ЧПУ [28], також до переваг використання швидкого прототипування відносяться:

- скорочення тривалості технічної підготовки виробництва нової продукції в 2-4 рази;
- зниження собівартості продукції, особливо в дрібносерійному або одиничному виробництві в 2-3 рази;
- значне підвищення гнучкості виробництва;
- виявлення недоліків на ранніх стадіях розвитку;
- отримання зворотного зв'язку з користувачем;
- підвищення конкурентоспроможності виробництва;
- наскрізне використання комп'ютерних технологій, інтеграція з системами САПР.

Незважаючи на переваги такий спосіб прототипування має ряд недоліків:

- відносно висока ціна установок і витратних матеріалів;
- відносно низька міцність моделей (в залежності від матеріалу);
- час виготовлення виробів [41].

З плином часу недоліки поступово усуваються – знижуються ціни, збільшується вибір технологій і матеріалів.

Основною відмінністю між технологіями прототипування є матеріали, що використовуються для створення прототипу, а також спосіб його нанесення. В цілому, суть технології прототипування одна, але основною відмінністю між ними є тип використовуваного матеріалу та способи його нанесення.

3D-друкування це побудова реального об'єкта по створеному на комп'ютері зразку 3D-моделі. Потім цифрова тривимірна модель зберігається в форматі STL-файлу, після чого 3D-принтер, на який виводиться файл для друку, формує реальний виріб [42].

Переваги 3D-друку перед звичними, ручними способами побудови моделей – висока швидкість, простота і відносно невелика вартість.

Наприклад, для створення 3D-моделі або будь-якої деталі вручну може знадобитися досить багато часу – від кількох днів до місяців. Адже сюди

входить не тільки сам процес виготовлення, а й попередні роботи – креслення і схеми майбутнього виробу, які все одно не дають повного бачення остаточного результату. В результаті значно зростають витрати на розробку, збільшується термін від розробки виробу до його серійного виробництва [43].

Існує значна кількість технологій 3D-друку, але найбільш популярною є технологія пошарового накладення розплавленої полімерної нитки (FDM – Fused Deposition Modeling), схему якої наведено на рисунку 2.9 [28].

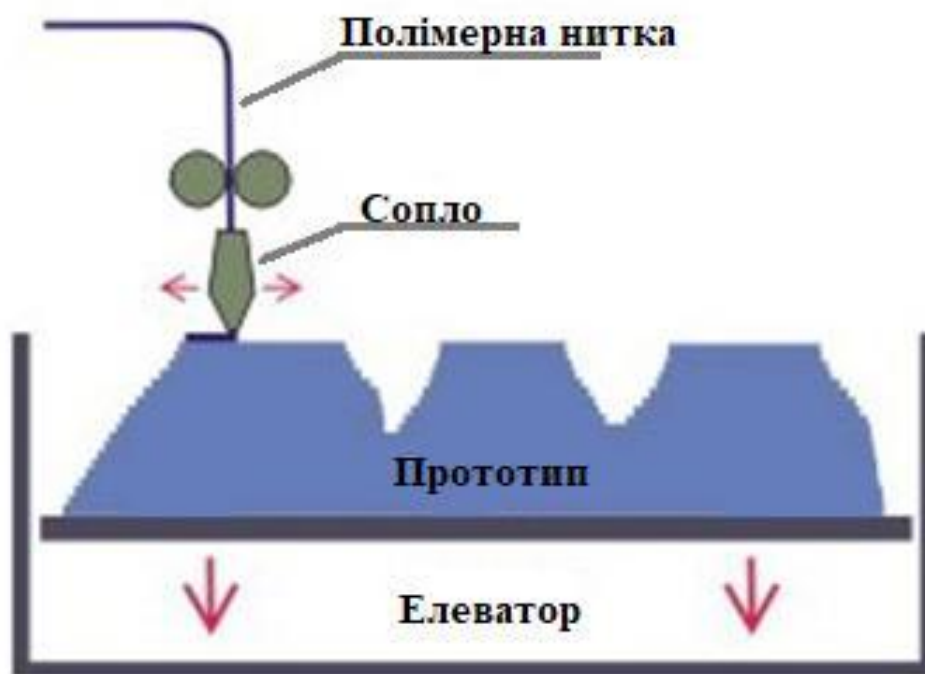


Рисунок 2.9 – Схема пошарове накладення розплавленої полімерної нитки [28]

Технологія друку тривимірного об'єкту, в свою чергу, дає можливість відмовитися від ручної праці в повній мірі. В результаті застосування 3D-технології, немає потреби в кресленнях і розрахунках на папері. Завдяки програмі, модель можна буде побачити на моніторі з усіх боків і прибрати похибки і недоліки не під час фінальної стадії, як це відбувається при ручному варіанті, а прямо при виробництві. А на всю роботу буде витрачено кілька годин, що у часовому еквіваленті буде швидше в  $n$ -ну кількість разів.

Також, варто додати, що помилки, що допускаються при ручній роботі, виключені при застосуванні 3D-технології [42].

Для побудови готової моделі на принтері необхідно спочатку створити 3D-модель об'єкта за допомогою програми CAD (наприклад Creo, AutoCAD) і зберегти в спеціальному форматі STL. Потім файл STL завантажується в програму різання для принтера, наприклад, Cura або Slic3r.

Програма-слайсер дозволяє задавати фізичні властивості моделі, такі як щільність заповнення або використання опорних конструкцій, оскільки на деякому шарі може виявитися, що окремі елементи "зависають" в повітрі, так як вони повинні кріпитися до верхніх шарів. Після такої пропередньої підготовки програма перетворює 3D-модель в G-код. Він містить інструкції для екструдера, за якими той повинен надавати форму кожному шару моделі. Код завантажується в принтер, пристрій запускається, і починається друк [44].

Сам процес друку – це ряд повторюваних циклів, пов'язаних зі створенням тривимірних моделей, нанесенням на робочий стіл (елеватор) принтера шару витратних матеріалів, переміщенням робочого столу вниз на рівень готового шару і видаленням з поверхні столу відходів. Цикли безперервно слідуєть один за іншим: на перший шар матеріалу наноситься наступний, елеватор знову опускається і так до тих пір, поки на робочому столі не виявиться готовий виріб. Так шар за шаром відтворюється повний набір перетинів моделі повторюючи форму необхідного виробу [43].

Взагалі технології 3D-друкування та сканування тісно пов'язані між собою, оскільки чудово співпрацюють та доповнюють одна одну. 3D-принтер виступає відмінним помічником при 3D-скануванні кісток і навіть окремих органів з найвищим рівнем деталізації. Надалі, отримані 3D-моделі і створені прототипи можуть бути використані в якості навчальних матеріалів або при створенні повноцінних біологічних протезів [33].

Використання 3D-принтерів в медицині дозволяє врятувати людські життя. Такі принтери можуть відтворити точну копію людського скелета для

відпрацювання прийомів, які гарантують проведення успішної операції. Все частіше 3D-принтери використовують в протезуванні та стоматології, так як тривимірний друк дозволяє отримати протези і коронки значно швидше класичною технологією виробництва [45].

Медичні тривимірні моделі можуть бути виготовлені з різних матеріалів, включаючи живі органічні клітини. Вибір того чи іншого матеріалу для медичного прототипування залежить від цілей і завдань, що стоять перед медиками, і проблем, пов'язаних зі здоров'ям пацієнта.

3D-принтер – це сучасний пристрій, що використовує цифрову 3D-модель для пошарового створення фізичного об'єкта. Друк може здійснюватися різними способами і з використанням різних матеріалів, але в основі будь-якого з них лежить принцип пошарового створення (вирощування) твердого об'єкта [46]. Для створення 3D-моделі можна використати нідерландський 3D-принтер LeapFrog Creatr HS, характеристики якого наведено в таблиці 2.4 [47, 48].

Leapfrog Creatr HS являє собою потужний і надійний 3D принтер, що забезпечує високу точність і швидкість 3D-друку. Він є один з найбільш швидких і точних приладів в своєму класі. Принтер має точністю друку в 50 мікрон, що дозволяє виконувати більш детальний друк. Його точність не схильна до зміни при перепадах температур і не погіршується з часом.

Корпус виконаний з високоякісного алюмінію, а два екструдера дозволяють друкувати відразу двома кольорами або видами пластику одночасно. Принтер може використовувати пластик різних типів для друку, що надає користувачам свободу дій. Також варто відзначити велику область побудови, яка дозволить створювати великі вироби, або проводити дрібносерійну 3D-друк [47, 49].

Принтер оснащений 4-дюймовим повно кольоровим ЖК-дисплеєм і всього однією ручкою для управління. Для роботи з ним не потрібні SD-карти, принтер працює від USB-накопичувача, або безпосередньо через комп'ютер (рис. 2.10).

Таблиця 2.4 - Технічні характеристики Leapfrog Creatr HS

| Параметр                             | Значення                                               |
|--------------------------------------|--------------------------------------------------------|
| Розмір друку                         | 270*280*180 мм                                         |
| Об'єм друку                          | max 14,8 л                                             |
| Сумісність з операційною системою    | Windows, Mac OS, Linux                                 |
| Програма керування                   | Repetier Host, Simplify 3D                             |
| Мова програми                        | Російська, англійська                                  |
| Розміри принтера                     | 60*50*50 см                                            |
| Вага принтера                        | 30 кг                                                  |
| Формат файлів друку                  | STL .GCODE .OBJ                                        |
| Робоча напруга                       | 220 V                                                  |
| Потужність                           | До 400 Вт                                              |
| Матеріал для друку                   | ABS; PLA; PVA; Laybrick; Nylon; PVA, HIPS              |
| Діаметр матеріалу                    | 1,75 мм                                                |
| Точність друку (товщина шару)        | 0,05-0,35 мм (50-350 мікрон)                           |
| Точність позиціонування              | Вісь Z: 0.005mm, вісь XY: 0.0169mm                     |
| Кількість екструдерів                | 2 шт                                                   |
| Діаметр сопел                        | 0,35 мм (0,5 мм опціонально)                           |
| Тип друкованого столу                | Наявний підігрів поверхні                              |
| Температура екструдера               | MAX 260°C                                              |
| Температура екструдера               | MAX 120°C                                              |
| Оптимальна робоча температура        | > 25 °C                                                |
| Рекомендована температура екструдера | 190-210 °C для PLA / 220-240°C для ABS                 |
| Швидкість руху друкованої голівки    | 30-300 мм / с при друку, 30-400 мм / с при переміщенні |



Рисунок 2.10 –Елементи керування принтеру

## 2.4 Висновки до розділу 2

У розділі досліджено особливості технології прототипування, що активно використовується для підвищення ефективності процесу розробки біомедичних виробів. Розглянуто методи та засоби віртуального та фізичного прототипування. Показано актуальність використання технологій 3D-сканування та 3D-друкування, що дозволяють створити цифровий двійник проєктованого виробу та виготовити його натурний макет. Окрім, того в розділі проаналізовано середовища автоматизованого проєктування, а також наявне обладнання, яке може використовуватись для прототипування кіберпротезу.

## **3 РОЗРОБКА ПРОТОТИПУ КІБЕРПРОТЕЗУ ВЕРХНЬОЇ КІНЦІВКИ ЛЮДИНИ**

### **3.1 Метод комплексного віртуального та фізичного прототипування**

Стадії та етап методу комплексного прототипування наведено на рисунку 3.1.

Можна виділити три основні стадії: розробка фізичної (механічної) частини прототипу, розробка електронної частини та розробка програмного забезпечення.

Етапами розробки механічної частини прототипу кіберпротезу є:

- оцифрування за допомогою сканера (наприклад, Gotcha) здорової кінцівки пацієнта для отримання загального виду 3D-моделі руки та її розмірів;
- робота в CAD-середовищі над створенням окремих механічних деталей кібер-протезу та загальної складальної одиниці з урахуванням кінематичних властивостей майбутнього виробу;
- друк окремих механічних деталей за допомогою 3D-принтеру (наприклад, LeapFrog Creatr HS) та складання фізичного прототипу.

Етапами розробки електронної частини прототипу кіберпротезу є:

- аналіз та вибір електронних компонентів;
- з'єднання електронних компонентів та встановлення в створену фізичну частину руки.

Розробка програмного забезпечення для функціонування прототипу кіберпротезу може розпочинатися вже після вибору електронних компонентів, оскільки передбачає використання програмних бібліотек, що пов'язані з конкретними компонентами. Вибір електронних компонентів важливий також при створення механічної частини кіберпротезу, оскільки обумовлює розміри посадкових місць.



Рисунок 3.1 – Метод комплексного віртуального та фізичного прототипування

### 3.2 Розробка механічної частини прототипу кіберпротезу

Створення та дослідження віртуального прототипу проєктованого виробу дозволяє: виявити помилки проєкту на ранніх етапах створення проєкту та, відповідно, зменшити час розробки та знизити витрати на доробку; дослідити більше альтернативних варіантів та шляхів розвитку проєкту; виконати дослідження прототипу, які інколи є неможливими в реальних умовах на натурному макеті; не проводити фізичні експерименти, що потребують значного часу та матеріальних витрат.

Для створення та дослідження віртуальних прототипів конструкцій роботизованих протезів використовуються сучасні системи автоматизованого проєктування (AutoCAD, PTC Creo, КОМПАС та ін.).

Згідно з [50] PTC Creo є одним з восьми найкращих середовищ автоматизованого проєктування в світі. Система має потужний функціонал та надає інтелектуальне середовище проєктування. Саме тому, ця CAD-система була обрана для розробки параметричної моделі кіберпротезу руки. У якості середовища проєктування використовувалася версія PTC Creo Parametric Educational Edition 4.0, завдяки зручному та простому способу отримання академічної ліцензії. Основні стадії та етапи інформаційної технології створення механічної частини кіберпротезу представлено на рисунку 3.2.

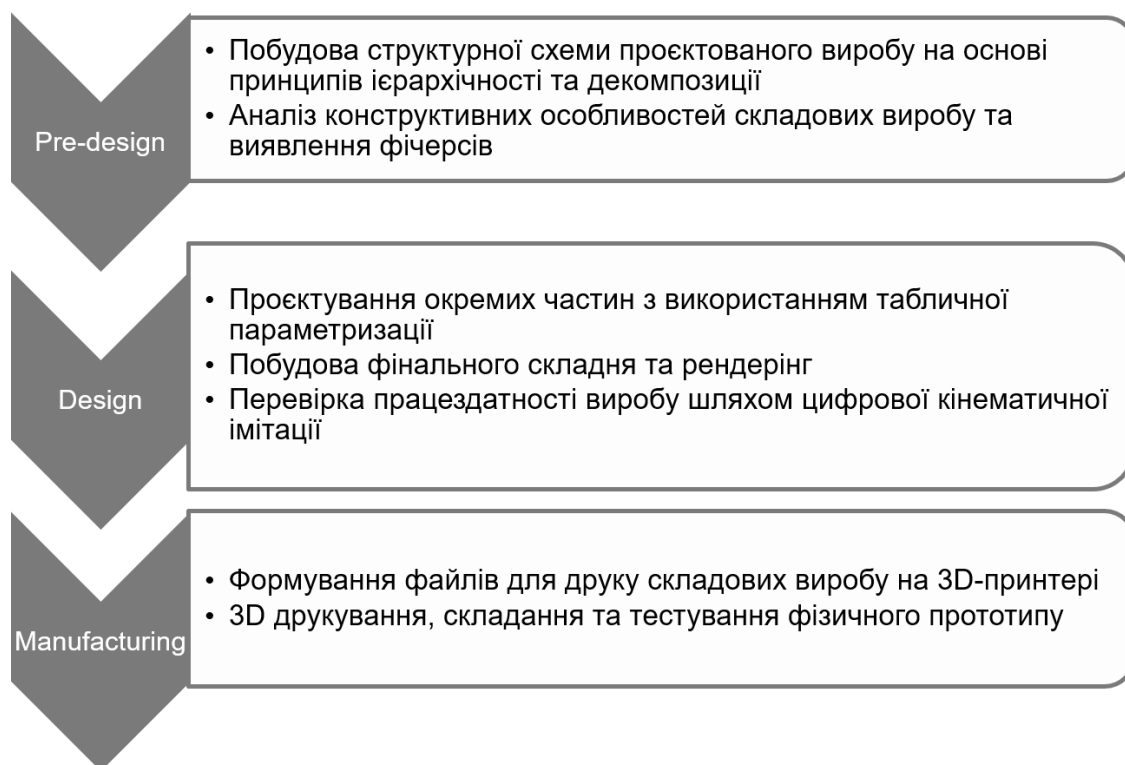


Рисунок 3.2 – Основні етапи інформаційної технології створення прототипу механічної частини кіберпротезу

### 3.2.1 Розробка віртуального прототипу конструктиву кіберпротезу

Структурна схема проєктованого кіберпротезу представлена на рисунку 3.3. Вона містить окремі деталі (пальці, долоня, основа великого пальця) та складальну одиницю долоня+основа.

Для побудови параметричної моделі, параметри кожного пальця можна представити у вигляді таблиці 3.1.

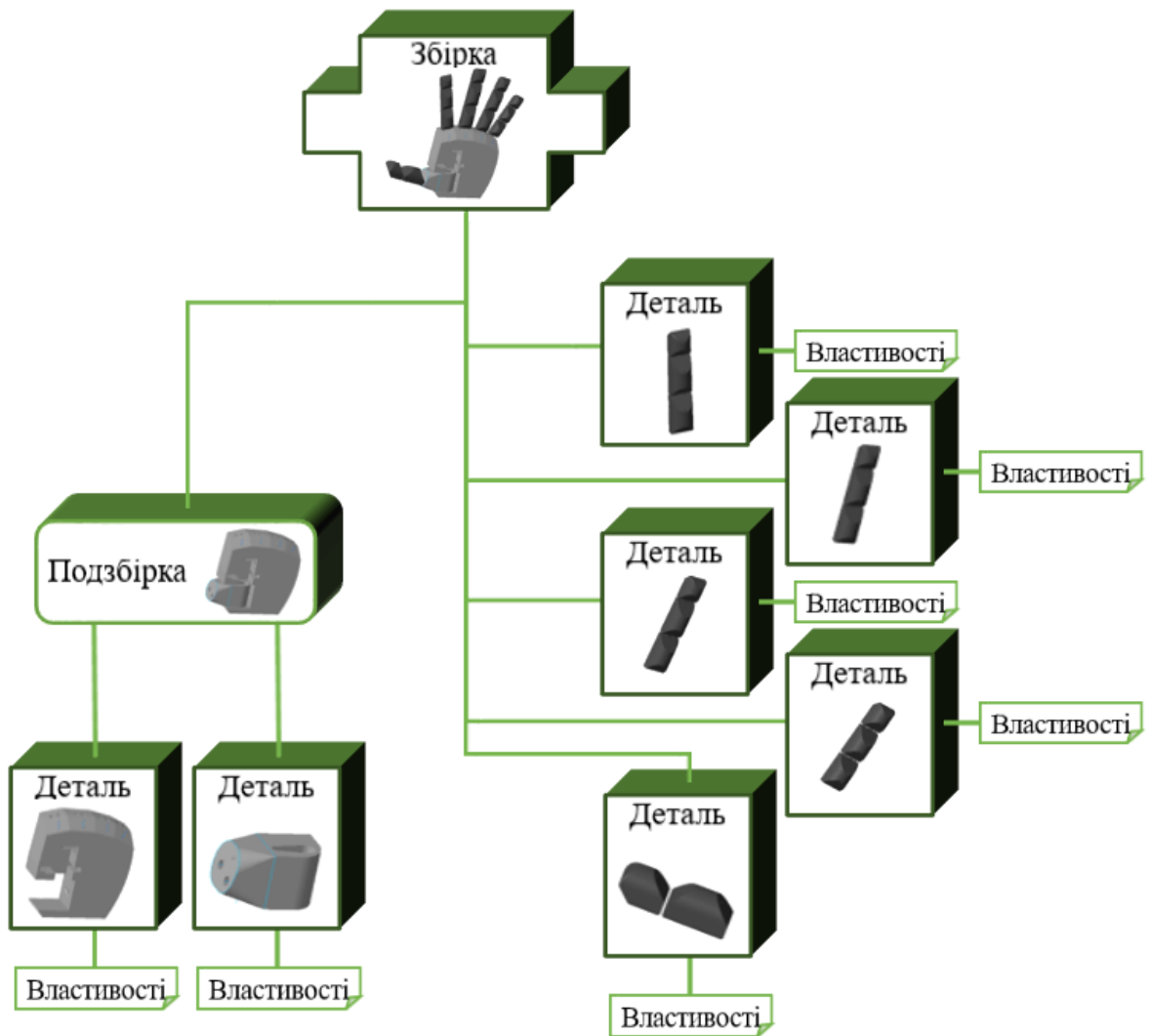


Рисунок 3.3 – Структурна схема проєктованого виробу

Таблиця 3.1 – Параметри пальців прототипу кіберфізичного протезу

| Назва параметру<br>пальця | вказівний | середній | безіменний | мізинець | великий |
|---------------------------|-----------|----------|------------|----------|---------|
| F_HEIGHT                  | 13,2      | 14,1     | 13,2       | 12       | 16      |
| F_WIDTH                   | 14        | 15       | 14         | 13       | 18      |
| F_LENGTH                  | 71,6      | 77,8     | 71,6       | 56,7     | 80      |
| F_F_HEIGHT                | 4,2       | 5,1      | 4,2        | 3        | 7       |

|             |      |      |      |       |        |
|-------------|------|------|------|-------|--------|
| F_WIDTH_END | 13   | 14   | 13   | 12    | 17     |
| SMALL_F     | 17,6 | 18,8 | 17,6 | 16    | 21,333 |
| F           | 26   | 28,5 | 26   | 19,35 | 28,333 |

В таблиці наведено наступні параметри:

- F\_HEIGHT - висота пальця;
- F\_WIDTH - ширина пальця;
- F\_LENGTH - довжина пальця;
- F\_F\_HEIGHT - висота основної частини пальця;
- F\_WIDTH\_END - ширина пальця у кінця;
- SMALL\_F - довжина крайньої (дрібної) фаланги пальця;
- F - довжина основної фаланги.

Параметри F\_HEIGHT, F\_WIDTH, F\_LENGTH вносяться проєктувальником самостійно, виходячи з габаритів необхідного пальця. У свою чергу, параметри F\_F\_HEIGHT, F\_WIDTH\_END, SMALL\_F, F обчислюються за рівняннями, які були виведені, виходячи з анатомії кінцівки середньостатистичної людини:

$$F\_F\_HEIGHT = F\_HEIGHT - 9;$$

$$F\_WIDTH\_END = F\_WIDTH - 1 \text{ мм};$$

$$SMALL\_F = (F\_HEIGHT / 3) * 4;$$

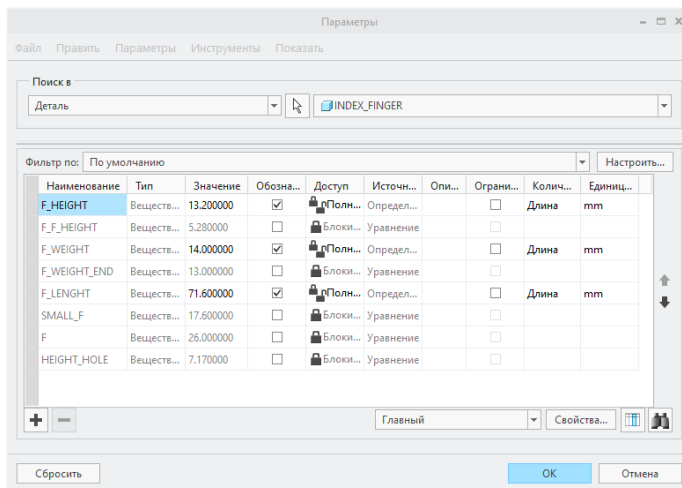
$$F = (F\_LENGTH - (SMALL\_F + 2 \text{ мм})) / 2,$$

де 2 мм – це сума відстаней між трьома фалангами, а 2 – це кількість фаланг.

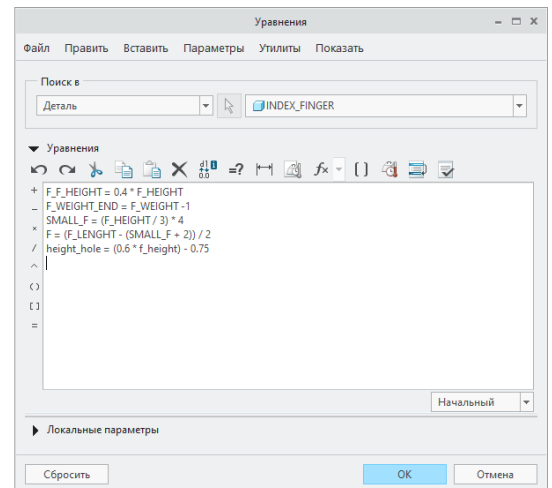
Параметри моделі пальця (рис. 3.4, а) і рівняння (рис. 3.4, б) можна задати у вкладці «Інструменти» для того, щоб полегшити введення потрібних значень при створенні моделей інших пальців.

Для побудови 3D-моделей використовувалися такі дії форм і проєктування як:

- «сполучити» - створює тривимірну геометрію за допомогою сполучення мінімум двох двовимірних перетинів, які паралельні один одному або проєціюються на паралельні поверхні;
- «витягнути» - створює тривимірну геометрію, проєктуючи двовимірний перетин на зазначеній по нормалі до площини ескізу відстані;



а



б

а – параметри; б – рівняння

Рисунок 3.4 – Вікна заданих параметрів і рівнянь вказівного пальця

- «фаска» - створює конструкторський елемент фаски кромки, використовуючи кромки, ланцюжки і поверхні в якості прив'язок;
- «заокруглення» - додає заокруглення в кромки або простір між поверхнями, використовуючи радіус або хорду.

Одержаний результат проєктування пальця можна побачити на рисунку 3.5, долоні – на рисунку 3.6, п'ясть великого пальця – на рисунку 3.7.

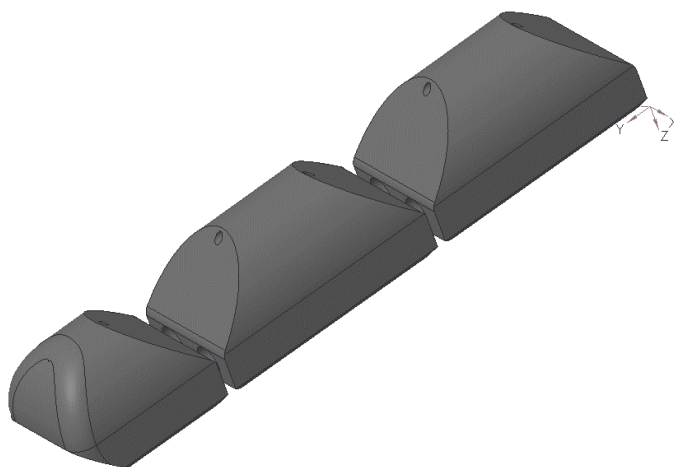


Рисунок 3.5 – Кінцевий результат проєктування 3D-моделі пальця

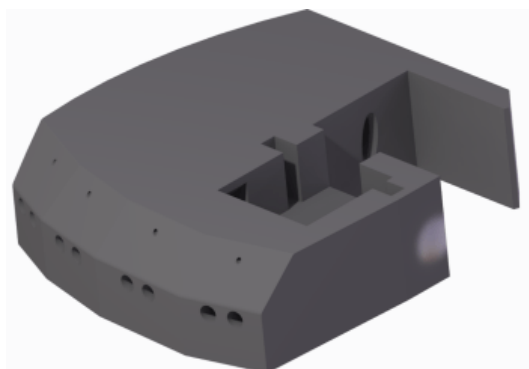


Рисунок 3.6 – Фінальна модель долоні

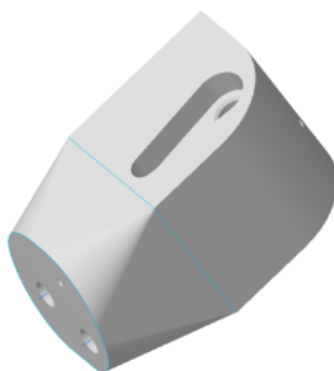


Рисунок 3.7 – Фінальна модель п'ястя великого пальця

Результати проєктування деталей в середовищі РТС Сгео наведено на рис. 3.8.

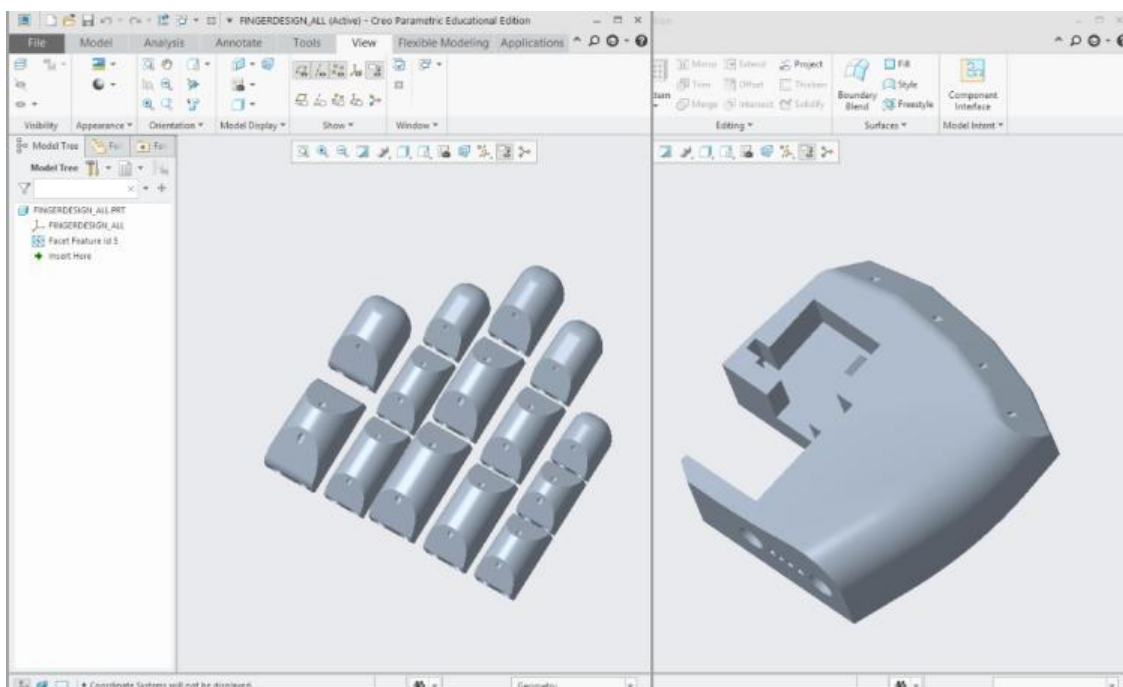


Рисунок 3.8 – Деталі кіберпротезу в середовищі PTC Creo Parametric

Також, після створення файлів усіх п'яти пальців та долоні, можна створити складень. Насамперед, це зручно для того, щоб подати на 3D-друк відразу всі пальці, а не кожен палець окремо, чи побачити як буде виглядати кисть руки у фіналі (рис. 3.9).



Рисунок 3.9 – Фінальний рендерінг моделі усієї кисті протезу руки

### 3.2.2 Виготовлення фізичного прототипу конструктиву кіберпротезу

На сьогоднішній день існує багато різних методів та засобів 3D-принтіngu [42, 43]. Питання вибору відповідного апаратного та програмного забезпечення для реалізації цієї адитивної технології за умови урахування часових та матеріальних обмежень, є надзвичайно важливим.

3D-принтер від компанії Leapfrog (рис. 3.10) забезпечує високу швидкість друку (до 300 мм.сек); оснащений двома екструдерами, платформою, що підігрівається, LCD-дисплеєм і використовує для друку різні види пластиків, такі як ABS / PLA / Нейлон / PVA / Laybrick та ін. Область побудови складає 270x260x180 мм [47-49], що відповідає розмірам проєктованого виробу.

Після розробки 3D-моделей, в середовищі PTC Creo було згенеровано файли в форматі STL, які є вхідними для програми Simplify3D [51]. В цій програмі було проведено перетворення даних для процесу 3D-друкування (рис. 3.11), а також виконано налаштування параметрів друку (рис. 3.12).

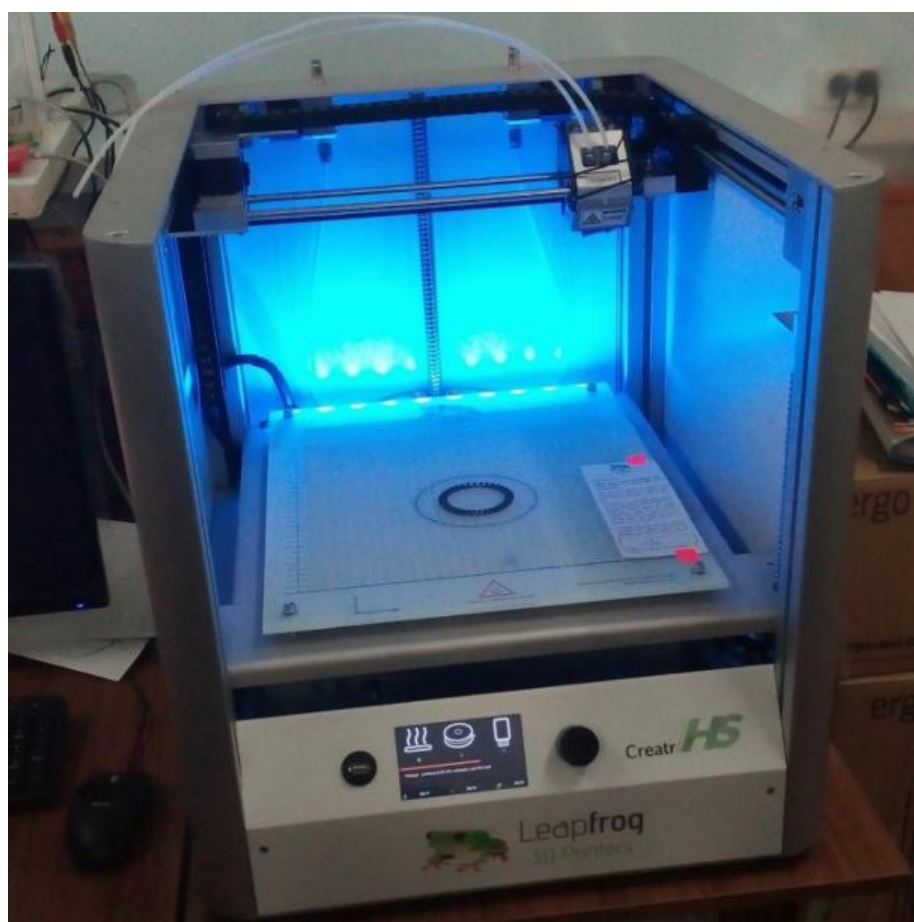


Рисунок 3.10 – Leapfrog Creatr HS

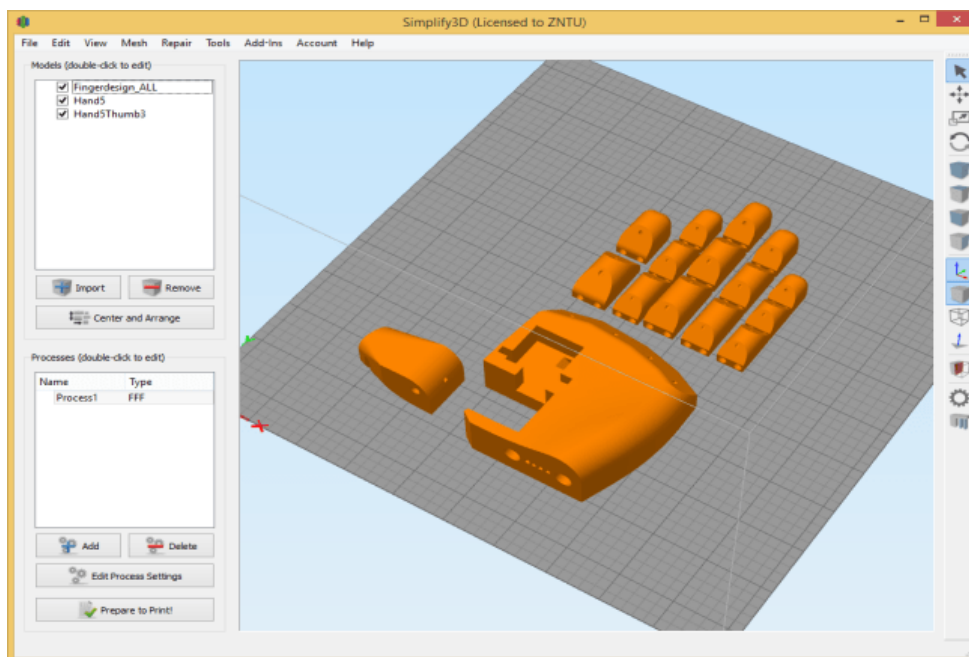


Рисунок 3.11 - 3D-модель кисті кіберпротезу в середовищі Simplify3D

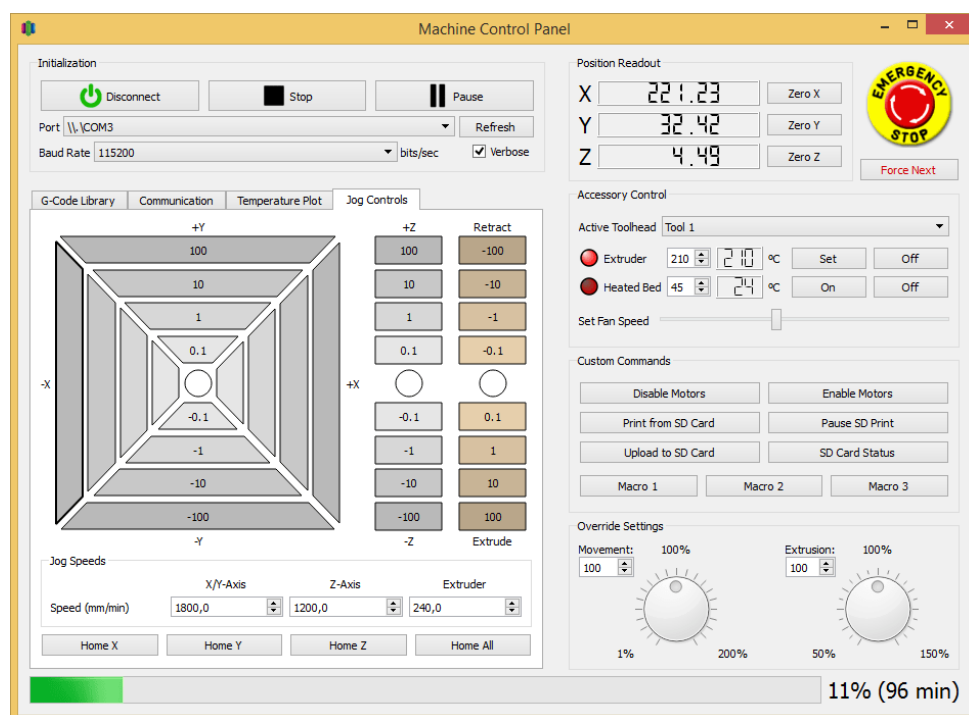


Рисунок 3.12 – Налаштування параметрів друкування в Simplify3D

В якості матеріалу використовувався ABS пластик, що стійкий до багатьох несприятливих умов експлуатації як волого-, масло- та кислотостійкість. Також він не є токсичним і має низьку вартість.

В результаті 3D-друкування було виготовлено фізичні прототипи деталей механічної частини проєктованого кіберпротезу, а саме – пальці, долоня та основа великого пальця.

### 3.3 Розробка електронної частини прототипу кіберпротезу

#### 3.3.1 Аналіз та вибір плати Arduino

Arduino – це комбінація апаратної та програмної частини для простої розробки електроніки. Апаратна частина включає в себе велику кількість видів плат Arduino з вбудованими мікроконтролерами, що програмуються, а також додаткові модулі. Програмна частина складається з середовища розробки прошивок Arduino IDE (програми для написання скетчів і завантаження їх на мікроконтролер Arduino), спрощеної мови програмування та великої кількості готових функцій та бібліотек (рис. 3.13) [52].

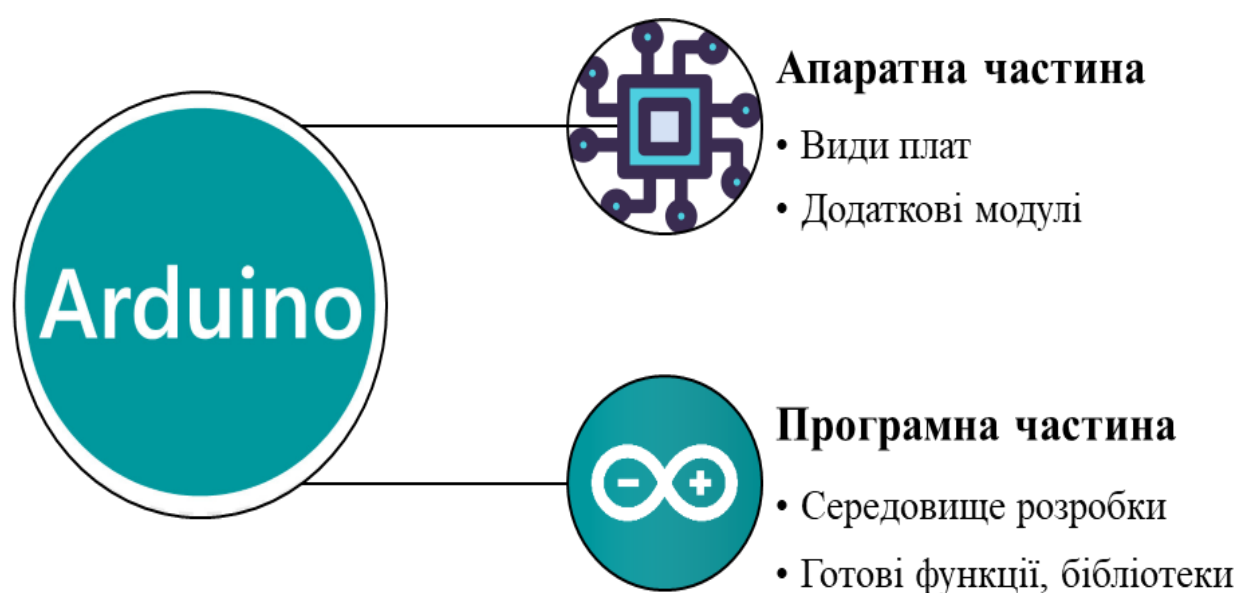


Рисунок 3.13 – Компоненти Arduino

Плати Arduino є універсальними контролерами, які можна використовувати для виконання будь-яких задач та створення готових електронних пристроїв різного призначення, від годинника з будильником до робота. До самої плати можна підключати різну периферію – кнопки, деякі види датчиків (температури, тиску, освітленості, прискорення і т.п.), світлодіоди, рідкокристалічні індикатори (цифро-буквені) написати програму і змусити взаємодіяти все це як завгодно.

Для Arduino є величезна кількість інструкцій і готових скетчів, що дозволяє з легкістю створювати, повторювати будь-який проєкт, доопрацювати або адаптувати його. Все необхідне є у відкритому доступі. За допомогою Arduino можна створити власні аналоги пристроїв і заощадити на цьому, або розробляти прототипи майбутніх пристроїв.

На сьогоднішній день існує велика кількість видів плат Arduino, починаючи з класичних Arduino UNO, Leonardo, Mini, Micro і Nano і закінчуючи специфічними Industrial 101, Tian і MKR1000. І всі вони мають характерні відмінності і різні сфери застосування [53].

В сімействі Arduino плата Arduino Uno є стандартною і можливо найбільш поширеною. Вона заснована на чіпі ATmega328, що має 32 КБ флеш-пам'яті, 2 КБ SRAM і 1 Кбайт EEPROM пам'яті. На периферії має 14 дискретних (цифрових) каналів введення / виводу і 6 аналогових каналів введення / виводу, це дуже різнобічно-корисні девайси, що дозволяють перекривати більшість аматорських завдань в області мікроконтролерної техніки. Arduino Uno є однією з найдешевших і найбільш часто використовуваних плат [54]. Більшість плат сімейства мають схожу з нею комплектацію та способи взаємодії.

Схожою на Arduino Uno є плата Arduino Leonardo, але вона з іншим мікро контролером, який знаходиться в тому ж класі, але має деякі відмінності позитивного характеру. Більша кількість аналогових входів (12 проти 6) для сенсорів, більше каналів ШІМ (7 проти 6), більше пінів з апаратним

перериванням (5 проти 2), роздільні незалежні Serial-інтерфейси для USB і UART. Arduino Leonardo має трохи інше розташування пінів ніж Arduino Uno, через що можлива несумісність з деякими платами розширення [54].

Плати відрізняються один від одного своїми формами, характеристиками і можливостями. В таблиці 3.2 наведено порівняльну характеристику плат сімейства Arduino [53, 54].

Оскільки для електронної частини розроблювального прототипу важливе використання маленької, але функціональної плати, було проаналізовано більш детально плати Arduino Nano, Arduino Mini та Arduino Micro.

Arduino Nano – це функціональний аналог Arduino Uno, але розміщений на мініатюрній платі. Відмінність полягає у відсутності власного гнізда для зовнішнього живлення і застосуванням mini-USB кабелю для взаємодії замість стандартного (рис. 3.14).

Arduino Mini є компактним ( $30 \times 18$  мм) аналогом Arduino Uno, але в іншому форм-факторі, через який виникають проблеми з підключенням до плати розширення Arduino. Передбачається з'єднання з додатковими модулями дротами і / або через макетну плату. На платі немає USB-порту, тому прошивати потрібно через окремий USB-Serial адаптер (рис. 3.15).

Таблиця 3.2 – Порівняння плат Arduino

| Параметри                      | Плата Arduino |             |            |             |                                 |            |             |              |
|--------------------------------|---------------|-------------|------------|-------------|---------------------------------|------------|-------------|--------------|
|                                | Uno           | Leonardo    | Nano       | Mega        | Due                             | Mini       | Micro       | MO           |
| Мікроконтроллер                | Atmega 328    | ATmega 32u4 | ATmega 328 | ATmega 2560 | Atmel SAM3XBE ARM Cortex-M3 CPU | ATmega 328 | ATmega 32u4 | Atmel SAMD21 |
| Робоча напруга (В)             | 5             | 5           | 5          | 5           | 3.3                             | 5          | 5           | 3.3          |
| Цифрові входи/виходи           | 14            | 20          | 14         | 54          | 54                              | 14         | 20          | 20           |
| Виходи в ШІМ                   | 6             | 7           | 6          | 14          | 12                              | н6         | 7           | 12           |
| Аналогові входи/виходи         | 6             | 12          | 8          | 16          | 12                              | 6          | 12          | 6            |
| Максимальний струм з піна (мА) | 40            | 40          | 40         | 40          | 800                             | 40         | 40          | 7            |
| Flash - пам'ять (КБ)           | 32            | 32          | 32         | 256         | 512                             | 32         | 32          | 256          |
| ОЗУ (КБ)                       | 2             | 2.5         | 2          | 8           | 96                              | 2          | 2.5         | 32           |
| EEPROM (КБ)                    | 1             | 1           | 1          | 4           | X                               | 1          | 1           | X            |
| Тактова частота (МГц)          | 16            | 16          | 16         | 16          | 84                              | 16         | 16          | 48           |
| Розміри (мм)                   | 69*53         | 75*53       | 18*45      | 102*53      | 102*53                          | 30*18      | 48*18       | 69*53        |
| USB                            | USB A-B       | micro-USB   | mini-USB   | USB A-B     | micro-USB                       | USB-Serial | micro-USB   | micro-USB    |



Рисунок 3.14 – Зовнішній вигляд Arduino Nano



Рисунок 3.15 – Зовнішній вигляд Arduino Mini

Arduino Micro – це Arduino Leonardo, виконана на компактній платі. Відмінність полягає у відсутності власного гнізда для зовнішнього живлення, але воно може бути підведено безпосередньо до контакту V<sub>i</sub>. В іншому, комплектація і способи взаємодії збігаються з Arduino Leonardo. Він також має один мікроконтролер ATmega32u4 і для прошивки через USB, і для виконання програм; також може виступати в ролі клавіатури або миші; надає таку саму кількість пам'яті, цифрових, аналогових і ШІМ-портів (рис. 3.16).

Arduino Nano, Mini, Micro є компактними та функціональними платами, що мають певні відмінності зі своїми базовими моделями, однак в основному їх комплектація та способи взаємодії збігаються з нею. Кожна з трьох має штиркові контакти, що дозволяють легко встановлювати її на макетну плату.

В таблиці 3.2 представлено порівняння плат Arduino Nano, Arduino Mini та Arduino Micro [55].



Рисунок 3.16 – Зовнішній вигляд Arduino Micro

Таблиця 3.3 – Порівняння мініатюрних плат Arduino

| Характеристики плати           | Arduino Micro | Arduino Nano | Arduino Mini |
|--------------------------------|---------------|--------------|--------------|
| Мікроконтролер                 | ATmega32u4    | ATmega32u    | ATmega32u    |
| Робоча напруга                 | 5 V           | 5 V          | 5 V          |
| Вхідна напруга (рекомендована) | 7-12 V        | 7-12 V       | 7-9 V        |
| Вхідна напруга (обмеження)     | 6-20 V        | 7-12 V       | 7-9 V        |
| Цифрові (I/O) піни             | 20            | 22           | 14           |
| Аналоговий вхід / вихід        | 12            | 8            | 6            |
| Постійний струм на I/O пін     | 20 mA         | 40 mA        | 40 mA        |
| Флеш-пам'ять                   | 32 KB         | 32 KB        | 32 KB        |
| SRAM                           | 2.5 KB        | 2 KB         | 2 KB         |
| EEPROM                         | 1 KB          | 1 KB         | 1 KB         |
| Тактова частота                | 16 MHz        | 16 MHz       | 16 MHz       |
| LED вбудовані                  | 13            | -            | -            |
| Довжина                        | 48 мм         | 18 мм        | 30 мм        |
| Ширина                         | 18 мм         | 45 мм        | 18 мм        |
| Вага                           | 13 г.         | 7 г.         | 2 г.         |

В результаті проведеного аналізу було визначено, що плата Arduino Mini, навіть незважаючи на свою компактність для виконання завдання не підходить, оскільки її можливостей не вистачає та на ній немає USB-порту, тому прошивати потрібно через окремий USB-Serial адаптер, що викликає додаткові незручності. Не найкращим варіантом була і Arduino Micro в

порівнянні з Arduino Nano вона є більшою та важчою. Тому було вирішено зупинитись на Arduino Nano та працювати безпосередньо з нею, більш детальний її опис наведено нижче.

Arduino Nano – це одна з найпоширеніших та найбільш мініатюрних плат сімейства Arduino (рис. 3.17). Існує кілька версій цих плат, що відрізняються між собою мікро-контролерами. У молодшій версії 2.X цієї плати використовується чіп ATmega168, а у версії 3.0 – ATmega328P. На відміну від свого молодшого побратима, Arduino Nano v 3.0 має вдвічі більші обсяги незалежної і flash пам'яті. І може похвалитися тактовою частотою в 16 МГц. Вона є повним аналогом Arduino Uno, тільки на ній відсутнє винесене гніздо зовнішнього живлення, в усьому іншому параметри збігаються. Живлення Arduino Nano може здійснюватися двома способами:

- через mini-USB або microUSB при підключенні до комп'ютера;
- через зовнішнє джерело живлення (павербанка або від адаптера, підключеного в розетку), що має напругу 6-20 В з низьким рівнем пульсацій [56, 57].

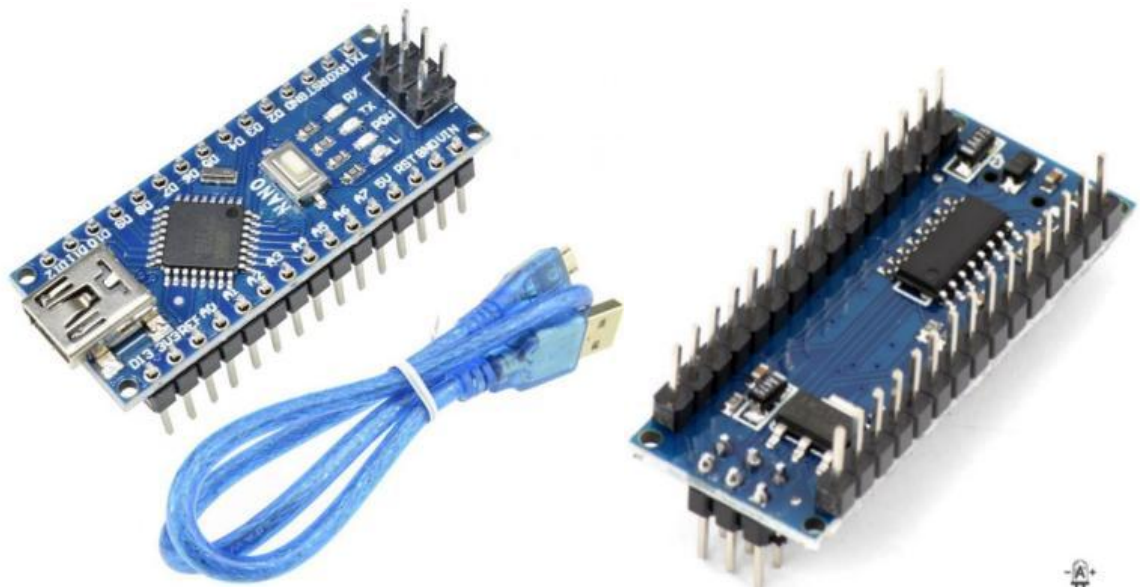


Рисунок 3.17 – Зовнішній вигляд Arduino Nano

На рисунку 3.18 наведено опис основних компонентів плати Arduino Nano [58].

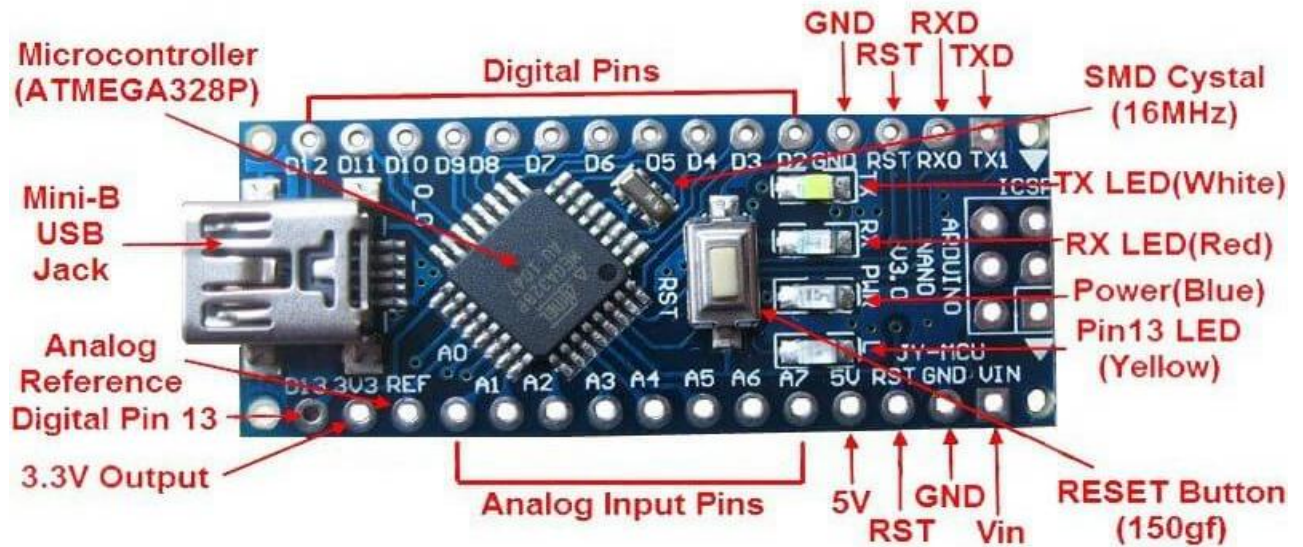


Рисунок 3.18 – Основні компоненти плати [58]

Основні технічні характеристики Arduino Nano наведено в таблиці 3.4 [59].

Таблиця 3.4 – Технічні характеристики Arduino Nano

| Характеристика                    | Значення                                               |
|-----------------------------------|--------------------------------------------------------|
| Мікроконтролер                    | ATmega328                                              |
| Гранична напруга живлення         | 5-20 В                                                 |
| Рекомендована напруга живлення    | 7-12 В                                                 |
| Цифрових вводів / виводів         | 14                                                     |
| Широтно-імпульсна модуляція (ШИМ) | 6 цифрових пінів можуть бути використані як виводи ШИМ |
| Аналогові виходи                  | 8                                                      |
| Максимальна сила струму           | 40 mAh з одного виводу і 500 mAh з усіх виводів        |
| Flash пам'ять                     | 16 Кб или 32 Кб, в залежності від чіпу                 |
| SRAM                              | 2 Кб                                                   |
| EEPROM                            | 1 Кб                                                   |
| Тактова частота                   | 16 МГц                                                 |

Плата має 14 цифрових пінів, що позначені з провідною буквою «D» (digital або цифровий). Вони можуть бути як входом так і виходом. Робоча

напруга цих пінів становить 5 В. Кожен з них має підтягуючий резистор, а подана на один з цих пінів напруга нижче 5 вольт все одно буде вважатися як 5 вольт (логічна одиниця). Аналогові піни на платі позначені провідною буквою «А». Ці піни є входами і не мають підтягуючих резисторів. Вони вимірюють напругу, що на них надходить і повертають значення від 0 до 1024 при використанні функції `analogRead()`. Ці піни вимірюють напругу з точністю до 0,005 В.

На платі поряд з деякими цифровими пінами іноді зображено значок тильди (~), що означає, що даний пін може бути використаний як вихід широтно-імпульсної модуляції (ШИМ). У Arduino Nano є 6 виходів ШІМ, це піни D3, D5, D6, D9, D10 і D11, для використання яких є спеціальна функція `analogWrite()` [57]. На рисунках 3.19-3.20 зображено розміщення пінів на платі.

Через свої габаритні розміри плата часто використовується в проєктах, в яких важлива компактність, однак через відсутність портів для роботи з нею необхідно використовувати плату розширення.

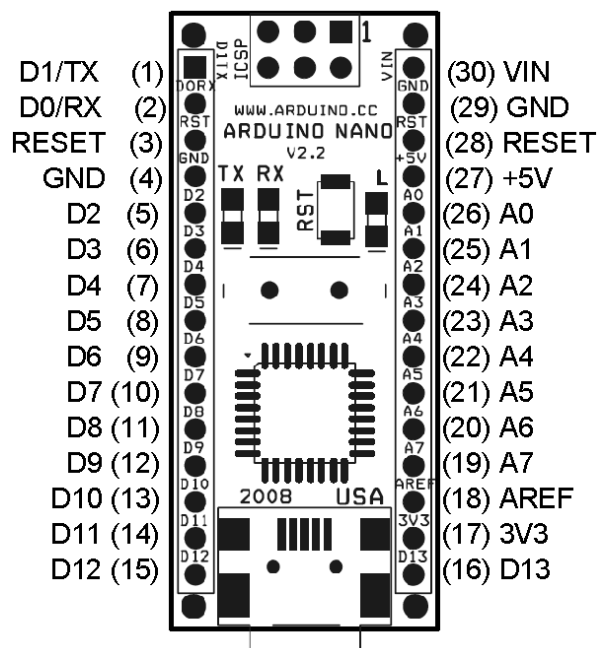


Рисунок 3.19 – Розташування пінів на платі



екранами (LCD -Shield), SD-картами (Data Logger Shield), мережеві (Ethernet Shield) і GPS-Shield, Shield з вбудованими реле для підключення до навантаження (рис. 3.21). В таблиці 3.5 наведено опис деяких видів Shield [60].

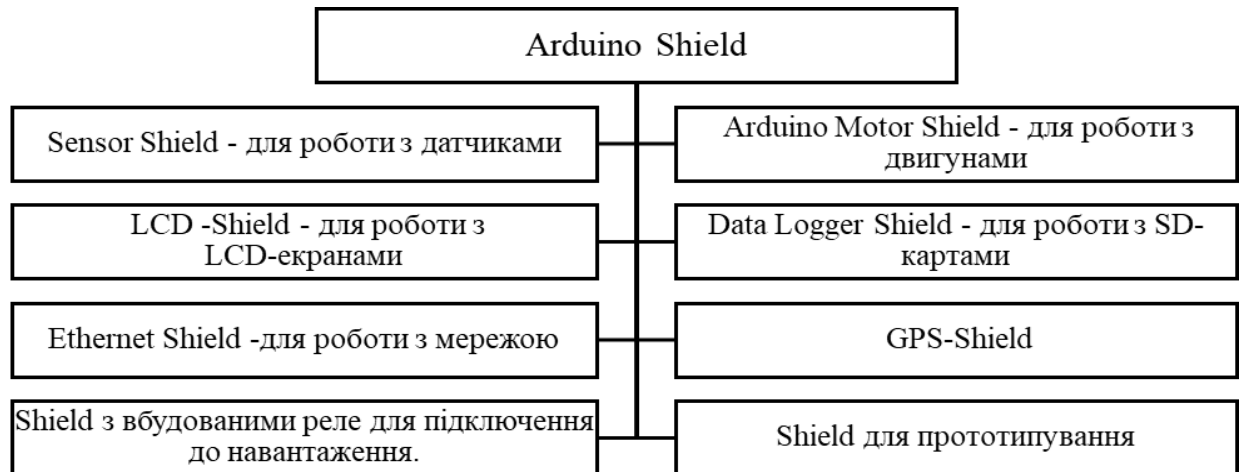


Рисунок 3.21 – Види Arduino Shield

В результаті аналізу Arduino Shield, виявлено, що для виконання поставленого завдання найкраще підходять плати розширення груп Arduino Sensor Shield та Arduino Motor Shield. Зазвичай Shield призначений для цілком конкретної версії контролера, тому для виконання поставленої задачі та роботи з платою Arduino Nano обрано плату розширення з першої групи, а саме Arduino Nano Shield V3.0.

Arduino Nano є дуже маленькою, що є безперечно її основною перевагою та дозволяє створювати на її основі компактні прототипи. Але для налагодження вона є не зручною та може знадобиться велика кількість входів і виходів для підключення периферійних пристроїв, тому для виправлення цього недоліку та полегшення підключення використовують плату розширення Arduino Nano Shield V3.0 (рис. 3.22), яка є платою введення / виведення, що полегшує підключення різних датчиків і виконавчих пристроїв. У центрі плати змонтовані роз'єми для установки Arduino Nano, а сама плата має розміри, ідентичні Arduino UNO.

Таблиця 3.5 – Опис видів Arduino Shield

| Назва плати розширення              | Основне завдання                                                                                                                                         | Додаткові можливості та версії                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Arduino Sensor Shield               | Надає зручні варіанти підключення до плати за рахунок додаткових роз'ємів живлення і землі, виведених на плату до кожного з аналогових і цифрових пінів. | Наявні роз'єми для підключення зовнішнього джерела живлення, світлодіод і кнопка перезапуску. Вони відрізняються кількістю і видом роз'ємів. Найбільш популярними сьогодні є версії Sensor Shield v4 і v5.                                                                                                                                                                                                |
| Arduino Motor Shield                | Забезпечити управління пристроями, що споживають досить високий для звичайної плати Arduino струм.                                                       | Дозволяє підключати до плати звичайний і серво двигуни є функція управління потужністю двигуна (з допомогою ШІМ) і зміни напрямку обертання. Загальним для всіх версій є наявність в схемі потужного транзистора, через який підключається зовнішнє навантаження, теплопровідних елементів, схеми для підключення зовнішнього живлення, роз'єми для підключення двигунів і піни для підключення до плати. |
| Плати розширення для прототипування | Підвищити компактність пристрою, коли всі необхідні компоненти розташовуються відразу над основною платою.                                               | На них розташовані контактні площадки для монтажу елементів, виведена кнопка скидання і є можливість підключення зовнішнього живлення.                                                                                                                                                                                                                                                                    |
| Arduino Data Logger Shield          | Збереження даних, отриманих з датчиків, з прив'язкою за часом.                                                                                           | Дозволяє не тільки зберегти дані і отримувати час зі вбудованих годин, але і підключити датчики в зручному вигляді шляхом пайки або на монтажній платі.                                                                                                                                                                                                                                                   |

Всі роз'єми для підключення аналогових і цифрових датчиків, порти UART, I2C зручно розведені на платі. По периметру також встановлені роз'єми для підключення шилтів. Крім цього на платі є додаткові роз'єми, що дозволяють підключити до будь-якого виходу мікроконтролера пристрій на гнучкому шлейфі з живленням і землею [61, 62].

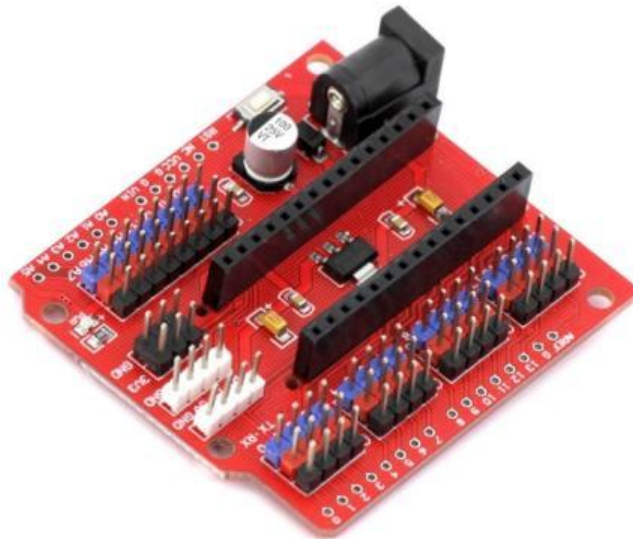


Рисунок 3.22 – Зовнішній вигляд розширення Arduino Nano Shield V3.0

Для використання плати не потрібно ніяких додаткових пристроїв, Arduino Nano v3.0 просто встановлюється в монтажні гнізда і можна починати роботу. Плата розширення підтримує зовнішнє живлення 7-12 В постійного струму, для підключення використовується стандартне гніздо. Крім того, присутній стабілізатор на 3.3 В і роз'єм для живлення пристроїв з такою напругою. Також на платі окремо виведена кнопка «Reset» [61].

Забезпечується підключення периферії до живлення 5 В і таким сигналам Arduino Nano:

- 14 цифрових входів-виходів з контактами живлення;
- 8 аналогових входів з контактами живлення;
- інтерфейси UART і I2C с контактами живлення;
- вхід опорного напруги АЦП [63].

На рисунку 3.23 зображено схему плати розширення.

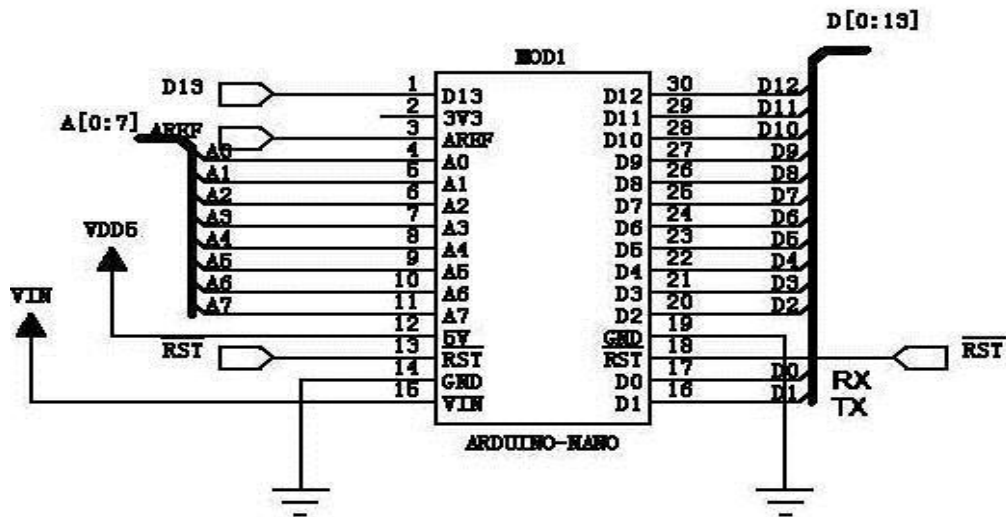


Рисунок 3.23 – Схема для плати розширення Arduino Nano Shield V3.0

### 3.3.3 Аналіз та вибір серводвигунів

Серводвигун – це мотор-редуктор, здатний повертати вихідний вал в задане положення (на заданий кут) і утримувати його в цьому положенні, всупереч опорам. Потрібно це в першу чергу для управління положеннями різних рухомих частин [64].

Офіційної класифікації серводвигунів не існує, виробники давно дотримуються декількох розмірів з загальноприйнятим розташуванням кріпильних елементів. Тож, за розмірами серводвигуни можна розділити на: маленькі, стандартні та великі. В таблиці 3.6 наведено характерні габарити двигунів кожної групи [65].

Таблиця 3.6 – Габарити серводвигунів для кожної групи

| Група      | Вага    | Лінійні розміри |
|------------|---------|-----------------|
| Маленькі   | 8-25 г  | 22×15×25 мм     |
| Стандартні | 40-80 г | 40×20×37 мм     |
| Великі     | 50-90 г | 49×25×40 мм     |

Однією з найважливіших характеристик розроблюваної електронної частини є компактність, саме тому вибір серводвигунів здійснювався серед маленьких.

Не менш важливою характеристикою при виборі серводвигуна є його момент сили, який показує, наскільки великий вантаж серводвигун здатний утримати в спокої на важелі заданої довжини. Якщо ця характеристика дорівнює  $5 \text{ кг} \times \text{см}$ , то це означає, що серводвигун утримає на вазі в горизонтальному положенні важіль довжини 1 см, на вільний кінець якого підвісили 5 кг. Або, що еквівалентно, важіль довжини 5 см, до якого підвісили 1 кг [65].

Серводвигун SG90 (або Tower Pro 9g SG90) є першою за популярністю моделлю серводвигунів, що найчастіше підключаються до проєктів Arduino, на другому місці знаходиться серводвигун моделі MG995 та MG996R, що є покращеною версією MG995. В таблиці 3.7 наведено порівняльну характеристику цих двох моделей [66-68].

Таблиця 3.7 – Порівняльна характеристика серводвигунів

| Характеристика            | SG90                       | MG995                                            | MG996R                                           |
|---------------------------|----------------------------|--------------------------------------------------|--------------------------------------------------|
| Швидкість обробки команди | 0,12с/60°                  | 0,20 с/60° (при 4,8 В),<br>0,16 с/60° (при 6 В)  | 0,17с/60° (при 4,8В)                             |
| Кут повороту              | від 0° до 180°             | Повертається на 120° (по 60 в кожному напрямку). | Повертається на 120° (по 60 в кожному напрямку). |
| Момент сили               | 1.2 кг/см при напрузі 4,7В | 8,5 кг/см (при 4.8 В),<br>10 кг/см (при 6 В)     | 9.4 кг/см (при 4.8 В)<br>11 кг/см (при 6.0 В)    |
| Живлення                  | 4,8 В                      | 4,8 В – 7,2 В                                    | 4,8 В – 7,2 В                                    |
| Робоча температура        | від -30С до 60С            | від 0С до -55С                                   | від 0С до -55С                                   |
| Розміри                   | 32x12x30 мм                | 40x20x42 мм                                      | 54x38x20 мм                                      |
| Вага                      | 9 г.                       | 55 г.                                            | 61 г.                                            |
| Група за розміром         | Маленькі                   | Середні                                          | Великі                                           |

В результаті порівнянь трьох серводвигунів, можна зробити висновок, що серводвигуни MG995 та MG996R за характеристиками є кращими та більшими за розмірами ніж SG90. Однак, серводвигун SG90 через свої маленькі розміри все ж таки є пріоритетним варіантом для створення прототипу. Крім того, моменту сили, що надає SG90, буде цілком достатньо для приведення в роботу захватного механізму руки.

Серводвигун SG90 (рис. 3.24), що використовується в основному для управління невеликими легкими механізмами, кут повороту яких обмежений діапазоном від 0 до 180 градусів, добре підходить для нескладних завдань під управлінням контролерів Arduino [66]. Для роботи з SG90 на нього необхідно встановити пластикові плечі (насадки) та закріпити спеціальними гвинтами [64]. Всі деталі наявні в комплекті у кількості трьох одиниць (рис. 3.25).



Рисунок 3.24 – Зовнішній вигляд серводвигуна SG90



Рисунок 3.25– Комплектація серводвигуна SG90

Основні характеристики SG90 наведено в таблиці 3.8. На рисунку 3.26 представлено схему серводвигуна [64-67, 69].

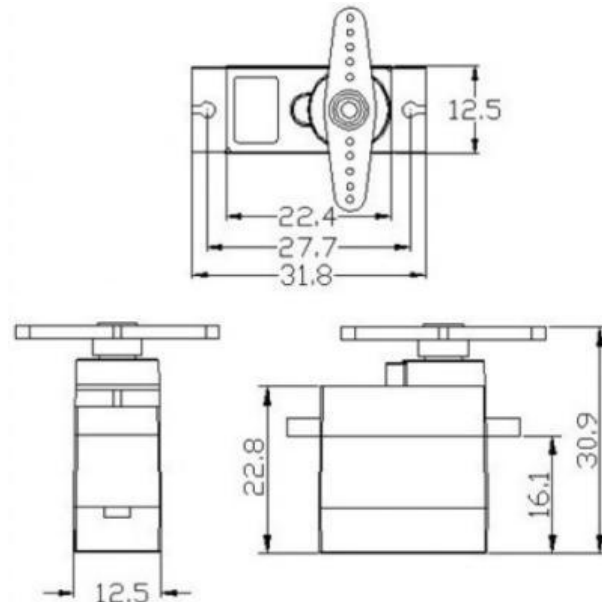


Рисунок 3.26 – Схема розмірів серводвигуна SG90

Таблиця 3.8 – Характеристики SG90

| Характеристика                       | Значення                        |
|--------------------------------------|---------------------------------|
| Вага                                 | 9 гр.                           |
| Розмір                               | 21.5мм X 11.8мм X 22.7мм        |
| Швидкість без навантаження           | 60 градусів / 0.12 сек. (4.8 В) |
| Момент (зусилля)                     | 1.2 кг * см при напрузі 4,7В    |
| Пусковий момент                      | 1.2 - 1.4 кг / см (4.8 В)       |
| Робоча температура                   | від -30 до +60                  |
| Робоча напруга                       | 4.8 - 6 В                       |
| Струм, що споживається при русі      | 50-80 мА                        |
| Струм, що споживається при утриманні | 5-10 мА                         |
| Ширина мертвої зони                  | 4 мікросекунди                  |
| Довжина кабеля                       | 23 см                           |
| Механіка                             | Аналогова                       |

Для роботи з SG90, його необхідно підключити до плати Arduino для цього важливо врахувати схему підключення серводвигуна:

- коричневий дрiт – земля (Ground, підключається до піну GND на платі Arduino)
- червоний дрiт – харчування +5 V (підключається до піну 5V на платі Arduino)
- жовтий (помаранчевий) провід – сигнал управління (підключається до цифрового піну Arduino).

Програмування серводвигуна передбачає:

- `#include <Servo.h>` - підключення бібліотеки Servo, яка полегшує роботу з різними серводвигунами і значно спрощує програмний код;
- `servo myservo` – оголошення серводвигуна;
- у розділі `setup ()` призначаємо пін управління серводвигуном за допомогою функції `attach ()`, для якої можливі два варіанти синтаксису: `servo.attach (pin)` і `servo.attach (pin, min, max)`. При цьому `pin` – номер піна, до якого приєднують серводвигун, `min` і `max` – довжини імпульсів в мікросекундах, що відповідають за кути повороту  $0^\circ$  і  $180^\circ$ . За замовчуванням виставляються рівними 544 мкс і 2400 мкс відповідно. Далі, всередині блоку програми, що виконується, керуємо становищем серводвигуну (рис. 3.27) [66].

```

1  #include <Servo.h> // Підключення бібліотеки
2
3  Servo myservo; //Оголошення сервоприводу
4
5  void setup()// призначаємо пін управління сервоприводом
6  {
7      myservo.attach(9);
8  }
9  //Керування роботою сервоприводу
10 void loop()
11 {
12     myservo.write(60); //встановлюємо сервопривід в положення 60 градусів
13     delay(100);
14 }
```

Рисунок 3.27 – Програмування серводвигуна SG90

Серводвигун SG90 приймає задане положення дуже швидко, щоб уповільнити його поворот необхідно використовувати цикл повороту з кроком в 1 градус і затримкою між кожним кроком.

До корисних функцій бібліотеки Servo, окрім вже вище вказаних належать:

- `writeMicroseconds ()` – повертає команду послати на серводвигун імпульс певної довжини, є аналогом команди `write ()`. Синтаксис наступний: `servo.writeMicroseconds (uS)`, де `uS` – довжина імпульсу в мікросекундах;
- `read ()` – читає поточне значення кута, в якому знаходиться серводвигун. Синтаксис наступний: `servo.read ()`, повертається ціле значення від 0 до 180;
- `attached ()` – перевірка, чи була приєднана змінна до конкретного піну. Синтаксис наступний: `servo.attached ()`, повертається логічна істина, якщо змінна була приєднана до якого-небудь піну, або брехня в зворотному випадку;
- `detach ()` – виробляє дію, зворотне дії `attach ()`, тобто від'єднує змінну від піна, до якого вона була приписана. Синтаксис наступний: `servo.detach ()`.

При роботі з серводвигунами можуть виникнути певні складнощі. Так недорогі не забезпечують точних налаштувань початкової і кінцевої позицій. Окрім того, часто в положенні 0 і 180 градусів чути характерний тріск або дзижчання, це говорить про те, що механічна частина серводвигуну вже знаходиться в граничній позиції, але датчик вважає, що задана позиція ще не досягнута. Проблему можна вирішити підбором скоригованих позицій. Наприклад, часто замість позиції 0 градусів потрібно використовувати позицію повороту серводвигуну в 10-15 градусів, а кінцеву 165-170 градусів. Це дозволить уникнути зайвих перевантажень і шуму при досягненні крайніх точок. Щоб вказати серводвигуну бажане положення необхідно надсилати керуючий сигнал. Керуючий сигнал – імпульси постійної частоти і змінної

ширини. Те, яке положення повинен зайняти серводвигун, залежить від довжини імпульсів. Коли сигнал надходить в керуючу схему, наявний в ній генератор імпульсів виробляє свій імпульс, тривалість якого визначається через потенціометр. Інша частина схеми порівнює тривалість двох імпульсів. Якщо тривалість різна, включається електромотор. Напрямок обертання визначається тим, який з імпульсів коротше. Якщо довжини імпульсів рівні, електромотор зупиняється. Найчастіше імпульси виробляються з частотою 50 Гц. Це означає, що імпульс випускається і приймається раз в 20 мс. Зазвичай при цьому тривалість імпульсу в 1520 мкс означає, що серводвигун повинен зайняти середнє положення. Збільшення або зменшення довжини імпульсу змусить його повернутися за годинниковою або проти годинникової стрілки відповідно. При цьому існує верхня і нижня межа тривалості імпульсу. У бібліотеці Servo для Arduino за замовчуванням виставлені наступні значення довжин імпульсу: 544 мкс – для  $0^\circ$  і 2400 мкс – для  $180^\circ$  (рис. 3.28) [70].

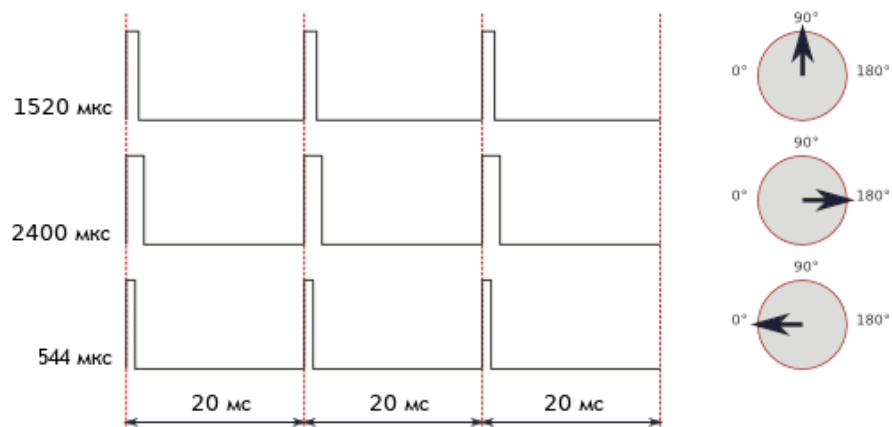


Рисунок 3.28 – Напрямок обертання серводвигуна

Слід звертати увагу на напругу, яка використовується при роботі серводвигуна. Якщо значення напруги перевищує допустимі значення з технічних характеристик серводвигуна, механічні частини можуть вийти з ладу або зубчасті колеса повернути в положення, що перевищує

максимальні значення і механізм не зможе продовжувати рух в зворотному напрямку. Іншими словами, завищена напруга швидше за все не призведе до згорання двигуна, але пошкодить механічні частини зубчастих механізмів [66].

### 3.4 Інтеграція компонентів електронної та механічної частин прототипу кіберпротезу

Для створення прототипу кіберпротезу буде використано:

- 5 серводвигунів Tower Pro 9g SG90;
- плата Arduino Nano;
- плата розширення Arduino Nano Shield V3.0.

Прототип руки закінчується захватним механізмом, відповідно, для приведення його в дію необхідно розмістити 5 серводвигунів на фізичній частині прототипу руки, по одному на кожен палець, та закріпити їх (рис. 3.29).



Рисунок 3.29 – Закріплення сервоприводів на механічній частині

Після чого необхідно підключити плату Arduino Nano та серводвигуни до плати Arduino Nano Shield V3.0 (рис. 3.30).

Для підключення Arduino Nano Shield V3.0 потрібно акуратно «надіти» його на основну плату. Зазвичай контакти Shield типу гребінки (тато) легко вставляються в роз'єми плати Arduino. Якщо сама плата спаяна неакуратно, то можуть виникнути проблеми з встановлення плати на Shield, в такому випадку потрібно акуратно підправити «штирки», головне діяти акуратно і не додавати зайвої сили (рис. 3.31).

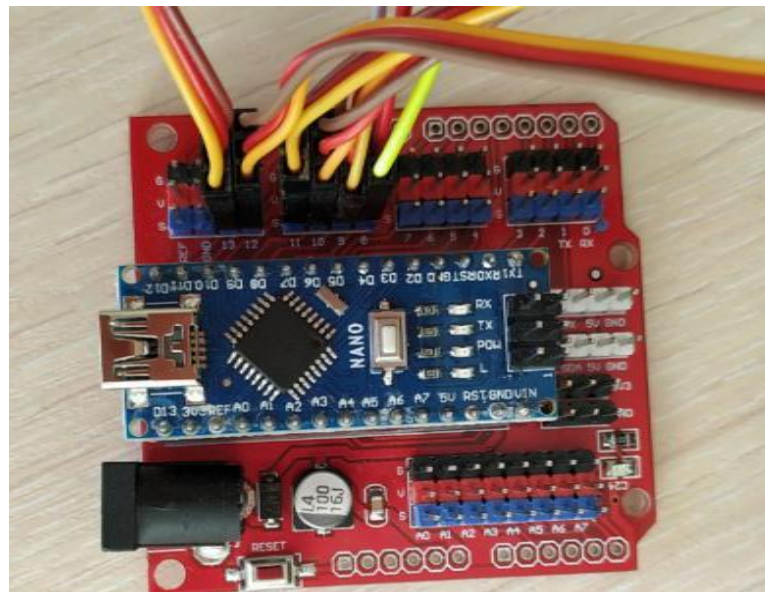


Рисунок 3.30 – Підключення компонентів до плати Arduino Nano Shield V3.0

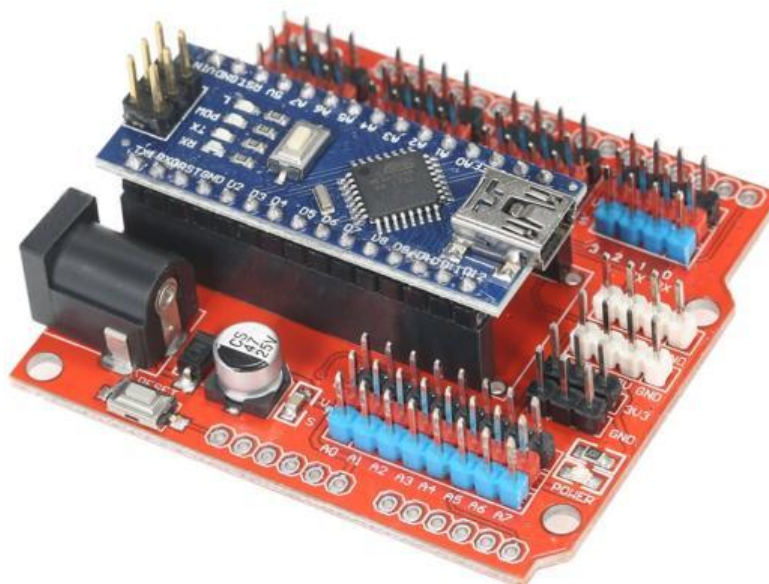


Рисунок 3.31 – Підключення Arduino Nano до плати розширення

Серводвигун має три контакти, які пофарбовані в різні кольори. Коричневий дріт веде до землі, червоний – до живлення +5 В, дріт помаранчевого або жовтого кольору – сигнальний. До Arduino він підключається через плату розширення як показано на рисунку 3.30. Помаранчевий провід (сигнальний) підключається до цифрового піну, коричневий і червоний – до землі та живлення відповідно.

Спрощену схему підключення сервоприводів та плати Arduino Nano наведено на рис. 3.32 з урахуванням шостого серводвигуна, що має забезпечити обертання кисті.

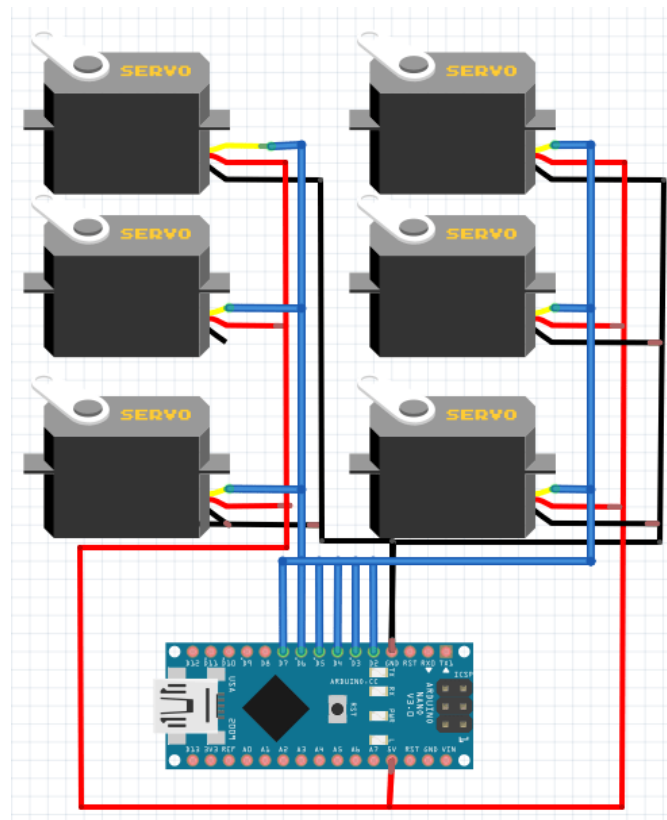


Рисунок 3.32 – Спрощена схема підключення

Кожному пальцю на руці відповідає свій серводвигун, що відповідно під'єднаний до різних пінів на платі. На рисунку 3.33 пронумеровано кожний з серводвигунів, а в таблиці 4.1 розписано піни, що їм відповідають.

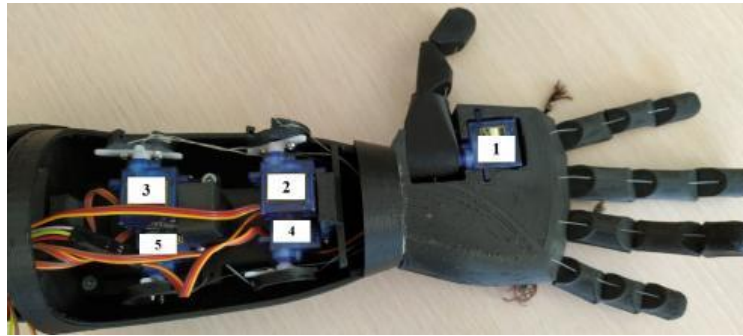


Рисунок 3.33 – Нумерація серводвигунів

Таблиця 3.9 – Відповідність серводвигунів пінам на платі

| Номер серводвигуна | Палець                        | Номер піна |
|--------------------|-------------------------------|------------|
| 1                  | Великий                       | pin 8      |
| 2                  | Перша фаланга великого пальця | pin 11     |
| 3                  | Вказівний та середній         | pin 9      |
| 4                  | Безіменний                    | pin 10     |
| 5                  | Мізинець                      | pin 12     |

### 3.5 Розробка програмного забезпечення для прототипу

На сьогоднішній день існує багато середовищ програмування для Arduino. Але Далі більш докладно розглянемо середовища програмування Arduino IDE.

Стандартне середовище розробки Arduino IDE використовується для роботи всіх різновидів плат Arduino з комп'ютером.

Після підключення плати до комп'ютера з нею можна працювати та програмувати її. Для того, щоб рука виконувала рухи необхідно написати та завантажити в контролер програмне забезпечення, код якого наведено у додатку А.

Розроблене програмне забезпечення реалізує роботу п'яти пальців одночасно, тобто їх згинання та розгинання. Для початку роботи необхідно підключити бібліотеку Servo, що дозволяє здійснювати програмне керування серводвигунами. Синтаксис є наступним `# include "Servo.h"`. Для роботи з серводвигунами вказується змінна типу Servo. Управління здійснюється наступними функціями:

- `attach ()` – приєднує змінну до конкретного піну, де `pin` – номер піна, до якого приєднують серводвигун;
- `write ()` – повертає команду серводвигуну прийняти деяке значення параметра. Синтаксис наступний: `servo.write (angle)`, де `angle` – кут, на який повинен повернутися серводвигун.

Якщо необхідно зігнути одночасно тільки конкретні пальці, наприклад, мізинець та безіменний, то потрібно винести зміни в `void loop ()`, тобто вказати значення відповідних пінів в ньому. Значення пінів для кожного з пальців були вказані в таблиці 3.9.

Алгоритм управління кіберпротезом верхніх кінцівок людини (рис.3.34) складається з наступних етапів:

Блок 1. Початок.

Блок 2. Ініціалізація сервоприводів.

Блок 3. Ввімкнення сервоприводів.

Блок 4. Перевірка поточної позиції сервопривода.

Блок 5. Повернення на початкову позицію.

Блок 6. Перевірка: чи слід продовжити роботу з експериментом?.

Блок 7. Виконання експерименту, тобто приведення сервоприводів в дію, для згинання пальців.

Блок 8. Завершення експерименту.

Програма керування кіберпротезом складається з наступних програмних компонентів:

- бібліотека для роботи з сервоприводами (`#include "Servo.h"`);

- клас керування кіберпротезом (class Finger);
- опис змінних (Servo servo, byte pos);
- опис функцій (init(int16\_t interval, byte pin); void update(); void setup(); void loop()).

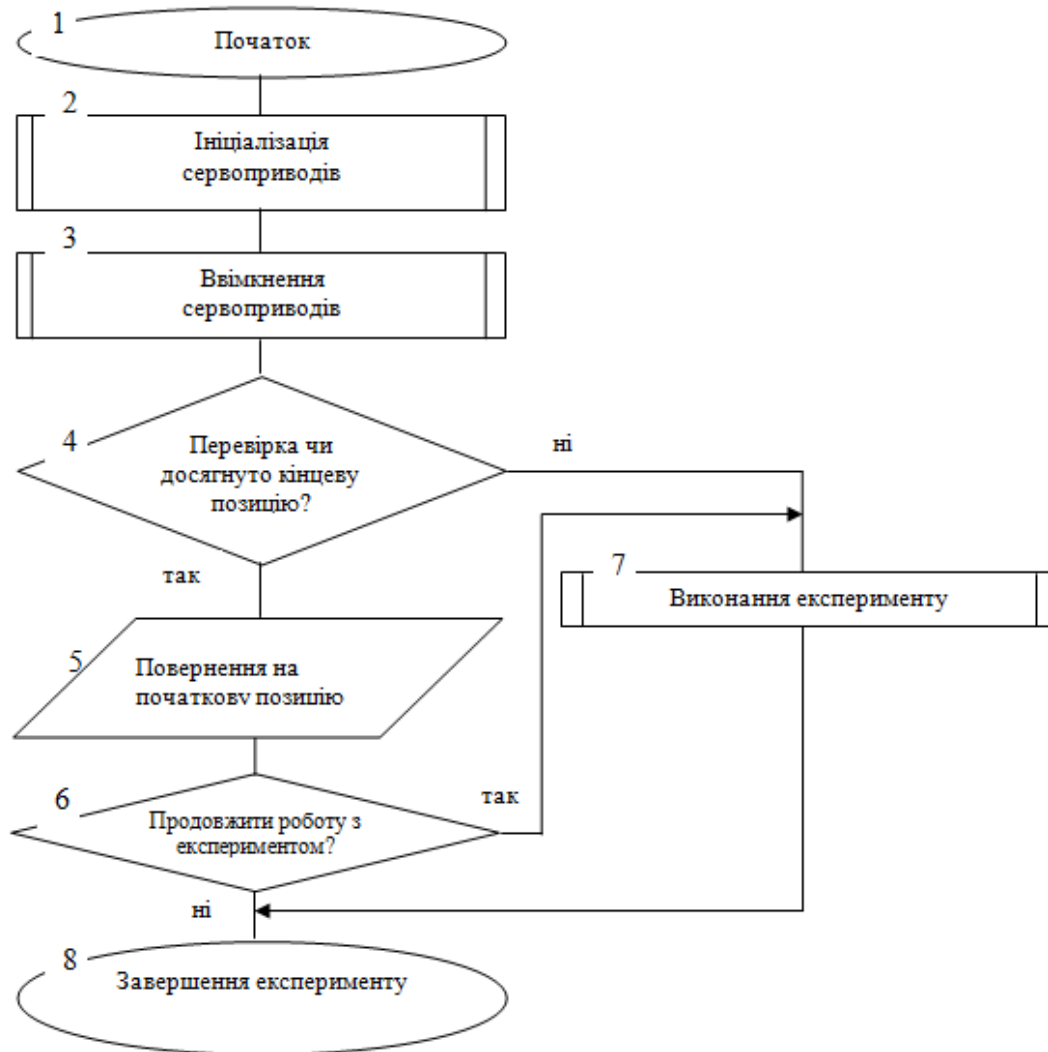


Рисунок 3.34 – Схема алгоритму програми управління кіберпротезом

Тестування роботи програми виконувалося для різних сценаріїв:

- два пальці зігнуті, а три розігнуті (рис. 3.35);
- один палець зігнутий (рис. 3.36);
- всі пальці зігнуті (рис. 3.37);
- всі пальці розігнуті (рис. 3.38).

Крім того, було виконано перевірку відповідності форми та розмірів кіберпротезу руці реальної людини на основі побудови цифрового двійника в програмі Skanect (рис. 3.39).



Рисунок 3.35 – Сценарій 1



Рисунок 3.36 – Сценарій 2



Рисунок 3.37 – Сценарій 3



Рисунок 3.38 – Сценарій 4

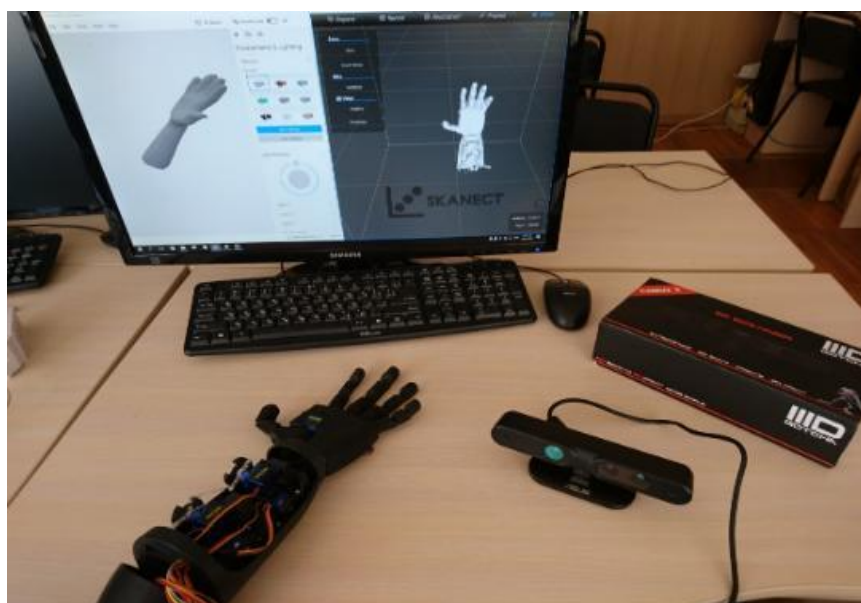


Рисунок 3.39 – Робота з цифровим двійником в програмі Skanect

### 3.6 Висновки до розділу

У даному розділі розроблено метод комплексного віртуального та фізичного прототипування кіберпротезу та інформаційну технологію створення механічної частини прототипу. Було спроектовано 3D-модель механічної частини кіберпротезу в середовищі Creo Parametric з застосуванням параметричного моделювання при розробці типових деталей (пальців). Виконано аналіз та вибір компонентів електронної частини прототипу: основної плати Arduino Nano, плати розширення Arduino Nano Shield V3.0, серводвигуна SG90. Розглянуто питання створення електронної частини розроблюваного прототипу на основі обраних компонентів, їх підключення до плати розширення, розташування елементів на фізичному прототипі, налаштування середовища програмування для роботи з платою та створення з його допомогою програми для функціонування прототипу. Функціональне тестування підтвердило працездатність прототипу кіберпротезу, а порівняння з цифровим двійником – відповідність форми та розмірів.

## 4 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

### 4.1 Планування розробки програмного виробу

Розроблене програмне забезпечення для керування кібер-протезом дозволяє підвищити точність та надійність роботи обладнання.

Очікувана кількість покупців 300 на рік. Планується продавати дану систему за 600 грн за копію.

Весь комплекс розробки програмного забезпечення можна поділити на етапи. Для кожного етапу вказується трудомісткість, кількість виконавців та тривалість робіт. У розробці приймають участь програміст впродовж 2,5 місяців і консультант впродовж 0,5 місяця.

Отримана інформація зведена в табл. 4.1., за даними якої складається зведений стрічковий графік планування розробки програмного продукту, який представляє собою таблицю, в першому стовпці якої розміщені в порядку збільшення термінів початку виконання всі види роботи, а навпроти – календарний період їх виконання.

Тривалість робіт розраховується за формулою 4.1:

$$T_{\eta} = \frac{Q}{R}, \quad (4.1)$$

де  $T_{\eta}$  – довжина циклу, днів;

$Q$  – трудомісткість, людино-днів;

$R$  – кількість виконавців, люд.

Необхідні початкові дані:

– час на розробку – 2,5 міс.;

– консультант працює – до 0,5 міс. (13 дн.) із заробітною платою 8000 грн./міс;

– програміст працює – 2,5 міс. (55 дн.) із заробітною платою 5700 грн./міс.

Таблиця 4.1 – Характеристика робіт по розробці програми

| Найменування роботи                                                   | Трудовісткість |                  | Виконавці          |              | Трива-<br>лість |
|-----------------------------------------------------------------------|----------------|------------------|--------------------|--------------|-----------------|
|                                                                       | чол.-<br>дн.   | % до<br>підсумку | спеціаль-<br>ність | кіл.<br>чол. | днів            |
| 1. Дослідження вимог до<br>проектowanego продукту                     | 10             | 14,7             | програміст         | 2            | 5               |
|                                                                       |                |                  | консультант        |              |                 |
| 2. Вибір інструментів<br>проектування, дослідження<br>існуючих рішень | 4              | 5,8              | програміст         | 1            | 4               |
| 3. Проектування загальної<br>структури програми                       | 10             | 14,7             | програміст         | 2            | 5               |
|                                                                       |                |                  | консультант        |              |                 |
| 4. Розробка схеми<br>функціонування програми                          |                | 2,9              | програміст         |              | 2               |
| 5. Створення програмного коду                                         | 3              | 33,8             | програміст         |              | 23              |
| 6. Тестування та налагодження<br>програмного продукту                 | 3              | 19,1             | програміст         |              | 13              |
| 7. Складання програмної<br>документації                               |                | 8,8              | програміст         |              | 3               |
|                                                                       |                |                  | консультант        |              |                 |
| Всього                                                                | 8              | 100,0            |                    |              | 55              |

Останнім етапом планування розробки програмного продукту є звідний стрічковий графік розробки програмного продукту є лінійний графік Ганта (табл. 4.2), побудований за даними таблиці 4.1.

Таблиця 4.2 – Лінійний графік Ганта розробки програмного продукту

| Найменування роботи                                             | 23.09 | 30.09 | 07.10 | 14.10 | 21.10 | 28.10 | 04.11 | 11.11 | 18.11 | 25.11 | 2.12 |
|-----------------------------------------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| 1. Дослідження вимог до проєктованого продукту                  |       |       |       |       |       |       |       |       |       |       |      |
| 2. Вибір інструментів проєктування, дослідження існуючих рішень |       |       |       |       |       |       |       |       |       |       |      |
| 3. Проєктування загальної структури програми                    |       |       |       |       |       |       |       |       |       |       |      |
| 4. Розробка схеми функціонування програми                       |       |       |       |       |       |       |       |       |       |       |      |
| 5. Створення програмного коду                                   |       |       |       |       |       |       |       |       |       |       |      |
| 6. Тестування та налагодження програмного продукту              |       |       |       |       |       |       |       |       |       |       |      |
| 7. Складання програмної документації                            |       |       |       |       |       |       |       |       |       |       |      |

————— – програміст

..... – консультант

## 4.2 Визначення витрат на розробку програми

Для визначення витрат на розробку програми складається калькуляція кошторисної вартості робіт яка включає наступні статті [1]:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок;
- витрати на спеціальне обладнання;
- матеріали і комплектуючі вироби;
- загальновиробничі витрати.

### 4.2.1 Розрахунок основної заробітної плати

Витрати по цій статті складаються з планового фонду заробітної плати усіх категорій працівників, зайнятих в розробці програмного продукту. Розрахунок заробітної плати ведеться на основі відомостей про трудомісткість (табл. 4.1). Результати розрахунків зведені в таблиці 4.3.

Таблиця 4.3 – Розрахунок основної заробітної плати

| Посада<br>виконавця | Кіль-<br>кість,<br>осіб | Деяний<br>оклад | Кіль-<br>кість,<br>днів<br>роботи | Премії та<br>доплати,<br>грн. | Середня<br>місячна<br>зарплата,<br>грн. | Сумарна<br>зарплата,<br>грн. |
|---------------------|-------------------------|-----------------|-----------------------------------|-------------------------------|-----------------------------------------|------------------------------|
| Програміст          | 1                       | 259,09          | 55                                | —                             | 5700                                    | 14250                        |
| Консультант         | 1                       | 307,69          | 13                                | —                             | 8000                                    | 4000                         |
| Всього              | 2                       |                 |                                   |                               |                                         | 18250                        |

Розрахунок заробітної плати проводиться за формулою:

$$ЗП_{\text{осн}} = \sum_1^i (ЗП_{\text{мес}_i} \times T_{pi}), \quad (4.2)$$

де  $ЗП_{\text{мес}_i}$  – щомісячна заробітна плата і-того працівника, грн;

$T_{pi}$  – час, витрачений на розробку програми і-тим працівником, міс.

#### 4.2.2 Розрахунок додаткової заробітної плати

Додаткову заробітну плату приймають рівною 10% від основної заробітної плати. Розрахунок основної та додаткової заробітної плати наведений в табл.4.4.

Таблиця 4.4 – Розрахунок додаткової заробітної плати

| Посада<br>виконавця | Додаткова заробітна<br>плата, грн. | Сума основної та додаткової<br>заробітної плати, грн. |
|---------------------|------------------------------------|-------------------------------------------------------|
| Програміст          | 1425                               | 15675                                                 |
| Консультант         | 400                                | 4400                                                  |
| Всього              | 1825                               | 20075                                                 |

#### 4.2.3 Відрахування на єдиний соціальний внесок

Відрахування на єдиний соціальний внесок визначають у відсотковому відношенні від суми основної і додаткової зарплат, з урахуванням премій і доплат. Премії і доплати в нашому випадку дорівнюють 0. Відрахування на соціальне страхування складають 22%. Розраховуємо за формулою 4.3:

$$ВІД = (ЗП_{осн} + ЗП_{дод}) \times 0,22, \quad (4.3)$$

де  $ЗП_{осн}$  – основна заробітна плата, грн.

$ЗП_{дод}$  – додаткова заробітна плата, грн.

Отже, відрахування на єдиний соціальний внесок дорівнюють:

$$ВІД = (18250 + 1825) \times 0,22 = 4416,5 \text{ грн.}$$

#### 4.2.4 Визначення витрат на матеріали

Матеріали, які знадобилися для розробки даної програми представлені в табл. 4.5.

Таблиця 4.5 – Витратні матеріали

| Матеріал           | Кількість  | Вартість, грн. |
|--------------------|------------|----------------|
| Папір формату А4   | 2 упаковки | 170            |
| Олівці             | 50 шт.     | 5              |
| CD-RW диск         | 5 шт.      | 30             |
| Тонер для принтера | 1 шт.      | 250            |

Витрати на матеріали розраховуються за формулою 4.4:

$$M = \sum_{i=1}^n (C_i \times N_i \times (1 + K_{Т.З.}) - C_{io} \times N_{io}), \quad (4.4)$$

де  $M$  – витрати на матеріали, покупні напівфабрикати і комплектуючі вироб, грн.;

$K_{ТЗ}$  – коефіцієнт, що враховує транспортно-заготівельні витрати (0,03 - 0,1);

$Ц_i$  – ціна  $i$ -го найменування матеріалу, напівфабрикату, комплектуючого, грн.;

$N_i$  – потреба в  $i$ -му матеріалі, напівфабрикаті, комплектуючого;

$Ц_{io}$  – ціна поворотних відходів  $i$ -го найменування матеріалу, грн.;

$N_{io}$  – кількість зворотних відходів  $i$ -го найменування

$n$  – кількість найменувань матеріалів, напівфабрикатів, комплектуючих.

Розрахуємо витрати на матеріали:

$$M = 2 \times (1 + 0,1) \times 170 + 50 \times (1 + 0,1) \times 5 + 30 \times (1 + 0,1) \times 5 + (1 + 0,1) \times 250 = 1089 \text{ грн.}$$

Разом витрати на матеріали склали 1089 грн.

#### 4.2.5 Витрати на спеціальне устаткування

У цю статтю входять витрати на оплату машинного часу ЕОМ. Витрати на оплату машинного часу ЕОМ для написання і відлагодження цієї програми визначають по формулі:

$$C_{mo} = P_{\text{ЭКС}} \times t_{mo} \quad (4.5)$$

де  $C_{mo}$  – витрати на оплату машинного часу, грн.;

$P_{\text{ЭКС}}$  – експлуатаційні витрати на одну годину машинного часу, грн.;

$t_{\text{МО}}$  – машинний час ЕОМ для написання і відладки цього програмного продукту, ч.

Для розрахунку експлуатаційних витрат на одну годину машинного часу складемо кошторис витрат на зміст і експлуатацію цього устаткування.

Амортизаційні відрахування визначають по формулі:

$$A = \Phi_{\delta} \times \frac{H_a}{100} \quad (4.6)$$

Де  $\Phi_{\delta}$  – балансова вартість обчислювальної техніки, грн.;

$H_a$  – норма амортизаційних відрахувань на повне відновлення обчислювальної техніки, %.

Балансова вартість комп'ютерної обчислювальної техніки визначається як ціна обчислювальної техніки, помножена на коефіцієнт, що враховує транспортно-заготівельні витрати і монтаж обладнання ( $K=0,05$ ).

Балансова вартість обчислювальної техніки у кількості однієї одиниці за ціною 20000 становить  $20000 \times (1+0,05) = 21000$  грн., а норма амортизаційних відрахувань – 25%.

Таким чином амортизаційні відрахування становлять  $A = 21000 \times 0,25 = 5250$  грн.

Стаття "Експлуатація обладнання" розраховується підсумовуванням витрат на силову електроенергію і допоміжні матеріали.

Витрати на електроенергію:

$$C_{\text{э}} = N_{\text{н}} \times \Phi_{\text{эф}} \times K_{\text{зв}} \times K_{\text{зм}} \times Ц_{\text{э}}, \quad (4.7)$$

де  $N_{\text{н}}$  – номінальна потужність ЕОМ, кВт;

$\Phi_{\text{эф}}$  – річний ефективний фонд часу роботи ЕОМ, год.;

$K_{\text{зв}}$  – середній коефіцієнт завантаження за часом;

$K_{зм}$  – коефіцієнт завантаження за потужністю;

$Ц_э$  – ціна одного кВт×год електроенергії, грн.

Номінальна потужність ЕОМ = 0,3 кВт.  $\Phi_{эф}$  = 440 год. Середній коефіцієнт завантаження за часом – 0,45. Коефіцієнт завантаження по потужності – 0,7. Ціна одного кВт×год електроенергії – 2,11 грн.

Таким чином,  $C_э = 0,33 \times 440 \times 0,45 \times 0,7 \times 2,11 = 96,50$  грн.

Зарплата обслуговуючого персоналу розраховується за формулою:

$$ЗП_{ОБСЛ} = \Phi ЗП_{Г} \times (1 + K_{ОТЧ}) \times \frac{t_{ОБСЛ}}{\Phi_{ЭФ.ОБСЛ}}, \quad (4.8)$$

де  $\Phi ЗП_{Г}$  – річний фонд зарплати обслуговуючих працівників, грн.;

$K_{ОТЧ}$  – коефіцієнт відрахувань на соціальне страхування і до інших фондів, %;

$t_{ОБСЛ}$  – час впродовж року, необхідний на техобслуговування ЕОМ, год/рік;

$\Phi_{ЭФ.ОБСЛ}$  – річний ефективний фонд часу обслуговуючого персоналу, год/рік.

Машину обслуговує один адміністратор з основною заробітною платою 8000 грн./міс. Річний фонд заробітної плати становить відповідно 96000 грн. На одну ЕОМ обслуговуючий витрачає по 6 % годині на місяць, що в рік становить 6 годин. Час, необхідний на технічне обслуговування ЕОМ протягом року, дорівнює 1750 год.

Таким чином,  $ЗП_{ОБСЛ} = 96000 \times (1 + 0,22) \times 6 / 1750 = 401,55$  грн.

Сума витрат по статті "Поточний ремонт устаткування" дорівнює 3% від балансової вартості устаткування, що дорівнює  $5250 \times 0,03 = 157,50$  грн.

Інші витрати складають 4% від суми попередніх:

$$(7875 + 722,17 + 401,55 + 236,25) \times 0,04 = 369,39 \text{ грн.}$$

Зведемо отримані дані в табл. 4.6.

Таблиця 4.6 – Кошторис витрат на зміст і експлуатацію устаткування

| Найменування статей витрат                         | Сума, грн. |
|----------------------------------------------------|------------|
| Амортизація устаткування                           | 5250       |
| Експлуатація устаткування                          | 96,50      |
| Зарплата обслуговуючого персоналу з відрахуваннями | 401,55     |
| Поточний ремонт устаткування                       | 157,50     |
| Інші витрати                                       | 5905,55    |
| Всього                                             | 5250       |

#### 4.2.6 Загальновиробничі витрати

До загальновиробничих витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати і так далі), зміст і експлуатацію будинків. Загальновиробничі витрати включаються у вартість розробки програми непрямым шляхом – у відсотках до основної заробітної плати розробників. В даному випадку загальновиробничі витрати складають 20% до основної заробітної плати розробників – 3650 грн. Результати калькуляції кошторисної вартості робіт по розробці програми і ціна програмного продукту наведені у таблиці 4.7.

Таблиця 4.7 – Калькуляція кошторисної вартості робіт по розробці програми і ціна програмного продукту

|   | Найменування статей витрат       | Сума, грн. | Питома вага до підсумку, % |
|---|----------------------------------|------------|----------------------------|
| 1 | Основна заробітна плата          | 18250      | 59,32                      |
| 2 | Додаткова заробітна плата        | 1825       | 5,93                       |
| 3 | Соціальні відрахування           | 4416,5     | 14,36                      |
| 4 | Витрати на матеріали             | 1089       | 3,54                       |
| 5 | Витрати на оплату машинного часу | 1535,04    | 4,99                       |
| 6 | Загальновиробничі витрати        | 3650       | 11,86                      |
| 7 | Разом витрати                    | 30765,54   | 100                        |
| 8 | Прибуток (30%)                   | 9229,66    | —                          |
| 9 | Всього                           | 39995,20   | —                          |

### 4.3 Оцінка економічної ефективності

Витрати на розробку програми складають 39995,20 грн. Очікується продати 300 копій за мінімальною ціною 600 грн. за копію. З урахуванням ризику 50% надходження від продажів складуть 90000 грн. Це значно перевищує ціну розробки в якій вже була врахована рентабельність 30%. Таким чином можна зробити висновок про економічну доцільність розробки даного програмного забезпечення для управління кібер-протезом руки.

## 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

### 5.1 Аналіз потенційних небезпек

Так як в дипломному проекті проводиться дослідження та практична реалізація прототипу кібер-протезу, тому можливі наступні потенційні небезпеки, що можуть виникнути під час роботи на робочому місці інженера-радіотехніка:

- підвищення рівню шуму, за рахунок роботи вентиляційних компонентів комп'ютерної техніки, пристроїв введення/виведення, ксероксів, принтерів, кондиціонерів та інших електричних пристроїв;
- погано організоване, захаращене робоче місце, нераціональне розміщення обладнання;
- небезпека ураження електричним струмом внаслідок порушення правил експлуатації електроприладів або їх несправності;
- підвищення рівнів електромагнітних полів;
- підвищення рівню іонізуючого випромінювання;
- підвищений рівень статичної електрики;
- підвищення або зниження температури або вологості повітря, а також його надмірна загазованість або запиленість;
- погане або недостатнє освітлення робочого місця природнім або штучним джерелом світла;
- надмірно великий об'єм інформації, що необхідно обробити за одиницю часу;
- тривале сидяче положення;
- навантаження на органи зору;
- інтелектуальне та нервово-емоційним навантаженням;

– кістково-м'язове напруження, що виникає через необхідність працівників тривалий час знаходитись в позах, що вимагають статичного напруження м'язів спини, шиї, рук, ніг, що призводить до їх стомлення.

## **5.2 Заходи по забезпеченню техніки безпеки**

В офісних приміщеннях для роботи використовується велика кількість різноманітних електроприладів: принтери, ксерокси, сканери, комп'ютери, проектори, освітлювальні прилади, кондиціонери, обігрівачі, тощо, які живляться від мережі з напругою 220 В. Саме тому небезпека ураження електричним струмом є однією з найбільших небезпек, яка може виникнути в разі недотримання правил користування ними.

Щоб запобігти випадкам ураження електричним струмом відповідно до Правил улаштування електроустановок (ПУЕ) [71], частини обладнання, що проводять електричний струм та можуть стати джерелами небезпеки, надійно ізолювані та розміщені в недоступному для співробітників місці. У випадку, якщо людина доторкнулась до неструмопровідних частин, які в наслідок пошкодженої ізоляції опинились під напругою, для забезпечення її безпеки, у приміщенні офісу використовується занулення та заземлення обладнання відповідно до ДСТУ 7237:2011.

В усіх приміщеннях, де розміщені робочі місця, заземлені конструкції (батареї опалення, водопровідні труби кабелі із заземленим відкритим екраном), надійно ізолювані діелектричними щиткам або сітками, щоб уникнути небезпечної ситуації потрапляння людини під напругу.

Відповідно до НПАОП 0.00-4.12-05, затвердженого наказом голови Держкомітету України з нагляду за охороною праці 26.01.2005 № 15, всі працівники проходять навчання та перевірку знань з питань охорони праці, як під час прийняття на роботу так і періодично, один раз на три роки.

До роботи допускаються тільки особи, які пройшли затверджений на

підприємстві курс охорони праці для роботи з комп'ютером [72].

Для захисту працівників від ураження електричним струмом проводяться організаційні та технічні заходи захисту.

Організаційні заходи включають в себе:

- призначення осіб, що будуть відповідати за організацію безпеки;
- навчання персоналу правилам безпеки та експлуатації обладнання;
- проведення інструктажів;
- використання попереджувальних плакатів і знаків (ДСТУ ISO

7010:2009).

- До технічних заходів забезпечення безпеки працівників належать:
- захисне заземлення, занулення, відмикання;
- електричний поділ мережі;
- вирівнювання потенціалів (ДСТУ 2815-94);
- обмеження сили струму;
- забезпечення недоступності неізольованих частин [73].

Для забезпечення безпеки працівників на підприємствах, де використовують комп'ютери забороняється:

- проводити ремонт та технічне обслуговування комп'ютера на робочому місці працівника;
- самостійно ремонтувати або намагатись виконати технічне налагодження комп'ютера без залучення компетентних спеціалістів;
- розміщувати на робочому місці зайві документи, деталі та предмети, що не потрібні для роботи;
- використовувати монітори з нечітким зображенням та монітори, у яких наявні поламки екрану;
- працювати з матричним принтером без антивібраційного покриття та зі знятою кришкою;

- експлуатація кабелів та проводів з пошкодженою ізоляцією або такою, що втратила захисні властивості в ході експлуатації;
- використання саморобних подовжувачів або нестандартного (саморобного) електронагрівального обладнання для опалення приміщення;
- користуватись пошкодженими розетками.

Також експлуатація електроустановок і електроустаткування проводиться згідно з НПАОП 40.1-1.01-97 «Правила безпечної експлуатації електроустановок» та НПАОП 40.1-1.21-98 «Правила безпечної експлуатації електроустановок споживачів», а також рекомендацій, що надавав виробник [72].

### **5.3 Заходи з виробничої санітарії та гігієни праці**

Робочі місця офісних працівників, обладнані персональними комп'ютерами, відповідають вимогам «Правил охорони праці під час експлуатації електронно – обчислювальних машин», що затверджені Наказом Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду від 26.03.2010 року №65 та «Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно – обчислювальних машин», затверджених постановою Головного державного санітарного лікаря України від 10.12.1998 року №7 (ДСанПіН 3.3.2-007-98). Дані правила поширюються на всіх суб'єктів господарювання незалежно від форм власності, які у своїй діяльності здійснюють роботу, пов'язану з персональними комп'ютерами [74].

Площу приміщень, в яких розташовуються персональні комп'ютери визначають у відповідності до чинних нормативних документів. Згідно з Вимогами щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроям розмір одного робочого місця обладнаного персональним комп'ютером становить не менше 6м<sup>2</sup>. Робочі місця розташовані на відстані не

менше ніж 1м від стіни з вікном, а також 1,4 м від стіни, відстань між бічними поверхнями комп'ютера є не меншою за 1,2 м Суміжні робочі місця працівників, у зв'язку з потребою особливої концентрації уваги, розділені перегородками висотою 1,5 – 2 м. [75].

Робочі місця заборонено створювати у підвальних або цокольних приміщення будинків. У приміщеннях, де працюють з комп'ютерами, щодня здійснюється вологе прибирання, з метою запобігання запиленості підлоги та меблів. Покриття підлоги є матовим з антистатичними властивостями, а сама поверхня рівною та неслизькою.

Правильна кольорова гама також забезпечує підвищення продуктивності та психологічному комфорту. Найбільш сприятливими для нервової системи є світлі та пастельні тони – зеленувато-блакитний, ясно-сірий, золотавий. У той час як яскраві, контрастні поєднання (синій і жовтогарячий, червоний і фіолетовий) будуть викликати втому та роздратування у працівників.

За потреби та з урахуванням площі приміщення в ньому можуть бути встановлені додатково шафи, полиці, тумби чи інші системами для зберігання документів або особистих речей співробітників.

Конструкція робочого місця користувача комп'ютером повинна забезпечити йому підтримку оптимальної пози для роботи та відповідати сучасним вимогам до ергономіка.

Стіл, за яким працює спеціаліст забезпечує оптимальне розміщення всього необхідного обладнання та документів на робочій поверхні. Для цього він має висоту поверхні 680 – 800 мм, ширину 600 – 1400 мм і глибину 800 – 1000 мм. Такі параметри забезпечують можливість виконання операцій в зоні досяжності працівника.

Робочий стілець працівника відповідно до п. 4.8 ДСанПіН 3.3.2.007-98 є підйомно-поворотним, з можливістю регулювання висоти та куту нахилу сидіння та спинки. Регулювання параметрів стільця здійснюватися легко та надійно фіксується. Поверхня для сидіння та спинка крісла вироблені з

напівм'якого, нековзкого та повітронепроникного покриття, що легко чиститься та не електризується. Поверхня для сидіння є плоскою з заокругленим краєм [76].

Для того, щоб знизити навантаження на органи зору між екраном комп'ютера та користувачем зберігається оптимальна відстань, що становить 600 – 700 мм, але не менше 600 мм з урахуванням літерно – цифрових знаків і символів. Також контролюється, щоб персональний комп'ютер та його комплектуючі не потрапляли під дію джерел тепла (батареї опалення, тощо) або під прямі сонячні промені. На рисунку 5.1 зображено приклад рекомендованого оформлення робочого місця.

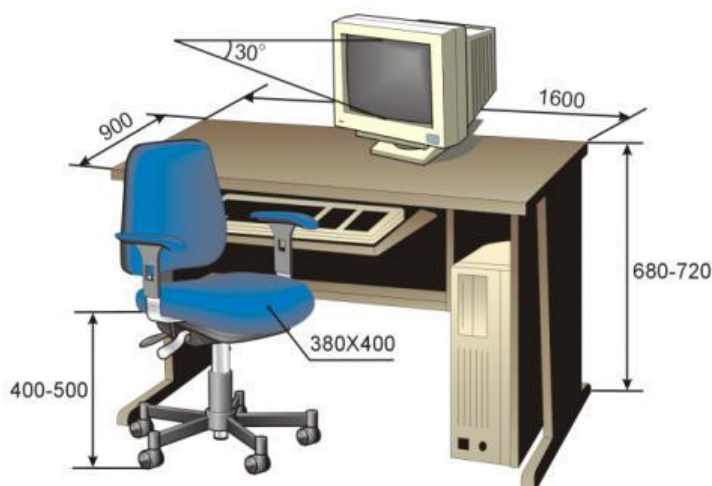


Рисунок 5.1 – Рекомендовані параметри для робочого місця

Тривала робота за комп'ютером може негативно вплинути на організм людини. Тому щоб мінімізувати ризики важливо мати правильно спроектоване робоче місце та виконувати ряд рекомендацій, що стосуються режиму праці та відпочинку. Саме тому працівникам рекомендується кожну годину робити перерву мінімум на 15 хвилин, під час якої виходити на свіже повітря, виконувати не складні гімнастичні вправи. Під час же самої роботи кожні 10–15 хвилин намагатись змінювати позу, рухати плечами, ногами та руками [74].

Всі приміщення обладнанні системами опалення та кондиціонування повітря, або припливно-витяжною вентиляцією, відповідно до норм та правил ДБН В.2.5-67:2013 «Опалення, вентиляція та кондиціонування». У приміщеннях забезпечується контроль параметрів мікроклімату та виконується його нормування, відповідно до встановлених санітарно-гігієнічних вимог державних санітарних норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень [77]. В усіх на робочих місцях працівників повинні забезпечуються оптимальні параметри мікроклімату:

- температура повітря в теплий період року становить 22–24°C, відносна вологість повітря — 40–60%, швидкість руху повітря — не вище за 0,2 м/с;
- температура повітря в холодний період року становить – 21–23°C, відносна вологість повітря — 40–60%, швидкість руху повітря — не більше 0,1 м/с.

Виконується регулярне очищення повітря, шляхом видалення забрудненого, нагрітого, іонізованого повітря з приміщення. Оскільки здатність обчислювальної техніки виділяти суттєву кількість тепла в середовище, призводить до підвищення температури та до зниження рівня вологості в приміщенні. Щоб мати змогу підтримувати необхідний об'єм чистого повітря в робочому просторі передбачено кондиціонування повітря. При недотриманні вказаних показників мікроклімату в офісних приміщеннях робочий день для робітників скорочується мінімум на 10% [78].

Для комфортної роботи співробітників кожне приміщення забезпечено елементами природного та штучного освітлення, відповідно до ДБН В. 2.5-28-2018 «Інженерне обладнання будинків і споруд. Природне і штучне освітлення» [79], оскільки тривала робота за комп'ютером чи документами при поганому рівні освітленості може призвести до перенапруження органів зору

Зазвичай використовують природне бокове одностороннє освітлення в день, а у ввечері – загальне рівномірне штучне освітлення. Робоче місце розміщене так, щоб природне світло було з лівого боку) та уникати потрапляння прямого світла в очі. Також у працівників є можливість для корегування рівня природнього освітлення за допомогою жалюзі або штор. Оскільки яскраве світло в районі периферійного зору може збільшити навантаження на органи зору та призвести до швидкої втоми, освітлення є рівномірним, тобто ступінь освітленості приміщення та яскравість монітору комп'ютера повинні бути приблизно однаковими.

Коефіцієнт пульсації не перевищує 5 %. Рівень освітлення робочого столу в зоні розташування документів знаходиться в межах 300 – 500 лк. Освітлювальні прилади приміщення не повинні створювати відблисків на поверхні екрана, а освітлення екрана має не перевищувати 300 лк.

Підвищення рівню шуму погіршує умови праці та негативно впливає на організм людини. В результаті у робітників спостерігаються підвищений рівень стомленості, роздратованість, сильні головні болі, зниження концентрації уваги, нервово-психічне навантаження. Тривала дія інтенсивного шуму (вище 80 дБА) на слух людини призводить до його часткової або повної втрати [72].

Основними джерелами шуму на робочих місцях в офісах є принтери, кондиціонери, системи вентиляції та комп'ютери, в яких джерелами шуму є вентилятори, що охолоджують систему. Рівень шуму та вібрацій на робочих місцях осіб, що працюють за персональним комп'ютером, визначають відповідно до ДСН 3.3.6.037-99. Санитарные нормы производственного шума, ультразвука и инфразвука. Для забезпечення дотримання допустимих рівнів шуму на робочих місцях застосовуються засоби звукопоглинання, вибір яких обґрунтовується спеціальними інженерно-акустичними розрахунками [80].

Також в офісних приміщеннях дотримуються вимог, що прописані у ДСанПіН 3.3.2.007-98 та ДСанПіН 3.3.6.096-2002 щодо рівня неіонізуючих електромагнітних випромінювань, електростатичних і магнітних полів, а також

підтримують в нормі рівень інтенсивності потоків інфрачервоного та ультрафіолетового випромінювання [82].

Розрахунок освітленості приміщення. Приміщення знаходиться на четвертому поверсі п'ятиповерхового цегляного будинку, що відповідає нормам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

Розміри приміщення становлять: довжина  $A = 9$  м, ширина  $B = 6$  м, висота  $H = 3,3$  м. Площа дорівнює  $S = 54$  м<sup>2</sup>, а об'єм  $V = 178,2$  м<sup>3</sup>.

В приміщенні наявне як природне так і штучне освітлення, що відповідає нормам ДБН В.2.5-28:2018 «Природне і штучне освітлення». Природне освітлення використовується в день, а у ввечері використовувати загальне рівномірне штучне освітлення.

Робоче місце розміщене таким чином, що природне світло знаходиться з лівого боку (п. 4.3 ДСанПіН 3.3.2.007-98), що дозволяє уникнути потрапляння прямого світла в очі. Джерелом природнього освітлення в приміщенні є вікна, на яких для корегування рівня природнього світла наявні жалюзі. Світильники загального і місцевого освітлення дозволяють створюють нормальні умови освітленості і відповідний контраст між екраном і навколишнім оточенням з урахуванням виду роботи та вимог видимості з боку працівника.

Щоб перевірити чи відповідає рівень освітленості в приміщенні нормативним вимогам зазначених у ДБН В.2.5-28:2018 необхідно виконати розрахунок оцінки штучного освітлення в приміщенні. Для цього використовується метод коефіцієнта використання світлового потоку, що використовують при розрахунку загального рівномірного освітлення горизонтальних поверхонь з урахуванням відбитих від стін, підлоги і стелі світлових потоків.

Згідно з ДБН В.2.5-28:2018 «Природне і штучне освітлення» освітлення в приміщенні нормується як  $E_n = 300$  Лк.

Висота робочої поверхні  $h_p = 0.8$  м;

Коефіцієнти відбиття поверхонь приміщення (від: стелі –  $\rho_c$ ; стін –  $\rho_{ст}$ ; підлоги –  $\rho_{п}$ ) у світлому приміщенні  $\rho_c=70\%$ ;  $\rho_{ст}=50\%$ ;  $\rho_{п}=30\%$ ;

Обираємо світильник типу ЛВП (IP=54 – ступінь захисту світильника,  $[L/h] = 1,3$  – числове значення коефіцієнта світильника).

Коефіцієнт запасу  $k_z$  враховує зниження рівня освітленості з часом в результаті забруднення та старіння ламп, світильників і поверхонь приміщення, приймається в залежності від виробничих умов. Приймаємо  $k_z = 1,4$ .

Коефіцієнт нерівномірності (мінімального) освітлення  $z$  (відношення середньої освітленості до мінімальної освітленості), для люмінесцентних ламп низького тиску, як правило дорівнює  $z = 1,1$ .

Розрахуємо кількість рядів світильників у приміщенні:

$$N_p = \frac{B}{(H - h_p) * [L/h]} = \frac{6}{(3,3 - 0,8) * 1,3} = 1,84 \approx 2$$

Визначимо максимально припустиму відстань між рядами світильників

$$L \frac{B}{N_{p \max}},$$

де  $B$  – ширина приміщення,  $N_p$  – кількість рядів

$$L_{\max} = \frac{6}{2} = 3 \text{ м}$$

Розраховуємо висоту підвісу світильника над робочою поверхнею

$$h = \frac{L_{\max}}{[L/h]}$$

$$h = \frac{3}{1,3} = 2,3 \text{ м}$$

Знайдемо висоту звисання світильника від стелі

$$h_3 = H - h_p - h$$

де  $H$  – висота приміщення,  $h_p$  – висота робочої поверхні,  $h$  – висота підвісу світильника над робочою поверхнею.

$$h_3 = 3,3 - 0,8 - 2,3 = 0,2 \text{ м}$$

Чисельне значення індексу приміщення визначаємо за рівнянням:

$$i = \frac{AB}{h(A+B)}$$

де  $A$  – довжина приміщення, м;

$B$  – ширина приміщення, м;

$h$  – висота розміщення світильників над робочою поверхнею, м

$$i = \frac{9 \cdot 6}{2,3 \cdot (9+6)} = \frac{54}{34,5} = 1,57$$

Значення коефіцієнта використання світлового потоку  $\eta$  вибирається в залежності від виду джерела світла, типу обраного світильника, коефіцієнтів відбиття поверхонь приміщення ( $\rho_c$ ,  $\rho_{ст}$ ,  $\rho_{п}$ ) та індексу приміщення. В нашому випадку  $\eta = 0,5$ .

Визначаємо сумарний світловий потік освітлювальної установки у даному виробничому приміщенні:

$$\Phi_{\Sigma} = \frac{E_n \cdot S \cdot k_z \cdot z}{\eta},$$

де  $\Phi_{\Sigma}$  – розрахункове значення сумарного світлового потоку у приміщенні, лм;

$E_n$  – нормоване значення освітленості, лк;

$S$  – площа освітлюваної поверхні, м<sup>2</sup>;

$k_z$  – коефіцієнт запасу;

$z$  – коефіцієнт нерівномірності (мінімальної) освітленості;

$\eta$  – коефіцієнт використання світлового потоку

$$\Phi_{\Sigma} = \frac{300 \cdot 9 \cdot 6 \cdot 1,4 \cdot 1,1}{0,5} = \frac{24948}{0,5} = 49896 \text{ лм}$$

Визначаємо умовну загальну кількість світильників у приміщенні:

$$N^* = AB/L_{\max}^2$$

$$N^* = \frac{9 \cdot 6}{9} = 6 \text{ шт}$$

Розрахуємо світловий потік умовного джерела світла:

$$\Phi_{\text{л}}^* = \frac{\Phi_{\Sigma}}{N_{\text{л}}}, \text{ лм}$$

де  $\Phi_{\Sigma}$  – розрахункове значення сумарного світлового потоку у приміщенні, лм;

$N_{\text{л}}$  – загальна кількість ламп у приміщенні, шт;

$$N_{\text{л}} = N^* \cdot n$$

$n$  – кількість ламп у світильнику (в ЛВП світильнику знаходиться 2 лампи)

$$N_{\text{л}} = 6 \cdot 2 = 12 \text{ шт.}$$

Тоді,

$$\Phi_{\text{л}}^* = \frac{49896}{12} = 4158 \text{ лм}$$

Обираємо тип стандартної лампи з найближчим значенням фактичного світлового потоку лампи  $\Phi_{\text{л}}$  і знайти коефіцієнт  $m$  (співвідношення між розрахунковим світловим потоком лампи  $\Phi_{\text{л}}^*$  та фактичним світловим потоком вибраної стандартної лампи  $\Phi_{\text{л}}$ ):

Обираємо лампу ЛБ 58,  $\Phi_{\text{л}} = 4300 \text{ лм}$

$$m = \frac{\Phi_{\text{л}}^*}{\Phi_{\text{л}}}$$

$$m = \frac{4158}{4300} = 0,96$$

Визначаємо оптимальну (фактичну) кількість світильників у приміщенні:

$$N = N^* \cdot m$$

$$N = 6 \cdot 0,96 = 12 \text{ шт.}$$

Фактична кількість ламп у приміщенні:

$$N_{\text{фл}} = N_{\text{ф}} \cdot n$$

$$N_{\text{фл}} = 12 \cdot 2 \text{ шт.}$$

Визначаємо загальну розрахункову освітленість  $E_p$  у приміщенні, що створюється при застосуванні стандартних ламп:

$$E_p = \frac{\Phi_{\text{л}} N_{\text{фл}} \eta}{S k_{\Sigma} \varepsilon}, \text{ лк}$$

$$E_p = \frac{4300 \cdot 12 \cdot 0,5}{9 \cdot 6 \cdot 1,4 \cdot 1,1} = \frac{25800}{83,16} = 310,2 \text{ лк}$$

Тобто виконується умова  $E_p = (-10\% \dots +20\%) \cdot E_n$ , лк

Отриманий результат порівнюємо з ДБН В.2.5-28:2018 й робимо висновок, що рівень штучного освітлення у приміщенні є достатнім.

## 5.4 Заходи безпеки у надзвичайних ситуаціях

### 5.4.1 Заходи з пожежної безпеки

Заходи з протипожежної безпеки розроблені для виробничого приміщення обладнаного ПК з ВДТ розроблені згідно вимог НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні» [82].

Виходячи з аналізу речовин та матеріалів, які використовуються при роботі у приміщенні, обладнаному ПК з ВДТ:

– згідно ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» [83] у приміщенні, обладнаному ПК з ВДТ можлива пожежа класу – А та Е; воно належить до категорії «Д» з пожежної небезпеки – простір у приміщенні, у якому перебувають тверді горючі речовини та матеріали.

Оскільки приміщення належить до категорії «Д» з пожежної небезпеки, тому відповідно до вимог ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» [83] воно має II ступінь вогнестійкості.

У разі виникнення пожежі у приміщенні обладнаному ПК з ВДТ для евакуації персоналу відповідно до вимог ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» передбачені виходи, по обидві сторони приміщення, з одного боку вікно (на пожежні сходи), а з іншого – вхідні двері. Згідно СНиП 2.09.02-85\* «Производственные здания», відстань від найбільш віддаленого робочого місця до найближчого евакуаційного виходу не обмежується.

Обладнання, силові та освітленні мережі приміщення, обладнаного ПК з ВДТ відповідають вимогам пожежної безпеки, оскільки виконані відповідно до вимог НПАОП 40.1-1.32-01 «Правила будови електроустановок. Електрообладнання спеціальних установок» [84], та мають ступінь захисту ізоляції обладнання IP44 яка відповідає класу пожежанебезпечної зони П-Па до якої належить приміщення.

Для виявлення початкової стадії загоряння, швидкого й точного оповіщення служби пожежної охорони про час і місце виникнення пожежі встановлені системи автоматичної пожежної сигналізації (АПС). Вони можуть самостійно пускати в хід установки пожежогасіння, коли пожежа не досягла великих розмірів. Системи АПС складаються з пожежних сповіщувачів, ліній зв'язку й прийомних пультів (станцій). Згідно вимог ДБН В.2.5-56:2014 «Системи протипожежного захисту», в приміщенні обладнаному ПК з ВДТ встановлена система пожежної й охоронної сигналізації «Фотон-ОФІС», яка забезпечує виявлення теплових і димових ознак пожежі і місця виникнення пожежі з точністю до місця розміщення датчика [72].

Оскільки приміщення що обладнане ПК з ВДТ має площу 54 м<sup>2</sup>, тому відповідно до вимог п. 3.8 розділу «Типові норми належності вогнегасників» ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги» [85] для гасіння електроустановок, що знаходяться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 3 штуки (з розрахунку один вогнегасник с величиною заряду вогнегасної речовини 3 кг. і більше, на 20 м<sup>2</sup> площі приміщення). Відстань між вогнегасниками та місцями можливих загорянь не перевищує 10 м.

#### **5.4.2 Заходи з цивільного захисту**

Укриття населення у захисних спорудах цивільного захисту

Укриття населення в захисних спорудах є комплексом заходів із їх

завчасним будівництвом, обладнанням приміщень та підтримкою їх у готовності до використання. Вони здатні захистити населення від наслідків дії ядерної, хімічної, бактеріологічної та звичайної зброї, від аварій та катастроф, деяких стихійних лих, військових дій або терористичних актів.

За своїм призначенням та захисними властивостями захисні споруди поділяються на сховища, протирадіаційні укриття та найпростіші укриття.

Сховища є надійним захистом від небезпечних чинників, що виникають в результаті надзвичайних ситуацій. Вони представляють собою комплекс інженерних споруд, що забезпечують необхідні умови життєдіяльності протягом певного часу.

За захисними властивостями від дії вибухової сили сховища поділяються на п'ять класів. За часом побудови на: побудовані завчасно і швидко споруджені. За наявністю фільтровентиляційного обладнання бувають промислового виготовлення та спрощеного (виготовлені з підручних матеріалів). За місцем заходження: вбудовані, ті що знаходяться в підвальних приміщеннях будинків, та відокремлені, які знаходяться поза будинками. Зазвичай їх будують заздалегідь та підтримують у належному стані, однак вони також можуть будуватись і в період загрози нападу або під час воєнних дій (швидко зведені). Сховища також класифікують за місткістю: малі здатні вміщувати 150-300 чоловік, середні – 300-600 чол. та великі – понад 600 чол.

Сховища мають забезпечувати надійний захист людей від усіх уражаючих факторів та їх безперервне перебування в них не менше двох діб. Вони розташовуються на місцевості, що знаходиться на відстані від ліній водостоку та каналізації, щоб уникнути їх затоплення. Всі входи та виходи мають бути з тим же ступенем захисту, що й основні приміщення, а на випадок їх завалу, повинні бути передбачені аварійні виходи. Висота основних приміщень укриття не менше 2,2 м, а рівень підлоги розташований вище ґрунтових води не менш як на 20 см.

Протирадіаційними укриття дозволяють захистити людей від

опромінення та впливу радіоактивних речовин в зонах радіаційного забруднення, від біологічних засобів у краплинно-рідинному вигляді та світлового випромінення ядерного вибуху. Вони можуть створюватись у підвальних поверхах будинків та споруд.

При їх виборі та підготовці обов'язково враховують захисні властивості будівельних матеріалів та окремих конструкцій, оскільки певна щільність та товщина шару матеріалу здатні ослаблювати та зменшити потік радіоактивних речовин. Що дозволяє людині, що знаходиться в приміщенні отримати меншу дозу опромінення. Так підвали в дерев'яних одноповерхових будинках здатні ослабити дозу радіації в 7 разів, а в житлових одноповерхових кам'яних (цеглиних) будинках — у 40, у двоповерхових — у 100, середня частина підвалу кількоповерхового кам'яного будинку — у 800—1000 разів. У випадку коли рівень радіації не високий для захисту від бактеріальних засобів, парів отруйних і сильнодіючих ядучих речовин можна використовувати кам'яні (цегляні) або дерев'яні будівлі.

Сховища та протирадіаційні укриття складаються з основних та допоміжних приміщень. Основними є приміщення для розміщення людей, пункти управління, медичні пункти, а до допоміжних відносяться приміщення, де розташовані фільтровентиляційне обладнання, системами водопостачання, опалення, санітарні вузли, приміщення для зберігання продовольства, тамбур-шлюзи (тамбури).

Найпростіші укриття для захисту або щілини є захисними спорудами, які можуть бути створеними населенням у найкоротший проміжок часу. Такі укриття здатні певною мірою захистити від ударної хвилі, радіаційного ураження та світлового випромінювання. Для будівництва щілин вибирають сухі місця, що знаходяться на підвищенні.

Найпростіші укриття типу щілини бувають двох видів відкритими та перекритими. Відкриті щілини дозволяють зменшити в 1,2–2 рази ймовірність ураження ударною хвилею, проникаючою радіацією та світловим

випромінюванням. Перекриті щілини є більш надійним захистом населення, оскільки вони здатні повністю захистити від світлового випромінювання, від ударної хвилі – у 1,5–3 рази, від проникаючої радіації і радіоактивного випромінювання – у 200-300 разів, а також надійно захищає від осколкових і кулькових бомб, від запалювальних засобів.

Будівництво укриття починається з розмітки та визначення її плану на місцевості. Щілини роблять у вигляді прямолінійних ділянок з такими розмірами: довжина до 15 м, глибина 1,8–2,0 м; ширина: зверху – 1,1-1,3 м, а на дні – 0,8 м. Стіни укриття укріплюють дошками, хмизом, жердинами або іншим підручним матеріалом. Перекриття щілини роблять з накату колод, за яким йде шар глини товщиною 10—15 см, що дозволяє захистити укриття від потрапляння дощових вод. Потім шар ґрунту 20—40 см, а зверху все вкривають дерном. Також такі укриття можна будувати зі збірних залізобетонних конструкцій. Входи до щілини роблять східчасті, під прямим кутом до осі укриття. У перекритій щілині вхід може бути з однієї або двох сторін з дверима та тамбуром. В нормі найпростіші укриття типу щілина повинна забезпечити захист від 10 до 15 чоловік. [86].

## ВИСНОВКИ

Як показали проведені дослідження, сучасні біоінженери активно працюють у напрямку створення кіберпротезів, які вже сьогодні використовуються в якості функціональних замінників втрачених кінцівок людини. Протези нового покоління покликані не лише відновлювати природний зовнішній вигляд пацієнта, але й заповнювати найважливіші втрачені функції. Тим не менш, залишається багато відкритих питань щодо удосконалення їх функцій, масо-габаритних характеристик, зовнішнього вигляду та алгоритмів управління. Застосування технологій прототипування сприяє економії часу та коштів на розробку, оскільки дозволяє виявляти небажані помилки та оптимізувати виріб на ранніх етапах реалізації проєкту.

В магістерській роботі досліджено особливості застосування технологій прототипування при розробці кіберпротезу верхньої кінцівки людини.

Для досягнення мети роботи розроблено віртуальний та фізичний прототипи кіберпротезу, а також програмне забезпечення для його функціонування на основі базових захоплень.

В ході розробки механічної частини прототипу було використано середовище проєктування PTC Creo та 3D-принтер Leapfrog Creatr HS. В ході розробки електронної частини було проведено аналіз та вибір всіх необхідних компонентів, що здійснювався за функціональними показниками та за розмірами. Було проведено інтеграцію механічної та електронної частин прототипу, а також розробку програмного забезпечення для функціонування прототипу.

Наукове значення роботи полягає в тому, що отримала подальший розвиток технологія прототипування на основі запропонованого методу комплексного віртуального та фізичного прототипування, а також інформаційної технології розробки механічної частини прототипу протезу, що відрізняється застосуванням параметричного моделювання.

Практичне значення роботи полягає в тому, що застосування запропонованого методу та технології дозволить підвищити ефективність процесу розробки за рахунок зменшення часу проєктування та кількості помилок, а також надасть можливості покращення форми та масогабаритних показників проєктованих протезів.

В подальшому, планується додати сенсори для контролю сили тиску частин протезу на об'єкт при захопленні, а також розширити перелік досяжних захоплень для удосконалення алгоритмів управління.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Belter J. T. Mechanical design and performance specifications of anthropomorphic prosthetic hands: a review / J. T. Belter, J. L. Segil, A. M. Dollar, R. F. Weir // *Journal of Rehabilitation Research & Development*. – 2013. – Vol. 50, Issue 5. – P. 599–617.
2. Pons J. L. The MANUS-HAND dextrous robotics upper limb prosthesis: mechanical and manipulation aspects / J. L. Pons, E. Rocon, R. Ceres, D. Reynaerts, B. Saro, S. Levin, W. Van Moorleghe // *Autonomous Robots*. – 2004. – Vol. 16.–P. 143–163.
3. Light, C. M. Development of a lightweight and adaptable multiple-axis hand prosthesis / C. M. Light, P. H. Chappell // *Medical Engineering & Physics*. – 2000. – Vol. 22 , Issue 10. – P. 679-684.
4. Zheng, J. Z. An investigation of grasp type and frequency in daily household and machine shop tasks / J. Z. Zheng, S. De La Rosa, A. M. Dollar // *Robotics and automation: International conference, Shanghai, China, 9-13 May 2011: proceedings*. – Los Alamitos: IEEE, 2011. – P. 4169–4175.
5. Parkhomenko, A. Investigation and realization of prototyping technologies for robotic-prostheses computer aided design / A. Parkhomenko, O. Gladkova, Ya. Zalyubovskiy // *The Experience of designing and application of CAD systems in microelectronics: International conference, Lviv, February 2019: proceedings*. – Los Alamitos: IEEE, 2019 – P. 7/5-7/8.
6. Liou, W. F. Rapid prototyping and engineering applications: a toolbox for prototype development / W. F. Liou. – CRC Press Taylor & Francis Group Boca Raton, 2007. – 592 p.
7. McEwen, A. Designing the Internet of things / A. McEwen, Cassimally. – Wiley, 2014. – 338 p.
8. Киберфизические системы [Электрон. ресурс]. Режим доступа: [http://www.tadviser.ru/index.php/Статья:Киберфизические\\_системы](http://www.tadviser.ru/index.php/Статья:Киберфизические_системы)

9. Cyber-Physical systems [Electronic resource]. Access mode: [https://link.springer.com/referenceworkentry/10.1007%2F978-3-642-35950-7\\_16790-1](https://link.springer.com/referenceworkentry/10.1007%2F978-3-642-35950-7_16790-1).
10. Cyber-Physical systems [Electronic resource]. Access mode: <https://ptolemy.berkeley.edu/projects/cps/>.
11. Кибер-физические системы в современном мире [Электрон. ресурс]. Режим доступа: <https://habr.com/ru/company/toshibarus/blog/438262/>.
12. Как работает киберпротез? [Электрон. ресурс]. Режим доступа: <https://postnauka.ru/faq/63738>
13. Биомехатроника: киберпротезы, дающие человеку сверхспособности. [Электрон. ресурс]. Режим доступа: <https://itc.ua/articles/biomehatronika-kiberprotezyi-dayushhie-cheloveku-sverhspособnosti/>.
14. Особливості біонічних протезів верхніх кінцівок [Електрон. ресурс]. Режим доступу: <http://esnuir.eenu.edu.ua/bitstream/123456789/13204/1/Yuriy%20Popadiukha.pdf>
15. Абу Ханиех, Я. Я. Адаптивный протез руки: съём сигналов управления / Я. Я. Абу Ханиех., Г. Н. Буров // Биомедицинская радиоэлектроника. – 2010. – № 4. – С. 20-24.
16. Воротников, С.А. Биометрическая система управления протезом руки / С. А. Воротников, В. С. Струнин, Н. А. Выборнов. // Прикаспийский журнал: управление и высокие технологии. – 2013. – № 3 (23). – С. 147–162.
17. Вонсевич, К. П. Особенности бионического протезирования верхних конечностей / К. П. Вонсевич, М. А. Безуглый // Приборостроение – 2016: 9-я Международная научно-техническая конференция, Минск, Республика Беларусь, 23-25 ноября 2016 г.: тезисы докладов. – Минск: БНТУ, 2016. – С. 36-37.
18. Прототипирование [Электрон. ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/Прототипирование>.

19. Юзабіліті-тестування [Електрон. ресурс]. Режим доступу: <https://uk.wikipedia.org/wiki/Юзабіліті-тестування>.
20. Варфел, Т. Прототипирование. Практическое руководство / Т. Варфел. – М.: Манн, Иванов и Фербер, 2013. – 240 с.
21. Бойцов, Б. В. Повышение качества подготовки производства применением технологий быстрого прототипирования // Б. В. Бойцов; М. Ю. Куприков, Ю. В. Маслов //Электронный журнал «Труды МАИ». – 2011. – № 49. – С.1 – 14.
22. Tinkercad Arduino – лучший онлайн симулятор ардуино на русском [Электрон. ресурс]. Режим доступа: <https://arduino-master.ru/program/simulyator-arduino-tinkercad-circuits/>.
23. Описание для ардуино виртуальный симулятор на русском. Tinkercad Dashboard – начальная страница [Электрон. ресурс]. Режим доступа: <https://sherstinka.ru/rezyume/opisanie-dlya-arduino-virtualnyi-simulyator-na-russkom-tinkercad/>.
24. Proteus. Интегрированная среда для разработки электронных устройств, в т.ч. на микроконтроллерах [Электрон. ресурс]. Режим доступа: <http://www.gaw.ru/html/cgi/txt/soft/avr/Proteus.htm>.
25. Proteus [Электрон. ресурс]. Режим доступа: <https://proteus.soft112.com/>.
26. Atmel Studio [Электрон. ресурс]. Режим доступа: [https://cxem.net/software/atmel\\_studio.php](https://cxem.net/software/atmel_studio.php).
27. Atmel Studio 7 [Электрон. ресурс]. Режим доступа: <https://www.microchip.com/mplab/avr-support/atmel-studio-7>.
28. Технологии быстрого прототипирования [Электрон. ресурс]. Режим доступа: [https://web.archive.org/web/20131017072744/http://www.vzrt.ru/rp\\_tec.php](https://web.archive.org/web/20131017072744/http://www.vzrt.ru/rp_tec.php)
29. Обзор популярных систем автоматизированного проектирования (CAD) [Электрон. ресурс]. Режим доступа: <https://www.pointcad.ru/novosti/obzor-sistem-avtomatizirovannogo-proektirovaniya>.

30. Autodesk within medical releases software for generative 3D modeling. [Electronic resource]. Access mode: <https://3dmedicalconference.com/uncategorized/autodesk-within-medical-releases-software-for-generative-3d-modeling-video/>.

31. Solid Edge для медицинских изделий. ПО для проектирования. [Электрон. ресурс]. Режим доступа: <https://solidedge.siemens.com/ru/industries/medical-devices/>.

32. Creo Parametric – Фундаментальное 3D-параметрическое CAD-решение [Электрон. ресурс]. Режим доступа: <http://ptsrussia.com/products/creo/blog-statej-creo/item/37-creo-parametric-fundamental.html>.

33. 3D-сканер: 3D-сканирование объектов и трехмерное моделирование [Электрон. ресурс]. Режим доступа: <https://koloro.ua/3d-skaner-3d-skanirovanie-obektov-i-trehmernoje-modelirovanie.html>.

34. Что такое 3D сканер и как он работает? [Электрон. ресурс]. Режим доступа: <https://make-3d.ru/articles/chto-takoe-3d-skaner-i-kak-on-rabotaet/>.

35. Принцип работы 3D сканера [Электронный ресурс]. Режим доступа до ресурсу: <https://setafi.com/elektronika/skaner/printsip-raboty-3d-skanera/>.

36. Как работает устройство 3d сканеров? Технологии и принципы сканирования [Электрон. ресурс]. Режим доступа: <https://robot-ik.ru/obzory/kak-rabotaet-ustroystvo-3d-skanerov-tehnologii-i-printsipy-skanirovaniya/>.

37. Как используются 3D сканеры здесь и сейчас [Электрон. ресурс]. Режим доступа: <https://make-3d.ru/articles/kak-ispolzuyutsya-3d-skanery-zdes-i-sejchas/>.

38. 3D сканер Gotcha [Электрон. ресурс]. Режим доступа: <https://shop.zanimatica.ru/3d-scanners/personalnye-skanery/3d-skaner-gotcha>.

39. 3d сканер Gotcha от 4ddynamics [Электрон. ресурс]. Режим доступа: <https://3dcorp.ru/gotcha.html>.

40. Технологии быстрого прототипирования. Описание и назначение [Электрон. ресурс]. Режим доступа: <https://studfiles.net/preview/2859272/page:3/>.

41. Быстрое прототипирование [Электрон. ресурс]. Режим доступа: [https://ru.wikipedia.org/wiki/Быстрое\\_прототипирование](https://ru.wikipedia.org/wiki/Быстрое_прототипирование).
42. Как работает 3D принтер и что такое 3D печать [Электрон. ресурс]. Режим доступа: <http://kakustroen.ru/tehnika/kak-rabotaet-3d-printer-i-chto-takoe-3d-pechat>.
43. Что такое 3D печать и 3D принтер [Электрон. ресурс]. Режим доступа: <https://make-3d.ru/articles/chto-takoe-3d-pechat/>.
44. Как работает 3D-принтер? [Электрон. ресурс]. Режим доступа: <https://ichip.ru/kak-rabotaet-3d-printer-prosto-o-slozhnom.html>
45. Сферы применения 3D печати [Электрон. ресурс]. Режим доступа: <https://3dsmart.com.ua/blog/sfery-primeneniya-3d-pechaty>.
46. 3D технологии: сканеры и принтеры [Электрон. ресурс]. Режим доступа: <https://media-tellur.ru/oborudovanie/3d-technologii-scanery-i-printery>.
47. 3D принтер Leapfrog Creatr HS [Электрон. ресурс]. Режим доступа: <https://3dtool.ru/product/3d-printerleapfrog-creatr-2-html/>.
48. 3D принтер Leapfrog Creatr HS [Электрон. ресурс]. Режим доступа: <https://3dsmart.com.ua/shop/3d-printers/3d-printer-leapfrog-creatr-hs>.
49. 3D принтер LeapFrog Creatr HS [Электрон. ресурс]. Режим доступа: <https://3ddevice.com.ua/product/leapfrog-creatr-hs-3d-printer/>.
50. Gaget L 2018 Top 8 of the best parametric modeling software in 2020 [Electronic resource]. Access mode: <https://www.sculpteo.com/blog/2018/03/07/top-8-of-the-best-parametric-modeling-software/>
51. Simplify3D [Electronic resource]. Access mode: <https://www.simplify3d.com/>
52. Arduino [Электрон. ресурс]. Режим доступа: <https://all-arduino.ru/>.
53. Сравнение плат Arduino (таблица) [Электрон. ресурс]. Режим доступа: <http://digitrode.ru/arduino-comparison.html#sel=3:17,4:22>.
54. Сравнение плат Arduino. Какую выбрать? [Электрон. ресурс]. Режим доступа: [https://ampermarket.kz/base/arduino\\_family/](https://ampermarket.kz/base/arduino_family/).

55. Все платы Ардуино: сравнительные таблицы [Электрон. ресурс]. Режим доступа: <https://arduinoplus.ru/arduino-vse-plati-sravnitelnaya-tablica/#i-9>.

56. Плата Arduino NANO v 3.0 : распиновка, схемы, драйвер [Электрон. ресурс]. Режим доступа: <https://arduinomaster.ru/platy-arduino/plata-arduino-NANO/>.

57. Arduino NANO [Электрон. ресурс]. Режим доступа: <https://all-arduino.ru/product/arduino-NANO/>.

58. Arduino NANO V3.0 — характеристики, распиновка, драйвера, описание платы [Электрон. ресурс]. Режим доступа: <https://micro-pi.ru/arduino-NANO-v3-0-обзор-драйвера/>.

59. Arduino NANO [Электрон. ресурс]. Режим доступа: <https://arduino.ua/prod166-arduino-NANO-v3-0-avr-atmega328-p-20au-s-kabelem-mini-usb-i-raspayannimi-razyomami>.

60. Arduino Shields – платы расширения для ардуино [Электрон. ресурс]. Режим доступа: [https://arduinomaster.ru/platy-arduino/shildy-i-platy-rasshireniya-arduino/#Arduino\\_Motor\\_Shield](https://arduinomaster.ru/platy-arduino/shildy-i-platy-rasshireniya-arduino/#Arduino_Motor_Shield).

61. Плата расширения NanoShield для Arduino Nanov3.0 [Электрон. ресурс]. Режим доступа: <https://freedelivery.com.ua/arduino-100/platy-rasshireniya-139/plata-rasshireniya-Nano-shield-dlya-arduino-Nano-v30-3580.html>.

62. Arduino NanoShield V3.0 (Плата расширения для Arduino Nano) [Электрон. ресурс]. Режим доступа: <https://ampermarket.kz/shields/arduino-Nano-shield-v3/>.

63. Плата расширения ввода/вывода Arduino NanoShield V3.0 [Электрон. ресурс]. Режим доступа: [http://robot-kit.ru/product\\_info.php/info/p1282\\_Plata-rasshireniya-Arduino-Nano-Shield-V3-0--Funduino-Nano-006-V3-0-.html](http://robot-kit.ru/product_info.php/info/p1282_Plata-rasshireniya-Arduino-Nano-Shield-V3-0--Funduino-Nano-006-V3-0-.html).

64. Сервопривод SG90 [Электрон. ресурс]. Режим доступа: [https://autohome.org.ua/market/moto\\_servo/servo-sg90-9g-detail](https://autohome.org.ua/market/moto_servo/servo-sg90-9g-detail).

65. Сервоприводы: подключение, управление, скетчи Ардуино [Электрон. ресурс]. – Режим доступа: <http://wiki.amperka.ru/робототехника:сервоприводы>.

66. Сервопривод Tower Pro 9g SG90 [Электрон. ресурс]. Режим доступа: [robotday.ru/2015/03/28/сервопривод-tower-pro-9g-sg90/](http://robotday.ru/2015/03/28/сервопривод-tower-pro-9g-sg90/).

67. Сервоприводы Ардуино SG90, MG995, MG996: схема подключения и управление [Электрон. ресурс]. Режим доступа: [https://arduinomaster.ru/motor-dvigatel-privod/servoprivody-arduino-sg90-mg995-shema-podklyuchenie-upravlenie/#\\_\\_SG90\\_MG995\\_MG996](https://arduinomaster.ru/motor-dvigatel-privod/servoprivody-arduino-sg90-mg995-shema-podklyuchenie-upravlenie/#__SG90_MG995_MG996).

68. Сервопривод MG996 [Электрон. ресурс]. Режим доступа: <http://arduino.org.ua/privody-motory/servoprivody/servoprivod-mg996/?sl=ru>.

69. Сервопривод Tower Pro SG90 micro servo 9g [Электрон. ресурс]. Режим доступа: <https://www.mini-tech.com.ua/servomotor-sg90>.

70. Сервоприводы: подключение, управление, скетчи Ардуино [Электрон. ресурс]. – Режим доступа: <http://wiki.amperka.ru/робототехника:сервоприводы>.

71. Правила улаштування електроустановок [Текст]. Міненерговугілля України, 2017. – 617с.

72. Атаманчук П. Охорона праці в галузі. Навчальний посібник / П. Атаманчук, В. Мендерецький, О. Панчук, Р. Билык. К: Центр навчальної літератури, 2017. – 322 с.

73. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом [Текст] : ДСТУ Б В.2.5-82:2016. – На заміну ДБН В.2.5-27-2006 ; чинний від 2017-04-01. – К. : ДП «УкрНДНЦ», 2016. – 109 с.

74. Вимоги охорони праці при виконанні робіт на персональному комп'ютері [Електронний ресурс]. Режим доступу до ресурсу: <https://selidovorada.gov.ua/novini/vimogi-okhoroni-pratsi-pri-vikonanni-robit-na-personalnomu-komp-yuteri>.

75. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроям [Електронний ресурс]. Режим доступу: <http://zakon.rada.gov.ua/laws/show/z0508-18>.

76. Мадиевский Ю.М. Курс лекций по «Эргономике». Ю.М. Мадиевский. – Под ред. д-ра. техн. наук., проф. Березуцкого В.В. – Х.: НТУ «ХПИ», 2015. – 294 с.

77. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. / ДСН 3.3.6.042-99 – Х.: Форт, 2000 – 24 с.

78. Охорона праці офісних працівників: організація та обладнання робочого місця [Електронний ресурс]. Режим доступу: <http://oppb.com.ua/news/ohorona-praci-ofisnyh-pracivnykiv-organizaciya-ta-obladnannya-robochogo-miscya>.

79. ДБН В.2.5-28:2018. Природне та штучне освітлення. Проблеми природного і штучного освітлення. / ДБН В.2.5-28:2018. – К.: Етіп, 2019. – 180с.

80. ДСН 3.3.6.037-99. Санитарные нормы производственного шума, ультразвука и инфразвука. / ДСН 3.3.6.037-99. – К.: МОЗ Украины. – 1999. – 98с.

81. Робота в офісі: основні санітарно-гігієнічні вимоги [Електронний ресурс]. Режим доступу: <https://te.dsp.gov.ua/roboata-v-ofisi-osnovni-sanitarno-gigiyenichni-vymogy/>.

82. Правила пожежної безпеки в Україні. 2015 / Харків : Форт, 2015 – 124 с.

83. ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги / ДБН В.1.1-7:2016 – К.: Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. – 39 с.

84. НПАОП 40.1-1.32-01 Правила будови електроустановок. Електрообладнання спеціальних установок / НПАОП 40.1-1.32-01 – Х.: Індустрія, 2010. – 118 с.

85. ДСТУ 4297:2004 Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги / ДСТУ 4297:2004 – Х.: Форт, 2004 – 48 с.

86. Васійчук В.О. Цивільний захист: Конспект лекцій для студентів всіх спеціальностей і форм навчання / В.О. Васійчук, В.Є. Гончарук, О.С. Дацько – Л: ВЛп, 2011. – 208 с.

## ДОДАТОК А

### А.1 Керування кіберпротезом.cc

```
#include "Servo.h"
class Finger
{
    Servo servo;
    byte pos;
    byte increment;
    int16_t update_interval;
    uint32_t last_update;
public:
    init(int16_t interval, byte pin);
    void Update();
};
const int16_t servo_interval = 10;
Finger  servo1,  servo2,  servo3,  servo4,  servo5,  servo6,
servo7;
void setup()
{
    Serial.begin(9600);
    servo1.init(servo_interval, 2);
    servo2.init(servo_interval, 3);
    servo3.init(servo_interval, 4);
    servo4.init(servo_interval, 5);
    servo5.init(servo_interval, 6);
    servo6.init(servo_interval, 7);

}
void loop()
{
    servo_update();
}
```

```

Finger::init(int16_t interval, byte pin)
{
    update_interval = interval;
    increment = 1;
    servo.attach(pin);
}

void Finger::Update()
{
    if ((millis() - last_update) > update_interval)
    {
        last_update = millis();
        pos += increment;
        servo.write(pos);
        Serial.println(pos);

        if ((pos >= 180) || (pos <= 0))
        {
            increment = -increment;
        }
    }
}

void servo_update()
{
    servo1.Update();
    servo2.Update();
    servo3.Update();
    servo4.Update();
    servo5.Update();
    servo6.Update();
}

```

## **A.2 Сценарий 1 – тестування прототипу кібер-протезу два пальці зігнуті**

```
#include "Servo.h"

Servo myservo, myservo2, myservo3, myservo4, myservo5;
int pos1 = 0;

void setup()
{
    Serial.begin(9600);
    myservo.attach(2);
    myservo2.attach(3);
    myservo3.attach(4);
    myservo4.attach(5);
    myservo5.attach(6);
}

void loop()
{
    for (pos1 = 0; pos1 <= 180; pos1 += 1) { // goes from 0
degrees to 180 degrees
        myservo.write(pos1);
        myservo2.write(pos1);
        delay(20);
    }
}
```

## **A 3 Сценарий 2 – тестування прототипу кіберпротезу - один палець зігнутий**

```
#include "Servo.h"

Servo myservo, myservo2, myservo3, myservo4, myservo5;
int pos1 = 0;

void setup()
```

```

{
    Serial.begin(9600);
    myservo.attach(2);
    myservo2.attach(3);
    myservo3.attach(4);
    myservo4.attach(5);
    myservo5.attach(6);
}

void loop()
{
    for (pos1 = 0; pos1 <= 180; pos1 += 1) { // goes from 0
degrees to 180 degrees
        myservo.write(pos1);
        delay(20);
    }
}

```

#### **А 4 Сценарий 3 – тестування прототипу кіберпротезу - всі пальці зігнуті**

```

#include "Servo.h"
Servo myservo, myservo2, myservo3, myservo4, myservo5;
int pos1 = 0;

void setup()
{
    Serial.begin(9600);
    myservo.attach(2);
    myservo2.attach(3);
    myservo3.attach(4);
    myservo4.attach(5);
    myservo5.attach(6);
}

```

```

void loop()
{
    for (pos1 = 0; pos1 <= 180; pos1 += 1) { // goes from 0
degrees to 180 degrees
        myservo.write(pos1);
        myservo2.write(pos1);
        myservo3.write(pos1);
        myservo4.write(pos1);
        myservo5.write(pos1);
        delay(20);
    }
}

```

#### **A.5 Сценарий 4 – тестування протипу кіберпротезу - всі пальці розігнуті**

```

#include "Servo.h"
Servo myservo, myservo2, myservo3, myservo4, myservo5;
int pos1 = 0;
void setup()
{
    Serial.begin(9600);
    myservo.attach(2);
    myservo2.attach(3);
    myservo3.attach(4);
    myservo4.attach(5);
    myservo5.attach(6);
}
void loop()
{
    myservo.write(pos1);
    myservo2.write(pos1);
    myservo3.write(pos1);
    myservo4.write(pos1);
    myservo5.write(pos1);
}

```