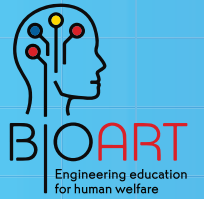




Co-funded by the  
Erasmus+ Programme  
of the European Union



# ПРОТОТИПУВАННЯ БІОМЕДИЧНИХ ПРИСТРОЇВ ТА КОНСТРУКЦІЙ

НАВЧАЛЬНИЙ ПОСІБНИК



2021

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Національний університет «Запорізька політехніка»**



**Пархоменко А.В., Гладкова О.М., Пархоменко А.В.**

**ПРОТОТИПУВАННЯ БІОМЕДИЧНИХ ПРИСТРОЇВ ТА  
КОНСТРУКЦІЙ**

**Навчальний посібник**

**ЖИТОМИР**  
**Видавець ПП «Євро-Волинь»**  
**2021**

**УДК 004.896**

**П18**

*Рекомендовано Вченою радою НУ Запорізька політехніка  
(протокол № 2 від 02.03.2020)*

**Рецензенти:**

Тарасов О.Ф., д.т.н., професор, завідуючий кафедрою комп'ютерних інформаційних технологій Донбаської державної машинобудівної академії;

Теслюк В.М., д.т.н., професор кафедри систем автоматизованого проектування Національного університету «Львівська політехніка».

**Пархоменко А. В., Гладкова О. М., Пархоменко А. В. .**

П18 Прототипування біомедичних пристроїв та конструкцій: навчальний посібник. – Житомир: ПП «Євро-Волинь», 2021. – 202 с.

ISBN 978-617-7992-11-9

Розглянуто особливості процесу прототипування при проектуванні біомедичних пристроїв та конструкцій. Викладено основні принципи віртуального та фізичного прототипування. Наведено практичні приклади створення прототипів кіберпротезу верхньої кінцівки людини та спінального імплантата в середовищі Сгео. Посібник призначений для студентів вищих навчальних закладів, які навчаються за спеціальностями галузей знань 12 «Інформаційні технології», 17 «Електроніка та телекомунікації» та 15 «Автоматизація та приладобудування», а також може використовуватись інженерами, аспірантами і студентами інших спеціальностей для отримання знань та практичного досвіду роботи з системами автоматизованого проектування.

**УДК 004.896**

*Видання здійснено за підтримки міжнародного освітнього проекту «Innovative Multidisciplinary Curriculum in Artificial Implants for Bio-Engineering BSc/MSc Degrees» (BIOART, реєстр. номер 586114-EPP-1-2017-1-ES-EPPKA2-CBHE-JP) програми Erasmus+ Європейського Союзу. Зміст матеріалу відображає думку авторів. Європейська комісія не несе відповідальності за використання інформації, що міститься у виданні.*

© Пархоменко А.В., Гладкова О.М., Пархоменко А.В., НУЗП 2021

© Видавець ПП «Євро-Волинь», видання, 2021

ISBN 978-617-7992-11-9

## ЗМІСТ

Вступ.....	5
1 Аналіз особливостей медичних кіберфізичних систем.....	7
1.1 Поняття кіберфізичних систем.....	7
1.2 Дослідження структурних та функціональних особливостей кіберфізичних протезів руки.....	10
1.3 Аналіз критеріїв та правил розробки кіберпротезів.....	16
1.4 Висновки до розділу 1.....	21
1.5 Контрольні питання.....	22
1.6 Практичні завдання.....	22
2 Дослідження існуючих технологій прототипування.....	23
2.1 Поняття прототипування.....	23
2.2 Метод комплексного віртуального та фізичного прототипування.....	26
2.3 Технології розробки віртуального прототипу.....	27
2.3.1 Прототипування електронної частини кіберпротезу...	27
2.3.2 Прототипування механічної частини кіберпротезу.....	32
2.3.3 Побудова візуальної моделі цифрового двійника .....	38
2.3.4 Практичне використання 3D сканеру Gotcha та програми Scanest.....	48
2.4 Технології розробки фізичного прототипу.....	53
2.4.1 Основи 3D друкування.....	53
2.4.2 Підготовка моделі до 3D друкування.....	63
2.4.3 Практичне використання 3D принтеру LeapFrog Creatr HS та програмного забезпечення Simplify3D.....	66
2.5 Висновки до розділу 2.....	85
2.6 Контрольні питання.....	85
2.7 Практичні завдання.....	86
3 Розробка прототипу кіберпротезу верхньої кінцівки людини.....	87
3.1 Розробка механічної частини прототипу кіберпротезу.....	87
3.1.1 Параметричне моделювання при проектуванні.....	87
3.1.2 Розробка віртуального прототипу конструктиву кіберпротезу.....	90
3.1.3 Виготовлення фізичного прототипу конструктиву кіберпротезу.....	119

3.2 Розробка електронної частини прототипу кіберпротезу.	120
3.2.1 Аналіз та вибір плати Arduino.....	120
3.2.2 Аналіз та вибір плати розширення для Arduino.....	129
3.2.3 Аналіз та вибір серводвигунів.....	133
3.3 Інтеграція компонентів електронної та механічної частин прототипу кіберпротезу.....	140
3.4 Розробка програмного забезпечення для прототипу.....	144
3.5 Висновки до розділу 3.....	151
3.6 Контрольні питання.....	152
3.7 Практичні завдання.....	152
4 Розробка віртуального прототипу спінального імплантанта..	153
4.1 Постановка проєктних завдань та вибір середовища розробки прототипу.....	153
4.2 Аналіз конструктивних та функціональних особливостей спінального імплантанта.....	156
4.3 Інтегроване середовище розробки віртуального прототипу спінального імплантанта.....	159
4.4 Основні етапи побудови базової 3D моделі спінального імплантанта в середовищі Creo Parametric.....	165
4.5 Практичне використання інструментів Program та Excel Analysis для роботи з параметричною моделлю в Creo Parametric.....	173
4.5.1 Робота з інструментом Excel Analysis.....	173
4.5.2 Побудова масиву елементів.....	180
4.5.3 Робота з програмою параметричної моделі.....	183
4.6 Висновки до розділу 4.....	189
4.7 Контрольні питання.....	190
4.8 Практичні завдання.....	190
Висновки.....	191
Перелік джерел посилання.....	192
Алфавітно-предметний покажчик.....	200

## ВСТУП

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

Якщо раніше людина, втративши кінцівку, відчувала значний дискомфорт і неприродність при носінні її замітника, 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 них. Показано, що важливими параметрами, що впливають на показники якості протезу є вага протезу 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 проектуванням, виготовленням та використанням протезів та імплантів. Основна з них полягає в складності розв'язку задачі керування, оскільки контролери протезу мають забезпечувати виконання різноманітних функцій 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 проектування механічної та електронної частин протезу. В результаті, вартість інтелектуальних протезів залишається досить високою, що не дозволяє радикально вирішити проблеми людей з особливими потребами. Тому, одним з напрямків поки залишається розробка більш дешевих та доступних пацієнтам роботизованих протезів кінцівок, що надають базовий набір рухів та можливість простого управління [5].

Як відомо, параметри та характеристики реального виробу можна покращити, а його вартість знизити на основі використання сучасних технологій прототипування [6-7]. Відповідно, актуальною задачею є розробка та дослідження віртуальних та фізичних прототипів протезів та імплантів для підвищення ефективності процесу автоматизованого проектування.

Створення та дослідження віртуального прототипу проектованого виробу дозволяє: виявити помилки проекту на ранніх етапах його реалізації 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 біомедичних пристроїв та конструкцій можуть використовуватися універсальні (PTC Creo, AutoCAD Mechanical Desktop, Solid Work та ін.) або спеціалізовані (наприклад, Autodesk Within Medical) системи автоматизованого проєктування. PTC Creo є однією з найкращих систем автоматизованого проєктування в світі, яка має потужний функціонал, надає інтелектуальне середовище проєктування, а також зручний та простий спосіб отримання академічної ліцензії. Саме тому, середовище PTC Creo Parametric Educational Edition 4.0 було обране для розробки віртуальних прототипів біомедичних конструкцій та пристроїв.

В першому розділі представлено огляд медичних кіберфізичних систем; дослідження структурних та функціональних особливостей існуючих кіберпротезів верхньої кінцівки людини. Виявлено основні критерії та правила розробки кіберпротезів.

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. Представлено аналіз та вибір компонентів для електронної частини прототипу, їх інтеграцію з механічною частиною. Показано особливості розробки програмного забезпечення для функціонування кіберпротезу та сценаріїв тестування досяжних захоплень.

В четвертому розділі представлено методика розробки віртуального прототипу спінального імпланта з використанням технології параметричного моделювання та інтегрованого середовищі розробки на основі інструментів Creo Parametric, Program та Excel Analysis.

# 1 АНАЛІЗ ОСОБЛИВОСТЕЙ МЕДИЧНИХ КІБЕРФІЗИЧНИХ СИСТЕМ

## 1.1 Поняття кіберфізичних систем

Кіберфізичні системи (КФС) – це системи, що складаються з різних природних об'єктів, штучних підсистем і керуючих контролерів, що дозволяють уявити таке утворення як єдине ціле. У КФС забезпечується тісний зв'язок і координація між обчислювальними та 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 й навпаки [8].

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

КФС – це системи взаємодіючих обчислювальних об'єктів, які знаходяться в інтенсивному зв'язку з навколишнім фізичним світом та його поточними процесами, надаючи та використовуючи одночасно послуги доступу та обробки даних, доступні в мережі Інтернет [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 й 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 дисципліні вбудованих систем, комп'ютерів та програмного забезпечення (ПЗ), вбудованих у пристрої (такі як машини, іграшки, медичні прилади та наукові інструменти), основна місія яких – не обчислення. КФС інтегрує динаміку фізичних процесів з технологіями ПЗ та мереж, надаючи абстракції та методи моделювання, проектування та аналізу для інтегрованого цілого [10].

Головним принципом роботи КФС можна назвати глибокий взаємозв'язок між фізичними та обчислювальними елементами. «Мозок» системи, реалізований зокрема на основі технологій штучного інтелекту, отримує дані від сенсорів в реальному світі, аналізує ці дані та використовує їх для подальшого управління фізичними елементами. Завдяки такій взаємодії кіберфізична система здатна ефективно працювати в умовах, що змінюються, як аналог людського організму, що пристосовується до існування 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 грід, реалізованими за допомогою Інтернету речей (Internet of Things, IoT), Індустрії 4.0 (Четверта промислова революція), промислового Інтернету речей (Industrial IoT), міжмашинної взаємодії (Machine-to-Machine, M2M), туманного і хмарного комп'ютингу (fog and cloud computing). Але лише цими технічними засобами не можна обмежувати уявлення про 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 дослідників, перспективи появи КФС і формування на їх основі Індустрії 4.0 є важливими для суспільства в цілому, тому повинні розглядатися не тільки в технічному, а й в більш широкому соціокультурному аспекті, з урахуванням демографічних та інших змін, що відбуваються [8].

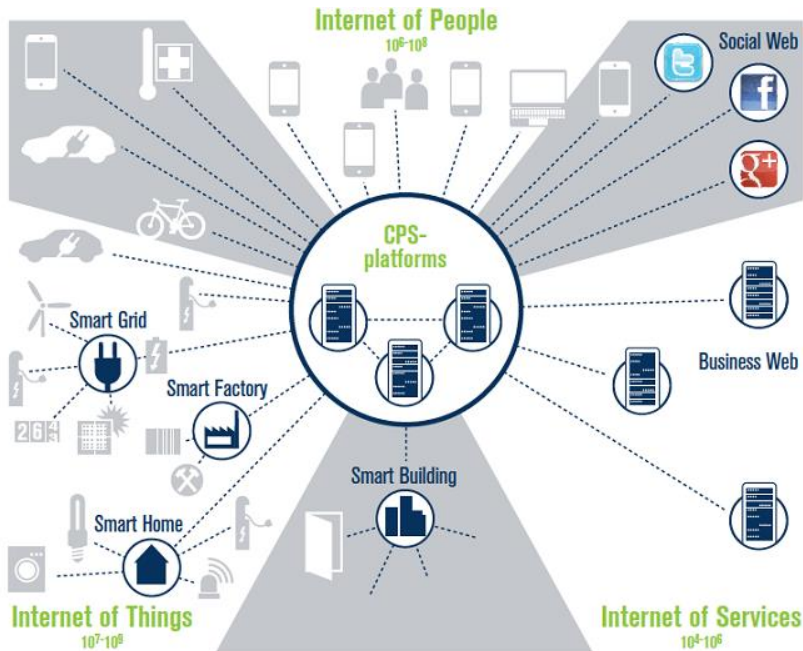


Рисунок 1.1 – Структура кіберфізичної платформи [8]

Галузь 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].

КФС використовуються в системах управління автономним транспортом, які в реальному часі отримують інформацію від учасників дорожнього руху та дорожньої інфраструктури, щоб уникнути аварії та підібрати оптимальний маршрут до місця призначення з урахуванням ситуації на дорозі; медичному обладнанні, яке дистанційно відстежує стан хворих і застосовується для дослідження людського організму; «розумних» будинках з ефективним споживанням електроенергії; засобах 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 т.д. 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. Кіберпротез, оснащений рушьями 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 потрібні свої налаштування. Перемикання режимів руху протеза вже не може бути автоматичним, оскільки намір змінити рух народжується в голові у людини, кіберпротезу 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 т.д.). Перетворені в сигнали для механізмів протеза, вони дозволяють перемикати 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 високою надійністю детектують ознаки того чи іншого наміру в корекції руху, які тут же транслюються до протезу як команди для перемикання його автоматики в потрібний режим: рух вгору по сходах, поворот направо і т.д. 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, представляють 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 Vincent та Michelangelo обмежена. У роботі [1] наведено властивості та характеристики для кожного кіберпротезу руки, що заявлені виробником, або зібрані з відео та вторинних джерел. Пристрій SensorHand, розроблений Отто Боком, також вказаний у авторах для порівняння можливостей руки одного ступеня свободи з сучасними багатофункціональними конструкціями рук.

Таким чином, дані, які представлені в статті [1] були зібрані з численних джерел, але експлуатаційні характеристики кожної руки було протестовано за допомогою єдиної процедури тестування,



а – рука Vincent від Vincent Systems; б – рука iLimb від Touch Bionics; в – iLimb Pulse від Touch Bionics; г – рука Bebionic від RSL Steeper; д – рука Bebionic v2 від RSL Steeper; е – рука Michelangelo від Отто Бока

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

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].

Наприклад, кіберпротез i-Limb від компанії Touch Bionics (рис. 1.3) повертає людині можливість використовувати руку: носити вантаж та брати пальцями дрібні предмети [13].



Рисунок 1.3 – i-Limb від Touch Bionics [13]

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.

Кіберпротез i-Limb є одним з найдоступніших, але далеко не самим високотехнологічним. Американське оборонне агентство DARPA фінансує відразу два проекти з біомехатроніки, яким вдалося просунути 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.

Управлятися Luke Arm може тією частиною тіла, якою забажає власник, наприклад ступнею, тобто щоб рухати кіберрукою, досить притупувати ногою [13].

Шведсько-італійський 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].

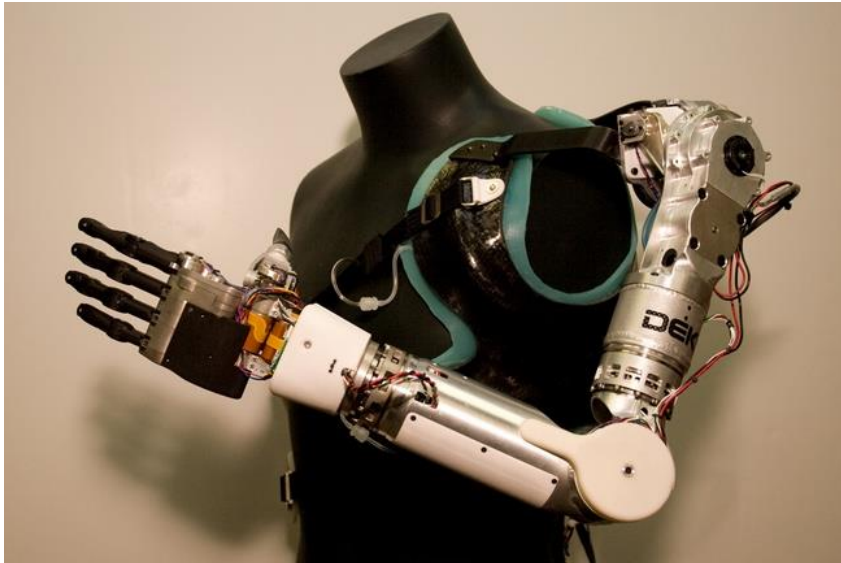


Рисунок 1.4 – Deka Luke Arm [13]

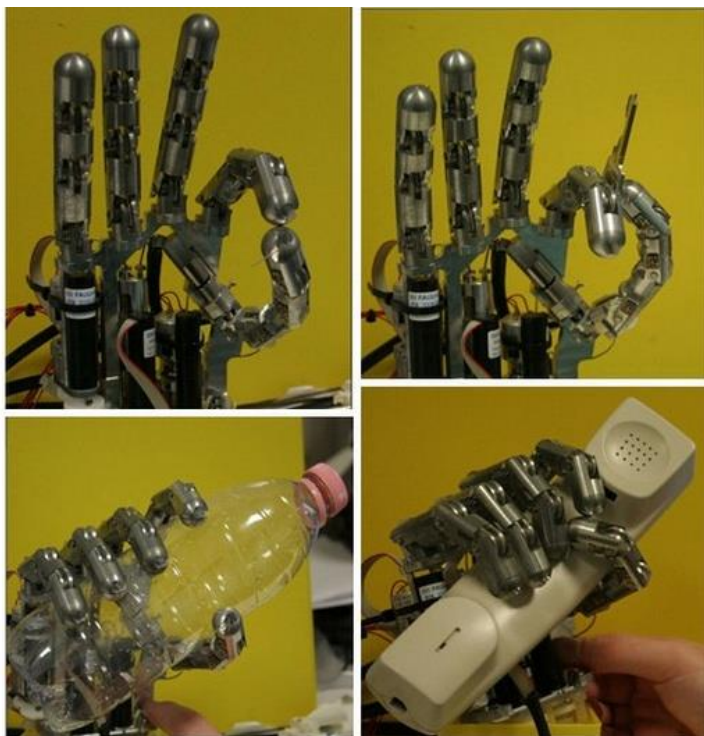


Рисунок 1.5 – Smart Hand [13]

У випадках, коли руку не втрачено, але через проблеми з нервовою системою вона втратила можливість рухатися, може використовуватися Possessed Hand (рис.1.6).

Цей пристрій одягається на руку і стимулює м'язи згідно заданої програми, а тому може допомогти і здоровим людям. Наприклад, бажаним навчитися грати на скрипці, ліпити з глини або жонглювати кулями. Це дозволяє не жертвувати здоровою кінцівкою заради кібернетичної, нехай навіть більш досконалої. Але проект 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.

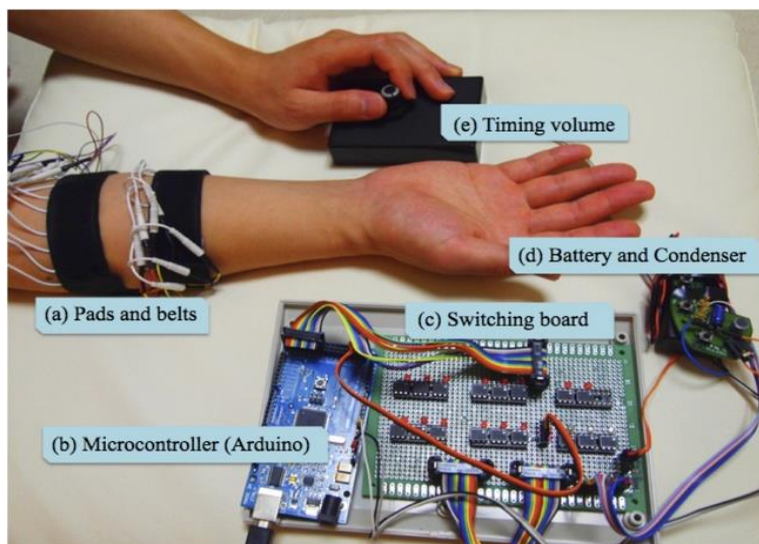


Рисунок 1.6 – Possessed Hand [13]

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 [14].

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

### 1.3 Аналіз критеріїв та правил розробки кіберпротезів

В роботі [17] вказано, що для проєктування та виготовлення якісного кіберпротезу проєктувальнику 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].



Рисунок 1.7 – Критерії розробки кіберпротезу

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.

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 протезів 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 обертанні 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 кіберпротезі. 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 непотрібна, 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, витратами 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 ступенів свободи, кількість компонентів) та збільшує розмір і вагу [1].

Таким чином, компроміс між характеристиками дизайну в основному вирішує проєктувальник через жорсткі 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 вирішення проблеми множинного введення/виведення сигналів, порівнювати особливості і характеристики з практичною функціональністю, а також робити постійні поліпшення та ітерації дизайну [1].

Таким чином, 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 всього протезу 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 розроблені з малою кількістю виконавчих механізмів з передачами, що дозволяють приймати всі функціональні позиції захоплення.

## 1.4 Висновки до розділу 1

Кіберфізичні протези вже сьогодні використовуються в якості функціональних заміників 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 проектування кіберпротезів на основі сучасних технологій прототипування є актуальною задачею.

### **1.5 Контрольні питання**

1. Дайте визначення КФС.
2. Які системи можна вважати попередниками КФС?
3. Що є головним принципом роботи КФС?
4. З яких мереж складається кіберфізична платформа?
5. В яких галузях застосовуються КФС?
6. Які типи протезів бувають?
7. Що таке кіберпротез?
8. Які кіберпротези представляють собою останні розробки в області комерційних міоелектричних рук?
9. Які вимоги висуваються до конструкції кіберпротезу при його проектуванні?
10. Які двигуни використовуються в якості приводу при розробці кіберпротезів?
11. Що таке досяжні захоплення?
12. Назвіть основні емпіричні правила для проектування кіберпротезів рук.

### **1.6 Практичні завдання**

*Завдання 1.* Виконайте дослідження останніх досягнень в галузі розробки кіберпротезів та підготуйте мультимедійну презентацію на тему «Сучасні тенденції в протезуванні кінцівок людини».

*Завдання 2.* Виконайте огляд джерел інформації та напишіть реферат на тему «Технології швидкого прототипування при проектуванні кіберпротезів».

## 2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ТЕХНОЛОГІЙ ПРОТОТИПУВАННЯ

### 2.1 Поняття прототипування

Прототипування – це створення схеми або готового макету, 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 неперевічених 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; а після того, як це було зроблено, 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, користувальницький 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 етапів.

Першим етапом є визначення вимог.

Другим етапом є розробка першого (ескізного) варіанту прототипу.

Третій етап – демонстрація та критична оцінка (вивчення прототипу замовником і кінцевим користувачем, отримання зворотнього зв'язку про необхідні зміни та доповнення).

Четвертий етап – 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 прототипування [20].

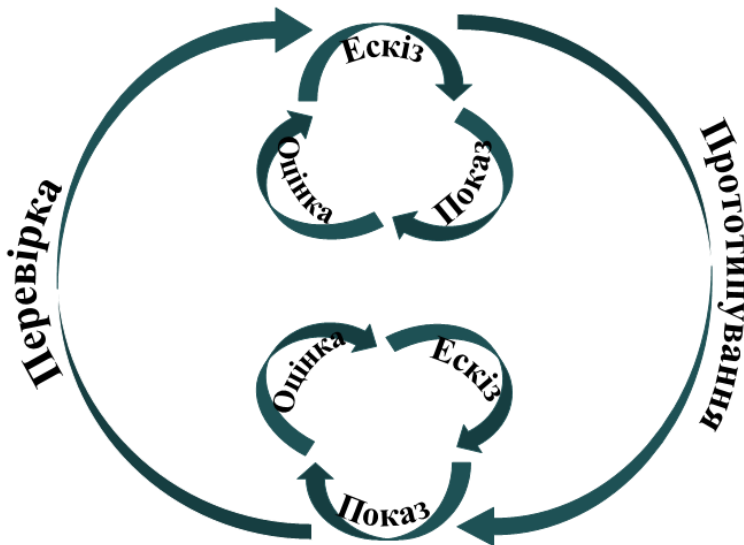


Рисунок 2.1 – Ітераційний процес прототипування [20]

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, причому на тестування і налагодження системи витрачають порівняно більшу кількість зусиль. Такий підхід використовується в машино- і приладобудуванні, програмуванні і в багатьох інших галузях техніки.

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

Найчастіше термін «прототипування» пов'язують з термінами «швидке прототипування» та «3D принтінг». Але, як свідчать проведені дослідження, на сьогоднішній день прототипування – це назва декількох суміжних технологій, які активно використовуються при проектуванні 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 проектування взаємопов'язаних електронної, механічної та програмної частин проєкту.

Для створення й дослідження віртуальних прототипів схем та конструкцій кіберпротезів можуть використовуватись сучасні системи автоматизованого проектування та моделювання (Altium Designer, EAGLE, PROTEUS, Tinkercad Circuits Arduino, PTC Creo, Solid Edge, AutoCAD, 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 проєкті. Крім того, створення фізичного прототипу – це зручний спосіб отримання зворотнього зв'язку з замовником [5].

## 2.2 Метод комплексного віртуального та фізичного прототипування

Схему методу комплексного прототипування наведено на рисунку 2.2. Можна виділити три основні стадії: розробка фізичної (механічної) частини прототипу, розробка електронної частини та розробка ПЗ.

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.



Рисунок 2.2 – Метод комплексного віртуального та фізичного прототипування

Етапами розробки електронної частини прототипу кіберпротезу є:

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

Розробка ПЗ 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.

## **2.3 Технології розробки віртуального прототипу**

### **2.3.1 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 Circuits Arduino, який є онлайн емулятором. Можна скористатись також 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 (рис. 2.3) – 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 реалізації власних проєктів [22]. Основним функціоналом Tinkercad Arduino Circuits є:

- візуальні редактори для схем та коду;
- набір 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 проекти;
- можливість інтеграції з рештою функціональності Tinkercad і швидким створенням для розроблюваного пристрою корпусу та інших конструктивних елементів (створену модель можна відразу відправити на 3D принтер);
- вбудовані посібники й величезне співтовариство з колекцією готових проєктів [22-23].

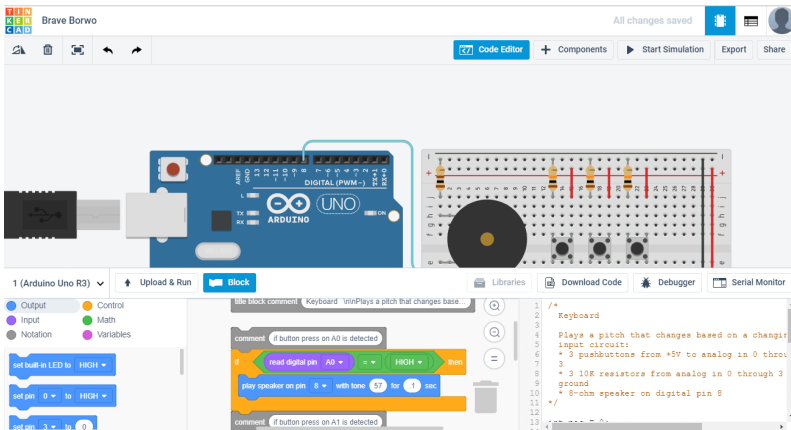


Рисунок 2.3 – Робоча область Tinkercad Circuits Arduino [22]

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 виконаних на основі мікроконтролерів (МК) різних сімейств (рис. 2.4). 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 МК. Proteus також має можливість імітувати взаємодію між ПЗ, що працює на МК, 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].

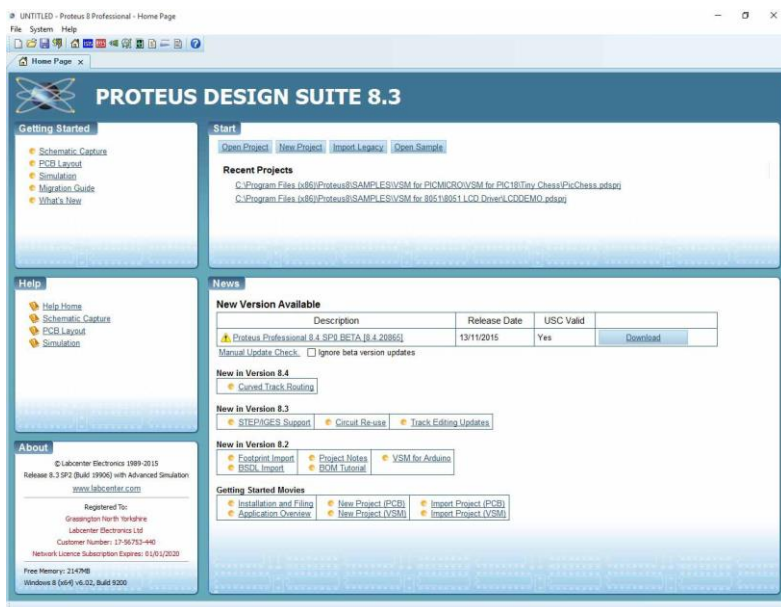


Рисунок 2.4 – Головне вікно Proteus Design Suite [24]

Proteus надає ряд переваг:

- виконання всіх етапів розробки електронного пристрою на основі МК в єдиному середовищі;
- можливість написання, налагодження та тестування вбудованого ПЗ ще до фізичного виготовлення дослідного зразка системи;
- генерація діагностичних повідомлень (наприклад, при виконанні непередбачуваної інструкції) як з боку CPU, так і з боку

моделей пристроїв введення-виведення, що дозволяють виявити складні в пошуку помилки програмування;

- прискорений 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 МК ARM Cortex-M та AVR, що засноване на Visual Studio [26].

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 МК, 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 (рис. 2.5).

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

- підтримка 500 + AVR і SAM пристроїв;
- велика бібліотека вихідного коду, включаючи драйвери, 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 ПЗ від Microchip;
- можливість налаштувати та протестувати продуктивність бездротових конструкцій за допомогою бездротового композера;
- запис і налагодження C/C++ і складання коду з вбудованим компілятором;
- додаткові функції налагодження включають складні точки переривання даних, підтримку трасування (пристрої SAM3, SAM4),

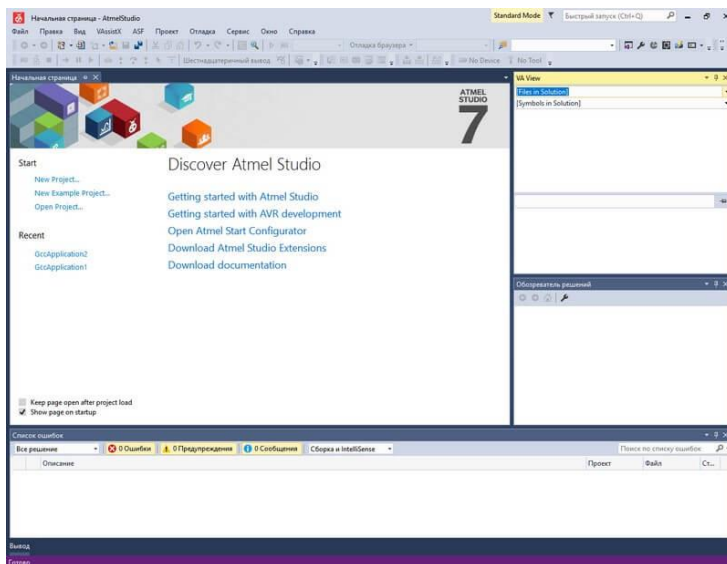


Рисунок 2.5 – Домашня сторінка Atmel Studio [26]

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 легкої розробки коду та налагодження;
- повний чіп моделювання для точної моделі CPU процесора, переривання, периферійних пристроїв і зовнішніх чинників [27].

Завдяки зв'язці програмних пакетів Atmel Studio і Proteus від фірми Labcenter Electronics можливе програмування МК 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 ПЗ, то можна використати Tinkercad Circuits Arduino та створювати проекти онлайн.

### 2.3.2 Прототипування механічної частини кіберпротезу

Побудова віртуального прототипу механічної частини кіберпротезу зазвичай відбувається на основі твердотільної моделі, створеної в системі автоматизованого проектування (САПР) або моделі з замкнутими поверхневими контурами [28].

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

Деякі середовища проектування є спеціалізованими для використання в області медицини, але більшість є універсальними.

САПР легкого рівня призначені для 2D-проектування та креслення, а також для створення тривимірних моделей окремих деталей без можливості роботи зі складальними одиницями.

Безумовний лідер серед базових САПР – AutoCAD, від компанії Autodesk. AutoCAD є найпоширенішою CAD-системою в світі, яка дозволяє проектувати як в двовимірному, так і в тривимірному середовищі. За допомогою AutoCAD можна будувати 3D моделі, створювати та оформляти кресленники й багато іншого. AutoCAD є універсальною САПР, тобто ця система не має чіткої орієнтації на певну проектну галузь, в ній можна виконувати різні проекти [29].

Система автоматизованого проектування AutoCAD має наступні особливості:

- це 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 розробників.

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 ВІМ-технологій [29].

САПР середнього рівня – це програми для 3D моделювання виробів, проведення розрахунків та автоматизації проектування різноманітних систем. Дані в таких системах можуть зберігатися як у звичайній файлової системі, так і в єдиному середовищі електронного документообігу та управління даними (PDM- і PLM-системах). Часто в системах середнього класу присутні модулі для підготовки керуючих програм для верстатів з ЧПУ (Числове програмне управління), а також інші програми для автоматизації технологічного проектування (САМ-системи). САПР середнього рівня – найпопулярніші системи на ринку. Вони вдало поєднують в собі співвідношення "ціна/функціональність", здатні вирішити переважну кількість проектних завдань і задовільнити потреби більшої частини клієнтів.

Autodesk Inventor – це професійний комплекс для 3D проектування промислових та медичних виробів і випуску документації. Розробник – компанія Autodesk [29].

Серед особливостей Inventor варто відзначити:

- просунуті інструменти 3D 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 галузях, включаючи дизайн, анатомічно точне проектування та інші;

- підтримку вітчизняних стандартів при проведенні розрахунків, моделюванні та оформленні документації;

- великі бібліотеки стандартних і часто використовуваних елементів;

- широкі можливості параметризації деталей і складальних одиниць, в тому числі управління складом виробу;

- вбудоване середовище створення правил проектування iLogic.

Для ефективного управління процесом розробки виробів, управління інженерними даними та організації колективної роботи

над проектами, 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 ПЗ дозволяє дизайнерам імплантів створювати пористі покриття для своїх імплантів, щоб сприяти осеоінтеграції, тобто зрощенню між кісткою та імплантом.

Ключові риси середовища Autodesk Within Medical:

– дозволяє дизайнерам розробляти пористі (для сприяння осеоінтеграції з різними конфігураціями розміру пор та поверхніми ґратки) та тверді частини (для сприяння фіксації пористого імпланта з кісткою, використовуючи шорсткі поверхні ґратки);

– 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 EBМ, дозволяючи отримати точний вихід.

Autodesk Within Medical може бути використаний для проектування пористих імплантів, де пористість може бути налаштована за допомогою ПЗ для забезпечення оптимальної осеоінтеграції. ґратки топології, що постачаються разом з цим ПЗ, були розроблені з урахуванням росту клітин і найкраще підходять для цієї функції [30].

SolidWorks – це програмний комплекс для автоматизації конструкторських робіт промислового підприємства від компанії Dassault Systemes. Риси системи, що вигідно відрізняють її від інших САD-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 проектування медичних виробів. Solid Edge допомагає як невеликим, так і крупним виробникам покращити процеси проектування 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 проектування нового покоління;
- прискорення проектування механічної частини, використовуючи 3D CAD;
- візуалізація конструкторських задумів (створення фотореалістичних зображень і анімації протезів чи органів, використовуючи 3D рендеринг);
- інтегрування електромеханічного проектування;
- підготовка проєктів до 3D друку;
- легке створювання моделі з листового металу;
- забезпечення відповідності державним вимогам і вимогам замовників;
- використання безпаперового документообігу;
- тісну інтеграцію з Microsoft SharePoint і PLM-системою Teamcenter для спільної роботи і управління даними;
- спільне використання даних про продукт, на основі хмарних сховищ [31].

Компас-3D – це система параметричного моделювання деталей і складань, що використовується в галузях 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 проектування окремих розділів проекту;

- гнучкий підхід до оснащення робочих місць проектувальників, що дозволяє заощадити при покупці;

- можливість інтеграції з системою автоматизованого проектування технологічних процесів ВЕРТИКАЛЬ і іншими системами для створення єдиного комплексу [29].

T-FLEX – САПР середнього рівня, побудована на основі ліцензійного тривимірного ядра Parasolid. Розробник системи – компанія ТопСистеми (Росія).

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].

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

PTC Creo – система 2D і 3D параметричного проектування складних виробів від компанії PTC. САПР PTC Creo широко використовується в самих різних галузях проектування систем [29, 32].

Вигідні відмінності системи від конкуруючих рішень:

- ефективна робота з великими і дуже великими складальними одиницями;

- 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 – 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 проєкту;

– повнофункціональне рішення 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 проєктом;

– підтримка міждисциплінарної розробки систем [29].

Останнім часом активно почали розвиватися "хмарні" САПР, які працюють у віртуальному обчислювальному 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 комп'ютерах з обмеженими ресурсами, оскільки 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.

Наприклад, САПР Fusion 360 від компанії Autodesk орієнтована на вирішення широкого кола завдань, починаючи від простого моделювання та закінчуючи проведенням складних розрахунків. Особливості Fusion 360:

- високий рівень інтерфейсу користувача;
- поєднання різних методів моделювання;
- розвинені інструменти роботи зі складальними одиницями;
- 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 однойменною компанією. Функціональні можливості Onshape:

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

### 2.3.3 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. 3D сканування дає можливість створити візуальну модель

цифрового двійника досліджуваного об'єкта 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. 2.6) [34].

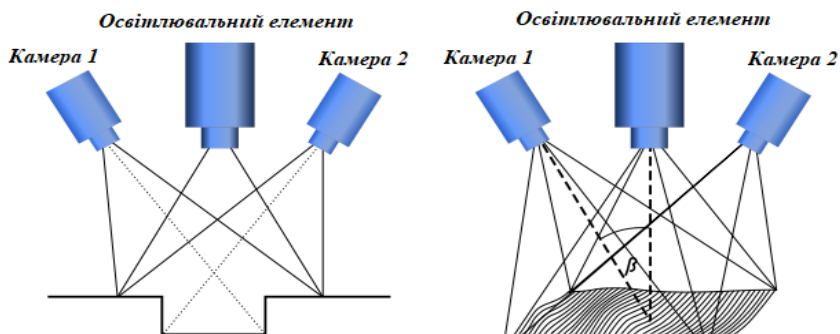


Рисунок 2.6 – Схематичне зображення камер та елементів освітлення на 3D сканері [34]

Принцип роботи 3D сканера визначається технологією сканування, в загальному сенсі – це здатність приладу визначати відстань до об'єкта, а отримані дані перетворювати в цифрове зображення (тривимірну модель) та передавати 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 особливі сигнали (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 2.7 разом з двома основними типами сканерів.

Активні безконтактні 3D сканери (лазерні). Для вивчення об'єкта використовують структурований світловий або лазерний промінь, який потрапляючи на об'єкт, відбивається й 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 не можливе.

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.

Їх 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].

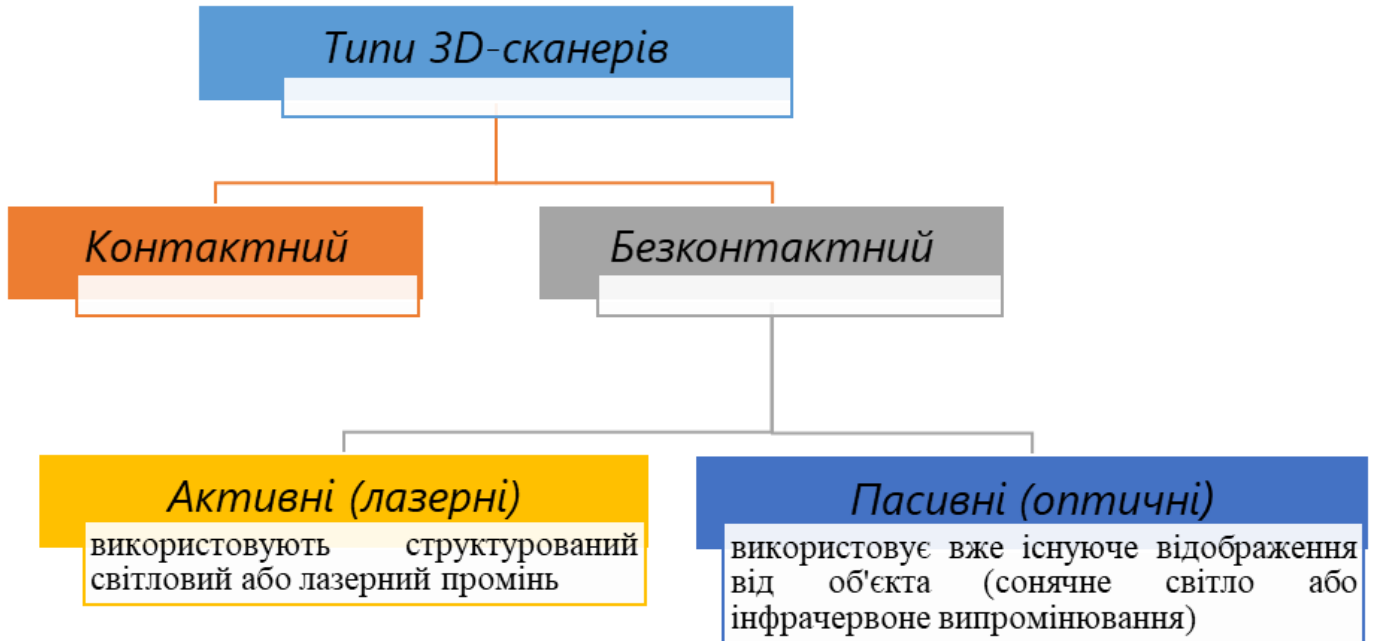


Рисунок 2.7 – Типи 3D сканерів та технологій сканування

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 надвеликих об'єктів (наприклад, будинки, пам'ятники, тощо).

До переваг контактних 3D сканерів можна віднести:

- простоту використання;
- незалежність від рівня освітлення;
- створення моделі високої точності;
- файл 3D моделі невеликий за обсягом.

До переваг безконтактних 3D сканерів можна віднести:

- енергоекономічність;
- 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. За 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].

Таким чином, сучасні 3D сканери – це електронні пристрої, що здатні аналізувати параметри фізичного об'єкта і на основі отриманих даних створювати точну 3D модель на екрані комп'ютера. Порівняльну характеристику декількох засобів наведено в таблиці 2.1.

Таблиця 2.1 – Порівняльна таблиця засобів 3D сканування

№ п/п	Характеристика	Назва апаратних засобів		
		3D сканер artec eva	3D сканер 4DDynamics Gotcha	3D сканер sense
1	Передача кольору	24 бит	повноколірний сканер	повноколірний сканер
2	Сканування об'єктів	Так	Так	Так
3	Сканування людини	Так	Так	Так
4	3D розширення, мм	до 0,5	0,5 – 5	х/у на відстані 0,5 – 0,9
5	Точність 3D точок, мм	до 0,1	0,5	1,0
6	Джерело світла	фотоспалах (не лазер)	внутрішній спалах	фотоспалах (не лазер)
7	Робоча відстань, м	0,4 – 1	0,5 – 3	0,45 – 1,6
8	Частота кадрів зйомки, кадрів в секунду	16	30	30
9	Вихідний формат даних	OBJ, STL, VRML, ASCII, AOP, PLY, PTX, X	PLY, OBJ, STL, VRML	STL, PLY, OBJ
10	Роздільна здатність	640x480	640x480	640x480
11	Інтерфейс	USB2.0	USB 2.0	USB 2.0

В якості прикладу можна розглянути 3D сканер Gotcha, виробництва 4D Dynamics, Бельгія (рис. 2.8).



Рисунок 2.8 – Зовнішній вигляд 3D сканера Gotcha

Безконтактний 3D сканер Gotcha від 4DDYNAMICS [32, 33] працює за принципом триангуляційного лазерного далекоміру. Використовується червоний лазер, який має достатню потужність, щоб отримувати чіткі відбиті сигнали від предметів на відстані до 2,5 метрів.

3D сканер Gotcha є зручним та досить простим у використанні. В ньому передбачена передача кольорової структури. Сканер має точність 1 мм та власне ПЗ, що дозволяє відносно недорого здійснювати 3D сканування з високою роздільною здатністю. В таблиці 2.2 наведено технічні характеристики цього сканера [38, 39].

Безконтактний 3D сканер при скануванні не контактує з об'єктом, а використовує стереозір. Сенсор PrimeSense 3D Depth ([www.4DDynamics.com](http://www.4DDynamics.com)) є комплексним рішенням, яке забезпечує сприйняття світла в тривимірному просторі.

Таблиця 2.2 – Технічні характеристики 3D сканера Gotcha

Характеристика	Значення/опис
Країна-виробник	Бельгія
Точність сканування, мм	1
Метод сканування	Безконтактний
Технологія сканування	Оптична
Тип проєктора	Інфрачервоний
Сканування текстур	Корольова текстура
Максимальна відстань до об'єкту, м	0.5
Мінімальна відстань до об'єкту, м	3
Фокусна відстань, мм	3-6
Фокусування в межах, м	0,5 – 3,0
Частота кадрів, кадрів/сек	30
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

Він переводить сприйняту інформацію в синхронізовані зображення, так само, як це роблять люди, з використанням природних процесів. Це робить дане рішення практичним, зручним, інтуїтивно зрозумілим і простим у використанні.

Канал когерентного лазерного випромінювання жорстко фіксований за просторовими координатами та охоплює простір в діапазоні, приблизно, 90 по вертикалі і 90 по горизонталі. Розгортка променя здійснюється з використанням керованих дзеркал і зі швидкістю близько 3 секунд на вертикальне переміщення в межах заданого кута і безперервне сканування з високою частотою кожної горизонтальної лінії.

Для створення тривимірного образу предмету використовується метод оптичної цифрової реєстрації положення відбитих від предмету сигналів лазерного випромінювання, спільно з інформацією про відстань до кожної зафіксованої точки.

Для реєстрації відображених сигналів використовується оптичний канал, аналогічний цифровому фотоапарату з роздільною здатністю 1.3 МРх. Як і для будь-якого цифрового фотоапарату, для отримання чіткого зображення потрібно вірно налаштувати різкість в умовах конкретної зйомки.

Таким чином, результатом сканування є хмара точок, кожна з яких дає інформацію про стан у видимій області оптичного каналу та відстані до неї. Саме ця інформація передається сканером для подальшої обробки.

Для попереднього налаштування на сканований предмет сканер має власну систему управління, яка не дозволяє використовувати його без підключення до комп'ютеру. Керування сканером здійснює ПЗ, встановлене на комп'ютері, до якого підключений сканер.

Сканер під'єднується до комп'ютера як зовнішній SCSI-пристрій і автоматично визначається при старті операційної системи. Це означає, що сканер повинен бути ввімкнений до включення комп'ютера, на якому встановлено ПЗ для роботи зі сканером (до відкриття ПЗ). В іншому випадку, операційна система не зможе правильно зареєструвати пристрій і ПЗ не буде мати доступу до управління сканером.

Оскільки діапазон просторового охоплення сканованої області фіксований і максимальна відстань до предмета обмежена 2,5 м, тому при скануванні предметів різного розміру найвигідніше вибирати ту відстань, при якій предмет займає не менше  $\frac{3}{4}$  площі кадру оптичного каналу сканера. Однак, для невеликих предметів такий вибір не завжди можливий, тому що паралакс оптичного і лазерного каналів дуже великий. Коли відстань до предмета наближається до подвоєного паралаксу, то окремі частини виявляються в мертвій зоні для лазера, хоча і відображаються по оптичному каналу. У зв'язку з цим рекомендується вибирати відстань до предмета в межах 0,8 - 2 м.

Оскільки площинні координати точок визначаються по відбитому лазерному променю, то оптичний канал повинен бути дуже точно сфокусований на сканованому предметі.

Фокусування оптичного каналу здійснюється за принципом визначення максимальної контрастності окремих ділянок одержуваного зображення і автоматично налаштовується перед

зйомкою кожного скану. Оскільки використовується принцип максимальної контрастності, то сканер не завжди може правильно встановити відстань до необхідної для нас точки, а визначає відстань до найбільш контрастних ділянок предмету.

Якщо предмет не має яскраво виражених контрастних ділянок – автофокусування буде неможливим і сканування проводитися не буде. Такі випадки трапляються при зйомці однотонних предметів, для яких автофокус може визначити довільну відстань, що не відповідає істині.

В цьому випадку, хоча сканування і буде здійснено, точки відображення не будуть реально сфокусовані і не розпізнаються оптичною системою сканера. Відповідно, хмара точок не буде передана в програму і з'явиться повідомлення про помилку: «There are no points for import».

Іноді при скануванні темного предмета, коли основна енергія лазера поглинається матеріалом предмета, може з'явитися повідомлення про помилку: «Laser barrier is close». Це означає, що сканер вимкнено, або обраної потужності лазера не вистачає для отримання нормального відбитого сигналу.

Таким чином, маючи на увазі все вищевикладене, необхідно вибирати умови зйомки в залежності від розміру предмету і властивостей його поверхні.

Сканований предмет необхідно розташовувати на відстані не менше 20 см від фону для того, щоб сканер не сфокусувався на ньому. Фон повинен бути темнішим за предмет, бажано чорним, щоб уникнути відбиття лазерного променя.

Паралельно з об'ємним скануванням проводиться цифрова фотозйомка предмету для фіксації текстури. Зйомка проводиться з роздільною здатністю 1.3 MPx.

ПЗ від 4DDynamics трансформує сенсор в повнофункціональний 3D сканер. Після завершення сканування, воно також дозволяє редагувати скановані зображення і об'єднати різні зображення, щоб створити 3D модель високої якості. В процесі роботи досить обнести сканований об'єкт, та ПЗ сканера автоматично сформує готовий файл в STL-форматі.

Для 3D сканування існує ряд програм.

David-3D – призначається для тривимірного сканування предметів і перетворення отриманих результатів з метою подальшого імпорту моделей в 3D редактори.

Artec Studio 10 – професійний інструмент для створення об'ємних моделей.

Autodesk 123D Catch – тривимірне сканування для мобільних телефонів на ОС Android.

Scanect – призначається для тривимірного сканування предметів і перетворення отриманих результатів з метою подальшого імпорту моделей в 3D редактори.

Photomodeler Scanner – дозволяє формувати високоточні stl-моделі на основі звичайних знімків, зроблених камерою смартфона або планшета.

3DAround – перетворює фото в форматі 2D в реалістичні тривимірні моделі.

### **2.3.4 Практичне використання 3D сканеру Gotcha та програми Scanect**

Для роботи зі сканером використовується програма Scanect, яка доступна для завантаження за посиланням <https://scanect.occipital.com/download/>.

Ця програма орієнтована на роботу з 3D сканерами саме цієї фірми, та є сумісною (універсальною) з різними моделями сканерів і володіє додатковими можливостями по обробці отриманих результатів.

Після запуску програми на екрані з'явиться її головне вікно (рис. 2.9), обираємо «New», встановлюємо необхідні параметри сканування, а саме, визначаємо об'єкт сканування (Scene: частини тіла, предмети, уся кімната, пів кімнати), габарити сканованого об'єкта, співвідношення сторін (Normal – співвідношення сторін однакове наприклад 1x1x1; Height x 2 – висота вдвічі більша, наприклад 1x2x1) і т.п. Далі натискаємо кнопку «Start».

Для того, щоб розпочати сканування у Вікні сканування (рис.2.10) необхідно у лівому верхньому куті вікна натиснути

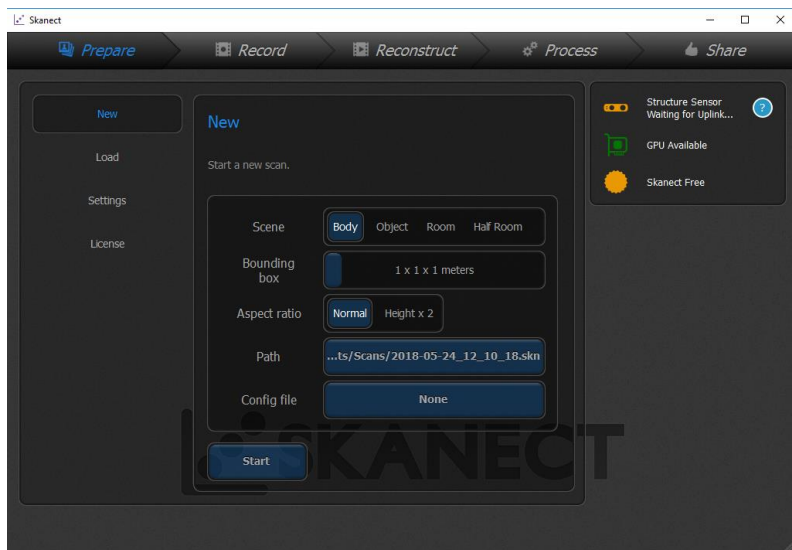


Рисунок 2.9 – Головне вікно програми «Skanect»

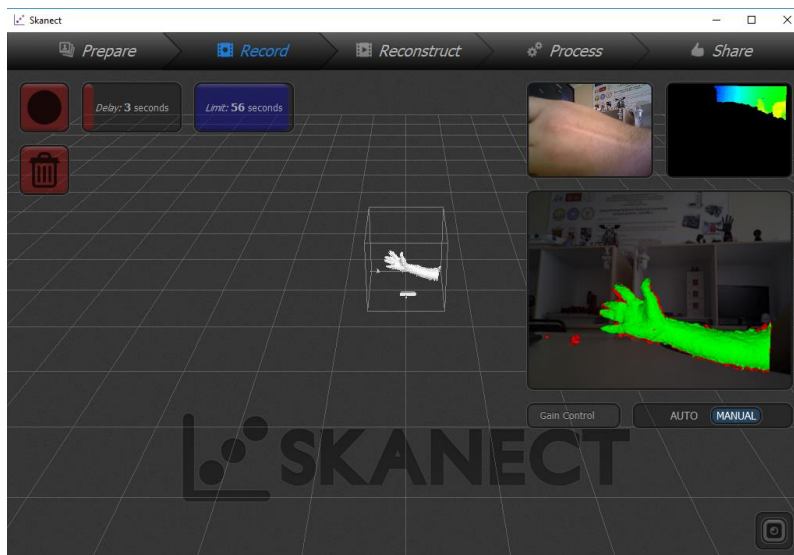





Рисунок 2.10 – Вікно сканування

кнопку . Поряд з цією кнопкою є повзунок для встановлення затримки початку сканування (Delay), за замовчанням встановлено затримку 3 секунди, та повзунок для обмеження часу сканування (Limit). Для того, щоб зупинити сканування у лівому верхньому куті треба натиснути кнопку . Якщо при скануванні об'єкта виникла помилка, то після того, як сканування зупинили, необхідно натиснути кнопку видалення поточного сканованого об'єкта .

Визначення геометрії сканованого об'єкта. У вікнах на рисунку 2.10 представлено зображення відсканованого об'єкта об'єднане полігонами як вид зверху (Top), спереду (Front), праворуч (Right) і ізометричне вікно тривимірного відображення, в якому можна розглядати тривимірне відображення об'єкта в різних ракурсах.

Симуляція відсканованої 3D моделі. На вкладці «Reconstruct» обрати в меню «Fusion», задати параметри Processor: GPU і Fidelity: High та натиснути «Run» (рис. 2.11).

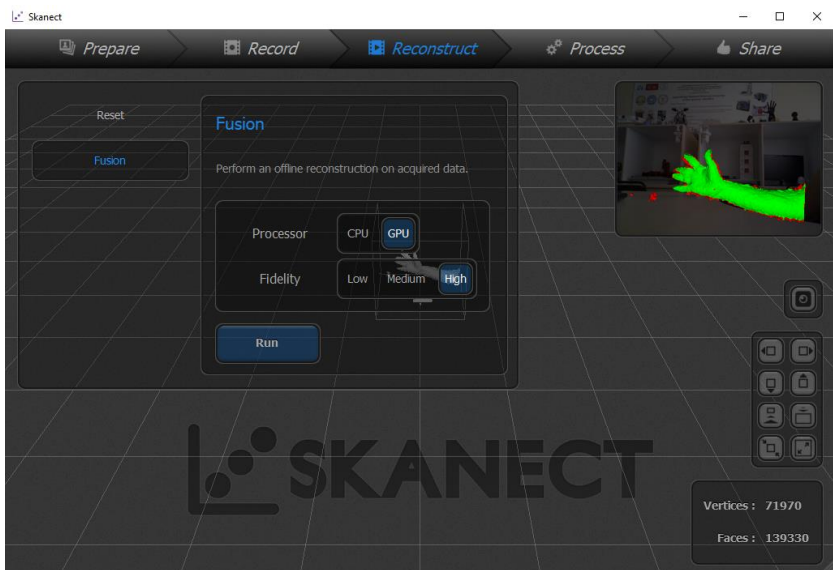


Рисунок 2.11 – Вікно налаштувань

На вкладці «Process» можна виконати деяку обробку відсканованої 3D моделі (рис. 2.12).

Пункт «Remove Parts» в меню «Geometry» видаляє всі об'єкти (рис. 2.13), які є меншими за відсканований. Таким чином за його допомогою можна видалити увесь відсканований разом з головним об'єктом «шум».

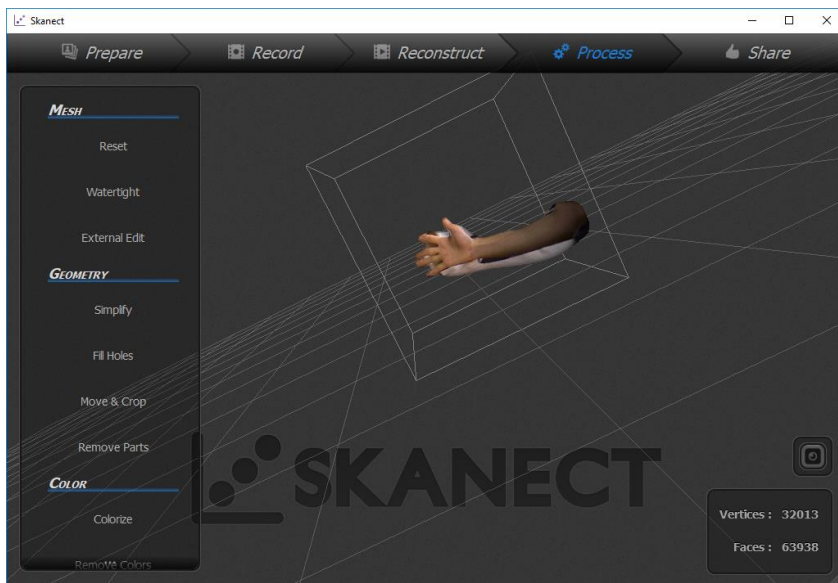


Рисунок 2.12 – Вікно обробки сканів

Для того, щоб представити відсканований об'єкт в кольоровому вигляді, можна скористуватися можливостями меню «Color» пунктом «Colorize» (рис. 2.12).

Для заповнення пустот в отриманій 3D моделі використовується «Fill Holes» з пункту меню «Geometry» (рис. 2.14). Можна встановити стратегію заповнення пустот (Strategy), згладжування (Smoothing) та межі (Limit). Редагувати положення відсканованого об'єкта (повертати відносно осей X, Y та Z, та переміщувати відносно осі Z) можна за допомогою пункту «Move&Crop» (рис. 2.15).

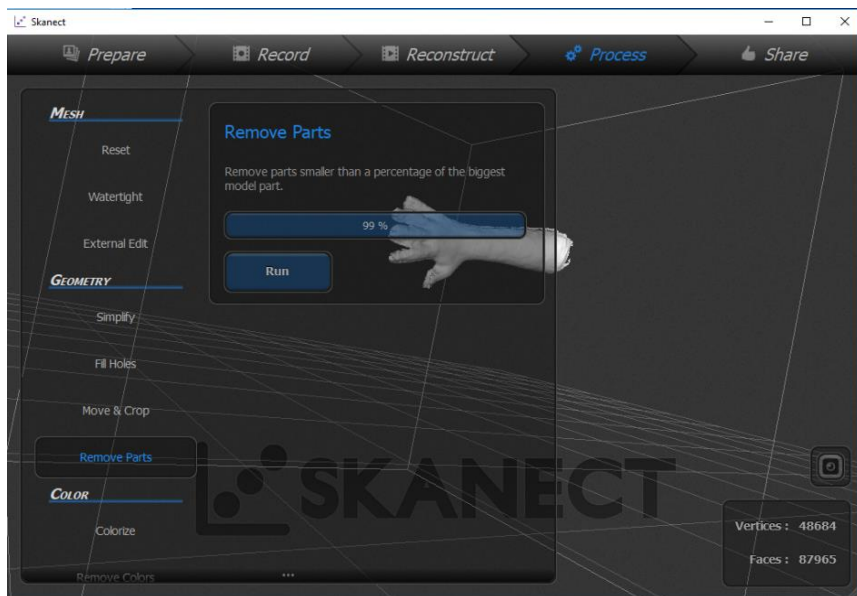


Рисунок 2.13 – Виконання очистки 3D моделі

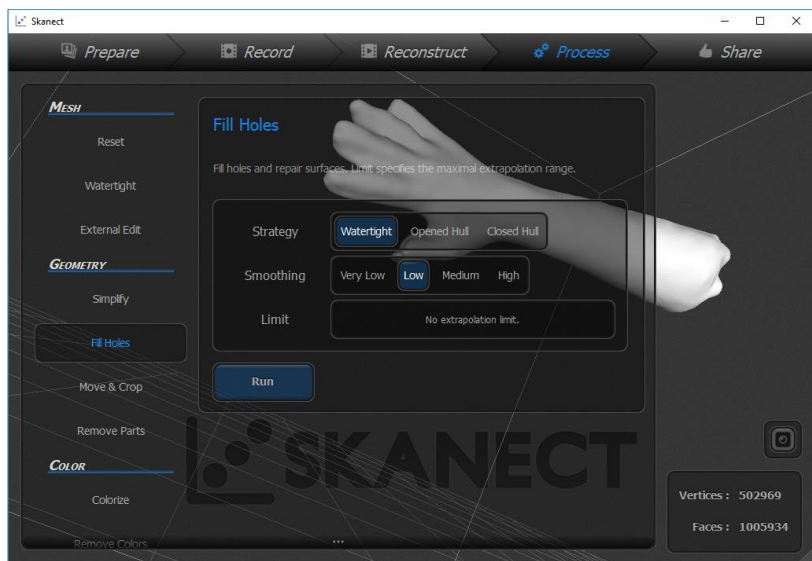


Рисунок 2.14 – Вікно заповнення порожніх ділянок поверхонь

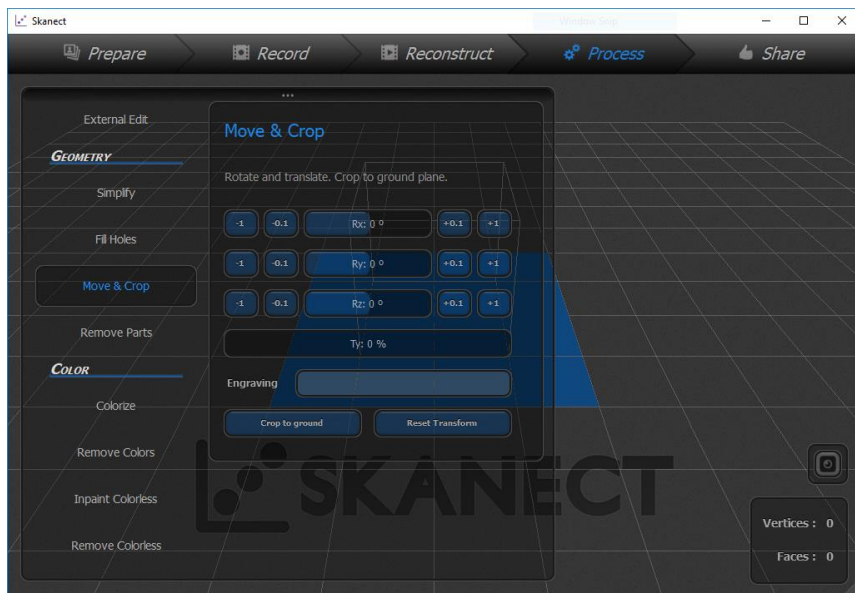


Рисунок 2.15 – Налаштування положення об'єкта

Збереження 3D моделі в форматі .STL. На вкладці «Share» обрати в меню «Local» пункт «Export Model». Формат моделі (Format) обрати .STL, повзунок #Faces перетягнути до крайнього лівого положення, таким чином буде встановлено максимальна можлива кількість. Далі натиснути «Export Preview (по PRO license)» (рис. 2.16), задати шлях збереження та ім'я моделі (рис. 2.17).

## 2.4 Технології розробки фізичного прототипу

### 2.4.1 Основи 3D друкування

Створення якісного фізичного 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 на задані [28].

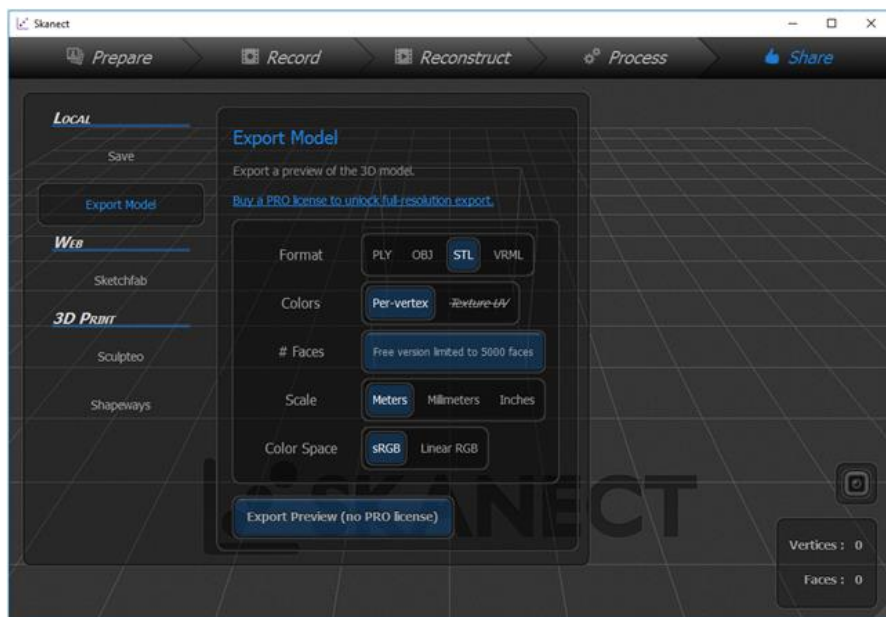


Рисунок 2.16 – Експорт 3D моделі

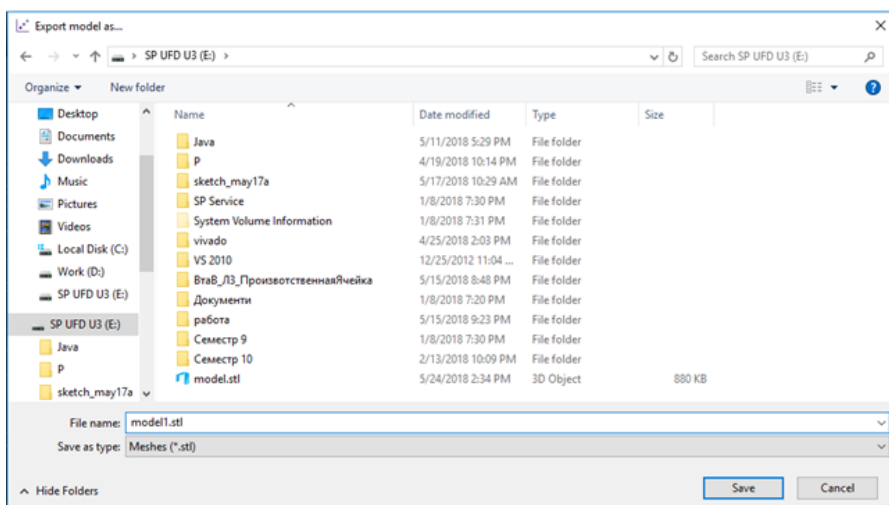


Рисунок 2.17 – Збереження 3D моделі

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 працюючої моделі 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 одиниць, 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].

До переваг використання швидкого прототипування відносяться:

- скорочення тривалості технологічної 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].

Таким чином, основні 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 друкування, деякі з яких представлено в таблиці 2.3.

Найбільш популярною є технологія пошарового накладення розплавленої полімерної нитки (FDM – Fused Deposition Modeling), схему якої наведено на рисунку 2.18 [28].

Таблиця 2.3 – Основні методи адитивного виробництва

Метод	Технологія	Матеріали
Екструзійний	Моделювання шляхом пошарового наплавлення (FDM або FFF)	Термопластики (полілактид (PLA), акрилонітрилбутадієнс-тирол (ABS), ін.)
Дротовий	Виробництво довільних форм електронно-променевою плавкою (EBF <sub>3</sub> )	Практично будь-які металеві сплави
Порошковий	Пряме лазерне спікання металів (DMLS)	Практично будь-які металеві сплави
	Електронно-променево плавлення (EBM)	Титанові сплави
	Вибіркове лазерне плавлення (SLM)	Титанові та кобальт-хромові сплави, нержавіюча сталь, алюміній
	Вибіркове теплове спікання (SHS)	Порошкові термопластики
	Вибіркове лазерне спікання (SLS)	Термопластик, металеві та керамічні порошки
Струменевий	Струменевий трьохвимірний друк (3DP)	Гіпс, пластик, металеві порошки, пісчані суміші
Ламінування	Виготовлення об'єктів методом ламінування (LOM)	Папір, металева фольга, пластикова плівка
Полімеризація	Стереолітографія (SLA)	Фотополімери
	Цифрова світлодіодна проєкція (DLP)	Фотополімери

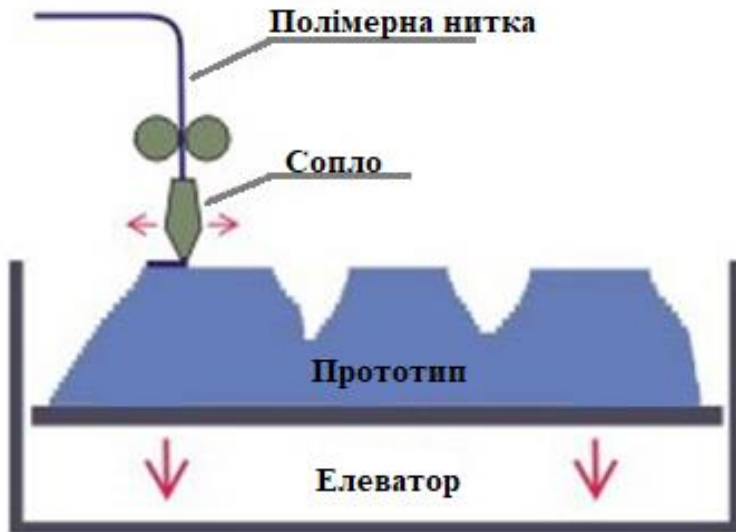


Рисунок 2.18 – Схема пошарового накладення розплавленої полімерної нитки [28]

Таким чином, технологія друку тривимірного об'єкту дає можливість відмовитися від ручної праці в повній мірі. В результаті застосування 3D технології, немає потреби в креслениках 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 проектуванні. А на всю роботу буде витрачено кілька годин, що у часовому еквіваленті буде швидше в п-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 CAD-системи (наприклад Creo, AutoCAD) і зберегти в спеціальному форматі STL (рис. 2.19). Потім файл STL завантажується в програму підготовки для принтера, наприклад, Cura або Slic3r.

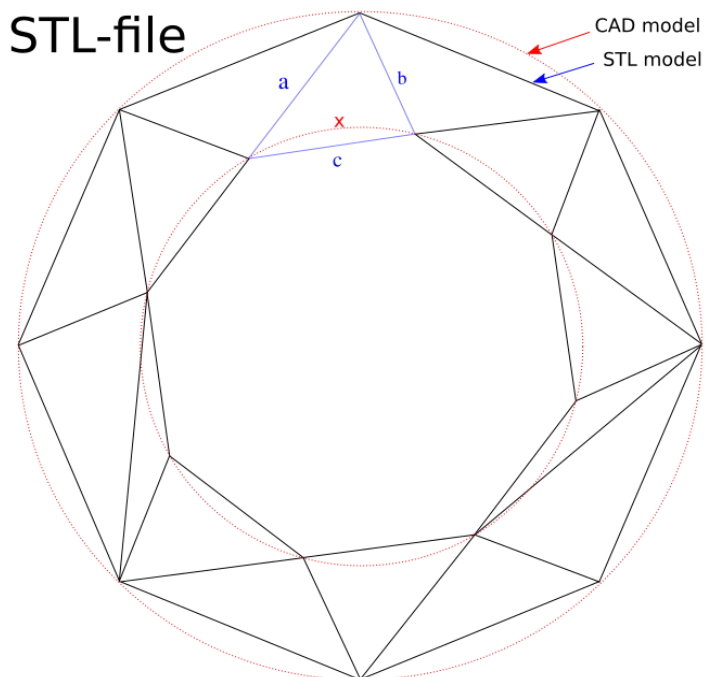


Рисунок 2.19 – Схема представлення форми в STL-файлі

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, оскільки 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 друкування [44].

Сам процес 3D друкування – 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 (рис. 2.20).

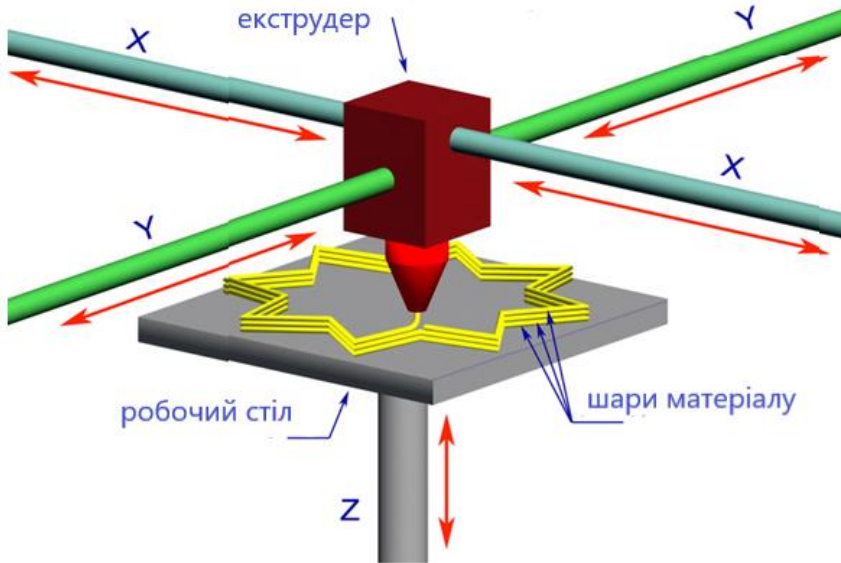


Рисунок 2.20 – Схема реалізації процесу 3D друкування

Цикли безперервно слідують один за другим: на перший шар матеріалу наноситься наступний, елеватор знову опускається і так доти, доки на робочому столі не з'явиться потрібна модель. Тобто 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].

Порівняльну характеристику декількох моделей 3D принтерів наведено в таблиці. 2.4. Порівняння декількох програм для підготовки до 3D друкування наведено в таблиці 2.5.

Таблиця 2.4 – Порівняльна таблиця 3D принтерів

№	Характеристика	Назва принтеру		
		3D принтер xyzprinting minimaker	3D принтер Leapfrog Creatr HS	3D принтер 3d systems trc-60
1	Діаметр сопла, мм	0,4	0,35	0,15
2	Технологія друку	FDM (FFF) наплавлення	Single Extruder:290x270x180 мм Dual Extruder:290x240x180 мм	HIWIN
3	Максимальна швидкість друку, мм/с	100	30-300	10-280
4	Робоча температура екструдера, °C	максимум 240	90	90
5	Кількість екструдерів	1	2	2
6	Область друку, мм	150x150x150	230x270x200	600x600x600
7	Додаткова інформація	Товщина шару: 0,1 – 0,4 мм; ПЗ: XYZware; Формат файлу stl, XYZ Format (.3w), .3m	Товщина шару: 0,02 – 0,35 мм	Товщина шару: від 10 до 400 мкм в залежності від діаметра сопла
8	Матеріал	PLA	PLA, ABS, Н IPS, Wood, Carbon, Flex, PET-G, PVA та інші	ABS, PLA, ABS+, CoPET, PET, Nylon, PVA, ABS+Flex, HIPS, Flex, PC та інші
9	Розміри, мм	400x336x362	633x486x523	1000x830x840

Таблиця 2.5 – Порівняльна таблиця програмних засобів для 3D друкування

№ з/п	Назва	Назва програмних засобів		
		Simplify3D	Kissliser 2	Cura
1	Швидкість	+	–	–
2	Створення підтримок	+	–	–
3	Зміна температури під час друкування	–	–	+
4	Розрахунок ширини екструзії	+	–	–
5	Друкування в декілька процесів	+	–	–
6	Повнофункціональний інтерфейс управління принтером	+	–	+
7	Інтерактивний попередній перегляд процесу друкування	+	+	+
8	Простота в користуванні	+	+	+
9	Сумісність	–	+	+

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, оскільки 3D друкування дозволяє отримати протези і коронки значно швидше ніж за класичною технологією виробництва [45].

Медичні тривимірні моделі (рис. 2.21) 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.



Рисунок 2.21 – Прототип зубних коронок, надрукований на 3D принтері [45]

#### 2.4.2 Підготовка моделі до 3D друкування

Першим кроком 3D друкування є підготовка 3D моделі. Не кожен 3D модель можна роздрукувати на конкретному принтері. В деяких випадках треба збільшити товщину стінки, враховувати похибки друку та змінювати діаметри отворів.

У процесі друку модель «ріжеться» на шари. Кожен шар складається з периметру та наповнення. Нижче наведено основні особливості, які треба враховувати під час підготовки 3D моделі до друкування 3D принтері.

Об'єкти, що перетинаються. Границі і ребра, що перетинаються, можуть привести до виникнення артефактів (рис. 2.22). Тому, якщо модель складається з декількох об'єктів, то їх необхідно об'єднати в один.

Плоска основа. Бажане, але необов'язкове правило. Плоска основа допоможе моделі краще триматися на столі принтера. Якщо модель відклеїться (цей процес називають деламінація), то порушиться геометрія основи моделі, а це може призвести до зміщення координат XY, що ще гірше. Якщо модель не має плоскої основи або площа основи замала, то її друкують на рафті – надрукованій підкладці. Рафт псує поверхню моделі, з якою стикається. Тому при можливості краще обійтися без нього.

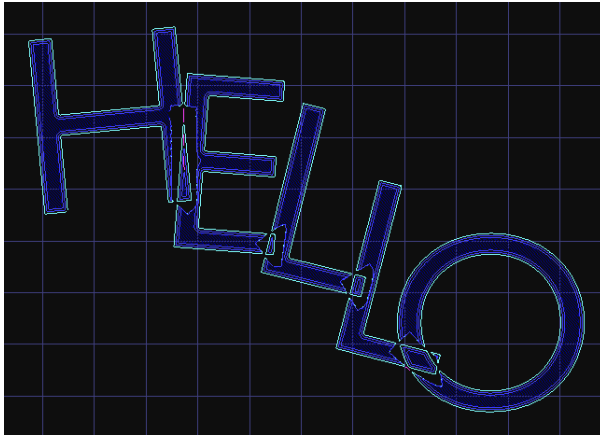


Рисунок 2.22 – Артефакти об'єктів, що перетинаються

Товщина стінок. Товщина стінки повинна дорівнювати або бути товще ніж діаметр сопла. Інакше принтер просто не зможе її надрукувати. Товщина стінки залежить від того, скільки периметрів буде друкуватися. При 3 периметрах і соплі 0,5 мм товщина стінок повинна бути від 0.5, 1, 1.5, 2, 2.5, 3 мм, а понад – може бути будь-якою. Тобто товщина стінки повинна бути кратна діаметру сопла, якщо вона менша за  $N \cdot d$ , де  $N$  – кількість периметрів,  $d$  – діаметр сопла.

Мінімум нависаючих елементів. Для кожного нависаючого елемента необхідна підтримуюча конструкція – підтримка (рис. 2.23).

Чим менше нависають елементів, тим менше підтримок потрібно, тим менше потрібно витратити матеріалу і часу друку на них і тим швидше буде друк. Крім того, підтримка псує поверхню, дотичну з нею. Можна надрукувати без підтримок стінки, які мають кут нахилу не більше 70 градусів.

Точність. Точність по осях  $X$  та  $Y$  залежить від люфтів, жорсткості конструкції, ременів, тобто від механіки принтера. І становить приблизно 0.3 мм у непрофесійних принтерів.

Точність по осі  $Z$  визначається висотою шару (0.1-0.4 мм). Звідси і висота моделі буде кратна висоті шару.

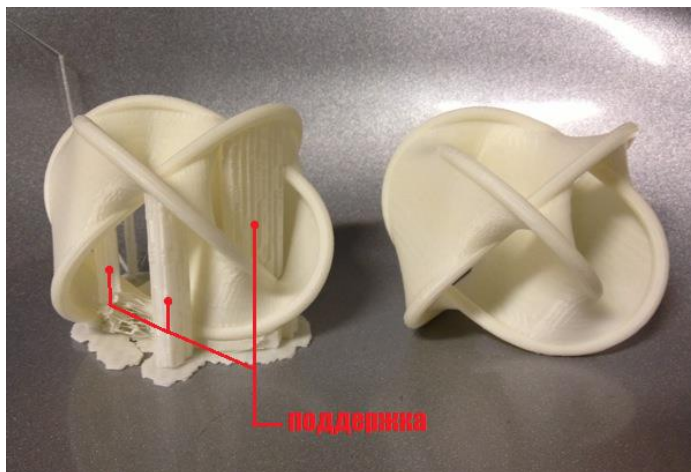


Рисунок 2.23 – Підтримки 3D моделі

Також необхідно враховувати, що після охолодження матеріал просідає, а разом з цим змінюється геометрія об'єкта.

Існує ще програмна сторона проблеми – не кожна програма-слайсер правильно обробляє внутрішні розміри, тому діаметр отворів краще збільшити на 0.1-0.2 мм.

Дрібні деталі. Дрібні деталі досить складно відтворюються на FDM принтері. Їх взагалі неможливо відтворити, якщо вони менші, ніж діаметр сопла. Крім того при обробці поверхні дрібні деталі стануть менш помітні або зникнуть зовсім.

Вузькі місця. Вузькі місця дуже складно обробляти. По можливості необхідно уникати таких місць, що потребують обробки, до яких неможливо підібратися. Звичайно, можна обробляти поверхню в ванні з розчинником, але тоді розплавляться дрібні елементи.

Великі моделі. При моделюванні необхідно враховувати максимально можливі габарити друку. У випадку якщо модель більша за ці габарити, то її необхідно розрізати, щоб надрукувати по частинам. Таким чином необхідно також передбачити з'єднання (наприклад «хвіст ластівки»).

### 2.4.3 Практичне використання 3D принтеру LeapFrog Creatr HS та програмного забезпечення Simplify3D

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 часто лежить принцип пошарового створення (виروشуння) твердого об'єкта [46].

Для створення 3D моделі можна використати 3D принтер LeapFrog Creatr HS (рис. 2.24). Характеристики пристрою наведено в таблиці 2.6 [47, 48].

Leapfrog Creatr HS являє собою потужний і надійний 3D принтер, що забезпечує високу точність і швидкість 3D друку до 300 мм/сек. Він є одним з найбільш швидких і точних приладів в своєму класі. Принтер має точність друку 50 мкм, що дозволяє виконувати високоточний 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. Принтер може бути оснащений як одним, так і двома екструдерами. Також варто відзначити велику область побудови, яка становить 270x260x180 мм, незалежно від кількості екструдерів. Все це дозволяє створювати великі вироби, або проводити 3D друкування для дрібносерійного виробництва [47].

Creatr HS оснащений платформою, що підігрівається, LCD-дисплеєм і використовує для друку різні види пластиків, такі як ABS / PLA / Нейлон / PVA / Laybrick та ін.

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. 2.25).

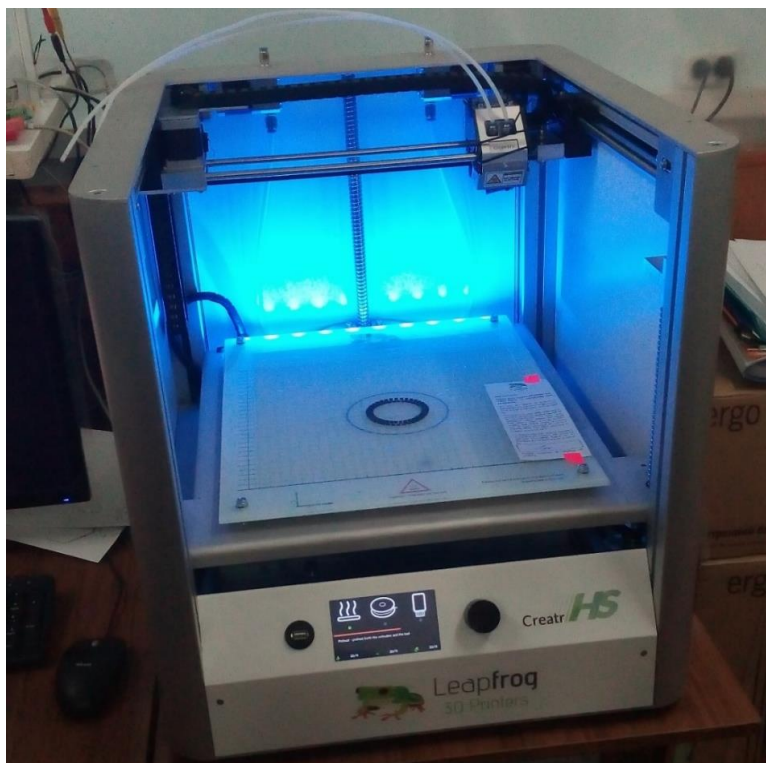


Рисунок 2.24 – 3D принтер Leapfrog Creatr HS

Таблиця 2.6 – Технічні характеристики Leapfrog Creatr HS

Характеристика	Значення
Розмір друку, мм	270*280*180
Об'єм друку, л	max 14,8
Сумісність з операційною системою	Windows, Mac OS, Linux
Програма керування	Repetier Host, Simplify 3D
Мова програми	Російська, англійська
Розміри принтера, мм	600*500*500
Вага принтера, кг	30

Продовження таблиці 2.6

Формат файлів друку	.STL .GCODE .OBJ
Робоча напруга, В	220
Потужність, Вт	до 400
Матеріал для друку	ABS; PLA; PVA; Laybrick; Nylon; PVA, HIPS
Діаметр матеріалу, мм	1,75
Точність друку (товщина шару), мкм	50-350
Точність позиціонування, мм	Вісь Z: 0.005, вісь XY: 0.0169
Кількість екструдерів	2 шт
Діаметр сопел, мм	0,35 (0,5 опціонально)
Тип друкованого столу	Найвний 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, мм / с	30-300 – при друкуванні, 30-400 – при переміщенні



Рисунок 2.25 – Елементи керування принтеру Creatr HS

Після створення 3D-моделі об'єкта за допомогою CAD-системи (наприклад Creo Parametric [49]), вона імпортується в програму підготовки до друкування.

Simplify3D – це платна програма, що свого часу замінила Repetier-Host і пакети Slic3r, дозволяючи імпортувати різні файли, змінювати масштаб моделі, відновлювати код моделі, створювати G-code і відправляти модель на друкування [50].

Simplify3D включає в себе алгоритми швидкої нарізки, які формують G-code навіть для найскладніших моделей. В програмі існує анімований попередній перегляд, який дозволяє побачити, як модель буде друкуватися шар за шаром. Ця функція дозволяє виправити або вдосконалити модель, перш ніж переходити до друку. Програма підходить для принтерів з подвійним екструдером та послідовного друку відразу декількох моделей, причому для кожної з них можна вибрати свої параметри друкування. Користувач повністю контролює кількість і нанесення підтримуючого матеріалу. У результаті час друку скорочується, як і витрата матеріалу. Параметри в межах однієї моделі можна змінювати, тобто збільшувати щільність заповнення окремих частин моделі, щоб вони вийшли міцніші. Висота шару в межах однієї моделі теж може відрізнятись. Також користувач може налаштовувати температуру і швидкість.

Для завантаження готової моделі для друку, потрібно в панелі <Models> натиснути на клавішу <Import> (рис. 2.26 «13»). Можна завантажити декілька моделей.

Для видалення моделі необхідно виділити цю модель і натиснути в цьому ж меню на <Remove> (рис. 2.26 «14»), або натиснути на клавіатурі клавішу <Delete>, обрана модель видалиться. Так само можна приховати або показати модель, для цього варто прибрати або поставити прапорець біля назви цієї моделі. Для розташування моделі по центру області друку треба натиснути <Center and Arrange> (рис. 2.26 «15»).

Для зміни ракурсу огляду потрібно натиснути ліву клавішу мишки і рухаючи мишкою міняти ракурс, або в головному меню зверху розгорнути вкладку <View> і обрати один з запропонованих, або в панелі інструментів, яка знаходиться праворуч, обрати один з чотирьох варіантів (рис. 2.26 «4»).

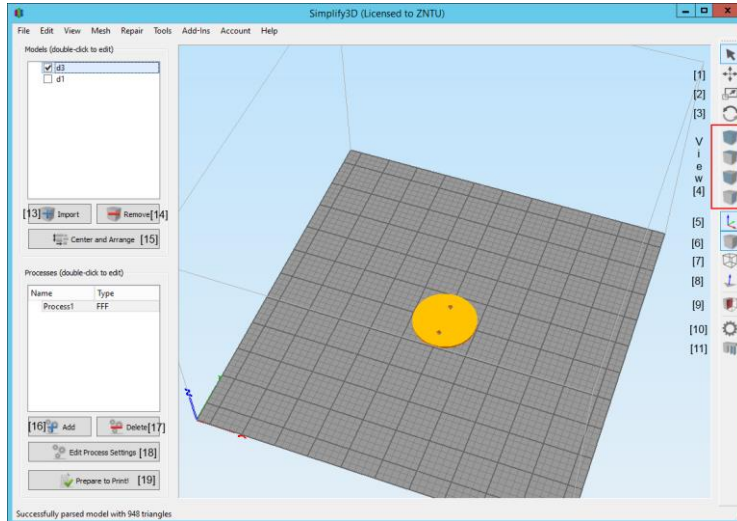


Рисунок 2.26 – Головне вікно Simplify3D

Якщо потрібно приховати систему координат, треба в панелі інструментів вибрати <Coordinate Axes> (рис. 2.26 «5»). Так само можна перейти в режим відображення тільки граней деталі, для цього в панелі інструментів вибрати <Wireframe> (рис. 2.26 «7»). Щоб перейти назад в режим відображення твердотільної моделі потрібно в панелі інструментів вибрати <Solid Model> (рис. 2.26 «6»).

<Show Normals> – показати нормалі (рис. 2.26 «8»), результат на рисунку 2.27.

<Cross Section> – налаштувати перетини (рис. 2.26 «9»), результат на рисунку 2.28.

Для розміщення моделі в області стола, в якій буде проводитися друк, необхідно затиснути клавішу <Ctrl>, затиснути ліву клавішу миші на деталі і перемістити її на потрібне місце, після чого відпустити клавіші, або в панелі інструментів вибрати <Translate Models> (рис. 2.26 «1»).

Для зміни розміру деталі в панелі інструментів вибрати <Scale Models> (рис. 2.26 «2»). Щоб повернути деталь потрібно в панелі інструментів вибрати <Rotate Models> (рис. 2.26 «3»).

Для зміни положення деталі потрібно в головному меню

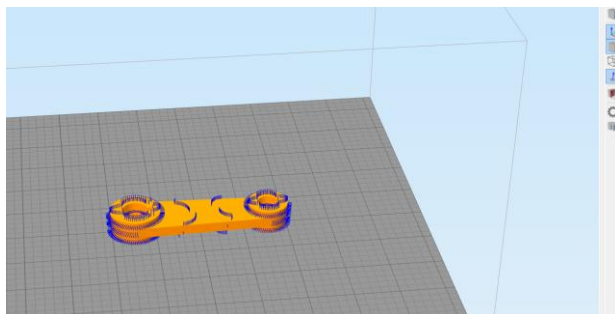


Рисунок 2.27 – Опція «Show Normals»

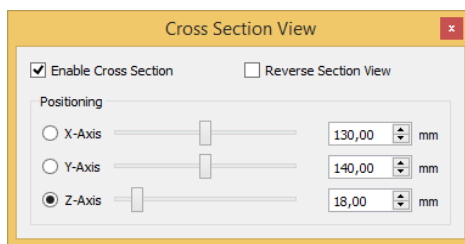


Рисунок 2.28 – Меню опції «Cross Section»

зверху у вкладці <Edit> вибрати <Place Surface on Bed> (рис. 2.26), далі натиснути клавіші <Ctrl> + <L>, після чого вибрати, на яку грань потрібно покласти деталь.

Для побудови деталі з опорною конструкцією потрібно в панелі інструментів вибрати <Customize Support Structures> (рис. 2.26 «11»), у вікні, що відкрилося (рис. 2.29), натиснути на <Generate Automatic Support>, також у полях <Support Type> – вибрати тип опори, <Support Pillar Resolution> – ширина опори на вигинах, <Max Overhang Angle> – максимальний кут опори.

Можна зберегти налаштування допоміжної конструкції, натиснувши на <Export Support>, а коли знову буде потрібна опорна конструкція, можна її завантажити, натиснувши <Import Support>. Клавіші <Clear All Support> – очистити всі параметри для опори, <Done> – погодитися з налаштуваннями та повернутися назад.

Для додавання налаштувань друку, потрібно в меню <Processes> натиснути <Add> (рис. 2.26 «16»), у вікні, що з'явилося (рис. 2.30) обрати налаштування:

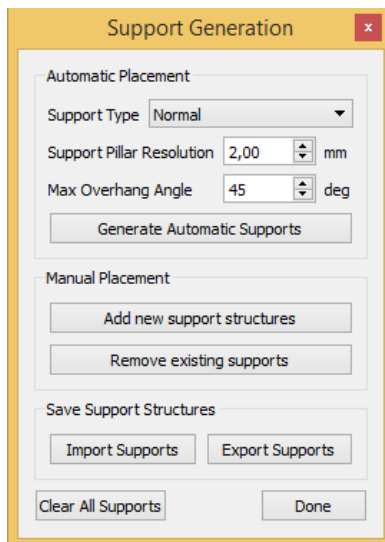


Рисунок 2.29 – Меню опції «Support Generation»

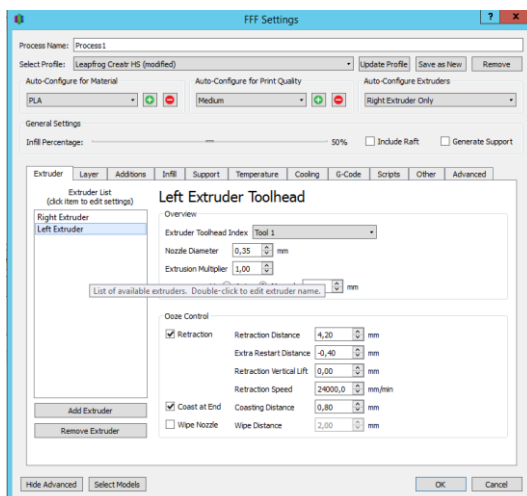


Рисунок 2.30 – Вікно параметрів друкування

- Process Name – ім'я налаштувань для друкування;
- Select Profile – вибрати профіль друкування;
- Auto-Configure Material – вибрати тип матеріалу;

- Auto-Configure for Print Quality – вибрати швидкість друкування;

- Auto-Configure Extruders – вибрати, яким екструдером буде виконуватись друк моделі;

- General Settings – задати рівень заповнення, <Include Raft> включити обведення і <Generate Support> – включити опорну конструкцію.

У нижній частині вікна знаходяться налаштування профілю друкування принтера: <Hide Advanced> – додаткові налаштування; <Select Models> – вибрати моделі.

Вкладка Extruder (Екструдер) (рис. 2.30) містить: Extruder list – вибір екструдера. Клавіші <Add Extruder> – додати екструдер, та <Remove Extruder> – видалити екструдер.

Область «Overview» містить:

- Extruder Toolhead Index – вибрати індекс екструдера;

- Nozzle Diameter – вказати діаметр сопел екструдера;

- Extruder Multiplier – параметр для компенсації жорсткості сировинного прутка в більшу або меншу сторону;

- Extruder Width – це товщина доріжки, викладеної в процесі екструзії.

Область «Ooze Control» містить:

- Retraction Distance – дистанція відводу;

- Extra Restart Distance – контролює, скільки додаткових екструзій визивається, коли екструдер скидається після втягування;

- Retraction Vertical Lift – піднімає головку екструдера на невелику відстань протягом кожного відводу, і знижує його відстань, щоб розпочати наступний сегмент;

- Retraction Speed – швидкість відводу;

- Coasting Distance – вимикає екструдер на заданій відстані;

- Wipe Distance – дистанція для очищення сопла;

- Retraction – відведення матеріалу;

- Coast at End – злив по закінченню друку;

- Wipe Nozzle – очистка сопла.

Область «Layer Settings» вкладки «Layer» (рис.2.31):

- Primary Extruder – яким екструдером буде проводиться друкування шарів;

- Primary Layer Height – визначає товщину кожного шару;

- Top Solid Layers – кількість верхніх шарів моделі;
- Bottom Solid Layer – кількість нижніх шарів моделі;
- Outline/Perimeter Shells – кількість зовнішніх стінок моделі;
- Outline Direction – визначає периметри, <Outside-In> – друкує спочатку внутрішні стінки моделі, а після зовнішні, та може дати трохи краще якість поверхні, але <Inside-Out> (навпаки, спочатку зовнішні потім внутрішні стінки моделі) є корисним при друку моделей з заходами, щоб забезпечити адгезію (прилипання моделі до поверхні стола);

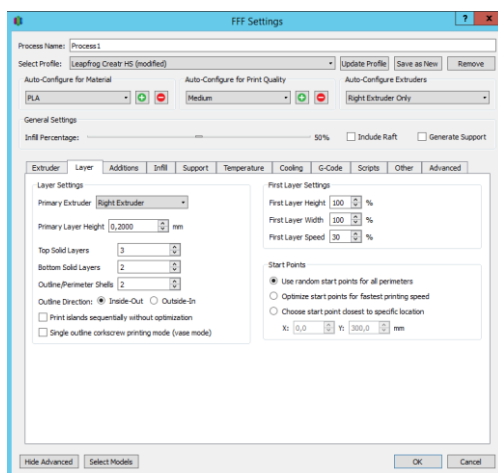


Рисунок 2.31 – Вкладка «Layer»

- Print islands sequentially without optimization – опція змінює порядок, в якому друкує "острова", або ізольовані райони конкретного шару;

- Single outline corkscrew printing mode (vase mode) – корисно для друкування одного контурного об'єкта без заповнення (виводить у вигляді однієї, поступово зростаючої спіралі).

Область «First Layer Settings» (рис. 2.31):

- First Layer Height – може бути використаний для зниження висоти першого шару, в ідеалі має бути 100%;

- First Layer Width – може бути використаний для зниження ширини першого шару;

–First Layer Speed – може бути використаний для зниження швидкості друку першого шару.

Область «Start Points» (рис. 2.31):

–Use random start point for all perimeters – старт друкування з випадкової точки;

–Optimize start point for fastest printing speed – оптимальна стартова точка для друкування;

–Chose start point closest to specific location – старт друкування із зазначених координат X та Y.

Вкладка «Additions», область «Use Skirt/Brim» (рис.2.32):

–Skirt Extruder – вибір екструдера для друку контуру моделі;

–Skirt Layers – кількість шарів контуру;

–Skirt Offset from Part – відстань від моделі до контуру;

–Skirt Outlines – кількість периметрів контуру.

Область «Use Raft» (рис.2.32):

–Raft Extruder – вибір екструдера для друку платформи, на якій розміститься модель;

–Raft Layers – кількість шарів платформи;

–Raft Offset from Part – відстань на яку платформа буде виступати за межі моделі;

–Separation Distance – розділова відстань між платформою та моделлю;

–Raft Infill – заповнення платформи;

–Disable raft base layer – відключити заповнення першого шару.

Область «Use Prime Pillar» (рис.2.32):

–Prime Pillar Extruder – вибір екструдера для друку основного стовпа;

– Pillar Width – ширина основного стовпа;

– Pillar Location – розміщення основного стовпа;

– Speed Multiplier – швидкість друку стовпа.

Область «Use Ooze Shield» (рис.2.32):

–Ooze Shield Extruder – вибір екструдера для друку щита;

–Offset from Part – відстань розміщення стінки щита від моделі;

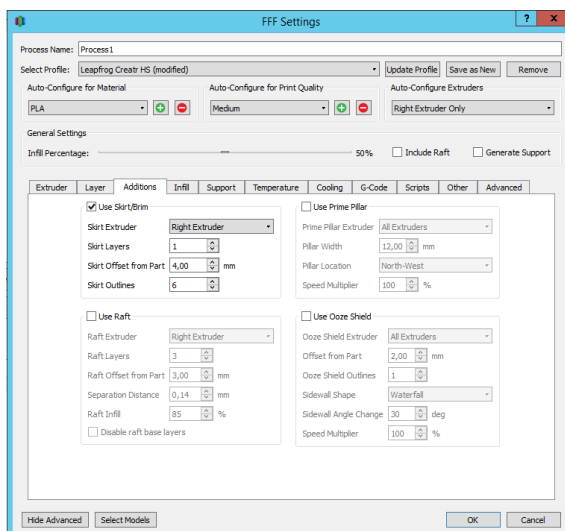


Рисунок 2.32 – Вкладка «Additions»

- Ooze Shield Outline – кількість периметрів щита;
  - Sidewall Share – форма щита;
  - Sidewall Angle Change – максимальний кут нахилу стінок щита;
  - Speed Multiplier – швидкість друку щита.
- Область «General» вкладки Infill (Заповнення) (рис.2.33):
- Infill Extruder – вибір екструдера для друку заповнення;
  - Internal Fill Pattern – форма внутрішнього заповнення моделі;
  - Extruder Fill Pattern – форма заповнення дна та кришки моделі;
  - Interior Fill Percentage – відсоток щільності заповнення внутрішнього шару;
  - Outline Overlap – відсоток перекриття внутрішнього шару стінок моделі;
  - Infill Extrusion Width – висота шару заповнення;
  - Minimum Infill Length – мінімальна довжина заповнення;
  - Print Sparse Infill Every – друк заповнення через N кількість шарів;

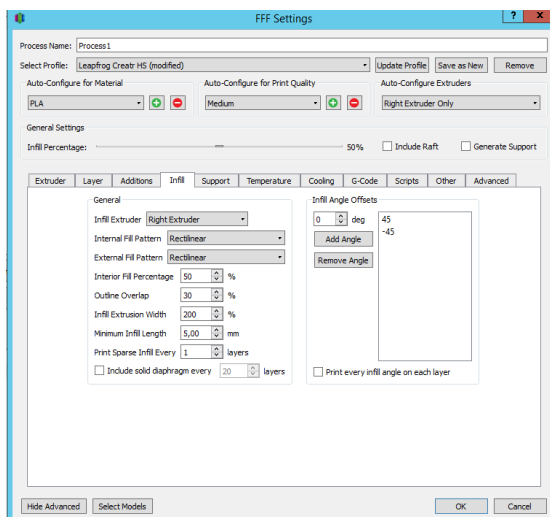


Рисунок 2.33 – Вкладка «Infill»

–Include solid diaphragm every – включити тверде заповнення через N кількість шарів;

Область «Infill Angle Offsets» (рис.2.33):

–<Add Angle> – додати кут заповнення;

–<Remove Angle> – видалити кут заповнення;

–Print every infill angle on each layer – друкувати на кожному шарі заповнення.

Вкладка Support (Опорна конструкція) (рис. 2.34) також містить декілька областей. Область «Support Material Generation»:

–Generate Support Material – включити опору;

–Support Extruder – вибрати екструдер для друку опори;

–Support Infill Percentage – співвідношення щільності друку основи опори у відсотках;

–Extra Inflation Distance – на скільки опори виходять за межі моделі;

–Dense Support Layers – щільність опори безпосередньо під шарами моделі, що нависають;

–Dense Infill Percentage – щільність заповнення опори під шарами моделі у відсотках;

–Print Support Every – друк опори через N кількість шарів.

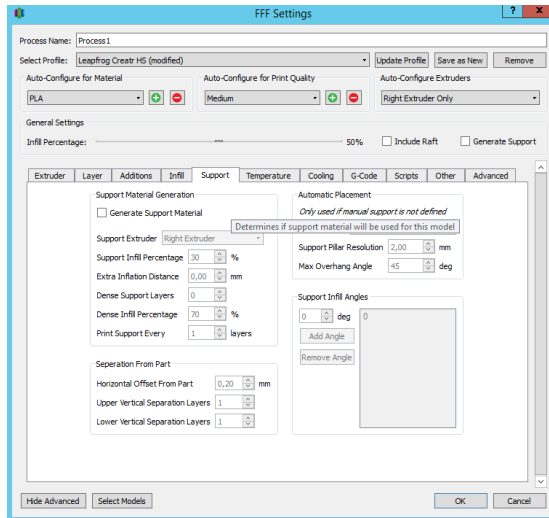


Рисунок 2.34 – Вкладка «Support»

Область «Separation From Part» (рис. 2.34):

- Horizontal Offset From Part – зсув по горизонталі від моделі;
- Upper Vertical Separation Layers – кількість верхніх розділових шарів;
- Lower Vertical Separation Layers – кількість нижніх розділових шарів.

Область «Automatic Placement»:

- Support Pillar Resolution – ширина опори на вигинах;
- Max Overhang Angle – максимальний кут опори.

Область «Support Infill Angle»:

- <Add Angle> – додати кут опори;
- <Remove Angle> – видалити кут опори.

Вкладку Temperature (Температура) наведено на рисунку 2.35. Область «Temperature Controller List»:

- <Add Temperature Controller> – додати налаштування температури;
- <Remove Temperature Controller> – видалити налаштування температури.

Область «Overview» (рис. 2.35):

- Temperature Identifier – ідентифікатор температури;

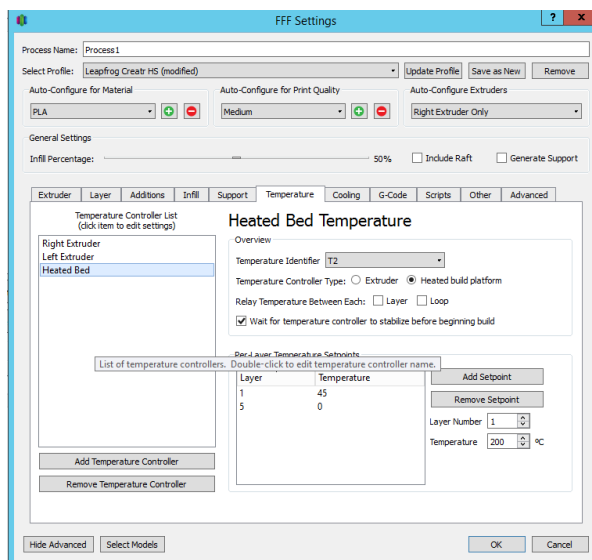


Рисунок 2.35 – Вкладка «Temperature»

–Temperature Controller Type – тип контролера, <Extruder>- екструдер, або <Heated build platform>- робоча поверхня;

–Relay Temperature Between Each – передача показників температури через кожний, <Layer>- шар і <Loop>- цикл;

–Wait for temperature controller to stabilize before beginning build - почекати регулятор температур перед початком друку;

Область «Per-Layer Temperature Setpoints» (рис.2.35):

–<Add Setpoint> - додати налаштування температури друку шару;

–<Remove Setpoint> – видалити налаштування;

–Layer Number – номер шару;

–Temperature – значення температури в Цельсіях.

Вкладка Cooling (Охолодження) представлена на рисунку 2.36. Область «Per-Layer Fan Controls»:

–<Add Setpoint> - додати номер шару та швидкість вентилятора;

–<Remove Setpoint> - видалити налаштування шару;

–Layer Number - номер шару;

–Fan Speed - швидкість обертів вентилятора.

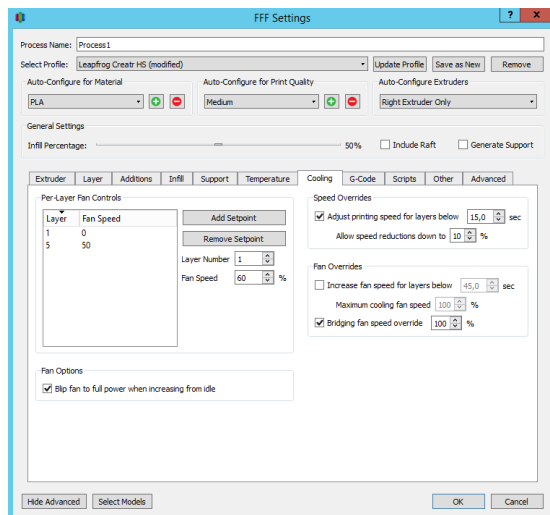


Рисунок 2.36 – Вкладка «Cooling»

Область «Speed Overrides»: Область «Fan Options»: <Blip fan to full power when increasing from idle> - максимальний обдув при переході з холостого ходу.

–<Adjust printing speed for layers bellow> - зменшення швидкості друку нижнього шару;

–Allow speed reductions down to - зниження швидкості до заданого.

Область «Fan Overrides» (рис. 2.36):

–<Increase fan speed for layers below> - збільшити швидкість обертання для шарів швидкість друку яких нижча за вказану;

–Maximum cooling fan speed - максимальна швидкість вентилятора;

–<Bridging fan speed override> - швидкість обдування під час друку мостів.

Вкладка: G-Code(G-код), Область «G-Cod Options» (рис. 2.37):

–5D firmware (include E-dimension) - прошивка на 5D, з E-виміром, 3 осі та 2 екструдера;

–Relative extrusion distance - відносна відстань екструзії;

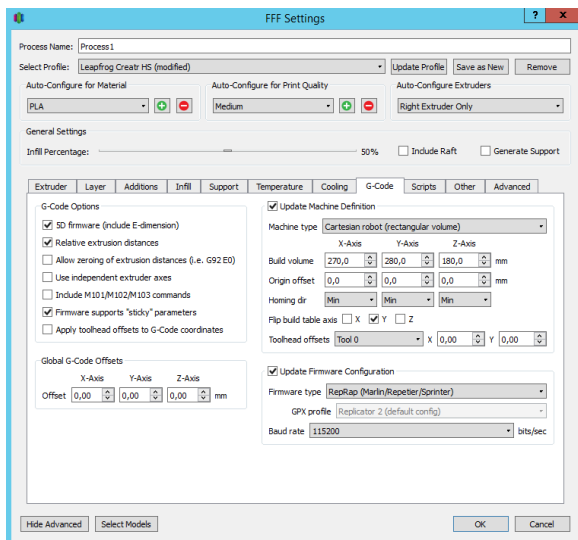


Рисунок 2.37 – Вкладка «G-Code»

- Allow zeroing of extrusion distance - дозволити обнулення відстані екструзії;
  - Use independent extruder axes – використовувати незалежні осі екструдера;
  - Include M101/M102/M103 commands - включити команди M101/M102/M103;
  - Firmware support “sticky” parameters - опора з “липкими” параметрами;
  - Apply toolhead offsets to G-Code coordinates - застосувати змішання до координат G-коду на робочій поверхні;
- Область «Global G-Code Offsets»: Offset - зсув по координатам X-Y-Z.

Область «Update Machine Definition»:

- включити оновлення визначень машини;
- Machine type - тип машини;
- Build volume - координати друкованої області (X-Y-Z);
- Origin offset - зсув по координатам X-Y-Z;
- Homing dir - координати робочої позиції;
- Flip build table axis – перевертає вісь таблиці побудови;

–Toolhead offsets - робоча поверхня і її координати;

Область «Update Firmware Configuration»:

–Firmware type - тип мікропрограми;

–GPX profile - GPX профіль підпрограми;

–Baud rate - швидкість передачі.

Вкладка: Script (Скрипти) (рис. 2.38) містить ще декілька вкладок:

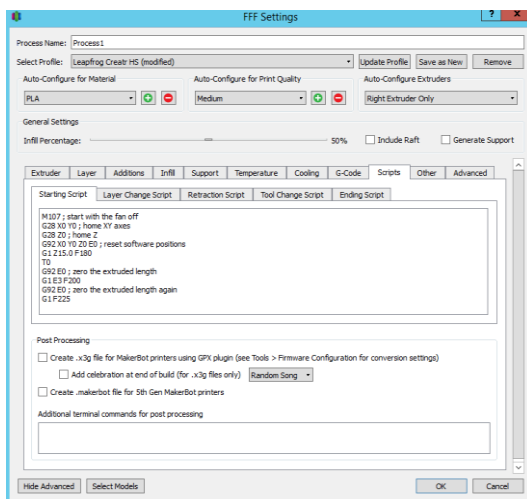


Рисунок 2.38 – Вкладка «Scripts»

–<Starting Script> - стартовий скрипт при друкуванні;

–<Layer Change Script > - скрипт при друкуванні кожного шару;

–<Retraction Script > - скрипт при переміщенні екструдера при друкуванні;

–<Tool Change Script > - скрипт при зміні екструдера;

–<Ending Script > - скрипт по закінченню друкування.

Вкладка: Other(Інші), область «Speed» (рис.2.39):

–Default Printing Speed - стандартна швидкість друку;

–Outline Underspeed - використовується, щоб уповільнити периметри у відсотках від швидкості друку за замовчуванням;

–Solid Infill Underspeed – швидкість друку твердого заповнення;

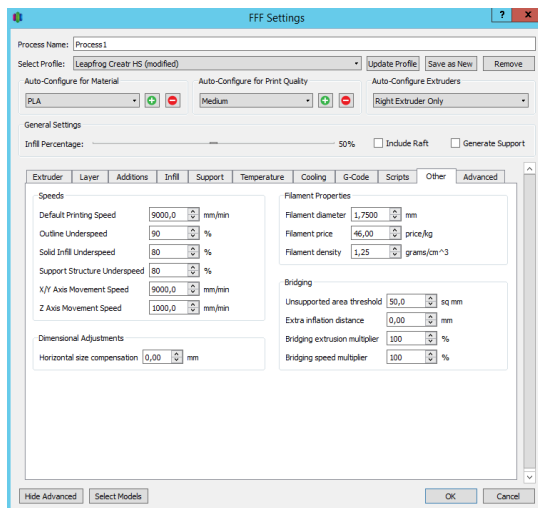


Рисунок 2.39 – Вкладка «Other»

–Support Structure Underspeed – швидкість друку опорної конструкції;

–X/Y Axis Movement Speed - швидкість руху по осях X и Y;

–Z Axis Movement speed - швидкість руху по осі Z.

Область «Dimensional Adjustments» (рис. 2.16): Horizontal size compensation - горизонтальна компенсація розміру у випадку усадки пластика.

Область «Filament Properties»:

–Filament diameter - діаметр прутка;

–Filament price - ціна за 1 кілограм;

–Filament density - щільність прутка.

Область «Bridging»:

–Unsupported area threshold - порогова площа нависаючої частини моделі без опори;

–Extra inflation distance - додаткова відстань, яка вказує наскільки міст заходить на площадки з яких він будується;

–Bridging extrusion multiplier - множник екструзії для друку мостів;

–Bridging speed multiplier – множник швидкості друку моста.

Вкладка: Advanced (Додатково) (рис. 2.40):

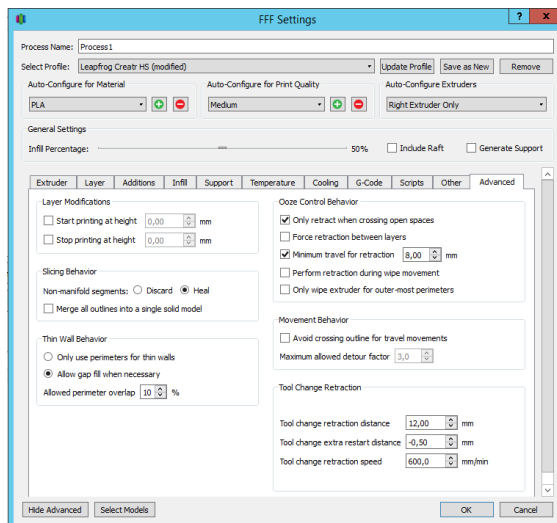


Рисунок 2.40 – Вкладка «Advanced»

Область «Layer Modifications»:

–<Start printing at height> - початок друку на висоті;

–<Stop printing at height> - закінчення друку на висоті.

Область «Slicing Behavior»:

–Non-manifold segments - якщо знайдена помилка в моделі,  
<Discard>- пропустити, або <Heal> - вирішити;

–<Merge all outlines into a single solid model> - об'єднати всі контури в одній суцільній моделі.

Область «Thin Wall Behavior»:

–<Only use perimeters for thin walls> - використовувати параметри для тонких стінок;

–<Allow gap fill when necessary> - дозволити заливку при розриві.

–Allowed perimeter overlap - периметр перекриття стінок.

Область «Ooze Control Behavior»:

–Only retract when crossing open spaces - відмовитися від втягування матеріалу при відводі сопла в рамках однієї моделі;

–Force retraction between layers - відстань втягування матеріалу між шарами;

–Minimum travel for retraction – мінімальна відстань для включення відводу;

–Perform retraction during wipe movement - виконувати підтягування при русі;

–Only wipe extruder for outer-most perimeters - згладжування місця відведення екструдера для зовнішніх периметрів.

Область «Movement Behavior»:

–Avoid crossing outline for travel movements - уникати перетинання при русі;

–Maximum allowed detour factor - максимально допустима кількість відхилень від початкового прямого шляху.

Область «Tool Change Retraction»:

–Tool change retraction distance - відстань для відведення матеріалу під час переключення між екструдерами;

–Tool change extra restart distance - додатковий відвід матеріалу при зміні інструменту;

–Tool change retraction speed - швидкість відведення інструмента при зміні екструдерів [51].

## 2.5 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 апаратне та ПЗ, яке може використовуватись для прототипування кіберпротезу.

## 2.6 Контрольні питання

1. Назвіть критерії оцінки якості ефективного прототипу.
2. Які типи прототипування існують? Назвіть переваги та недоліки віртуального та фізичного прототипів?

3. З яких етапів складається процес створення прототипу? Назвіть основні стадії методу комплексного прототипування.
4. Які програми схемотехнічного моделювання та програми-емулятори використовуються для прототипування електронної частини кіберфізичних систем?
5. Дайте загальну характеристику універсальних та спеціалізованих CAD-систем для створення віртуальних прототипів біомедичних пристроїв та конструкцій.
6. Які переваги та недоліки хмарних САПР?
7. Що таке цифровий двійник?
8. В чому полягає принцип роботи 3-D сканера? Які базові типи сканерів існують?
9. Дайте характеристику існуючих технологій сканування.
10. Опишіть особливості використання 3D сканерів та 3D принтерів в біоінженерії та медицині.
11. Що таке RP-технологія? Які переваги даної технології? Дайте загальну характеристику технологій 3D друкування.
12. Які особливості треба врахувати при підготовці 3D моделі до друкування?

## 2.7 Практичні завдання

*Завдання 1.* Виконайте сканування здорової кінцівки людини та обробку отриманої моделі за допомогою сканера Gotcha, ПЗ Skanect за методикою, викладеною в п. 2.3.4.

*Завдання 2.* Ознайомтеся з будовою 3D принтера LeapFrog Creatr HS та функціональними можливостями програми Simplify3D.

*Завдання 3.* Створіть 3D модель простої деталі, виконайте підготовку до друкування згідно з п.2.4.2 та виведіть її на друк згідно з п. 2.4.3.

## 3 РОЗРОБКА ПРОТОТИПУ КІБЕРПРОТЕЗУ ВЕРХНЬОЇ КІНЦІВКИ ЛЮДИНИ

### 3.1 Розробка механічної частини прототипу кіберпротезу

Основні стадії та етапи інформаційної технології створення механічної частини кіберпротезу представлено на рисунку 3.1.

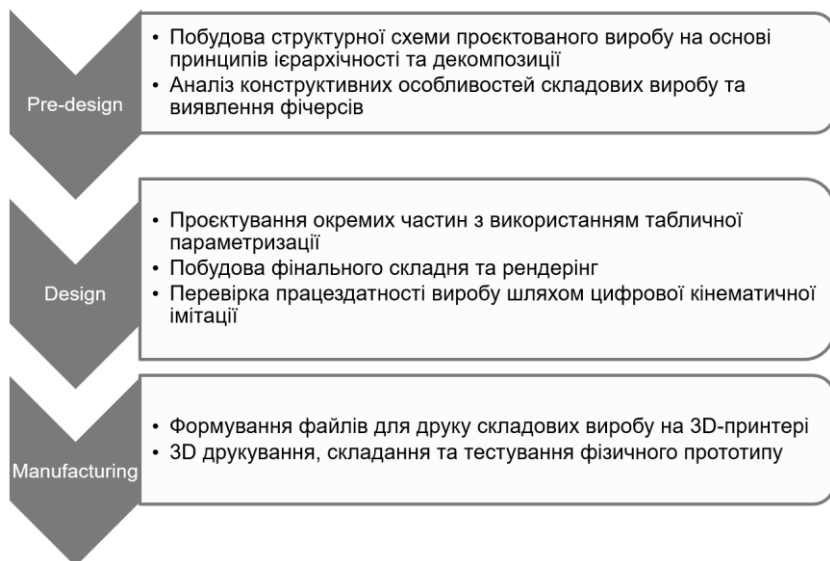


Рисунок 3.1 – Основні етапи інформаційної технології створення прототипу механічної частини кіберпротезу

Структурна схема проєктованого кіберпротезу представлена на рисунку 3.2. Вона містить окремі деталі (пальці, долоня, основа великого пальця) та складальну одиницю долоня+основа.

#### 3.1.1 Параметричне моделювання при проєктуванні

Для побудови параметричної моделі прототипу кіберпротезу в середовищі Creo Parametric Educational Edition 4.0 параметри кожного пальця необхідно представити у вигляді таблиці 3.1.

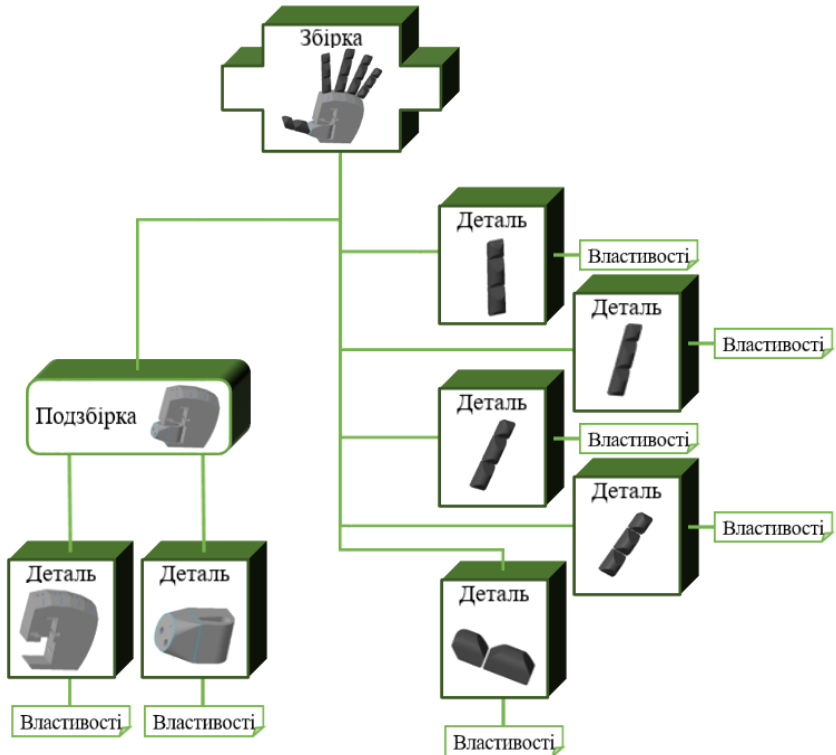


Рисунок 3.2 – Структурна схема проектованого виробу

В таблиці наведено наступні параметри для кожного пальця:

- 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 проектувальником самостійно, виходячи з габаритів необхідного пальця.

Таблиця 3.1 – Параметри пальців прототипу кіберфізичного протезу

Назва параметру 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	21,33
F	26	28,5	26	19,35	28,33

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:

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

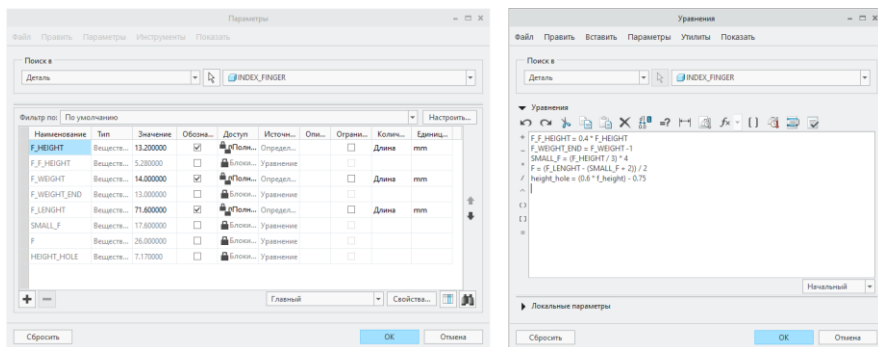
$$F\_WIDTH\_END = F\_WIDTH - 1 \text{ мм};$$

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

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

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

Параметри моделі пальця (рис. 3.3, а) і рівняння (рис. 3.3, б) можна задати у вкладці «Інструменти» для того, щоб полегшити введення потрібних значень при створенні моделей інших пальців.



а

а – параметри; б – рівняння

Рисунок 3.3 – Вікна заданих параметрів і рівнянь вказівного пальця

### 3.1.2 Розробка віртуального прототипу конструктиву кіберпротезу

Для побудови 3D моделей компонентів конструктиву будуть використовуватися такі проектні операції 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, проектуючи двовимірний перетин на зазначеній по нормалі до площини ескізу відстані;

- «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.

Після визначення параметрів, починаємо будувати модель пальця. Для цього створимо файл моделі вказівного пальця, типу «Деталь», підтипу «Тверде тіло» з шаблоном за замовчуванням і назвемо його «index\_finger».

Для початку створюємо основну форму пальця за допомогою операції «Сполучити». Для цього нам потрібно накреслити перший перетин, яке зображено на рисунку 3.4, після чого додаємо другий перетин, замінивши F\_WEIGHT на F\_WEIGHT\_END яке

розташоване паралельно першого перетину на відстані, що дорівнює параметру  $F\_LENGHT$ . Завершивши сполучення, ми отримуємо основну форму пальця, звужену до кінця (рис 3.5).

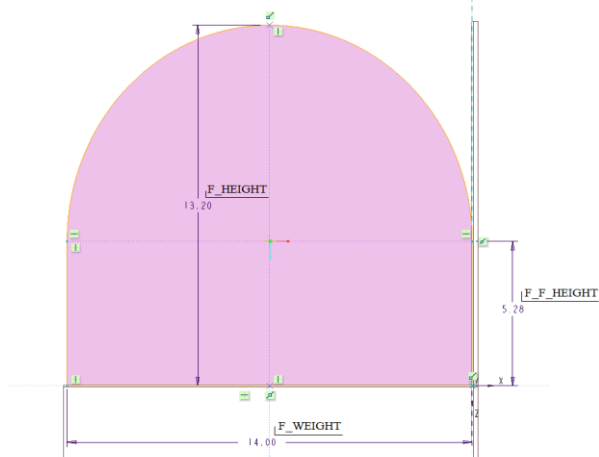


Рисунок 3.4 – Ескіз перетину сполучення



Рисунок 3.5 – Основна форма пальця

Далі нам потрібно розбити цю форму на фаланги пальців з відстанню між ними в 1 мм. Для цього за допомогою операції «Витягнути» креслимо наступний ескіз (рису. 3.6), попередньо виділивши дно форми.

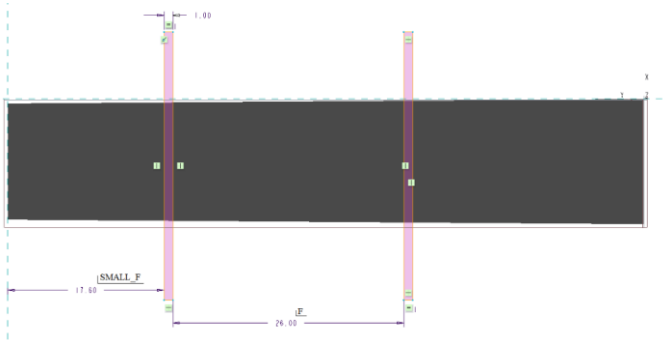


Рисунок 3.6 – Ескіз поділу на фаланги

Відокремивши одна від одної фаланги, робимо їх форму більш наближеною до анатомічної, так щоб при стисненні пальця нічого не заважало і зовні вони були схожі. Для цього робимо згини на кожній фаланзі за допомогою того ж витягування, але вже з прив'язкою до орієнтації «RIGHT». Одержаний результат можна побачити на рисунку 3.7.

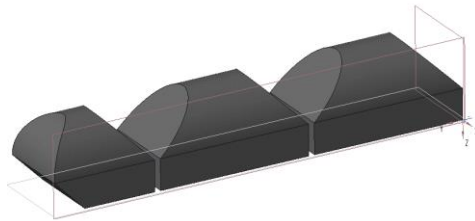


Рисунок 3.7 – Надання потрібної форми фалангам

Тепер потрібно додати отвори для опорної частини (на ній тримаються фаланги, а також вона випрямляє їх назад після стиснення) і для волосіні (яка закріплюється на кінці пальця і тягне дрібну фалангу до основи, за допомогою сервоприводів, тим самим стискаючи палець). Для цього за допомогою операції «Витягнути» ми робимо наскрізні отвори в основній частині фаланг, для опорної частини, і дрібні наскрізні отвори на згинах фаланг, так щоб при натягуванні волосіні палець спокійно міг зігнутися (рис 3.8).

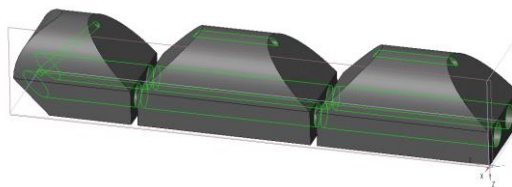


Рисунок 3.8 – Вигляд того, як повинні йти отвори в пальці

Залишилося додати заокруглення для плавності згинання, а також, для додання більш акуратної форми пальцю, заокруглення на кінець пальця і фаску уздовж верхньої частини пальця. Кінцевий результат всього проектування моделі пальця можна побачити на рисунку 3.9.

Модель вказівного пальця кіберфізичної руки готова. Щоб створити інші пальці потрібно зберегти копію даної моделі під іншим ім'ям, після чого відкрити і ввести інакші три основних параметри у вікні «Параметри», натиснути кнопку «Регенерувати» на вкладці «Модель» та зберегти. При створенні великого пальцю треба ще видалити одну фалангу, оскільки за анатомією у людини всього дві фаланги у великому пальці.

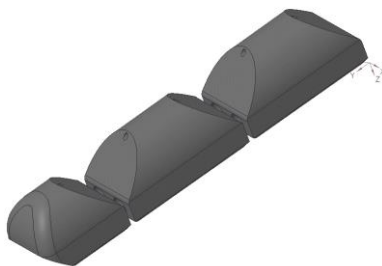
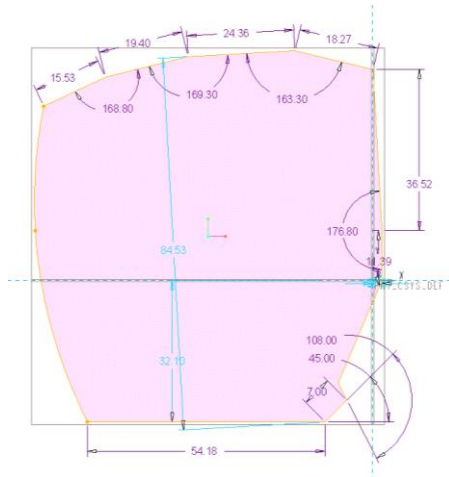


Рисунок 3.9 – Кінцевий результат проектування пальця

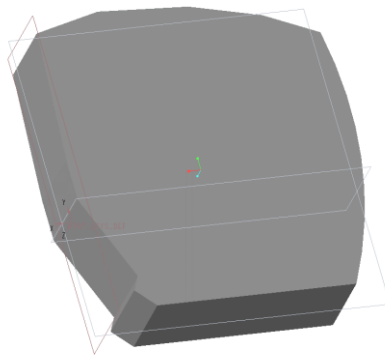
Усі пальці повинні кріпитися на долоні. Створимо новий файл типу «Деталь», підтипу «Тверде тіло» з шаблоном за замовчуванням і назвемо його «palm».

Далі потрібно створити основну частину долоні, з якої у подальшому буде формуватися більш анатомічна форма. Основні побудови будуть виконуватися у площині FRONT.

На рисунку 3.10, а зображено кресленик основної частини долоні. Використовуючи операцію «Витягнути», витягуємо цей перетин на 22 мм. Завершуємо дану дію та бачимо результат (рис. 3.10, б).



а

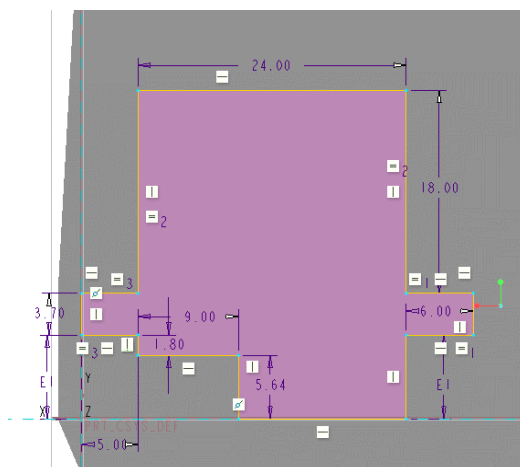


б

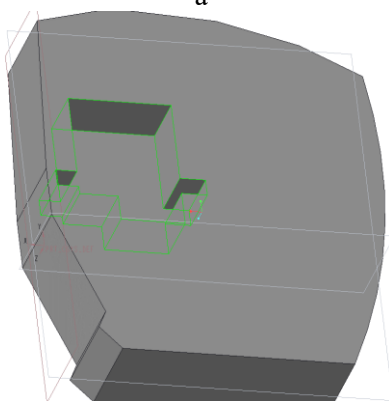
а – кресленик перетину, що витягується; б – результат операції

Рисунок 3.10 – Основна частина долоні

Наступним етапом проектування є вилучення місця під сервопривід (сервомотор) великого пальця. Для початку треба визначити розміри та форму самого мотору. Виходячи з цих даних, завдяки «Витягнути» з використанням видалення матеріалу, видаляємо спочатку частину тіла мотора. Перетин тіла мотора (рис. 3.11) витягуємо на висоту тіла мотору 12,5 мм.



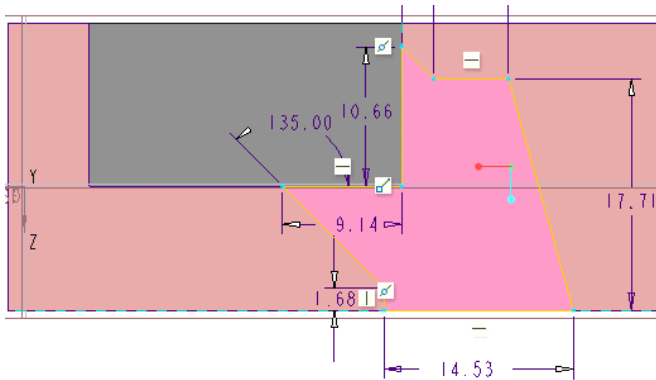
а



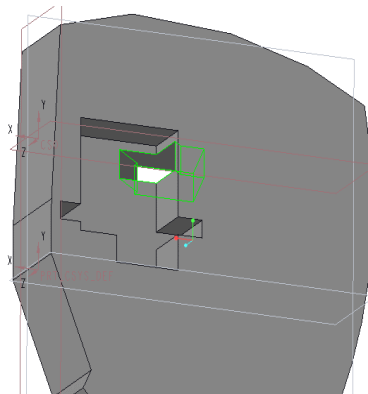
б

а – кресленник перетину, що витягується; б – вигляд моделі у результаті  
Рисунок 3.11 – Основна частина сервоприводу

Але у мотору є додадкова частинка та дрiт, який треба вивести з iншого боку долонi, так щоб вiн не заважав стисненню чи iншим рухам. Цю частину можна побудувати додавши ще одну площину, окрiм трьох основних. Для цього, вибравши операцiю «Площина» у вкладцi «Модель» будемо площину паралельно основнiй площинi TOP на вiдстанi 27,36 мм. Обравши дану площину, починаємо будувати перетин додаткової частини сервомотору (рис 3.12, а). Витягуємо цей перетин на вiдстань рiвну 6 мм у бiк площини TOP. Результат на рисунку 3.12, б.



а



б

а – кресленник перетину, що витягується; б – вигляд моделі у результаті  
Рисунок 3.12 – Додаткова частина сервоприводу

Далі зробимо виїмку під тіло обертальної частини великого пальця. Для цього від площини TOP у протилежний бік від виїмки сервомотору видалимо частину, яка починається на площині RIGHT та має ширину 34,2815 мм, висоту рівну висоті всієї деталі та глибину 26,0839 мм. Результат можна побачити на рисунку 3.13.

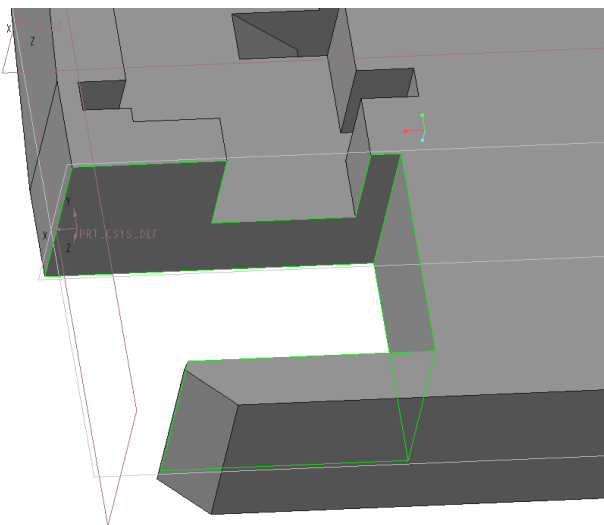


Рисунок 3.13 – Виїмка під частину великого пальця

Пальці повинні кріпитися до долоні, та при цьому функціонувати вільно. Ось чому потрібно зробити продовження для каналів опорної частини пальців і для каналів, по яких тягнеться волосінь (рис 3.8). Робимо це послідовним поєднанням дій «Система координат», «Система координат зміщення», «Крива по точкам» та «Протягнути».

Вибираємо «Система координат». Розташовуємо її на відстані 2,5 мм від верхнього ребра, та на відстані, що дорівнює половині від ширини грані пальця (використовуємо умови). Тепер, використовуючи операцію «Система координат зміщення», побудуємо потрібні точки для пальця, які можна побачити у таблиці 3.2.

Таблиця 3.2 – Координати та позначення точок для «Система координат зміщення»

Палець	Назва точки	X	Y	Z	Палець	Назва точки	X	Y	Z
Вказівний	PNT0	0.00	0.00	0.00	Середній	PNT6	0.00	0.00	0.00
	PNT1	0.00	0.00	-10.00		PNT7	0.00	0.00	-10.00
	PNT2	20.00	0.00	-20.00		PNT8	13.10	0.00	-15.00
	PNT3	26.10	0.00	-30.00		PNT9	17.70	0.00	-25.00
	PNT5	12.40	10.01	-88.30		PNT10	18.50	10.00	-80.50
	PNT4	13.10	10.00	-85.51		PNT11	18.70	10.00	-83.50
	PNT23	3.00	19.50	-10.00		PNT27	3.00	19.50	-10.00
	PNT24	-3.00	19.50	-10.00		PNT28	-3.00	19.50	-10.00
	PNT25	3.00	9.00	0.00		PNT29	3.00	9.00	0.00
	PNT26	-3.00	9.00	0.00		PNT30	-3.00	9.00	0.00
	PNT39	3.00	9.00	-10.00		PNT41	3.00	9.00	-10.00
	PNT40	-3.00	9.00	-10.00		PNT42	-3.00	9.00	-10.00
Безіменний	PNT12	0.00	0.00	0.00	Мізинець	PNT17	0.00	0.00	0.00
	PNT13	2.50	0.00	-10.00		PNT18	3.00	0.00	-25.00
	PNT14	10.00	0.00	-35.00		PNT19	12.60	10.00	-74.80
	PNT15	13.20	10.00	-77.60		PNT20	13.80	10.00	-77.40
	PNT16	13.90	10.00	-80.40		PNT35	3.00	19.50	-10.00
	PNT31	3.00	19.50	-10.00		PNT36	-3.00	19.50	-10.00
	PNT32	-3.00	19.50	-10.00		PNT37	3.00	9.00	0.00
	PNT33	3.00	9.00	0.00		PNT38	-3.00	9.00	0.00
	PNT34	-3.00	9.00	0.00		PNT45	3.00	9.00	-10.00
	PNT43	3.00	9.00	-10.00		PNT46	-3.00	9.00	-10.00
	PNT44	-3.00	9.00	-10.00					

Точки проставлено, тепер з'єднаємо точки послідовно за номерами завдяки операції «Крива по точкам», як наприклад PNT0-PNT1-PNT2-PNT3-PNT4-PNT5 – це буде крива каналу для волосінь, а PNT23-PNT39-PNT25 та PNT24-PNT40-PNT26 криві для каналів опорної частини пальців.

Оскільки було вирішено поєднати канали для волосінь великого та вказівного пальців при виконанні дії «Протягнути», то в якості початкового перетину було обрано коло діаметра 1,25 мм, а наприкінці, у ділянці з'єднання кисті до зап'ястя, було обрано коло 2,5 мм.

Для каналів опорної частини було обрано коло діаметру 3,5 мм. Результат утворення каналів можна побачити на рисунку 3.14, а, для відображення яких було обрано стиль показу «Схована лінія», що дозволяє відобразити модель у каркасному вигляді з виділеними сірим кольором схованими лініями. Так само виконуємо і для інших пальців. Канали для усіх пальців можна побачити на рисунку 3.14, б.

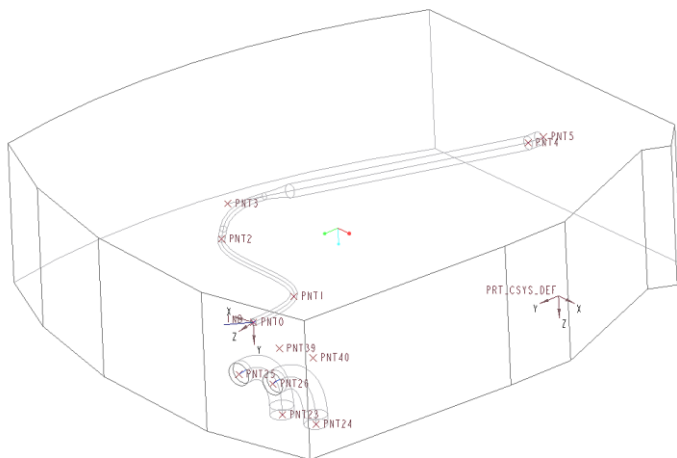
Залишилося лише зробити отвори для кріплення з зап'ястям, зріз для кращого згинання пальців та деякі косметичні покращення у вигляді фасок і заокруглень. Оберемо грань, до якої треба кріпити зап'ястя та виберемо операцію «Витягнути».

Витягуємо з видаленням матеріалу на 5 мм за перетином, зображеним на рисунку 3.15. Також робимо фаску під кутом 45° та висотою у 9 мм як зображено на рисунку 3.16. Згладжуємо ребра цих граней, для більш плавного згинання пальців (рис. 3.17).

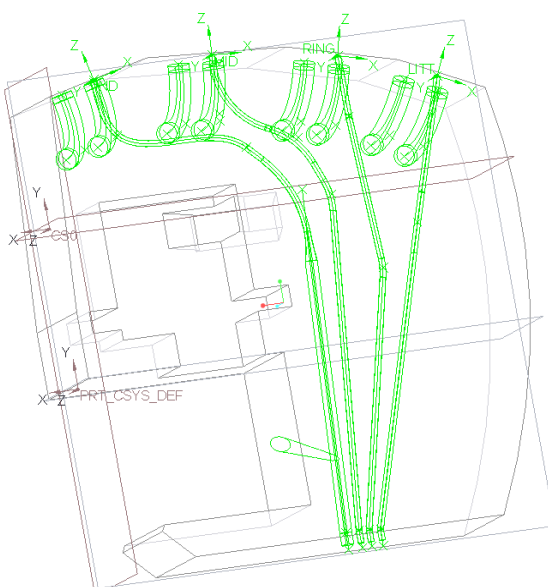
Наступною частиною кисті, яку ми спроекуємо, буде п'ясть великого пальця. Оскільки це рухома частина долоні, вона проєктується окремо та рухається за допомогою мотору.

Створимо новий файл типу «Деталь», підтипу «Тверде тіло» з шаблоном за замовчуванням і назвемо його «metacarpus». За допомогою операції «Витягнути» робимо частину моделі за ескізом зображеним на рисунку 3.18 на 24 мм.

Додамо площину для побудови додаткової частини моделі. Оберемо операцію «Площина» («Модель»>«Опорні елементи»). Ліву сторону моделі після витягування обираємо у якості прив'язки та обираємо «Зміщення» на відстань 14 мм та завершуємо побудовання.



а



б

а – для вказівного пальця; б – усіх п'яти пальців  
 Рисунок 3.14 – Зображення каналів опорної частини і каналів  
 волосінь

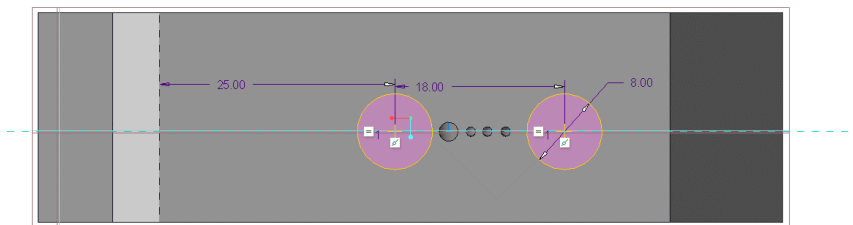


Рисунок 3.15 – Кресленик перетину пазів для кріплення кисті до зап'ястя

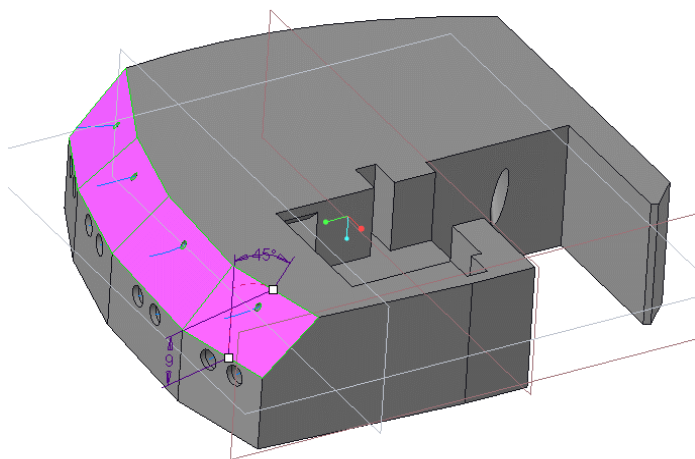
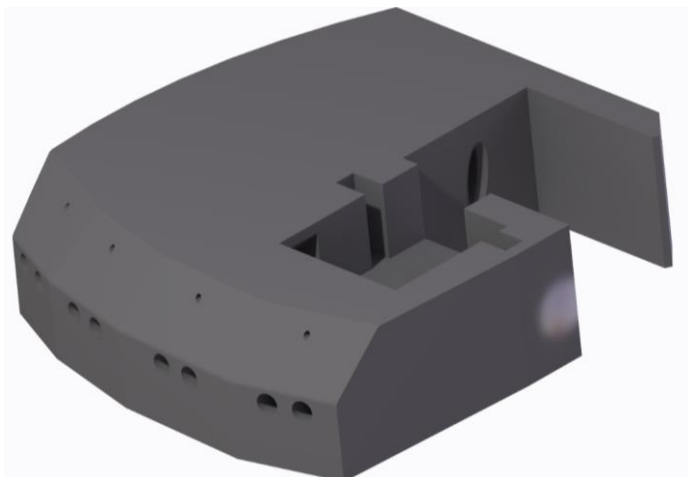
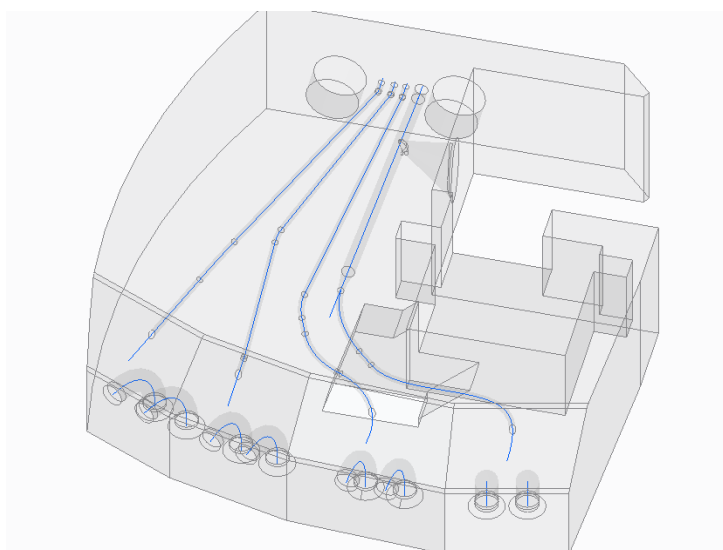


Рисунок 3.16 – Фаска для правильного згинання пальців

Результат можна побачити на рисунку 3.19. На цій площині будемо перетин, для цього оберемо «Ескіз» («Модель» > «Опорні елементи») та побудуємо ескіз так як зображено на рисунку 3.20, а – з прив'язками до сторін частини моделі. Це перетин єднальної площини (рис. 3.20, б) з фалангами великого пальця, так що тут враховані й габарити фаланг.



а



б

а – у вигляді реалістичної моделі; б – у вигляді каркасної моделі  
Рисунок 3.17 – Фінальна модель кисті протезу руки

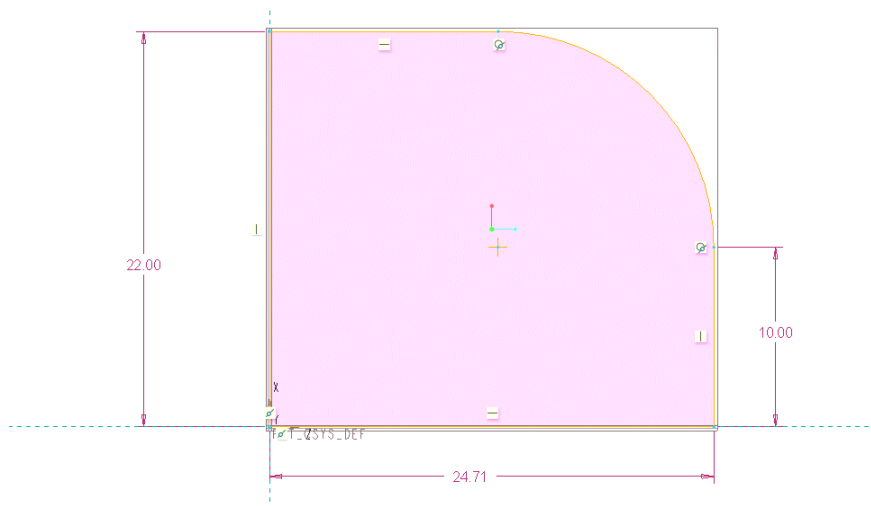


Рисунок 3.18 – Перетин частини п'ястя

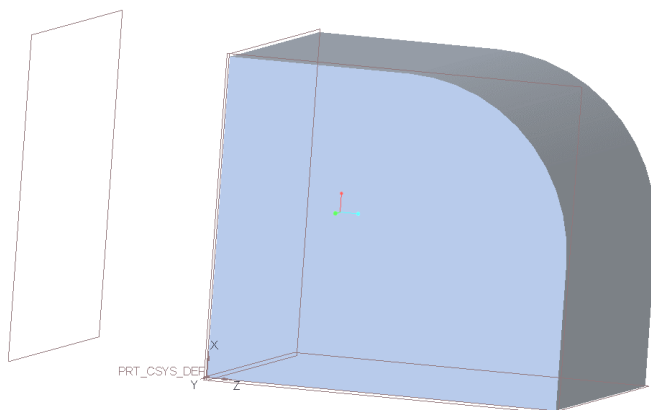
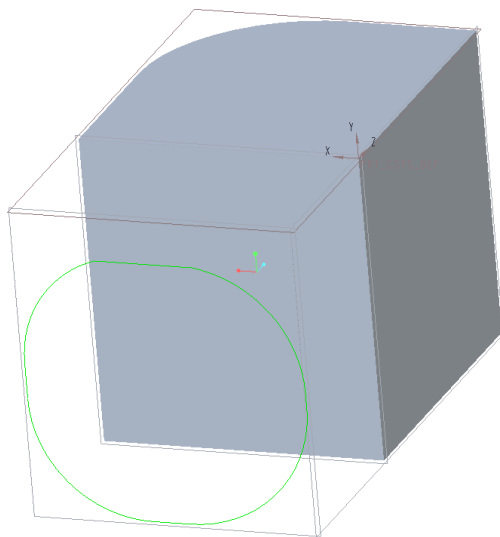
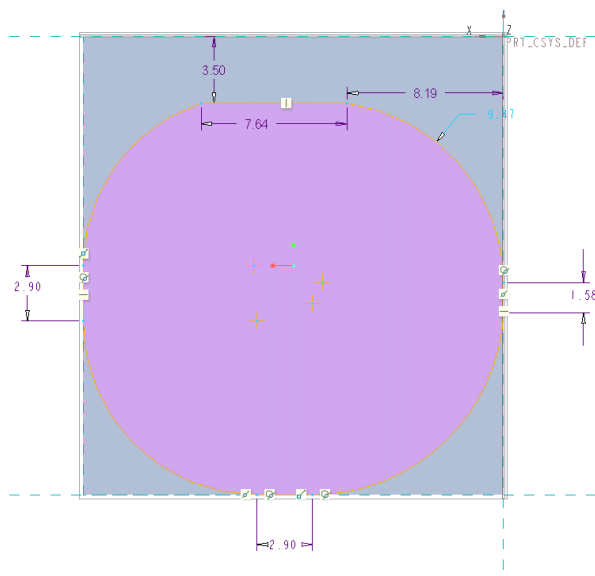


Рисунок 3.19 – Побудована додаткова площина

Далі потрібно трохи повернути перетин для більшої анатомічності. Оберемо ескіз (рис. 3.20) та застосуємо операцію «Пересунути» («Гнучке моделювання» > «Перетворити») до нього. Обираємо у якості початку криву-основу та обертаємо її на  $13^\circ$ , як зображено на рисунку 3.21, та завершуємо операцію.



а

б

а – ескіз перетину; б – результат побудови перетину  
Рисунок 3.20 – Побудова перетину на додатковій площині

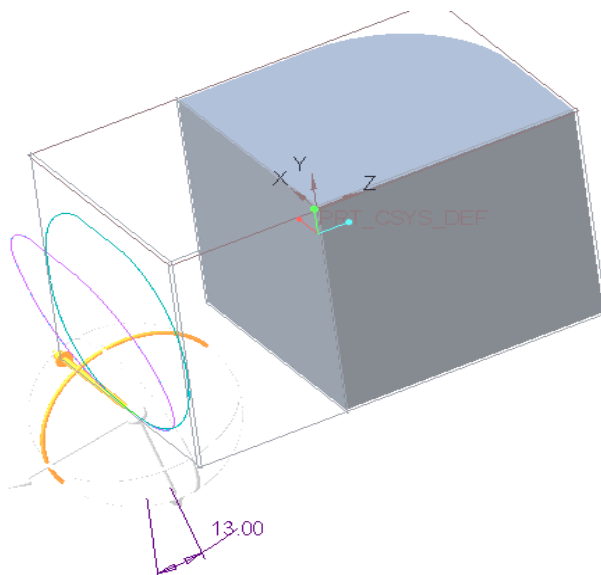
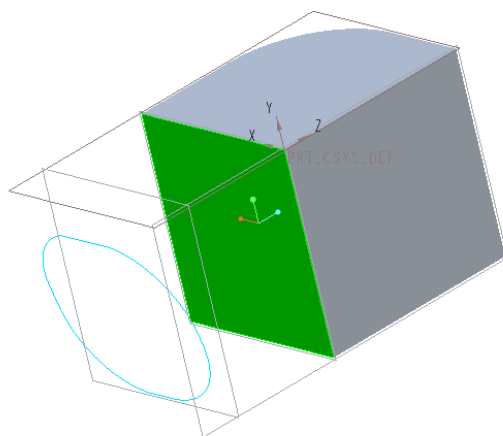


Рисунок 3.21 – Пересунення першого ескізу

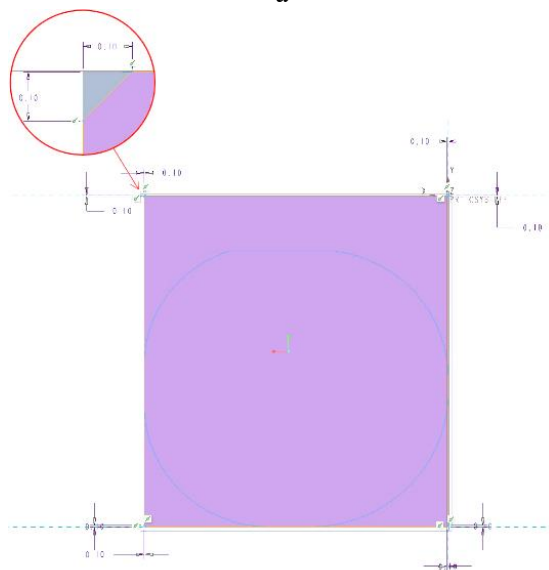
Далі, нам потрібно сполучити цей ескіз з часткою моделі. Перед цим будуємо другий ескіз майбутнього сполучення. Обираємо бічну сторону моделі (рис 3.22, а) та обравши «Ескіз» («Модель» > «Опорні елементи») будуємо ескіз, так як зображено на рисунку 3.22, б.

Оберемо «Сполучити» («Модель» > «Форми»). Перейдемо до вкладки «Перетин» та у вікні, що з'явилося, вибираємо «Вибрані перетини» та обираємо спочатку перший перетин (рис. 3.21), натискаємо кнопку «Вставити» та обираємо другий перетин (рис. 3.22). Перевіряємо, щоб кількість вершин на перетинах була однаковою, та сполучення було правильним (якщо воно скручене, то треба розташувати точку початку під фіолетовою стрілкою у правильному положенні на контурі перетину), як зображено на рисунку 3.23.

Зробимо отвір під частину сервомотору. Для цього оберемо нижню площину деталі, оберемо «Витягнути» («Модель» > «Форми») з прив'язками до прямих стін моделі та накреслимо ескіз, як зображено на рисунку 3.24, а.



а



б

а – площина, на якій розміщено ескіз; б – кресленик ескізу

Рисунок 3.22 – Побудова другого ескізу

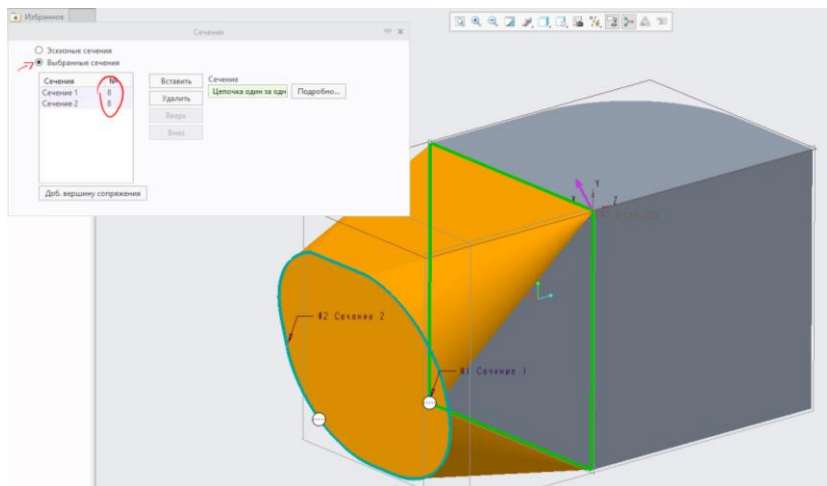


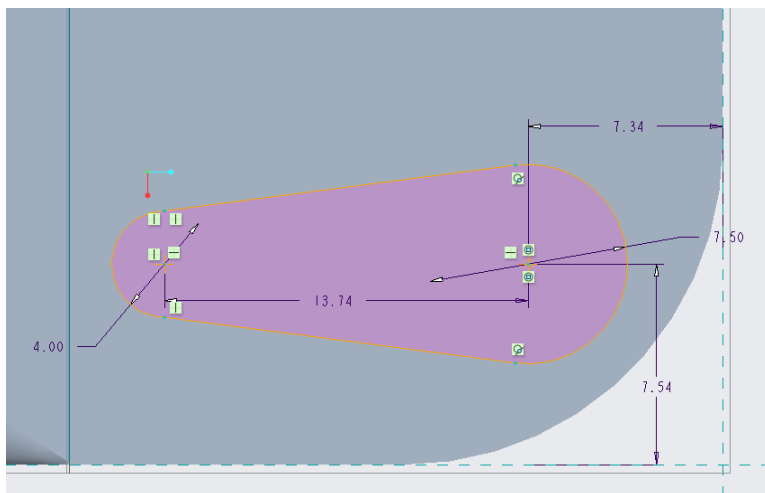
Рисунок 3.23 – Сполучення перетинів

Далі, оберемо глибину витягування 3,95 мм, натиснемо «Видалити матеріал» та у результаті отримаємо виїмку, як зображено на рисунку 3.24, б.

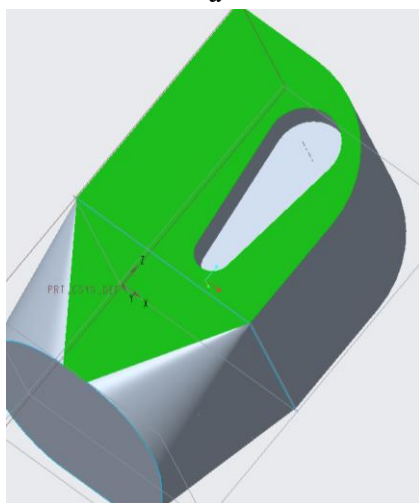
Також зробимо потрібний отвір ще для однієї частини сервомотору. Оберемо площину виїмки та натиснемо «Отвір» («Модель» > «Проектування»), та розмістимо його так, як зображено на рисунку 3.25, обравши «Додати зенківку» (діаметр зенківки 5,27 мм, кут –  $90^\circ$ ) і наскрізний отвір діаметру 4,47 мм.

Наступним кроком буде зробити канал для волосінь та канали для опорної частини з місцем кріплення. Зробимо систему координат для початку побудови цих каналів. Оберемо поверхню моделі, де вона з'єднується з фалангами пальців та оберемо «Система координат» («Модель» > «Опорні елементи») та розмістимо її, як вказано на рисунку 3.26. Усі потрібні дані для побудови теж зображено на цьому ж рисунку.

Зробимо канал для волосінь завдяки «Витягнути» («Модель» > «Форми») з прив'язками до осі отвору для сервоприводу і створеної системи координат за ескізом, зображеним на рисунку 3.27 (коло діаметром 1,25 мм на відстані 0 мм від системи координат).



а



б

а – ескіз виїмки; б – вигляд самої виїмки

Рисунок 3.24 – Виїмка під частину сервомотора

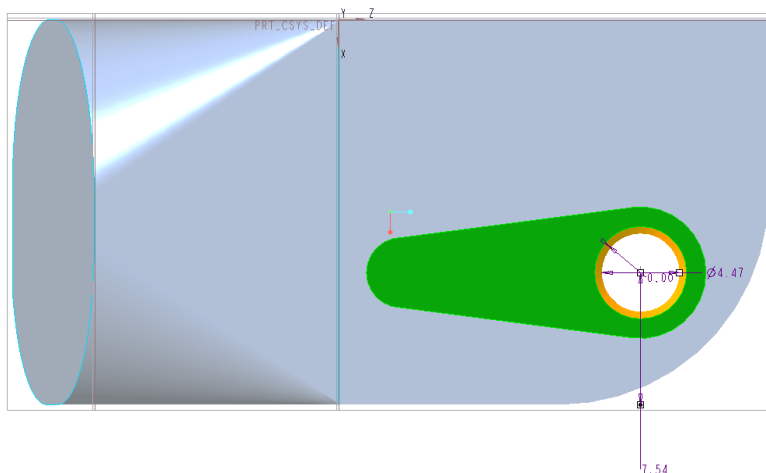


Рисунок 3.25 – Додатковий отвір сервомотору

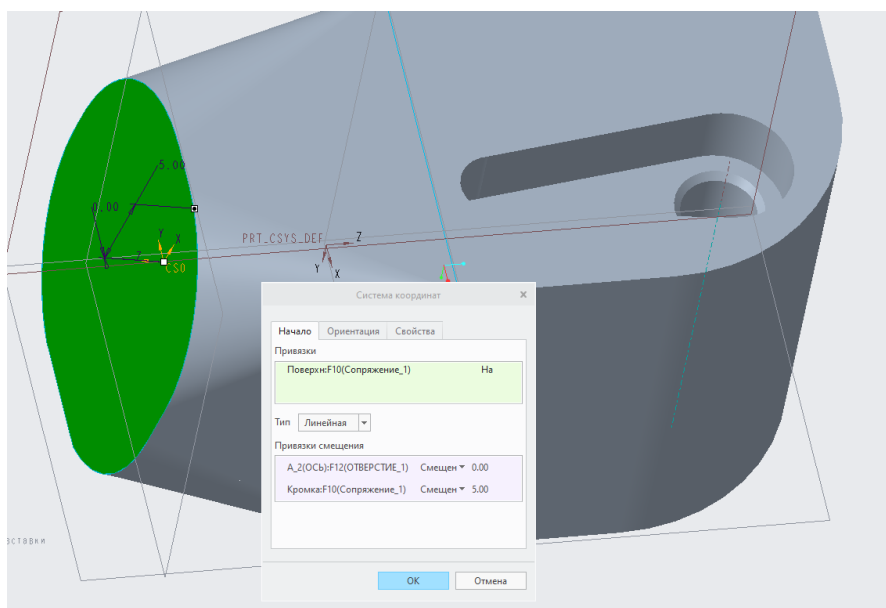


Рисунок 3.26 – Побудова системи координат

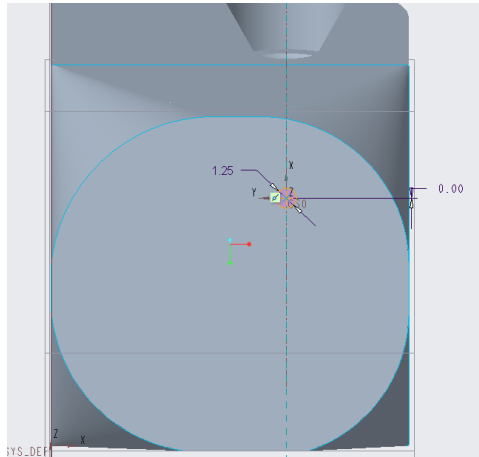


Рисунок 3.27 – Побудова каналу для волосінь

Далі, побудуємо канали для опорної частини. Зробимо дві точки, через які буде йти допоміжна вісь. Для цього оберемо «Точка» («Модель» > «Опорні елементи») та розмістимо її на новій вісі координат (рис 3.28), а іншу точку розмістимо на нижній кромці, як показано на рисунку 3.29, зі зміщенням у відношенні 0,8.

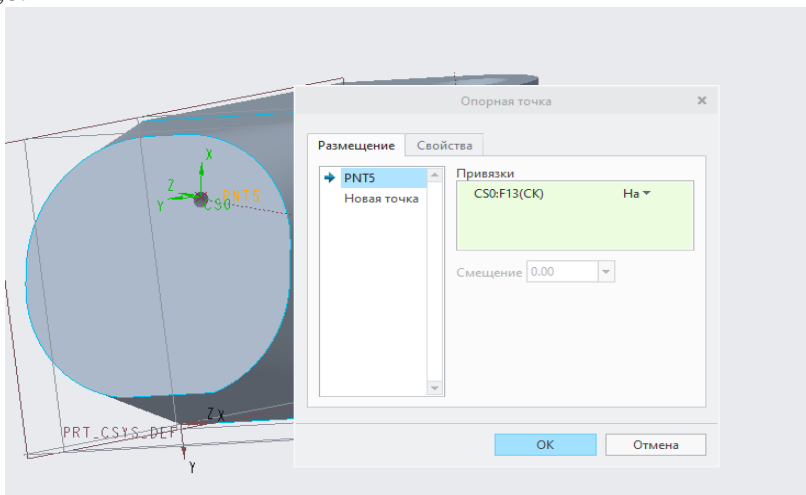


Рисунок 3.28 – Перша точка для побудови допоміжної вісі

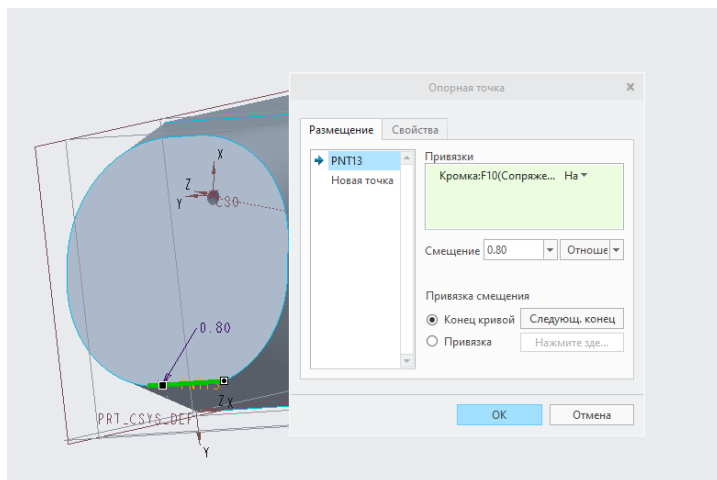


Рисунок 3.29 – Друга точка для побудови допоміжної вісі

Побудуємо допоміжну вісь. Оберемо «Вісь» («Модель» > «Опорні елементи») та у якості прив'язок оберемо точки, які щойно додали (рис 3.30). Далі за допомогою «Ескіз» («Модель» > «Опорні елементи») розмістимо ще одну систему координат, так як зображено на рисунку 3.31.

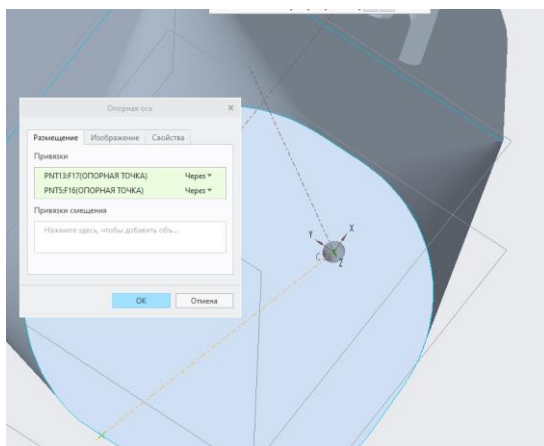


Рисунок 3.30 – Розміщення додаткової вісі

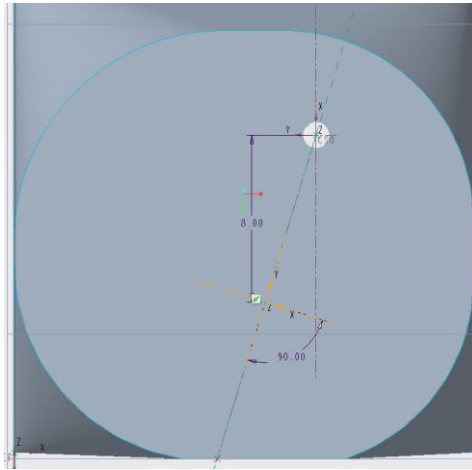


Рисунок 3.31 – Розміщення додаткової системи координат у режимі ескізу

Наступним кроком розмістимо «Систему координат зміщення» («Модель» > «Опорні елементи») у якості прив'язки обрав останню систему координат, та додавши точки, як вказано на рисунку 3.32.

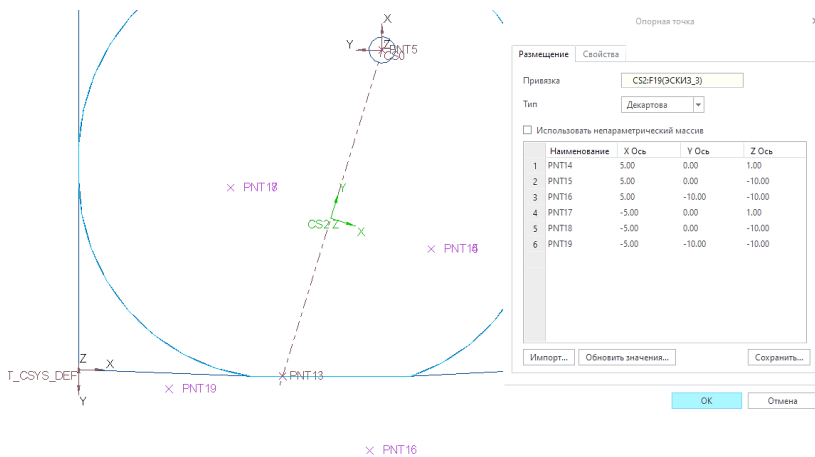


Рисунок 3.32 – Побудова точок для кривих опорної частини

Обравши «Крива по точкам» («Модель» > «Опорні елементи»), з'єднаємо PNT14-PNT15(обравши тип з'єднання «Пряма лінія» з додаванням заокруглення радіусу 9 мм)-PNT16(обравши тип з'єднання «Пряма лінія» без додавання заокруглення) у першу криву та PNT17-PNT18(обравши тип з'єднання «Пряма лінія» з додаванням заокруглення радіусу 9 мм)-PNT19(обравши тип з'єднання «Пряма лінія» без додавання заокруглення) у другу криву для каналів опорної частини пальців. Отримані криві можна побачити на рисунку 3.33.

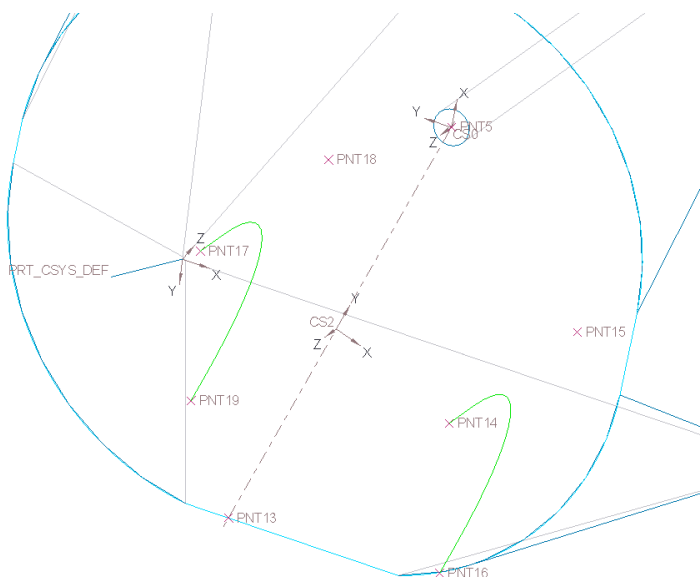
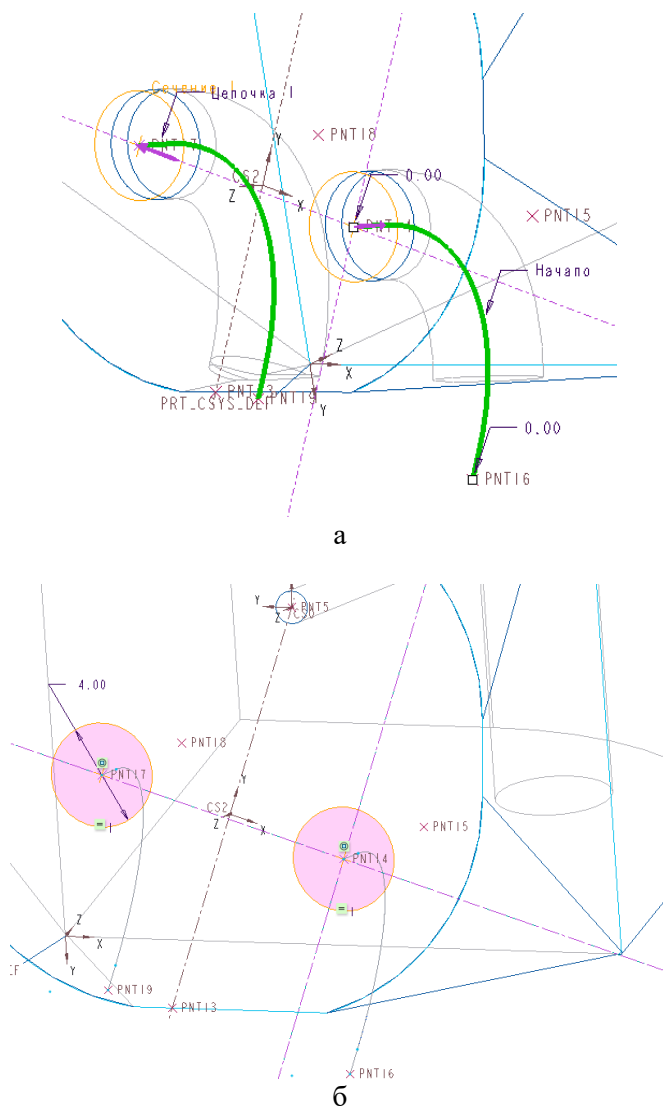


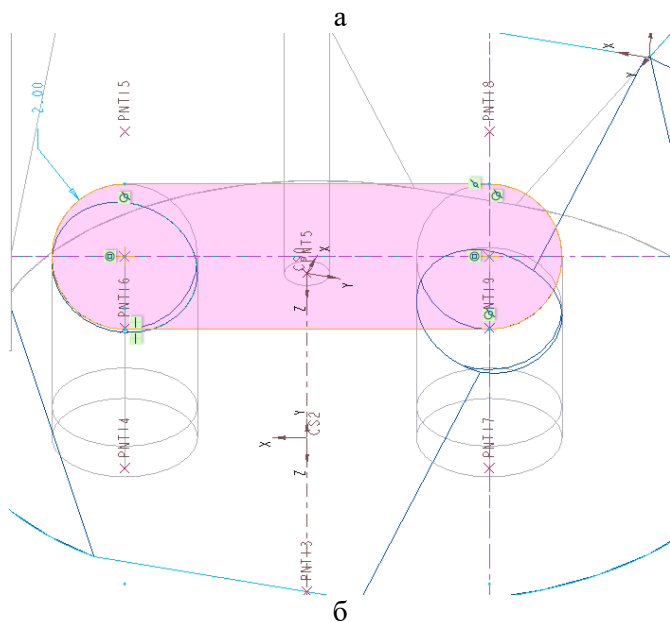
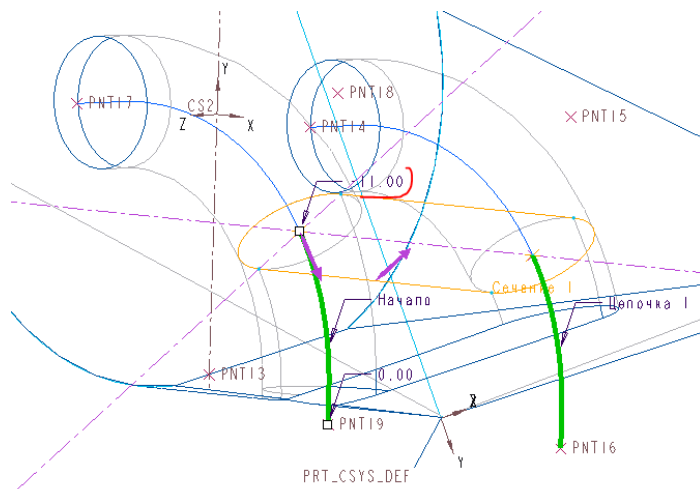
Рисунок 3.33 – Побудова кривих опорної частини

Далі, оберемо «Протягнути» («Модель» > «Форми»), попередньо обравши криві (рис. 3.34, а), та намалюємо ескіз двох однакових кіл діаметра 4 мм, як зображено на рисунку 3.34, б.

Побудуємо місце де будуть закріплені опорні частини. Також обираємо криві і «Протягнути» («Модель» > «Форми») але при побудові ескізу обираємо інший кінець кривих та відстань 11 мм (рис. 3.35, а) та креслимо ескіз, який показано на рисунку 3.35, б.

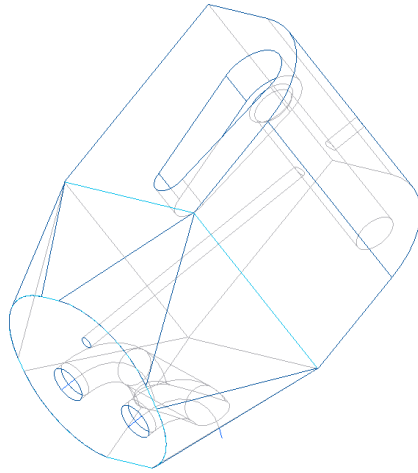


а – криві, по яким йде протягування; б – ескіз який буде протягуватися  
 Рисунок 3.34 – Побудова каналів опорної частини

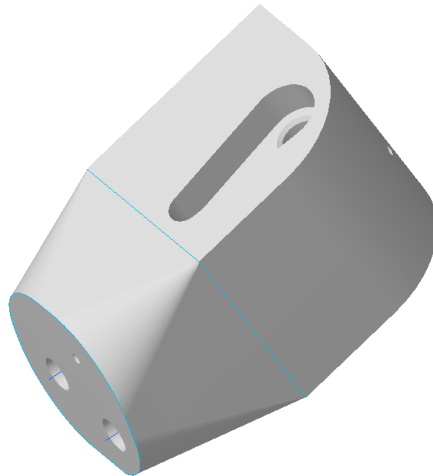


а – криві, по яким йде протягування; б – ескіз який буде протягуватися  
Рисунок 3.35 – Побудова місця закріплення опорної частини

Модель п'ястя великого пальця готова. На рисунку 3.36 можна побачити каркас цієї моделі (рис 3.36, а), та результат візуалізації моделі (рис 3.36, б).



а



б

а – у вигляді каркасної моделі; б – у вигляді реалістичної моделі

Рисунок 3.36 – Фінальна модель п'ястя великого пальця протезу

Після створення файлів усіх п'яти пальців та долоні, можна створити складальну одиницю. При створенні нового файлу, потрібно вибрати тип «Складальна одиниця», далі відкрити файл кожного з пальців і розмістити їх в просторі. Насамперед, це зручно для того, щоб подати на друк відразу всі пальці, а не кожен палець окремо, чи побачити як буде виглядати кисть руки у фіналі (рис. 3.37).

Після проектування моделі пальця та інших моделей можна виконати побудову кресленика для звітності. Зробимо для прикладу кресленик фаланг вказівного пальця. Для цього можна, не виходячи з файлу моделі, натиснути «Файл» > «Створити» і вибрати тип «кресленик». Вибравши далі шаблон «a4\_drawing», автоматично побудуються три основних види на аркуші А4. Залишилося тільки проставити анотації розмірів, підібрати потрібний масштаб і кресленик готовий (рис 3.38).

Таким чином, було спроектовано модель частини протезу руки. Було враховано те, що треба додати виїмку для сервомотору, канали для волосінь і канали для опорних частин. Параметричне моделювання дозволило спроектувавши один палець, швидко спроектувати інші. Долоню та п'ясть було спроектовано з урахуванням того, щоб рух великого пальця був більш плавними і йому нічого не заважало. Завдяки можливостям середовища Creo Parametric можна створити складальну одиницю та відправити її на друк.



Рисунок 3.37 – Фінальний рендерінг моделі

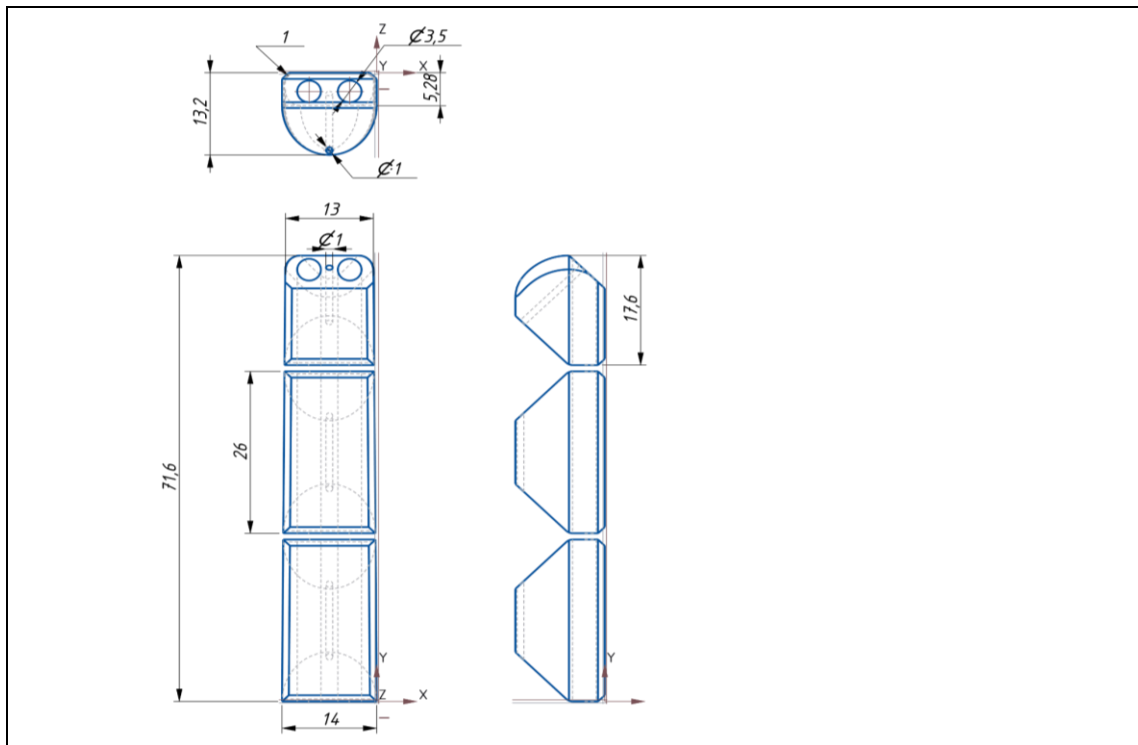


Рисунок 3.38 – Фінальний кресленик моделі пальця

### 3.1.3 Виготовлення фізичного прототипу конструктиву кіберпротезу

Після розробки 3D моделей, в середовищі PTC Creo було згенеровано файли в форматі STL, які є вхідними для програми Simplify3D [50]. В цій програмі було проведено перетворення даних для процесу 3D друкування (рис. 3.39), а також виконано налаштування параметрів друку (рис. 3.40).

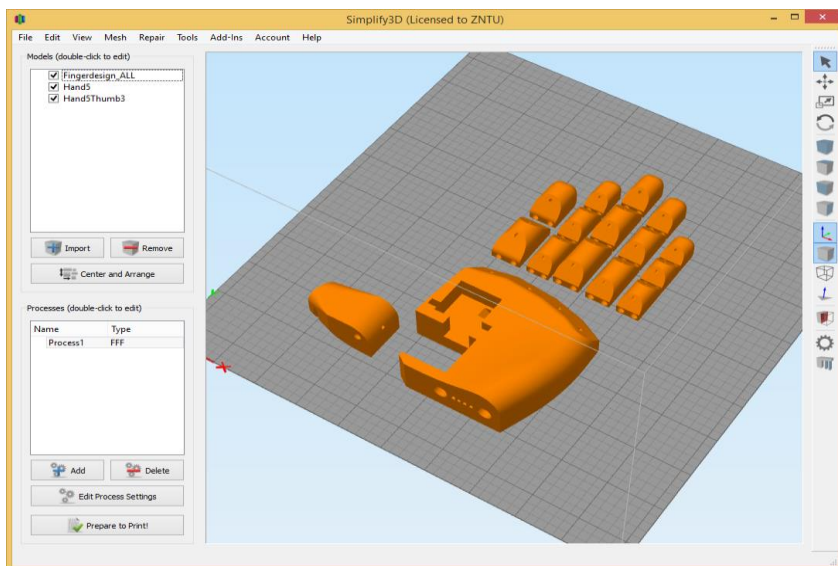


Рисунок 3.39 – 3D модель кисті кіберпротезу в середовищі Simplify3D

В якості матеріалу використовувався ABS пластик, що стійкий до багатьох несприятливих умов експлуатації, таких 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.

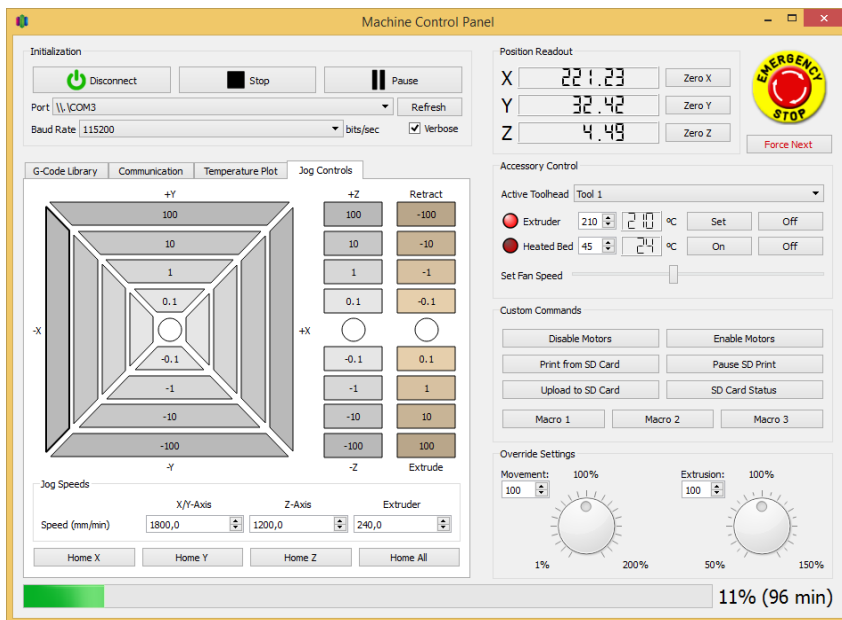


Рисунок 3.40 – Налаштування параметрів друкування в Simplify3D

## 3.2 Розробка електронної частини прототипу кіберпротезу

### 3.2.1 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 програмованими МК, а також додаткові модулі. Програмна частина складається з середовища розробки прошивок Arduino IDE (програми для написання скетчів і завантаження їх на МК Arduino), спрощеної мови програмування та великої кількості готових функцій та бібліотек (рис. 3.41) [52].

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.

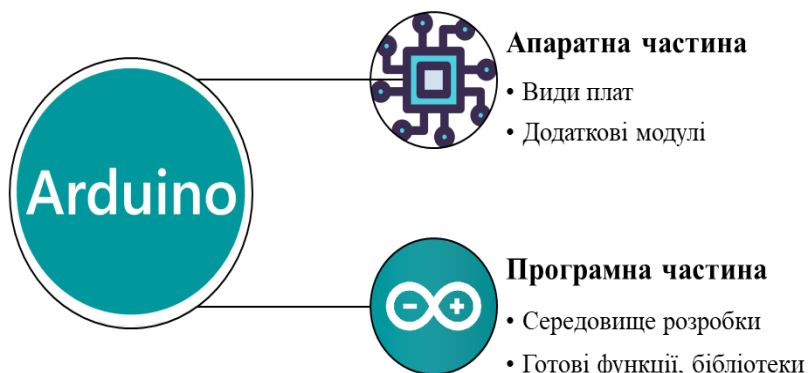


Рисунок 3.41 – Компоненти Arduino

До самої плати можна підключати різну периферію – кнопки, деякі види датчиків (температури, тиску, освітленості, прискорення і т.п.), світлодіоди, рідкокристалічні індикатори, після чого написати програму і організувати взаємодію всіх компонентів.

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. 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 Кб EEPROM пам'яті. На периферії є 14 дискретних (цифрових) каналів введення / виводу і 6 аналогових каналів введення / виводу, що дозволяють перекривати більшість 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 МК, який знаходиться в тому ж класі, але має деякі відмінності позитивного характеру. Це більша кількість аналогових входів (12 проти 6) для сенсорів, більше каналів широтно-імпульсної модуляції (ШИМ) (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 одна від одної своїми формами, характеристиками і можливостями. В таблиці 3.3 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 застосуванні mini-USB кабелю для взаємодії замість стандартного (рис. 3.42).

Arduino Mini є компактним (30 × 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 та / або через макетну плату. На платі немає USB-порту, тому прошивати потрібно через окремий USB-Serial адаптер (рис. 3.43).

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 Vi. В іншому, комплектація і способи взаємодії збігаються з Arduino Leonardo.

Таблиця 3.3 – Порівняння плат Arduino

Характеристика	Плата Arduino							
	Uno	Leonardo	Nano	Mega	Due	Mini	Micro	MO
Мікроконтролер	Atmega 328	ATmega 32u4	ATmega 328	ATmega2560	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	6	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
ОЗП, 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

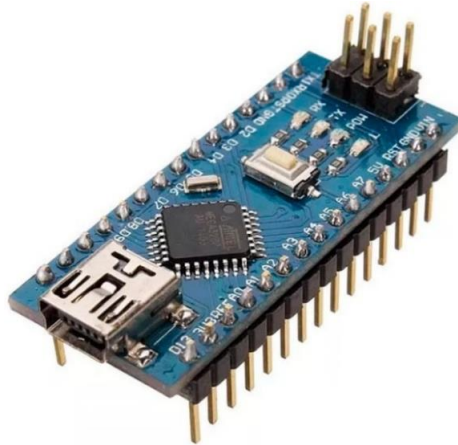


Рисунок 3.42 – Зовнішній вигляд Arduino Nano

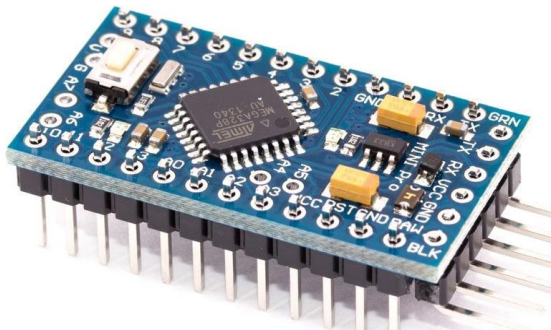


Рисунок 3.43 – Зовнішній вигляд Arduino Mini

Arduino Micro також має один МК ATmega32u4 для прошивки через USB і для виконання програм; також може виступати в ролі клавіатури або миші; надає таку саму кількість пам'яті, а також цифрових, аналогових і ШІМ-портів (рис. 3.44).

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 3.4 представлено порівняння плат Arduino Nano, Arduino Mini та Arduino Micro [55].

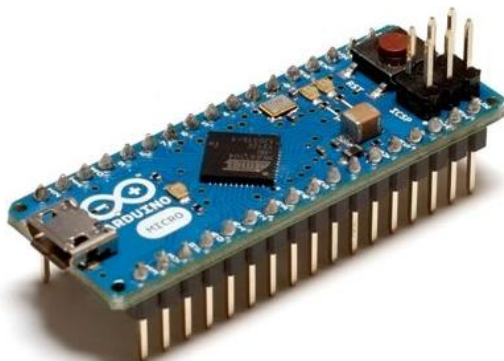


Рисунок 3.44 – Зовнішній вигляд Arduino Micro

В результаті проведеного аналізу було визначено, що плата Arduino Mini, навіть незважаючи на свою компактність, для виконання проєкту зі створення прототипу кіберпротезу не підходить, оскільки її функціональних 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. 3.45). Існує кілька версій цих плат, що відрізняються між собою використовуваними МК. 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 об'єми незалежної і flash пам'яті, а також тактову частоту 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 (павербанк або від адаптера, підключеного в розетку), що має напругу 6-20 В з низьким рівнем пульсацій [56, 57].

Таблиця 3.4 – Порівняння мініатюрних плат Arduino

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

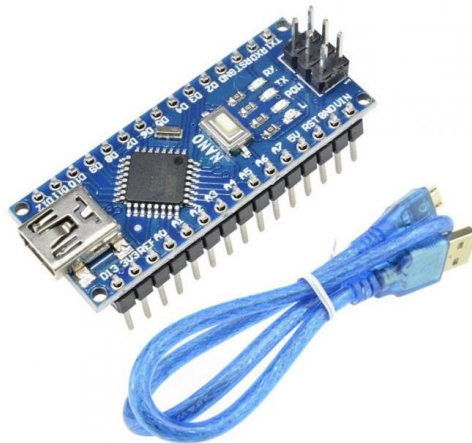


Рисунок 3.45 – Зовнішній вигляд Arduino Nano

На рисунку 3.46 наведено опис основних компонентів плати Arduino Nano [58].

Основні технічні характеристики Arduino Nano наведено в таблиці 3.5 [59].

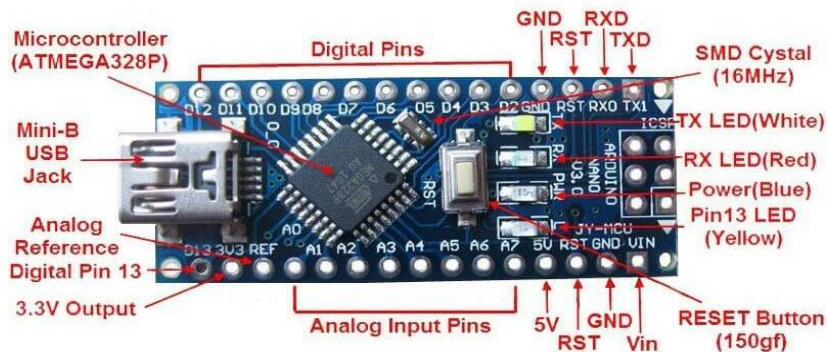


Рисунок 3.46 – Основні компоненти плати [58]

Таблиця 3.5 – Технічні характеристики Arduino Nano

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

Плата має 14 цифрових пінів, що позначені літерою «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 В все одно буде вважатися як 5 В (логічна одиниця). Аналогові піни на платі позначені літерою «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 ШІМ. 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 рисунку 3.47 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.

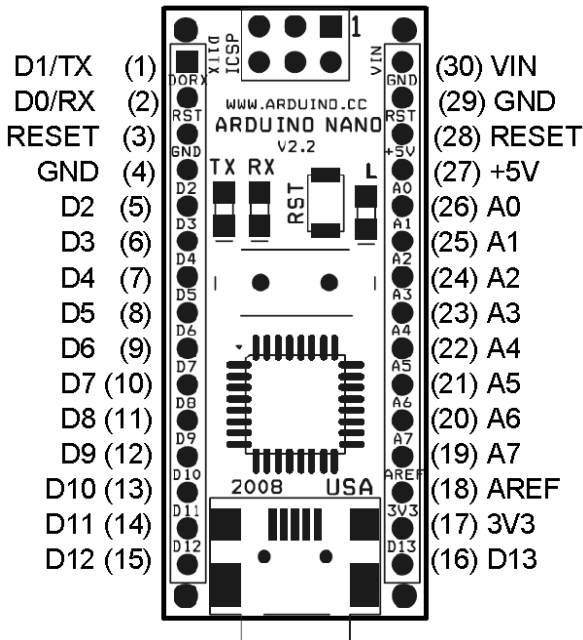


Рисунок 3.47 – Розташування пінів на платі Arduino Nano

### 3.2.2 Аналіз та вибір плати розширення для Arduino

Одною з ключових переваг Arduino є популярність. Її активно підтримують виробники електронних пристроїв, що випускають спеціальні версії різних плат, що розширюють базову функціональність контролера. Такі плати називаються платами розширення (інша назва: arduino shield, шилд) та служать для виконання найрізноманітніших завдань [60].

Плата розширення Arduino – це закінчений пристрій, що призначений для виконання певних функцій і підключається до основного контролера за допомогою стандартних роз'ємів. На платі розширення встановлені всі необхідні електронні компоненти, а взаємодія з МК і іншими елементами основної плати відбувається через стандартні піни Arduino. Найчастіше живлення на шилд теж подається з основної плати Arduino, хоча в багатьох випадках є можливість живлення з інших джерел. На будь-якій платі розширення залишаються кілька вільних пінів, які можна використовувати на свій розсуд, підключивши до них будь-які інші компоненти.

Найбільш популярним прикладами шилдів є плати розширення для роботи з датчиками (Sensor Shield), двигунами (Arduino Motor Shield), LCD-екранами (LCD -Shield), SD-картами (Data Logger Shield), мережеві (Ethernet Shield) і GPS-Shield, Shield з вбудованими реле для підключення до навантаження (рис. 3.48). В таблиці 3.6 наведено опис деяких видів Shield [60].

В результаті аналізу Arduino Shield, виявлено, що для реалізації прототипу кіберпротезу 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 може бути 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 знадобитися велика кількість входів і виходів для підключення периферійних пристроїв,

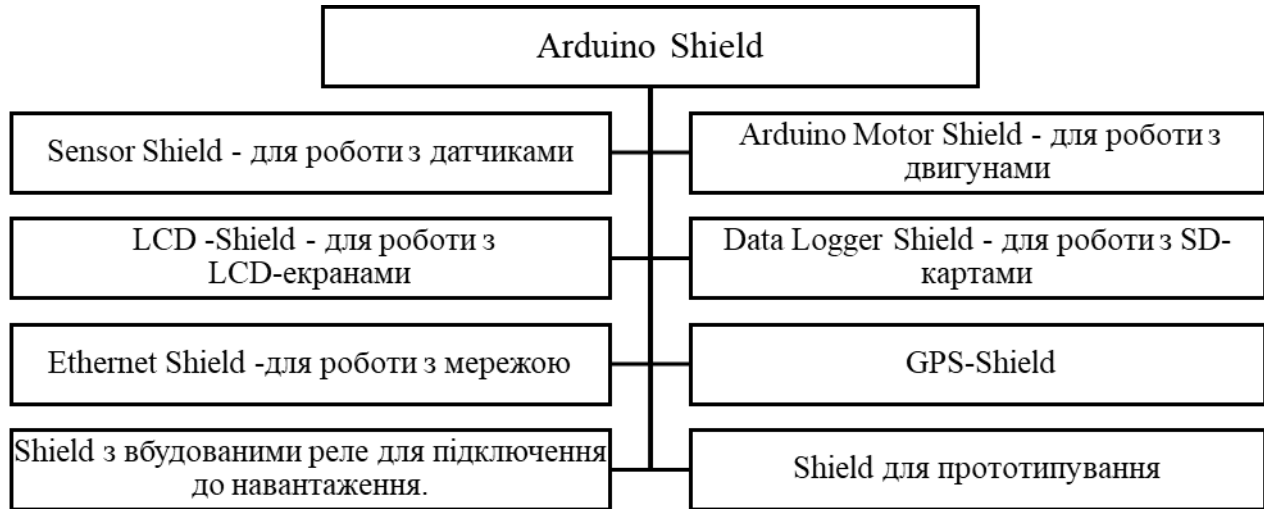


Рисунок 3.48 – Види Arduino Shield

Таблиця 3.6 – 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.

тому для виправлення цього недоліку та полегшення підключення використовують плату розширення Arduino Nano Shield V3.0 (рис. 3.49), 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.

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

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].

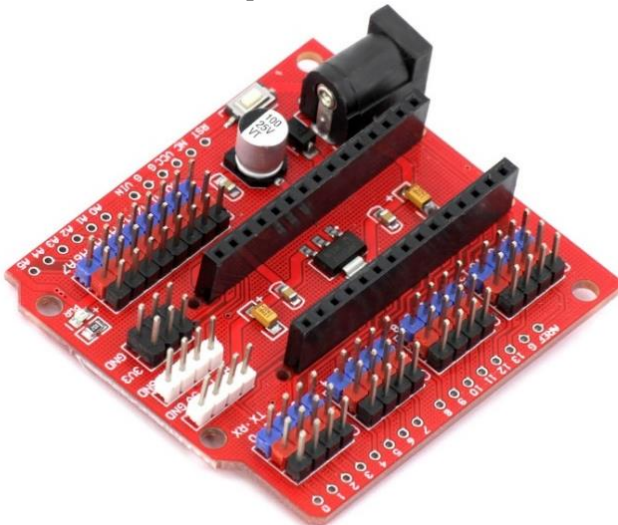


Рисунок 3.49 – Зовнішній вигляд плати розширення Arduino Nano Shield V3.0



Таблиця 3.7 – 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 прототипу протезу є компактність, саме тому вибір серводвигунів має здійснюватися 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 довжиною 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 3.8 наведено порівняльну характеристику цих двох моделей [66-68].

В результаті порівняння 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 протезу руки.

Серводвигун SG90 (рис. 3.51), 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. 3.52).

Таблиця 3.8 – Порівняльна характеристика серводвигунів

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

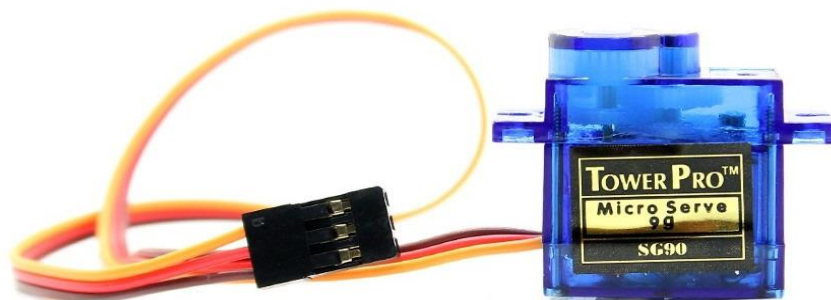


Рисунок 3.51 – Зовнішній вигляд серводвигуна SG90



Рисунок 3.52– Комплектація серводвигуна SG90

Основні характеристики SG90 наведено в таблиці 3.9. На рисунку 3.53 представлено ескіз серводвигуна з розмірами [64-67, 69].

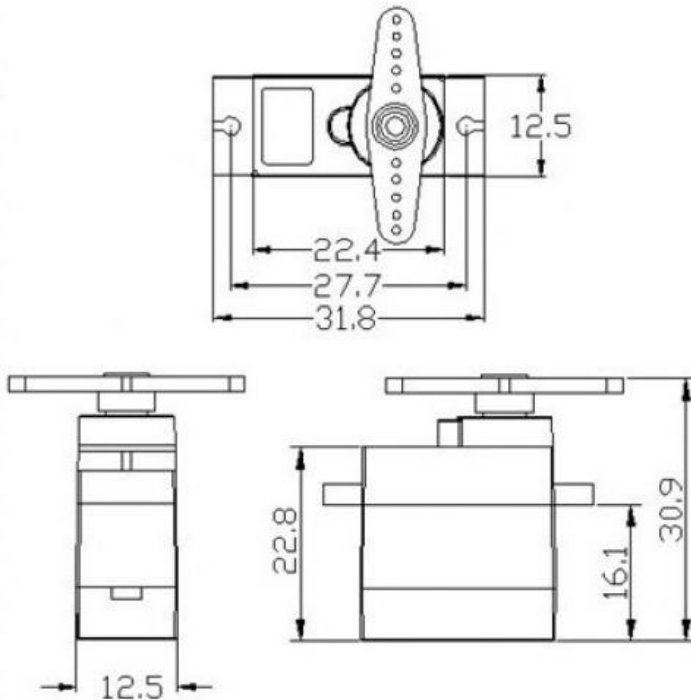


Рисунок 3.53 –Ескіз серводвигуна SG90

Таблиця 3.9 – Характеристики серводвигуна SG90

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

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

- коричневий дріт – земля (Ground, підключається до піна GND на платі Arduino)
- червоний дріт – харчування +5 V (підключається до піна 5V на платі Arduino)
- жовтий (помаранчевий) дріт – сигнал управління (підключається до цифрового піна 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. 3.54) [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 3.54 – Програмування серводвигуну 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 піна, або неправда – в зворотному випадку;

• `detach ()` – виробляє дію, зворотно 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 пристрої 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. Проблема 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 – це 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 ° і 2400 мкс – для 180 ° (рис. 3.55) [70].

Слід звертати увагу на напругу, яка використовується при роботі серводвигуна. Якщо значення напруги перевищує допустимі значення, вказане в технічних характеристиках серводвигуна, механічні частини можуть вийти з ладу або зубчасті колеса провернути 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].

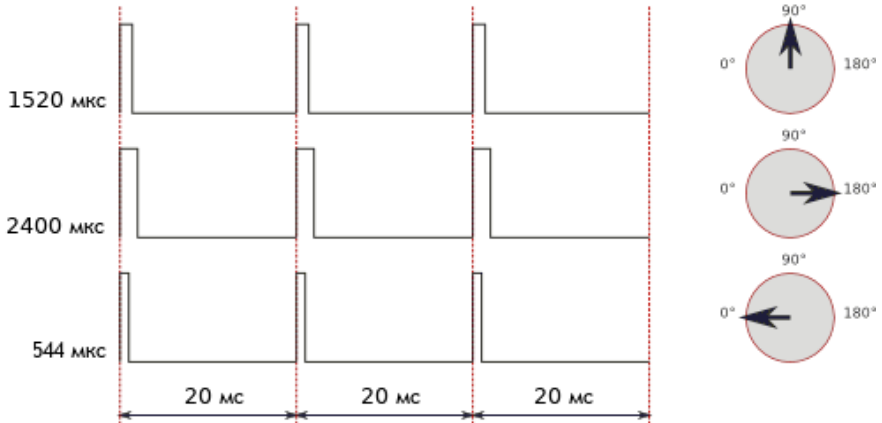


Рисунок 3.55 – Напрямок обертання серводвигуна

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

Таким чином, для створення прототипу кіберпротезу буде використано:

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

Прототип протезу 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. 3.56). Після чого необхідно підключити плату Arduino Nano та серводвигуни до плати Arduino Nano Shield V3.0 (рис. 3.57).



Рисунок 3.56 – Закріплення сервоприводів на механічній частині

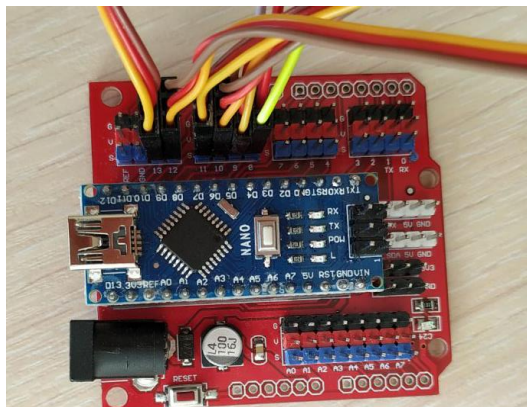


Рисунок 3.57 – Підключення компонентів до плати Arduino Nano Shield V3.0

Для підключення Arduino Nano Shield V3.0 потрібно акуратно «надіти» його на основну плату. Зазвичай контакти Shield типу гребінки 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 штиреві контакти, головне діяти акуратно і не додавати зайвої сили (рис. 3.58).

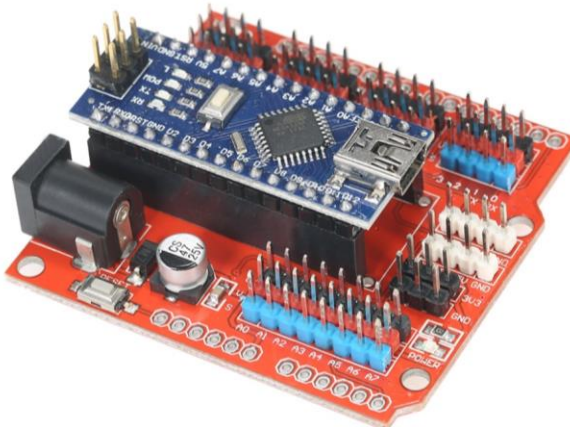


Рисунок 3.58 – Підключення Arduino Nano до плати розширення

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 3.56. 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 рисунку 3.59 з урахуванням шостого серводвигуна, що має забезпечити обертання кисті.

Кожному пальцю на руці відповідає всій серводвигун, що відповідно під'єднаний до різних пінів на платі. На рисунку 3.60 пронумеровано кожний з серводвигунів, а в таблиці 3.10 розписано піни, що їм відповідають.

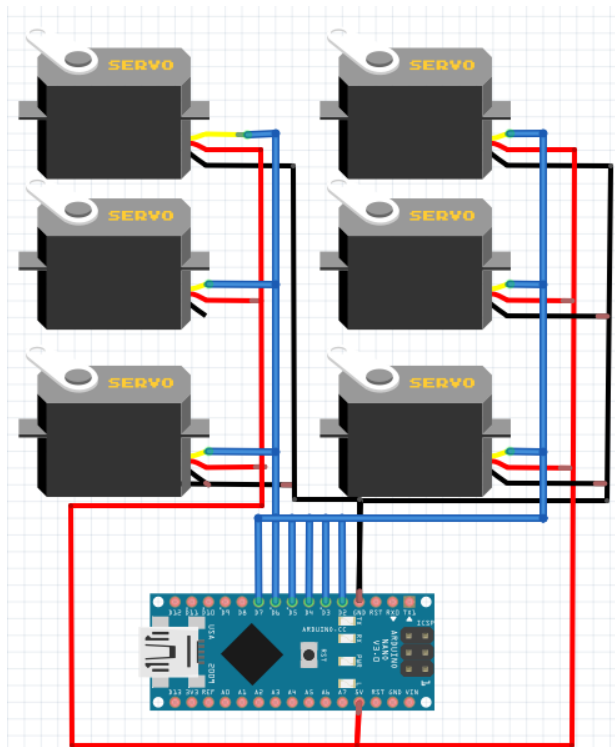


Рисунок 3.59 – Схема підключення серводвигунів

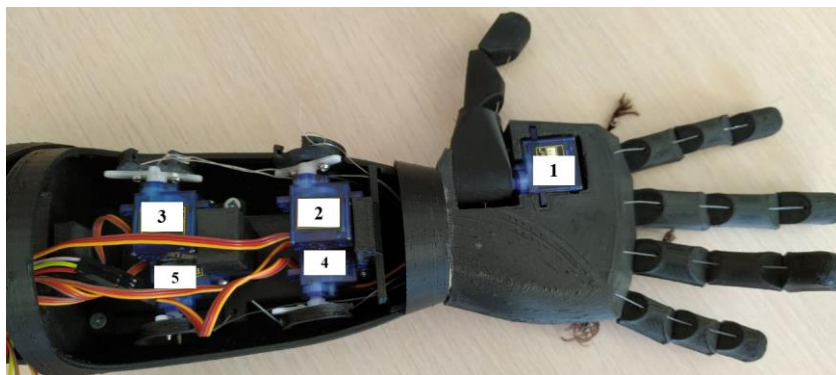


Рисунок 3.60 – Нумерація серводвигунів

Таблиця 3.10 – 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

### 3.4 Розробка програмного забезпечення для прототипу

На сьогоднішній день існує багато середовищ програмування для Arduino: Arduino IDE, Programino, B4R, CodeBlocks for Arduino, Visuino, S4A, ArduBlock та інші.

Стандартне середовище розробки Arduino IDE використовується для роботи всіх видів Arduino з комп'ютером. Щоб почати роботу, потрібно спочатку завантажити Arduino IDE з офіційного сайту і встановити його (рис. 3.61). Зручніше завантажувати Windows Installer, особливо якщо середовище розробки буде встановлено на постійному робочому комп'ютері. Якщо викачаний архів, то його потрібно розпакувати і запустити файл Arduino.exe.

#### Download the Arduino Software



Рисунок 3.61 – Завантаження середовища розробки Arduino IDE

Щойно середовище буде встановлено, потрібно його запустити та налаштувати роботу з платою Arduino. Для цього потрібно підключити до комп'ютера саму плату через USB. На платі загориться світлодіод «ON» і почне блимати світлодіод «L». Це означає, що на плату подано живлення і МК почав виконувати прошивку на заводі програму «Blink».

Для налаштування Arduino IDE на роботу з конкретною платою Arduino треба дізнатися, який номер COM-порту привласнив комп'ютер цій платформі. Зайдіть в «Диспетчер пристроїв» Windows і розкрийте вкладку «Порти (COM і LPT)». У списку з'явиться встановлена плата і буде вказано номер порту, до якого підключається плата (рис. 3.62).

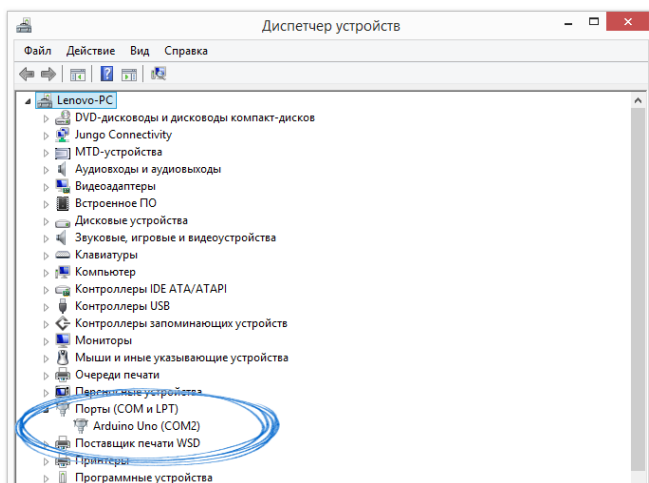


Рисунок 3.62 – Визначення номеру COM-порту пристрою

Після цього потрібно запустити Arduino IDE, перейти в меню Інструменти >> Порт, і вказати порт (рис. 3.63), до якого приєднана Arduino.

Потім потрібно вибрати модель підключеної плати в меню Інструменти >> Плати. Важливо пам'ятати, що якщо до комп'ютера буде підключатися інша плата, налаштування знову потрібно буде поміняти на відповідний пристрій (рис. 3.64).

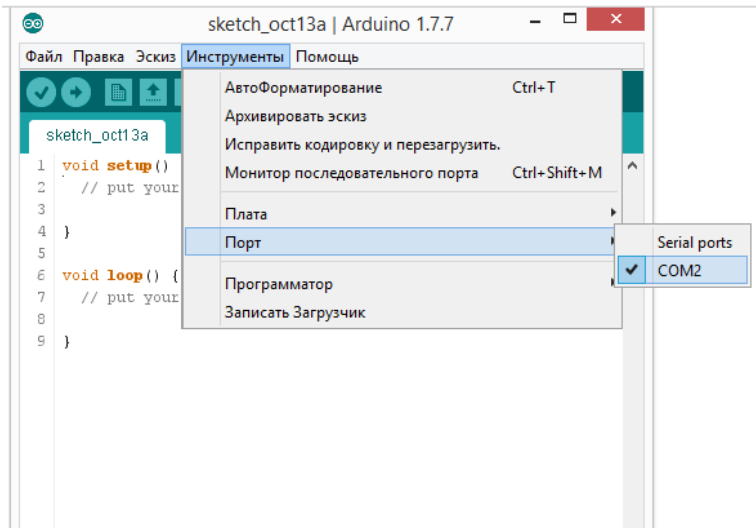


Рисунок 3.63 – Налаштування порту в Arduino IDE

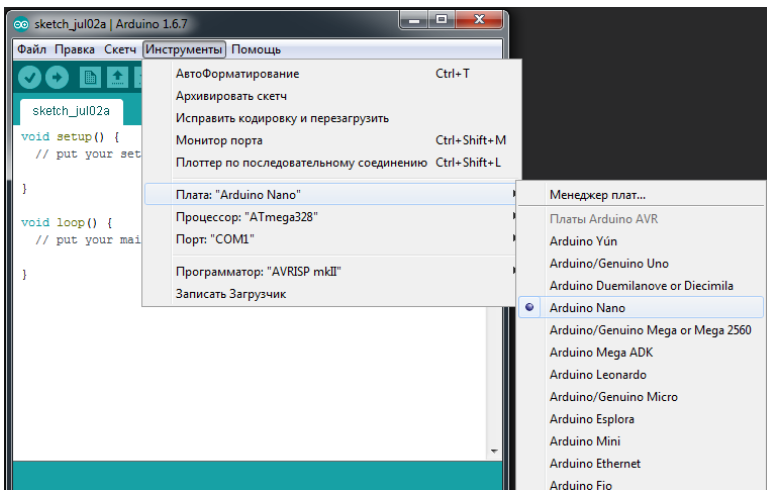


Рисунок 3.64 – Налаштування плати в Arduino IDE

Після підключення плати до комп'ютера з нею можна працювати та програмувати її. Для того, щоб прототип протезу руки виконував рухи, необхідно написати та завантажити на плату ПЗ (приклад коду наведено на рисунку 3.65).

```

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

void setup()
{
  Serial.begin(9600);
  myservo.attach(8);
  myservo2.attach(11);
  myservo3.attach(9);
  myservo4.attach(10);
  myservo5.attach(12);
}
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);
}
}

```

Рисунок 3.65 – Приклад коду

Розроблене ПЗ реалізує роботу п'яти пальців одночасно, тобто їх згинання та розгинання. Для початку роботи необхідно підключити бібліотеку Servo, яка 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 внести зміни в void loop (), тобто вказати значення відповідних пінів в ньому. Значення пінів для кожного з пальців були вказані в таблиці 3.10.

Алгоритм управління кіберпротезом верхньої кінцівки людини (рис. 3.66) складається з наступних етапів:

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.

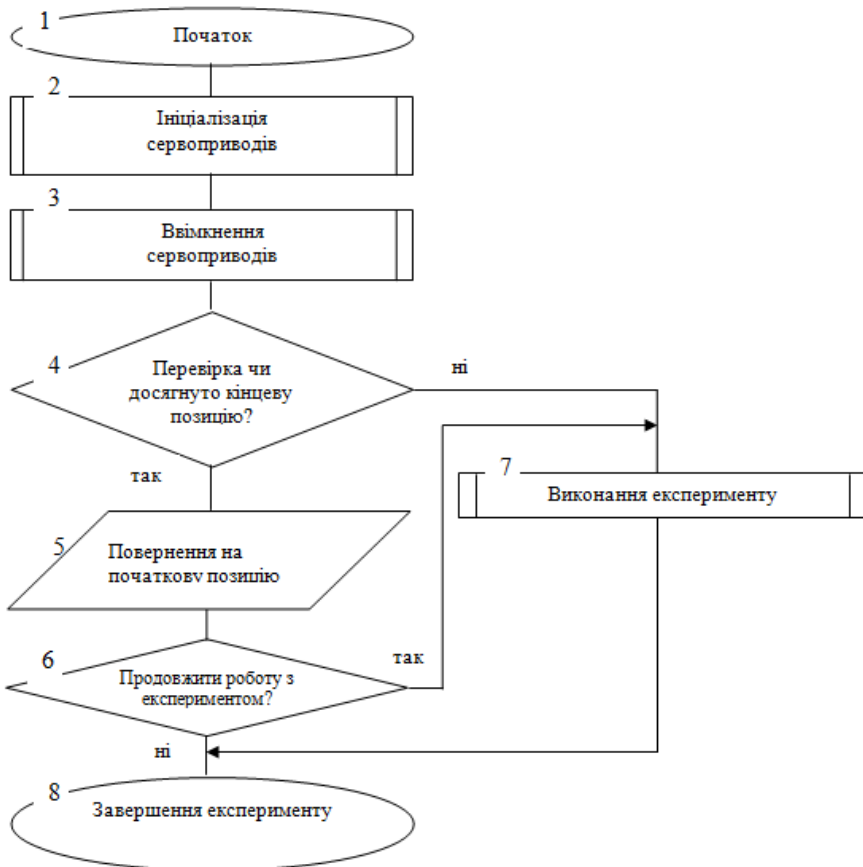


Рисунок 3.66 – Схема алгоритму програми управління кіберпротезом

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()`).

Тестування роботи програми виконувалося для різних сценаріїв: два пальці зігнуті, а три розігнуті (рис. 3.67); один палець зігнутий (рис. 3.68); всі пальці зігнуті (рис. 3.69); всі пальці розігнуті (рис. 3.70).



Рисунок 3.67 – Сценарій: два пальці зігнуті



Рисунок 3.68 – Сценарій: один палець зігнутий



Рисунок 3.69 – Сценарій: всі пальці зігнуті



Рисунок 3.70 – Сценарій: всі пальці розігнуті

Крім того, можливо виконати перевірку відповідності форми та розмірів кіберпротезу руці реальної людини на основі побудови візуальної моделі цифрового двійника в програмі Skanest (рис. 3.71).

Функціональне тестування підтвердило працездатність прототипу кіберпротезу, а порівняння з цифровим двійником – відповідність форми та розмірів.

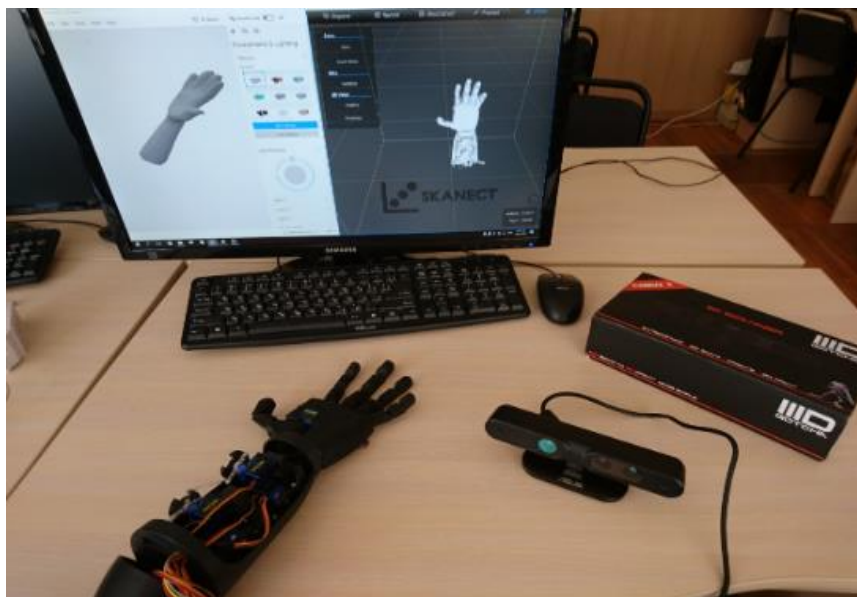


Рисунок 3.71 – Робота з цифровим двійником в програмі Skanect

### 3.5 Висновки до розділу 3

У даному розділі представлено 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.

### 3.6 Контрольні питання

1. Охарактеризуйте основні стадії та етапи інформаційної технології створення механічної частини кіберпротезу.
2. Що таке параметричне моделювання? Які типи параметричного моделювання існують?
3. Які основні проєктні операції використовуватися для побудови 3D моделі конструктиву кіберпротезу
4. Яким чином можна задати параметри моделі пальця і параметричні рівняння в середовищі Creo Parametric?
5. Як будується складальна одиниця на основі моделей пальців та долоні?
6. Файл з яким розширенням є вхідним для програми Simplify3D? Охарактеризуйте базові функції програми.
7. Який матеріал доцільно використати для виготовлення прототипу конструктиву кіберпротезу на 3D принтері?
8. Які різновиди плати Arduino можливо використати при прототипування кіберпротезу?
9. Що таке плата розширення та які її різновиди існують.
10. Опишіть технічні особливості та основи програмування серводвигунів.
11. Які основні етапи функціонування ПЗ для прототипу кіберпротезу.
12. Як використовується цифровий двійник?

### 3.7 Практичні завдання

*Завдання 1.* Виконайте побудову віртуального та фізичного прототипів кіберпротезу руки на основі методик цього розділу.

## 4 РОЗРОБКА ВІРТУАЛЬНОГО ПРОТОТИПУ СПІНАЛЬНОГО ІМПЛАНТА

### 4.1 Постановка проєктних завдань та вибір середовища розробки прототипу

Малорухомий спосіб життя або навпаки інтенсивні фізичні навантаження, травми, хронічні захворювання, надлишкова вага, неправильне положення тіла у підлітків, вікові зміни у літніх людей – все це часто призводить до патологічних процесів та захворювань хребта людини [71]. В той же час, завдяки новітнім засобам діагностики, а також розробці сучасних стабілізуючих біомедичних конструкцій, кількість спінальних операцій в світі неухильно зростає, а їх ефективність і безпека підвищуються з кожним роком [72].

Для забезпечення функціональної стабілізації хребта розроблено сучасні методики лікування на основі використання динамічних та ригідних систем від відомих компаній Paradigm Spine GmbH, HumanTech Germany GmbH, Ulrich Germany GmbH (Германия); Medyssey Co., Ltd. Diomedical Co., Ltd. (Южная Корея); Zimmer GmbH (Швейцария); SpinalKinetics, Inc. (США) та ін.. Зокрема, активно використовуються наступні біомедичні конструкції:

- ендопротези міжпозвонокового диску для імітації анатомічної структури та біомеханічних властивостей природнього диску;
- спінальні імпланти для динамічної стабілізації хребта;
- імпланти для транспедикулярної фіксації та багато інших [71].

Тим не менш, подальший розвиток хірургії хребта є актуальною задачею та вимагає розробки нових або модернізованих біомедичних конструкцій, що будуть відповідати індивідуальним потребам пацієнтів, сучасним тенденціям в медицині, а також світовим стандартам якості [73-74]. Через складність хребта людини, спінальні імпланти можуть відрізнятися за розмірами, формою та функціональністю [75].

Удосконалення існуючих спінальних імплантів має виконуватися з урахуванням появи сучасних матеріалів для їх

виготовлення, покращених технологій протезування, нових технологічних можливостей виробництва, розвинених технологій автоматизованого проектування та прототипування.

Як показали проведені дослідження, основними завданнями, що вирішуються розробниками біомедичних конструкцій є:

- забезпечення функціональності, біологічної сумісності з живими тканинами організму та зручності для пацієнта;
- гарантування конструкційної міцності імпланта для всіх варіантів його взаємодії з тілом або навколишнім середовищем;
- забезпечення зручного доступу хірургічного інструменту до потрібних точок конструкції імпланта для полегшення роботи лікарів в процесі протезування;
- урахування індивідуальних потреб пацієнта та конкретних особливостей його організму;
- забезпечення технологічності виробництва та зменшення собівартості продукції.

Проектування біомедичних конструкцій – це складний процес, що вимагає постійного урахування індивідуальних особливостей організму пацієнта, існуючих медичних показань та протипоказань. В той же час, для зниження собівартості виробництва, доцільно створювати загальну базову конструкцію, яка може бути представлена у вигляді дискретного розмірного ряду, а також унікальну індивідуальну частину, геометрія якої визначається фізичними особливостями будови тіла пацієнта.

Універсальні (PTC Creo, AutoCAD Mechanical Desktop, Solid Work та ін.) або спеціалізовані CAD-системи (наприклад, Autodesk Within Medical) (рис. 4.1) широко використовуються для створення 3D моделей біомедичних пристроїв та конструкцій в середовищах моделювання [5, 76-77].

Також, постійно розширюються можливості сучасних CAS-технологій (Computer Assisted Spine Surgery) на основі навігаційних (Medtronic Stealth, BrainLab, 7D Surgical та Stryker) та робототехнічних систем (Mazor Renaissance, MazorX, Globus Excelsius GPS та Brainlab Cirq.) [78-80].



Рисунок 4.1 – Класифікація середовищ розробки біомедичних пристроїв та конструкцій

Згідно з [49] РТС Creo є одним з найкращих середовищ параметричного моделювання. Система має потужний функціонал та надає інтелектуальне середовище проектування. Дослідження показали, що важливими для проектування біомедичних конструкцій функціями системи Creo є:

- твердотільне 3D моделювання.
- пряме моделювання.
- функції аналізу.
- модельно-орієнтоване проектування.
- робота з імпортованими сторонніми даними.
- розробка власного ПЗ.

Саме тому, ця система може бути рекомендована в якості основи для створення інтегрованого середовища розробки біомедичних конструкцій. Актуальною задачею є організація інтегрованого середовища розробки з використанням параметричного моделювання та програмування для підвищення ефективності процесу автоматизованого проектування індивідуальної конструкції спінального імпланта.

## 4.2 Аналіз конструктивних та функціональних особливостей спінального імпланта

Coflex® – імплант у формі латинської букви «U», розроблений на початку 90-х років французьким ортопедом Жаком Самані з міста Ліон для функціональної динамічної стабілізації (підтримки) хребетного стовпа після декомпресійної хірургії (рис. 4.2) [81].

Технологію Coflex® використовують в 60 різних країнах, більш ніж 163 000 пацієнтів, що підтверджує її надійність, безпеку і ефективність [82].

Процедура установки імпланта Coflex виконується малоінвазивним способом, коли хірург робить лише невеликий розріз шкіри, зберігаючи м'язи і зв'язки недоторканими (рис. 4.3) [81].



Рисунок 4.2 – Спінальний імплант Coflex [81]

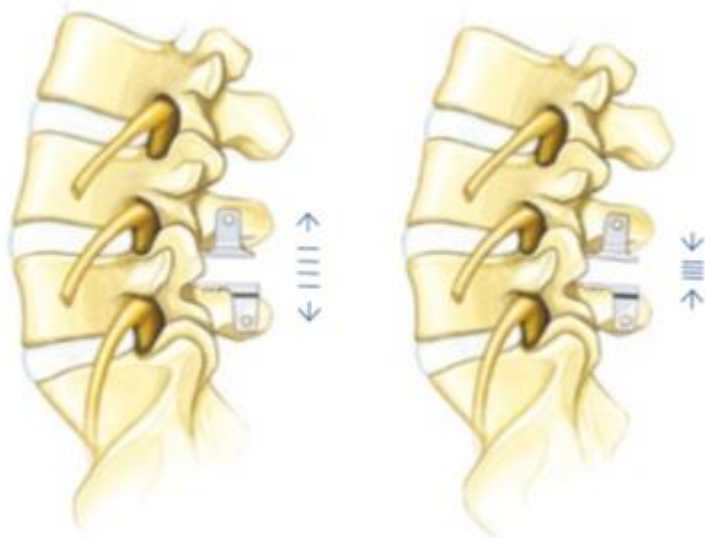


Рисунок 4.3 – Встановлення імпланта Coflex [81]

Цей імплант призначений для вирішення декількох важливих задач.

Перша з них – це стабілізація/підтримка. В результаті операції декомпресії часто розвивається втрата стійкості в хребетному стовпі. Однак ця втрата може бути успішно компенсована установкою імпланта Coflex. Імплантація Coflex також дозволяє і хірургу, і пацієнту бути впевненими в збереженні нормальної висоти прооперованого сегмента. Хребетні суглоби ефективно розвантажуються, нервові волокна/корінці, що подразнюються здавленням, вивільнюються і набувають надійний захист.

Друга функція – збереження/підтримання рухливості. На додаток до стабілізації функція підтримки рухливості є дуже важливою для запобігання розвитку дегенеративних змін в суміжних сегментах. Коли пацієнт відхиляється назад, імплант Coflex пружно стискається, рівномірно перерозподіляючи навантаження по хребцях, обмежуючи тиск на фасеточні суглоби і зменшуючи грижове випинання диска [81].

Імплант Кофлекс виготовляється зі сплаву титану класу 5 з маркуванням Ti6Al4V, що складається з титану (89.6%), алюмінію

(6%), ванадію (4%), заліза (0.25%), різних окислів (0.2%). Така комбінація титану та легуючих добавок робить сплав виключно міцним і стійким до деформацій. Цей матеріал довів свою надійність протягом багатьох років випробувань в медицині і володіє чудовою біосумісністю.

Імплант Кофлекс виготовляється шляхом фрезерування виробу з єдиного полікристалу з хаотичним розташуванням зерен. Тому виріб, що отримується, позбавлений слабких місць щодо можливих непередбачених деформацій. Тривалі динамічні випробування імпланта на стенді показують, що навантаження в 15 кг на 1 см<sup>2</sup>, повторене 10 мільйонів разів (моделювання експлуатації терміном в 30 років), не призводить до руйнування або незворотних деформацій імпланта.

Імплант має п'ять стандартних розмірів залежно від міжкостистої відстані (8, 10, 12, 14, 16 мм) (рис. 4.4). Однак варіації розмірів інших частин імпланта не передбачені для повного врахування індивідуальних особливостей пацієнта [83].

Таким чином, популярність імпланта Soflex® та описаний набір його особливостей обумовили вибір цього біомедичного пристрою як об'єкта проєктування в цьому розділі.

#### coflex-F® Implant



Color Code	Size	Article Number
	16 mm	RPI 00016
	14 mm	RPI 00014
	12 mm	RPI 00012
	10 mm	RPI 00010
	8 mm	RPI 00008

Рисунок 4.4 – Стандартні типорозміри спінального імпланта Soflex

### 4.3 Інтегроване середовище розробки віртуального прототипу спінального імпланта

Діаграму діяльності розробника індивідуального спінального імпланта представлено на рисунку 4.5.

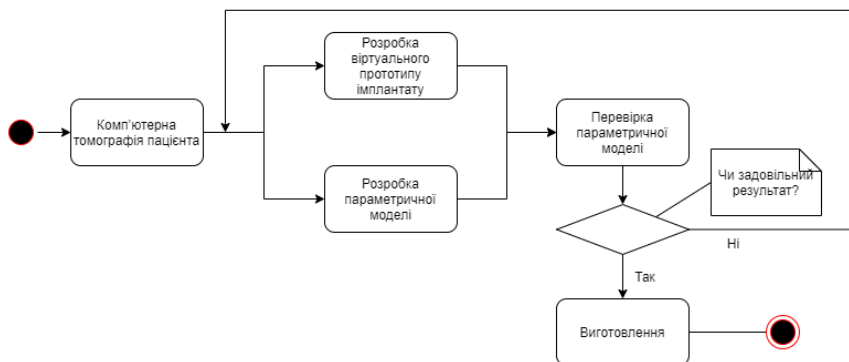


Рисунок 4.5 – Діаграма діяльності розробника індивідуального спінального імпланта

На першому етапі вхідні дані для проєктування індивідуального імпланта можна отримати за допомогою комп'ютерної томографії пацієнта. Виявлення та аналіз порушень проводить лікар. Він приймає рішення про особливості протезування та формулює вимоги щодо модифікації базової конструкції імпланта хребта.

Другий етап процесу проєктування – розробка віртуального прототипу спінального імпланта в середовищі тривимірного моделювання. Наступним етапом є створення параметричної 3D моделі для подальшої регенерації індивідуального дизайну відповідно до значень параметрів, отриманих від лікаря. Після верифікації параметричної моделі її можна застосовувати для проєктування окремого імпланта хребта.

Архітектура інтегрованого середовища розробки базується на організації взаємодії Creo Parametric з інструментами Excel Analysis та Program (рис. 4.6).

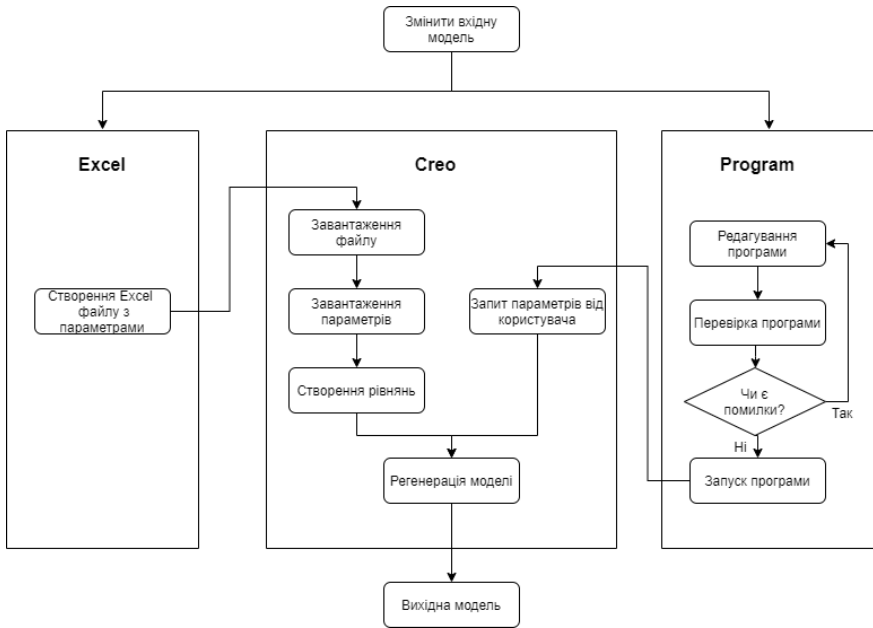


Рисунок 4.6 – Архітектура інтегрованого середовища розробки

Файл Excel містить інформацію про параметри базового проектування (рис. 4.7) для подальших змін. Набір параметрів електронної таблиці такий (рис. 4.8):

WIDTH – ширина основної частини деталі;

WIDTH\_TOP – ширина верхньої частини деталі;

FLANGE\_TOP – висота верхньої частини деталі;

FLANGE\_BOTTOM – висота нижньої частини деталі;

WIDTH\_BOTTOM – ширина відступів для нижньої частини деталі.

Дані комірок створеного файлу Excel завантажуються в середовище Creo Parametric за допомогою інструмента аналізу Excel (рис. 4.9). Створюються відношення між параметрами моделі та значеннями електронної таблиці Excel. Важливим етапом є перевірка створених відношень між 3D моделлю та файлом Excel та їх створення відповідно до порядку в таблиці Excel.

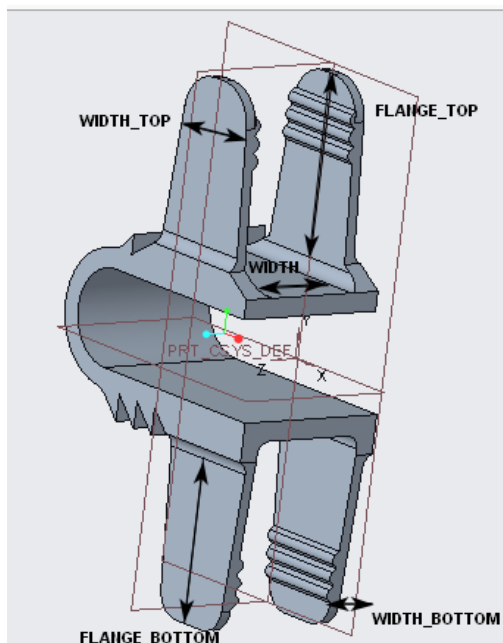


Рисунок 4.7 – Основні параметри 3D моделі базової конструкції

	A	B	C
1	WIDTH	8,00	
2	WIDTH_TC	2,00	
3	FLANGE_T	18,00	
4	FLANGE_E	20,00	
5	WIDTH_B	3,00	
6			
7			

Рисунок 4.8 – Опис параметрів частин імпланту в Excel таблиці

У результаті будується параметричний опис моделі (рис. 4.10). Після цього можна регенерувати модель відповідно до заданих параметрів. Результати побудови базової та індивідуальної конструкцій спінального імпланту наведено на рисунку 4.11.

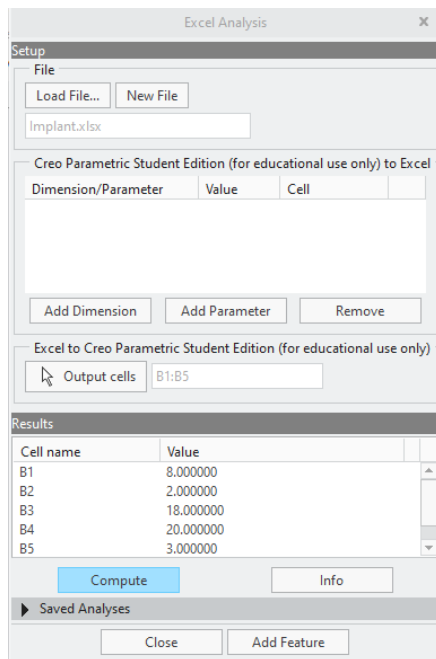


Рисунок 4.9 – Чисельні результати обробки даних за допомогою інструменту аналізу Excel

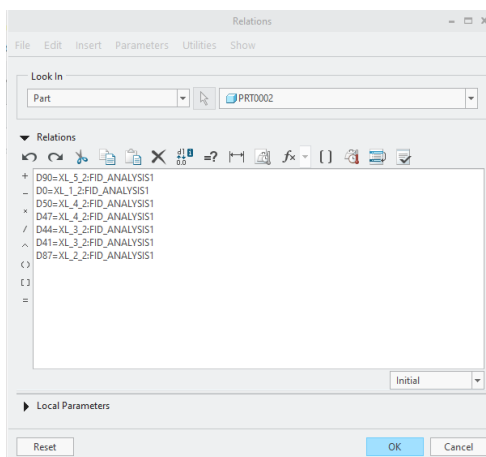


Рисунок 4.10 – Параметричний опис моделі

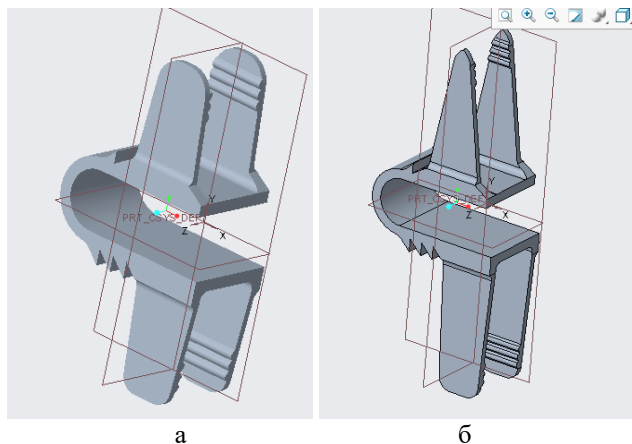


Рисунок 4.11 – Базова модель (а) та індивідуальна конструкція (б) спінального імпланта

Інструмент Program дозволяє розробити параметризований опис унікальних частин імпланта хребта (наприклад, масиву зубів) для швидкого редагування їх геометрії. Для взаємодії з проектувальником створюється блок із запитом щодо побудови масиву зубців з подальшим введенням їх розмірів (висота, ширина і відстань між ними).

Фрагмент програмного коду для побудови унікальної частини імпланта показаний на рисунку 4.12. Змінна D117 відповідає за висоту елемента, D118 - за ширину елемента, D119 - за відстань між елементами.

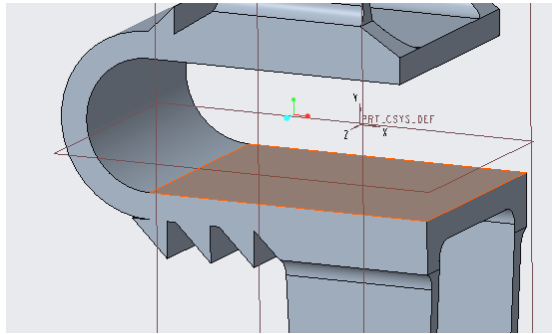
```

INPUT
  ADD_PATTERN STRING
  "Do you want add the pattern? (y/n)"
  IF ADD_PATTERN == "y"
    D117 NUMBER
    "Add height:"
    D118 NUMBER
    "Add width:"
    D119 NUMBER
    "Add dimension:"
  END IF
END INPUT

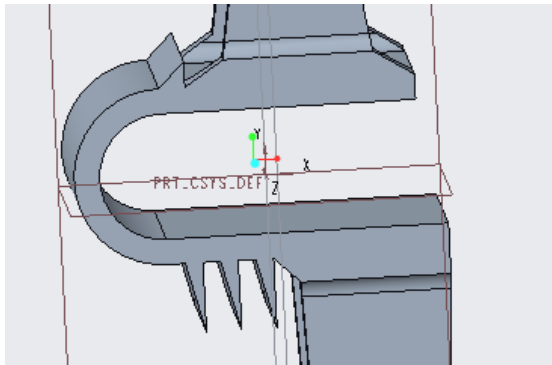
```

Рисунок 4.12 – Фрагмент коду розробленої програми

Результат регенерації 3D моделі імпланта з унікальною частиною наведено на рисунку 4.13.



а



б

Рисунок 4.13 – Вихідна геометрія зубців (а) та результат роботи програми (б)

Впровадження інтегрованого середовища розробки біомедичних пристроїв, заснованого на взаємодії інструментів Creo Parametric, Excel Analysis та Program, дозволяє підвищити ефективність процесу автоматизованого проектування спінальних імплантів з урахуванням індивідуальних особливостей пацієнтів. Запропонований підхід відкриває шляхи для подальшого покращення форми та масогабаритних параметрів імплантів хребта, і його можна рекомендувати для розробки інших біомедичних пристроїв.

#### 4.4 Основні етапи побудови базової 3D моделі спінального імпланту в середовищі Creo Parametric

Вхідними даними для розробки є кресленик U-імпланту Coflex® для хребта людини. Головний вид деталі наведено на рисунку 4.14, вид зверху – на рисунку 4.15.

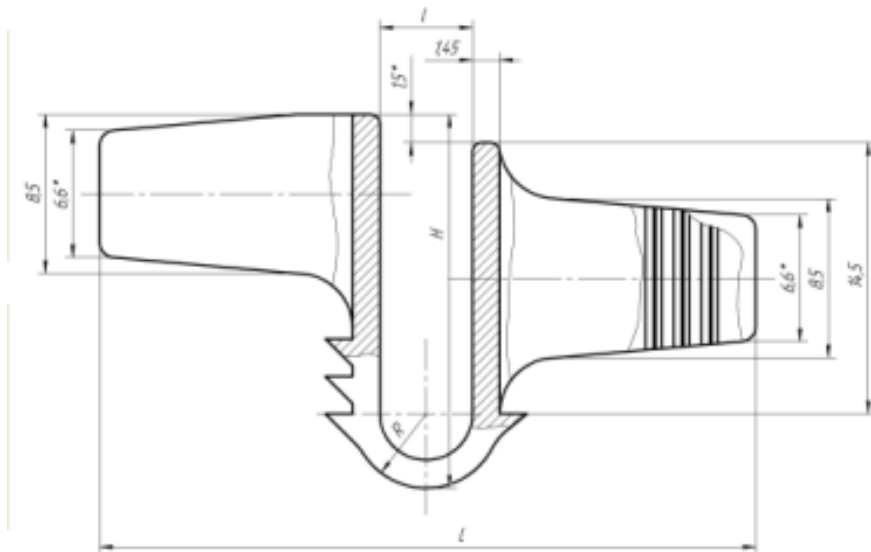


Рисунок 4.14 – Головний вид деталі

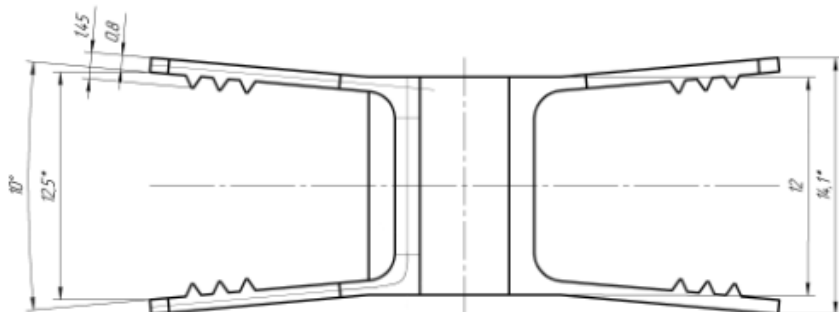


Рисунок 4.15 – Вид зверху деталі

Основні етапи процес побудови моделі спінального імпланту представлено на рисунках 4.16 - 4.30.

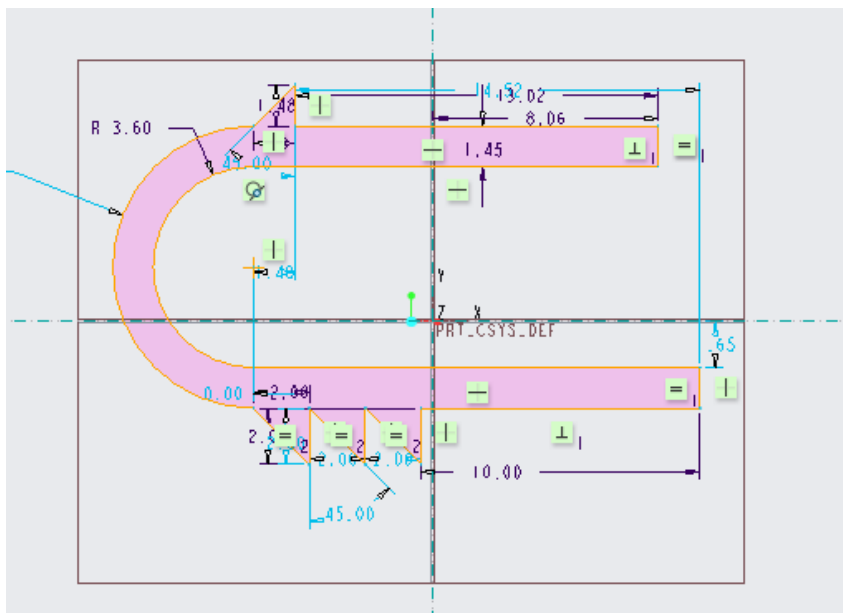


Рисунок 4.16 – Створення основного ескізу моделі

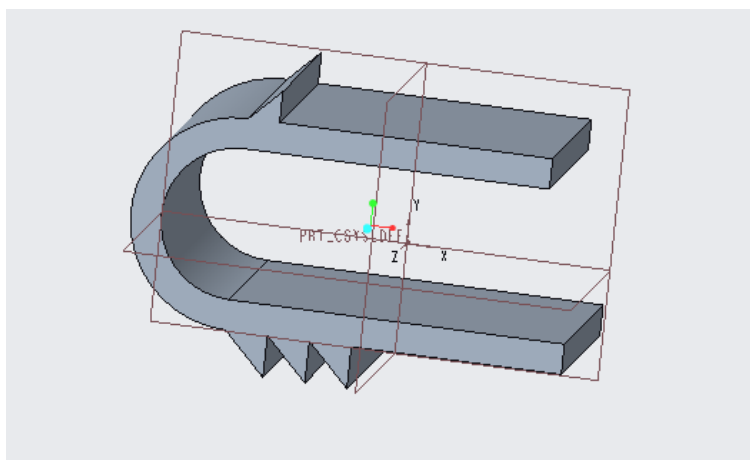


Рисунок 4.17 – Витягування ескізу основної частини

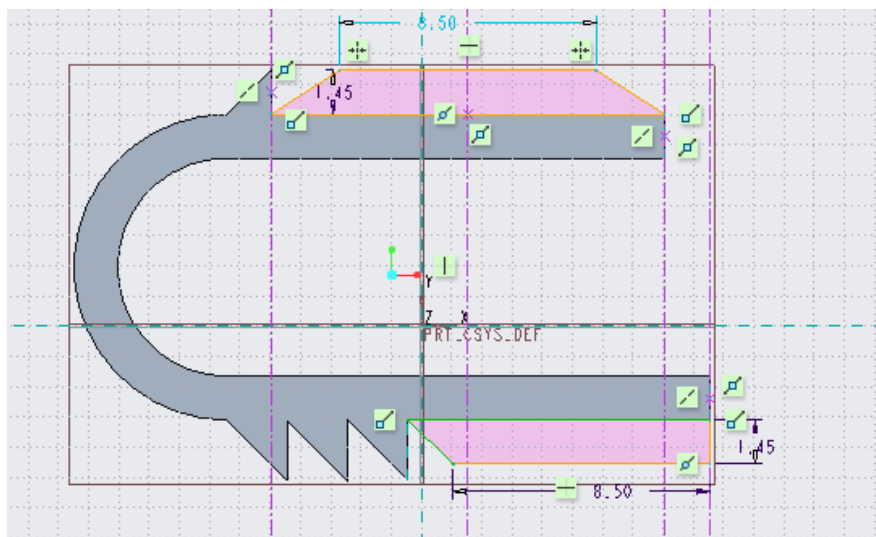


Рисунок 4.18 – Додавання ескізу бокових виступів

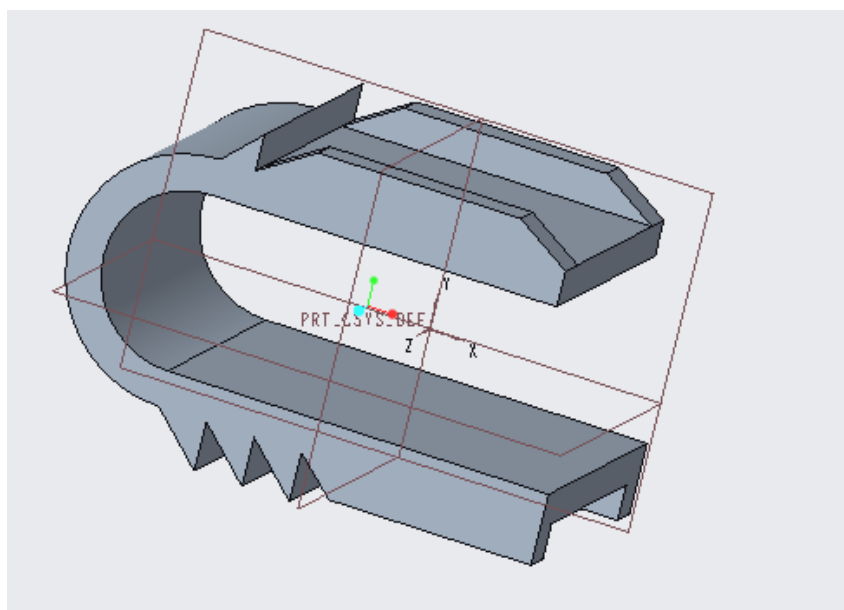


Рисунок 4.19 – Вигляд моделі після додавання усіх бокових виступів

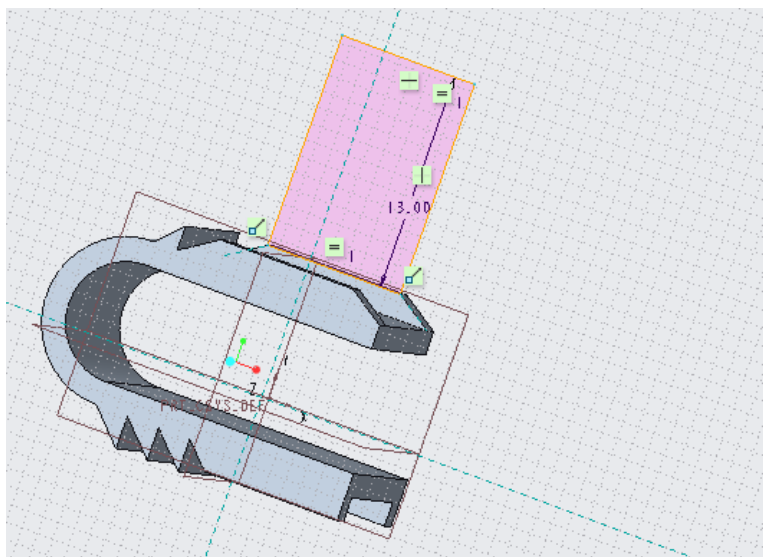


Рисунок 4.20 – Створення ескізу бокових частин

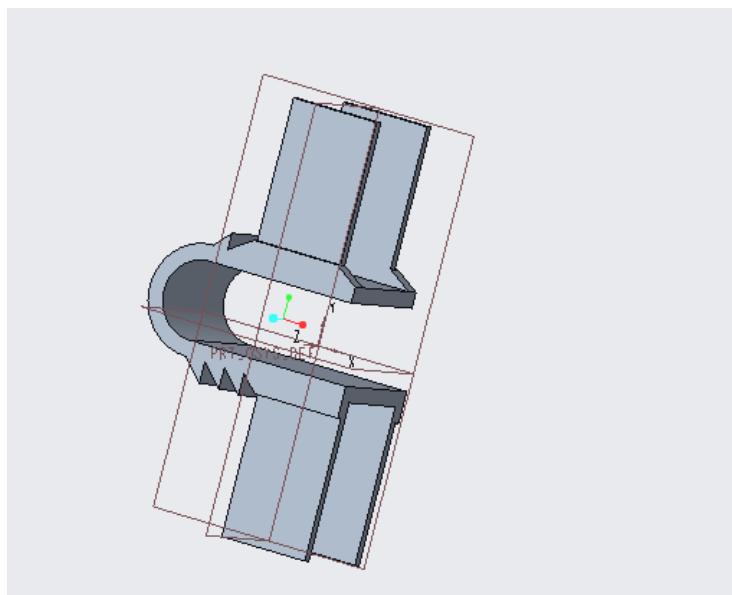


Рисунок 4.21 – Вигляд моделі після додавання усіх бокових частин

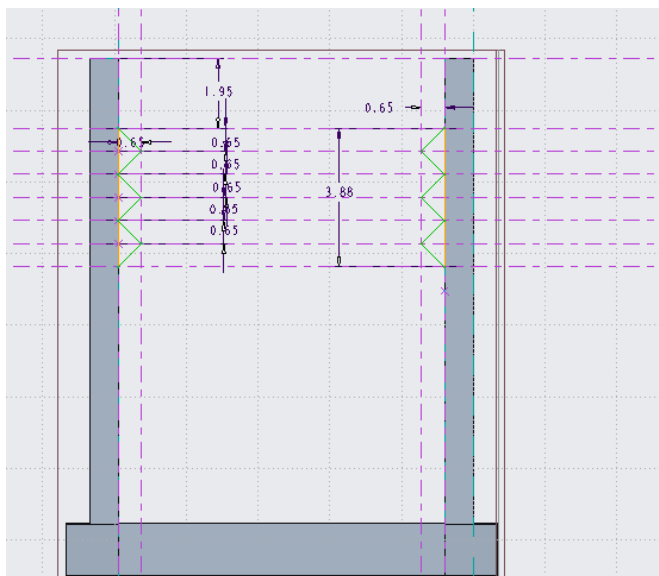


Рисунок 4.22 – Створення різьблення на боковій частині

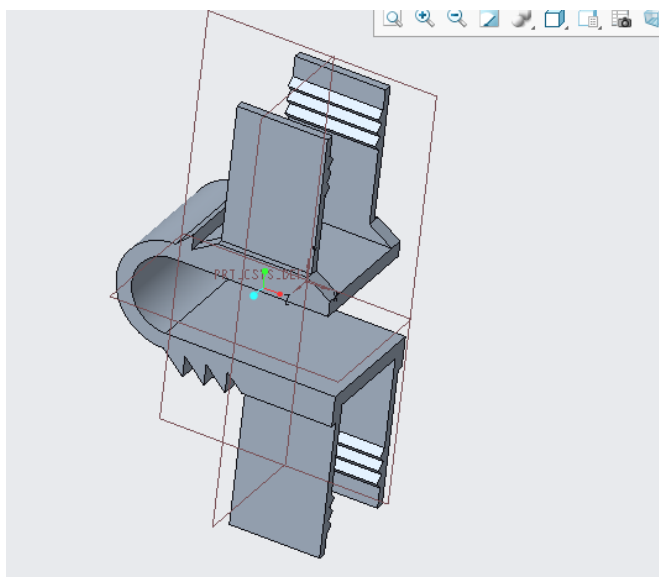


Рисунок 4.23 – Вигляд моделі після додавання різьблення

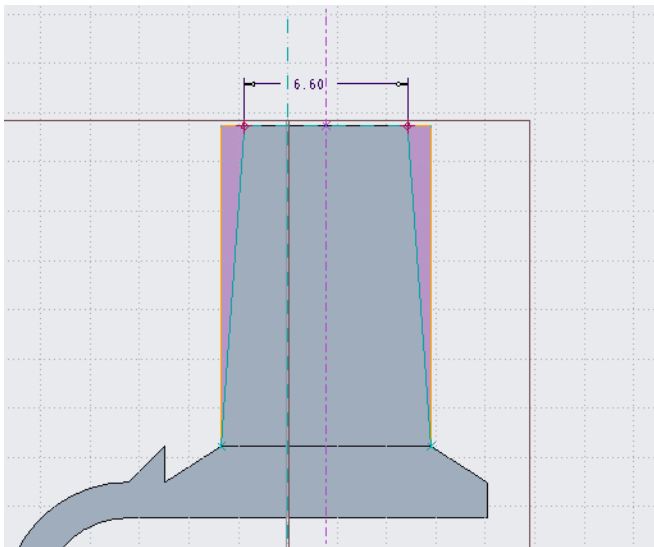


Рисунок 4.24 – Надання форми боковій частині

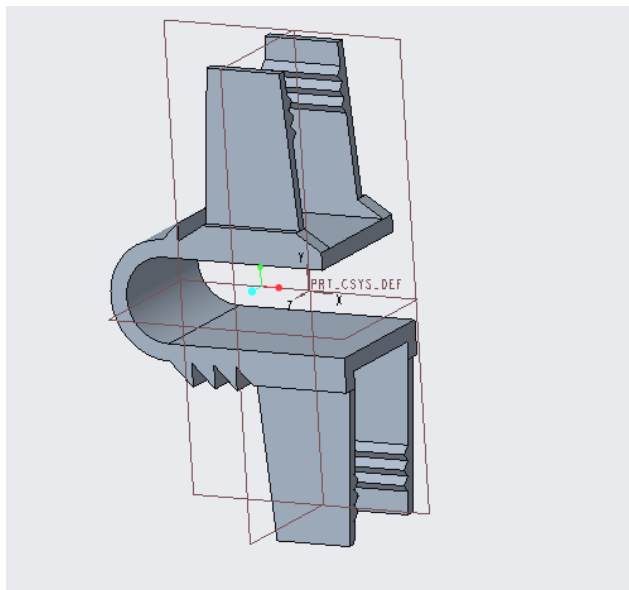


Рисунок 4.25 – Вигляд моделі після надання форми боковим частинам

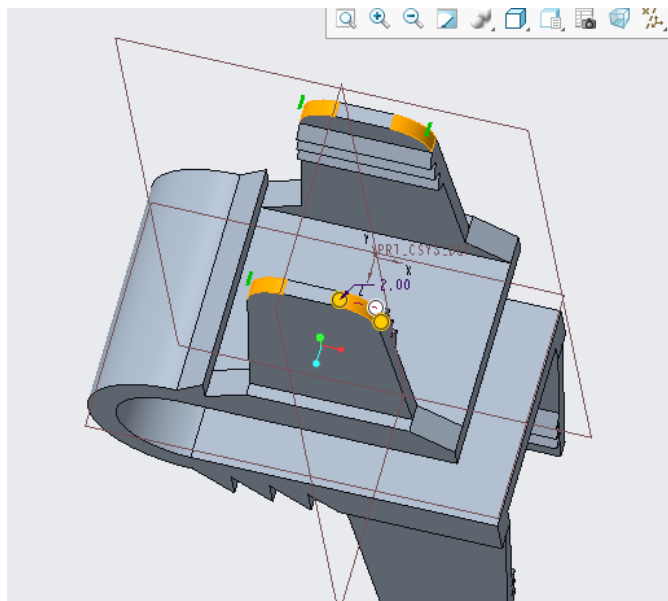


Рисунок 4.26 – Створення заокруглень для бокових частин

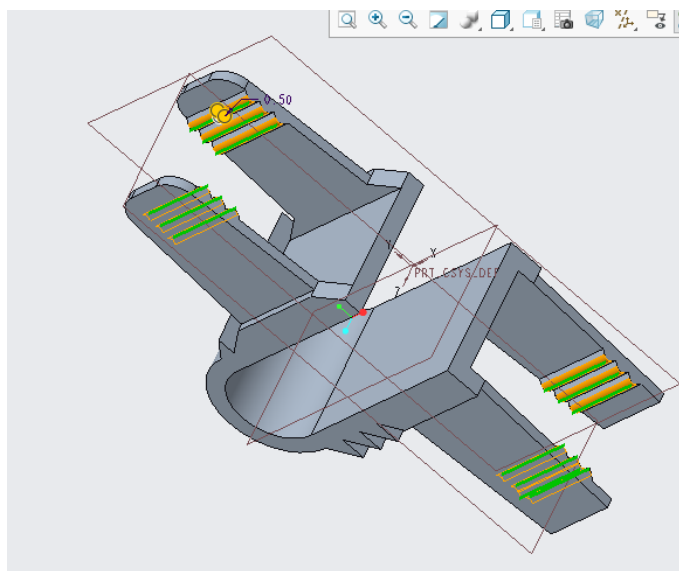


Рисунок 4.27 – Створення заокруглень для різьблення

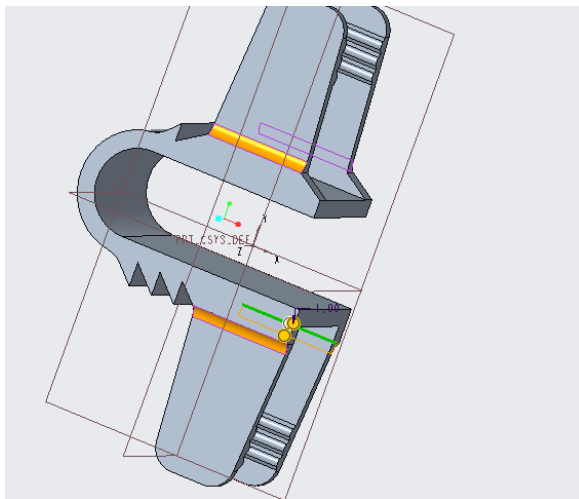


Рисунок 4.28 – Створення заокруглень для зовнішньої сторони бокових частин

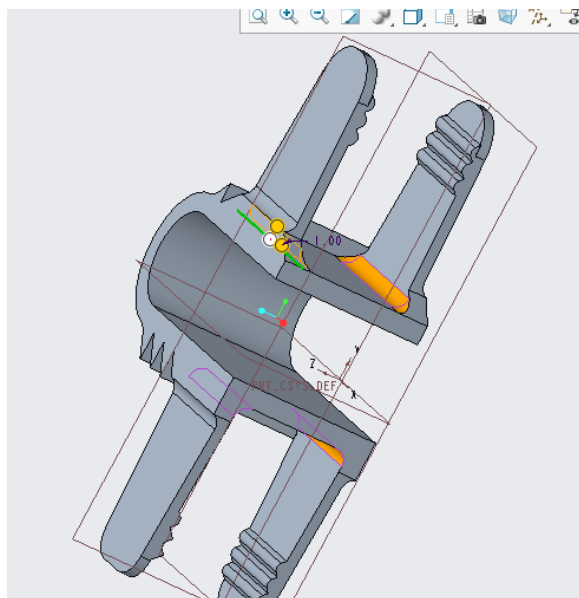


Рисунок 4.29 – Створення заокруглень для внутрішньої сторони бокових частин

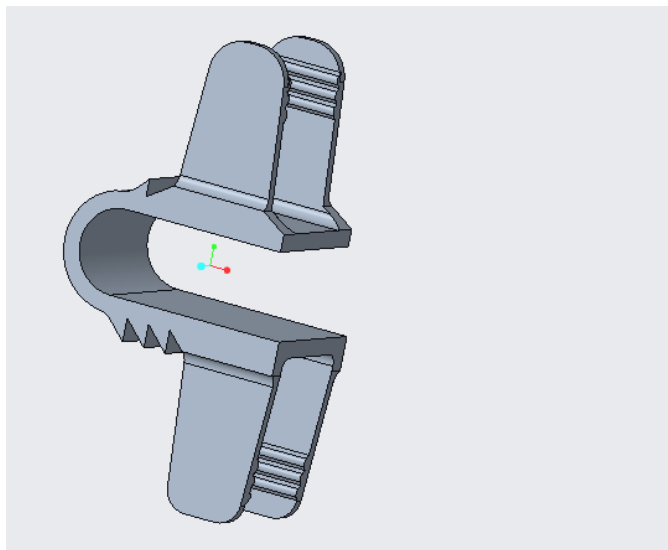


Рисунок 4.30 – Вигляд готової моделі

## 4.5 Практичне використання інструментів Program та Excel Analysis для роботи з параметричною моделлю в Creo

### 4.5.1 Робота з інструментом Excel Analysis

Створити нову твердотільну модель: FILE → NEW.

У вікні New обрати: PART(Деталь) та тип SOLID (Тверде тіло).

Зняти чекбокс з Use default template.

Задати ім'я файлу: main.

Клікнути кнопку ОК.

У вікні New File Option обрати шаблон: mmns\_part\_solid.

У вікні моделі що відкрилось ввімкнути відображення назв площин, для цього в панелі інструментів обрати інструмент: View → Show → Plan Tag Display.

Виконати ескіз для видавлювання основної частини:

Model→Shapes→Extrude

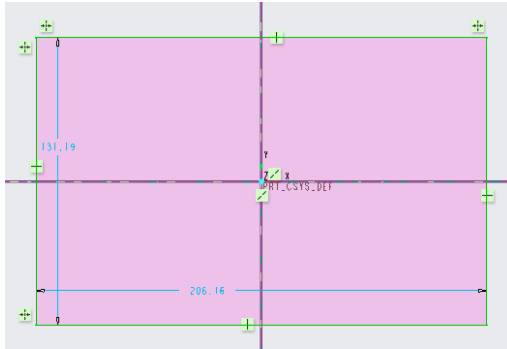
<ПКМ> Define Internal Sketch (Задати внутрішній ескіз).

Обирати FRONT-поверхню, як Поверхню ескізу.

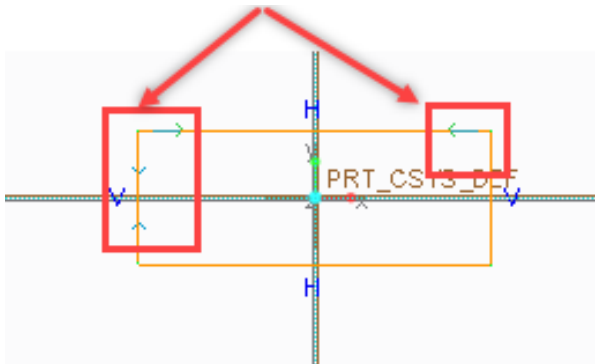
Натиснути кнопку Sketch(Ескіз) у вікні Sketch.

За допомогою інструменту Sketch→Sketching→ Centerline накреслити дві центральні лінії на перетині поверхонь Top та Right (див. рис. 4.31, а).

За допомогою інструменту Sketch→Sketching→ Rectangle накреслити довільний прямокутник (рис.4.31, а) таким чином, щоб він був симетричний відносно центральних ліній (у протилежних кутах фігури з'являться позначки, рис. 4.31, б).



а



б

Рисунок 4.31 – Побудова прямокутника (а) та позначки (б)

В панелі інструментів на табі Sketch натиснути кнопку ОК. В табі Extrude також натиснути Apply (прапорець). Як результат буде побудовано паралелепіпед довільних розмірів (рис.4.32). Зберегти файл (main.prt).

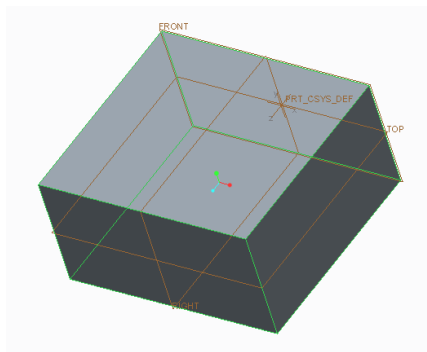


Рисунок 4.32 – Побудова паралелепіпеда

Створити файл Excel, зберегти його в директорії, де зберегли побудовану модель main.prt (1. Не обов'язково у директорію, де зберігається деталь. 2. Важливо, щоб шлях до файлу містив виключно англійські літери. Якщо папка містить українські або російські літери, то Creo не зможе завантажити файл.). Далі заповнити таблицю так як вказано на рисунку 4.33.

	A	B	C	D
1	HIGH	1,45		
2	LENGTH	53,1		
3	WIDTH	12		
4				
5				

Рисунок 4.33 – Створення Excel таблиці

В середовищі Creo в панелі інструментів обрати інструмент Analysis->Custom->Excel Analysis. У вікні Excel Analysis (рис. 4.34) натиснути кнопку Load File... та вказати шлях до створеного файлу Excel. Після вибору цього файлу він повинен відкритися (закривати його не потрібно). Перейти в середовищі Creo до вікна Excel Analysis (рис. 4.34) та натиснути кнопку Output cells, у файлі Excel виділити комірки B1:B3.

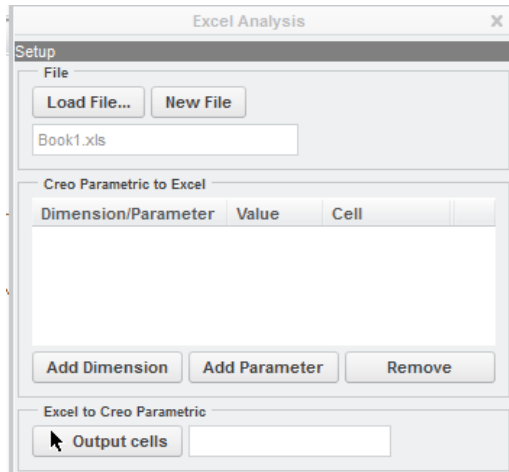


Рисунок 4.34 – Завантаження Excel файлу

Повернутися до середовища Creo, де повинно з'явитися вікно Menu Manager (рис. 4.35), у якому обрати пункт меню Done Sel.

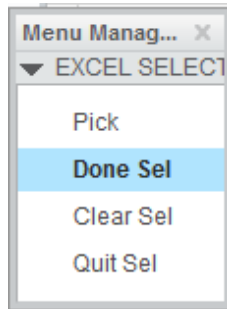


Рисунок 4.35 – Завершення вибору комірок

Щоб завантажити данні обраних комірок в Creo, необхідно у вікні Excel Analysis натиснути кнопку Compute (рис. 4.36). Після чого дані відобразилися в області Results.

Далі натиснути Add Feature, з'явиться підказка (рис. 4.37), в якій відображається назва Excel аналізу, який буде додано до моделі, в цій підказці натиснути прапорець.

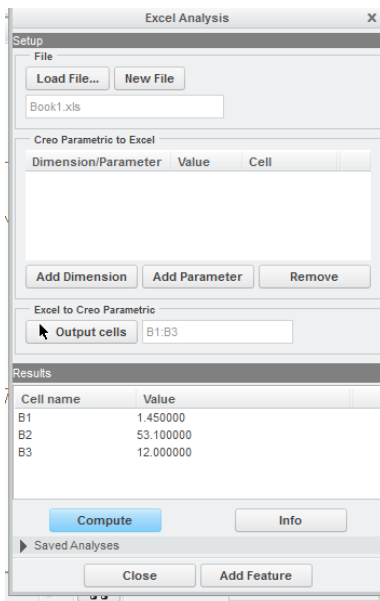


Рисунок 4.36 – Завантаження даних

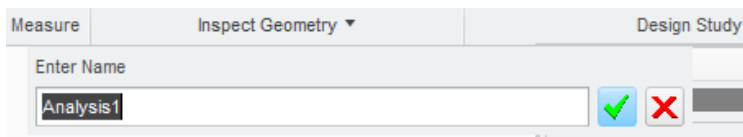


Рисунок 4.37 – Створення назви Excel-аналізу

В дереві моделі з'явиться новий елемент під назвою Analysis1. Закрити вікно Excel Analysis.

Далі створити відношення між параметрами моделі та значеннями у таблиці. Для цього необхідно перейти в панелі інструментів Tools->Model Intent->Relations.

Натиснути лівою кнопкою миші на 3D модель створеного паралелепіпеда, на ній з'являться умовні позначки розмірів (якщо модель була обрана до того як обрали інструмент Relations, то ці позначки з'являться одразу).

У вікні Relations, яке відкрилося написати "d1=" та перейти Insert->From List (рис. 4.38).

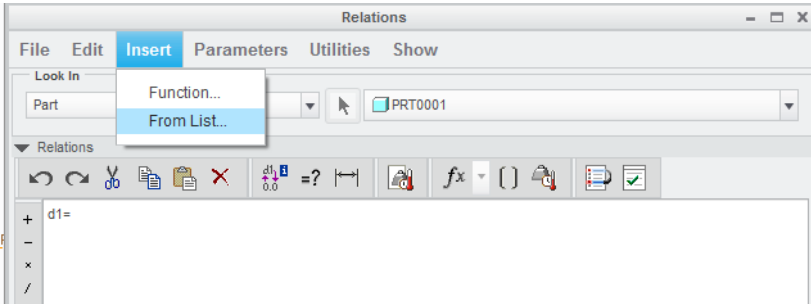


Рисунок 4.38 – Створення відношень між параметрами моделі та значеннями у таблиці

У вікні Select Parameter обрати Look In – Feature та обрати Analysis, (якщо немає параметру Feature, замість цього оберіть: Пошук в – Деталь, Фільтр по – Поточний та всі підрівні). Після цього у вікні з'являться значення з таблиці Excel. Обрати необхідне значення зі списку та натиснути кнопку Insert Selected (рис. 4.39).

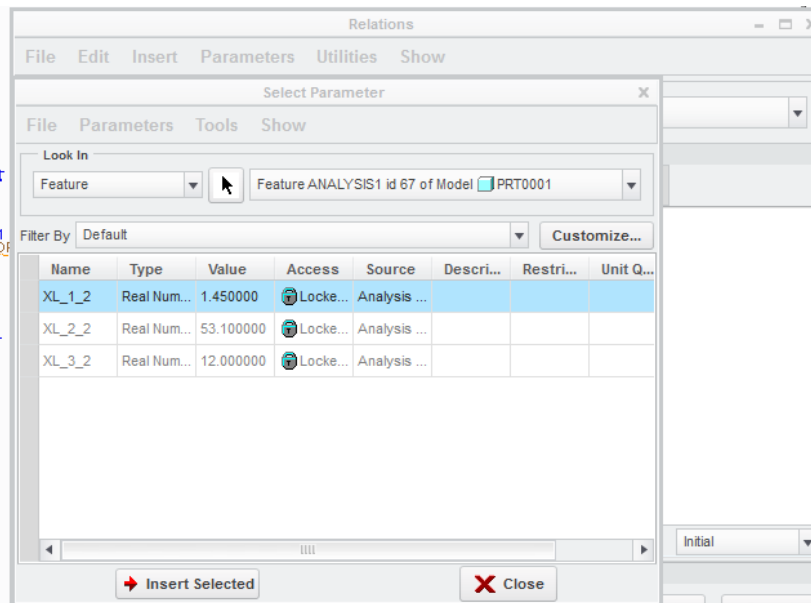


Рисунок 4.39 – Вибір параметрів

Повторити дії для інших двох параметрів, у результаті чого буде отримано три рівняння (рис. 4.40).

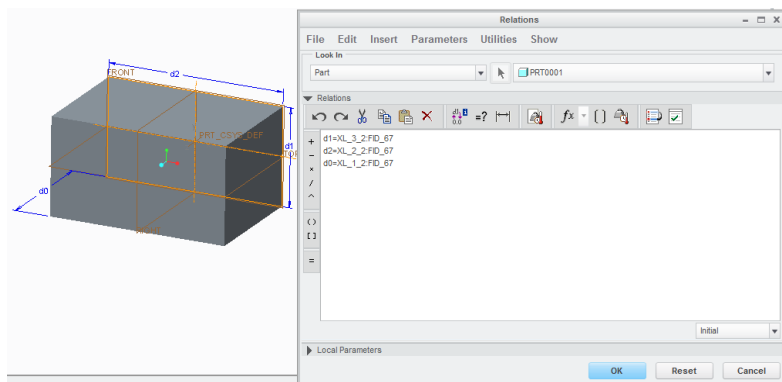




Рисунок 4.40 – Отримані рівняння

Для завершення створення відношень між моделлю та файлом Excel, в панелі інструментів вікна Relations обрати спочатку Execute/Verify Relations , підтвердити дії у вікні що з'явилося, після чого натиснути кнопку Sort Relations  для того, щоб рівняння відповідали сортуванню, яке задано в таблиці Excel і також підтвердити сортування.

Після чого у вікні Relation натиснути кнопку OK. Вікно закрилось у середовищі Creo, для регенерації моделі натиснути сполучення клавіш Ctrl+G. Результат на рисунку 4.41. Таким чином були створені відношення параметрів моделі та даних у файлі Excel. Щоб змінити параметри моделі, треба змінити значення в комірках Excel файлу, зберегти та у вікні Creo натиснути Ctrl+G.

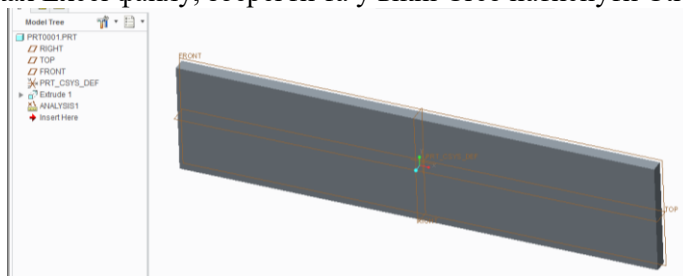


Рисунок 4.41 – Результат регенерації моделі

## 4.5.2 Побудова масиву елементів

Наступний крок – побудова масиву елементів.

В панелі інструментів обрати Model→Shapes→Extrude.

<ПКМ> Define Internal Sketch (Задати внутрішній ескіз).

Обираємо top поверхню деталі, як Поверхню ескізу.

Натискаємо кнопку Sketch (Ескіз) у вікні Sketch.

Для побудови додаткового посилання (з лівого краю моделі блакитна перпендикулярна до моделі пунктирна лінія), як зображено на рисунку 4.42, треба використати інструмент Sketch->Setup->References. Відкриється вікно References, на моделі клікнути по лівому краю моделі, на самій моделі з'явиться пунктирна лінія та у вікні References з'явиться назва посилання. Закрити вікно References.

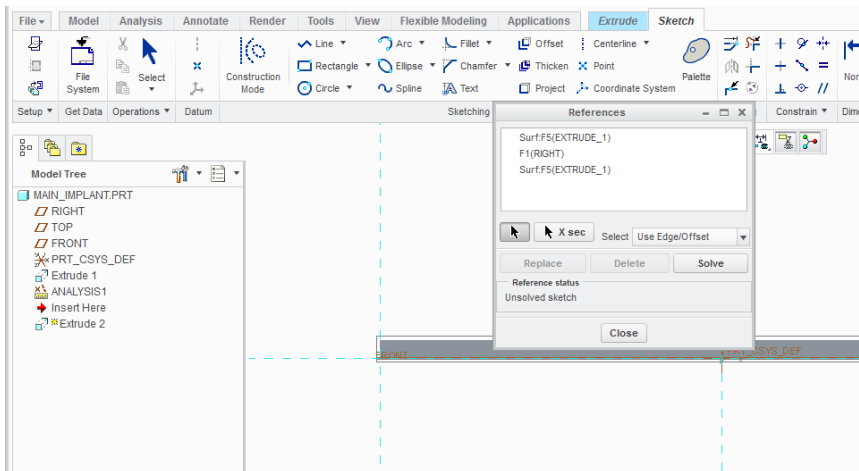


Рисунок 4.42 – Побудова додаткового посилання

Перейти в ескіз, накреслити контур елемента видавлювання за допомогою інструментів Centerline та Line, як зображено на рисунку 4.43.

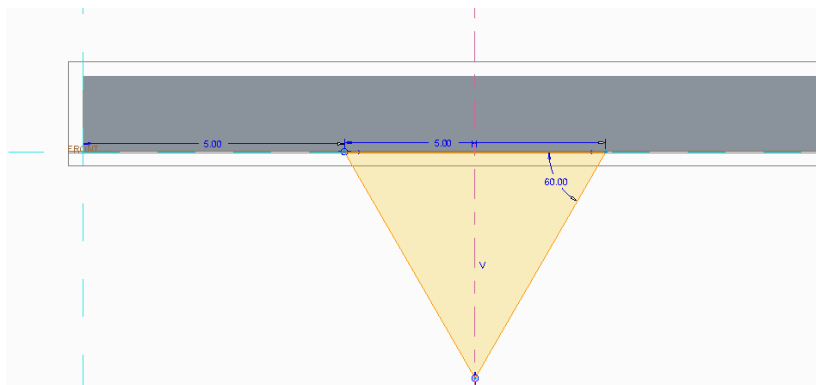



Рисунок 4.43 – Побудова контуру елемента видавлювання

В ескізі використовуючи інструмент Normal  встановити розміри, як зображено на рисунку 4.43. Щоб встановити розмір, треба ЛКМ натиснути на перший елемент потім на другий елемент і СКМ натиснути між цими елементами.

В панелі інструментів на табі Sketch клікнути ОК. На табі Extrude встановити параметр «Extrude to selected point, curve, plane or surface», після чого обрати протилежну поверхню деталі до якої буде виконуватись видавлювання елемента трикутника (див. рис. 4.44). Отриманий результат зображено на рисунку 4.45.

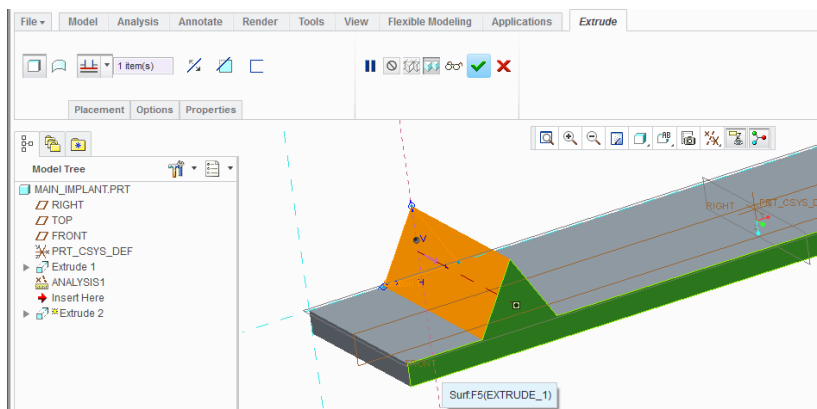


Рисунок 4.44 – Видавлювання елемента

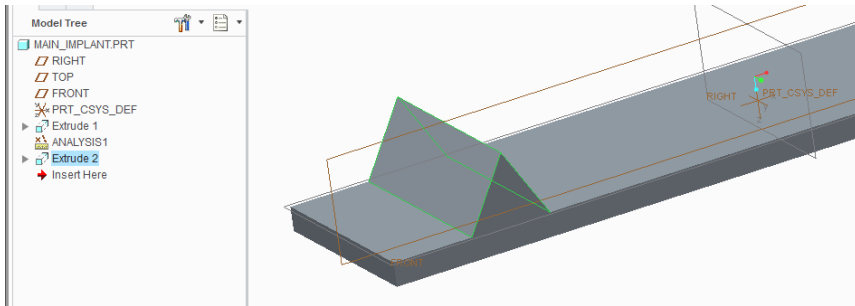


Рисунок 4.45 – Результат побудови елемента

Щоб побудувати масив елементів, необхідно виділити елемент в дереві моделі, який необхідно розмножити (Extrude2) (див. рис. 4.45), та обрати інструмент Model->Editing->Pattern.

В панелі інструментів в табі Pattern встановити параметр Detention, та обрати розмір як зображено на рисунку 4.46 та змінити його значення з 5.0 на 7.0. Це означає, що відстань між елементами масиву буде 7 мм.

Далі, треба змінити кількість елементів масиву на 3 (див. рис. 4.47).

Натиснути ОК. Результат зображено на рисунку 4.48.

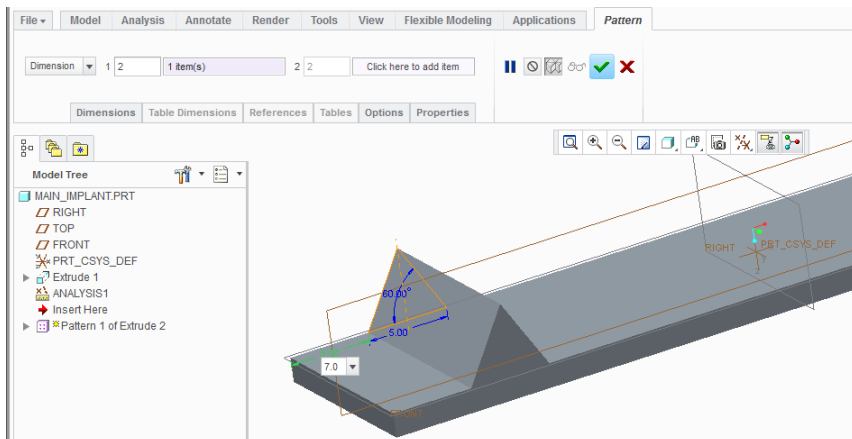


Рисунок 4.46 – Встановлення розмірів для масиву елементів

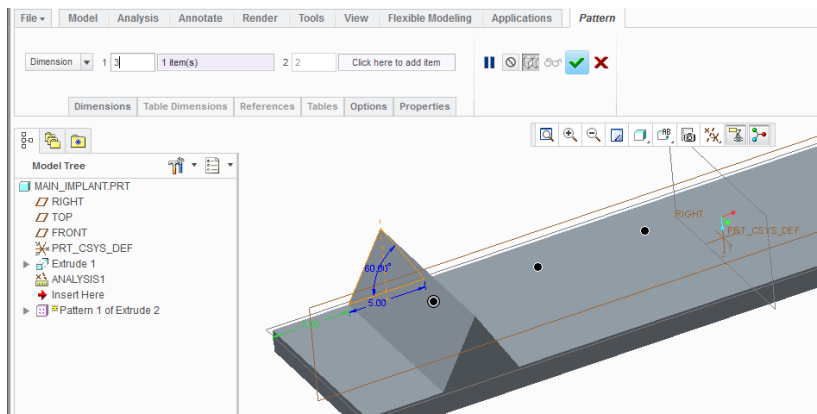


Рисунок 4.47 – Зміна кількості елементів масиву

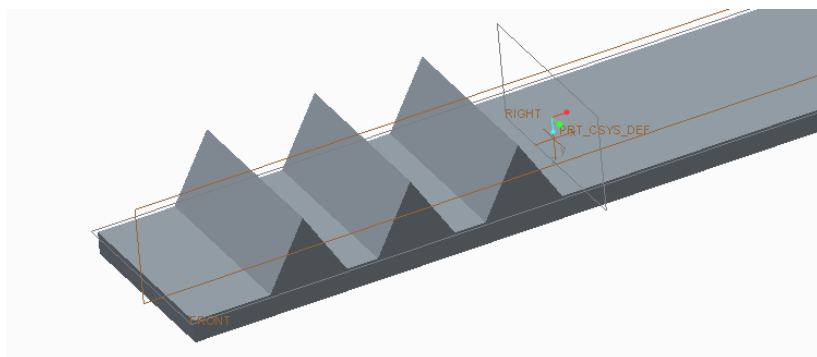


Рисунок 4.48 – Результат створення масиву елементів

### 4.5.3 Робота з програмою параметричної моделі

Виконаємо редагування програми побудови нашої моделі, щоб в подальшому мати можливість швидко редагувати масив елементів безпосередньо виконуючи цю програму.

Для взаємодії з користувачем в Steo використовуються системні підказки.

Змінні INPUT можуть бути вказані на початку програми. Типовим використанням змінної INPUT є привласнення значення для якогось параметру моделі. Цей параметр згодом використовується у відношеннях або як вхідне значення для імен моделей, що використовуються у складальних одиницях.

Оператор INPUT повинен вказувати ім'я та тип змінної. Імена змінних завжди повинні починатися з символу.

Підтримуються такі типи змінних:

- Number – число;
- String – рядок символів (це дозволяє вводити параметри або імена моделей, але не атрибути користувача);
- YES\_NO – так або ні.

Якщо для змінної не вказано тип, значенням за замовчанням системи є Number.

Щоразу, коли потрібне введення значення, система пропонує ввести значення вхідної змінної.

Замість того, щоб використовувати системні підказки, можна налаштувати підказки для певних вхідних змінних. Потім під час виконання проекту з'являються підказки, коли необхідно ввести ту чи іншу змінну.

Правила використання підказок:

- підказка повинна бути в лапках;
- підказка повинна слідувати одразу після відповідної вхідної змінної.

Наприклад:

```
INPUT //початок блоку оголошень змінних
THICKNESS NUMBER //оголошення змінної thickness типу
число
"Enter wall thickness for the cylinder" // текст повідомлення
INCLUDE_VALVE YES_NO //булеве значення
"Is valve to be included for analysis" // тексту повідомлення
STOCK_ID STRING //змінна типу рядок
"Enter the part's stock ID" // вивід тексту повідомлення
...
END INPUT //кінець блоку
```

Список введення в PROGRAM може включати оператори IF - ELSE - ENDIF. Коли умова IF має значення FALSE, вам не буде запропоновано ввести значення.

Наприклад:

```

INPUT
INCLUDE_HOLE YES_NO
"Should the hole be included?:"
IF INCLUDE_HOLE == YES
HOLE_DIA NUMBER
"Enter diameter for hole"
ELSE
...
ENDIF
...
END INPUT

```

Для редагування моделі треба обрати інструмент Model->Model Intent-> Program. У вікні меню, що відкрилось натиснути на пункт Edit Design (рис. 4.49).

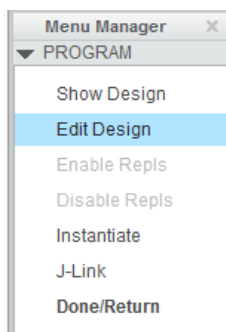


Рисунок 4.49 – Редагування моделі

Таким чином в текстовому редакторі відкриється програма розробленої моделі.

Додати на початку програми блок з запитом до користувача чи необхідно будувати масив, та на якій відстані елементи будуть розташовані один від одного.

Для того, щоб дізнатись яка змінна в програмі відповідає за відстань між елементами, треба знайти блок коду, який буде масив. Він виглядає як на рисунку 4.50.

```

ADD FEATURE (initial number 7)
INTERNAL FEATURE ID 108

NO. ELEMENT NAME          INFO                                STATUS
-----
1  Type                   Dimension                            Defined
2  Regen Method           General                                Defined
3  1st Dir Dime           3,                                    Defined
4  1st Dir Inst           3,                                    Defined
5  2nd Dir Dime           3,                                    Optional
6  Dimension-Va           3,                                    Defined
7  2nd Dir Inst           3,                                    Optional

LEADER OF A (3 X 1) DIM GENERAL PATTERN

MAIN PATTERN DIMENSIONS:
d4 = (Displayed:) 5.00
      ( Stored:) 5.0 ( 0.01, -0.01 )
d5 = (Displayed:) 60.00
      ( Stored:) 60.0 ( 0.5, -0.5 )
d6 = (Displayed:) 5.00
      ( Stored:) 5.0 ( 0.01, -0.01 )
d7 = (Displayed:) 7.00
      ( Stored:) 7.0 ( 0.01, -0.01 )
END ADD

ADD FEATURE (initial number 8)
INTERNAL FEATURE ID 68
PARENTS = 40(#5)

```

Рисунок 4.50 – Блок програмного коду, що буде масив

Змінна d7 відповідає за відстань між елементами та дорівнює 7мм, як було вказано при побудові масиву. Всі інші змінні це розміри побудованого трикутника (сторона трикутника, кут). Змінна d4 – це відстань від краю моделі до першого трикутника.

На початку програми потрібно додати дві змінні, перша строкова для запису чи необхідно будувати масив, в якості другої змінної буде використовуватись вже існуюча d7, а саме відстань між елементами масиву, яку треба змінювати.

Код, який треба додати на початку програми в блок INPUT, буде виглядати наступним чином:

```

INPUT
add_pattern string
"Do you want add the pattern? (y/n)"

if add_pattern == "y"
d7 number
"Assign dimension:"
end if
END INPUT

```

Крім того в програмі моделі на початку побудови масиву треба відкрити цикл IF (рис. 4.51) і у кінці побудови масиву закрити цикл IF (рис. 4.52).

```

if add_pattern == 'y'
ADD FEATURE (initial number 7)
INTERNAL FEATURE ID 108

NO. ELEMENT NAME INFO STATUS
-----
1 Type Dimension Defined
2 Regen Method General Defined
3 1st Dir Dime Defined
4 1st Dir Inst 3, Defined
5 2nd Dir Dime Optional
6 Dimension-Va Defined
7 2nd Dir Inst optional

LEADER OF A (3 X 1) DIM GENERAL PATTERN

MAIN PATTERN DIMENSIONS:
d4 = (Displayed:) 5.00
( Stored:) 5.0 ( 0.01, -0.01 )
d5 = (Displayed:) 60.00
( Stored:) 60.0 ( 0.5, -0.5 )
d6 = (Displayed:) 5.00
( Stored:) 5.0 ( 0.01, -0.01 )
d7 = (Displayed:) 7.00
( Stored:) 7.0 ( 0.01, -0.01 )
END ADD

```

Рисунок 4.51 – Відкриття циклу IF

```

4.1.2 View Direction Side 1
4.1.3 Orientation Bottom
4.1.4 Reference Surf:F5(EXTRUDE_1)
4.2 Sketch Defined
5 Feature Form Solid
6 Direction Side 1
7 Depth Defined
7.1 Side One Defined
7.1.1 Side One Depth None
7.2 Side Two Defined
7.2.1 Side Two Depth To Reference
7.2.2 Reference Surf:F5(EXTRUDE_1)

SECTION NAME = Section 1
MEMBER (3, 1) IN A (3 X 1) DIM GENERAL PATTERN

MAIN PATTERN DIMENSIONS:
d4 = (Displayed:) 5.00
( Stored:) 5.0 ( 0.01, -0.01 )
d5 = (Displayed:) 60.00
( Stored:) 60.0 ( 0.5, -0.5 )
d6 = (Displayed:) 5.00
( Stored:) 5.0 ( 0.01, -0.01 )
d7 = (Displayed:) 7.00
( Stored:) 7.0 ( 0.01, -0.01 )
END ADD
end if

MASSPROP
END MASSPROP

```

Рисунок 4.52 – Закриття циклу IF

Ділі потрібно зберегти зміни у текстовому файлі та закрити його. Після чого в Creo з'явиться вікно (рис. 4.53), у якому натиснути ОК.

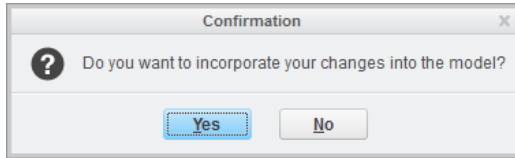


Рисунок 4.53 – Підтвердити застосування зроблених змін

Для виконання програми, в меню Menu Manager (якщо меню закрито, то щоб відкрити його знову оберіть Model->Operations->Regenerate) необхідно натиснути на пункт Enter, після чого обрати параметри з якими треба працювати (add\_pattern та d7), далі натиснути Done Sel (див. рис. 4.54).

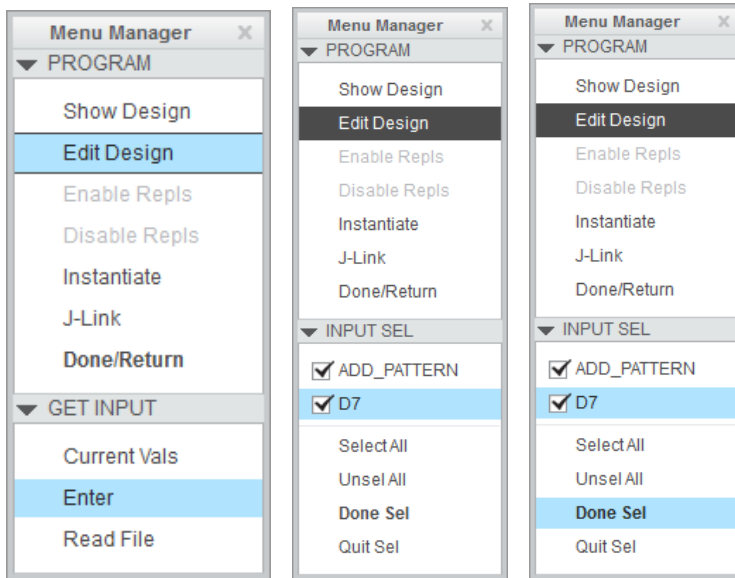


Рисунок 4.54 – Послідовність вибору параметрів

Першим з'явиться запит до користувача – чи потрібно будувати масив. В поле вводу ввести англійську літеру “у” (див.рис. 4.55).

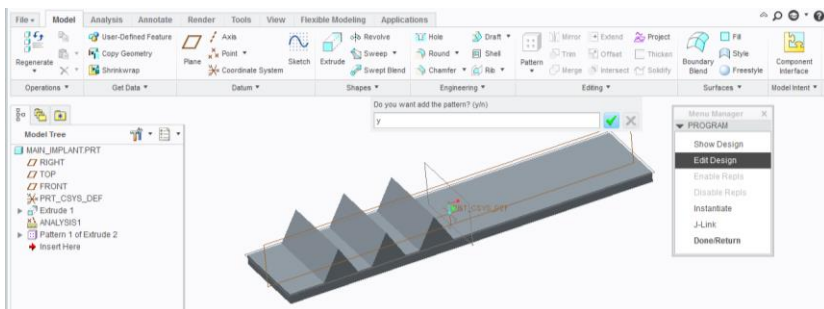


Рисунок 4.55 – Запит на побудову масиву

Наступний запит на встановлення відстані між елементами, в поле вводу ввести 10 (рис. 4.56). У результаті відстань між елементами зміниться з 7 мм на 10 мм.

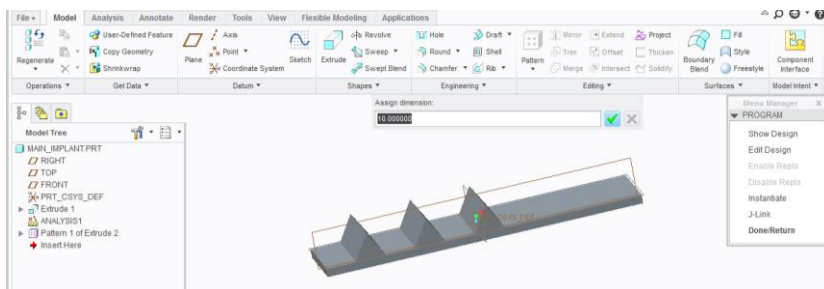


Рисунок 4.56 – Запит на встановлення відстані між елементами

Щоб закрити вікно меню оберіть пункт Done/Return.

#### 4.6 Висновки до розділу 4

В даному розділі було представлено методику розробки 3D моделі спінального імпланта в Creo. Показано як за заданими в Excel файлі параметрами за допомогою інструменту Excel Analysis можливо побудувати параметричну модель імпланта. Наведено базові прийоми роботи з програмою побудованої моделі, а саме внесення змін до програми та її подальше виконання.

#### 4.7 Контрольні питання

1. Які біомедичні конструкції використовуються для функціональної стабілізації хребта людини?
2. Назвіть завдання, що вирішуються розробниками біомедичних конструкцій.
3. Які типи середовищ розробки біомедичних пристроїв та конструкцій ви знаєте?
4. Визначте конструктивні, технологічні та функціональні особливості імпланта Coflex.
5. Які основні етапи діяльності розробника індивідуального спінального імпланта?
6. Охарактеризуйте архітектуру інтегрованого середовища розробки індивідуального спінального імпланта.
7. Визначте особливості використання інструментів Excel Analysis та Program при розробці індивідуального спінального імпланта.
8. Охарактеризуйте основні етапи побудови базової 3D моделі спінального імпланта в середовищі Creo Parametric.
9. Назвіть особливості використання інструментів Program та Excel Analysis для роботи з параметричною моделлю в Creo.
10. Як здійснюється побудова масиву елементів на основі розробки програми для швидкого редагування параметрів?
11. Які ви бачите шляхи подальшого покращення форми та масогабаритних параметрів імплантів хребта людини?

#### 4.8 Практичні завдання

*Завдання 1.* За заданим креслеником імпланта Coflex® виконайте розробку його 3D моделі в середовищі за методикою, викладеною в п. 4.4.

*Завдання 2.* Виконайте розробку параметричної моделі на основі методики, викладеної в п. 4.5. Зберігайте основні параметри моделі в файлі Excel, а масиви зробіть з використанням інструменту Program.

## ВИСНОВКИ

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.

Розроблено віртуальний та фізичний прототипи кіберпротезу, а також ПЗ для його функціонування на основі базових захоплень. В ході розробки механічної частини прототипу було використано середовище проєктування PTC Creo, 3D сканер Gotcha 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. Зокрема, було обрано основну плату Arduino Nano, плату розширення Arduino Nano Shield V3.0, серводвигун SG90. Описано інтеграцію механічної та електронної частин прототипу, а також розробку ПЗ для функціонування прототипу.

Показано переваги використання параметричного моделювання при проєктуванні віртуального прототипу спінального імпланту на основі інтегрального середовища розробки, що об'єднує інструменти Creo Parametric, Excel Analysis та Program.

Застосування описаних технологій прототипування дозволить підвищити ефективність процесу розробки біомедичних пристроїв та конструкцій за рахунок зменшення часу проєктування та кількості помилок, а також надасть можливості покращення форми та масогабаритних показників проєктованих виробів.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. 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. 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 Moorlehem // *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\\_\(Cyber-Physical\\_System,\\_CPS\)](http://www.tadviser.ru/index.php/Статья:Киберфизические_системы_(Cyber-Physical_System,_CPS)).

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. Каплан, А. Как работает киберпротез? [Электрон. ресурс] / А. Каплан. – Режим доступа: <https://postnauka.ru/faq/63738>

13. Пятковский, Ю. Биомехатроника: киберпротезы, дающие человеку сверхспособности. [Электрон. ресурс] / Ю. Пятковский. – Режим доступа: <https://itc.ua/articles/biomehatronika-kiberprotezyi-dayushhie-cheloveku-sverhsposobnosti/>.

14. Попадюха, Ю. Особливості біонічних протезів верхніх кінцівок [Електрон. ресурс] / Ю. Попадюха. – Режим доступу: <http://esnuir.eenu.edu.ua/bitstream/123456789/13204/1/Yuriy%20Pora%20diukha.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://arduinomaster.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://pts-russia.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-trehmernoemodelirovanie.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-pechati>.

46. 3D технологии: сканеры и принтеры [Электрон. ресурс]. – Режим доступа: <https://media-tellur.ru/oborudovanie/3d-technologii-skanery-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://3ddevice.com.ua/product/leapfrog-creatr-hs-3d-printer/>.

49. 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/>

50. Simplify3D 3D printing software for German RepRap 3D printers [Electronic resource]. – Access mode: <https://www.makepartsfast.com/simplify3d-3d-printing-software-german-reprap-3d-printers/>

51. Simplify3D. Часть 3. Настройки процесса печати [Электрон. ресурс]. – Режим доступа: <https://3deshnik.ru/blogs/volandzel/simplify3d-chast-3-nastrojki-processa-pechati>

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. Modern methods and implants in spine neurosurgery [Electronic resource]. – Access mode: <http://carmel74.ru/service/sovremennye-metody-lecheniya-pozvonochnika/>.

72. Babu, M. A. Device innovation in neurosurgery / M. A. Babu, R. F. Heary, B. V. Nahed // Neurosurgery. – 2012. - vol. 70(4). - P. 789–795.

73. The current testing protocols for biomechanical evaluation of lumbar spinal implants in laboratory setting: a review of the literature / [S. A. Gonzalez-Blohm, J. J. Doulgeris, W. E. Lee, et.al.] // Biomed. Res. Int. – 2015. - P.1-15.

74. Demir, T. Mechanical testing standards of orthopedic implants / T. Demir, and M. Özkaya // Musculoskeletal research and basic science. Springer. - 2016. - P. 61–91.

75. Dr. Spine. Spine Surgery. What is a spinal implant and who can benefit? [Electronic resource]. – Access mode: <http://surl.li/iwxc>.

76. Computer-aided design and manufacture of implants using CAD/CAM systems from Delcam [Electronic resource]. – Access mode: <https://sapr.ru/article/24393>.

77. Information technology of robotic prosthesis computer-aided design based on parametric modeling // [A. V. Parkhomenko, O. M. Gladkova, Y. I. Zalyubovskiy, et.al.] // 2021 IOP Conf. Ser.: Mater. Sci. Eng., 1016, 012016.

78. Computer-guided applications for dental implants, bone grafting, and reconstructive / [M. Rinaldi, A. Esposti, A. Mottola, et.al.] // Surger. – 2015. - 568 p.

79. What's new in spine and orthopedic computer-assisted surgery (CAS) [Electronic resource]. – Access mode: <https://blog.smarttrak.com/whats-new-in-spine-and-orthopedic-computer-assisted-surgery-cas>.

80. Malham, G. M. What should my hospital buy next? – Guidelines for the acquisition and application of imaging, navigation, and robotics for spine surgery / G. M. Malham, T. Wells-Quinn // J. Spine Surg. – 2019. - vol. 5 (1). - P. 155–165.

81. Минимальное вмешательство со значительным результатом – вот новая концепция лечения по стабилизации позвоночного столба с помощью импланта Coflex® (Кофлекс®) [Электрон. ресурс]. – Режим доступа: <http://mst.ru/information/patients/spine/degens/coflex/>.

82. What is coflex? How it works [Electronic resource]. – Access mode: <https://coflexsolution.com/how-it-works/>.

83. Coflex-F® Малоинвазивный имплант для спондилодеза на одном уровне [Электрон. ресурс]. – Режим доступа: <https://germesrf.com/catalog/rashodnye-materialy/Nejroxirurgiya/35>.

## АЛФАВІТНО-ПРЕДМЕТНИЙ ПОКАЖЧИК

### Б

Біомедична конструкція, 153

### В

Вбудовані системи, 7

### Д

Діаграма діяльності, 159

### І

Інтегроване середовище, 159

Інтернет речей, 8

Інформаційна технологія, 87

### К

Кіберпротез, 10

Кіберфізичні системи, 7

Критерії розробки, 17

### М

Макетування, 55

Метод

адитивного виробництва, 57

комплексного прототипування, 26

### П

Параметрична модель, 87, 159

Параметричний опис, 161

Плата

Arduino, 120

розширення, 129

Принтер

програмне забезпечення, 62

технічні характеристики, 61, 67

Програма

слайсер, 59

сканування, 48

емулятор, 27

схемотехнічного моделювання, 28

Прототип

механічна частина, 26

електронна частина, 27, 120

фізичний, 53, 55, 119

віртуальний, 90

Прототипування

процес, 23

типи, 25

### С

Серводвигун

групи, 134

підключення, 137, 143

програмування, 137, 144

характеристики, 135

Сканер

безконтактний, 40, 44

контактний, 40

лазерний, 40

оптичний, 40

переваги, 42

принцип роботи, 39

технічні характеристики, 43, 45

Система

автоматизованого

проектування

високого рівня, 36

легкого рівня, 32

середнього рівня, 33

спеціалізована, 34

універсальна, 32

хмарна, 38

Спінальний імплант, 156

### Т

Технологія

пошарового нанесення,

тривимірного друкування, 56, 60

тривимірного сканування, 40

швидкого прототипування, 55

Тривимірна(ний)

модель, 63

принтер 61, 66

сканер 39, 43

### Ф

Файл

STL, 47, 59

Excel, 160

### Ц

Цифровий двійник, 38

### Я

Якості прототипу, 23

Навчальне видання

Пархоменко Анжеліка Володимирівна  
Гладкова Ольга Миколаївна  
Пархоменко Андрій Валентинович

**ПРОТОТИПУВАННЯ БІОМЕДИЧНИХ ПРИСТРОЇВ ТА  
КОНСТРУКЦІЙ**

Навчальний посібник

Редактор Пархоменко А. В.  
Дизайн обкладинки Андреев М. О.

Формат 60x84/16. Ум.-друк. арк. 11,74.  
Папір офсетний, 80 г/м. кв. Друк цифровий.  
Тираж 100 прим. Видавець і виготівник ПП «Євро-Волинь»  
м. Житомир вул. Крошенська, буд. 45, кв. 34  
Свідоцтво серія ДК № 7208 від 07.12.2020