

Харківський національний університет радіоелектроніки

Факультет Електронної та біомедичної інженерії

Кафедра Мікроелектроніки, електронних приладів та пристроїв

Рівень вищої освіти перший (бакалаврський)

Спеціальність 171 Електроніка
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма «Електронні пристрої та системи»
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Кузнецову Ігорю Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Інтелектуальна система керування протезом руки з функцією голосового контролю»

затверджена наказом університету від «26» травня _____ 2025 р. № 415 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 12 червня 2025 р.

3. Вихідні дані до роботи:

3.1 Напрямок живлення 5В;

3.2 Діапазон робочої температури варується від - 40°C до + 85°C,

3.3 Максимальний та мінімальний струм споживання схеми дорівнюють 4,46 А та 1,12 А відповідно;

3.4 максимальна потужність дорівнює 21 Вт;

3.5 мінімальна ємність акумулятора становить в 5000 мА·год для безперервної роботи в активному режимі 1,12 години та 4,46 годин безперервної роботи в сплячому режимі.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Актуальність застосування голосового інтерфейсу для медичних пристроїв _____

2. Вибір апаратних та програмних засобів _____

2.2 Функціональна схема девайса _____

2.3 Розробка електричної схеми, вибір елементної бази _____

3 Розробка друкованої плати _____

4 Реалізація програмного забезпечення інтелектуальної системи голосового контролю _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____

Додаток А Відомість атестаційної роботи бакалавра _____

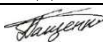
Додаток Б Кресленик електронної принципової схеми приладу _____

Додаток В Кресленик друкованих плат _____

Додаток Г Лістинг інтелектуальної системи керування _____

Додаток Д Презентація «інтелектуальна система керування протезом руки з функцією голосового контролю» _____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доцент. Пащенко Олександр Георгійович		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз отриманого завдання на кваліфікаційну	1.04.2025р.	виконано
2.	Аналітичний огляд медичної та технічної літератури	15.04.2025р.	виконано
3.	Структурування матеріалу	27.04.2025р.	виконано
4.	Вибір апаратних та програмних засобів	02.05.2025р.	виконано
5.	Опис обраного функціоналу, функціональна схема	05.05.2025р.	виконано
6.	Вибір елементної бази. Розробка електричної схеми	09.05.2025р.	виконано
7.	Розробка друкованої плати	14.05.2025р.	виконано
8.	Реалізація програмного забезпечення	18.05.2025р.	виконано
9.	Оформлення пояснювальної записки	22.05.2025р.	виконано
10.	Оформлення презентації доповіді	24.05.2025р.	виконано
11.	Рецензування, нормоконтроль	1.06.2025р.	виконано
12.	Дистанційний захист кваліфікаційної роботи	11.06.2025р.	виконано

Дата видачі завдання 1 квітня 2025 р.

Студент 
(підпис)

Керівник роботи  доц. Пащенко О.Г.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить: 68 с., 51 рис., 5 додатки, 23 джерел, 0 табл.

ПРОТЕЗ, СИСТЕМА, СХЕМА, ГОЛОС, РОЗРОБКА, КОНСТРУКЦІЯ, МІКРОПРОЦЕССОР, РУХ, ЕЛЕМЕНТ, СЕРВОПРИВІД.

Об'єкт розроблення – розробка приладу з голосовим інтерфейсом для медичного застосування, а саме протез кінцівки з голосовим управлінням, що використовує мікроконтролер ESP-32 для забезпечення керування та комунікації.

Мета роботи – розробка та обґрунтування методу голосового інтерфейсу створення електронної схеми на базі мікроконтролера з використанням 3D-моделювання.

Методи дослідження й перелік апаратури – аналіз існуючих рішень у сфері біомедичної інженерії. Проектування та моделювання протеза за допомогою програмного забезпечення. Проектування електронної принципової схеми, та друкованої плати (Kicad). Програмування голосового інтерфейсу з використанням середовища Visual Studio.

У ході виконання роботи було розроблено прилад з голосовим інтерфейсом для медичного застосування, використання мікроконтролеру дає оптимальний баланс між потужністю та енергоефективністю. Базова конструкція протеза виготовлена з легких термопластичних матеріалів.

Описаний метод створення стандартизованих медичних приладів є перспективним. У майбутньому його впровадження дозволить підтвердити ефективність цього підходу у побуті, а також в інших напрямках ортезування та протезування. Це відкриває можливості для його широкого застосування та масштабного рівня виробництва.

ABSTRACT

The explanatory note contains: 68 p., 51 fig., 5 appendices, 23 sources, 0 tables.

PROSTHESIS, SYSTEM, SCHEME, VOICE, DEVELOPMENT, DESIGN, MICROPROCESSOR, MOVEMENT, ELEMENT, SERVODRIVE.

The object of development is the development of a device with a voice interface for medical use, namely a voice-controlled limb prosthesis that uses an ESP-32 microcontroller to provide control and communication.

The purpose of the work is to develop and substantiate a voice interface method for creating an electronic circuit based on a microcontroller using 3D modeling.

Research methods and equipment list – analysis of existing solutions in the field of biomedical engineering. Design and modeling of the prosthesis using software. Design of electronic schematic diagram and printed circuit board (Kicad). Programming of voice interface using Visual Studio environment.

During the work, a device with a voice interface for medical use was developed, the use of a microcontroller provides an optimal balance between power and energy efficiency. The basic design of the prosthesis is made of lightweight thermoplastic materials.

The described method for creating standardized medical devices is promising. In the future, its implementation will allow confirming the effectiveness of this approach in everyday life, as well as in other areas of orthosis and prosthetics. This opens up opportunities for its widespread use and large-scale production.

ЗМІСТ

ВСТУП	7
1 АКТУАЛЬНІСТЬ ЗАСТОСУВАННЯ ГОЛОСОВОГО ІНТЕРФЕЙСУ ДЛЯ МЕДИЧНИХ ПРИСТРОЇВ.....	9
1.1 Медико-технічні особливості протезування	9
1.2 Сучасні технології протезування верхніх кінцівок	12
1.3 Можливості застосування 3D-друку для створення компонентів протезів.....	15
2 ВИБІР АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ.....	21
2.1 Опис обраного функціоналу протезу	21
2.2 Функціональна схема девайса	22
2.3 Розробка електричної схеми, вибір елементної бази	24
3 РОЗРОБКА ДРУКОВАНОЇ ПЛАТИ.....	41
4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ГОЛОСОВОГО КОНТРОЛЮ	54
ВИСНОВКИ.....	65
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	66
ДОДАТОК А.....	Ошибка! Закладка не определена.
ДОДАТОК Б КРЕСЛЕНИК ЕЛЕКТРОННОЇ ПРИНЦИПОВОЇ СХЕМИ ПРИЛАДУ	Ошибка! Закладка не определена.
ДОДАТОК В КРЕСЛЕННЯ ДРУКОВАНИХ ПЛАТ.....	Ошибка! Закладка не определена.
ДОДАТОК Г ЛІСТІНГ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ КЕРУВАННЯ	Ошибка! Закладка не определена.
ДОДАТОК Д ПРЕЗЕНТАЦІЯ «ІНТЕЛЕКТУАЛЬНА СИСТЕМА КЕРУВАННЯ ПРОТЕЗОМ РУКИ З ФУНКЦІЄЮ ГОЛОСОВОГО КОНТРОЛЮ»	Ошибка! Закладка не определена.

ВСТУП

Розвиток медичних технологій відкриває нові можливості для покращення якості життя людей з інвалідністю, зокрема через створення сучасних, високотехнологічних протезів. Завдяки інноваціям у біомеханіці, матеріалознавстві та штучному інтелекті сучасні протези стають дедалі функціональнішими, ергономічнішими та зручнішими у використанні. Вони дозволяють не лише відновлювати основні рухові функції, а й значно підвищувати рівень незалежності користувачів у повсякденному житті.

Проте, попри значний прогрес у цій сфері, одним із ключових викликів залишається ефективне та зручне керування такими пристроями. Традиційні методи управління протезами, зокрема механічні або електроміографічні системи, мають низку обмежень. Наприклад, механічні системи можуть бути громіздкими та вимагати значних фізичних зусиль для контролю, тоді як електроміографічні – складними в налаштуванні, дорогими та залежними від рівня розвитку м'язової активності користувача.

У цьому контексті одним із перспективних напрямів розвитку є використання голосового управління. Така технологія може стати не лише зручним, а й інтуїтивно зрозумілим рішенням для багатьох користувачів, адже вона дозволяє значно спростити взаємодію з протезом без необхідності застосування фізичних зусиль. Використання голосових команд дає змогу людині більш природно та ефективно контролювати свої рухи, що особливо важливо для людей із обмеженою м'язовою активністю або додатковими фізичними обмеженнями.

Таким чином, подальше вдосконалення методів керування протезами, зокрема впровадження голосового управління та інших інноваційних підходів, може зробити ці пристрої ще більш доступними та функціональними для широкого кола користувачів. Це не лише покращить рівень комфорту та

мобільності людей з інвалідністю, а й сприятиме їхній соціальній адаптації та інтеграції в суспільство.

У даній роботі розглядається розробка керованого голосом протеза верхньої кінцівки на основі мікроконтролера ESP32. Його було обрано через високі обчислювальні можливості, підтримку бездротових технологій, сумісність із сучасними системами розпізнавання голосу та іншими електронними модулями.

Робота спрямована на створення доступного, ефективного та простого у використанні пристрою, що може бути інтегрований у повсякденне життя користувачів. Мета роботи полягає у розробці та реалізації концепції та розгляд і подальшим створенням електронних вузлів в Spice подібних програмах, які будуть реагувати на голосові команди та виконувати заданні програм-ним шляхом рухи. Для досягнення цієї мети дослідимо методи обробки голосових сигналів, алгоритми управління сервоприводами, розробку друкованих плат.

Актуальність теми обумовлена зростаючою потребою у доступних та інноваційних рішеннях для людей з втратою кінцівок. Використання сучасних мікроконтролерів та алгоритмів голосового управління дозволяє створити протез, який не лише відтворює основні рухові функції, але й значно покращує якість життя користувачів.

1 АКТУАЛЬНІСТЬ ЗАСТОСУВАННЯ ГОЛОСОВОГО ІНТЕРФЕЙСУ ДЛЯ МЕДИЧНИХ ПРИСТРОЇВ

1.1 Медико-технічні особливості протезування

Протези – це спеціальні механічні або біонічні пристрої, призначені для заміни втрачених сегментів кінцівок чи інших частин тіла. Вони виконують важливу функцію компенсації рухових можливостей, допомагаючи людям з інвалідністю відновлювати активність та незалежність у повсякденному житті.

Сучасні протези можуть мати як суто механічну конструкцію, так і високотехнологічні рішення, зокрема електронні та біонічні системи, які дозволяють користувачам здійснювати складні рухи завдяки сенсорам, штучному інтелекту та нейроінтерфейсам. Основне призначення протезів полягає не лише у відновленні функцій ушкодженого органа, а й у покращенні якості життя людини, її соціальній адаптації та відчутті комфорту.

Окрім функціонального аспекту, протези відіграють важливу роль у косметичній корекції, допомагаючи приховати фізичні недоліки та забезпечити природний вигляд. Завдяки сучасним технологіям зовнішній вигляд протезів може максимально наближатися до натуральних частин тіла, що позитивно впливає на психологічний стан користувачів.

Косметичні протези, або так звані пасивні протези, не виконують функції захвату або ж забезпечують лише пасивне утримання предметів (вмикаються і вимикаються за допомогою збереженої руки). Попри це, вони мають велике значення для осіб з ампутацією [3].

Втрата частини чи всієї кінцівки є фізичним дефектом, який косметичний протез допомагає приховати від оточення. Це відіграє важливу роль у психологічній стабілізації, особливо людей, чия професійна діяльність

пов'язана з публічними контактами (наприклад, працівники інформаційних служб, тощо). Такий аспект не можна ігнорувати в процесі реабілітації.

Окрім психосоціальних функцій, протез верхньої кінцівки допомагає збалансувати масу тіла. Якщо людина після ампутації не користується протезом, з часом її плече з боку втраченої кінцівки піднімається вище, що може спричинити викривлення хребта. Під час ходьби маса протеза відіграє важливу роль у гармонізації рухів плечей і таза. У людей без протеза обмежена амплітуда рухів, що призводить до характерної «скованої» ходи. Однак, якщо людина не зацікавлена у косметичному протезі й не вбачає в ньому необхідності, його забезпечення таким пристроєм не має сенсу, оскільки він має лише незначну функціональну користь, наприклад, допомагаючи пасивно утримувати предмети [3].

Для дітей косметичне протезування має інше значення. Навіть мінімальна допоміжна функція протеза сприяє його сприйняттю та підготовці до використання функціонального протеза в майбутньому. Раннє протезування значно полегшує навчання користуванню пристроєм, а згодом людина самостійно вирішує, який тип протеза їй підходить [3].

Серед функціональних протезів розрізняють не тільки й пасивні є й ще активні протези. Наступним рівнем технологічного розвитку функціональних протезів верхніх кінцівок є протези з тяговим управлінням.

На відміну від пасивних-косметичних протезів, які потребують спеціальних насадок для виконання різних завдань, у тягових протезах механізм захвату приводиться в рух гнучкими тягами, що передають зусилля до механізмів протеза. Для управління протезом передпліччя необхідно передавати зусилля для руху кисті, що здійснюється за допомогою:

- м'язів плеча, які забезпечують розкриття протезної кисті через рухи згинання і розгинання;
- добре розвиненого м'язового каналу двоголового м'яза плеча;
- рухів плечового пояса з використанням ремінця, до якого кріпиться тяга;

– вплив висоти ампутації на вибір протеза.

Рівень ампутації відіграє ключову роль у виборі типу функціонального протеза. Для кукси плеча управління повинно включати щонайменше три функції такі як активний рух кисті протеза, згинання передпліччя, фіксацію та розблокування ліктьового суглоба.

Для діяльності захвату джерелами сили є м'язи плечового пояса і збережена мускулатура кукси плеча в плечовому суглобі. Три названі функції можуть виконуватися абсолютно незалежно один від одного за допомогою 35 трьох тяг передачі зусиль при використанні трьох різних і незалежних видів руху. Є також можливість активно виконувати всі три функції при використанні тільки двох тяг передачі зусиль, коли одна з тяг виконує послідовно дві функції. Тому розрізняють кріплення протеза плеча з трьома і з двома тягами [3].

Міоелектричні протези складаються з трьох основних конструктивних вузлів, де індивідуальна ортопедична частина, виготовлена за зліпком або обчерком, наприклад, гільза плеча чи передпліччя. Механічні напівфабрикати, до яких належать механічний плечовий шарнір, механічний або електромеханічний ліктьовий шарнір, механічний або електромеханічний шарнір пронації-супінації, а також електромеханічна кисть. Електричні та електронні напівфабрикати, такі як акумулятор (батарея), керуючі електроди, підсилювальні каскади, вимикачі та мотори для приводу шарнірів.

Принцип роботи міоелектричних протезів, поверхневі електроди, розташовані на шкірі або введені під шкіру чи всередину м'язів у визначених місцях, зчитують біоелектричні потенціали, що виникають при скороченні м'язів. Ці сигнали передаються на електронну систему управління протезом. Біоелектричні потенціали мають мікрвольтний діапазон і можуть бути зафіксовані високочутливими вимірювальними приладами [3].

Інтенсивність активності м'язів, наприклад, згиначів або розгиначів плеча чи передпліччя, використовується як команда для перемикання функцій

протеза: згинання та розгинання ліктьового шарніра, відкривання та закривання кисті або її пронації-супінації. Таким чином, управління протезом здійснюється не за допомогою кнопок або перемикачів, а через напруження м'язів. Оскільки м'язова активність контролюється нервовою системою, можна сказати, що протез керується "волею" користувача, що робить його більш природним у використанні [3].

Отриманий від електродів сигнал надходить у мініатюрний підсилювач, а далі через мікроконтролер передається на двигун протеза, переводячи його в потрібний робочий режим. Електронна система управління дозволяє реалізовувати різні схеми перемикання: цифрову, пропорційну, порогову, залежну від інтенсивності або тривалості імпульсу тощо.

1.2 Сучасні технології протезування верхніх кінцівок

Сучасні технології протезування верхніх кінцівок значно просунулися завдяки розвитку біоніки, штучного інтелекту, 3D-друку та нейроінтерфейсів. Основним напрямком розвитку цієї галузі є біонічні протези. Безумовно, можна сказати, що біонічні протези це ті ж міоелектричні протези, проте краще.

Зараз відомими виробниками біонічних протезів є: Ossur (Ісландія), Ottobock (Німеччина), Vincent Systems (Німеччина), Taska (Нова Зеландія), та Steeper (Великобританія). Завдяки сучасним матеріалам, збалансованому розміщенню двигунів, сенсорних датчиків та містких акумуляторів, розробники протезів змогли створити біонічний протез, здатний до більшості повсякденних дій [2].

i-Digits Quantum – це біонічна кисть, яка розроблена для людей із частковою ампутацією кисті. Вона оснащена інтелектуальними сенсорами, які автоматично адаптують силу хвату на 30%, підвищують точність рухів на 25% та прискорюють реакцію на команди користувача на 20%. Ергономічний

дизайн і технологія Touch Bionics забезпечують природність та плавність рухів (рис 1.1).



Рисунок 1.1 – Протез i-Digits Quantum

Веbionic 3 є головним продуктом компанії Ottobock має на своєму забезпечені 14 варіантів рухів і хвату, включаючи природний рух пальців. Індивідуальне налаштування сили та швидкості кожного пальця. Високу чутливість сенсорів що дозволяє точно керувати приладом. Легка та міцна конструкція з карбоновими елементами має великий плюс для повсякденного користування (рис 1.2).



Рисунок 1.2 – Протез VeBionic 3 з різними за довжиною насадками

Taska Hand – це єдина у світі водонепроникна біонічна рука зі ступеню захисту IP67, яка дозволяє виконувати складні маніпуляції навіть у вологих умовах. Удосконалені моторизовані приводи підвищують силу хвату на 40%, а адаптивна технологія запобігає пошкодженню крихких предметів при цьому маючи 23 режими хвату для адаптації до різних завдань. Конструкція з ударостійких матеріалів збільшує довговічність пристрою в екстремальних умовах (рис 1.3).



Рисунок 1.3 – Водостійкий протез Taska Hand

Кожен із цих виробників пропонує сучасні рішення які адаптуються до потреб користувачів. Össur та Ottobock зосереджуються на інтелектуальному управлінні, Vincent Systems – на легкості та точності рухів, Taska – на міцності та водостійкості, а Steeper – на інтеграції біонічних технологій для руки та ліктя.

Таким чином, розвиток протезування є важливим напрямом медицини та інженерії, який сприяє створенню все більш досконалих рішень, що поєднують ергономіку, естетику та інноваційні технології.

1.3 Можливості застосування 3D-друку для створення компонентів протезів

Зважаючи на те, що голосом керований протез позиціонується як доступний та простий у виготовленні пристрій, розглянемо можливі альтернативи та потенційні недоліки традиційного підходу до виготовлення оболонок протезів із композитних матеріалів.

Розглянемо спочатку які бувають види 3D друку, які можливо використовувати не витрачаючи космічні суми на розходні матеріали та технічні рішення. Види 3D друку поділяють відповідно до сфер використання, а в свою чергу там вибирають відповідно матеріалу, що ми бачимо нижче:

Спочатку потрібно навести що типи 3D-друку залежать першочергово від типу матеріалу наприклад полімери метали та інше (рис 1.4).

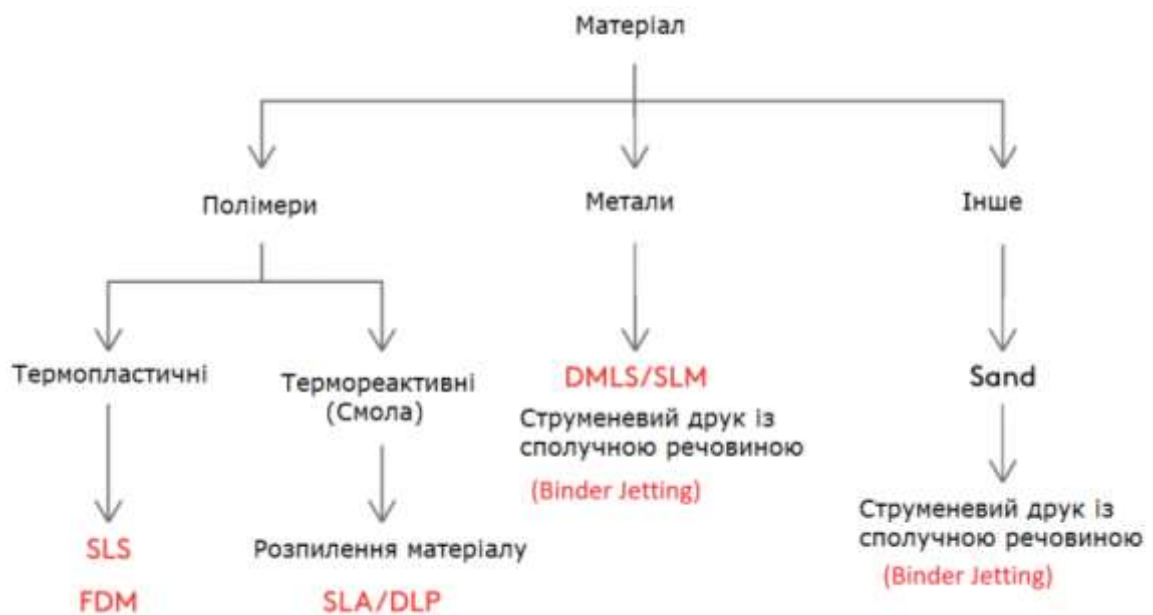


Рисунок 1.4 – Типи 3D-друку

Для ортезування та протезування обирають як полімери так і металеві матеріали, проте використання металічних матеріалів в 3D-друці має свої ключові недоліки.

Вартість процесу 3D друку металічними матеріалами може бути дуже дорогим у порівнянні з іншими методами виробництва. Вартість матеріалів, обладнання та послуги знаходяться на високому рівні, що робить його недоступним для багатьох компаній та осіб.

Обмеженість типів відповідних металів для друку встановлює свої обмеження, хоча при цьому доступність металевих матеріалів постійно збільшуються, вибір все ще обмежений порівняно з іншими матеріалами. Це може обмежувати можливості дизайну та виробництва.

Водночас для досягнення високої якості друку потрібно дотримуватися строгих параметрів та використовувати високоякісні матеріали. Навіть невеликі відхилення можуть призвести до дефектів або втрати міцності виготовленого виробу.

За цих умов обмежена швидкість виробництва металевими матеріалами буде досить повільною у порівнянні з іншими методами виробництва. Це може

бути проблемою для виробництва великих обсягів приладів або в сферах, де потрібно швидко реагувати на змінні умови. При цьому багато друкарських процесів вимагають післяобробки для досягнення бажаного вигляду та міцності виробу. Це може включати обробку поверхні, обробку теплом або інші послідовні операції, що збільшують час та витрати на виробництво.

Через ці недоліки вибір падає на полімери, тому залишається вибір між технологією друку, розглянемо їх.

FDM-друк: що це таке та які його особливості.

FDM (Fused Deposition Modeling) – це одна з найпопулярніших технологій 3D-друку, яка заснована на пошаровому наплавленні термопластичного матеріалу. Суть цього процесу полягає у подачі розплавленої нитки пластику (філаменту) через екструдер – друкувальну головку, оснащену нагрівальним елементом. Температура нагрівального елемента забезпечує плавлення матеріалу, після чого він видавлюється через сопло та накладається шарами на платформу принтера, створюючи тривимірну модель.

До переваг FDM-друку відносять доступність та економічність, цей тип друку є відносно недорогим методом у порівнянні з іншими технологіями 3D-друку, такими як SLA або SLS. При цьому є великий вибір бюджетних принтерів і матеріалів для друку. Завдяки простоті технології та широкому спектру доступних термопластичних матеріалів, FDM дозволяє отримувати стабільні результати навіть при масовому виробництві.

Надруковані моделі можна легко обробляти механічним способом (шліфування, свердління, різання). У більшості випадків подальша обробка деталей не вимагає складних або небезпечних процедур.

Проте в цієї технології є й недоліки такі як менша точність у порівнянні з іншими технологіями. Через використання екструзійного сопла можуть виникати похибки у розмірах деталей, друковані об'єкти можуть мати видимі шари, так звані «сходинки» що впливає на естетику виробу.

Через перепади температури під час друку можливе деформаційне викривлення моделі, при цьому за недостатньої температури може виникати неоекструзія яка збільшує пористість матеріалу моделі що впливає на міцність виробів.

SLA-друк та його відмінності від FDM.

Стереолітографія (SLA) – це технологія адитивного виробництва 3D-друку, яка використовує рідкі фотополімери та затверджує їх під впливом лазерного випромінювання або іншого джерела світла. У процесі фотополімеризації рідкий матеріал твердне пошарово, створюючи готовий виріб з високою точністю.

Сильними сторонами цього методу є лазерна технологія, друк якої дозволяє отримати моделі з мінімальними похибками та високою роздільною здатністю, що ідеально підходить для створення складних та дрібних деталей, де важлива точність.

Відсутність видимих шарів на надрукованих моделях забезпечує естетичну привабливість що підходить для виготовлення прототипів, ювелірних виробів та медичних імплантів.

Проте на відміну від технології FDM-друку SLA має наступні недоліки, де більшість SLA-матеріалів менш міцні, ніж термопластики, які використовуються в FDM-друку. Загалом більшість фотополімерів не підходять для деталей, що піддаються механічному навантаженню.

При цьому висока вартість SLA-принтерів та фотополімерних смол, з процесом друку що займає більше часу, а постобробка вимагає додаткових етапів очищення та затвердіння не додає цьому методу плюсів.

З урахуванням усіх факторів вибір падає на технологію FDM-друку, де потрібність в міцних, функціональних деталях, які будуть витримувати навантаження, і при цьому мати доступний варіант 3D-друку, технологія FDM є оптимальним рішенням. Завдяки простоті експлуатації, широкому вибору

матеріалів та низькій вартості, цей метод є ідеальним для швидкого та серійного виробництва створення робочих механічних деталей.

Наступним кроком буде вибір матеріалу для друку оболонки протезу та його механічних частин. Є п'ять основних матеріалів FMD друку такі як: PLA, ABS, PETG, Nylon, TPU. Зробимо оглядовий аналіз їх всіх та висунемо висновок щодо вибору матеріалу [13].

PLA (Полілактид) – це біорозкладний термопластичний полімер, виготовлений із відновлюваних джерел, таких як кукурудзяний крохмаль або цукрова тростина. Доволі безпечний не виділяє токсичних речовин при нагріванні, має низьку температуру плавлення 180-220°C. Проте має серйозні недоліки такі як низька термостійкість починає деформуватися за температури 50-60°C. Доволі крихкий в порівнянні з іншими типами пластичних матеріалів. Схильний до гідролізу, може поглинати вологу, що впливає на якість друку.

ABS (Акрилонітрилбутадієнстирол) – термопластичний листовий конструкційний матеріал. До його переваг відносять ударостійкий та міцність, добре витримує механічні навантаження, витримує згинання та розтягнення без розриву. Термостійкий не деформується при температурі до 90-100°C. Легко обробляється – можна шліфувати, фарбувати, клеїти, полірувати ацетоном. Його недоліками є висока температура плавлення 220-250°C, потребує нагріву столу 90-110°C при 3D-друку. При цьому виділяє токсичні випари стіролу, тому потрібна вентиляція при друці та обробці.

PETG (Поліетилентерефталат) – це ударостійкий, гнучкий та хімічно стійкий термопластик, який поєднує в собі переваги PLA та ABS. До переваг відносять високу міцність, гнучкість – не такий крихкий, як PLA, і менш ламкий, ніж ABS, стійкість до вологи та хімікатів – не вбирає воду, не реагує на більшість кислот і лугів. Має гарна адгезію між шарами – менше шансів на розшарування при 3D-друці. Безпечний для харчових продуктів – можна використовувати для контейнерів та пляшок. Проте має і свої недоліки складніший у друці, ніж PLA – можливе «ниткування» (stringing), потребує

точної налаштування температури. М'якший, ніж ABS – хоч і міцний, але більш податливий до подряпин. Менш термостійкий, ніж ABS – деформується при 70-80°C (але краще, ніж PLA).

Nylon (нейлон) – високоміцний, зносостійкий та гнучкий термопластик. Перевагами нейлона є витримування великих навантажень. Зносостійкість – ідеально підходить для рухомих деталей (шестерні, підшипники). Термостійкість – може витримувати температуру до 100°C і більше. Має низький коефіцієнт тертя, що підходить для механічних з'єднань. Основними його недоліками є те що він вбирає вологу з повітря, що впливає на якість друку (потрібне зберігання у герметичних контейнерах). Складність у друці – потребує високої температури екструдера 240-270°C і нагрітого столу 80-100°C, має погану адгезію можливе скручування, потрібні спеціальні підкладки або клей [13].

TPU (Поліуретан) – гетероланцюговий полімер, що поєднує в собі властивості гуми та пластику, роблячи його ідеальним для гнучких і ударостійких деталей. Має наступні переваги легко згинається та розтягується без розривів, витримує удари, скручування та механічне навантаження, не тріскається та не рветься при тривалому використанні. Стійкість до хімікатів та води – не боїться масел, жирів, пального та вологи. Його недоліками є складність у друці – через гнучкість може застрягати у екструдері. Потребує низької швидкості друку 20-40 мм/с для якісного результату. Не всі принтери підтримують TPU – важливо мати екструдер із прямою подачею (Direct Drive).

Таким чином, перспективним матеріалом для виготовлення є PETG (поліетилентерефталат), з огляду на його універсальність, а також відносно прості умови експлуатації та 3D-друку.

2 ВИБІР АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ

2.1 Опис обраного функціоналу протезу

Верхня кінцівка протеза оснащена механізмом згинання, який реалізується за допомогою системи тросів. Ці троси намотуються на спеціальний циліндр, що обертається під дією сервопривода. Коли сервопривод активується, він приводить у рух циліндр, який, у свою чергу, закручує троси, створюючи тягове зусилля. Це зусилля викликає згинання фаланг пальців протезної кінцівки у потрібному напрямку. Така конструкція забезпечує плавний та контрольований рух, дозволяючи користувачу виконувати точні й природні жести.

Також верхня кінцівка протеза імітує роботу ліктьового суглоба завдяки розміщеному між накладкою на плече та протезом передпліччя сервоприводу. Цей сервопривод виконує роль штучного м'яза, забезпечуючи обертання у зоні ліктя та дозволяючи згинати або розгинати руку. Він точно керує кутом повороту, що дозволяє досягти природності рухів та адаптувати силу й швидкість згину відповідно до потреб користувача. Така конструкція дає змогу ефективно імітувати функціональність справжнього ліктьового суглоба, забезпечуючи комфорт і свободу рухів у повсякденному житті.

Іншими словами можна сказати що система приводу, яка включає два серводвигуни для імітації згину ліктя та рухів пальців. Один сервопривод розташовується у зоні ліктьового суглоба та відповідає за його обертання, тоді як другий керує тросовим механізмом, що згинає пальці. Тросова система, яка передає зусилля від серводвигуна до фаланг пальців, забезпечуючи точність, контроль і плавність рухів при стисканні та розжиманні.

Для з'єднання ліктьового суглобу використаємо два карданних шарніра, водночас відбувається створення рухомого моменту за допомогою сервоприводу з насадкою на кукусу (рис 2.1).



Рисунок 2.1 – Конструктивна схема протеза

2.2 Функціональна схема девайса

Виходячи з вище описаного функціоналу протезу в розділі 2.1 можна зробити наступний висновок, що схема цього протезу буде складатися з декількох основних функціональних блоків, кожен з яких виконує окрему роль у забезпеченні рухливості та керованості протезної кінцівки.

Електронна система управління, яка включає мікроконтролер або мікропроцесор може використовувати модулі штучного інтелекту для підвищення точності розпізнавання команд, а також може бути адаптована під індивідуальні потреби користувача методами загального оновлення. Сенсори та програмне забезпечення для обробки сигналів від користувача. Ця система

приймає команди, аналізує їх та надсилає сигнали керування до відповідних приводів. В цій схемі мікрофон використовується для прийому звукових сигналів із зовнішнього середовища. Далі звуковий розшифровувач обробляє аудіосигнали, перетворюючи їх з аналогового сигналу в цифровий формат для подальшої інтерпретації [1].

Міст УАПП до УПШ перетворює сигнали між інтерфейсами, забезпечуючи їхню сумісність. UART являє собою сполучену ланку для обміну даними між мікроконтролером та пристроями периферії. Регулятор напруги 3.3 В знижує напругу живлення до безпечного рівня для мікроконтролера та інших компонентів схеми. USB використовується для підключення пристрою до комп'ютера або джерела живлення через свою шину. Акумулятор забезпечує автономне живлення пристрою, в залежності від робочої напруги його можна підключити як через USB так і через UART (за умови що його робоча напруга дорівнює 3.3 В). Індикаторний світлодіод сигналізує про стан пристрою наприклад пройшла команда повороту, проте нічого не відбулося а світлодіод видав зелений колір та певне блимання, що дає зрозуміти що команда пройшла перевірку, проте має механічні порушення [1].

Загальна схема протезу матиме модульну структуру, що дозволяє змінювати або вдосконалювати окремі елементи конструкції, не змінюючи основу систему. Завдяки цьому забезпечується гнучкість, ремонтпридатність і можливість подальшої модернізації (рис 2.2).

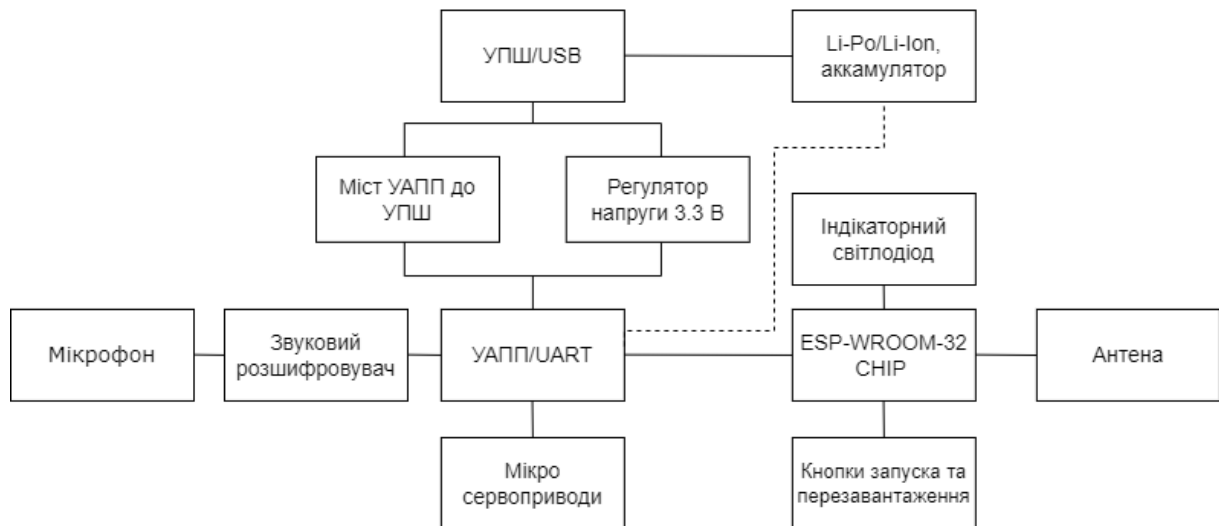


Рисунок 2.2 – Функціональна схема протезу

2.3 Розробка електричної схеми, вибір елементної бази

Для розробки системи керування з голосовим управлінням необхідно створити електричну схему, підібрати відповідну елементну базу та обґрунтувати вибір мікроконтролера. Основні функціональні блоки системи включають мікроконтролер, модуль захоплення голосу (мікрофонний підсилювач або аудіомодуль), драйвери двигунів (серводвигуни або безколекторні), сенсори, блок живлення, елементи інтерфейсу користувача (кнопки, світлодіоди) та захисні компоненти.

При виборі мікроконтролера було проаналізовано кілька варіантів: ESP32, STM32, PSoC та Raspberry Pi Pico W. STM32 має високу продуктивність та підтримує RTOS, але не має вбудованого бездротового зв'язку. PSoC забезпечує гнучку аналогову обробку, проте складний у налаштуванні та дорожчий. Raspberry Pi Pico W має Wi-Fi, проте менш енергоефективний та має обмежену підтримку голосових модулів. У порівнянні з ними ESP32-WROOM-32 вирізняється найбільш збалансованими характеристиками: вбудовані Bluetooth і Wi-Fi, потужний двоядерний процесор, підтримка інтерфейсу I2S

для підключення цифрових мікрофонів, низьке енергоспоживання та велика спільнота розробників.

Для реалізації цих функцій пропонується наступна елементна база. В якості мікроконтролера використовується ESP32-S3-WROOM-1, який виступає центральним керуючим елементом. Для захоплення голосу використовується цифровий мікрофон INMP441 з інтерфейсом I2S. Для керування рухами частин протеза підходить серводвигуни типу MG996R.

В якості альтернативи можна використати: для захоплення голосу – модуль на зразок DFPlayer Mini для відтворення звуків; для керування рухами частин протеза – більш точні модуль, як-от SG90 micro servo. Для зчитування положення чи руху протеза застосовується IMU-сенсор, наприклад, MPU-6050 або точніший BNO055.

Ручне керування реалізується за допомогою двох кнопок NTC013-AT1J-A160T від виробника T-MEC, призначений для поверхневого монтажу. Корпус перемикача має компактні розміри та розрахований на роботу при напрузі до 16 В постійного струму до 50 мА, а максимальний опір контактів не перевищує 500 мОм [21].

В схемі використовується два індикаторних світлодіода, де перший червоний дискретний світлодіод із дифузованою лінзою, призначений для наскрізного монтажу. Він має круглу форму з куполоподібною верхівкою та діаметром 3 мм, відповідає за індикацію лінійного регулятора напруги. Реальним його аналогом є 264-10SDRD/S530-A3, електричні характеристики включають типову пряму напругу 2,0 В (максимум 2,4 В) та зворотну напругу до 5 В. Максимальний прямий струм становить 25 мА. Розсіювання потужності обмежено 60 мВт. Температурний діапазон експлуатації становить від -40°C до +85°C [17].

У ролі візуальної перевірки програмного забезпечення інтелектуальної системи використовується програмований RGB світлодіод SK6812MINI для поверхневого монтажу. Працює в діапазоні напруги від 3,7 В до 5,5 В

споживаючи струм 12 мА на канал. Робоча температура може варуватися від -40 до +80 °С [14].

Для регулятора напруги вибрано лінійний стабілізатор напруги SGM2212-3.3XKC3G/TR. Цей стабілізатор забезпечує максимальний вихідний струм до 800 мА при вхідній напрузі в діапазоні від 2.7 В до 20 В. Типове падіння напруги становить 0.39 В, а максимальне – 0.61 В, що дозволяє ефективно працювати навіть при невеликій різниці між вхідною та вихідною напругою. Діапазон робочих температур становить від -40°С до +125°С [16].

В якості мікросхеми УППШ до УАПШ виступає CP2102N-A01-GQFN28 від Silicon Labs. Ця мікросхема включає в себе контролер функцій USB 2.0 Full-Speed, USB-трансивер, осцилятор та універсальний асинхронний приймач/передавач (UART) у компактному корпусі розміром 5 на 5 мм. Робоча напруга варується від 2,2 В до 3,6 В та має низьке споживання струму в робочому режимі 9,5 мА. Температурний діапазон експлуатації становить від -40°С до +85°С [20].

USB C роз'єм який використовується для живлення всього приладу та завантаження програмного забезпечення. Він поєднує в собі компактний дизайн, симетричну форму та універсальність, що робить його ідеальним вибором. Робоча напруга становить до 5 В за струму від 1,5 до 3 А, потужністю до 15 Вт.

Отже, обраним мікроконтролером є ESP32-S3-WROOM-1, оскільки він ідеально підходить для голосового керування, має достатню обчислювальну потужність, гнучкість у програмуванні (Arduino IDE, PlatformIO, ESP-IDF) та дозволяє забезпечити автономну, компактну та енергоефективну роботу пристрою.

При цьому вище перелічені елементи, на які можна опиратися при зміні елементної бази, мають наступні характеристики:

- мікрофон INMP441 від компанії InvenSense, який працює напряду від цифрового процесора DSP до контролерів з використанням кодеків. Має

наступні фізичні характеристики: діапазон напруги живлення складає 1,62 – 3,63 В, струм споживання 1,4 мА, чутливість -26 dBFS, частотний діапазон 60 Гц – 15000 Гц, температурний діапазон -40°C ... + 85 °C;

- сервопривід MG996 11 кг, кут повороту становить 180 градусів, робоча швидкість має наступні значення 0,17 сек/60 градусів, 4,8В без навантаження та 0,13 сек/60 градусів, 6В без навантаження. Пусковий момент вар'юється 9.4 кг/см при живленні 4,8В до 11 кг/см при живленні 6В. При цьому робоча напруга становить 4,8 – 7,2В.

Розрахуємо максимальне споживання схеми в активному режимі роботи виходячи з даташитів на обрану елементну базу, щоб в подальшому обрати відповідну батарею, акумулятор за ємністю та вирахувати її час безперервної роботи. За наступною формулою розрахуємо загальний струм схеми:

$$I_{all} = I_{esp} + (2 \cdot I_{nrc}) + (2 \cdot I_{1N5819HW}) + (2 \cdot I_{L8050QLT1G}) + I_{SK6812} + I_{CP2102N} + I_{SGM2212} + I_{redled} + (6 \cdot I_{LESD5D5.0CT1G}) + I_{INMP441} + (2 \cdot I_{servo}); \quad (2.1)$$

$$I_{all} = 0,5 + (2 \cdot 50 \cdot 10^{-3}) + (2 \cdot 1 \cdot 10^{-3}) + (2 \cdot 0,8) + 12 \cdot 10^{-3} + 13,7 \cdot 10^{-3} + 800 \cdot 10^{-3} + 25 \cdot 10^{-3} + (6 \cdot 1 \cdot 10^{-3}) + 2,5 \cdot 10^{-3} + (2 \cdot 700 \cdot 10^{-3}) = 4,46 \text{ А}.$$

Виходячи з максимального споживання схеми потрібно обрати батарею, акумулятор з мінімальною ємністю в 5000 мА·год для роботи в 1,12 години безперервної роботи в активному режимі. Проте схема більшу частину часу знаходиться в сплячому режимі, що знижує її споживання від двох до десяти разів [22].

Живлення реалізується на основі літієво-іонного акумулятора 7,4 В з перетворювачами напруги до 5 В та 3,3 В для живлення різних компонентів.

Розрахуємо споживання схеми в сплячому режимі роботи, яке розраховується за формулою 2.1, щоб розрахувати максимальний час безперервної роботи:

$$I_{sleep} = (107,9 \cdot 10^{-3}) + 0 + (2 \cdot 1 \cdot 10^{-3}) + (2 \cdot 1 \cdot 10^{-3}) + 0 + 195 \cdot 10^{-6} + (10 \cdot 10^{-3}) + 0 + (6 \cdot 1 \cdot 10^{-3}) + (0,8 \cdot 10^{-3}) + (2 \cdot 500 \cdot 10^{-3}) = 1,12 \text{ А}.$$

З розрахунку видно, що схема в сплячому режимі за акумулятора ємністю в 5000 мА буде працювати 4,46 годин.

Знаючи максимальне споживання струму з формули 2.1 розрахуємо максимальну потужність (2.2):

$$\begin{aligned}
 P_{\max} = & (V_{esp} \cdot I_{esp}) + 2(V_{ntc} \cdot I_{ntc}) + 2(V_{1N581} \cdot I_{1N5819HW}) + 2(V_{L8050} \cdot I_{L8050QLT1G}) + \\
 & (V_{SK6812} \cdot I_{SK6812}) + (V_{CP2102N} \cdot I_{CP2102N}) + (V_{SGM2212} \cdot I_{SGM2212}) + (V_{led} \cdot I_{led}) + \\
 & 6(V_{LESD5D5} \cdot I_{LESD5D5.0CT1G}) + (V_{INMP441} \cdot I_{INMP441}) + 2(V_{servo} \cdot I_{servo}); \quad (2.2) \\
 P_{\max} = & (3,3 \cdot 0,5) + 2(3,3 \cdot 0,05) + 2(5 \cdot 10^{-3}) + 2(5 \cdot 0,8) + (3,3 \cdot 12 \cdot 10^{-3}) + \\
 & (3,6 \cdot 9,5 \cdot 10^{-3}) + (5 \cdot 800 \cdot 10^{-3}) + (2 \cdot 25 \cdot 10^{-3}) + 6(5 \cdot 0,001) + \\
 & (3,3 \cdot 2,5 \cdot 10^{-3}) + 2(5 \cdot 700 \cdot 10^{-3}) = 21 \text{ Вт}.
 \end{aligned}$$

Створення електричної принципової схеми буде залежати безпосередньо від реалізація програмного забезпечення протезу, тож проведемо моделювання плати на платформі circuit.io в середовищі розробки схем в якості репрезентації, на котру будемо спиратися при розробці схеми в середовищі KiCad (рис 2.3).

Після попереднього аналізу функціональних вимог до пристрою було сформовано базову апаратну конфігурацію, яка дозволяє оглянути ключові вузли протеза: модулі збору вхідних даних, логіку обробки сигналів та систему виконавчих елементів. На макеті реалізовано зв'язок мікроконтролера з серводвигунами, що моделюють рухову активність протеза, а також підключено модуль живлення та датчики.

Окрему увагу було приділено питанню енергозабезпечення: зважаючи на енергоспоживання сервоприводів, використання лише USB-живлення є недостатнім, тому в макеті застосовано Li-Po акумулятор із стабілізацією напруги за допомогою лінійного регулятора. Це дозволяє забезпечити стабільну роботу навіть при пікових навантаженнях. Наступним кроком стане деталізація елементної бази та перенесення перевіреної конфігурації у схему, що буде реалізована в KiCad.

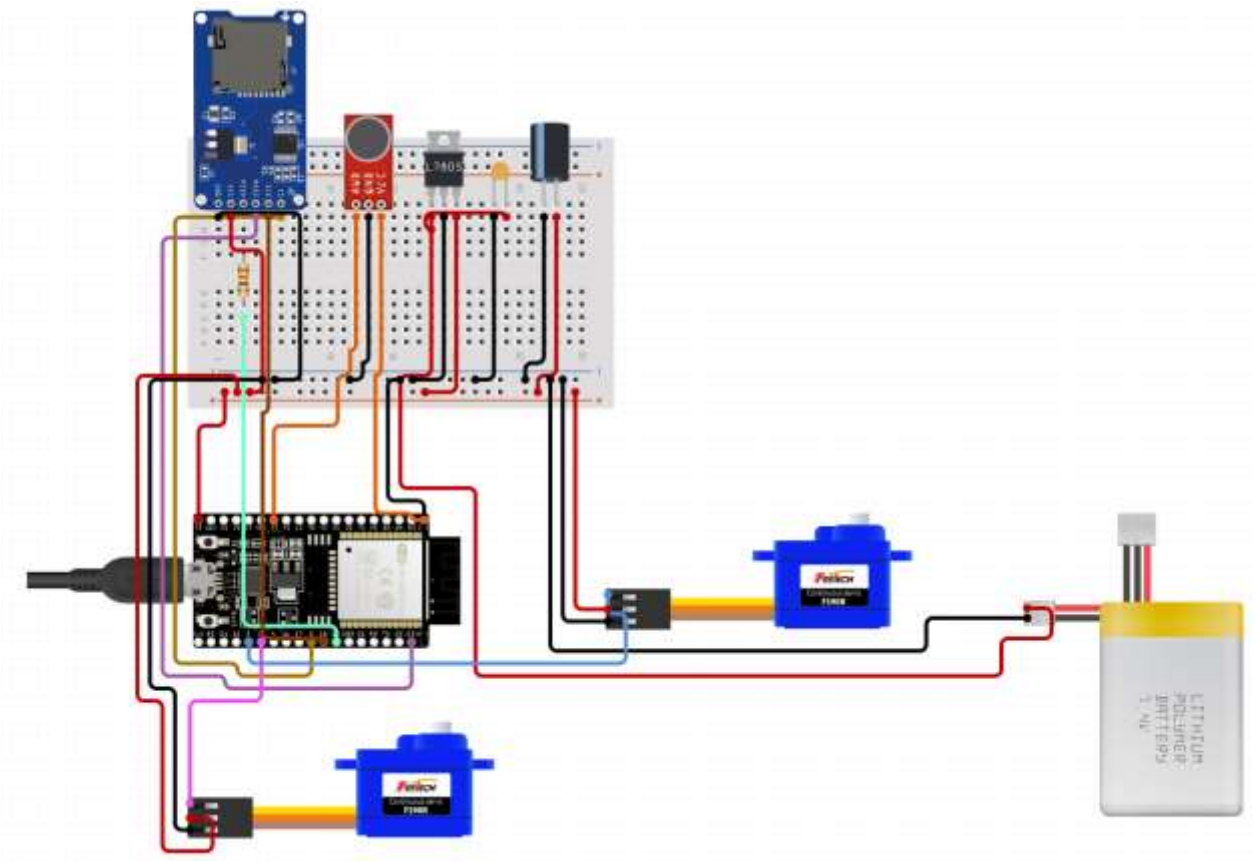


Рисунок 2.3 – Електрична схема в середовищі `circuito.io`

Де на рис 2.10 зображено ESP32-DevKitC малогабаритна плата розробки побудовану на базі ESP32-S3-WROOM-1. Вона має комбінований набір Wi-Fi та Bluetooth, а також оптимізовану розпіновку для створення прототипів на макетній платі.

Два мікросерводвигуна безперервного обертання SG90. Вони обертаються на 360 градусів повністю вперед або назад, замість того, щоб рухатися в одному положенні.

За допомогою літій-полімерного акумулятору відбувається живлення електронної схеми. Даний акумулятор має два активних елементи та видає 7,4 В. Оскільки це двоелементна батарея, тобто не є сумісною з одноелементними зарядними пристроями, вона потребує спеціальний зарядний пристрій.

L7805 це регулятор напруги 5 який отримує діапазон напруг на вхід та видає єдину та стабільну напругу на виході. До якого під'єднаний керамічний

конденсатор на 100 нФ це пасивний двополосний електричний компонент, який накопичує електричну енергію. У цій системі він передає електроенергію стабілізують напругу та потік потужності. Електролітичний конденсатор на 1 мкФ/50 В виконує таку ж саму задачу, а саме стабілізують напругу та потік потужності.

Роз'єм для електричного мікрофона SparkFun де цей модуль має на собі, мікрофон та підсилювач дозволяє виявляти стандартні звуки такі як голос, стукіт, оплески, шуми та інше. Повернуті значення рахуються від 0 (без звуку) до 1023 (максимальний звук). Потрібно припаяти 3 штекерні контакти для встановлення на макетну плату. Де AUD це контакт для обернутих значень, VCC живлення модулю, GND це заземлення.

Модуль Micro SD Дозволяє передавати дані на стандартну карту micro SD та з карти на плату модулю, модуль Micro SD може знадобитися якщо реалізація програмного забезпечення буде потребувати LLM для обробки голосових команд.

Використовуючи загальну документацію для мікропроцесора ESP32-S3-WROOM-1 та ESP32-S3-DevKitC-1 v1.1 модулю створимо власну схему в програмному забезпеченні Kicad [22, 8].

Створюємо новий проект та додаємо обраний мікропроцесор, для кожного піна мітку відповідно її призначення за документацією. Як наприклад для GND додаємо заземлення, а для піна 3V3 додаємо два фільтруючих конденсатора та резистор перемичку з нульовим опором.

Пін CHIP_PU відповідає вмикання мікросхему, коли на ній високий рівень тобто логічна одиниця, і скидається(перезавантажується), коли низький рівень тобто логічний ноль.

Коли ESP32 використовує системне живлення 3,3 В, шини живлення потребують деякого часу для стабілізації, перш ніж CHIP_PU буде підключено, і мікросхема буде ввімкнена. Тому CHIP_PU має бути переведений у високий рівень після того, як лінії 3,3 В будуть підключені.

Щоб скинути налаштування мікросхеми, підтримуйте напругу скидання у діапазоні від $-0,3$ до $0,25$ В. Щоб уникнути перезавантажень, спричинених зовнішніми перешкодами, потрібно зробити доріжку CHIP_PU якомога коротшою [9].

При цьому механічний перемикач SW_MEC_5E підключений до малоємного конденсатора який невілює помилку брязкіту контактів цього механічного перемикача. Якщо їхні контакти швидко замикаються та розмикаються протягом кількох мілісекунд, перш ніж стабілізуватися при натисканні або відпусканні. Цей дребезг може призвести до реєстрації кількох небажаних сигналів, а конденсатор в свою чергу згладжує стрибки напруги, тобто небажанні натискання (рис.2.4).

Крім того, передбачено можливість ручного перевodu мікроконтролера в режим програмування. Це досягається шляхом натискання кнопки BOOT, утримуючи її під час короткочасного натискання кнопки RESET. Такий підхід є стандартним для завантаження прошивки через UART інтерфейс за допомогою інструментів, таких як esptool або ESP-IDF Flash Tool.

Ця послідовність призводить до того, що пін GPIO0 при запуску перебуває у низькому логічному рівні, що сигналізує мікроконтролеру про необхідність входу у режим завантаження (bootloader mode). Якщо під час скидання GPIO0 знаходиться в логічній "1", то мікроконтролер просто завантажується з флеш-пам'яті у нормальному робочому режимі.

У режимі завантаження мікроконтролер слухає UART-порт зазвичай U0TXD/U0RXD і готовий приймати прошивку. Завдяки цьому режиму, розробник може легко оновлювати або змінювати прошивку без потреби у зовнішньому програматорі.

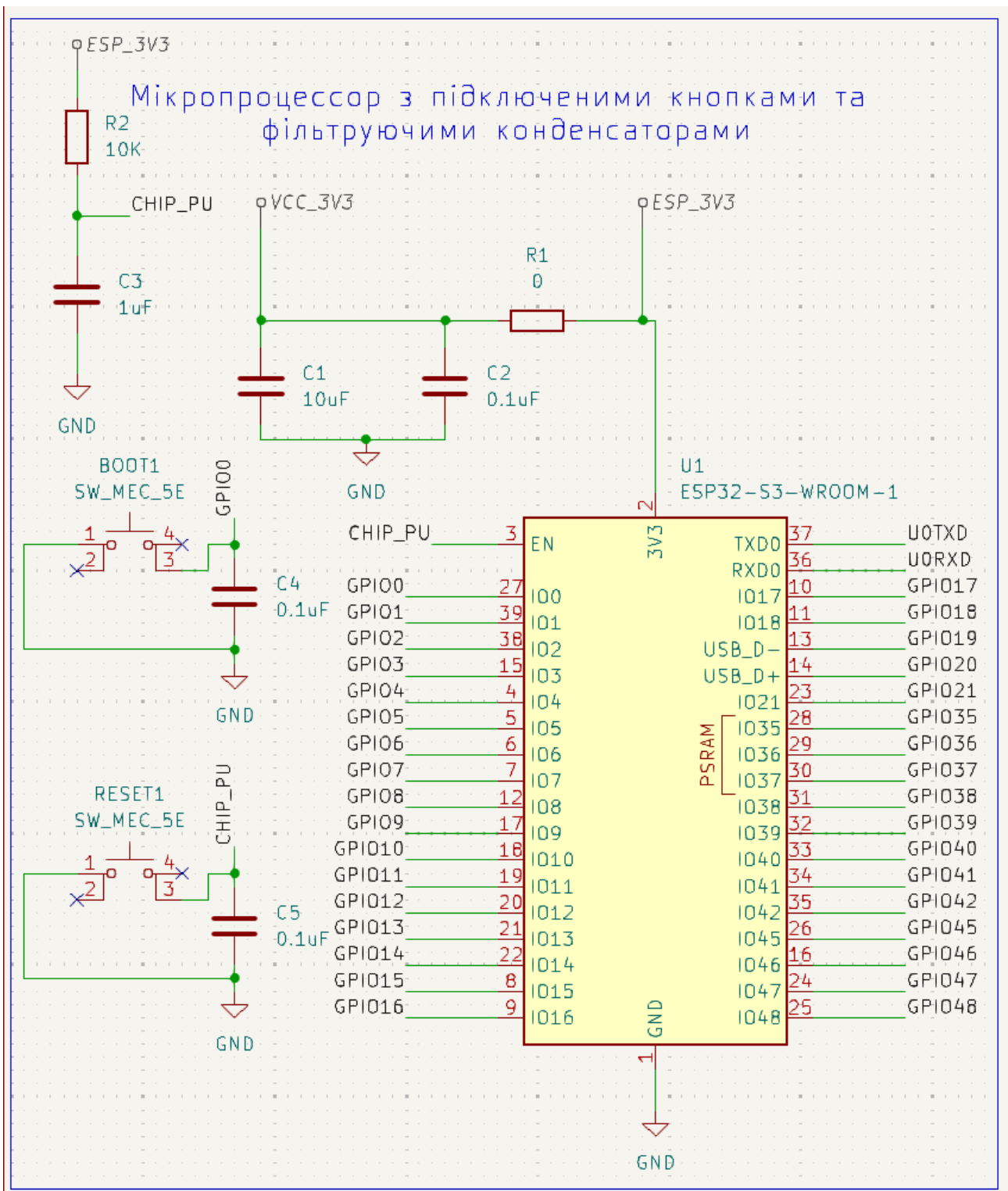


Рисунок 2.4 – Мікропроцесор з підключеними кнопками та фільтруючими конденсаторами

USB-UART та ESP USB виконують різні, але пов'язані функції в роботі мікропроцесора.

Інтерфейс USB-UART працює з конектором USB-C 2.0, який передає сигнали USB з комп'ютера в мікросхему CP2102N-Axx-xQFN28 (рис 2.6), де вона покриває послідовний зв'язок UART, що використовується мікроконтролером. Однак він не забезпечує живлення плати.

З іншого боку, ESP USB відноситься до вбудованої функціональності USB. Цей інтерфейс дозволяє пряме підключення USB без необхідності використання мікросхеми перетворювача. Він може обробляти програмування та послідовний зв'язок, а також забезпечує живлення плати через порт USB. Крім того, ESP USB підтримує розширені ролі USB-пристроїв, такі як робота в якості HID /CDC/MSD (наприклад, клавіатури або миші) або пристрою масової пам'яті. Проте ESP USB може бути повільнішим або менш стабільним, в порівнянні з USB-to-UART інтерфейсом (рис 2.5).

LES5D5.0CT1G – це діод TVS (Transient Voltage Suppression) з низькою ємністю, спеціально розроблений для захисту від електростатичного розряду у високошвидкісних лініях передачі даних, таких як USB. Проте стандартні бібліотеки KiCad не мають його в наявності, тому завантажимо цей елемент з відкритих джерел [23].

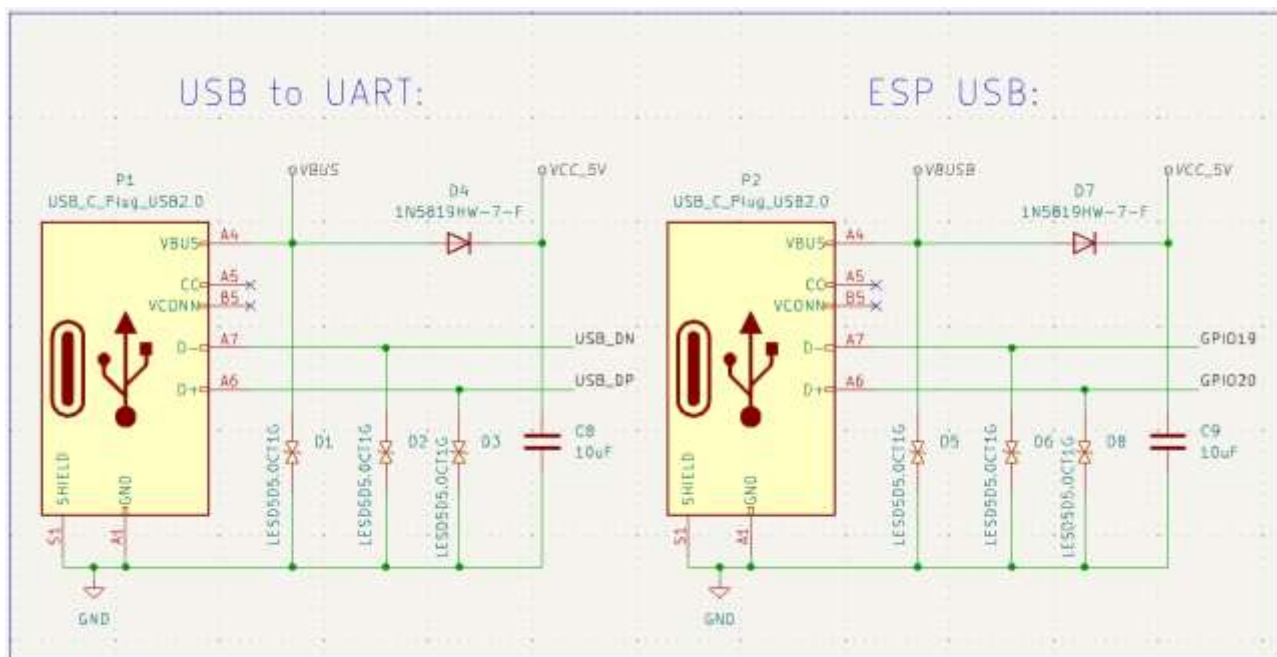


Рисунок 2.5 – USB to UART та ESP USB

Мікросхема CP2102N-Axx-xQFN28 працює як високоінтегрований контролер мостів USB-UART забезпечуючи просте рішення використовуючи мінімум компонентів та місця на друкованій платі. Він в основному відповідає за програмування ESP32 шляхом передачі прошивки через послідовний завантажувач та за забезпечення послідовного зв'язку для налагодження (рис 2.6).

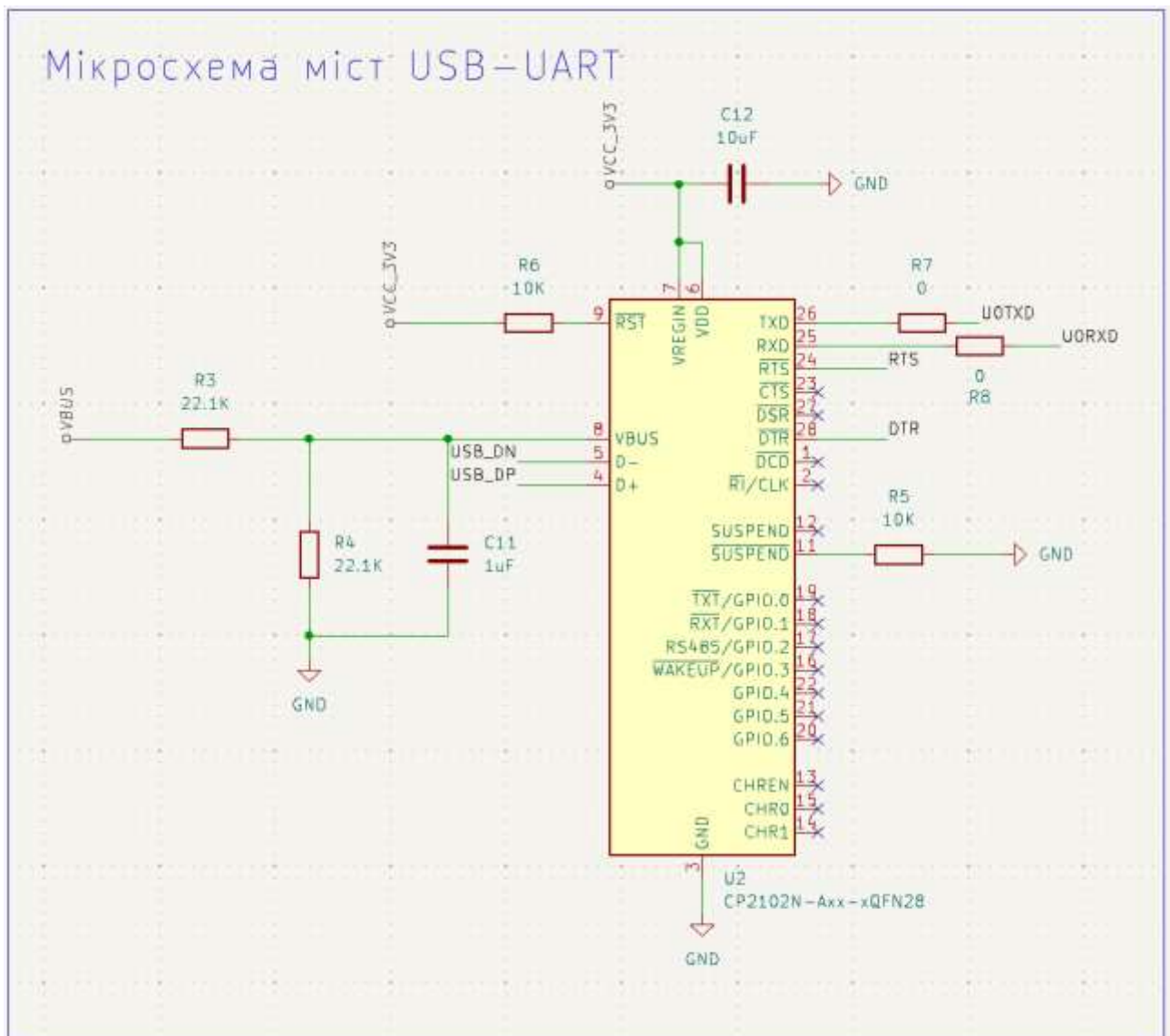


Рисунок 2.6 – Мікросхема міст USB-UART (CP2102N-Axx-xQFN28)

DTR або Data Terminal Ready це сигнал вказує, що термінал (наприклад, комп'ютер або USB-UART міст) готовий до зв'язку. У випадку ESP32, його

зазвичай використовують, щоб вивести мікроконтролер з режиму сну або підготувати до перезавантаження.

RTS або Request to Send це сигнал вказує, що пристрій хоче надіслати дані. У випадку з ESP32, RTS часто використовується для скидання (RESET) мікроконтролера, коли надходить нова прошивка.

Цей каскад використовуються для автоматичного скидання ESP32 та переведення його в режим завантажувача (програмування) без ручного натискання кнопок. Схема використовує два NPN-транзистори (Q1 та Q2) для керування контактами EN (скидання) та GPIO0 (вибір режиму завантаження) ESP32.

Коли обидва контакти, DTR та RTS, мають високий рівень, мікропроцесор працює в звичайному режимі. Якщо RTS стає низьким, а DTR залишається високим, контакт EN стає низьким, що призводить до скидання. Якщо DTR стає низьким, а RTS залишається високим, контакт EN стає низьким. Коли обидва контакти, DTR та RTS, разом стають низькими, обидва контакти, EN та GPIO0 стають низькими, що скидає ESP32, утримуючи GPIO0 низьким, таким чином переводячи його в режим завантажувача. Це дозволяє автоматичне перепрошивання прошивки. Після спалаху сигнали повертаються до високого рівня, і ESP32 відновлює нормальну роботу (рис 2.7).

SK6812MINI – це мініатюрний інтелектуальний RGB-світлодіод із вбудованим драйверним чіпом, розроблений для компактних або обмежених у просторі освітлювальних застосувань. Це менший варіант популярного світлодіода SK6812 (який сам по собі схожий на WS2812), який часто використовується в адресованих світлодіодних матрицях, носимій електроніці та компактних дисплеях (рис 2.7).

Забезпечує повний 24-бітний колір (8 бітів на канал: червоний, зелений, синій). Використовує однопровідний протокол для передачі даних, що дозволяє послідовне з'єднання багатьох світлодіодів. В випадку цієї схеми працює від напруги 3,3 В, проте може працювати і більшому спектрі напруг до 5,5В.

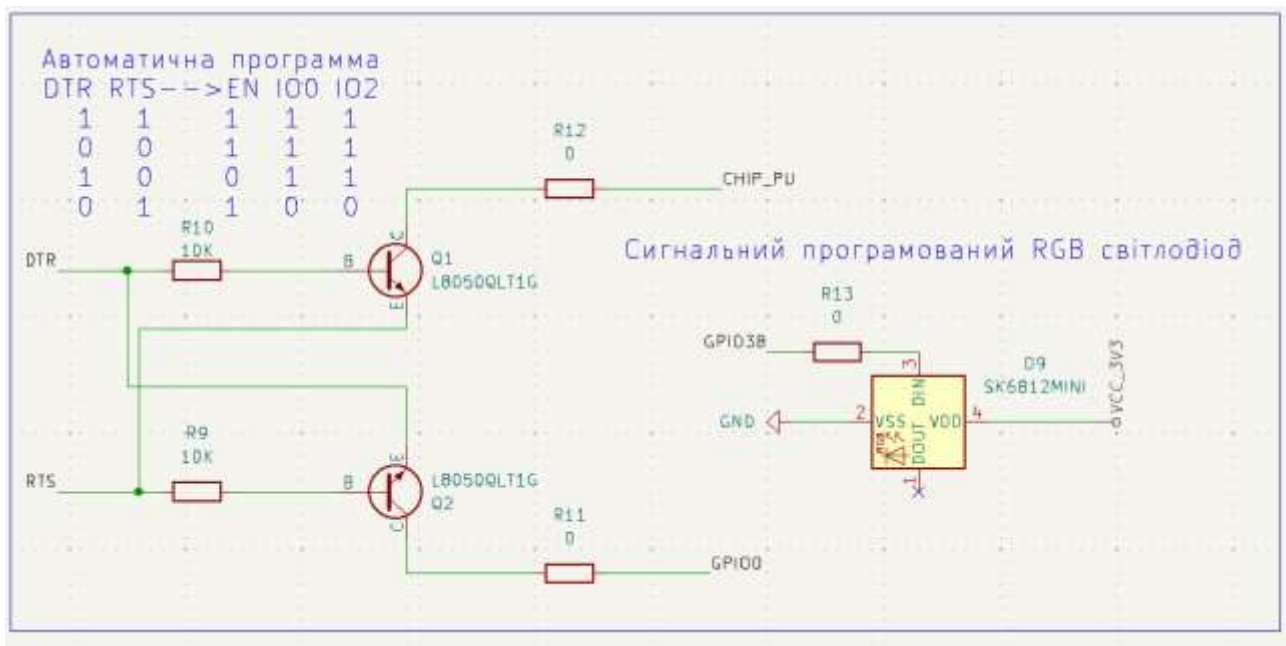


Рисунок 2.7 – Автоматична програма DTR, RTS та Сигнальний програмований RGB світлодіод

SGM2212-3.3XKC3G/TR – це лінійний стабілізатор з низьким рівнем шуму, низьким струмом спокою, високим PSRR, швидкою перехідною характеристикою та низьким падінням напруги, розроблений з використанням технології CMOS.

Проте в стандартних бібліотеках KiCad немає цього лінійного регулятора напруги, загальним пошуком не було виявлено цього електричного елемента, тому створимо його за допомогою вбудованого редактора елементів. Використовуючи прямокутник створюємо форму елемента, задаємо чотири виводи відповідно документації [7, 16] (рис 2.8).

Створюємо загальний каскад лінійного регулятора з фільтруючими конденсаторами та сигнальним світлодіодом (рис 2.9).

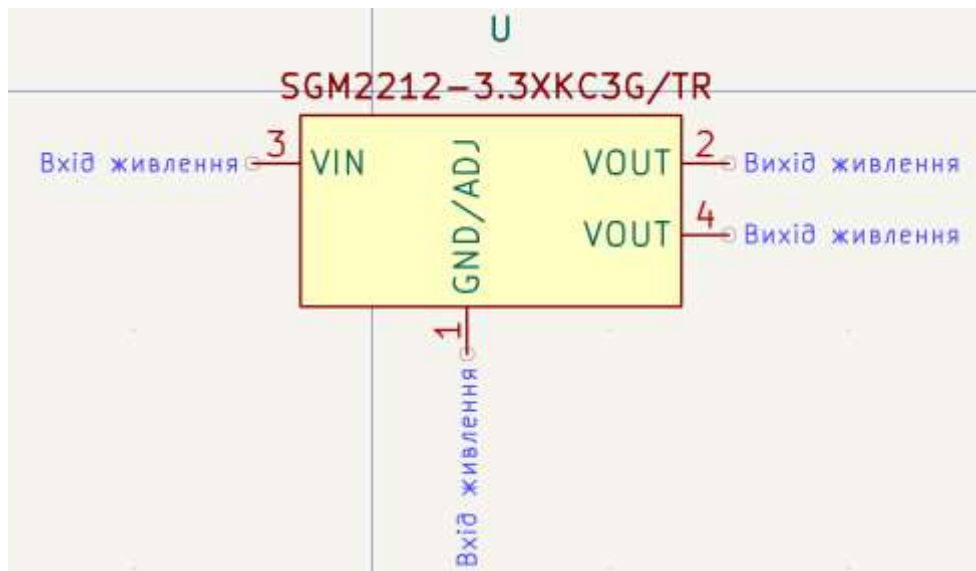


Рисунок 2.8 – SGM2212-3.3XKC3G/TR лінійний стабілізатор

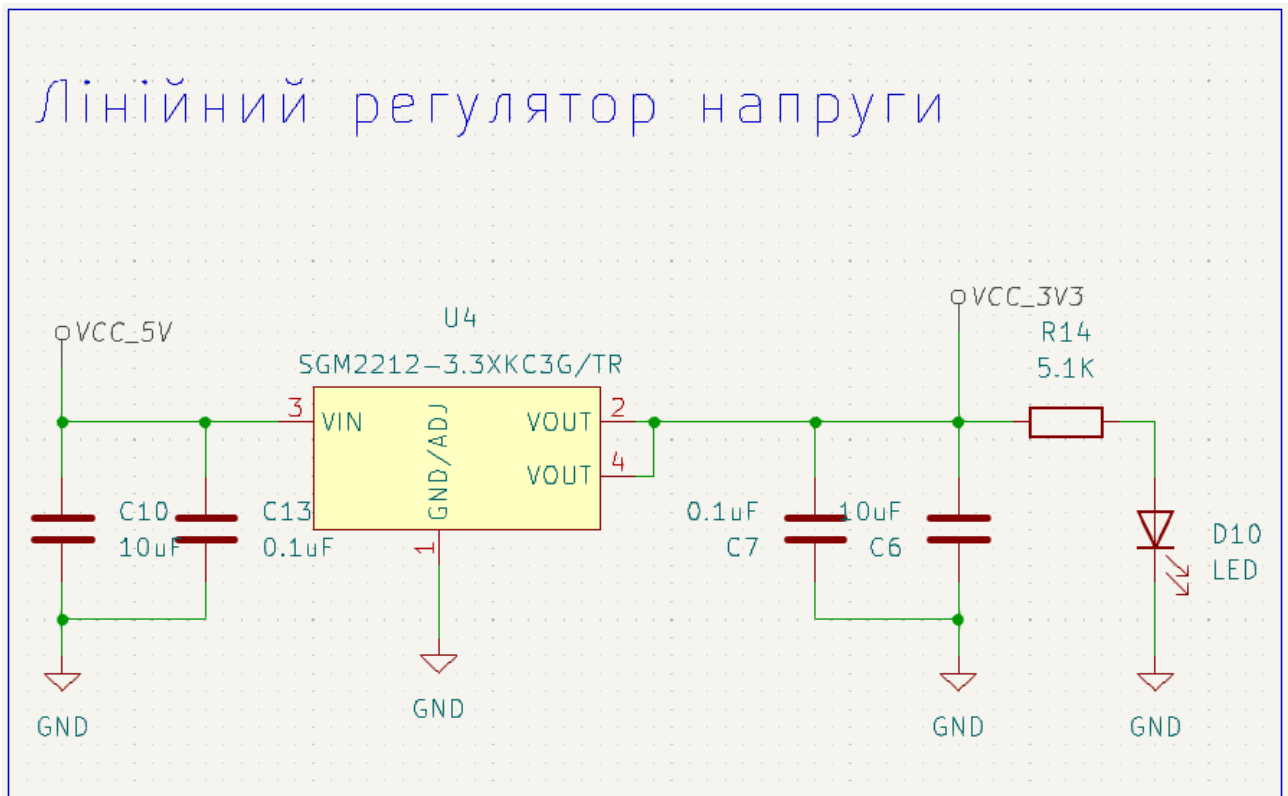


Рисунок 2.9 – Лінійний регулятор напруги

Pin Header являє собою два 22-контактні роз'єми. Роз'єм J1 містить контакти живлення, такі як ESP_3V3, VCC_5V та GND, а також контакти керування та загального призначення, такі як CHIP_PU та GPIO4 – GPIO18.

Роз'єм J2 містить додаткові контакти GPIO0 – GPIO42 та лінії зв'язку, такі як U0TXD та U0RXD. GND також підключено до обох кінців кожного роз'єму для заземлення (рис 2.10).

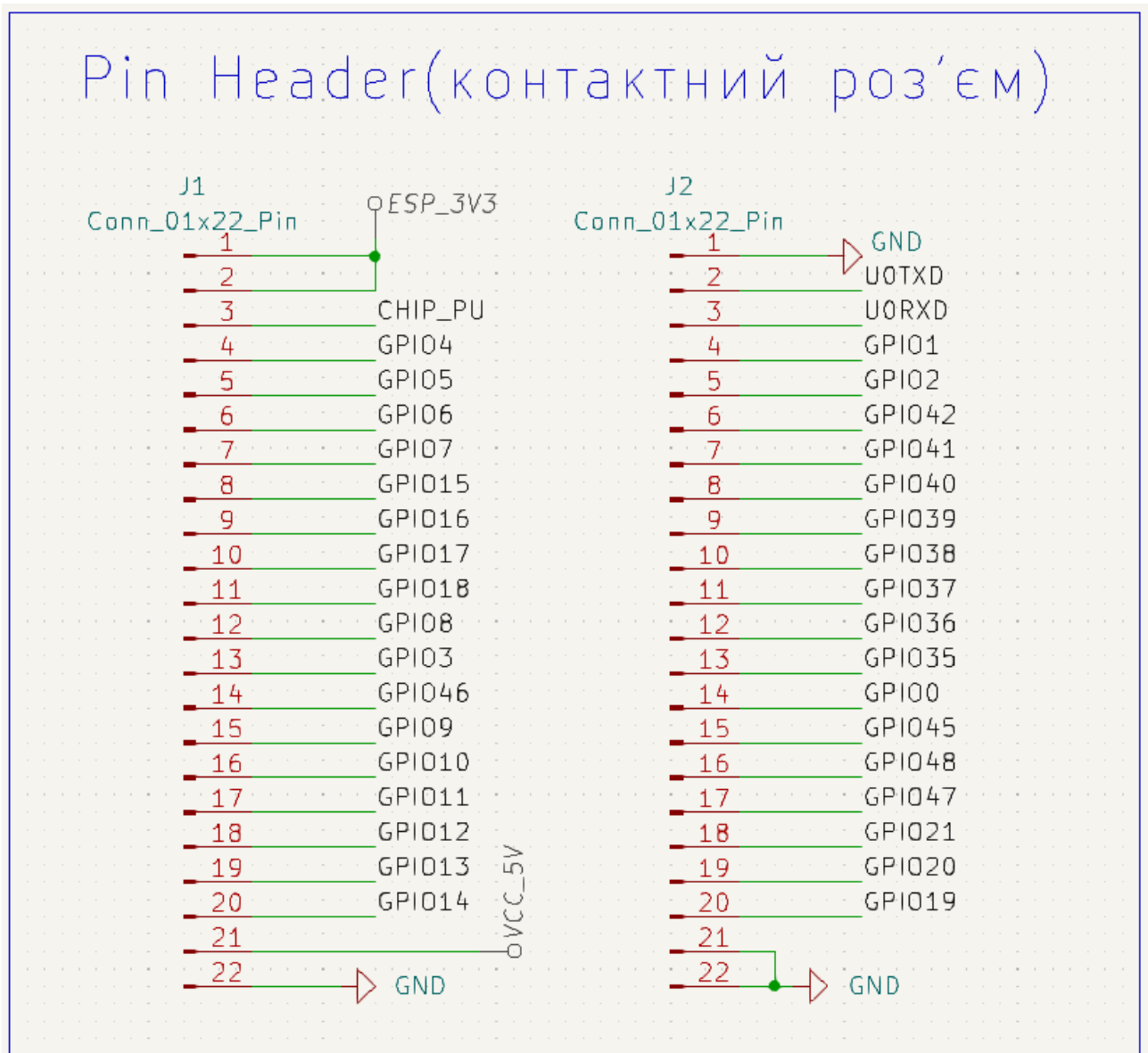


Рисунок 2.10 – Контактні роз'єми відповідно виводів мікропроцесора

Модуль INMP441ACEZ-R7 – це цифровий MEMS-мікрофон із інтерфейсом I2S (Inter-IC Sound), який дозволяє безпосередньо підключати його до мікроконтролерів або мікропроцесорів без потреби в аналого-цифровому перетворювачі. Характеристики самого чипу наведені вище за текстом (рис 2.11).

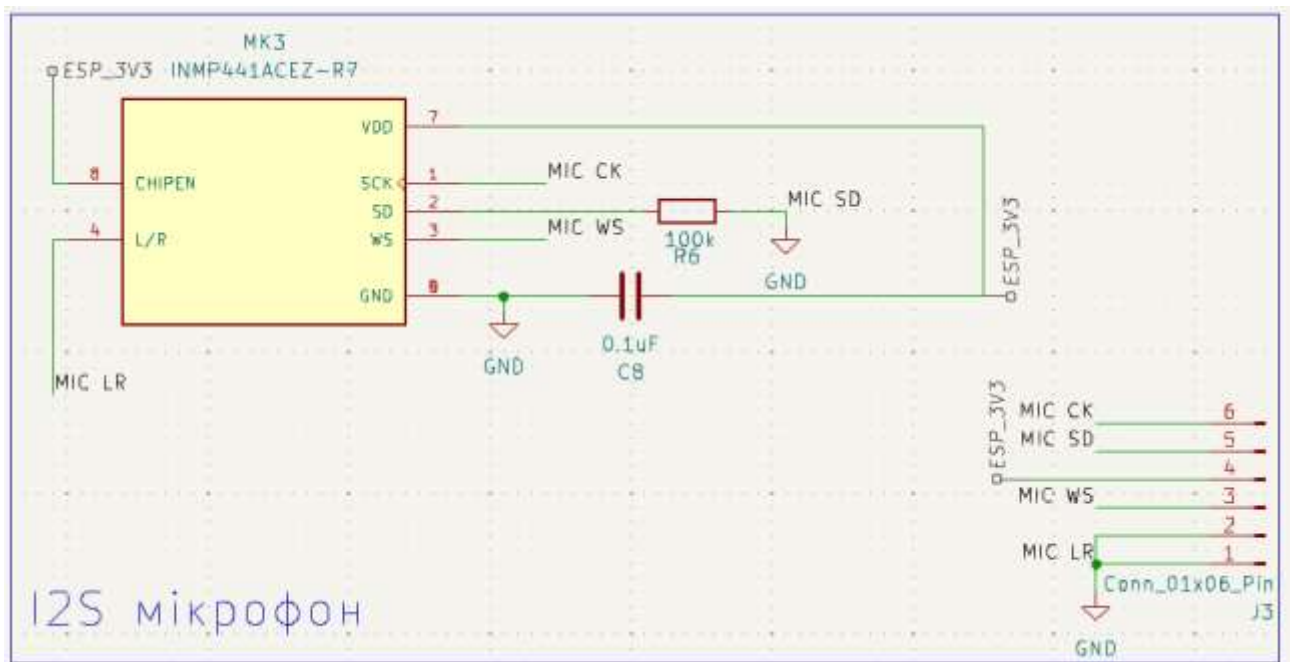


Рисунок 2.11 – Модуль I2S мікрофона INMP441ACEZ-R7

Загальна схема підключення модулю до контактних роз'ємів ESP32 де, MIC SD під'єднується до GPIO10, MIC WS під'єднується до GPIO11, MIC SCK під'єднується до GPIO12, VDD під'єднується до ESP_3V3, GND та L/R до GND.

Двоє серводвигунів під'єднані до мікроконтролера за допомогою контактів GPIO сигналів керування та зовнішнього джерела живлення 5 В для роботи. Кожен серводвигун має три з'єднання: перший контакт (PWM) першого серводвигуна підключений до GPIO13, а перший контакт другого серводвигуна до GPIO14, обидва з яких використовуються для подачі сигналів керування ШІМ від мікроконтролера. Другі контакти обох серводвигунів підключені до спільного джерела живлення VCC_5V, а їхні треті контакти підключені до спільної землі. Така схема являє собою типовий метод живлення та керування серводвигунами, забезпечуючи отримання ними достатнього живлення, зберігаючи при цьому спільне заземлення для надійної передачі сигналу між мікроконтролером та серводвигунами (рис 2.12).

Серводвигун та його джерело живлення

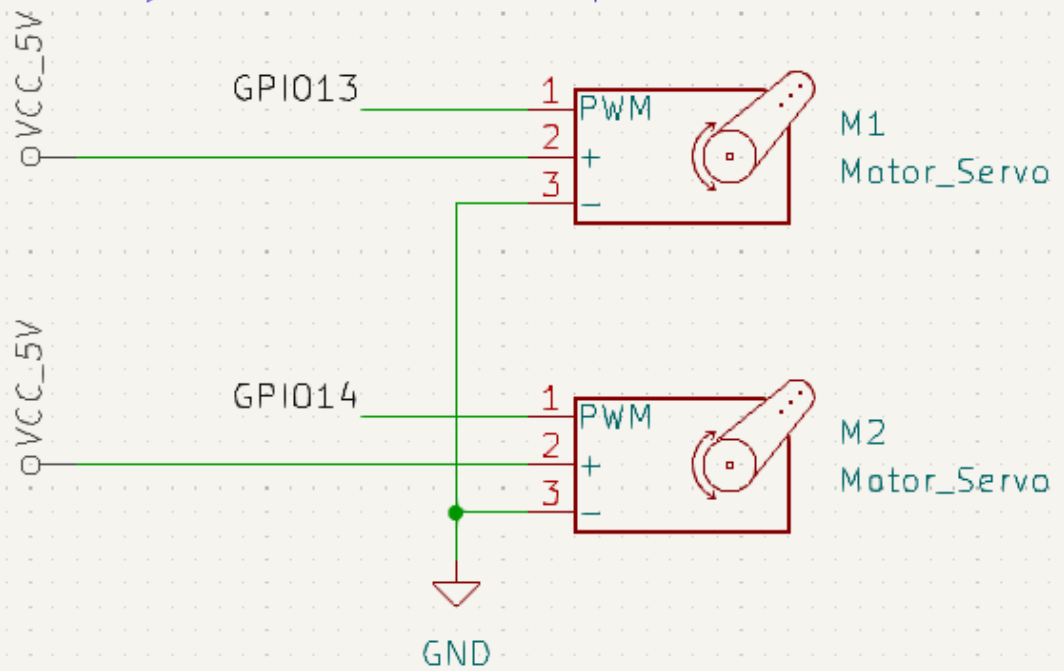


Рисунок 2.12 – Загальна схема підключення серводвигунів

В додатку А позначено креслення загальної принципової схеми.

3 РОЗРОБКА ДРУКОВАНОЇ ПЛАТИ

На основі визначених технічних умов та чіткого розуміння того, що має робити пристрій, які компоненти будуть використовуватися, які розміри повинна мати плата, яке живлення буде використовуватись, які є обмеження по споживанню струму, температурі, та іншим параметрам, можна приступити до розробки друкованої плати, зі створеною електричною принциповою схемою.

Першочерговим кроком в її розробці буде, додавання посадкових місць так званих footprint кожному елементу плати. Як ось з мікропроцесором ESP32-S3-WROOM-1 (рис 3.1).

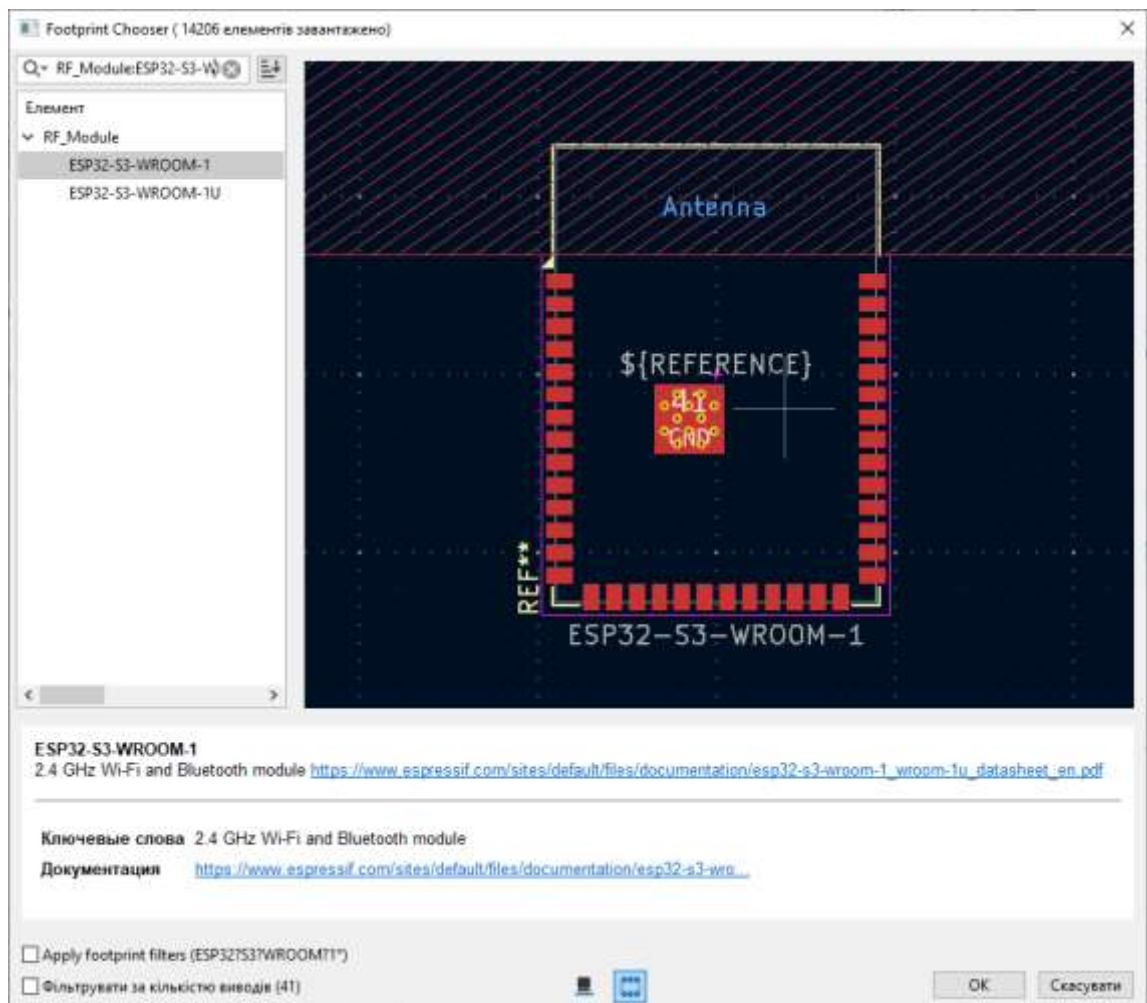


Рисунок 3.1 – Вибір посадкового місця мікропроцесора

Проте дрібні компоненти такі як конденсатори та резистори мають багато різних посадкових місць. Сам номінал не визначає розмір компонента, тому необхідно оцінити тип монтажу, потужність, робочу напругу та обмеження за розміром.

Перш за все, слід визначити тип монтажу: SMD (поверхневий) або ТНТ (в отвори). Плата розробляється з прицілом на сучасне виготовлення та компактність тому варто вибирати SMD. Це найпоширеніший варіант у більшості нових проєктів.

Далі потрібно врахувати робочі умови. Для резисторів важливий показник потужності. Наприклад, резистори з розміром 0402 зазвичай витримують до 0.063 Вт, 0603 до 0.125 Вт, а 0805 до 0.25 Вт (рис 3.2).

Для керамічних конденсаторів головне максимальна робоча напруга, яка прямо залежить від їх розміру: 0402 зазвичай до 16 В, 0603 до 25 В, 0805 до 50 В. Якщо напруга і потужність не визначені, найкращим універсальним вибором буде розмір 0603, він зручний у роботі та достатньо компактний (рис 3.3).

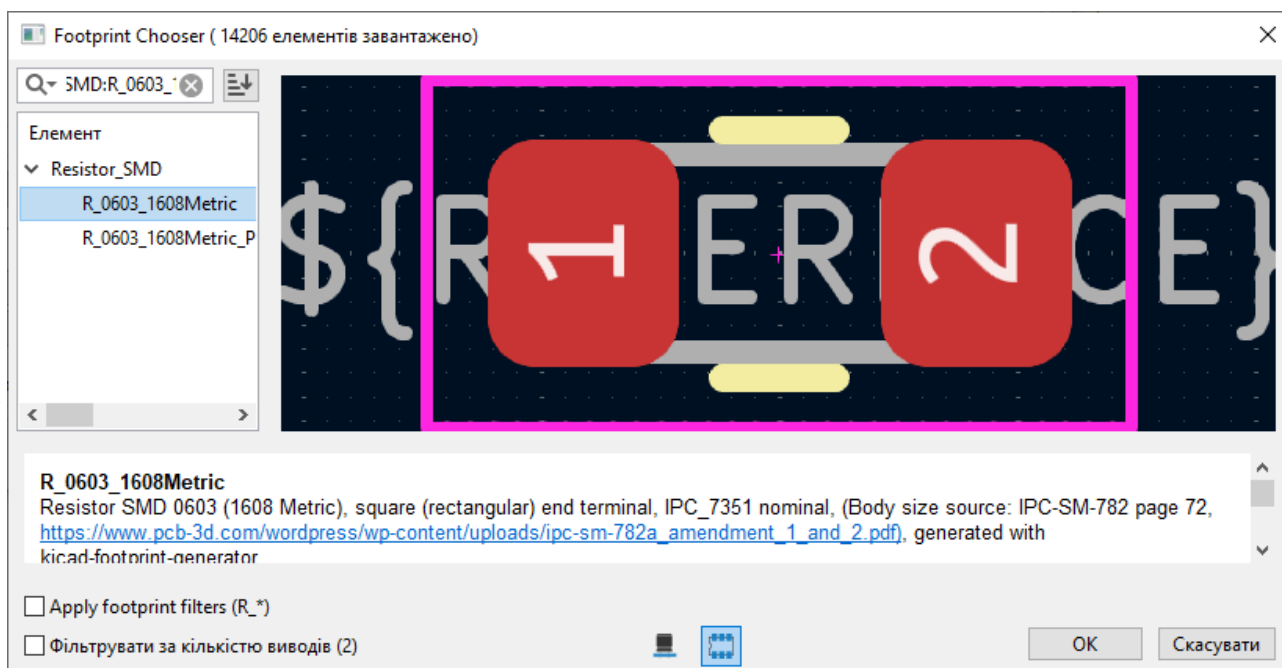


Рисунок 3.2 – Посадкове місце резистора з номіналом 10 кОм (1%)

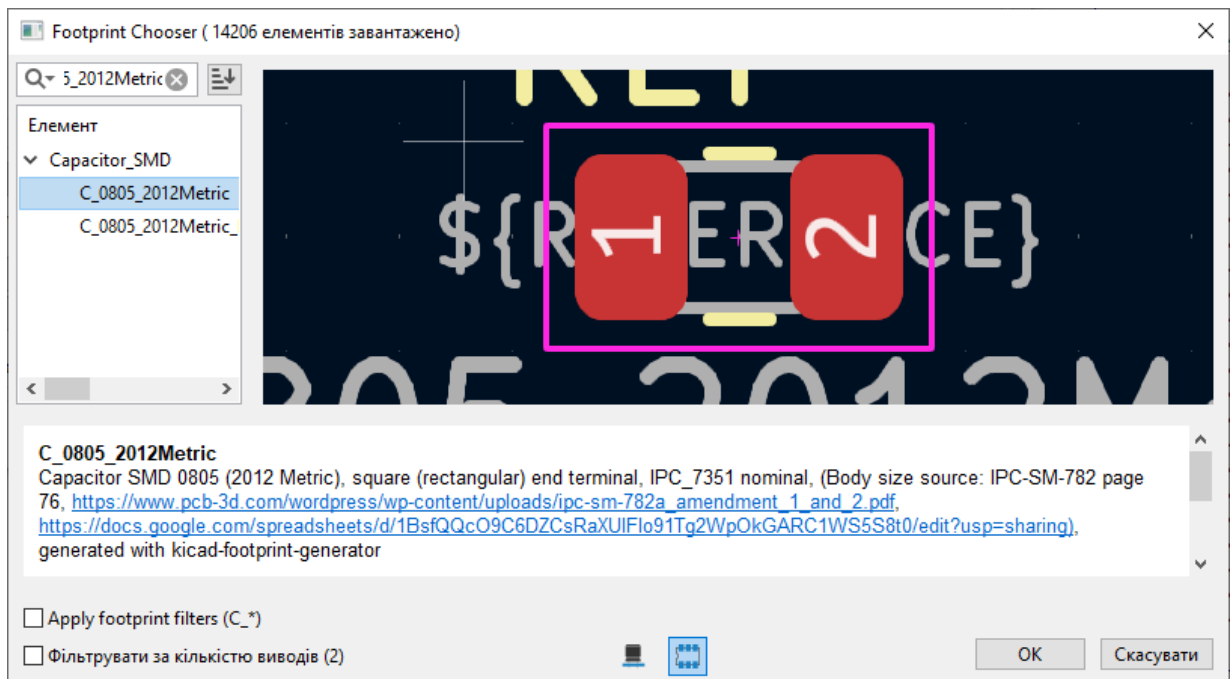


Рисунок 3.3 – Посадкове місце конденсатора з номіналом 1 мкФ 16В
(10%)

Елементи які не мають посадкових місць, такі як ті що створювалися в вбудованому редакторі символів KiCad та імпортовані з інтернет простору вище в попередньому розділі.

З огляду на те потрібно створити відповідні посадкові місця. Використовуючи відповідну документацію в якій можна знайти фізичні розміри посадкових місць та загальні розміри керамічних корпусів [16, 18, 14, 21, 15, 19].

Додаємо контактні площадки налаштовуючи розміри та загострюючи їх кути. Шовкографієвим шаром (F.Silkscreen) позначаємо на платі загальні розміри корпусу відповідного елемента та його номер чи назву, наприклад R2 та BOOT1. Іншим шаром (F.Fab) позначаємо область з інформацією про монтаж посадкового місця зверху плати. Останнім шаром (F.Courtyard) область встановлення посадкового місця (рис 3.4).

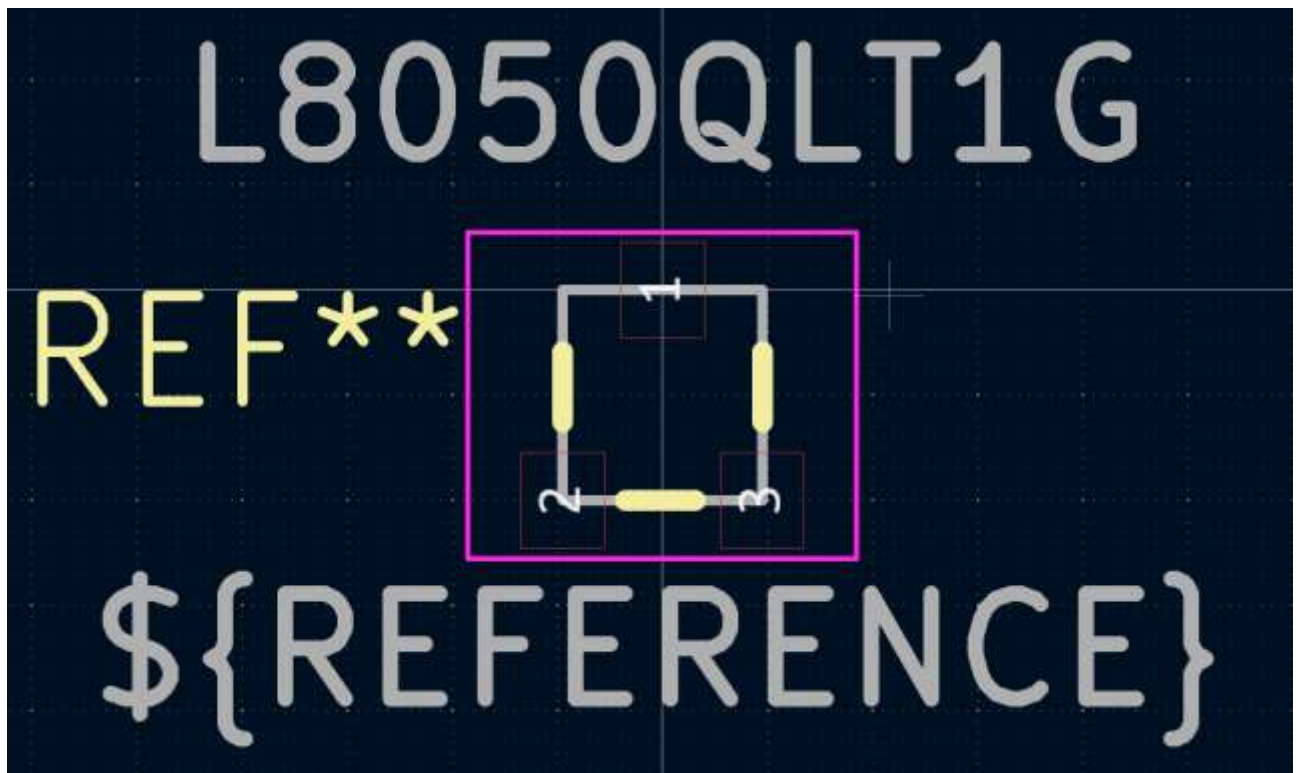


Рисунок 3.4 – Створення посадочного місця для транзистора
L8050QLT1G

Щоб не створювати посадкові місця імпортуємо їх з інтернет джерел попередньо перевіривши відповідність розмірів та контактних площин з документацією та елементами еквівалентної схеми. Для більшої наглядності в програмному забезпеченні KiCad є можливість додавання 3D моделей елементів, згідно з чим додаємо їх до посадкових місць які не мають своїх 3D моделей, деякі вбудовані посадкові місця в стандартних бібліотеках KiCad також їх не мають. При цьому не обов'язково використовувати моделі самих елементів, можна використовувати подібні до оригіналу (рис 3.5).

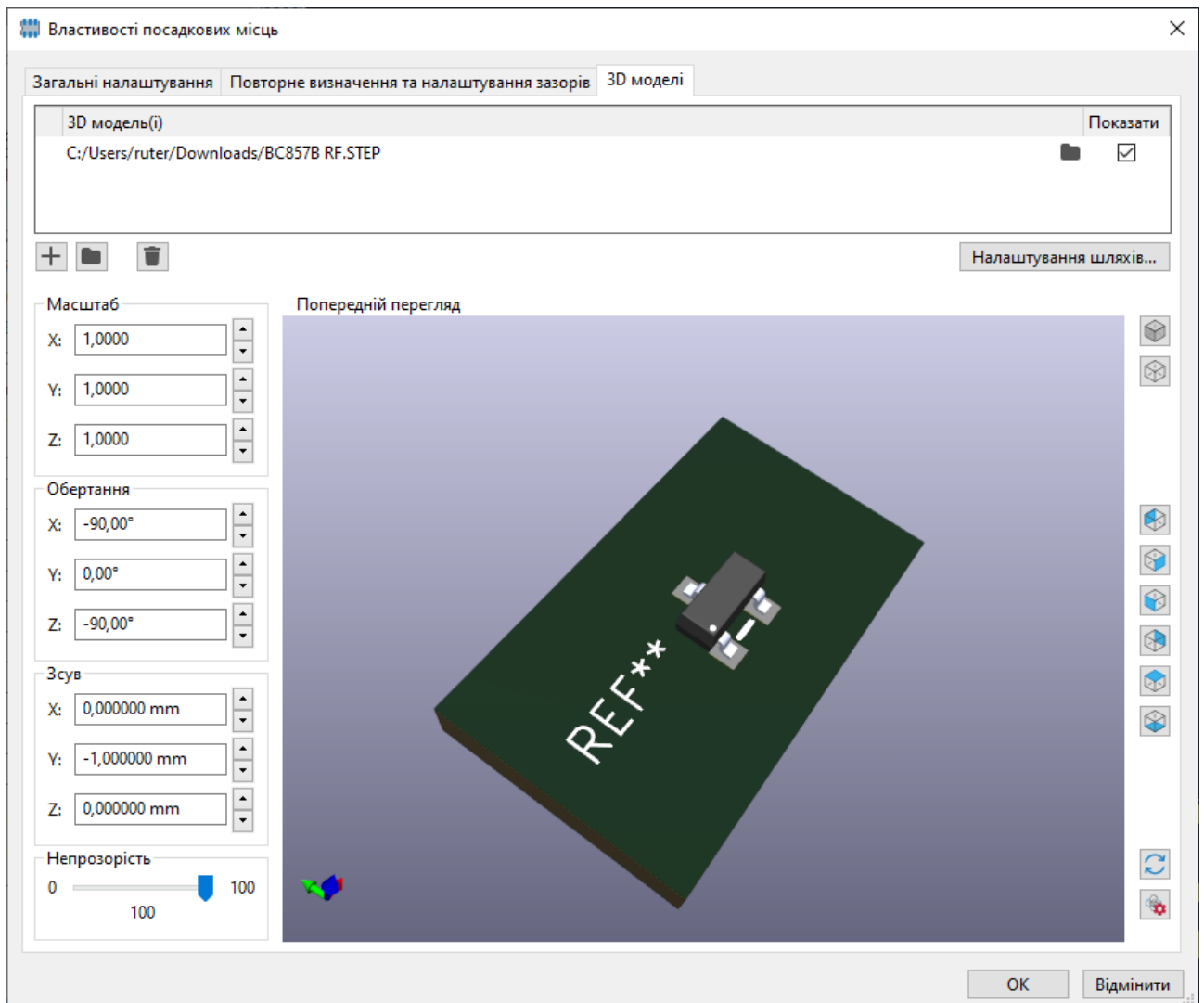


Рисунок 3.5 – 3D модель транзистора BC857B використана для посадкового місця L8050QLT1G

Після додавання усіх посадкових місць у редакторі друкованих плат виставляємо в налаштуваннях плати два шару міді та оновлюємо плату з принципової схеми [4].

Відповідно до правил компоновання друкованих плат розділові конденсатори слід розміщувати якомога ближче до контактів живлення інтегральних схем, щоб придушити шум та коливання напруги (рис. 3.6, рис. 3.7).

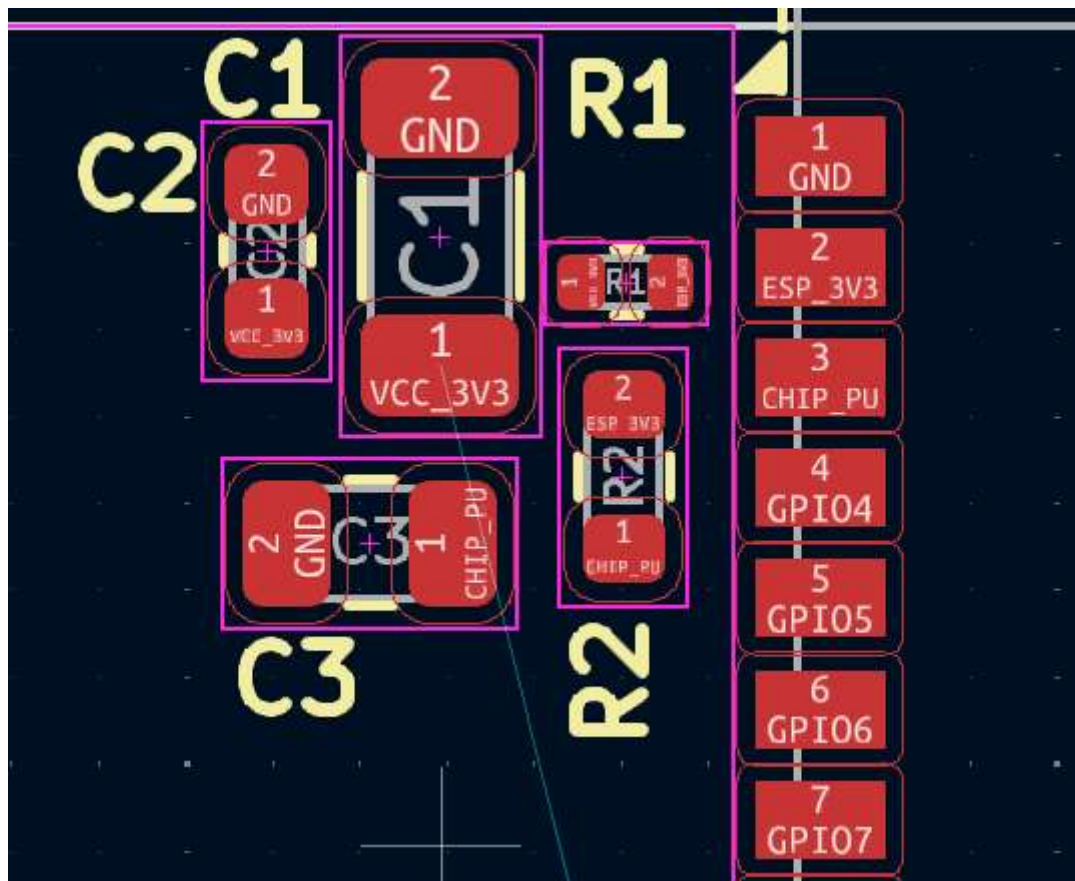


Рисунок 3.6 – Мікропроцесор з каскадом фільтруючих конденсаторів

Ширина мідних доріжок повинна бути достатньою для роботи зі струмом, не спричиняючи надмірного нагрівання або падіння напруги. Сигнальні лінії зазвичай використовують ширину від 2 до 6 міл, тоді як лінії електропередач потребують набагато ширших доріжок, зазвичай 6–10 міл. Розробники використовують калькулятори ширини доріжок для визначення відповідної ширини на основі струму, товщини міді та підвищення температури. В нашому випадку ширина доріжок становить 0,2 мм (рис. 3.8).

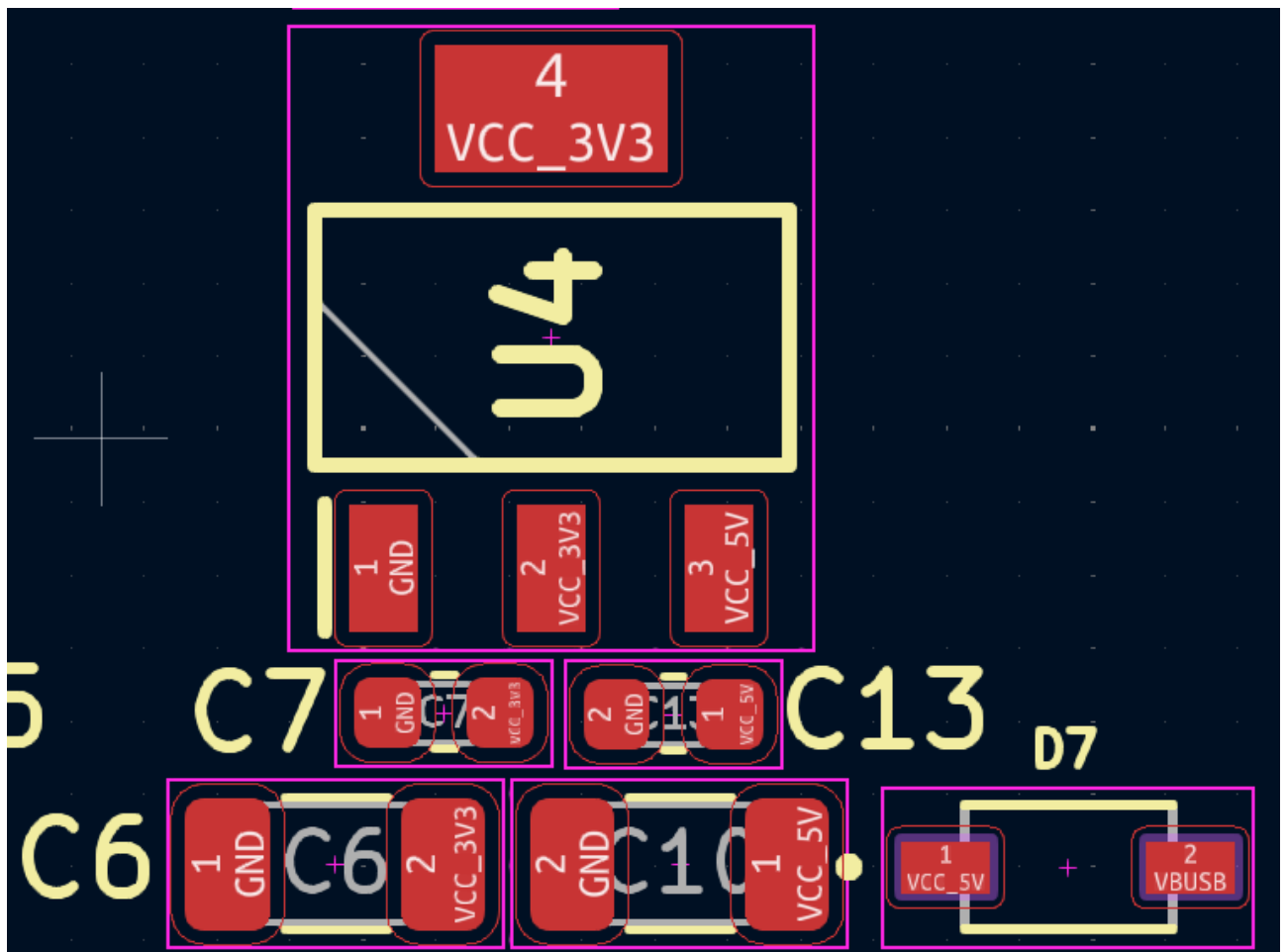


Рисунок 3.7 – Блок лінійного регулятора напруги

Диференціальні пари (наприклад, USB DN та DP) повинні бути прокладені з однаковою відстанню та однаковою довжиною для підтримки цілісності сигналу [5].

Аналогічно, забезпечення чистого та безперервного зворотного шляху для високошвидкісних сигналів, зазвичай через безперервну заземлювальну площину, має вирішальне значення для зменшення електромагнітних перешкод та відбиттів сигналу (рис 3.9).

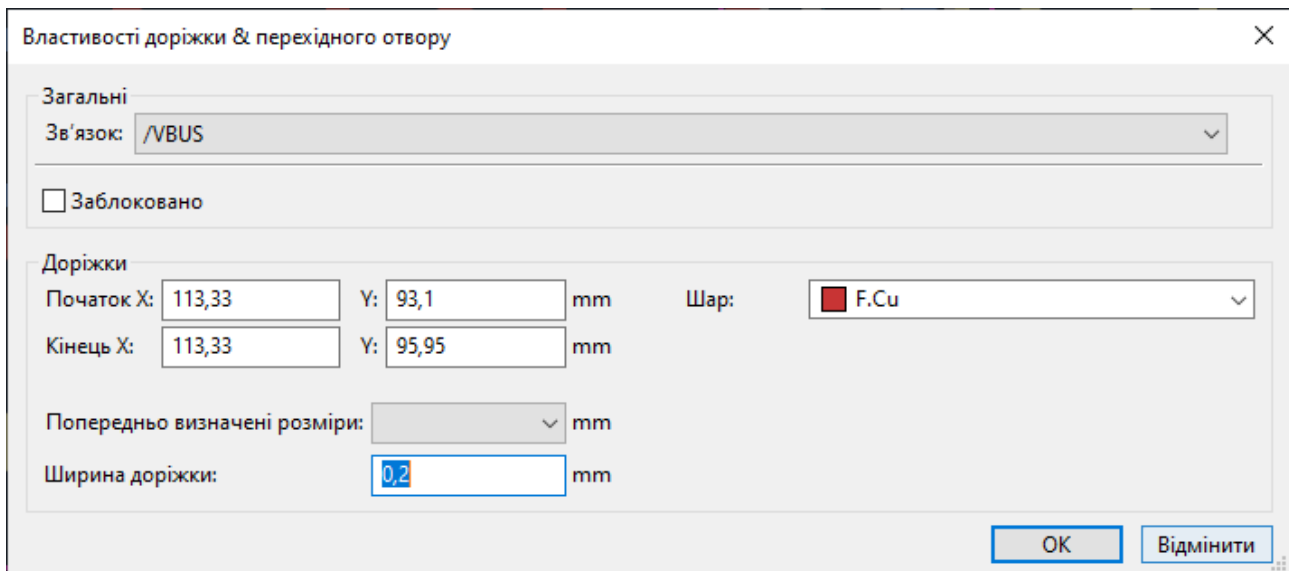


Рисунок 3.8 – Ширина доріжки електропередачі

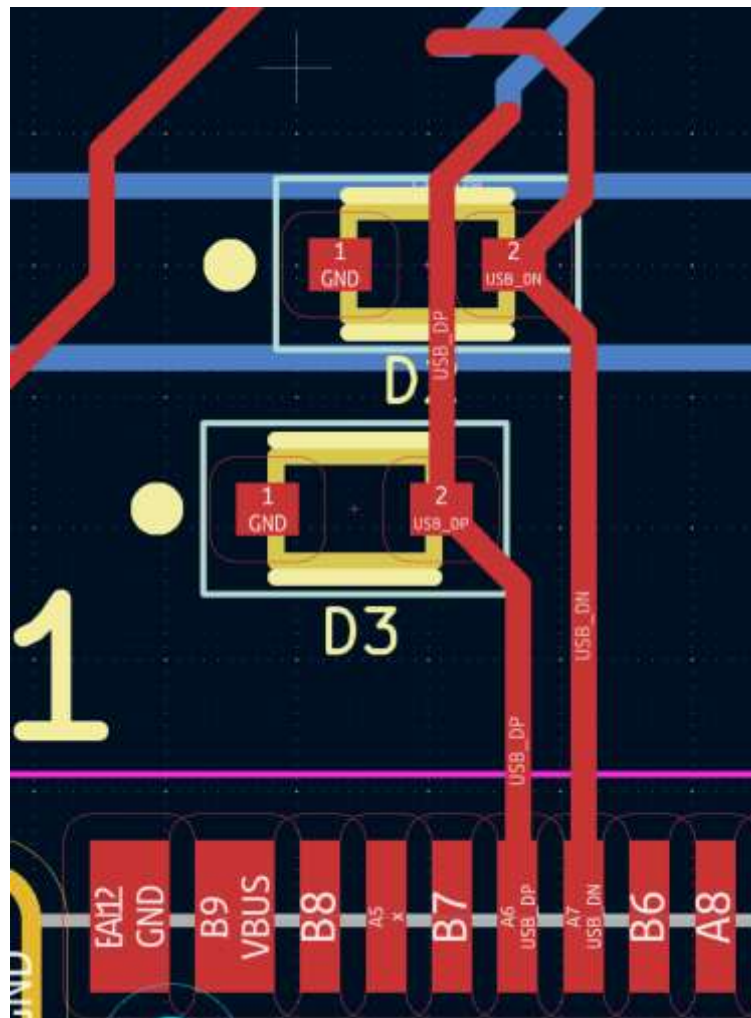


Рисунок 3.9 – Диференціальна пара USB_DN та USB_DP

Розміри перехідних отворів визначаються діаметром отвору та навколишньою контактною площадкою. Стандартні наскрізні перехідні отвори зазвичай мають розмір отвору 0,3 мм та контактну площадку 0,5 мм (рис 3.10) [6].

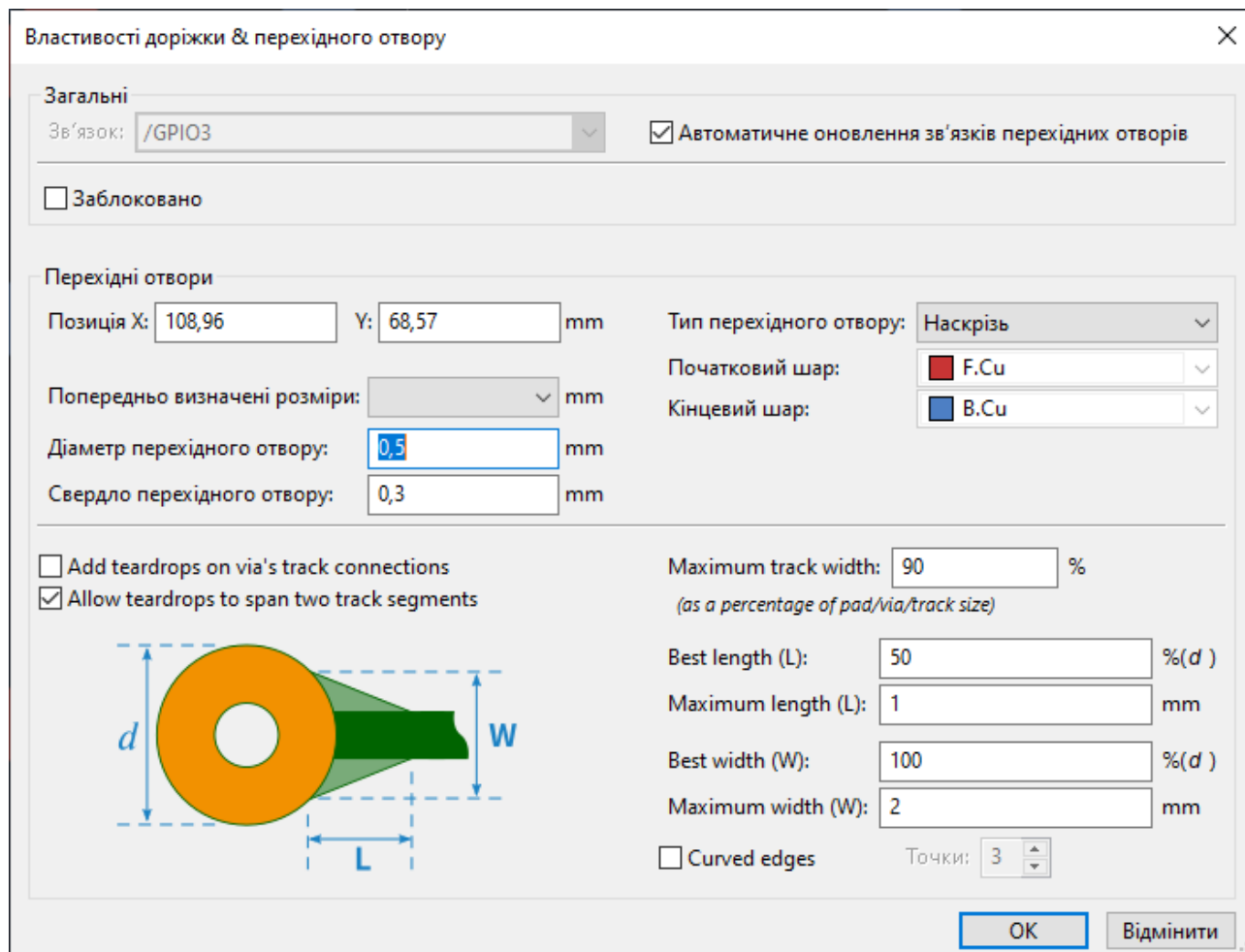


Рисунок 3.10 – Розмір перехідних отворів

Шаром Edge_Cuts визначаємо контур або периметр плати, по суті описуючи фізичну форму та розмір кінцевої виготовленої друкованої плати який не може бути більше ширини оболонки протезу до якої плата буде кріпитися. Друкована плата має наступні фізичні характеристики, де її висота дорівнює 61,31 мм а ширина 57,15 мм.

Заземлення реалізуємо за допомогою суцільної мідної заливки площини плати, додаючи заповнену зону яка буде впливати на обидва шари плати та мати зв'язок з заземленням. Використання окремих площин для живлення та

заземлення допомагає підтримувати низькоімпедансні шляхи та ефективний розподіл струму. Зшивання перехідних отворів використовується для з'єднання заземлювальних площин на кількох шарах, покращуючи шляхи повернення сигналу та характеристики електромагнітних перешкод (рис 3.11).

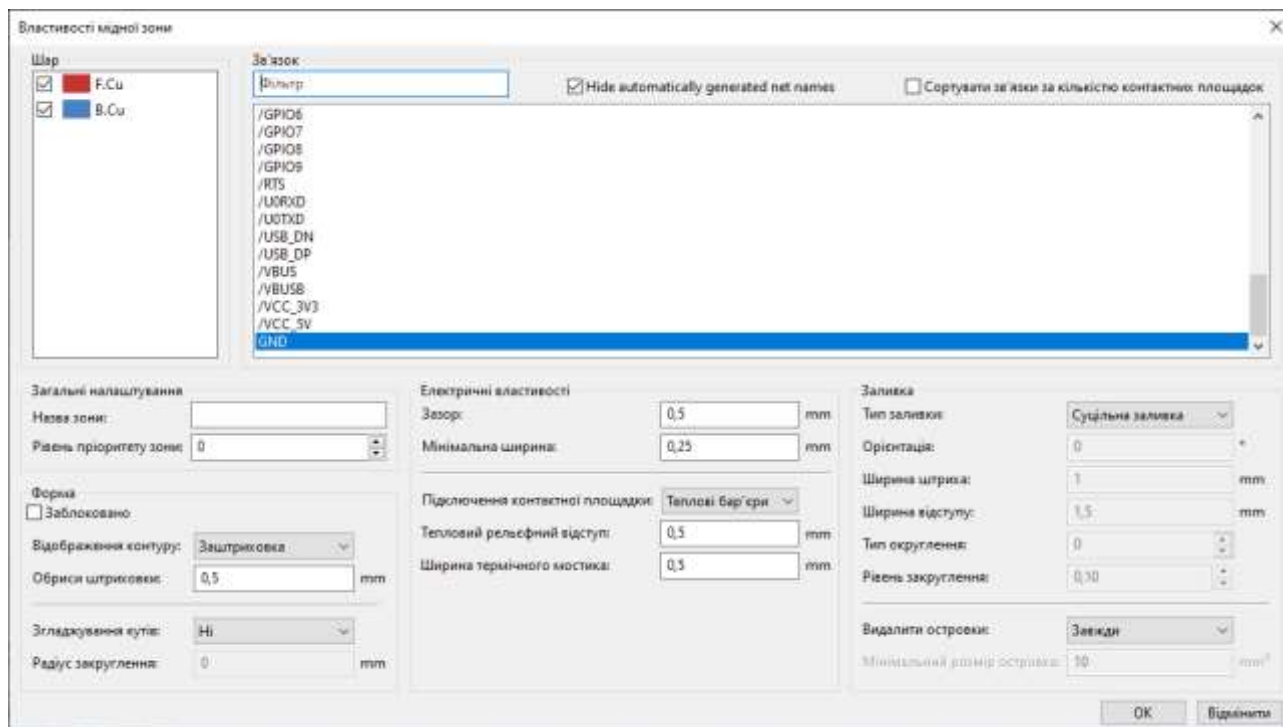


Рисунок 3.11 – Загальне заземлення за допомогою площин перехідних отворів

Після розміщення компонентів на платі у редакторі PCB Layout, було увімкнено режим трасування, де з урахуванням правил дизайну (DRC), таких як мінімальна ширина доріжок, відступи між ними та допустимі кути, вручну прокладалися з'єднання між контактами компонентів. Для зручності використовувалися сітка та прив'язка, а також автоматичне вирівнювання доріжок під час обведення перешкод.

Після виконаної роботи отримано конкретний результат – розроблено схему друкованої плати, а також створено її тривимірне зображення, що дозволяє наочно оцінити конструкцію та компонування елементів (рис 3.12, 3.13).

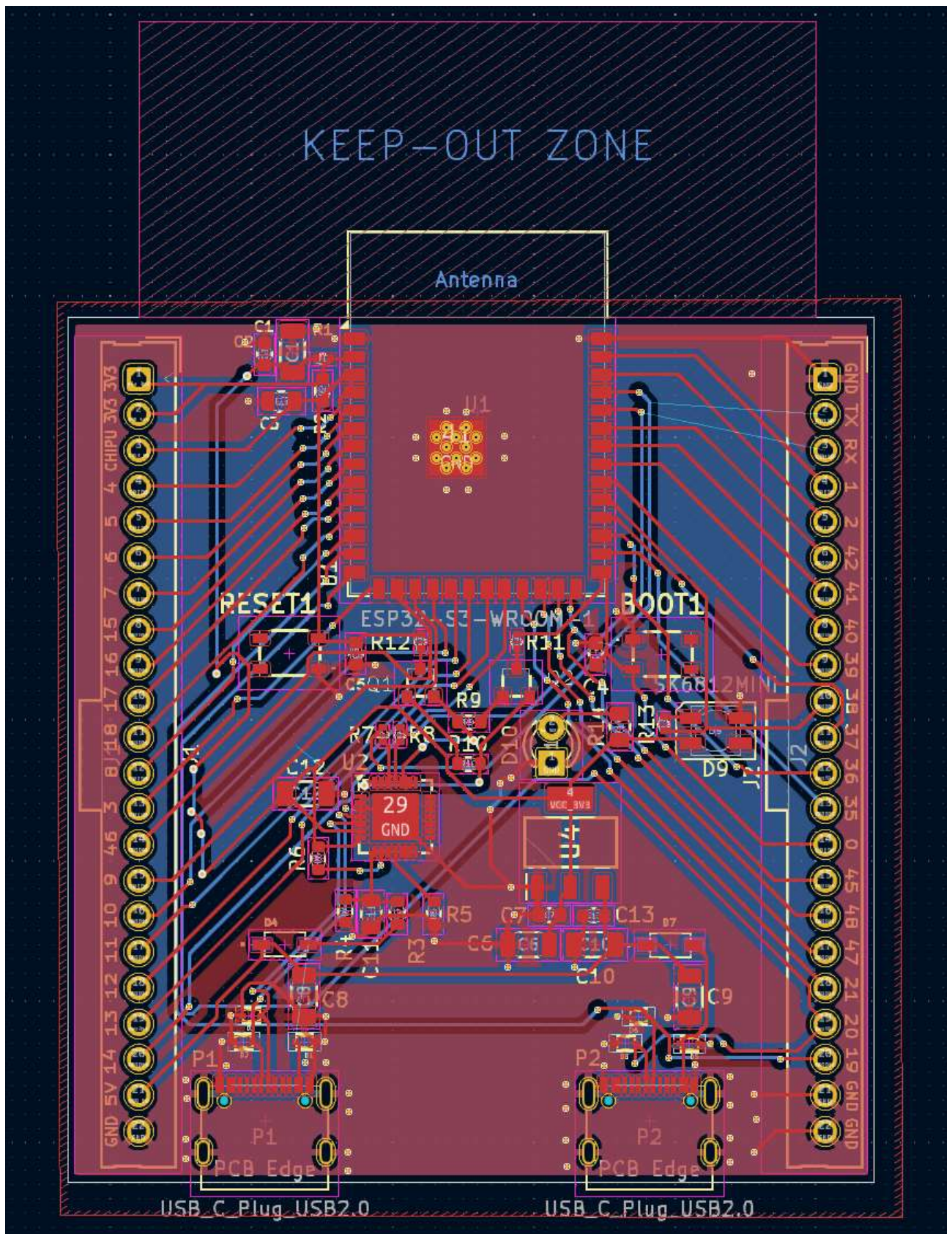


Рисунок 3.12 – Схема друкованої плати

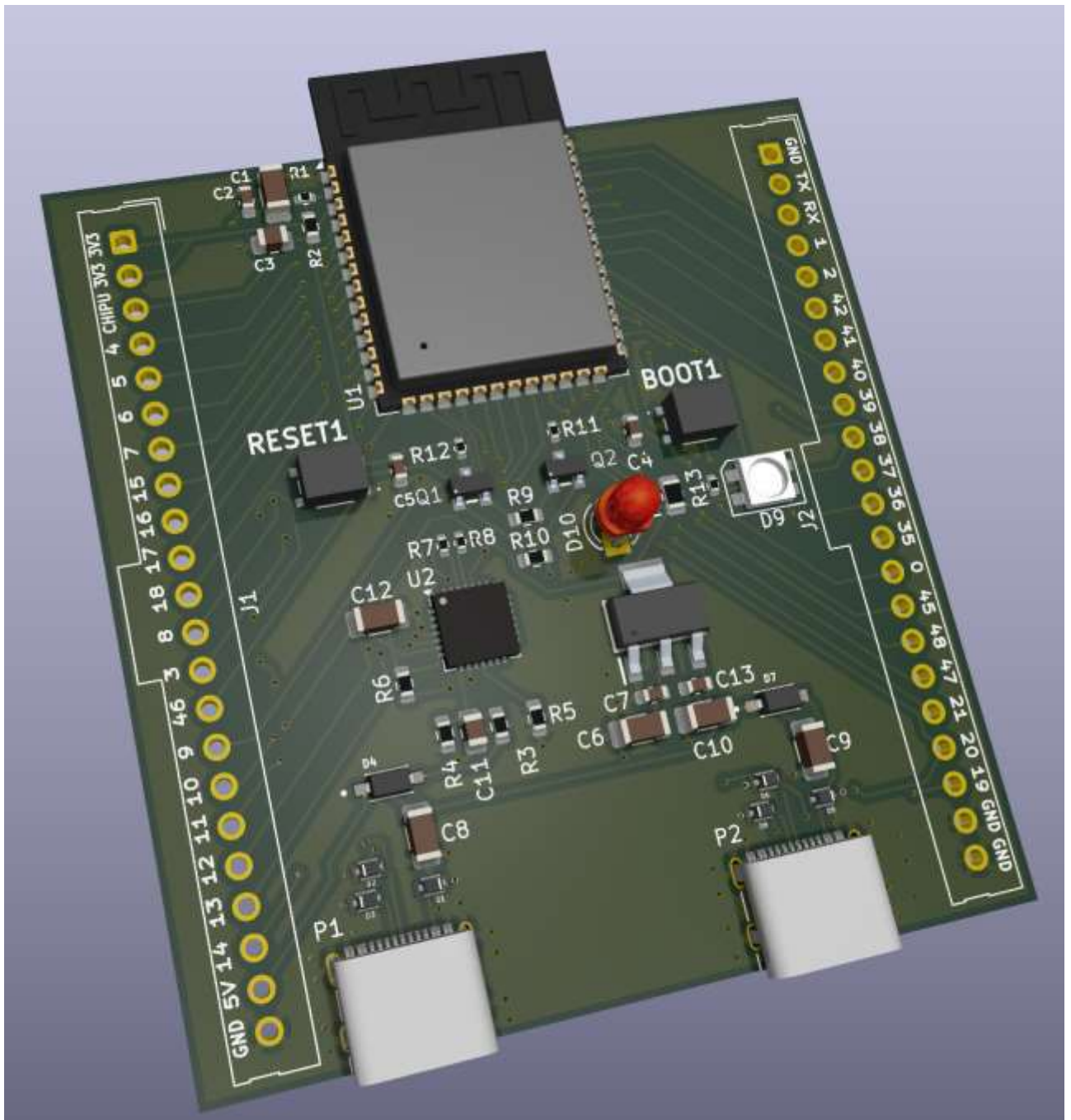


Рисунок 3.13 – Плата в тривимірному просторі

Аналогічно до раніше застосованих методів, виконується розводка друкованої плати для зовнішнього пристрою – модуля I2S мікрофона INMP441ACEZ-R7. Це дозволяє забезпечити коректне з'єднання усіх необхідних ліній та дотримання вимог до трасування сигналів (рис 3.14, 3.15).

В додатку Б наведені креслення обох плат модулів мікропроцесора ESP32-S3-WROOM-1 та мікрофона INMP441ACEZ-R7.

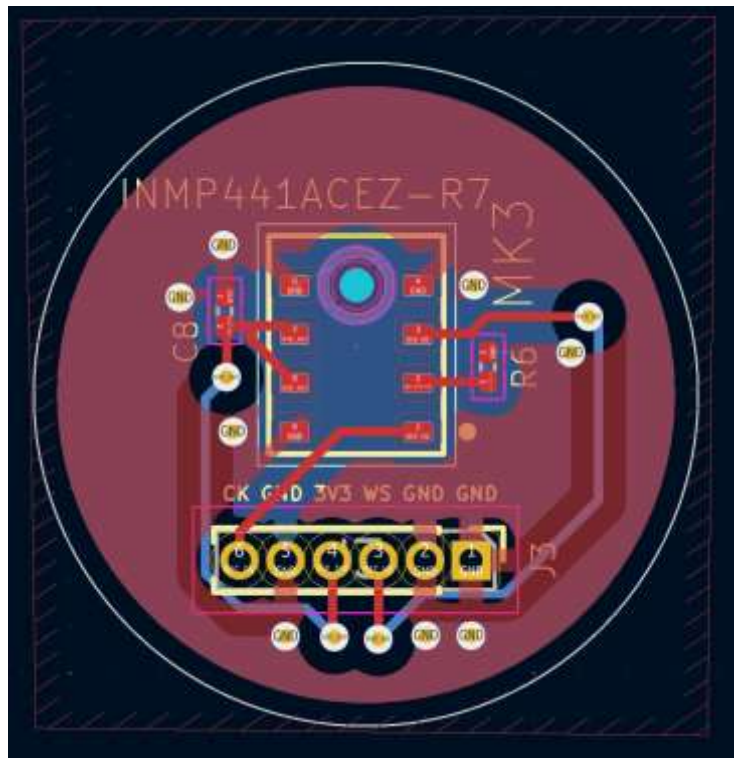


Рисунок 3.14 – Схема друкованої плати модуля I2S мікрофона
INMP441ACEZ-R7

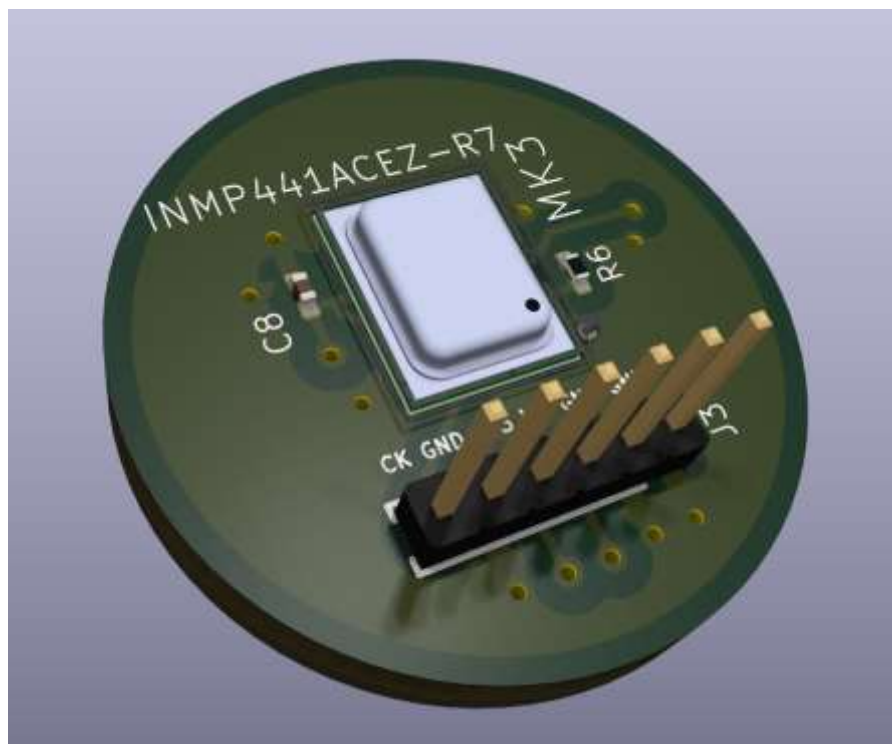


Рисунок 3.15 – Плата модуля в тривимірному просторі

4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ГОЛОСОВОГО КОНТРОЛЮ

В якості програмного каркасу чи framework-у використано ESP-Skainet від компанії Espressif Systems. ESP-Skainet – це голосовий фреймворк розроблений спеціально для мікроконтролерів серії ESP32. Його мета забезпечити ефективне розпізнавання голосових команд у реальному часі у вбудованих системах з низьким енергоспоживанням та обмеженими ресурсами такими як відсутність постійного доступу до інтернету.

Однією з ключових функцій ESP-Skainet є визначення ключового слова активації так зване wake word detection, яке дозволяє запускати голосове керування за допомогою фраз на кшталт “Hi ESP” або “Hi Lexin”. Після активації в роботу вступає модуль розпізнавання команд, MultiNet, що здатна розпізнавати заздалегідь визначений набір голосових інструкцій, наприклад: "увімкни світло", "пауза", "гучніше" тощо. Усі ці дії виконуються офлайн, без потреби в хмарних сервісах чи інтернет-з'єднанні.

Фреймворк оптимізований для Audio Front-End для обробки звуку, забезпечують хорошу продуктивність навіть у пристроях з обмеженою пам'яттю та обчислювальними можливостями. AFE включає функції VAD (визначення наявності голосу), фільтрацію шуму та луна-придушення [11].

Робота ESP-Skainet виглядає наступним чином: користувач подає голосовий сигнал через мікрофон, WakeNet аналізує його на наявність активаційної фрази. Якщо така фраза присутня, активується MultiNet, що визначає конкретну команду. Далі виконується відповідна дія, закладена в логіку пристрою (рис 4.1).

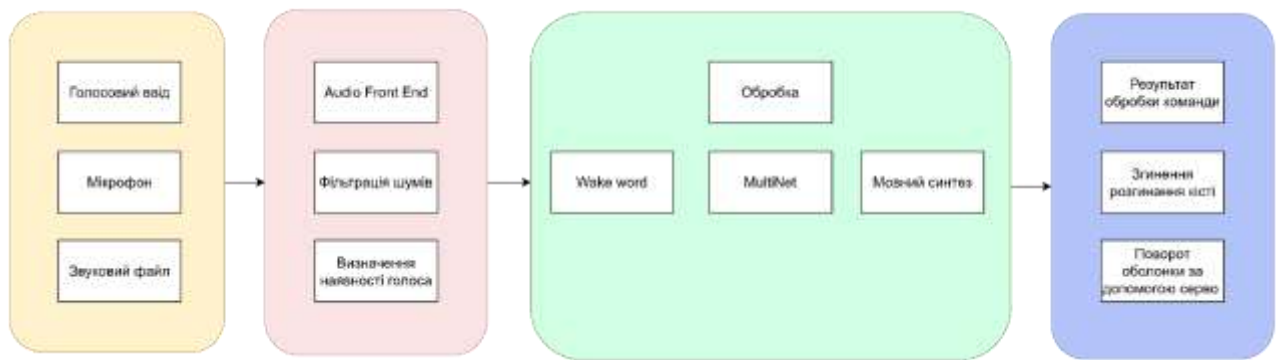


Рисунок 4.1 – Огляд функціоналу ESP-Skainet

Розробка інтелектуальної системи керування починається з завантаження git-у та клонування відповідної бібліотеки через термінал Windows (рис 4.2) чи Windows PowerShell [10].

```

Command Prompt
Microsoft Windows [Version 10.0.19045.5854]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ruter>git clone --branch ESP32-S3-Devkit-C --recursive git@github.com:0015/esp-skainet.git
fatal: destination path 'esp-skainet' already exists and is not an empty directory.

C:\Users\ruter>_

```

Рисунок 4.2 – Заздалегідь завантажена гілка проекту

Клонувавши бібліотеку встановлюємо ESP-IDF в розширення для Visual Studio Code та перевіряємо працездатність через термінал (рис 4.3).

```

ESP-IDF 5.4 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-ad5d4ee340c39bc556afa783e091e773
Setting PYTHONHOMEUSERSITE, was not set
Using Python in C:\Espressif\python_env\idf5.4_py3.11_env\Scripts\
Python 3.11.2
Using Git in C:\Espressif\tools\idf-git\2.44.0\cmd\
git version 2.44.0.windows.1
Activating ESP-IDF 5.4
Setting IDF_PATH to 'C:\Espressif\frameworks\esp-idf-v5.4.1'.
* Checking python version ... 3.11.2
* Checking python dependencies ... OK
* Deactivating the current ESP-IDF environment (if any) ... OK
* Establishing a new ESP-IDF environment ... OK
* Identifying shell ... cmd.exe
* Detecting outdated tools in system ... Found tools that are not used by active ESP-IDF version.
For removing old versions of idf-driver, idf-python-wheels use command 'python.exe C:\Espressif\frameworks\esp-idf-v5.4.1\tools\idf_tools.py uninstall'
To free up even more space, remove installation packages of those tools.
Use option python.exe C:\Espressif\frameworks\esp-idf-v5.4.1\tools\idf_tools.py uninstall --remove-archives.
Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

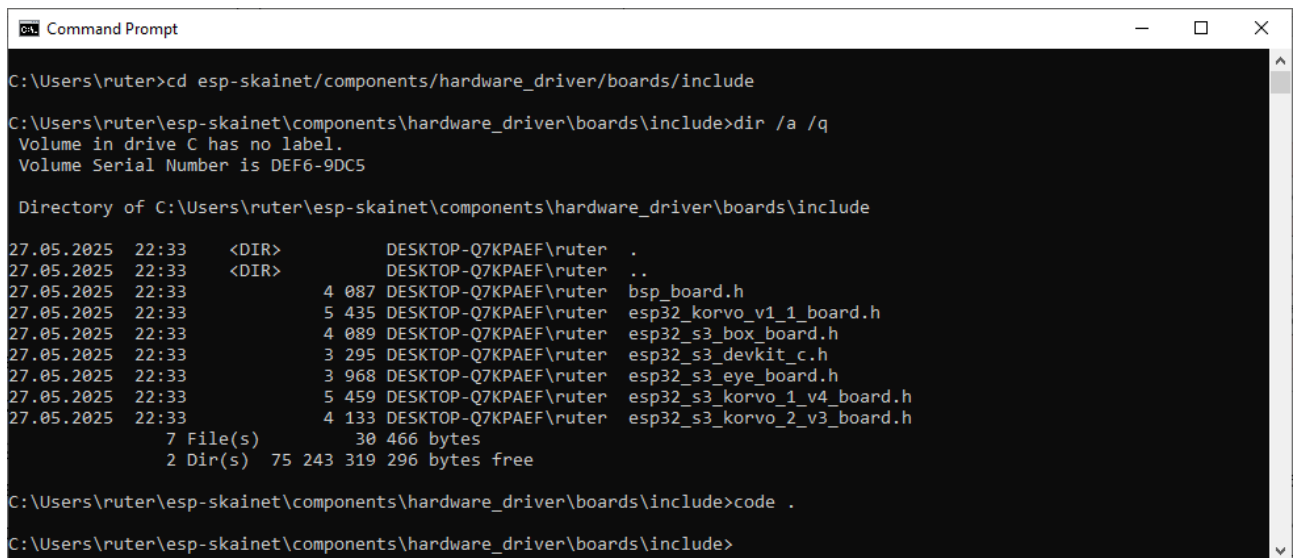
idf.py build

C:\Espressif\frameworks\esp-idf-v5.4.1>idf.py --version
ESP-IDF v5.4.1-dirty

```

Рисунок 4.3 – Перевірка роботи ESP-IDF

Команду `cd esp-skainet/components/hardware_driver/boards/include` виконуємо в терміналі для переходу в папку [10]. Командою `dir /a /q` викликаємо усі файли цієї папки включно з прихованими та системними з показанням власника кожного файлу. Командою `code .` запускаємо Visual Studio Code з терміналу поточну папку тобто VS Code відкриє її як проєкт (рис 4.4).



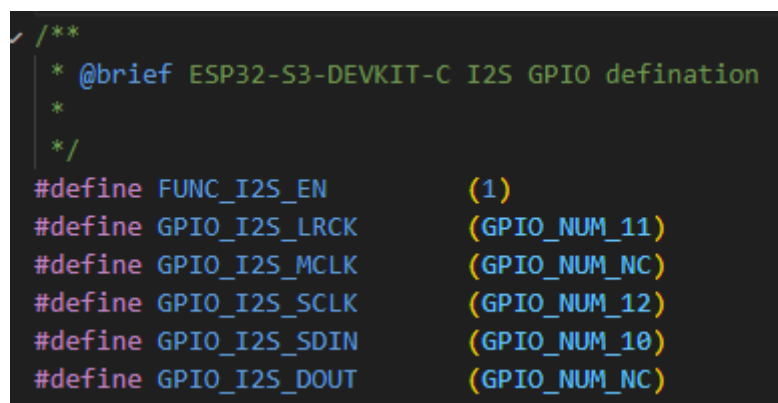
```
Command Prompt
C:\Users\ruter>cd esp-skainet/components/hardware_driver/boards/include
C:\Users\ruter\esp-skainet\components\hardware_driver\boards\include>dir /a /q
Volume in drive C has no label.
Volume Serial Number is DEF6-9DC5

Directory of C:\Users\ruter\esp-skainet\components\hardware_driver\boards\include
27.05.2025  22:33    <DIR>          DESKTOP-Q7KPAEF\ruter  .
27.05.2025  22:33    <DIR>          DESKTOP-Q7KPAEF\ruter  ..
27.05.2025  22:33         4 087 DESKTOP-Q7KPAEF\ruter  bsp_board.h
27.05.2025  22:33         5 435 DESKTOP-Q7KPAEF\ruter  esp32_korvo_v1_1_board.h
27.05.2025  22:33         4 089 DESKTOP-Q7KPAEF\ruter  esp32_s3_box_board.h
27.05.2025  22:33         3 295 DESKTOP-Q7KPAEF\ruter  esp32_s3_devkit_c.h
27.05.2025  22:33         3 968 DESKTOP-Q7KPAEF\ruter  esp32_s3_eye_board.h
27.05.2025  22:33         5 459 DESKTOP-Q7KPAEF\ruter  esp32_s3_korvo_1_v4_board.h
27.05.2025  22:33         4 133 DESKTOP-Q7KPAEF\ruter  esp32_s3_korvo_2_v3_board.h
                7 File(s)          30 466 bytes
                2 Dir(s)  75 243 319 296 bytes free

C:\Users\ruter\esp-skainet\components\hardware_driver\boards\include>code .
C:\Users\ruter\esp-skainet\components\hardware_driver\boards\include>
```

Рисунок 4.4 – Відкриття компонентів esp skainet

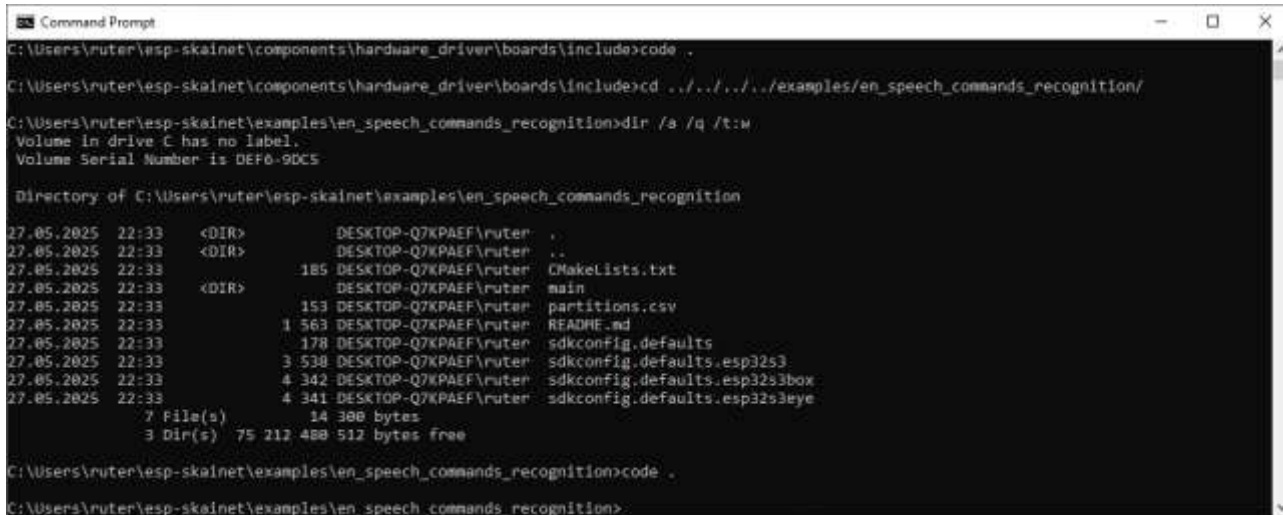
Відкриваємо наступний файл `esp32_s3_devkit_c.h` та виставляємо в параметрі `I2S GPIO defination` відповідно до схеми підключення модуля мікрофона до контактних роз'ємів плати мікропроцесора та включаємо саму функцію `I2S` (рис 4.5)



```
/**
 * @brief ESP32-S3-DEVKIT-C I2S GPIO defination
 *
 */
#define FUNC_I2S_EN          (1)
#define GPIO_I2S_LRCK       (GPIO_NUM_11)
#define GPIO_I2S_MCLK       (GPIO_NUM_NC)
#define GPIO_I2S_SCLK       (GPIO_NUM_12)
#define GPIO_I2S_SDIN       (GPIO_NUM_10)
#define GPIO_I2S_DOUT       (GPIO_NUM_NC)
```

Рисунок 4.5 – Підключення контактних роз'ємів

Наступним кроком відкриваємо у терміналі папку `cd ../../../../examples/en_speech_commands_recognition/`, яка відповідає приклади розпізнавання мовних команд на англійській мові. Командою `dir /a /q` викликаємо усі файли цієї папки та командою `code .` запускаємо проект (рис 4.6).



```
Command Prompt
C:\Users\ruter\esp-skainet\components\hardware_driver\boards\include>code .
C:\Users\ruter\esp-skainet\components\hardware_driver\boards\include>cd ../../../../examples/en_speech_commands_recognition/
C:\Users\ruter\esp-skainet\examples\en_speech_commands_recognition>dir /a /q /t:m
Volume in drive C has no label.
Volume Serial Number is DEF0-9DC5

Directory of C:\Users\ruter\esp-skainet\examples\en_speech_commands_recognition
27.05.2025 22:33 <DIR>          DESKTOP-Q7KPAEF\ruter  .
27.05.2025 22:33 <DIR>          DESKTOP-Q7KPAEF\ruter  ..
27.05.2025 22:33 <DIR>          185 DESKTOP-Q7KPAEF\ruter  CMakeLists.txt
27.05.2025 22:33 <DIR>          153 DESKTOP-Q7KPAEF\ruter  main
27.05.2025 22:33          153 DESKTOP-Q7KPAEF\ruter  partitions.csv
27.05.2025 22:33          1 563 DESKTOP-Q7KPAEF\ruter  README.md
27.05.2025 22:33          178 DESKTOP-Q7KPAEF\ruter  sdkconfig.defaults
27.05.2025 22:33          3 538 DESKTOP-Q7KPAEF\ruter  sdkconfig.defaults.esp32s3
27.05.2025 22:33          4 342 DESKTOP-Q7KPAEF\ruter  sdkconfig.defaults.esp32s3box
27.05.2025 22:33          4 341 DESKTOP-Q7KPAEF\ruter  sdkconfig.defaults.esp32s3eye
                7 File(s)          14 388 bytes
                3 Dir(s)  75 212 488 512 bytes free

C:\Users\ruter\esp-skainet\examples\en_speech_commands_recognition>code .
C:\Users\ruter\esp-skainet\examples\en_speech_commands_recognition>
```

Рисунок 4.6 – Приклади розпізнавання мовних команд

Відкривши файл `main.c` виставляємо наш цільовий пристрій `esp32s3` та обираємо метод завантаження прошивки. Переходимо до `sdkconfig` обираючи в `Audio Media HAL` плату аудіообладнання, тобто `ESP32 S3 DevKit C` (рис 4.7).

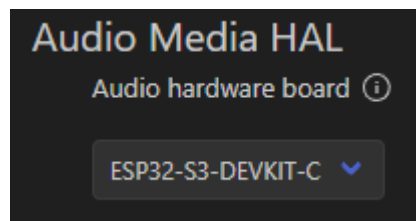


Рисунок 4.7 – Вибір плати аудіообладнання

Якщо перейти до вкладки `Add English speech commands` в `sdkconfig` можна змінити стандарте слово для пробудження. При цьому `ESP32 S3 DevKit C` працює на базі `WakeNet version 9`, яка може підтримувати тільки два слова для пробудження(рис 4.8).

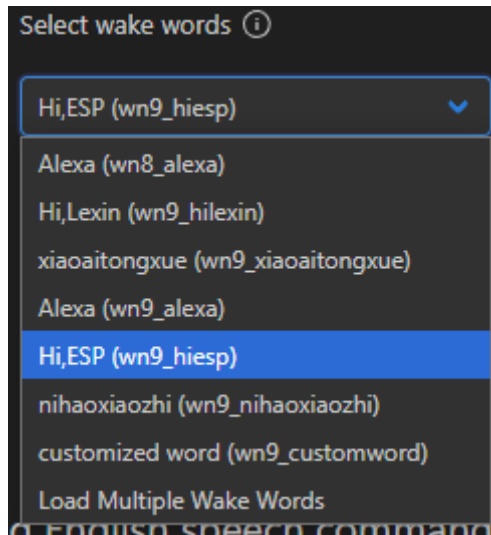


Рисунок 4.8 – Слово для пробудження

Якщо спуститися нижче будуть поточні зареєстровані аудіокоманди в обсязі 31 штуки, з можливим доданням до 199 команд з існуючими, проте вони представлені у вигляді спеціальних фонетичних символів, як наприклад команди1 «TELL ME A JOKE», 2 «SING A SONG», 3 «PLAY NEWS CHANNEL», та 4 «TURN ON MY SOUNDBOX» (рис 4.9).

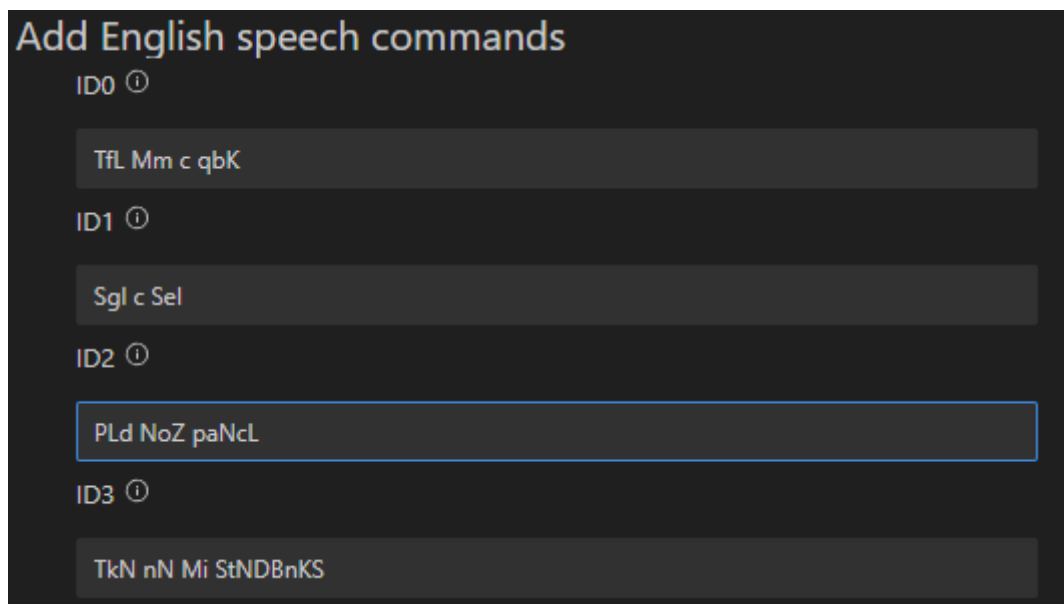


Рисунок 4.9 – Додавання голосових команд

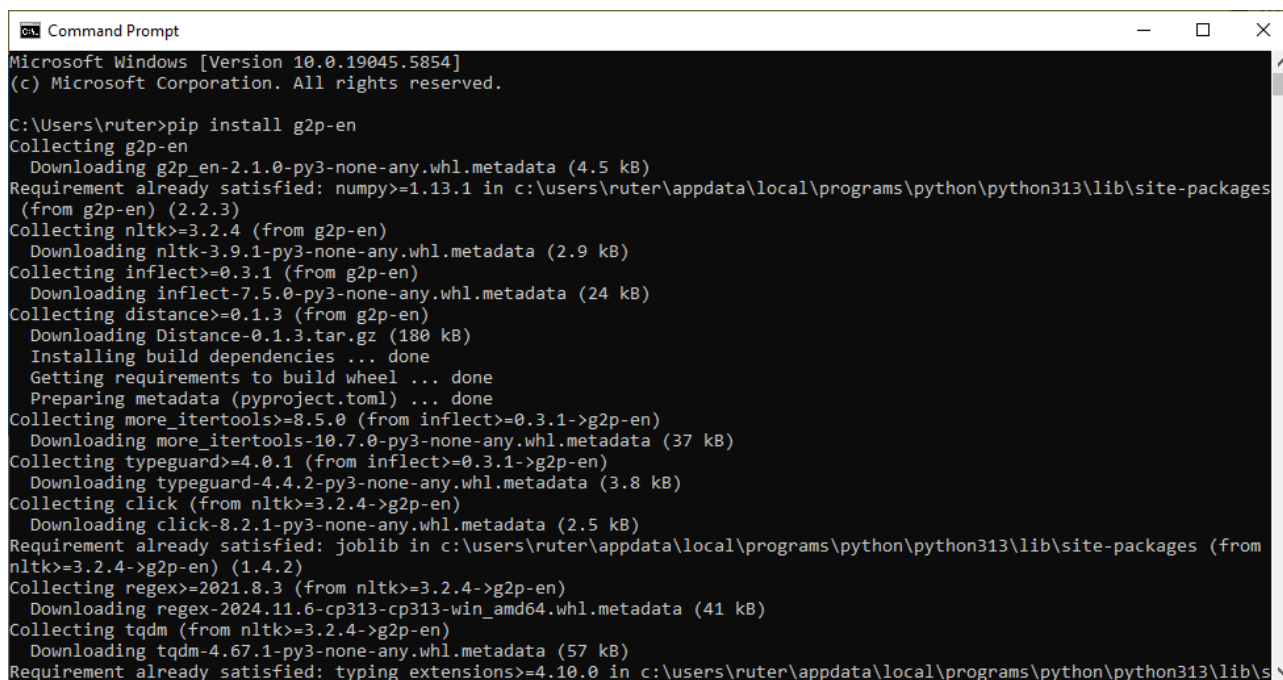
Перейшовши в main.c бачимо декілька основних функції як void feed_Task та detect_Task. Де в Feed task працює Audio Front-End для створення

відфільтрованої аудіо інформації, WakeNet входить в комплект AFE та повертає результати виявлення через інтерфейс вибірки. В Detect task коли виявляється WakeWord, попередньо зареєстровані голосові команди шукаються через Multinet.

Multinet формує результат на основі ймовірності. Коли вводяться подібні голосові дані, система відображає ймовірність найбільш відповідних команд. Чим вища ця ймовірність, тим точніше введені користувачем дані збігаються з розрахунками Multinet.

Щоб додати свої команди потрібно перетворити звичайні слова на фонетичні символи. Зробимо це за допомогою спеціального скрипта наданого Espressif Systems [12].

В термінал вводимо наступну команду котра завантажить `pip install g2p_en` цей скрипт, який являє собою простий модуль Python для перетворення англійської графеми на фонему (рис 4.10).



```
Microsoft Windows [Version 10.0.19045.5854]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ruter>pip install g2p-en
Collecting g2p-en
  Downloading g2p_en-2.1.0-py3-none-any.whl.metadata (4.5 kB)
Requirement already satisfied: numpy>=1.13.1 in c:\users\ruter\appdata\local\programs\python\python313\lib\site-packages
(from g2p-en) (2.2.3)
Collecting nltk>=3.2.4 (from g2p-en)
  Downloading nltk-3.9.1-py3-none-any.whl.metadata (2.9 kB)
Collecting inflect>=0.3.1 (from g2p-en)
  Downloading inflect-7.5.0-py3-none-any.whl.metadata (24 kB)
Collecting distance>=0.1.3 (from g2p-en)
  Downloading Distance-0.1.3.tar.gz (180 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting more_itertools>=8.5.0 (from inflect>=0.3.1->g2p-en)
  Downloading more_itertools-10.7.0-py3-none-any.whl.metadata (37 kB)
Collecting typeguard>=4.0.1 (from inflect>=0.3.1->g2p-en)
  Downloading typeguard-4.4.2-py3-none-any.whl.metadata (3.8 kB)
Collecting click (from nltk>=3.2.4->g2p-en)
  Downloading click-8.2.1-py3-none-any.whl.metadata (2.5 kB)
Requirement already satisfied: joblib in c:\users\ruter\appdata\local\programs\python\python313\lib\site-packages (from
nltk>=3.2.4->g2p-en) (1.4.2)
Collecting regex>=2021.8.3 (from nltk>=3.2.4->g2p-en)
  Downloading regex-2024.11.6-cp313-cp313-win_amd64.whl.metadata (41 kB)
Collecting tqdm (from nltk>=3.2.4->g2p-en)
  Downloading tqdm-4.67.1-py3-none-any.whl.metadata (57 kB)
Requirement already satisfied: typing_extensions>=4.10.0 in c:\users\ruter\appdata\local\programs\python\python313\lib\s
```

Рисунок 4.10 – Завантаження g2p-en

Multinet_g2p знаходиться у папці `esp-sr/tool` переходимо туди через термінал. Командою `dir /a /q` викликаємо усі файли цієї папки (рис 4.11).

```
Command Prompt
Directory of C:\Users\ruter\esp-skainet\components\esp-sr\tool
27.05.2025 22:34 <DIR>          DESKTOP-Q7KPAEF\ruter  .
27.05.2025 22:34 <DIR>          DESKTOP-Q7KPAEF\ruter  ..
27.05.2025 22:34 <DIR>          DESKTOP-Q7KPAEF\ruter  fst
27.05.2025 22:34                2 124 DESKTOP-Q7KPAEF\ruter  multinet_g2p.py
27.05.2025 22:34                1 537 DESKTOP-Q7KPAEF\ruter  multinet_pinyin.py
27.05.2025 22:34                1 806 DESKTOP-Q7KPAEF\ruter  README.md
27.05.2025 22:34                31 DESKTOP-Q7KPAEF\ruter  requirements
                4 File(s)          5 498 bytes
                3 Dir(s)  74 845 347 840 bytes free

C:\Users\ruter\esp-skainet\components\esp-sr\tool>
```

Рисунок 4.11 – Скрипт Multinet_g2p

Цей скрипт вимагає pandas та numpy, завантажуюмо їх також наступною командою: `python -m pip install pandas` та `python -m pip install numpy` (рис 4.12 та 4.13).

```
Command Prompt - python -m pip install pandas
                4 File(s)          5 498 bytes
                3 Dir(s)  74 845 347 840 bytes free

C:\Users\ruter\esp-skainet\components\esp-sr\tool>python -m pip install pandas
Collecting pandas
  Downloading pandas-2.2.3-cp313-cp313-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: numpy>=1.26.0 in c:\users\ruter\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2.2.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\ruter\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in c:\users\ruter\appdata\local\programs\python\python313\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Downloading pandas-2.2.3-cp313-cp313-win_amd64.whl (11.5 MB)
----- 11.5/11.5 MB 15.9 MB/s eta 0:00:00
Downloading pytz-2025.2-py2.py3-none-any.whl (509 kB)
Downloading tzdata-2025.2-py2.py3-none-any.whl (347 kB)
Installing collected packages: pytz, tzdata, pandas
```

Рисунок 4.12 – Завантаження бібліотеки pandas

```
Command Prompt
C:\Users\ruter\esp-skainet\components\esp-sr\tool>python -m pip install numpy
Requirement already satisfied: numpy in c:\users\ruter\appdata\local\programs\python\python313\lib\site-packages (2.2.3)

[notice] A new release of pip is available: 24.3.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\ruter\esp-skainet\components\esp-sr\tool>
```

Рисунок 4.13 – Завантаження бібліотеки numpy

Тепер використовуючи скрипт `Multinet_g2p` перетворимо слова на фонетичні символи[12], наступною командою `python multinet_g2p.py -t "hello`

world" та отримуємо результат "hcLb WkLD" який вставляємо у sdkconfig до вкладки Add English speech commands (рис 4.15). Також додаємо 4 наступні команди "take it", "release it", "move upward", "move downward" (рис 4.14).

```
C:\Users\ruter\esp-skainet\components\esp-sr\tool>
C:\Users\ruter\esp-skainet\components\esp-sr\tool>python multinet_g2p.py -t "hello world"
in: hello world
out: hcLb WkLD;

C:\Users\ruter\esp-skainet\components\esp-sr\tool>python multinet_g2p.py -t "take it"
in: take it
out: TdK gT;

C:\Users\ruter\esp-skainet\components\esp-sr\tool>python multinet_g2p.py -t "release it"
in: release it
out: RmLmS gT;

C:\Users\ruter\esp-skainet\components\esp-sr\tool>python multinet_g2p.py -t "move upward"
in: move upward
out: MoV cPWkD;

C:\Users\ruter\esp-skainet\components\esp-sr\tool>python multinet_g2p.py -t "move downward"
in: move downward
out: MoV DtNWkD;
```

Рисунок 4.14 – Фонетична транскрипція

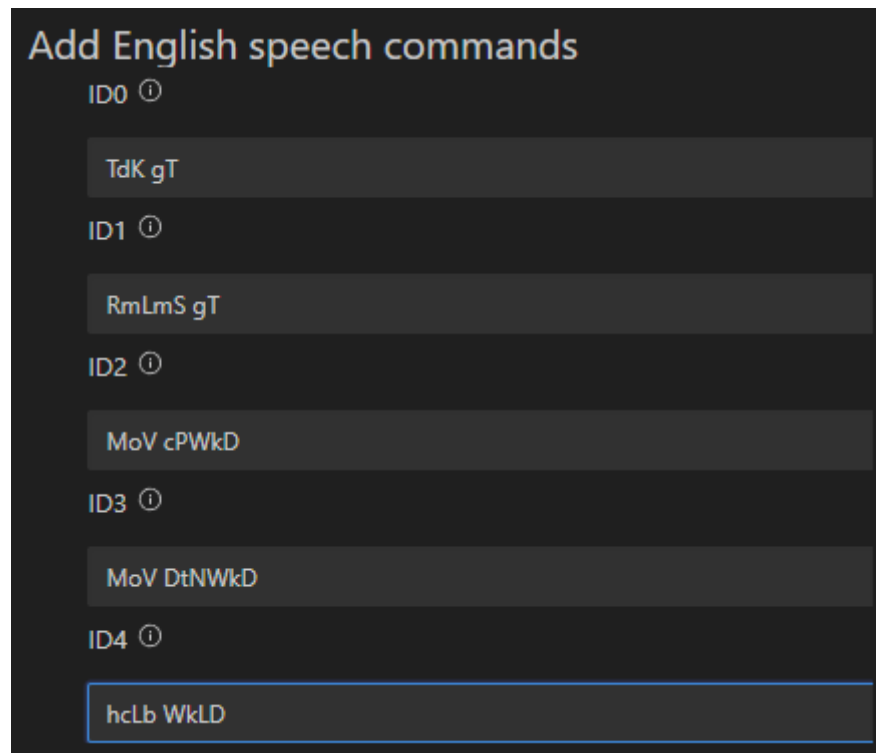


Рисунок 4.15 – Фонетична заміна

Переходимо в main.c та створюємо в умові mn_state == ESP_MN_STATE_DETECTED команду розпізнавання та виконання дії. Де if (mn_result->num > 0) є умовою, яка перевіряє, чи взагалі розпізнано якісь голосові команди. Оператор switch використовується для вибору дії на основі ID першої найбільш ймовірної розпізнаної команди. mn_result->command_id[0] повертає числовий ідентифікатор цієї команди. Кожен case відповідає певному ID команди (рис 4.16).

```
if (mn_result->num > 0){
    switch (mn_result->command_id[0])
    {
        case 0:
            detect_flag = 2;
            set_led_color(15,0,0);
            break;
        case 1:
            detect_flag = 2;
            set_led_color(15,15,0);
            break;
        case 2:
            detect_flag = 2;
            set_led_color(15,0,15);
            break;
        case 3:
            detect_flag = 2;
            set_led_color(0,15,15);
            break;
        case 4:
            detect_flag = 2;
            set_led_color(0,0,15);
            break;
    }
}
```

Рисунок 4.16 – Виконання дії відповідно команди

Підключаємо серводвигуни виставляючи в параметрах GPIO defination відповідно до контактних роз'ємів плати мікропроцесора. Також задаємо частоту ШИМ, максимальний кут повороту серводвигуна, початковий кут другого серводвигуна та мінімальну максимальну ширину імпульсу (4.17).

```
#define SERVO1_GPIO_PIN 13
#define SERVO2_GPIO_PIN 14
#define SERVO_FREQ_HZ 50
#define SERVO_MIN_PULSE_WIDTH_US 500
#define SERVO_MAX_PULSE_WIDTH_US 2500
#define SERVO_MAX_DEGREE 180
#define SERVO2_INITIAL_DEGREE 0
```

Рисунок 4.17 – Команда для препроцесора, яка визначає макровизначення

`servo1_set_angle` та `servo2_set_angle`, ці допоміжні функції створені для простого керування кутами серводвигунів. Вони приймають кут у градусах і перетворюють його на відповідну ширину імпульсу, а потім встановлюють це значення для компаратора MCPWM (рис 4.18).

```
static mcpwm_cpr_handle_t comparator1 = NULL;
static mcpwm_cpr_handle_t comparator2 = NULL;

static inline uint32_t example_angle_to_compare(int degree) {
    return (degree - 0) * (SERVO_MAX_PULSE_WIDTH_US - SERVO_MIN_PULSE_WIDTH_US) / (SERVO_MAX_DEGREE - 0) + SERVO_MIN_PULSE_WIDTH_US;
}

void servo1_set_angle(int degree) {
    if (comparator1 != NULL) {
        ESP_ERROR_CHECK(mcpwm_comparator_set_compare_value(comparator1, example_angle_to_compare(degree)));
    }
}

void servo2_set_angle(int degree) {
    if (comparator2 != NULL) {
        ESP_ERROR_CHECK(mcpwm_comparator_set_compare_value(comparator2, example_angle_to_compare(degree)));
    }
}
```

Рисунок 4.18 – Допоміжні функції `servo1_set_angle` та `servo2_set_angle`

У case 0 і case 1 додано цикл `for`, який плавно переміщає серводвигун з кроком 5 градусів і невеликою затримкою `vTaskDelay` для більш природного руху. У case 2 і case 3 просто встановлюється початковий кут другого сервопривода.

```

if (mn_result->num > 0){
    switch (mn_result->command_id[0])
    {
        case 0:
            detect_flag = 2;
            set_led_color(15,0,0);
            for (int angle = 0; angle <= 135; angle += 5) {
                servo1_set_angle(angle);
                vTaskDelay(pdMS_TO_TICKS(20));
            }
            break;
        case 1:
            detect_flag = 2;
            set_led_color(15,15,0);
            for (int angle = 135; angle >= 0; angle -= 5) {
                servo1_set_angle(angle);
                vTaskDelay(pdMS_TO_TICKS(20));
            }
            break;
        case 2:
            detect_flag = 2;
            set_led_color(15,0,15);
            servo2_set_angle(SERVO2_INITIAL_DEGREE + 25);
            break;
        case 3:
            detect_flag = 2;
            set_led_color(0,15,15);
            servo2_set_angle(SERVO2_INITIAL_DEGREE - 25);
            break;
        case 4:
            detect_flag = 2;
            set_led_color(0,0,15);
            break;
    }
}

```

Рисунок 4.19 – Виконання дії відповідно голосової команди

За відсутності самої плати Esp32 S3 Devkit C перевірити роботу кода можна зробити збірку проекту, яка скомпілює вихідний код, зв'яже його та готує бінарний файл (.bin), який пізніше можна прошити на плату мікропроцесора (рис 4.20).

```

-- Configuring done (63.2s)
-- Generating done (0.5s)
-- Build files have been written to: C:/Users/ruter/esp-skainet/examples/en_speech_commands_recognition/build/bootloader

```

Рисунок 4.20 – Збірка проекту

ВИСНОВКИ

В ході виконання атестаційної роботи було розглянуто актуальність застосування голосового інтерфейсу для медичних пристроїв, медико-технічні особливості протезування, як це впливає на сучасні технології протезування верхніх кінцівок та які не стандартні можливості застосування 3D-друку в протезуванні.

Розроблено функціонал протезу з його функціональною схемою, обрано елементну базу з обґрунтуванням вибору відповідних компонентів за різними параметричними даними, такими як: струм споживання, напруга (пряма та непряма), робочі температури, фізичні розміри. Розраховано загальні параметри, такі як: максимальний та мінімальний струм споживання схеми, час безперервної роботи відповідно ємності акумулятора та максимальна потужність схеми.

Побудовано з вибраної елементної бази еквівалентну схему модулю ESP32 S3 Devkitc 1 та модуля мікрофона INMP441, зроблено схему загального підключення контактних роз'ємів серводвигунів.

Розроблено схему друкованої плати відповідно правил компоновання елементів, електромагнітної сумісності та урахуванням правил дизайну (DRC), з використанням різної ширини доріжок згідно з струмом сигнальних ліній та ліній електропередач.

Реалізовано програмне забезпечення інтелектуальної системи керування на базі ESP-Skainet, налаштовано та створено її проект. Задано вибрані контактні роз'єми плати мікропроцесора з зовнішньою апаратурою. За допомогою скрипта Multinet_g2p додано в конфігурацію проекту власні голосові команди та програмно створено їм дію, зібравши проект.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кузнецов І. В., Бабиченко О.Ю. Протез верхньої кінцівки керований голосом на базі мікроконтролера ESP32. Матеріали 29-го Міжнародного молодіжного форуму «Радіоелектроніка та молодь у ХХІ столітті». Зб. матеріалів форуму. Т. 1. – Харків: ХНУРЕ. 2025. – С. 96–98.
2. Манічева Н., Чеботарьова Г., Рибченко К. // Аналітичний огляд розвитку реабілітаційної інженерії та високотехнологічного протезування. // Modern technologies of biomedical engineering. // Матеріали II міжнародної науково-технічної конференції, 17–19 травня 2023 року. – 5 с.
3. Протезування та штучні органи: Конспект лекцій [Електронний ресурс] : навч. посіб. для студ. спеціальності 163 «Біомедична інженерія» / І. Ю. Худецький, Ю. В. Антонова-Рафі, Г. В. Мельник, Є. В. Сніцар ; КПІ ім. Ігоря Сікорського. Електронні текстові дані. Київ : КПІ ім. Ігоря Сікорського, 2021. – 184 с.
4. B. Groenen., Guide to basic PCB design using KiCad., March 2021 – 11p.
5. David L. Jones., PCB Design Tutorial., Freely distributable for educational and personal use., Revision A – 29.06.2004 – 25 p.
6. IPC-2221A., Generic Standard on Printed Board Design., A standard developed by IPC., May 2003 – 124 p.
7. URL: https://dl.espressif.com/dl/schematics/SCH_ESP32-S3-DevKitC-1_V1.1_20220413.pdf [Електронний ресурс. Режим доступу 08.05.2025]
8. URL: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32s3/esp32-s3-devkitc-1/user_guide.html#related-documents [Електронний ресурс. Режим доступу 11.05.2025]

9. URL: <https://docs.espressif.com/projects/esp-hardware-design-guidelines/en/latest/esp32/schematic-checklist.html> [Электронный ресурс. Режим доступа 07.05.2025]
10. URL: <https://github.com/0015/esp-skainet/tree/ESP32-S3-Devkit-C> [Электронный ресурс. Режим доступа 20.05.2025]
11. URL: <https://github.com/0015/esp-skainet?tab=readme-ov-file> [Электронный ресурс. Режим доступа 19.05.2025]
12. URL: <https://github.com/espressif/esp-sr/tree/master/tool> [Электронный ресурс. Режим доступа 20.05.2025]
13. URL: <https://www.3dprinter.ua/osnovni-materiali-shho-vikoristovuyutsya-dlya-3d-druku/> [Электронный ресурс. Режим доступа 25.04.2025]
14. URL: <https://www.alldatasheet.com/datasheet-pdf/view/1643962/ETC/SK6812MINI-012.html> [Электронный ресурс. Режим доступа 13.05.2025]
15. URL: <https://www.alldatasheet.com/datasheet-pdf/view/178400/DIODES/1N5819HW-7-F.html> [Электронный ресурс. Режим доступа 15.05.2025]
16. URL: <https://www.alldatasheet.com/datasheet-pdf/view/1994926/SGMICRO/SGM2212-3.3.html> [Электронный ресурс. Режим доступа 26.04.2025]
17. URL: <https://www.alldatasheet.com/datasheet-pdf/view/230967/EVERLIGHT/264-10SDRDSLASHS530-A3.html> [Электронный ресурс. Режим доступа 15.05.2025]
18. URL: https://www.alldatasheet.com/datasheet-pdf/view/695296/LRC/L8050QLT1G_15.html [Электронный ресурс. Режим доступа 12.05.2025]

19. URL: <https://www.alldatasheet.com/datasheet-pdf/view/769823/LRC/LESD5D5.0CT1G.html> Электронный ресурс. Режим доступа 15.05.2025]

20. URL: <https://www.alldatasheet.com/datasheet-pdf/view/851450/SILABS/CP2102N-A01-GQFN20.html> Электронный ресурс. Режим доступа 16.05.2025]

21. URL: https://www.nh-technology.de/T-Mec/PDF/Series_TC/NTC013.pdf Электронный ресурс. Режим доступа 14.05.2025]

22. URL: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf [Электронный ресурс. Режим доступа 11.05.2025]

23. URL: <https://www.snapeda.com/parts/LESD5D5.0CT1G/Leshan%20Radio/view-part/> Электронный ресурс. Режим доступа 25.04.2025]