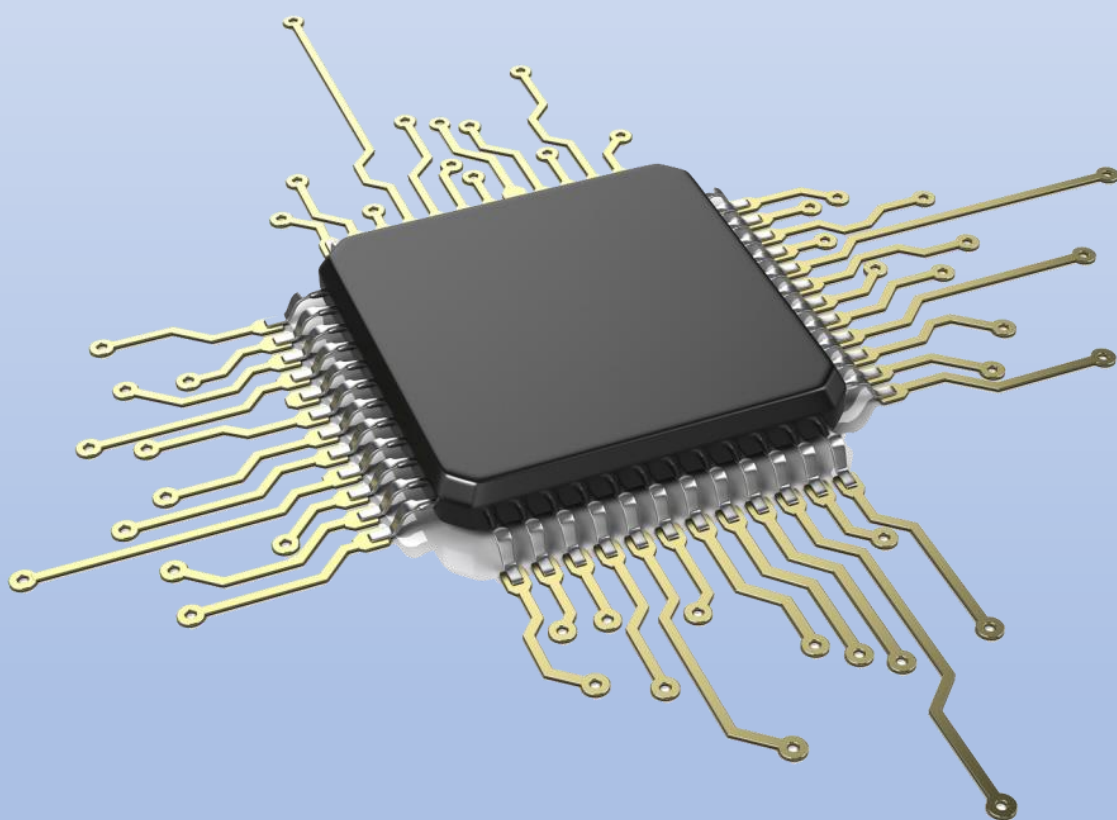


Р.М. Галаган

**Комп'ютерне
проєктування
електронних схем**



Комп'ютерний практикум

2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Р. М. Галаган

КОМП'ЮТЕРНЕ ПРОЄКТУВАННЯ ЕЛЕКТРОННИХ СХЕМ КОМП'ЮТЕРНИЙ ПРАКТИКУМ

Навчальний посібник

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою
«Комп'ютерно-інтегровані системи та технології в приладобудуванні»
спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» та
174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

Електронне мережне навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2023

Рецензенти: *Чмельов В.О.*, к.т.н., доцент кафедри РТС, КПІ ім. Ігоря Сікорського
Павловський О.М., к.т.н., доцент кафедри КІОНС, КПІ ім. Ігоря Сікорського

Відповідальний
редактор: *Лисенко Ю.Ю.*, к.т.н., доцент кафедри АСНК, КПІ ім. Ігоря Сікорського

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № 6 від 30.03.2023 р.)
за поданням Вченої ради приладобудівного факультету
(протокол № 2/23 від 27.02.2023 р.)*

Навчальний посібник призначений для докладного вивчення системи автоматизованого проєктування Altium Designer та допомагає оволодіти навиками наскрізного проєктування принципових схем та друкованих плат електронних пристроїв. У посібнику пояснюється, як розробляти умовні графічні позначення електронних компонентів та їх посадкові місця. Описуються принципи розробки та оформлення принципових схем та створення на їх основі друкованих плат. Комп'ютерні практикуми структуровані за принципом «від простого до складного»: спочатку студенти навчаються розроблювати компонентні модулі та додавати їх до власної бібліотеки, після чого переходять до розробки принципових схем та друкованих плат електронних пристроїв.

Для здобувачів ступеня бакалавра за спеціальностями 151 «Автоматизація та комп'ютерно-інтегровані технології» та 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка». Може бути корисним для аспірантів, викладачів та фахівців у сфері приладобудування, автоматизації, неруйнівного контролю, технічної діагностики тощо, які займаються розробкою та проєктуванням електронних приладів та систем різного призначення.

Реєстр. № НП 22/23-555. Обсяг 20,7 авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Перемоги, 37, м. Київ, 03056
<https://kpi.ua>

Свідectво про внесення до Державного реєстру видавців, виготовлювачів
і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

© Р. М. Галаган, 2023
© КПІ ім. Ігоря Сікорського, 2023

ЗМІСТ

ЗМІСТ	3
СКОРОЧЕННЯ, ПОЗНАЧЕННЯ ТА ТЕРМІНИ	5
ВСТУП	6
ОСОБЛИВОСТІ НАВЧАЛЬНОГО ПОСІБНИКА.....	8
КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО СИСТЕМУ АВТОМАТИЗОВАНОГО ПРОЄКТУВАННЯ СХЕМ ТА ДРУКОВАНИХ ПЛАТ ALTIUM DESIGNER	10
КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 1. ОСНОВИ РОБОТИ В СЕРЕДОВИЩІ ALTIUM DESIGNER	72
КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 2. РОЗРОБКА КОМПОНЕНТНИХ МОДУЛІВ ПАСИВНИХ ДИСКРЕТНИХ ЕЛЕКТРОННИХ КОМПОНЕНТІВ.....	92
КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 3. РОЗРОБКА КОМПОНЕНТНИХ МОДУЛІВ АКТИВНИХ ДИСКРЕТНИХ ЕЛЕКТРОННИХ КОМПОНЕНТІВ	120
КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 4. РОЗРОБКА КОМПОНЕНТНОГО МОДУЛЯ ЦИФРОВОЇ МІКРОСХЕМИ K155ЛА3	136
КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 5. РОЗРОБКА КОМПОНЕНТНОГО МОДУЛЯ ТАЙМЕРА 555.....	148
КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 6. РОЗРОБКА КОМПОНЕНТНИХ МОДУЛІВ ОПЕРАЦІЙНИХ ПІДСИЛЮВАЧІВ LMH6714MF ТА AD642.....	163
КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 7. РОЗРОБКА КОМПОНЕНТНИХ МОДУЛІВ З'ЄДНУВАЧІВ ТА РОЗ'ЄМІВ	173
КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 8. РОЗРОБКА КОМПОНЕНТНОГО МОДУЛЯ ДРАЙВЕРА СИЛОВИХ ТРАНЗИСТОРІВ MD1822.....	189
КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 9. РОЗРОБКА 3D-МОДЕЛЕЙ КОРПУСІВ ЕЛЕКТРОННИХ КОМПОНЕНТІВ ТА ЇХ ДОДАВАННЯ ДО БІБЛІОТЕЧНИХ КОМПОНЕНТНИХ МОДУЛІВ.....	204
КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 10. РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ ТА ДРУКОВАНОЇ ПЛАТИ ПІДСИЛЮВАЧА НА ОДНОМУ ТРАНЗИСТОРІ.....	243
КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 11. РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ ТА ДРУКОВАНОЇ ПЛАТИ ГЕНЕРАТОРА ТАКОВИХ ІМПУЛЬСІВ НА БАЗІ МІКРОСХЕМИ ТАЙМЕРУ 555.....	276
ЗАВДАННЯ №1 ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ НА КОМП'ЮТЕРНИХ ПРАКТИКУМАХ.....	319
ЗАВДАННЯ №2 ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ НА КОМП'ЮТЕРНИХ ПРАКТИКУМАХ.....	323
ЗАВДАННЯ №3 ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ НА КОМП'ЮТЕРНИХ ПРАКТИКУМАХ.....	343
ЛІТЕРАТУРА.....	378
ДОДАТОК А. РЕКОМЕНДОВАНІ ІНТЕРНЕТ-РЕСУРСИ.....	379

ДОДАТОК Б. СТАНДАРТИ З РОЗРОБКИ УМОВНИХ ГРАФІЧНИХ ПОЗНАЧЕНЬ, ЕЛЕКТРИЧНИХ СХЕМ, ПОСАДКОВИХ МІСЦЬ ЕЛЕКТРОННИХ КОМПОНЕНТІВ ТА ДРУКОВАНИХ ПЛАТ	381
ДОДАТОК В. СТВОРЕННЯ ПРАВИЛ ПРОЄКТУВАННЯ У РЕДАКТОРІ ПЛАТ ALTIUM DESIGNER	386
ДОДАТОК Г. РЕКОМЕНДАЦІЇ З ВИКОРИСТАННЯ МИШКИ ТА ГАРЯЧИХ КЛАВІШ В ALTIUM DESIGNER 17.1	387
ДОДАТОК Д. РОЗМІРИ НАЙБІЛЬШ ПОПУЛЯРНИХ ТИПІВ SMD КОРПУСІВ ПАСИВНИХ ЕЛЕКТРОННИХ КОМПОНЕНТІВ	391
ДОДАТОК Е. РЕКОМЕНДАЦІЇ ЩОДО РОЗРОБКИ КОНТАКТНИХ МАЙДАНЧИКІВ ДЛЯ SMD-РЕЗИСТОРІВ	393
ДОДАТОК Ж. РІЗНІ НЕСТАНДАРТНІ СИТУАЦІЇ, ЩО ПРОЯВЛЯЮТЬСЯ ПІД ЧАС РОБОТИ В ALTIUM DESIGNER, ТА СПОСОБИ ЇХ ПОДОЛАННЯ.....	394
ДОДАТОК К. АНАЛІЗ ТЕХНІЧНОЇ ДОКУМЕНТАЦІЇ (НА ПРИКЛАДІ МІКРОСХЕМИ AD5600).....	397
ДОДАТОК Л. ДОДАТКОВІ ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ	401

СКОРОЧЕННЯ, ПОЗНАЧЕННЯ ТА ТЕРМІНИ

- ГТІ** – генератор тактових імпульсів
ДП – друкована плата
КМ – контактний майданчик
КМОН – комплементарний метал-оксидний напівпровідник
ЛКМ – ліва клавіша миші
ОП – операційний підсилювач
ПКМ – права клавіша миші
ПЛІС – програмована логічна інтегральна схема
ПМ – посадкове місце
САПР – система автоматизованого проектування
ТТЛ – транзисторно-транзисторна логіка
ЧПУ – числове програмне управління
УГП – умовне графічне позначення
CNC – computer numerical control (числове програмне управління)
Datasheet – технічна документація
DIP – dual inline package
EDA – electronic design automation (автоматизація проектування електронних засобів)
ECAD – electronic computer-aided design (САПР електронних пристроїв)
MCAD – mechanical computer-aided design (САПР механічних пристроїв)
Pad – контактний майданчик
Pitch – відстань (крок) між контактними майданчиками
SOIC – small outline integrated circuit
SOP – small outline package
Footprint – посадкове місце
→ – використання стрілки у тексті означає послідовність дій (зазвичай застосовується у навчальному посібнику, коли необхідно показати дії із елементами меню, щоб виконати деяку команду)

ВСТУП

Навчальний посібник містить протоколи 11-ти комп'ютерних практикумів та завдання для 3-х індивідуальних самостійних робіт, які виконуються студентами під час вивчення дисципліни «**Комп'ютерне проєктування електронних схем**».

У розвитку сучасних технологій ключову роль відіграють системи автоматизованого проєктування електронних схем (англ. **electronic computer-aided design, ECAD**). Особливо це стосується приладобудівної галузі, електроніки, електротехніки, робототехніки, систем автоматизації та управління. Для розробки принципових схем та друкованих плат приладів та систем існує велика кількість САПР: Altium Designer, OrCAD, P-CAD, DipTrace, EAGLE тощо. Уміння застосовувати САПР електронних пристроїв дозволить студентам підвищити ефективність розробки схем сучасних приладів та автоматизованих систем.

Метою проведення комп'ютерних практикумів є формування умінь та набуття досвіду практичного використання САПР **Altium Designer**¹ для розробки принципових схем та друкованих плат.

Призначення комп'ютерних практикумів полягає у тому, що вони мають виявити:

- ступінь знання студентом теоретичної частини дисципліни;
- вміння користуватися літературними джерелами;
- навички використання комп'ютерів та інформаційних технологій для досягнення поставленої мети;
- навички самостійної роботи для вирішення поставлених завдань;
- навички використання теоретичних знань на практиці.

Перед тим як виконувати комп'ютерні практикуми, студент має обов'язково ознайомитись із лекціями.

У кожного студента має бути власне робоче місце. Комп'ютерні практикуми виконуються студентами під наглядом викладача. Послідовність комп'ютерних практикумів визначається навчальним посібником, однак може бути змінена викладачем за потреби. Деякі із комп'ютерних практикумів виконуються протягом декількох занять.

¹ Під час виконання комп'ютерних практикумів використовується версія програми із англomовним інтерфейсом.

Робота над кожним окремим комп'ютерним практикумом складається із трьох частин: *допуск, виконання та захист комп'ютерного практикуму*. Отримана оцінка буде залежати від реалізації усіх трьох частин.

Перш ніж перейти до виконання комп'ютерного практикуму, студент повинен продемонструвати знання протоколу та пояснити дії, які йому доведеться виконувати під час роботи над комп'ютерним практикумом, а також розуміння результату, який необхідно досягти.

Усі пункти комп'ютерного практикуму виконуються послідовно згідно протоколу, який заздалегідь видається студенту (у друкованому або електронному вигляді). Результатом виконання комп'ютерного практикуму є побудовані умовні графічні позначення електронних компонентів та їх посадкові місця, принципові схеми та топологія друкованої плати.

Після виконання комп'ютерного практикуму студент повинен його обов'язково захистити, пояснивши результати та відповівши на задані теоретичні запитання.

ОСОБЛИВОСТІ НАВЧАЛЬНОГО ПОСІБНИКА

Цей навчальний посібник у першу чергу призначений для використання в електронному форматі. Він містить гіперактивні посилання² на ресурси, що розміщені у мережі «Інтернет». Це дозволяє швидко ознайомитись із додатковою корисною інформацією, економить час студентам на пошук та не дублює вже існуючу інформацію.

Однак необхідно враховувати, що деякі посилання із часом можуть припинити працювати внаслідок різних причин. Тому, якщо посилання не працює, то ви можете спробувати самостійно пошукати необхідну інформацію, звертаючи увагу на контекст, де воно використовувалось. Наприклад, в одному з комп'ютерних практикумів є посилання на відеоролик в YouTube, в якому пояснюється, як за допомогою САПР SolidWorks розробити тривимірну модель корпусу деякої мікросхеми. Якщо це посилання раптом не працюватиме, то у такому випадку ви самостійно можете знайти³ на YouTube подібні відеоролики, в яких описується розробка тривимірних моделей, та ознайомитись із ними.

Важливо розуміти, що навчальний посібник не може вмістити усю необхідну інформацію, щоб докладно описати усі аспекти роботи із програмою **Altium Designer**. Тому, щоб краще оволодіти навичками розробки бібліотечних компонентів, принципових схем та друкованих плат в **Altium Designer**, необхідно також самостійно ознайомитись із додатками⁴, літературними джерелами та інформаційними ресурсами (особливо зі спеціалізованими тематичними каналами на YouTube), на які є посилання у цьому навчальному посібнику.

² Оригінальна версія навчального посібника містить перевірені посилання. Тому завантажуйте цей посібник виключно із достовірних джерел, оскільки в електронних версіях, на жаль, можливі модифікації посилань сторонніми особами з метою переадресації на інші ресурси, що несуть потенційну небезпеку. Тож, переходячи за тим чи іншим посиланням, звертайте увагу на його формат, напис адреси ресурсу (наприклад, у посібнику не використовуються короткі посилання), контекст його використання в тексті тощо. Пам'ятайте, що усі ресурси, на які є посилання у посібнику, пов'язані із електронікою, розробкою електронних схем та плат, документацією САПР **Altium Designer**, технічною документацією на електронні компоненти тощо.

³ Вважається, що студенти на старших курсах навчання повинні мати навички самостійного пошуку інформації за допомогою різних пошукових систем.

⁴ Зокрема, у [Додатку А](#) наведені посилання на корисні інформаційні ресурси, які допоможуть краще розібратись в електроніці та схемотехніці загалом, а також навчитись розробляти електричні схеми та друковані плати у програмі **Altium Designer**.

Також у навчальному посібнику застосовується специфічна термінологія, яка не є широко вживаною. Інколи важко знайти правильні лаконічні україномовні відповідники англійським термінам або поняттям, тому у деяких випадках вживаються слова-запозичення та калька. Якщо читач цього навчального посібника знатиме більш кращі україномовні терміни, то автор буде вдячний за зворотний зв'язок і допомогу у виправленні.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО СИСТЕМУ АВТОМАТИЗОВАНОГО ПРОЄКТУВАННЯ СХЕМ ТА ДРУКОВАНИХ ПЛАТ ALTIUM DESIGNER

1. Структура та особливості САПР Altium Designer

Altium Designer – це комплексна система автоматизованого проектування, яка надає можливість реалізовувати складні проєкти електронних пристроїв, починаючи від етапу створення принципової схеми і завершуючи розробкою друкованої плати та передачею інформації для її виготовлення на виробництво.

Головна ідея САПР **Altium Designer** – *наскрізне проектування*, яке дозволяє виконувати розробку з урахуванням особливостей кожного етапу проектування.

В основі САПР **Altium Designer** 17.1 лежить програмна платформа *Design Explorer* (DXP), що об'єднує у собі різні модулі для реалізації всіх функцій наскрізного автоматизованого проектування:

- редактор схем;
- редактор друкованих плат та автотрасувальник;
- редактор бібліотек моделей електронних компонентів;
- програма моделювання різноманітних електронних схем;
- текстовий редактор списку з'єднань і описів мовою VHDL;
- редактор синтезу логіки для ПЛІС та інші модулі.

До складу програмного комплексу **Altium Designer** входить увесь необхідний інструментарій для розробки, редагування та налагодження проєктів на базі електричних схем та програмованої логіки. Редактор схем дозволяє створювати багатоієрархічні та багатоканальні схеми будь-якої складності, а також виконувати змішане цифро-аналогове моделювання.

Якщо ж у системі відсутні потрібні користувачу функції, то існує спеціальний набір Software Development Kit, що надає можливість створювати власні допоміжні модулі, які потім будуть інтегровані в єдину систему.

Важливим кроком в досягненні потрібного результату під час створення будь-якого проєкту є процес налагодження роботи схеми пристрою ще до його втілення «у залізо», тому до складу **Altium Designer** включена програма моделювання, яка

дозволяє розробникові відразу після завершення розробки принципової схеми виконати аналіз електричних характеристик, статистичний аналіз, змінити за потреби параметри тощо.

Altium Designer підтримує велику кількість типів аналізу, зокрема:

- частотний аналіз в режимі малого сигналу;
- аналіз перехідних процесів;
- розрахунок спектральної щільності внутрішнього шуму;
- аналіз передатних функцій за постійним струмом;
- статистичний аналіз вихідних електричних параметрів схеми методом Монте-Карло;
- аналіз впливу змін значень параметрів елементів схеми і температури на роботу схеми;
- спектральний аналіз Фур'є;
- можливості по математичній обробці розрахованих сигналів: їх додавання, віднімання, застосування до них різних математичних функцій.

Інтерфейс **Altium Designer** є дуже гнучким і може бути налаштований найбільш оптимальним чином для розробника. Наприклад, можна виконати налаштування для одночасного перегляду схеми і плати на одному моніторі (рис. 1), що дуже зручно під час пошуку помилок та налагодження.

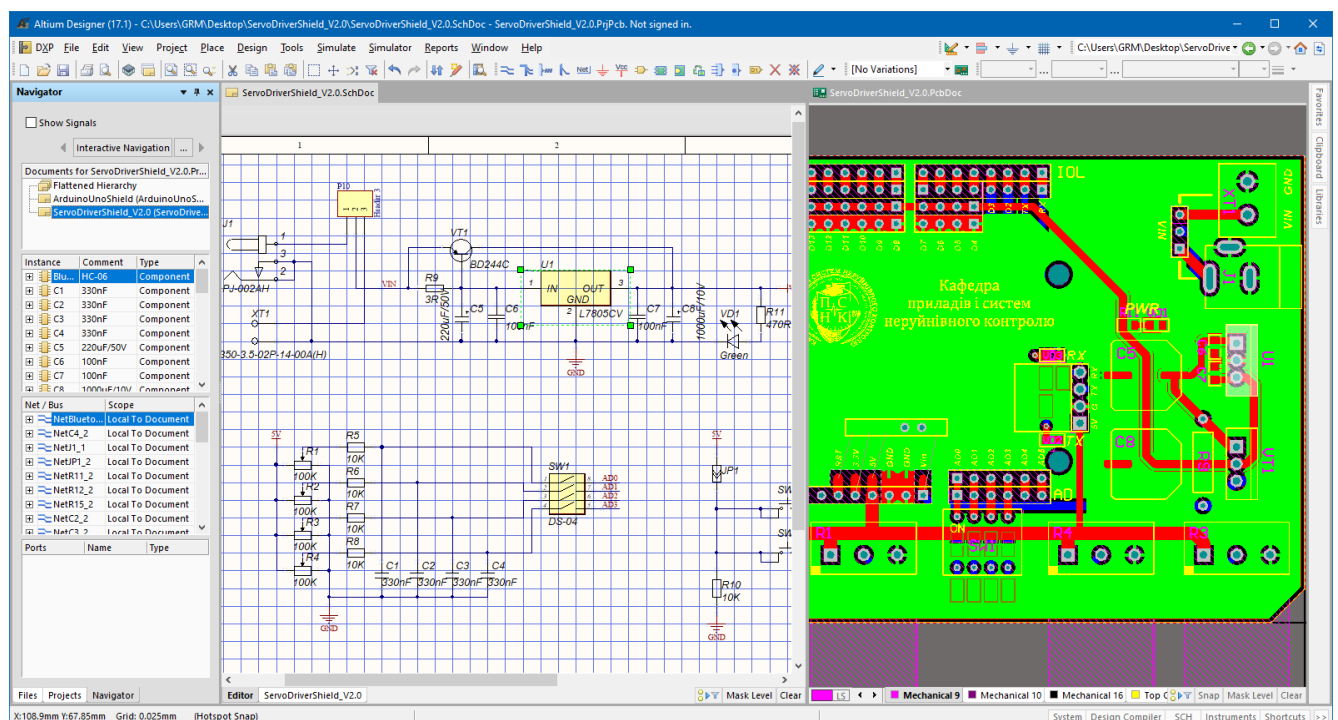


Рис. 1. Режим одночасного перегляду схеми і плати

Проте найкраще працювати у середовищі **Altium Designer** із використанням декількох моніторів (рис. 2).



Рис. 2. Приклад використання кількох моніторів під час роботи із **Altium Designer**

Завдяки тому, що **Altium Designer** підтримує можливість роботи із тривимірними моделями, розробник ще на етапі проєктування може бачити, як виглядатиме плата та розміщені на ній компоненти після виготовлення (рис. 3).

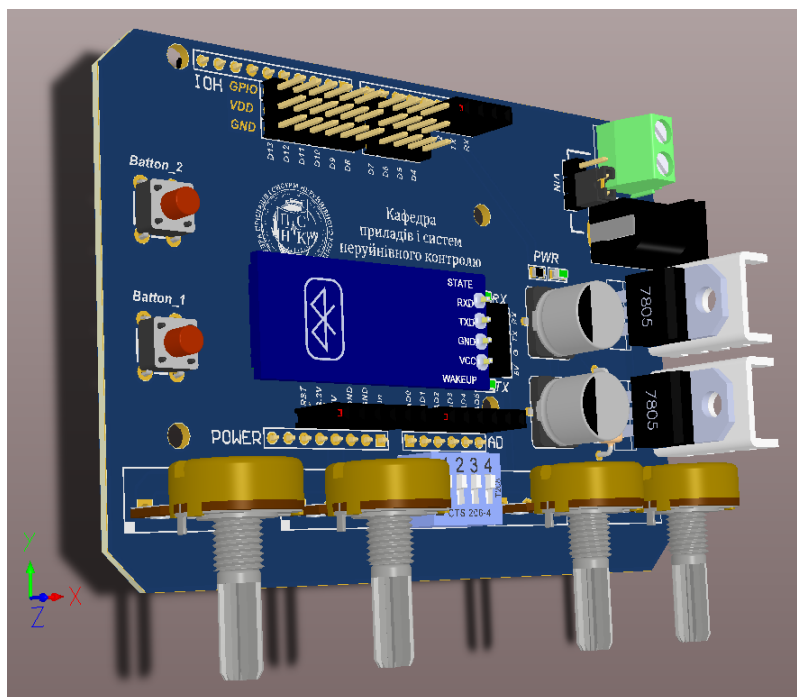


Рис. 3. Приклад тривимірної моделі плати

Важливою особливістю **Altium Designer** є величезна бібліотека, що містить сотні тисяч готових компонентів, у багатьох з яких є моделі ПМ, SPICE- та IBIS-

моделі, а також тривимірні моделі корпусів. Більшість з перерахованих вище моделей можна створити внутрішніми засобами програми.

У **Altium Designer** є спеціальний майстер для імпортування проєктів і готових бібліотек з інших САПР електронних компонентів (зокрема, Protel X, P-CAD 200X тощо), що розширює можливості зі створення власних бібліотек (така функція особливо корисна під час роботи з проєктами, отриманими від інших розробників, які використовували власні бібліотеки компонентів).

Насамкінець, варто відзначити потужну [довідникову систему Altium Designer](#), яка доступна на офіційному сайті компанії. Не буде перебільшенням сказати, що у довідці та на форумі ви знайдете відповіді на практично усі ваші запитання.

2. Основні принципи розробки компонентних модулів

Електронним компонентом називається складова частина електронного пристрою, що має вигляд закінченого виробу (рис. 4). По-суті, електронні компоненти є [базовими будівельними блоками електронного пристрою](#) та мінімальною одиницею заміни.

Електронні компоненти діляться за функціональними ознаками (пасивні та активні), інтеграцією (дискретні та інтегральні), технологією монтажу (для поверхневого монтажу, для монтажу в отвори тощо), типом і розташуванням виводів (аксіальні, радіальні, з матричним розташуванням виводів тощо), типом і матеріалом корпусу та іншими ознаками.

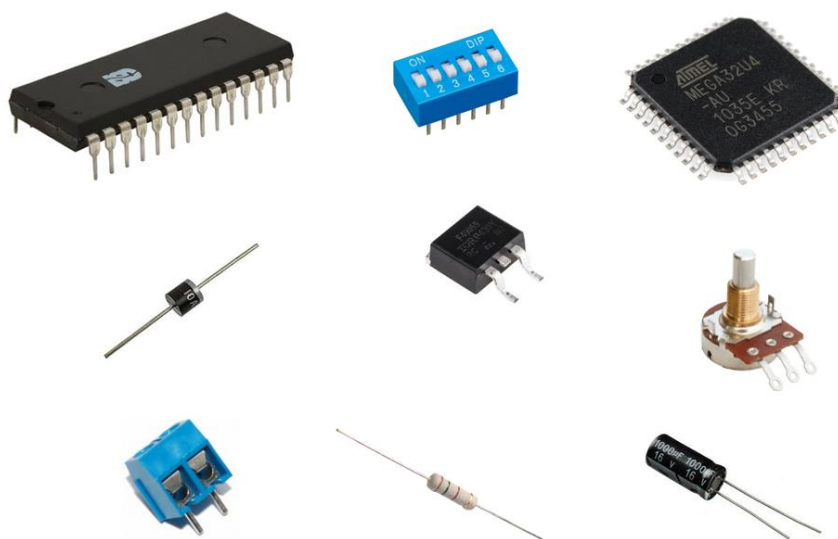


Рис. 4. Зображення різних електронних компонентів

У реальному світі електронний компонент є матеріальним об'єктом, який має явну форму. Однак, під час проєктування електронних схем та друкованих плат за допомогою САПР із використанням комп'ютерів, електронні компоненти абстрагуються у віртуальному світі у набір моделей, які по-різному представляють компонент у різних областях проєктування: на схемі – це умовне графічне позначення (або символ), на платі – посадкове місце тощо.

Тому під час роботи із САПР електронних компонентів проєктують не самі електронні компоненти, а так звані компонентні модулі (або бібліотечні компоненти). **Компонентний модуль** – це набір взаємопов'язаної конструкторської, технологічної та схемотехнічної інформації про електронний компонент, що монтується на друковану плату. Зазвичай типовий компонентний модуль містить:

- умовне графічне позначення електронного компонента, яке складається з набору графічних символів (примітивів), виводів та додаткової інформації (позиційне позначення, тип, номінал тощо);
- посадкове місце компоненту, яке є фрагментом друкованої плати і містить дані, що відносяться до конструкції та технології виготовлення друкованої плати, зокрема: параметри отворів, форми КМ тощо;
- дані про маркування, яке наноситься на друковану плату (може містити текстову і графічну інформацію, зокрема контури або умовні зображення електронних компонентів);
- текстову і довідкову інформацію, що використовується під час проєктування, а також формування специфікацій, переліків та інших документів, що входять до складу конструкторської документації на виріб;
- модель SPICE та/або IBIS для віртуального моделювання;
- тривимірну модель корпусу для візуалізації.

Перелік складових компонентного модуля може відрізнятися від наведеного вище і визначається конкретними вимогами проєкту, проте **компонентний модуль має містити як мінімум УГП та графічне зображення ПМ**. Під час комплексного проєктування пристрою використовуються усі перераховані складові частини.

На рис. 5 на прикладі транзистора показано, який взаємозв'язок існує між поняттям «електронний компонент» та «компонентний модуль». На принциповій схемі електронний компонент представляється його УГП, а на друкованій платі – зображенням ПМ для заданого типу корпусу.



Рис. 5. Представлення електронного компонента на схемі та платі

Таким чином, електронний компонент – це те, що фізично монтується на плату. А компонентний модуль – це результат об'єднання УГП та ПМ (а також за потреби й іншої інформації, наприклад, тривимірної моделі корпусу, допоміжної текстової інформації, математичних моделей для моделювання тощо), що зберігається у бібліотеці. Саме тому його ще називають бібліотечним компонентом.

Під час розробки компонентного модуля важливою задачею є вміння об'єднувати УГП електронного компонента, зображення його ПМ та інших моделей, що в майбутньому значно спростить поетапну реалізацію проєкту електронного пристрою, у якому буде використовуватись цей компонентний модуль.

Проте необхідно відзначити, що взаємозв'язок між УГП та ПМ в різних САПР забезпечується по-різному.

Наприклад, в САПР **P-CAD 2006** типове УГП створювалось один раз, після чого могло бути привласнене кільком різним бібліотечним компонентам. Після створення окремо УГП (symbol) та окремо ПМ (pattern), вони об'єднувались за допомогою програми Library Executive в окремий компонентний модуль. У такому

випадку важливо було іменувати УГП найбільш інформативно, щоб його легко можна було відшукати у бібліотеці, наприклад: резистор 0,25 Вт; резистор 0,5 Вт; резистор підлаштовувальний; конденсатор керамічний; транзистор біполярний р-п-р; елемент І; елемент АБО-НЕ тощо.

В **Altium Designer** використовується інший підхід. Хоча у цій САПР для створення компонентного модуля використовуються дві бібліотеки (з розширенням ***.SchLib** – для створення і зберігання УГП (symbol), та з розширенням ***.PcbLib** – для створення і зберігання графічних зображень ПМ (footprint)), а також окремі файли інших моделей, проте саме **бібліотека з розширенням *.SchLib є основною, а те, що в ній зберігається, називається або символом, або компонентом⁵** (component).

Важливо!

У САПР Altium Designer бібліотека з розширенням *.SchLib є основою компонентного модуля, до якого можна підключати ПМ та інші моделі у вигляді посилань.

Якщо деякий елемент бібліотеки ***.SchLib** містить тільки УГП, то він називається *символом*, але якщо до символу підключені інші моделі (наприклад, модель ПМ, тривимірна модель тощо), то він стає компонентом (або компонентним модулем, або бібліотечним компонентом).

Зазвичай основною метою є створення саме компонентних модулів та їх збереження у бібліотеці. У такому разі назва компонентного модуля у бібліотеці ***.SchLib** має з самого початку відображати не найменування УГП, а реальну назву самого електронного компоненту. Наприклад, компонентний модуль називатиметься не «резистор» (назва УГП), а «МЛТ 0.125» (конкретна назва електронного компоненту); не «транзистор біполярний п-р-п», а «BC547», не «підсилювач», а «AD797» тощо.

Проте наявність окремо зразків УГП (або символів) у бібліотеці ***.SchLib** може бути досить корисною, що дасть змогу швидко на основі існуючого зразка створити новий компонентний модуль. Наприклад, можна створити символ резистора з назвою «Резистор 0,125» без ПМ та інших моделей, який не буде

⁵ Не плутайте із електронним компонентом, про який йшла мова раніше. Назва «компонент» у цьому випадку відповідає терміну «компонентний модуль».

використовуватись у проєктах, а буде використовуватись тільки для створення нових компонентних модулів резисторів потужністю 0,125 Вт. Наявність окремо зразків УГП, звичайно, дещо захаращуватиме бібліотеку, але, окрім швидкого пошуку, дає ще одну перевагу – під час редагування зразка УГП усі наступні нові компонентні модулі будуть містити останню версію зображення. Варто відзначити, що зображення УГП у вже існуючих (попередніх) компонентних модулях залишаться незмінними.

Якщо розробник приймає рішення зберігати у бібліотеці окремо зразки УГП, то важливо їх іменувати найбільш інформативно, щоб не було дублювання елементів у бібліотеці. Наприклад, зразки УГП цифрових мікросхем можна називати: «Елемент І»; «Елемент АБО-НЕ» тощо. Зразок має сенс створювати тільки тоді, коли його можна назвати настільки інформативно, що користувачі бібліотек з великою ймовірністю зможуть скористатися цим зразком, орієнтуючись за назвою.

Другою важливою бібліотекою є бібліотека з розширенням ***.PcbLib**, в якій створюють ПМ електронних компонентів. Оскільки ПМ є частиною топології плати та призначене для монтажу електронного компонента, тому містить відомості про тип (штирові або планарні) та форму КМ, а також конструктив (форму) корпусу. ПМ створюється на основі кресленика корпусу електронного компонента, тому важливо уважно ознайомитись із технічною документацією, щоб правильно його розробити. У бібліотеці зазвичай ПМ називають за типом корпусу (тобто важливо уміти розбиратися у класифікації корпусів). Після створення ПМ за певними правилами підключається до компонентного модуля безпосередньо у бібліотеці з розширенням ***.SchLib**.

Тривимірна модель корпусу може бути створена у зовнішній програмі (наприклад, SolidWorks, AutoCAD, CATIA тощо), після чого її необхідно підключити⁶ до компонентного модуля, а може бути й створена засобами самої програми **Altium Designer**.

⁶ Якщо тривимірна модель корпусу створюється у зовнішній програмі, то вона має бути збережена в окремому файлі (зазвичай це файл у форматі step, проте можуть бути й інші варіанти), після чого дані із цього файлу імпортуються у компонентний модуль. Зверніть увагу, що тривимірна модель спочатку підключається до моделі ПМ у бібліотеці ***.PcbLib**, а не до моделі УГП у бібліотеці ***.SchLib**. Це пов'язано із тим, що тривимірну модель необхідно коректно повернути, вирівняти та розмістити саме відносно ПМ.

2.1. Створення УГП електронного компонента

Умовне графічне позначення⁷ – це графічне зображення, яке використовується для позначення електронного компонента на електричних схемах (рис. 6).

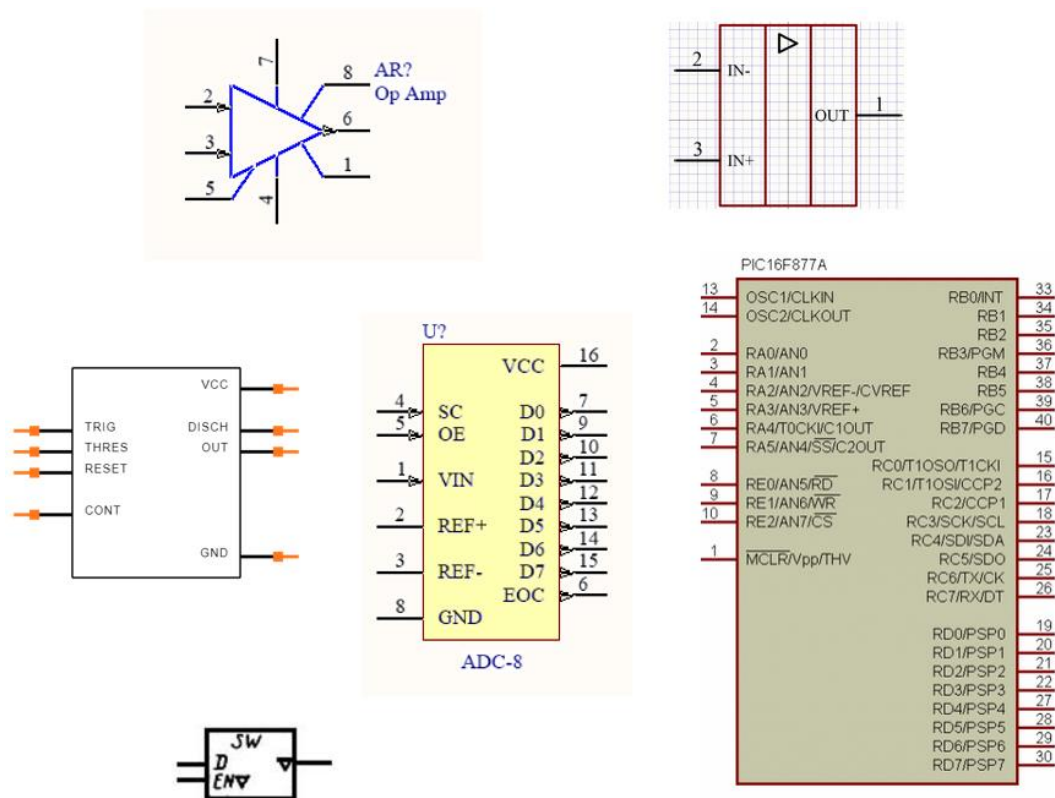


Рис. 6. Приклади умовних графічних позначень

Під час розробки УГП важливо розуміти, що сьогодні існують різні стандарти, які визначають їх зовнішній вигляд та правила побудови. Докладніше про це можна прочитати у [Додатку Б](#).

Проте, незалежно від стандарту, формування УГП у тій чи іншій САПР електронних компонентів можна розділити на 3 етапи:

- створення зображення (або контуру) УГП за допомогою графічних примітивів;
- встановлення виводів та їх нумерування;
- додавання позиційного позначення та інших атрибутів.

⁷ В закордонних літературних джерелах та стандартах здебільшого застосовується термін «символ» (symbol).

Для створення контуру УГП використовуються зазвичай такі графічні примітиви, як лінії, прямокутники, кола, дуги, криві, текстові поля тощо.

Виводи є окремими графічними об'єктами, що входять до складу УГП. На принциповій схемі компоненти з'єднуються між собою лініями електричного зв'язку, які можуть бути проведені лише між выводами компонентів. Таким чином, якщо в УГП відсутні виводи, його не можна застосувати для побудови схем. Виводи можуть відображатись різними стилями, проте усі вони мають розташовуватись тільки на лініях, що визначають контури УГП.

Позиційне позначення – це текстове поле для ідентифікації компонента на схемі. Зазвичай позиційне позначення розміщується над контуром УГП та складається із двох частин: літерної (формується на основі сімейства компонента) та цифрової (вказує порядковий номер компонента та/або секції).

Атрибути УГП є текстовими полями та використовуються для відображення додаткових відомостей про компонент. Зокрема, до атрибутів належать номінал, розсіювана потужність, назва електронного компонента, тип корпусу мікросхеми, виробник тощо. У САПР **Altium Designer** перелік атрибутів та можливість їх відображення можна налаштовувати. Згодом атрибути можуть бути використані під час формування технічної документації.

УГП в САПР **Altium Designer** створюється у редакторі символів **Schematic Library Editor**.

2.2. Створення посадкового місця електронного компонента

Посадкове місце електронного компонента (footprint, land pattern) – це комбінація КМ, що використовуються для монтажу та з'єднання окремих електронних компонентів на ДП. Окрім цього, ПМ також містить графічну та текстову інформацію, які полегшують розміщення, позиціонування та пошук компонентів під час виготовлення та експлуатації ДП.

Процес створення ПМ складається із 3-х етапів:

- розміщення КМ та їх нумерування;
- додавання графічних примітивів, що визначають контури корпусу, контури забороненої для трасування зони тощо, а також розміщення текстових написів (наприклад, позиційного позначення);

- розміщення та позиціонування тривимірної моделі корпусу електронного компонента⁸ відносно КМ.

ПМ електронного компонента в САПР **Altium Designer** створюється у редакторі **PCB Library Editor**.




3. Поняття проєкту в Altium Designer

Все, що розробляється в **Altium Designer**, зазвичай називається проєктом. Проєкт є набором файлів, що містять вихідні дані та зберігаються в одній папці. У цій же папці розташовується спеціальний файл, який називається **файлом проєкту**. У ньому міститься опис наявних документів (тобто інших файлів), відносних шляхів до них та зав'язків між ними. Також у цій же папці за замовчуванням будуть згодом зберігатись файли звітів, sam-файли тощо.

Важливо!

Рекомендується для кожного проєкту створювати свою папку з унікальною назвою. Це зручно не тільки з огляду на структурування даних, але також і тому, що, наприклад, згодом папку можна перенести на інший комп'ютер, відкрити проєкт і одержати доступ до всіх його документів.

Основними проєктами⁹ в **Altium Designer** є проєкт плати, проєкт бібліотеки, проєкт програмованої логічної інтегральної схеми (FPGA), проєкт логічного ядра та скрипт-проєкт. Відповідно є наступні файли проєктів:

- файл проєкту друкованої плати **.PrjPCB** ();
- файл проєкту бібліотеки **.LibPkg** ();
- файл проєкту FPGA **.PrjFpg** ();

⁸ Насправді третій етап не зовсім належить до процесу створення посадкового місця, оскільки тривимірна модель корпусу електронного компоненту використовується для візуалізації та уявлення про його габарити. Зазвичай тривимірна модель потрібна під час розробки ДП, розуміння її габаритів та уникнення просторових конфліктів між електронними компонентами.

⁹ У цьому навчальному посібнику основна увага буде приділена створенню проєктів бібліотек та проєктів плат.

- файл проєкту логічного ядра **.PrjCor** (📁);
- файл скрипт-проєкту **.PrjScr** (📄).

Для того, щоб створити новий проєкт, необхідно у меню **File** вибрати пункт **New** → **Project**. Після чого у вікні, що відкриється, обирають тип проєкту.

4. Створення проєкту плати в Altium Designer

Для створення нового проєкту плати необхідно обрати тип проєкту **PCB Project**.

На рис. 7 наведено типову структуру проєкту плати¹⁰ в **Altium Designer**, а на рис. 8 – вміст папки деякого проєкту на диску.

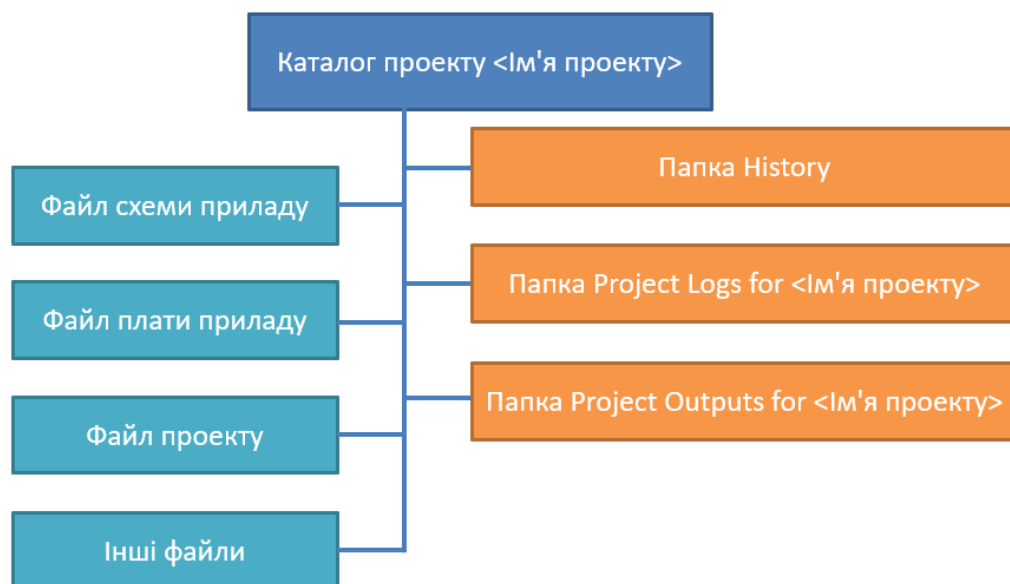







Рис. 7. Типова структура проєкту плати в **Altium Designer**

History	30.11.2020 19:16	Папка с файлами	
Project Logs for PulseGenerator	23.11.2020 19:34	Папка с файлами	
Project Outputs for PulseGenerator	01.12.2020 11:50	Папка с файлами	
PulseGenerator.OutJob	23.11.2020 20:44	Altium Output Jo...	14 КБ
PulseGenerator.PcbDoc	30.11.2020 17:42	Protel PCB Docu...	720 КБ
PulseGenerator.pdf	23.11.2020 20:19	Foxit PDF Reader ...	324 КБ
PulseGenerator.PrjPcb	30.11.2020 19:16	Altium PCB Project	59 КБ
PulseGenerator.PrjPcbStructure	23.11.2020 20:19	Файл "PRJPCBSTR...	1 КБ
PulseGenerator.SchDoc	22.11.2020 23:40	Altium Schematic ...	46 КБ

Рис. 8. Вміст папки проєкту плати на диску

¹⁰ Структура проєкту бібліотеки буде розглянута дещо пізніше.

У папці проекту плати **Altium Designer** зазвичай містяться такі файли:

- файл проекту друкованої плати ***.PrjPCB** ();
- файл електричної схеми ***.SchDoc** ();
- файл плати ***.PcbDoc** ();
- файл з розширенням ***.Cam** () для засобу перегляду і редагування Gerber-файлів (такий файл містить в собі результати імпортування Gerber-файлів у програму класу CAM);
- файл з розширенням ***.OutJob** (), в якому містяться налаштування відображення результатів розробки плати у файли документації, Gerber-файли, NC-Drill файли тощо.

У вкладеній папці **History** зберігаються архіви із попередніми версіями проекту. Їх можна використати для відновлення даних.




У вкладеній папці **Project Logs for <Ім'я проекту>** зберігаються згенеровані журнали роботи.

У вкладеній папці **Project Outputs for <Ім'я проекту>** зберігаються згенеровані звіти про пошук помилок, файли документації, Gerber-файли тощо.

5. Робота із бібліотеками

Бібліотеки електронних компонентів є важливою складовою будь-якого проекту електронною пристрою. САПР **Altium Designer** містить як величезну колекцію готових бібліотек (які розміщені на сервері компанії), так і надає користувачеві інструменти для розробки власних бібліотек.

Коли користувач розробляє власну бібліотеку, то він повинен спочатку створити проект бібліотеки. У папці проекту бібліотеки **Altium Designer** зазвичай містяться такі файли:

- файл проекту бібліотеки ***.LibPkg** (), у якому містяться усі необхідні зв'язки та посилання між різними моделями;
- файл бібліотеки УГП електронних компонентів ***.SchLib** ();
- файл бібліотеки графічних зображень ПМ електронних компонентів («футпрінтів») ***.PcbLib** ().

На основі проекту бібліотеки після виконання процедури компіляції створюють єдиний файл інтегрованої бібліотеки ***.IntLib**, який можна використовувати під час розробки схем і плат.

Користь від інтегрованих бібліотек полягає у тому, що усі типи моделей (УГП, ПМ, тривимірні моделі тощо) містяться не у різних бібліотеках і файлах, а в одному файлі інтегрованої бібліотеки¹¹. Додатковою користю від створення інтегрованої бібліотеки є те, що під час її компіляції виконується перевірка на помилки. Це дає змогу вчасно усунути помилки, що згодом пришвидшує розробку кінцевого виробу.

Бібліотеки можуть містити невелику кількість компонентів, а можуть містити і тисячі компонентів. Розробнику важливо навчитись використовувати потужні можливості пошуку в бібліотеках. Розглянемо на прикладі, як це можна робити.

Доступ до бібліотек виконується за допомогою панелі ***Libraries*** (рис. 9), яку можна відкрити або натиснувши кнопку **System** в правому нижньому куті основного вікна і вибравши ***Libraries*** з контекстного меню, або натиснувши на спеціальній вкладці ***Libraries***, що знаходиться у правій частині вікна Редактора схем. Вибір необхідної бібліотеки відбувається із випадного меню, що розташоване у верхній частині панелі.

Найпростішим способом пошуку компоненту в бібліотеці є використання фільтру, який розташований у рядку нижче назви обраної бібліотеки у панелі ***Libraries***. За замовчуванням у рядку фільтру задано груповий символ «*», що встановлює режим відображення усіх компонентів, які містяться у бібліотеці. Наприклад, якщо замість символу «*» ввести слово «diode», то буде відображено список усіх доступних у поточній бібліотеці компонентів, що мають у своїй назві текст «diode».

Більш гнучким є розширений пошук компонентів. Для того, щоб його використовувати, необхідно відкрити діалогове вікно ***Libraries Search***. Це можна зробити або натиснувши кнопку **Search** на панелі ***Libraries***, або обравши із меню ***Tools*** пункт ***Find Component***. Розширений пошук дає змогу шукати як компонентні модулі, так і ПМ, тривимірні моделі тощо. Пошук може бути виконаний як у підключених до проєкту, так і у зовнішніх бібліотеках, що зберігаються на диску. Також можна шукати як за повним співпадінням, так і частковим, шукати не тільки

¹¹ Фактично, можете розглядати інтегровану бібліотеку як своєрідний архів, у якому в одному файлі зібрана уся необхідна інформація.

за назвою компонента, а й за його параметрами (наприклад, номіналом, описом, URL-адресою тощо).

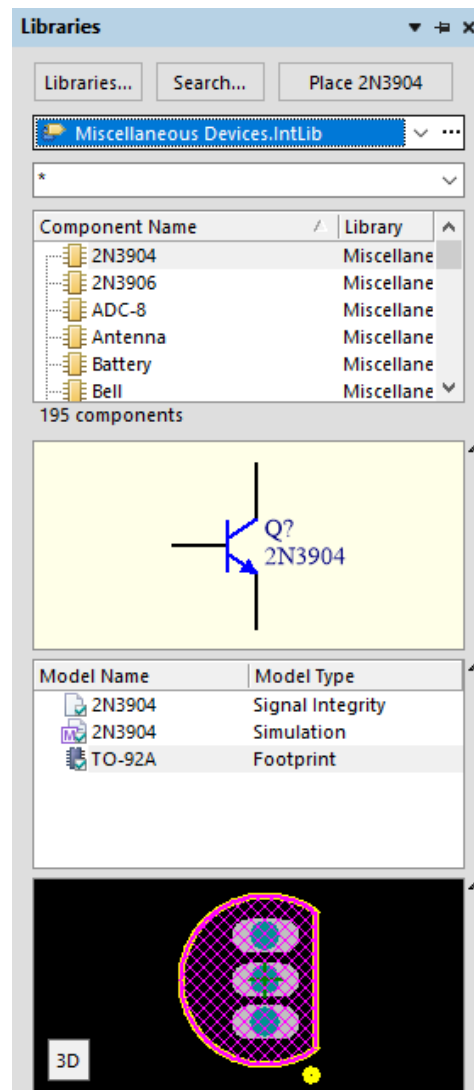


Рис. 9. Панель *Libraries*

6. Редактор схем

Для створення принципових схем в **Altium Designer** використовується редактор схем **Schematic Editor** (рис. 10).

Редактор схем дозволяє проектувати принципові схеми будь-якої складності. Він є не просто графічним редактором, в якому створюється кресленик схеми, але й додатково містить інструменти для аналізу та налагодження схеми, групового редагування, синхронізації із Редактором плат тощо.

Процес створення схеми у Редакторі схем зазвичай складається з наступних ключових етапів:

- розміщення УГП електронних компонентів на схемі;

- з'єднання компонентів шляхом додавання ліній зв'язку¹² (електричних ланцюгів) між виводами або шляхом розміщення ідентифікаторів ланцюгів для підключення виводів;
- пошуку помилок, аналіз та виправлення схеми, якщо це потрібно.

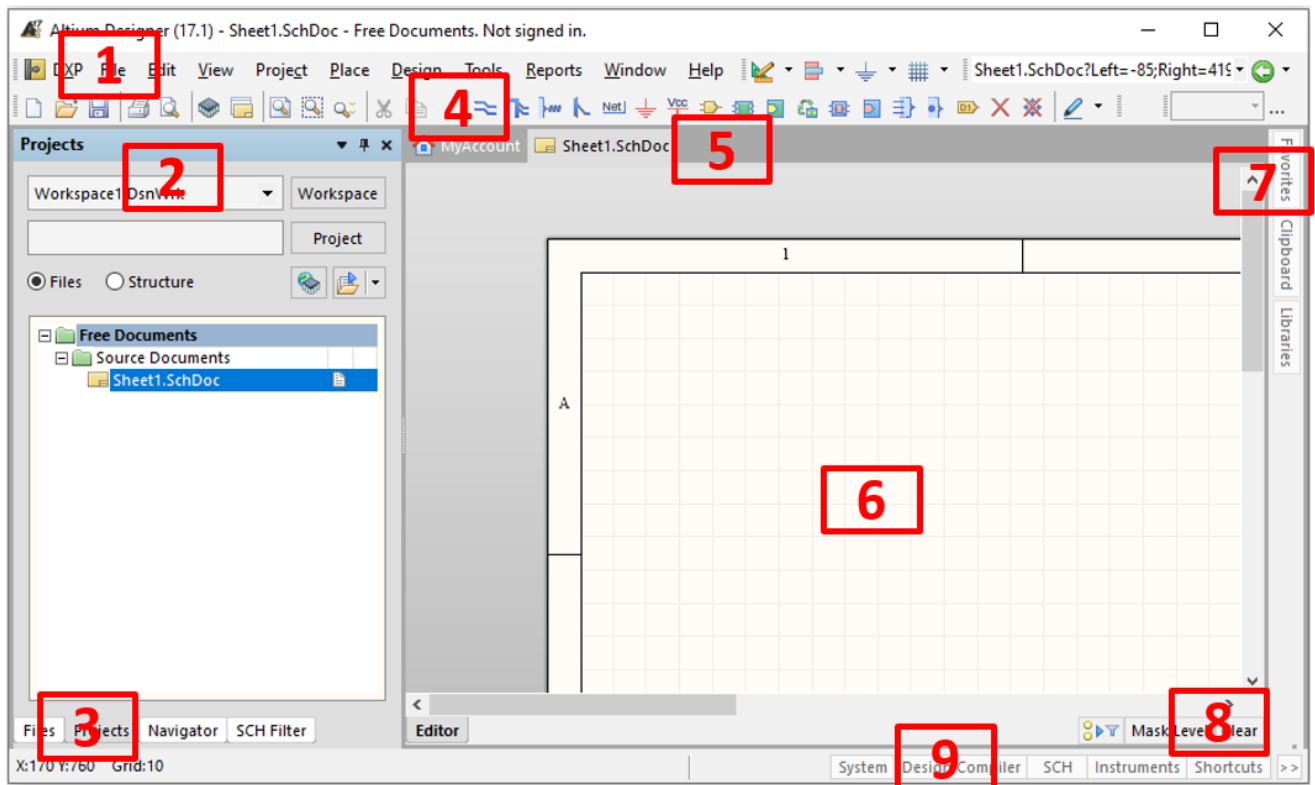


Рис. 10. Редактор схем: 1 – головне (системне) меню, 2 – допоміжна ліва бічна панель, що має кілька режимів відображення, 3 – закладки бічної панелі, 4 – панель інструментів, 5 – закладки відкритих документів, 6 – робоча область, 7 – закладки для відображення відкритих правих бічних панелей, 8 – налаштування відображення елементів в документі, 9 – кнопки швидкого виклику бічних панелей

Для налаштування Редактора схем необхідно виконати команду **Design** → **Document Options**. У вікні, що з'явиться, можна задати орієнтацію документа, формат аркушу, колір, змінити систему вимірювання тощо.

Наприклад, зміна кольору фону відбувається у вкладці **Sheet Options** в полі **Options** у рядку **Sheet Color**. А розмір робочої області можна обрати або зі стандартних у списку **Standard Styles**, або встановити власний, попередньо активувавши прапорець **Use Custom style**.

¹² Якщо проєкт містить компоненти із великою кількістю виводів, то задавати усі зв'язки шляхом розміщення окремих ліній недоцільно. У такому випадку ланцюги об'єднують у шину.

7. Редактор плат

Для створення топології ДП в **Altium Designer** використовується редактор плат **PCB Editor** (рис. 11), що має інтуїтивно зрозумілий інтерфейс та містить потужні інструменти як для ручного, так і автоматичного трасування.

Процес створення плати у Редакторі плат зазвичай складається з наступних ключових етапів:

- створення контуру плати;
- розміщення ПМ електронних компонентів на платі;
- трасування (інтерактивне ручне, автоматичне або змішане);
- пошук та виправлення помилок¹³.

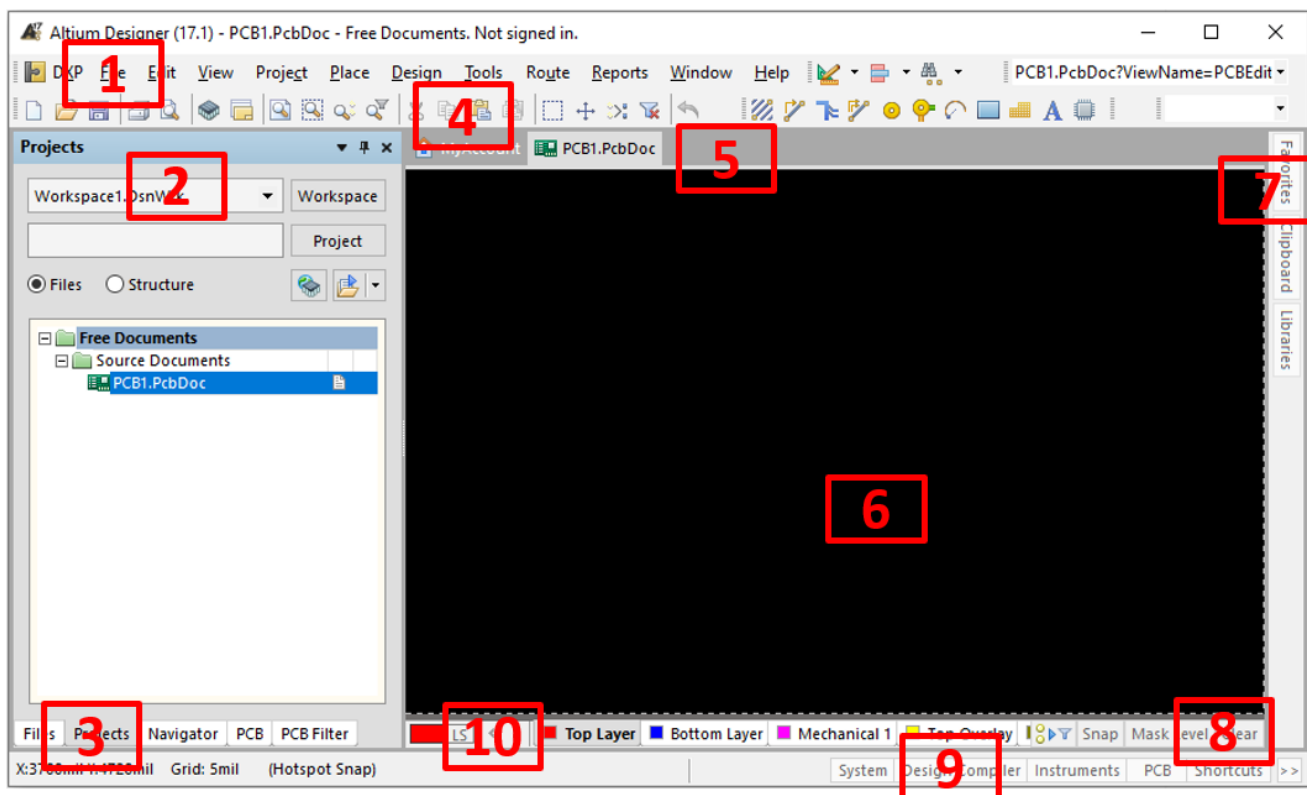


Рис. 11. Редактор плат: 1 – головне (системне) меню, 2 – допоміжна ліва бічна панель, що має кілька режимів відображення, 3 – закладки бічної панелі, 4 – панель інструментів, 5 – закладки відкритих документів, 6 – робоча область, 7 – закладки для відображення відкритих правих бічних панелей, 8 – налаштування відображення елементів у документі, 9 – кнопки швидкого виклику бічних панелей, 10 – закладки шарів друкованої плати

¹³ Цей етап може бути одним з найдовших під час розробки плати.

Для налаштування Редактора плат необхідно виконати команду **Design** → **Board Options**. Одним з найважливіших параметрів у вікні налаштувань **Board Options** є одиниці вимірювань, які можна задати в області **Measurement Unit**. Причому вибір цей здійснюється зі списку, який містить лише два варіанти: дюймова (**Imperial**) або метрична (**Metric**).

Також під час роботи над проєктом плати важливо правильно налаштувати відображення сітки, оскільки вона відіграє важливу роль під час розміщення ПМ та трасування. Для того, щоб отримати доступ до редагування параметрів сітки, виконують команду **Tools** → **Grid Manager**. У вікні, що відкриється (рис. 12), необхідно натиснути кнопку **Menu** у лівому нижньому куті та обрати необхідний пункт. Зокрема, налаштувати крок та колір сітки можна, обравши пункт **Properties**.

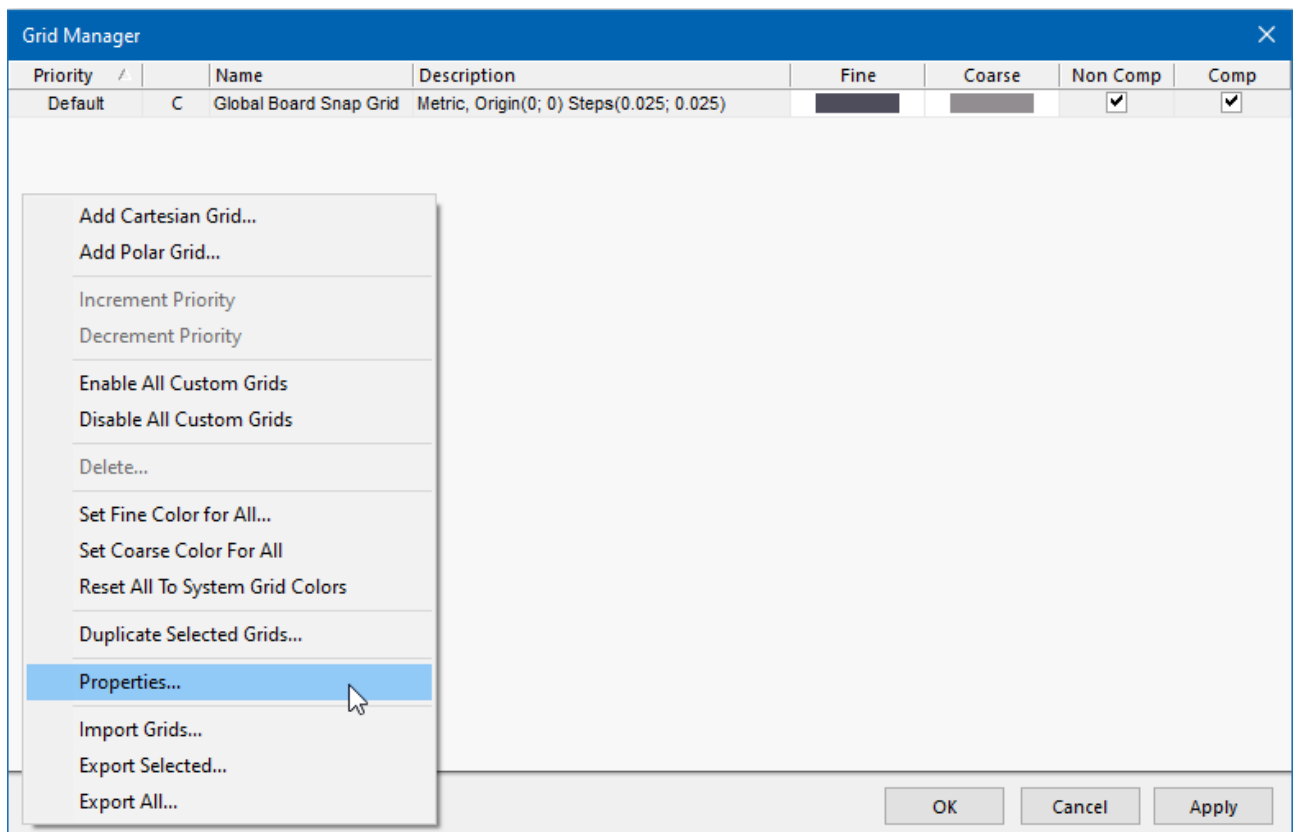


Рис. 12. Вікно **Grid Manager**

За допомогою Редактора плат ви можете як створити окремий проєкт плати без прив'язки до принципової схеми, так і редагувати плату, дані в яку передаються із існуючої принципової схеми. Загалом, перший варіант зручно використовувати, коли вам потрібно швидко створити маленьку та дуже просту плату. Проте, якщо ви працюєте над великим проєктом, то другий варіант є невід'ємною частиною процесу розробки.

Важливо!

Створення і редагування тільки плати без її прив'язки до проєкту (в якому має бути й принципова схема) не відповідає ідеології наскрізного проектування Altium Designer. Саме тому в Altium Designer не передбачені ефективні інструменти для трасування плати поза проєктом. Як буде показано далі, у такому випадку доводиться «викручуватись» та виконувати певні налаштування, а також змиритися із тим, що неможливо адекватно застосувати повну перевірку дотримання правил проектування плати. Тому буває легше і швидше створити принципову схему і зібрати усе в один проєкт, або ж, якщо потрібна тільки плата, то скористатись іншими системами автоматизованого проектування.

7.1. Основні способи трасування у PCB Editor

Процес прокладання провідників, який ще називають трасуванням, є одним з найважливіших під час проектування плати. Трасування полягає у визначенні шляху, який з'єднує виводи електронних компонентів на платі. Створення цього шляху виконується за допомогою розміщення на провідних шарах плати таких об'єктів, як траси, дуги та перехідні отвори, які спільно формують неперервне електричне з'єднання між виводами електронних компонентів на ДП.

В **Altium Designer** доступні два режими трасування: інтерактивне ручне та автоматичне із використанням автотрасувальника.

Більш докладніше про те, яким чином на практиці виконується ручне та автоматичне трасування, буде описано у комп'ютерних практикумах. Тут лише розглянемо важливі поради та рекомендації, які необхідно враховувати під час трасування.

7.1.1. Автоматичне трасування

Для автоматичного трасування використовується вбудований топологічний автотрасувальник **Situs**, який використовує прогресивні алгоритми створення топології плати. Він дозволяє виконати оптимальне трасування всієї плати або її частини. Однак для того, щоб отримати якісні результати роботи автотрасувальника, розробник спочатку повинен налаштувати правила трасування згідно необхідних йому вимог.

Основні команди, які використовуються для автотрасування, знаходяться у меню **Route**, пункт **Auto Route**:

- **All** – трасувати усю плату;
- **Net** – трасувати один ланцюг;
- **Net Class** – трасувати клас ланцюгів;
- **Connection** – трасувати усі ланцюги, що приєднані до деякого виводу;
- **Area** – трасувати усі провідники, що з'єднують виводи у виділеній області;
- **Room** – трасувати усі з'єднання між компонентами у вибраній області розміщення («кімнаті»);
- **Component** – трасувати усі з'єднання, які сполучаються із обраним компонентом;
- **Component Class** – трасувати усі з'єднання, які сполучаються із обраним класом компонентів;
- **Connections on selected components** – трасувати усі провідники всередині обраних компонентів;
- **Connections between selected components** – трасувати усі провідники між обраними компонентами.

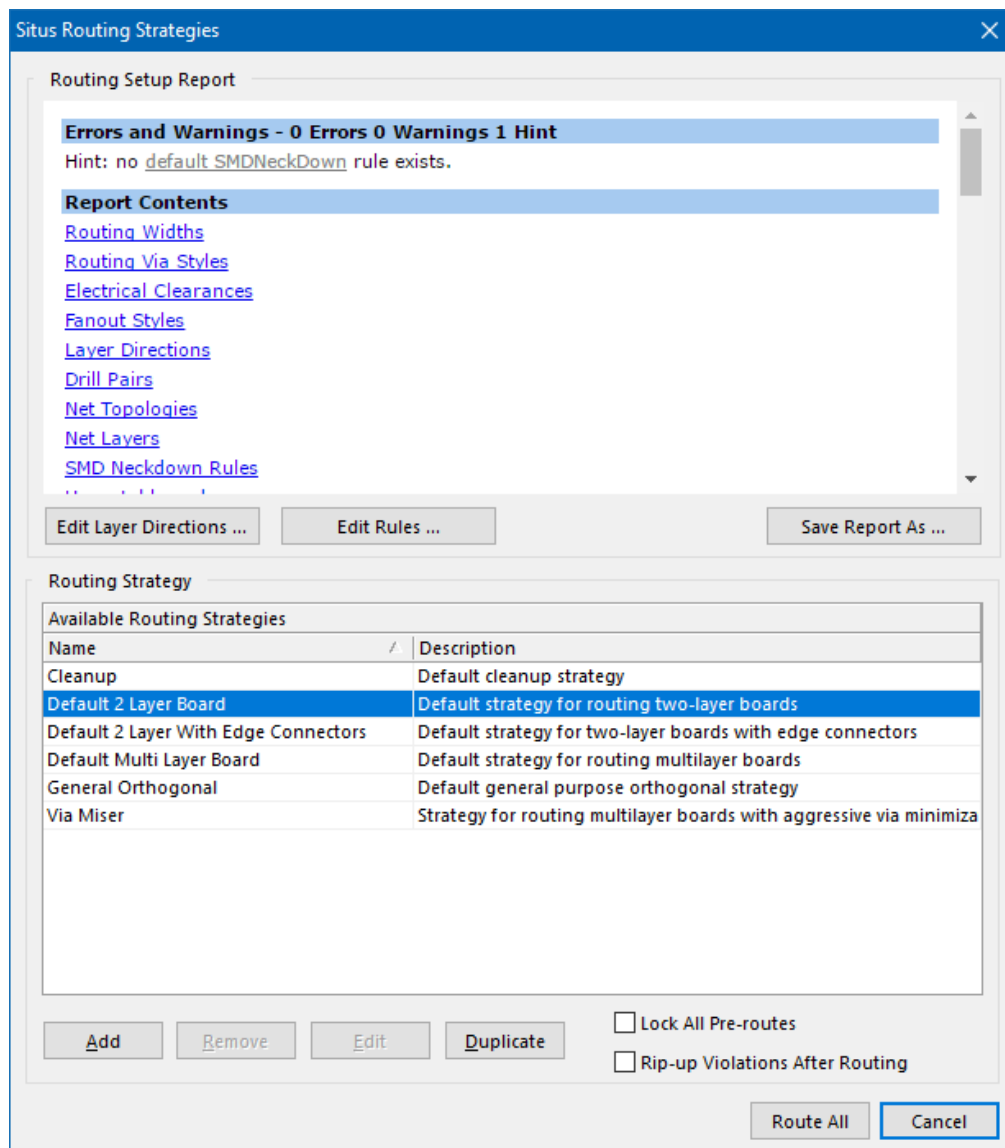
Якщо обрати пункт **All** (трасування усієї плати), то відкриється вікно **Situs Routing Strategies** (рис. 13).

У нижній половині вікна в області **Routing Strategy** можна обрати одну із задалегідь заданих еталонних стратегій трасування:

- **Cleanup** – «очищення» топології;
- **Default 2 Layer Board** – проста двошарова плата;
- **Default 2 Layer With Edge Connectors** – двошарова плата з торцевим роз'ємом;
- **Default Multilayer Board** – багатошарова плата;
- **General Orthogonal** – переважно ортогональне трасування;
- **Via Miser** – трасування з мінімізацією числа перехідних отворів.

Також можна увімкнути деякі параметри, для чого необхідно встановити галочки в наступних пунктах:

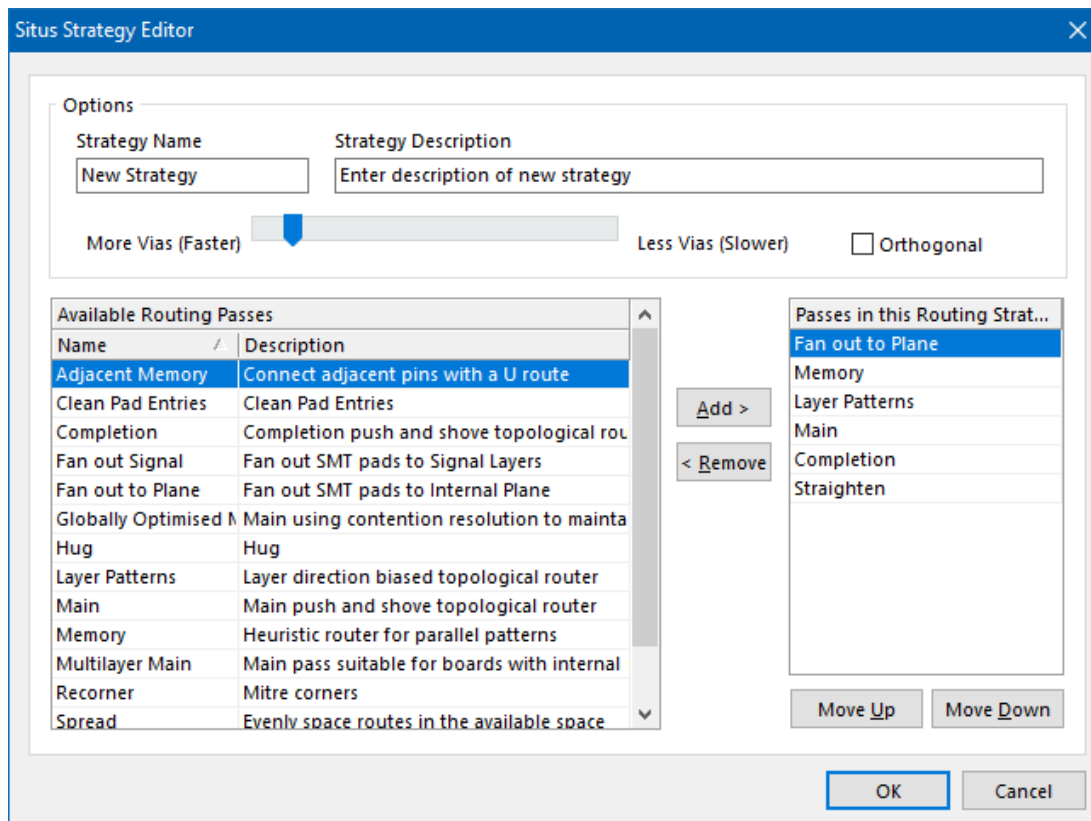
- **Lock All Pre-routes** – дозволяє зафіксувати усі раніше прокладені траси;
- **Rip-up Violations After Routing** – видаляє конфліктні ділянки топології після трасування.

Рис. 13. Вікно *Situs Routing Strategies*

Зверніть увагу, що серед доступних стратегій немає стратегії трасування тільки на одній стороні плати (такий підхід часто використовується для простих проєктів плат, які можна виготовити у домашніх умовах). Якщо необхідно виконати трасування тільки на одному шарі, то можна скористатись одним із двох підходів. Перший полягає у тому, що у вікні *Situs Routing Strategies* необхідно натиснути кнопку **Edit Layer Directions** і для або верхнього, або нижнього шару плати змінити значення поля **Current Setting** із **Automatic** на **Not Used**. Інший підхід полягає у зміні правила проєктування *Routing Layers*¹⁴.

¹⁴ Для зміни цього правила необхідно у меню *Design* обрати пункт *Rules*. У лівій частині вікна *PCB Rules and Constraints Editor* зі списку *Routing* активувати пункт *Routing Layers* та прибрати галочку навпроти того шару плати, який не повинен використовуватись під час трасування.

Якщо ж розробника не влаштовують існуючі стратегії трасування, то він має можливість створити власну. Для цього необхідно натиснути кнопку **Add** у нижній частині вікна *Situs Routing Strategies* (див. рис. 13), після чого з'явиться нове вікно *Situs Strategy Editor* (рис. 14).

Рис. 14. Вікно *Situs Strategy Editor*

Доступні параметри, які можна додати до власної стратегії трасування, знаходяться у списку *Available Routing Passes*:

- **Adjacent Memory** – з'єднує виводи U-подібними провідниками;
- **Clean Pad Entries** – «очищує» підходи до КМ;
- **Completion** – намагається завершити трасування, для чого використовуються методи розриву і розштовхування перешкод;
- **Fan Out Signal** – створення коротких сегментів доріжок для КМ поверхового монтажу з переходами на сигнальні шари;
- **Fan Out to Plane** – створення коротких сегментів доріжок для КМ поверхового монтажу з переходами на внутрішні шари живлення та землі;
- **Globally Optimized Main** – забезпечує оптимальне трасування: під час першої ітерації ігнорує суперечливі моменти та порушення, після чого

на другій ітерації виконує заново трасування з'єднань, використовуючи алгоритм оцінки «вартості» конфліктних ситуацій, доки не залишиться жодних порушень;

- **Hug** – ущільнює прокладені доріжки;
- **Layer Pattern** – використовує шаблони трасування з урахуванням переважаючого напрямку на шарі;
- **Main** – головна процедура топологічного трасування з використанням методів розриву та розштовхування перешкод;
- **Memory** – знаходить усі ланцюги, пов'язані з пристроями пам'яті або схожі на такі, та використовує евристичний алгоритм;
- **Recorner** – додає згладжування прямих кутів;
- **Spread** – рівномірно використовує доступне для трасування місце;
- **Straighten** – виконує загальне очищення топології.

7.1.2. Інтерактивне ручне трасування плати, яка є частиною проєкту

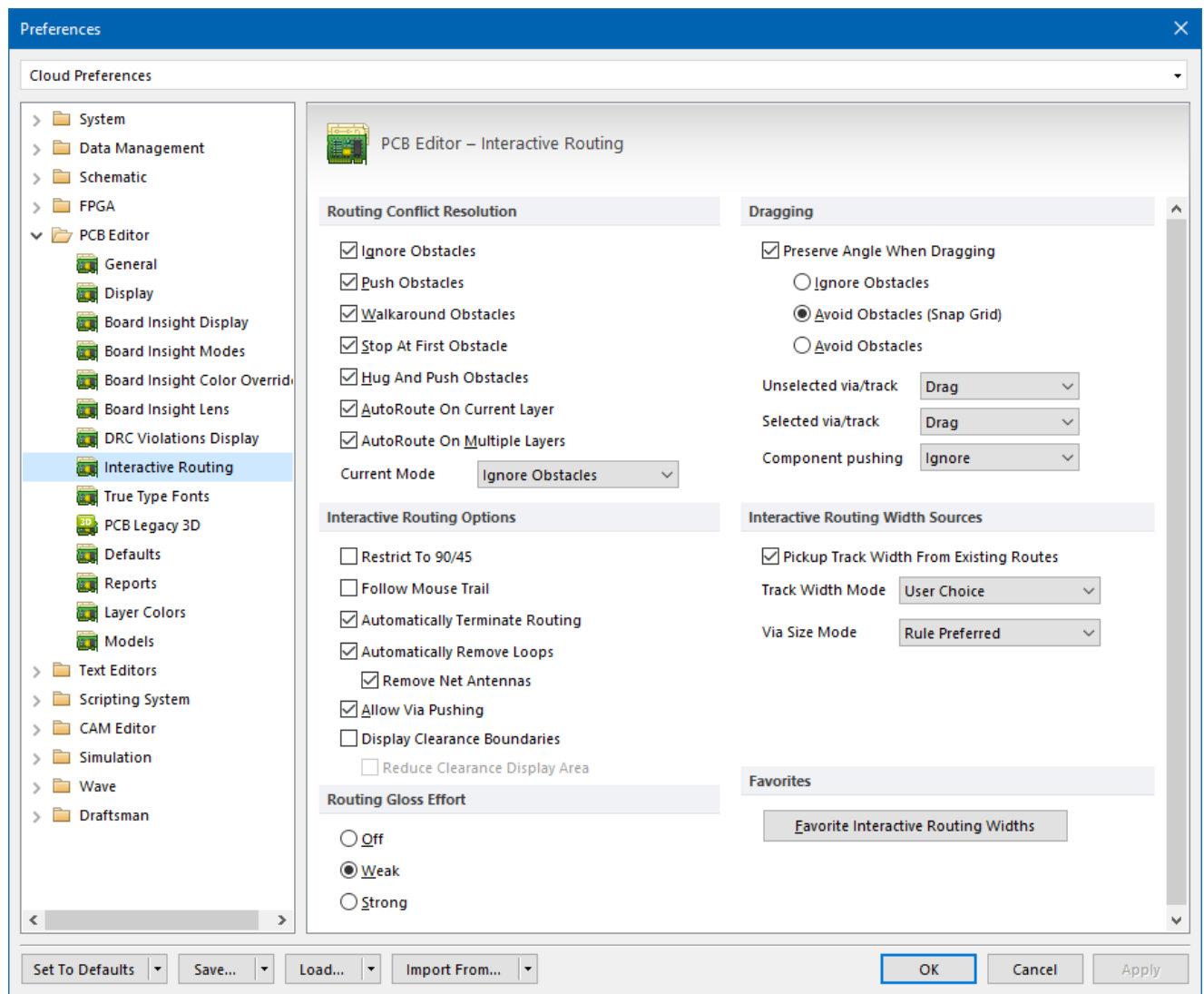
За усіх переваг автоматичного трасування бувають ситуації, коли його недостатньо. У деяких випадках вам буде необхідно повністю контролювати розміщення провідних доріжок на платі. З цією метою ви можете виконати трасування усієї плати або її частини вручну. Засоби інтерактивного трасування **Altium Designer** забезпечують високу ефективність та гнучкість завдяки таким можливостям, як управління курсором під час прокладання трас, розштовхування або огинання перешкод, автоматичне слідування за існуючим з'єднанням відповідно до правил проєктування тощо.


У випадку, коли плата створюється як частина проєкту **Altium Designer**, то зазвичай дані для розміщення електронних компонентів та інформація про з'єднання (nets) між їх виводами передаються у Редактор плат із Редактора схем. Ідея «з'єднань» є вкрай важливою у проєкті. Під час трансферу даних від схеми до плати передається, зокрема, і список з'єднань (netlist), який використовується системою під час ручного трасування для того, щоб: 1) наочно показати, які КМ мають бути електрично з'єднані на платі доріжками під час трасування; 2) уникнути помилкового прокладання доріжок між КМ, які не повинні з'єднуватись між собою. Також список з'єднань відіграє важливу роль під час перевірки правил проєктування.



Змінити налаштування, які будуть використовуватись під час інтерактивного трасування плати, можна у розділі ***Interactive Routing*** (рис. 16), що доступний у розкриттій списку ***PCB Editor*** вікна налаштувань ***Preferences (DXP → Preferences)***.

33

Рис. 16. Розділ *Interactive Routing* вікна *Preferences*

Для того, щоб увімкнути режим інтерактивного трасування плати, необхідно виконати команду **Interactive Routing** із меню **Route** або натиснути відповідну піктограму на панелі інструментів .

Розглянемо важливі поради та рекомендації, які необхідно враховувати під час інтерактивного ручного трасування:

1. Під час прокладання траси незавершений її сегмент відображається у заштрихованому вигляді, тоді як завершений сегмент відображається суцільним кольором, який відповідає кольору поточного шару. Для завершення прокладання сегмента доріжки до поточної позиції курсору необхідно клацнути ЛКМ або натиснути клавішу **<Enter>**.
2. Для того, щоб скасувати останній розміщений сегмент траси, необхідно натиснути клавішу **<Backspace>**.

3. Для того, щоб завершити прокладання поточної доріжки із переходом у режим прокладання нової, необхідно клацнути ПКМ або натиснути клавішу <Esc>.
4. Якщо ви довели трасу до деякої точки в ручному режимі, а потім хочете, щоб програма автоматично завершила з'єднання, то необхідно клацнути ЛКМ, утримуючи клавішу <Ctrl>. Однак ця функція не спрацює за наявності нерозв'язних конфліктів.
5. Натискання комбінації клавіш <Shift + R> під час прокладання траси перемикає режими вирішення конфліктів, пов'язаних з перешкодами на платі, такими як КМ та інші траси: **Push** (розштовхування), **Walkaround** (огинання), **Hug and Push** (розштовхування та огинання) та **Ignore** (Ігнорування).
6. Натискання клавіші <Пробіл> призводить до зміни напрямку вигину траси в усіх режимах, окрім режиму «під будь-яким кутом».
7. Натискання комбінації клавіш <Shift + Пробіл> перемикає різні режими вигину трас. Доступні наступні режими: під будь-яким кутом, 45°, 45° з дугою, 90° і 90° з дугою.
8. Перемальовування екрану в будь-який момент роботи із Редактором плат здійснюється натисканням клавіші <End>.
9. Щоб показати усі об'єкти у межах робочої області, необхідно виконати із меню **View** команду **Fit Board** або послідовно натиснути клавіші <V, F>.
10. Клавішами <Page Up> та <Page Down> здійснюється масштабування відносно позиції курсору. Іншим підходом до зміни масштабу є обертання колеса миші з одночасним утриманням клавіші <Ctrl>.
11. Щоб видалити сегмент траси, необхідно спочатку вибрати його, клацнувши на ньому ЛКМ, після чого натиснути клавішу .
12. Якщо траса прокладається повторно, то усі зайві сегменти видаляються автоматично.
13. Якщо необхідно повністю скасувати поточне трасування, то із меню **Route** виконують команду **Un-Route** → **All**.

Далі розглянемо основні режими інтерактивного ручного трасування, які можна перемикати за допомогою комбінації клавіш <Shift + R> під час прокладання траси:

- **Ignore Obstacle** (ігнорування перешкод) – фактично, ручний режим, у якому не застосовуються раніше створені правила проєктування, тобто траса прокладається так, як мишкою веде проєктувальник (навіть поверх наявних об'єктів);
- **Walkaround Obstacle** (огинання перешкод) – під час прокладання траси конфліктні (зустрічні) елементи топології (як от: інші траси, КМ, перехідні отвори тощо) огинаються з урахуванням мінімальних зазорів і за оптимальною траєкторією;
- **Push Obstacle** (розштовхування перешкод) – під час прокладання траси пріоритет віддається останній, а усі конфліктні елементи топології розштовхуються з урахуванням правил (якщо ж взагалі неможливо вирішити конфліктну ситуацію, то в кінці поточної траси з'являється індикатор блокування);
- **Hug and Push Obstacle** (огинання та розштовхування перешкод) – у цьому режимі в першу чергу програма намагається огинати конфліктний об'єкт, а у випадку неможливості цього зробити – об'єкт відштовхується (якщо цей режим не може обійти або відштовхнути перешкоду, не спричинивши порушення правил, то з'являється індикатор, який показує, що маршрут заблокований);
- **Stop At First Obstacle** (зупинка на першій перешкоді) – у цьому режимі процес трасування зупиняється на першій перешкоді, яка заважає;
- **AutoRoute On Current Layer** (автотрасування на поточному шарі) – увімкнення режиму автотрасування лише на поточному шарі;
- **AutoRoute On Multiple Layers** (автотрасування на декількох шарах) – увімкнення режиму автотрасування на декількох шарах.

7.1.3. Інтерактивне ручне трасування плати, яка не є частиною проєкту

Якщо ви створюєте тільки плату без принципової схеми, то під час її трасування (як ручного, так і автоматичного) можна застосовувати підходи, описані вище. Проте це не так просто і спочатку потрібно виконати певні налаштування.

Найбільшою проблемою є те, що дані про з'єднання КМ вам доведеться вносити вручну. Якщо цього не зробити, то: 1) ви не зможете запустити

автотрасування; 2) у ручному режимі вам доведеться реально «викручуватись», щоб прокласти трасу між двома КМ, між якими немає з'єднання.

У попередньому підпункті описувалось, що поняття «з'єднань» є дуже важливим для процесу трасування. Якщо створюється проєкт, то дані про з'єднання переносяться із принципової схеми. У випадку ж використання тільки **PCB Editor** після розміщення компонентів ніяких з'єднань не буде. Їх потрібно створити вручну.

Зауваження!

На жаль, в **Altium Designer 17.1** Редактор плат не має зручного інструменту для створення з'єднань, наприклад, за допомогою мишки. Це означає, що неможливо встановити мишку в одну точку і провести лінію з'єднання до іншої.

Для створення з'єднань необхідно із меню **Design** виконати команду **Netlist** → **Edit Nets**. Відкриється вікно **Netlist Manager**, представлене на рис. 17.

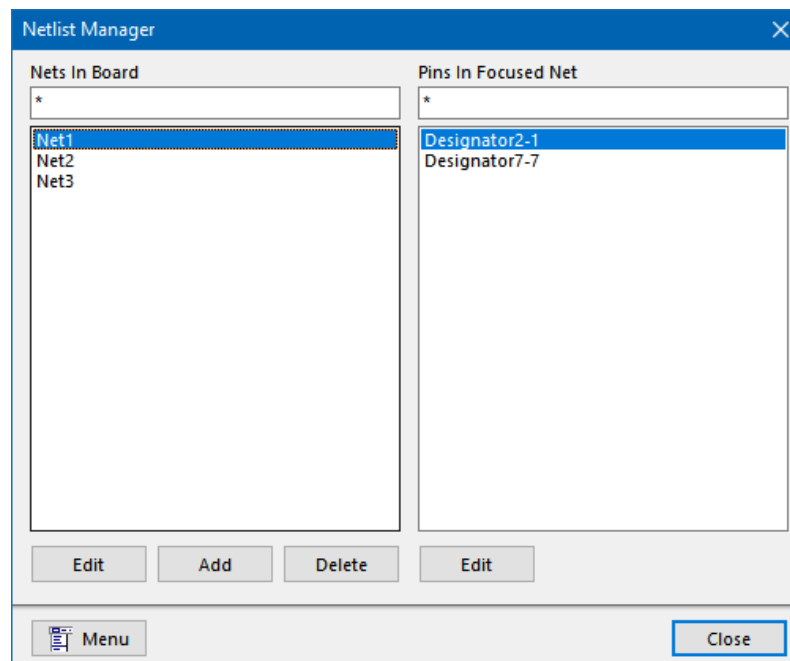

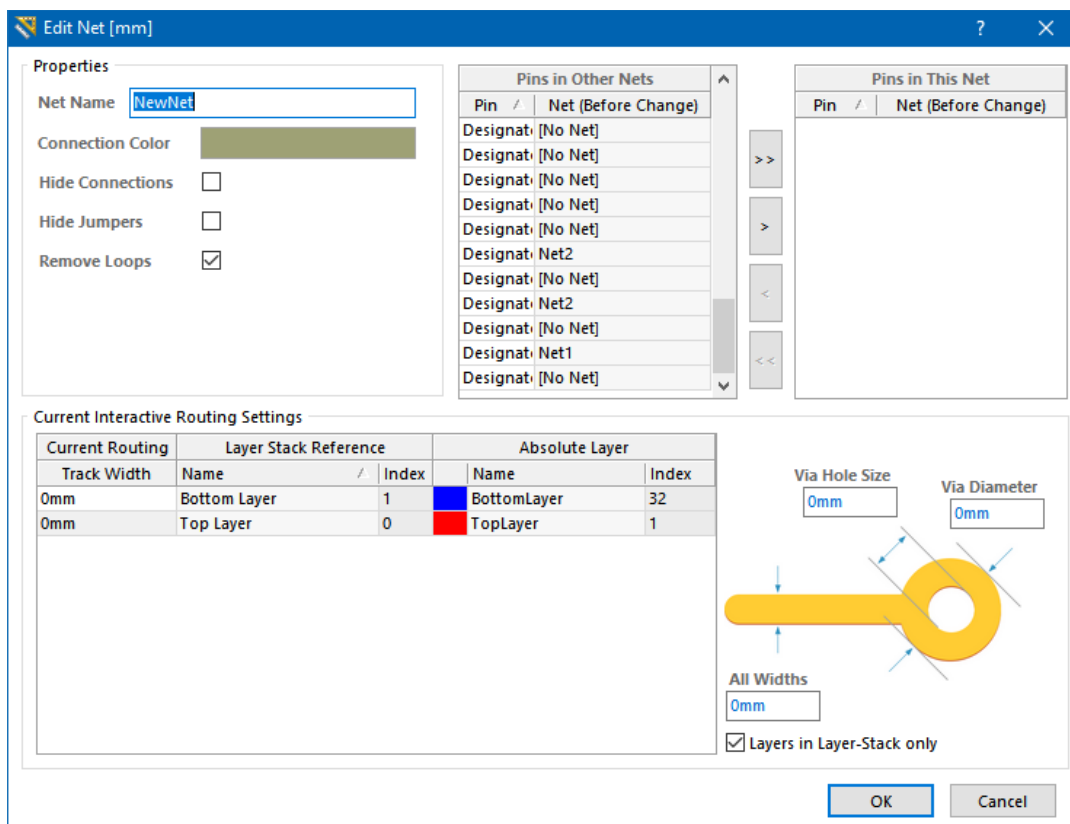


Рис. 17. Вікно **Netlist Manager**

У лівій частині вікна є список вже створених з'єднань, а у правій частині – перелік КМ, що відповідають цьому з'єднанню. Для того, щоб створити нове з'єднання і додати перелік відповідних йому КМ, необхідно натиснути кнопку **Add**. Відкриється нове вікно **Edit Net**, у якому і відбуваються усі необхідні маніпуляції із новим з'єднанням (рис. 18). У полі **Net Name** задається ім'я, а зі списку **Pins in Other Nets** обирається необхідний вивід мікросхеми, який за допомогою

натискання кнопки  переноситься у список *Pins in This Nets*¹⁵. З'єднання має містити мінімум два виводи, але може і більше. Натискання кнопки **ОК** призводить до створення нового з'єднання.

Рис. 18. Вікно *Edit Net*

Описаним вище способом створюються усі необхідні з'єднання на платі. Якщо їх небагато, то все робиться досить швидко, проте зі збільшенням кількості електронних компонентів на платі цей процес стає дуже трудомістким. Причому ще й необхідно контролювати правильність зв'язків та не наробити помилок.

Після створення усіх з'єднань можна запускати автотрасування. Або ж можна виконувати ручне трасування за правилами, описаними у попередньому підзаголовку.

Однак, припустімо, ви, як розробник, хочете виконати ручне трасування і не хочете створювати з'єднання. Ви можете запустити інструмент *Interactive Routing*, після чого почати проводити трасу із одного КМ, проте за замовчуванням ви не

¹⁵ А можна піти й дещо іншим шляхом. Спочатку просто створити набір назв з'єднань, залишивши на цьому етапі їх пустими (фактично, не додаючи КМ). А потім на самій платі відкривати властивості потрібного КМ і у спеціальному полі вікна властивостей обирати із випадного списку, до якого зі створених з'єднань має належати цей КМ.

зможете її з'єднати із іншим КМ. Так відбувається тому, що **Altium Designer** аналізує зв'язаність КМ і не дає з'єднати ті КМ, які не пов'язані між собою. Як можна це обійти? [Існує кілька способів](#):

1. Під час прокладання траси за допомогою комбінації клавіш **<Shift + R>** обрати режим **Ignore Obstacle** (оскільки зазвичай стоїть інший режим за замовчуванням), що дасть вам змогу провести трасу до будь-якого примітиву на платі, зокрема й іншого КМ.
2. Змінити проєктні правила, за якими відбувається перевірка. Для цього необхідно у меню **Design** обрати пункт **Rules**. У лівій частині вікна **PCB Rules and Constraints Editor** зі списку **Electrical** обрати пункт **Clearance**, після чого у правій частині прибрати галочку у стовпчику **Enabled** (рис. 19). Те ж саме виконати для пункту **Short-Circuit**.

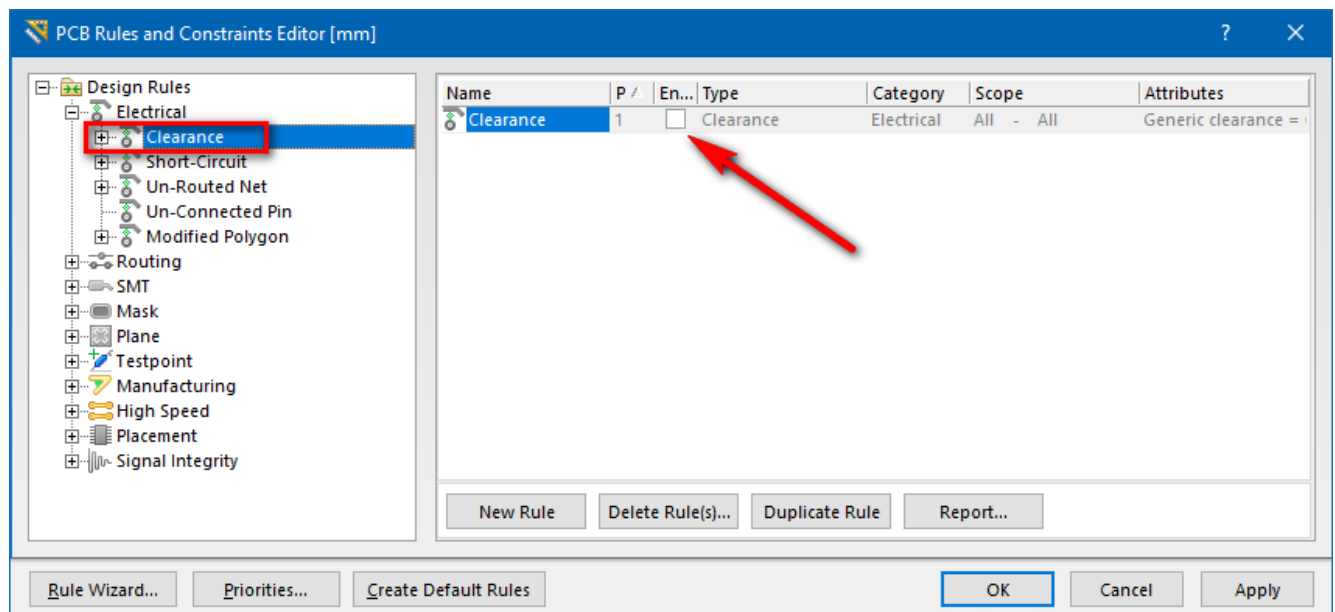


Рис. 19. Вікно **PCB Rules and Constraints Editor**

Важливо!

Так робити, як описано вище, не варто. Вказані підходи пояснені лише для ознайомлення. Краще створюйте проєкти, в яких вам буде доступний повний інструментарій для аналізу, пошуку помилок і зручного трасування.

7.2. Вибір ширини траси

Зазвичай у кожному редакторі ДП задається деяке значення ширини друкованого провідника за замовчуванням. Однак це не означає, що саме таке

значення необхідно використовувати під час проєктування плати. Важливо розуміти, що ширина траси має розраховуватись і задовольняти деяким критеріям.

Зокрема, під час вибору ширини друкованого провідника враховують тип матеріалу, з якого він виготовляється, його товщину, довжину, температуру оточуючого середовища, шар плати, на якому він знаходиться, а також максимальний струм, що протікає в ньому. Також можуть бути враховані й інші параметри, наприклад, мінімальний зазор до інших провідників на платі, який фактично пов'язаний зі щільністю провідників на платі.

Існують різні документи та стандарти, в яких наведені або формули, або графіки, за допомогою яких і визначають ширину доріжки. Також існують різноманітні онлайн-калькулятори, що також дозволяють це зробити. Розробник має узгодити із замовником, які стандарти розробки плати будуть використовуватись, після чого вже визначатись із керівним документом, із якого і будуть братись дані для розрахунків¹⁶.

Одним із поширених стандартів, який можна використовувати під час розробки ДП, є **IPC-2221A «Generic Standard on Printed Board Design»**. У цьому стандарті, зокрема, наведені графіки та формули для необхідних розрахунків ширини доріжки. На основі цього стандарту створено калькулятор розрахунку ширини доріжки у програмі **Electrodoc**, яка доступна на пристроях **Android** та **iOS**.

Для того, щоб керувати налаштуваннями ширин доріжок, необхідно спочатку відкрити вікно **Preferences (DXP → Preferences)**, після чого у розкритому списку розділу **PCB Editor** обрати пункт **Interactive Routing** (див. рис. 16 у розділі [7.1.2](#)). В області **Interactive Routing Width Sources** можна налаштувати джерела даних, із яких буде браться поточна ширина доріжки під час трасування (а також тут можна зробити те саме і для перехідного отвору). А в області **Favorites** можна налаштувати перелік типорозмірів ширин доріжок, серед якого розробник зможе «на льоту» обирати необхідну ширину під час інтерактивного ручного трасування завдяки натисканню комбінації клавіш **<Shift + W>**¹⁷.

Також ширини доріжок регламентуються правилами проєкту, про що буде докладніше описано в одному із комп'ютерних практикумів.

¹⁶ Перелік основних стандартів, що регламентують розробку ДП, можете подивитись у [Додатку Б](#).

¹⁷ Однак потрібно пам'ятати, що якщо обрати із цього списку значення, яке виходить за межі ширин, налаштованих правилами проєктування, то відбудеться обмеження до нижнього або верхнього діапазону відповідно.

7.3. Створення областей металізації

Під час виготовлення плат часто використовують області металізації або області, що заповнені міддю (copper pour). Термін «мідь» застосовується тому, що зазвичай саме цей метал використовується на платах для створення провідних доріжок. Области металізації бувають суцільними та сітчастими.

Суцільні області металізації на платі, як правило, приєднуються до ланцюга землі GND і використовуються для екранування ланцюгів. Однак можуть підключатись і до інших ланцюгів (наприклад, потужних джерел живлення), якщо через ці ланцюги протікатиме великий струм. Сітчасті ж області використовуються в основному для з'єднання з землею на платах з аналоговими електронними компонентами.

Додатковим наслідком використання областей металізації є забезпечення тепловідведення та мінімізація кількості витравлюваної міді, а, отже, зменшення кількості травильної рідини, яка використовується під час виробництва.

Відмітною особливістю області металізації є відступ (backoff або stand-off) – певна відстань між областю металізації та будь-якими доріжками або КМ. В цьому проміжку відсутня мідна заливка. Тому область металізації виглядає так, ніби вона обтікає інші компоненти, проте містить невеликі теплові з'єднання із тими КМ, які належать до одного із нею ланцюга (рис. 20).

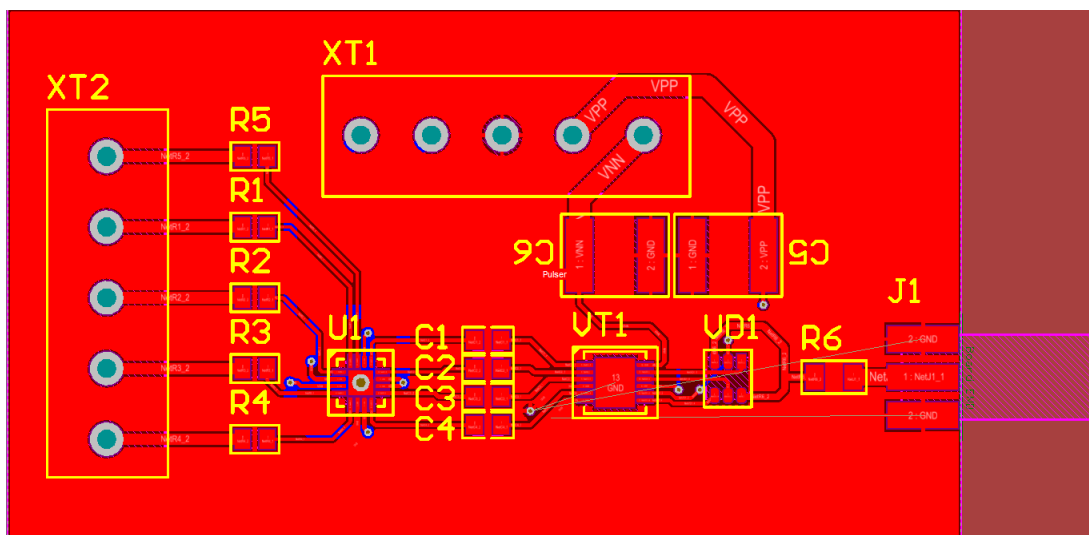


Рис. 20. Приклад верхнього шару плати з областю металізації

Для створення області металізації використовується команда *Polygon Pour* із меню *Place*.

Якщо внаслідок внесених у проєкт змін необхідно сформуванати полігони заново, то виконують команду ***Tools*** → ***Polygon Pour*** → ***Repour All***.

7.4. Робота із шарами плати

Шар (англ. layer) ДП – це окрема сторона пластини діелектрика із нанесеним на неї провідним малюнком.

Багат шарова ДП є набором пластин діелектрика із нанесеним на кожну пластину власним провідним малюнком, які виробник компонує в єдине ціле за допомогою виконання послідовності різних хімічних і механічних технологічних операцій.

Шари ДП є **конструктивними** (фізично реалізованими) **шарами** плати як готового виробу. Зазвичай у конструкції ДП виділяють сигнальні, екранувальні шари та окремо шар шовкографії¹⁸.

У Редакторі плат поняття «шар» є значно ширшим, ніж те, яке використовується під час виготовлення ДП, і включає у себе не тільки конструктивні шари плати, але й безліч додаткових шарів, які виконують допоміжні, технологічні, проєктні чи інші функції. Фактично, це не «шари друкованої плати», а «шари проєкту друкованої плати». Цих шарів завжди більше, ніж конструктивних.

У проєкті ДП шари діляться на:

- електричні (сигнальні і живлення) шари;
- неелектричні (механічні і спеціальні) шари.

Розглянемо докладніше призначення кожного із типів шарів у Редакторі плат.

Електричні шари використовуються для створення провідних доріжок та полігонів металізації.

Неелектричні механічні шари використовуються для зберігання та відображення розмірів, заводських позначень, інструкцій зі складання, тривимірних моделей корпусів електронних компонентів тощо. Ці шари можуть бути вибірково включені у вихідні дані для друку і у Gerber-файли. Крім того, можна призначати пари шарів: у цьому випадку об'єкт, що був розміщений на одному із верхніх парних шарів у Редакторі посадкових місць, буде перенесений на нижній парний шар, якщо компонент розмістити на нижній стороні плати.

¹⁸ Шовкографія – це процес нанесення фарби на плату шляхом продавлювання її через сітчастий шаблон (найчастіше це шовкова сітка з малюнком). Малюнок трафарету на шовку формується фотоекспонуванням. Фотошаблон для шовкографії, по суті, мало чим відрізняється від фотошаблонів, які застосовуються в інших фотохімічних процесах на платі.

До неелектричних спеціальних шарів можна віднести верхні та нижні шари шовкографії, шари паяльних паст і захисних масок, шари свердління (Drill Drawing, Drill Guide), шари з зонами заборони трасування (Keep-Out), шари для розміщення КМ із отворами і перехідних отворів багатошарових ДП (Multi-Layer), шари електричних з'єднань, шари маркерів помилок, шари сіток, шари отворів та інші.

Нижче у табл. 1 наведено перелік шарів, який за замовчуванням доступний у Редакторі плат та Редакторі посадкових міць, із коротким описом та рекомендаціями із використання.

Таблиця 1. Перелік та короткий опис шарів плати у проєкті **Altium Designer**

Назва шару	Призначення	Примітка
Шари топології та розміщення КМ компонентів		
Top Layer	Сигнальний шар на верхній стороні плати. Розміщення КМ, доріжок, полігонів, зон заборони розміщення топології тощо	Має парний шар ¹⁹ Bottom Layer
Bottom Layer	Сигнальний шар на нижній стороні плати. Розміщення КМ, доріжок, полігонів, зон заборони розміщення топології тощо	Має парний шар Top Layer
Internal Layer	Сигнальний внутрішній шар плати. Розміщення доріжок	Використовується у проєктах багатошарових плат
Plane Layer	Шар типу Plane. Зазвичай використовуються для розміщення полігонів землі або живлення на внутрішніх шарах плати	Використовується у проєктах багатошарових плат
Шари графіки захисного покриття та нанесення пасти для КМ		
Top Paste	Верхній шар пасти	Має парний шар Bottom Paste
Bottom Paste	Нижній шар пасти	Має парний шар Top Paste
Top Solder	Верхній шар маски	Має парний шар Bottom Solder
Bottom Solder	Нижній шар маски	Має парний шар Top Solder

¹⁹ Парними називаються шари, графіка на яких взаємно змінюється під час перенесення ПМ електронного компонента на протилежний бік ДП. Такий підхід є дуже зручним, оскільки автоматизує процес перенесення даних і позбавляє розробника головного болю, якби він це робив вручну. Зазвичай у Редакторі посадкових місць практично усі елементи ПМ створюються на верхніх шарах **Top**. Якщо таке ПМ згодом розмістити на верхній стороні плати, то все співпадає. Але що робити, якщо це ПМ потрібно перенести на нижню сторону плати? Створювати його заново у Редакторі посадкових місць, тільки використовуючи шар **Bottom**? Або вручну змінювати на платі для усіх примітивів, з яких складається ПМ, їх приналежність до шару **Bottom**? Зрозуміло, що це не оптимально. Саме у таких випадках у нагоді стають парні шари.

Продовження табл. 1

Назва шару	Призначення	Примітка
Шари шовкографії		
Top Overlay	Верхній шар для написів (зокрема, позиційного позначення, номіналу) та графіки компонента (зокрема, лінії умовного корпусу електронного компонента)	Має парний шар Bottom Overlay
Bottom Overlay	Нижній шар для написів та графіки компонента	Має парний шар Top Overlay
Інші спеціальні шари		
Drill Guide	Шар, що містить інформацію про центри отворів	Зазвичай ці шари використовуються тільки у Редакторі плат, тоді як у Редакторі посадкових місць – ні
Drill Drawing	Шар, що містить графіку типів отворів	
Keep-Out Layer	Шар для вказівки зон заборони розміщення компонентів та топології	Його дія поширюється на усі шари ДП
Multi-Layer	Віртуальний шар для вказівки того, що об'єкти, які його використовують, належать до усіх зовнішніх і внутрішніх сигнальних шарів	Наприклад, цей шар використовується для КМ, що мають наскрізні отвори, та для перехідних отворів
Механічні шари		
Mechanical 1 - Mechanical 32	Механічні шари використовуються розробником на власний розсуд для збереження та відтворення необхідної графічної, текстової, конструкторської та іншої інформації	Наприклад, шари можуть містити тривимірні моделі корпусів електронних компонентів, графіку контуру плати, вирізів на платі, варіантів креслеників ДП тощо

Зауваження!

Абсолютний мінімум шарів, який необхідно буде використовувати у проєкті найпростішої односторонньої ДП, що містить тільки компоненти поверхневого монтажу і може бути виготовлена в домашніх умовах, – це Top Layer (або Bottom Layer). Зі зростанням складності плати, урізноманітненням номенклатури електронних компонентів та необхідності виготовлення її на автоматизованих виробництвах кількість використовуваних шарів зростатиме.

Розглянемо докладніше особливості використання та специфіку деяких шарів плати, що використовуються в **Altium Designer**.

У Редакторі посадкових місць усі елементи та графіка зазвичай створюються на шарах типу **Top**. У Редакторі плат завдяки властивостям парності ПМ можуть бути без проблем розміщені як на верхній стороні плати (**Top**), так і на нижній (**Bottom**).

Шари типу **Solder** містять апертури паяльної (або захисної) маски, яка використовується для захисту ДП від механічних впливів та оточуючого середовища. Якщо на цих шарах намальовано об'єкт, то в паяльній масці з'явиться виріз, що відповідає контуру цього об'єкта²⁰. Тому цей шар ще називають інверсним, оскільки наявність графічного об'єкту на ньому свідчить не про наявність у цьому місці паяльної маски, а про її відсутність.

Шари типу **Paste** використовуються для позначення місць нанесення паяльної пасти для електронних компонентів, що монтуються на поверхню. Під час виготовлення самої ДП цей шар не використовується. Проте він може використовуватись пізніше для створення спеціальних трафаретів, за допомогою яких наноситься паяльна паста.

У шарах типу **Plane** завжди відображається інверсний малюнок провідного покриття на шарі. Тобто, якщо ви на такому шарі розмістите деякий об'єкт (наприклад, лінію, дугу, полігон тощо), то в цій області провідне покриття буде відсутнє у реальній виготовленій ДП. До цього типу шарів належать **Power Plane** та **Ground Plane**.

Зазвичай шари **Internal Layer** та **Plane Layer** не використовуються у Редакторі посадкових місць (за винятком компонентів, які встановлюються між внутрішніми шарами), а тільки у проєктах багатошарових плат у Редакторі плат.

На шарі **Overlay** розміщують написи та графіку компонента. Тут важливим є наступне правило: графіка на цьому шарі не повинна перетинатись (накладатись) із КМ. Також на цьому шарі зазвичай створюється вказівник на перший контакт електронного компонента.

Шар **Keep-Out Layer** використовується для розміщення графіки зон заборони топології для усіх шарів. Майстер створення ДП, який доступний у Редакторі плат, використовує саме цей шар для того, щоб розмістити контури ДП²¹.

²⁰ Якщо покрити паяльною маскою усю плату, то будуть також покриті КМ, до яких потрібно припаювати виводи електронних компонентів. А це унеможливить процес припаювання. Тому зазвичай вирізи у паяльній масці потрібні саме для КМ, щоб до них можна було припаяти виводи мікросхеми. Проте [для КМ в Altium Designer такі області регламентуються налаштуваннями примітиву Pad та правилами](#). Тому використання додаткових примітивів на шарах типу **Solder** не є обов'язковим (до речі, те саме стосується і шару типу **Paste**), а обумовлюється конкретними потребами, наприклад, необхідністю відкриття захисної маски на полігонах з одночасним нанесенням паяльної пасти.

²¹ Використання саме цього шару для розміщення ліній контуру не є обов'язковим. Як буде показано у комп'ютерних практикумах, лінії контуру, які використовуватимуться на виробництві

Як було зазначено вище, деякі шари за замовчуванням складають пару. Однак інколи необхідно створити користувацьку пару шарів. *Створювати парні шари необхідно безпосередньо під час налаштування властивостей шарів проєкту плати.* Для налаштування парних шарів необхідно у Редакторі плат із меню **Design** обрати команду **Board Layers & Color**, а потім у лівому нижньому куті нового вікна **View Configurations**, що відкриється, натиснути кнопку **Layer Pair** (рис. 21).

Зауваження!

У Редакторі посадкових місць користувацькі пари шарів не налаштовуються, оскільки ці дані не передаються у проєкт плати під час розміщення компонентів.

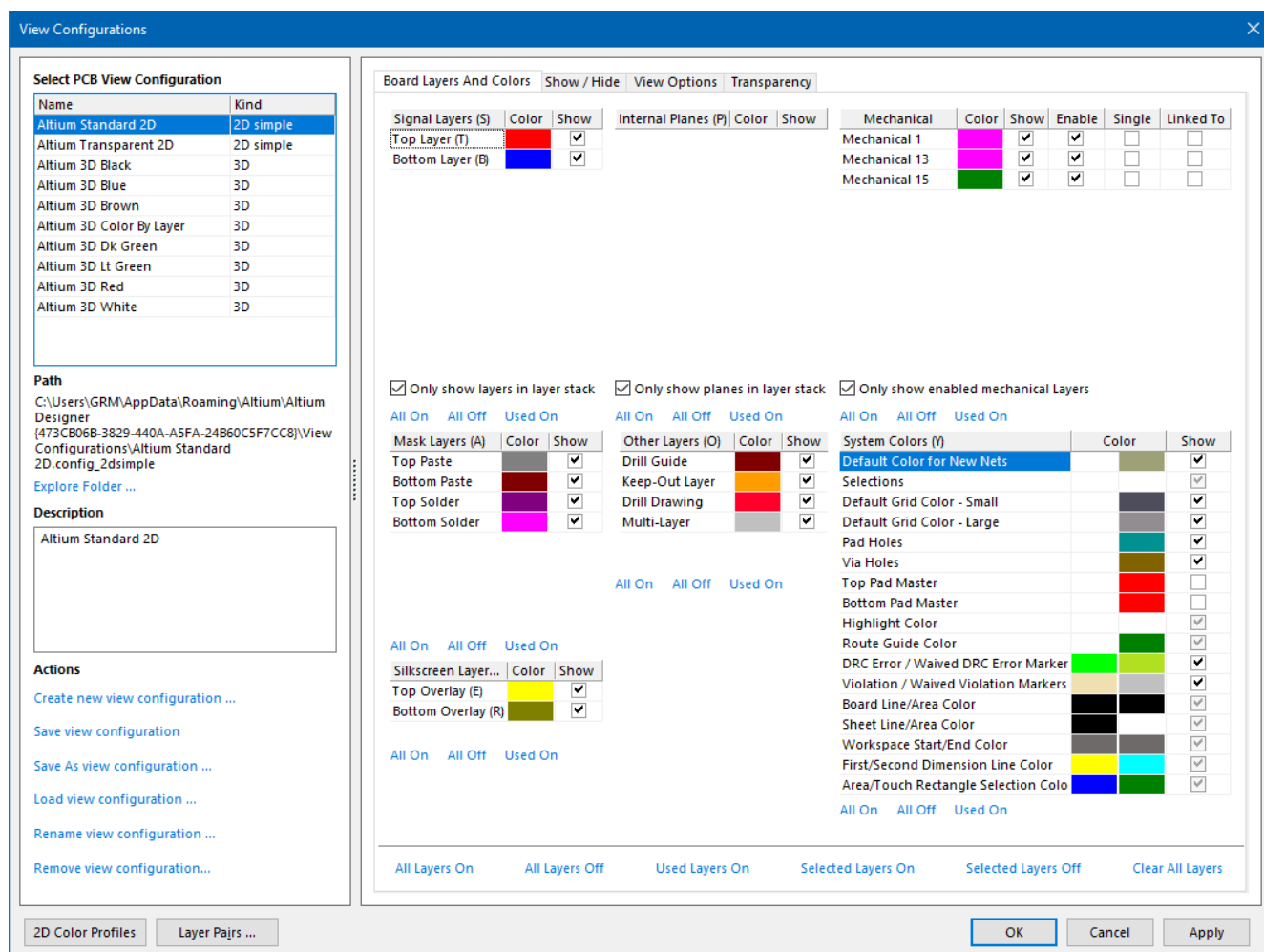


Рис. 21. Вікно **View Configurations**

під час безпосереднього виготовлення самої плати, можуть бути розміщені на будь-якому механічному шарі.

У цьому ж вікні в області **Select PCB View Configuration** можна обирати одну з доступних конфігурацій відображення шарів плати (ця інформація дублюється у вигляді випадного списку на панелі інструментів Редактора плат). Конфігурація відображення являє собою набір параметрів, що керують зовнішнім виглядом робочого простору в 2D і 3D режимі. Конфігурація, яка використовувалася під час збереження документа, також зберігається разом з файлом. Таким чином, при відкритті файлу під час наступного запуску програми **Altium Designer** буде використана пов'язана з цим файлом конфігурація відображення. Якщо документу не привласнена жодна конфігурація відображення, то під час його відкриття використовується конфігурація за замовчуванням.

До речі, користувач може створити і зберегти власну конфігурацію відображення, для чого використовуються відповідні елементи в області **Actions** все того ж вікна **View Configurations** (див. рис. 21).

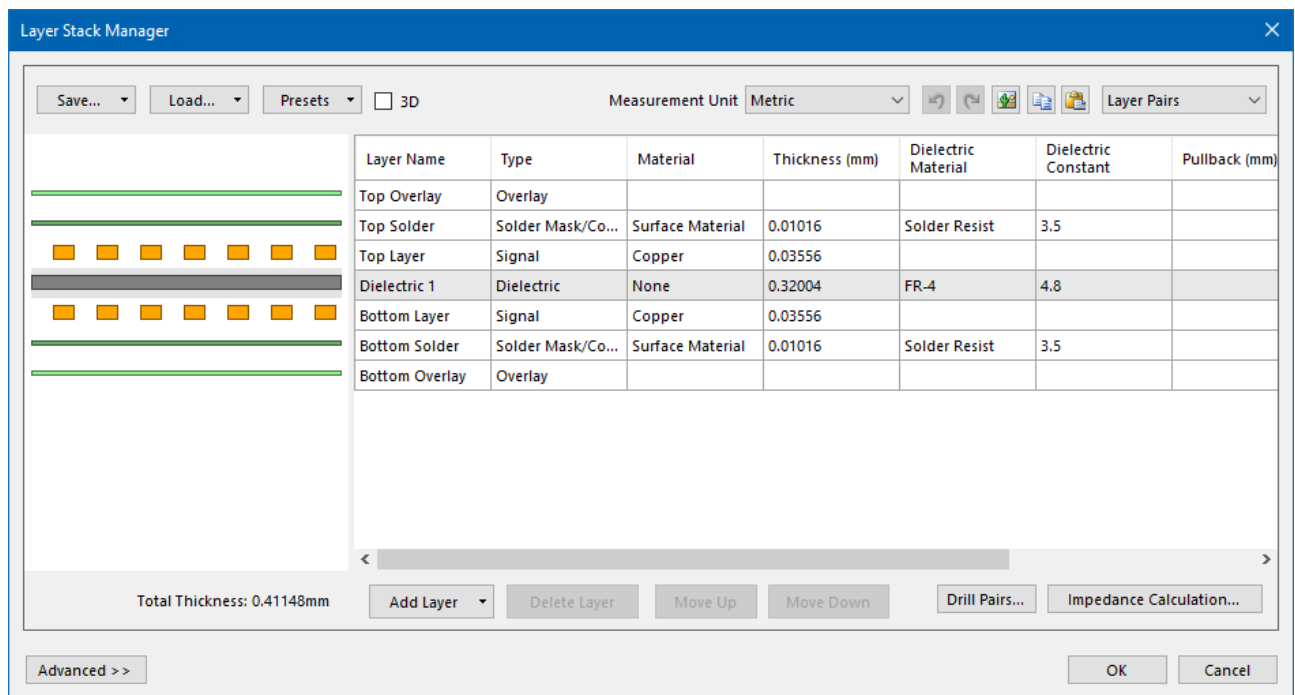
Окрім налаштування конфігурації відображення, у вікні **View Configurations** можна також вказувати видимість (або прихованість) шарів, задавати їх кольори, а також встановлювати системні параметри кольору, які будуть застосовуватися до всіх документів, а не тільки до поточної конфігурації відображення.

7.5. Налаштування структури плати

Найпростіша двостороння плата складається із діелектричної основи, на верхній та нижній стороні якої містяться шари провідного матеріалу, яким зазвичай є мідь. Ці шари прийнято називати сигнальними, оскільки саме на них формуються провідники, які з'єднують виводи електронних компонентів.

Проте плати бувають не тільки одно- або двосторонніми, але й можуть бути багатошаровими. Для того, щоб налаштувати фізичну структуру плати (зокрема, кількість внутрішніх шарів), використовують вікно **Layer Stack Manager** (рис. 22), яке викликається із меню **Design** вибором пункту **Layer Stack Manager** (послідовність клавіш <D, K>).

У вікні **Layer Stack Manager** можна вказати товщину шару, тип матеріалу, деякі фізико-механічні характеристики (зокрема, діелектричні властивості). Для того, щоб додати новий шар, використовується кнопка **Add Layer**. Нові шари вставляються після виділеного (або обраного) у даний момент шару. Переміщувати шари можна, використовуючи кнопки **Move Up** та **Move Down**.

Рис. 22. Вікно *Layer Stack Manager*

7.6. Створення контуру плати

Для того, щоб створити контур ДП, в **Altium Designer** використовується декілька підходів, які будуть розглянуті далі.

7.6.1. Створення контуру плати вручну

Для того, щоб створити контур плати вручну, спочатку необхідно обрати один із механічних шарів, який буде використовуватись тільки для збереження контуру плати. Номер механічного шару обирається на власний розсуд, проте до нього висувається наступна умова: на цьому шарі не має бути ніяких інших графічних примітивів, окрім ліній контуру плати. Після вибору необхідного механічного шару на ньому за допомогою ліній (line) та дуг (arc) створюють заданий контур.

Важливо!

Контур плати обов'язково має бути замкненим.

Малювати контур можна як безпосередню ведучи лінію за допомогою курсору мишки, так і використовуючи режим «стрибків». Перший варіант є швидким, але не завжди зручним, оскільки буває складно позиціонувати курсор в необхідну

точку. Однак під час малювання лінії її координати можна вводити з клавіатури. Для цього, перебуваючи у режимі малювання, необхідно натиснути клавішу <J>, після чого у меню, що з'явиться, обрати пункт *New Location*. Далі відкриється вікно *Jump To Location*, у якому задаються координати наступної точки, до якої має бути проведена лінія. Після введення координат та закриття вікна *Jump To Location* завершити формування контуру можна натисканням ЛКМ.

Після того, як намальовані усі лінії контуру, необхідно виконати команду *Design* → *Board Shape* → *Define from selected objects*. Ця дія призведе до створення контуру плати.

7.6.2. Створення контуру плати за допомогою інструменту *PCB Board Wizard* з використанням шаблону

Спеціальний інструмент *PCB Board Wizard* дає змогу обрати стандартний промисловий типорозмір плати або ж створити свій власний. Майстер створення контуру плати *PCB Board Wizard* розташований в області *New from Template* нижньої частини лівої бічної панелі *Files* (рис. 23).

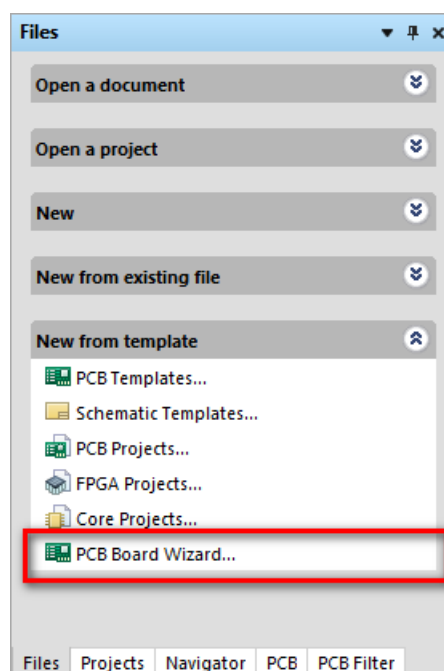


Рис. 23. Панель *Files*

Після запуску майстра у початковому вікні потрібно натиснути *Next*, у наступному вікні вибрати необхідну систему одиниць вимірювань (наприклад, *metric*) і також натиснути *Next*. Наступне вікно називатиметься *Choose Board Profiles*, у лівій його частині присутній список шаблонів плат (рис. 24).

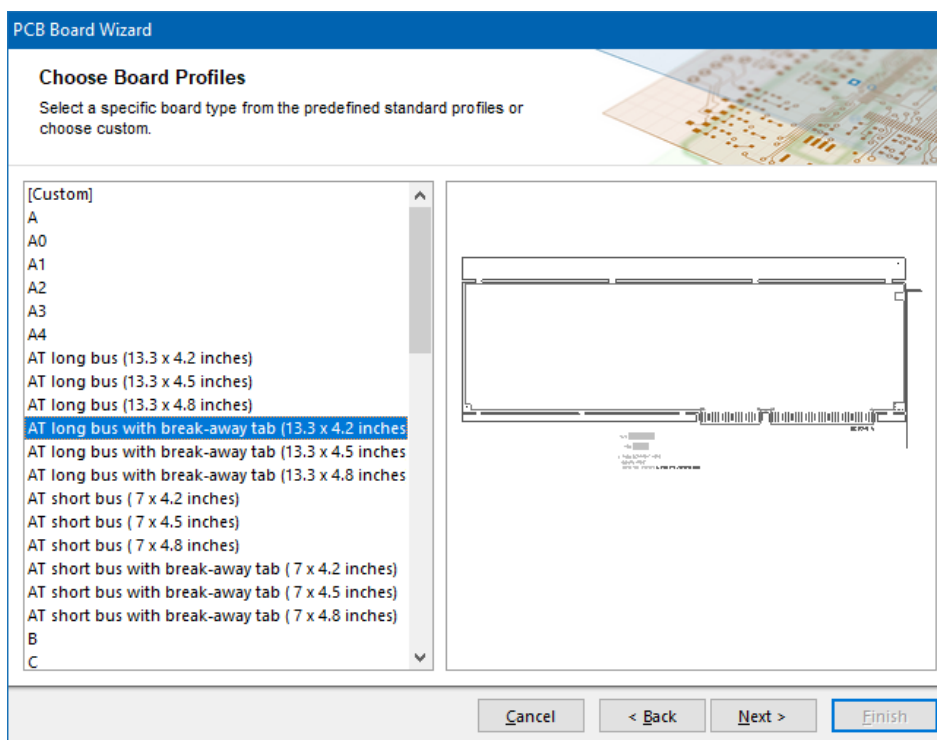


Рис. 24. Вікно вибору профіля плати

Із наведеного переліку шаблонів можна обрати потрібний, хоча варто зауважити, що їх кількість невелика.

Далі, натискаючи послідовно кнопку **Next**, з'являтимуться нові вікна із можливістю налаштувати додаткові параметри плати. Насамкінець, після усіх налаштувань, необхідно натиснути кнопку **Finish**.

7.6.3. Створення контуру плати за допомогою інструменту *PCB Board Wizard* без використання шаблону

Процедура запуску майстра аналогічна попередній. Однак тепер у вікні **Choose Board Profiles** замість шаблону необхідно обрати параметр **Custom** (див. рис. 24). Якщо він обраний, то на наступному кроці з'явиться нове графічне вікно²² із можливістю налаштувати геометричні параметри контуру плати (рис. 25): форма, розміри, шар розмірних ліній, ширина лінії, що обмежує контур плати на шарі **Keep-Out**, ширина розмірних ліній, зазор від краю плати для забороненої зони і додаткові опції.

²² Це вікно не з'являється, коли використовується шаблон, оскільки для шаблону задані усі геометричні параметри плати.

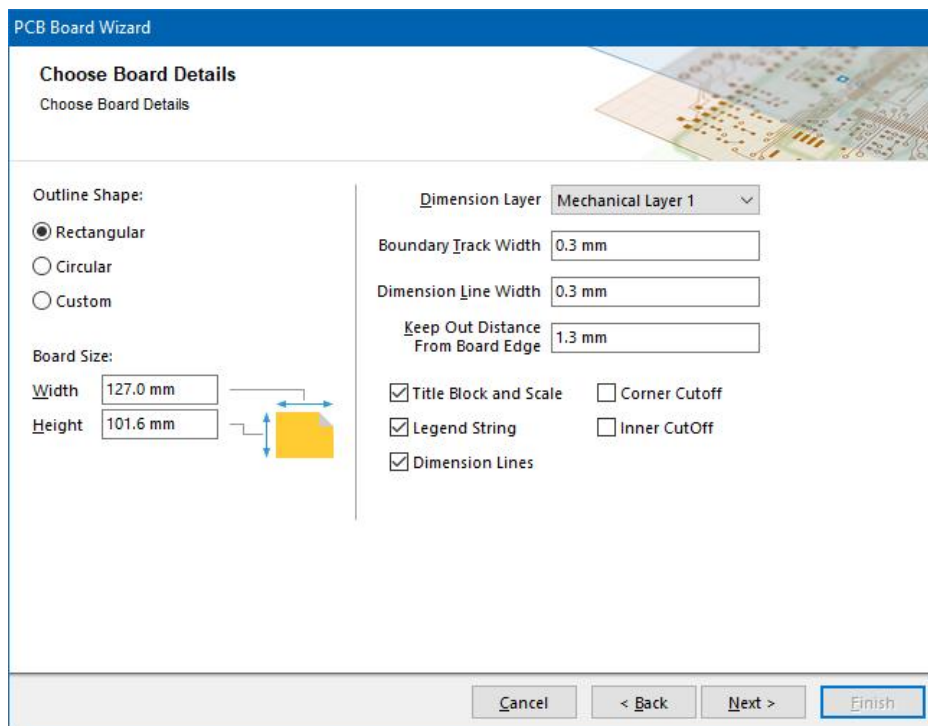


Рис. 25. Налаштування геометричних параметрів контуру плати

Якщо увімкнені опції **Corner Cutoff** та **Inner Cutoff**, то на наступних кроках буде запропоновано задати розміри вирізів²³.

Спочатку задаються зовнішні прямокутні вирізи у платі (рис. 26).

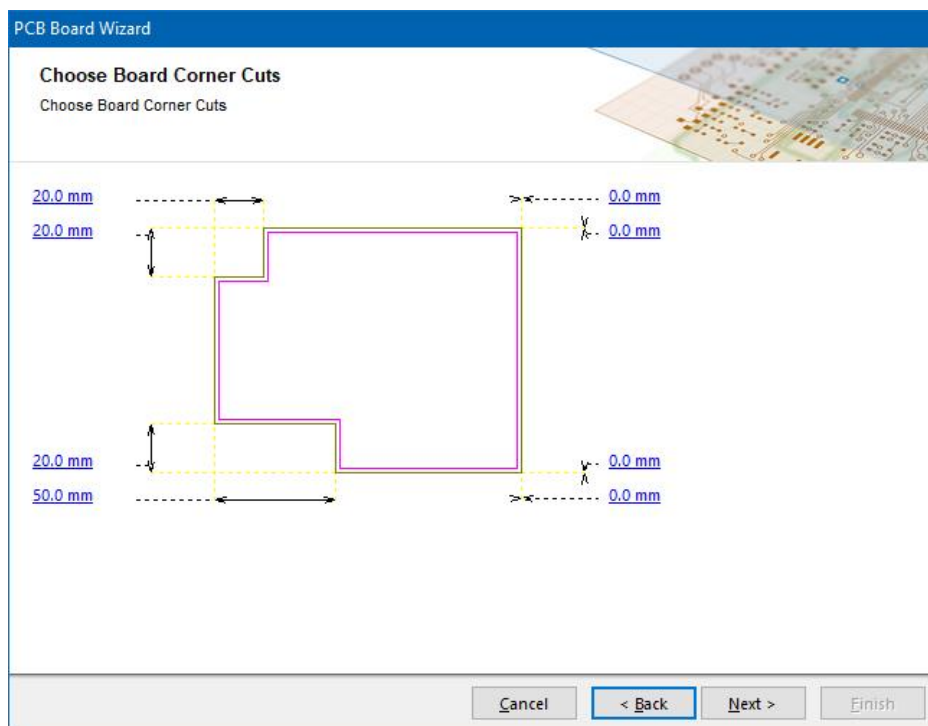


Рис. 26. Вікно налаштування зовнішніх вирізів

²³ Якщо ці опції не обрані, то на наступних кроках графічних вікон із можливістю налаштування вирізів не буде.

Потім задається внутрішній виріз за допомогою координат, причому внутрішній виріз може бути тільки один і тільки прямокутної форми (рис. 27). Згодом можна коригувати вирізи стандартними інструментальними засобами **Altium Designer**.

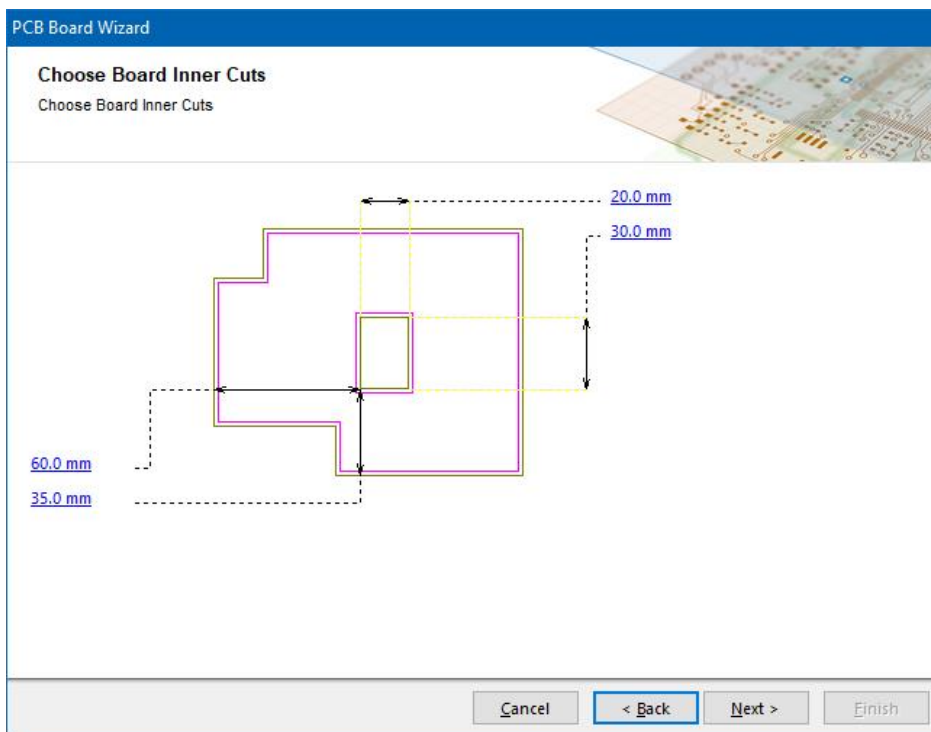


Рис. 27. Вікно налаштування внутрішніх вирізів

7.6.4. Імпортування контуру плати із формату DWG/DXF

Altium Designer 17.1 підтримує можливість імпортування двовимірної моделі плати, що збережена у форматі **DXF** або **DWG** (зазвичай створюються в САПР **AutoCAD**). Для того, щоб імпортувати дані, необхідно, перебуваючи у Редакторі плат, у меню **File** обрати пункт **Import**, після чого відкриється список підтримуваних форматів (рис. 28).

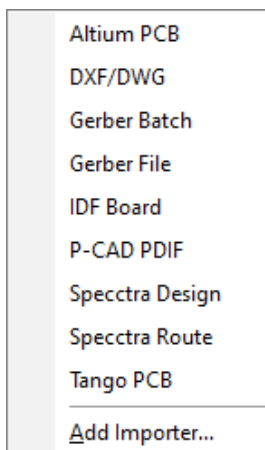


Рис. 28. Формати, із яких можна імпортувати дані в **Altium Designer**

Якщо обрати із цього списку рядок DWG/DXF, то відкриється стандартне вікно пошуку файлу на диску. Після вибору та завантаження необхідного файлу із контуром плати²⁴ з'явиться нове вікно, у якому виконуються деякі налаштування: обираються одиниці вимірювання, механічний шар, на який буде перенесена інформація, що імпортується, товщина ліній контуру, початок координат тощо (рис. 29). Якщо у самому файлі присутні кілька шарів графіки, то для невикористовуваних шарів необхідно встановити опцію *Not Imported* у стовпчику *PCB Layer*.

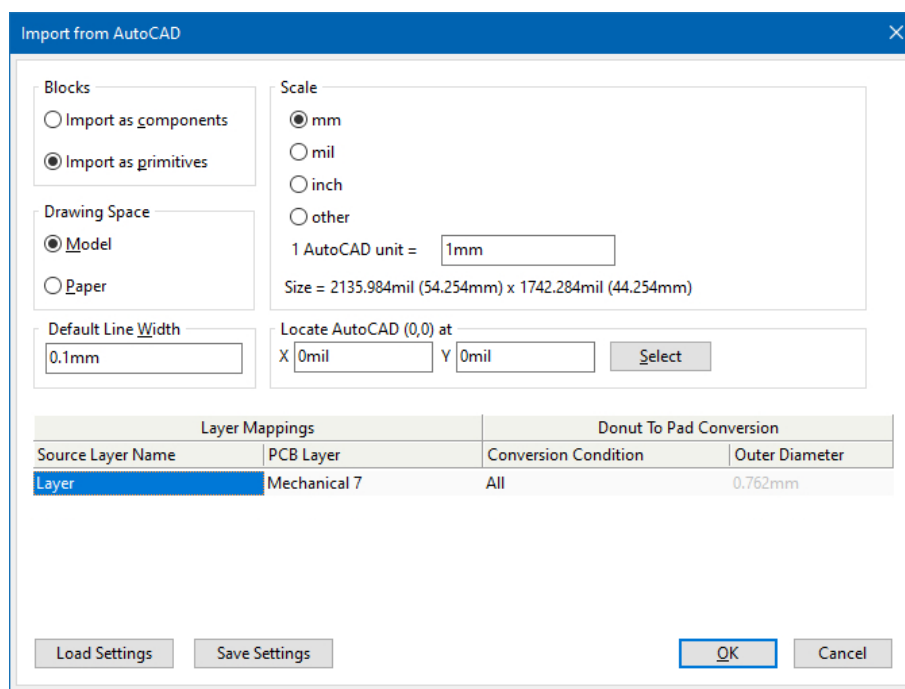


Рис. 29. Вікно *Import from AutoCAD*

Після встановлення усіх опцій і натискання кнопки **ОК**, у робочій області Редактора плат на обраному механічному шарі з'явиться імпортований контур. Однак це ще не завершення процесу. Далі програмі необхідно вказати, що цей контур є межами плати. Для цього виділяють увесь контур і виконують команду *Design* → *Board Shape* → *Define from selected objects*.

7.6.5. Імпортування контуру плати із формату STEP

Окрім розглянутих вище способів сформувати контур плати можна і за допомогою імпортування даних із формату **STEP**, який використовується для збереження тривимірних моделей плати.

²⁴ Вважається, що контур плати має бути намальованим на одному графічному шарі та обов'язково має бути замкненим.

Спочатку необхідно вказати шлях до моделі, для чого у вікні **Preferences** (див. рис. 16) у розділі **PCB Editor** обирають пункт **Models**. За допомогою елементів керування графічного інтерфейсу вказують шлях до розташування моделей.

Тепер, щоб підключити модель плати у форматі STEP, потрібно в Редакторі плат перейти у тривимірний режим відображення (гаряча клавіша <3>). Після цього модель плати підключається²⁵ за допомогою команди **Place** → **3D Body**. У вікні, що відкриється (рис. 30), обирають опцію **Generic 3D Model**, вказують механічний шар для розміщення контуру, а також в області **Generic 3D Model** встановлюють перемикач **File** та за допомогою кнопки **Select** обирають необхідну модель плати для підключення.

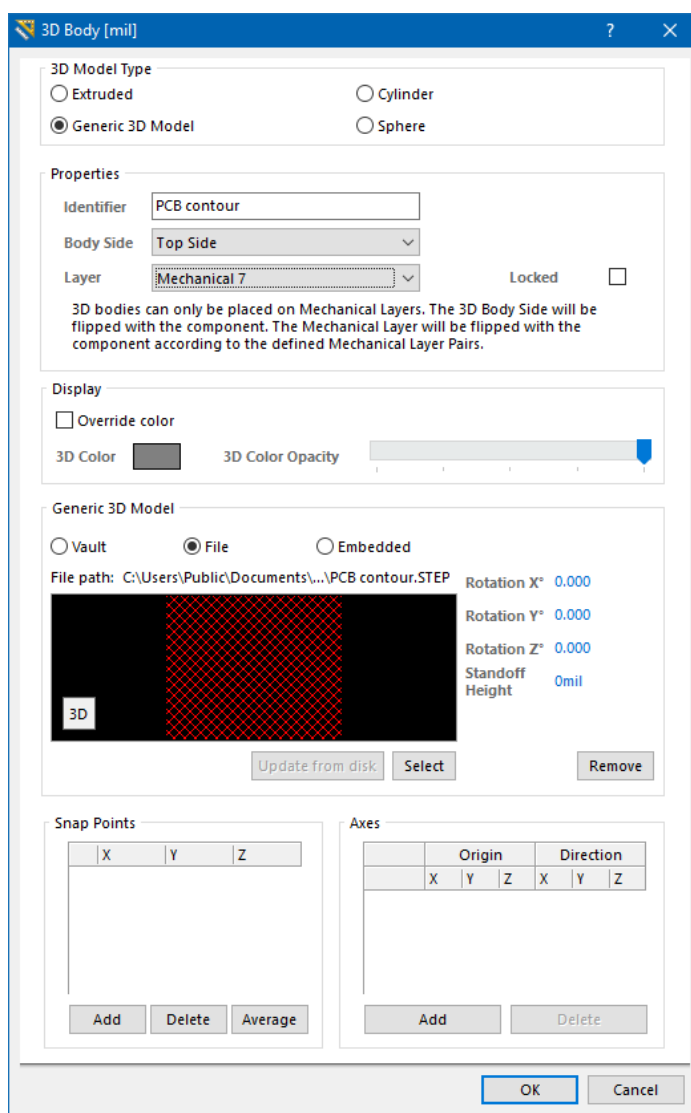


Рис. 30. Вікно **3D Body** із обраною опцією **Generic 3D Model**

²⁵ Зазвичай можливість відображення тривимірних моделей в **Altium Designer** увімкнена за замовчуванням, проте інколи потрібно зробити деякі початкові налаштування.

Для того, щоб розмістити обрану 3D модель плати, після виконання усіх налаштувань необхідно натиснути кнопку **ОК** та завершити розміщення натисканням ЛКМ. Після розміщення моделі програма пропонує встановити наступну модель, однак у нашому випадку слід відмовитися від цього натисканням кнопки **Cancel**. Для того, щоб вказати, що додана тривимірна модель визначає саме контур плати, необхідно виконати команду *Design* → *Board Shape* → *Define from 3D Body* і послідовно двічі клацнути ЛКМ на доданій моделі. Врешті буде відображене повідомлення, в якому пропонується створити контур плати на основі обраної моделі, з чим слід погодитися.

8. Налаштування параметрів перевірки та компіляція проєкту

Компіляція проєкту є одним із важливих етапів розробки. В основі цього процесу лежить механізм порівняння, який визначає відмінності між принциповою схемою і ДП та формує список змін для усунення всіх невідповідностей. Особливістю такого підходу є не тільки безпосередня робота з принциповою схемою і ДП (без створення проміжного списку з'єднань), але і можливість використовувати механізм порівняння для синхронізації схеми та плати на будь-якому етапі проектування. Механізм порівняння також дозволяє знаходити відмінності між вихідними та цільовими файлами і виконувати оновлення (синхронізацію) в обох напрямках.

Під час компіляції також застосовуються комплексні правила проектування та електричні правила. Після усунення всіх помилок скомпільована принципова схема може бути передана у документ ДП з формуванням серій запитів на внесення змін **Engineering Change Order (ECO)**.

Основні налаштування проєкту та правил перевірки задаються у вікні *Options for PCB Project <ім'я_проєкту>* (рис. 31), що викликається із меню *Project* вибором команди *Project Options*²⁶. До параметрів проєкту відносяться параметри модуля перевірки на наявність помилок, параметри матриці з'єднань, генератора класів, компаратора, звіту про зміни, шляхи вихідних даних і опції списку з'єднань, налаштування друку, шляхи пошуку тощо. САПР **Altium Designer** використовує ці параметри під час компіляції проєкту.

²⁶ Причому вікно налаштувань викликається однаково як із Редактора схем, так і з Редактора плат.

Для налаштування компіляції важливими є дві перші вкладки: **Error Reporting** та **Connection Matrix**.

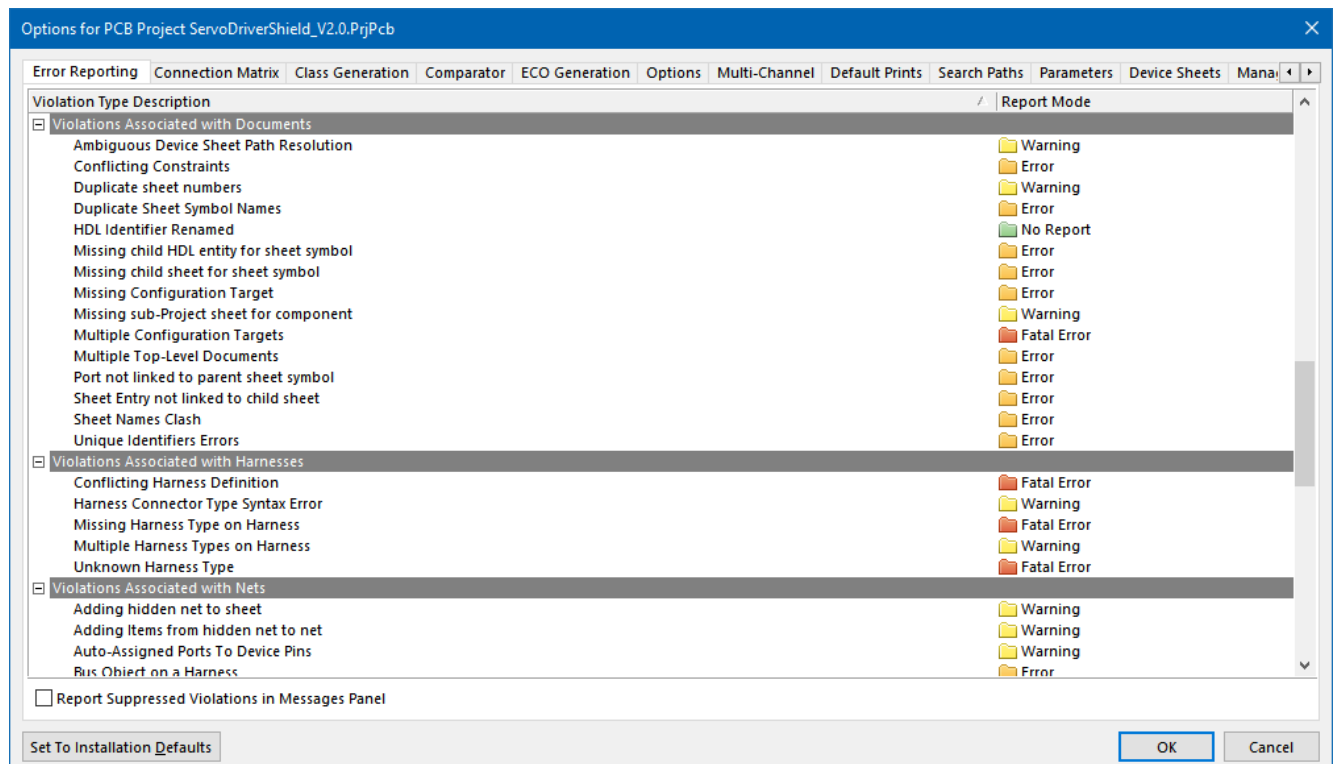
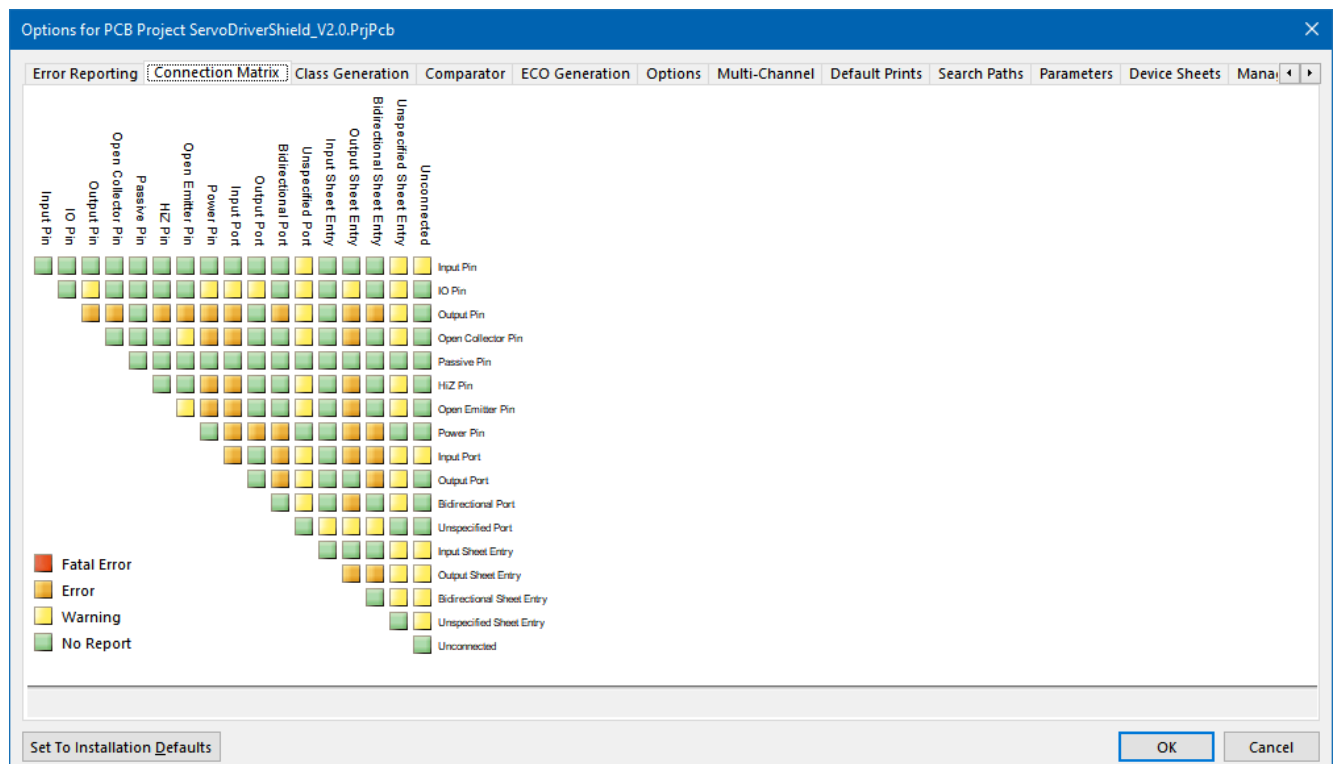


Рис. 31. Вкладка **Error Reporting**

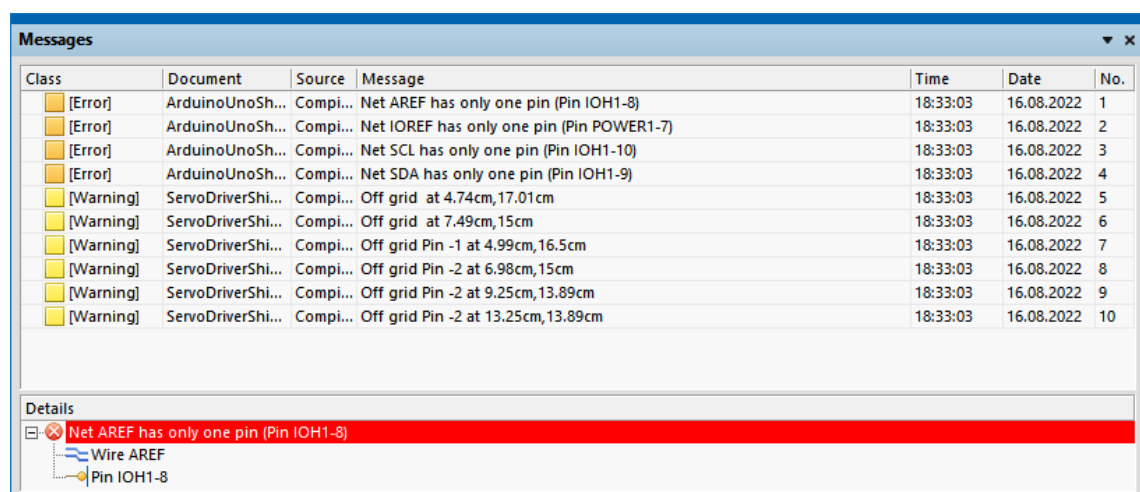
У вкладці **Error Reporting** (див. рис. 31) показані усі можливі помилки, які перевірятимуться у поточному (відкритому) проєкті, та вказано, яким чином система буде реагувати, коли їх виявить. Їх можна залишити без змін або змінювати окремі правила за потребою. Наприклад, правило **Nets with multiple names** (ланцюг, що містить кілька назв) за замовчуванням має значення **Warning**. Якщо розробник проєкту до одного й того ж ланцюга приєднає мітку землі, а в іншій частині схеми (якщо ланцюг довгий) додасть іншу мітку з іменем, що відрізняється від мітки землі, то за стандартних налаштувань буде виведено лише повідомлення **Warning**; причому невідомо, яка з міток буде привласнена в кінцевому результаті під час трасування плати. Щоб цього уникнути, даному правилу можна привласнити значення **Fatal Error**. Подібним чином розробник може налаштувати й інші правила під власні потреби.

Іншою важливою вкладкою є **Connection Matrix** (рис. 32), у якій перераховані усі можливі комбінації з'єднань між різними виводами, портами та переходами у схемі поточного проєкту. За допомогою матриці з'єднань можна задати рівень значимості підключень кожного типу.

Рис. 32. Вкладка *Connection Matrix*

На перетині відповідних стовпчиків та рядків можливо задати наступні реакції системи у випадку виявлення порушень: **Fatal Error**, **Error**, **Warning**, **No Report**, які мають свій колір. Змінювати правила з'єднань можна шляхом натискання ЛКМ на відповідному прямокутнику (водночас змінюватиметься колір).

Після того, як налаштовані описані вище правила, можна переходити до наступного етапу – компіляції проєкту. Для цього необхідно перейти у меню **Project** та виконати команду **Compile PCB Project <ім'я_проєкту>**. Після компіляції проєкту з'являється вікно **Messages**, в якому відображаються усі знайдені помилки (рис. 33).

Рис. 33. Вікно *Messages*

Спочатку відображаються **Fatal Error**, потім **Error**, а вже потім виконується аналіз повідомлень **Warning**. Виділивши відповідний рядок, можна переглянути деталі помилки або повідомлення у субвікні **Details**. Якщо ж у ньому подвійно клацнути ЛКМ на назві елемента, який обумовив помилку, то програма автоматично відобразить та підсвітить його на схемі.

9. Перенесення проєктних даних із Редактора схем у Редактор плат

Процес передачі даних від принципової схеми до ДП запускається командою **Design → Update PCB Document <ім'я_проєкту>** за умови, що наразі активним є документ принципової схеми. Під час цього процесу відбувається компіляція проєкту та виконуються наступні кроки:

1. Формується список всіх компонентів, що використовуються у поточному проєкті.
2. Здійснюється пошук ПМ для кожного компонента у доступних бібліотеках, після чого вони розміщуються на ДП. Якщо ПМ недоступне, то генерується повідомлення про помилку.
3. Створюється список усіх ланцюгів у проєкті (тобто підключених виводів компонентів). **Altium Designer** додає кожен ланцюг на ДП, а потім намагається додати виводи, що належать цим ланцюгам. Якщо вивід додати не вдається, то генерується помилка. Зазвичай це пов'язано з відсутністю моделі ПМ для того чи іншого компонентного модуля.
4. Передаються додаткові проєктні дані, які включають «кімнати» розміщення, класи ланцюгів і компонентів, а також правила проектування ДП.

Тому перед тим, як передавати дані із принципової схеми у порожню ДП, необхідно упевнитися у відсутності помилок та доступності всіх необхідних бібліотек для схеми і плати.

10. Синхронізація між схемою та платою

Altium Designer підтримує двонаправлений зв'язок між принциповою схемою та платою, що забезпечує єдність інтерфейсу користувача і моделі даних для всього

процесу проектування та підвищує його ефективність. Ця концепція є однією із ключових у цій САПР.

Під час передачі проєкту із Редактора схем у Редактор плат перший раз або під час внесення змін до існуючого проєкту на схемі або на платі, необхідно підтримувати синхронізацію цих двох областей проектування. **Altium Designer** містить ефективні інструменти синхронізації даних, які спрощують задачу підтримки синхронізації, завдяки чому розробник може більше зосередитися на творчій складовій процесу проектування. [Синхронізація проєктних даних здійснюється безпосередньо між редакторами схем і плат](#). Система використовує модуль порівняння (компаратор), щоб порівнювати всі аспекти проєкту і виводити список відмінностей.

Існує два підходи до виконання синхронізації: двостороння та одностороння. Якщо обрана двостороння синхронізація, то зміни можуть бути внесені одночасно в обидва напрямки, тобто від схеми у плату і навпаки (використовується команда *Project* → *Show Differences*). Одностороння синхронізація діє лише в одному напрямку і виконується за допомогою команд *Design* → *Update PCB* <ім'я_проєкту> та *Design* → *Update Schematics* <ім'я_проєкту>.

11. Налаштування правил проектування

Редактор друкованих плат є середовищем, робота якого заснована на правилах²⁷. Це означає, що під час внесення змін до проєкту, наприклад, розміщення доріжок, переміщення компонентів або виконання автоматичного трасування, **Altium Designer** відстежує кожну дію і перевіряє дотримання правил проектування. Якщо порушується якесь правило, то програма негайно повідомляє про це. Завдяки потужній і повністю візуалізованій системі завдання та перевірки правил проектування користувач отримує повний контроль над процесом розробки топології.

Усі вимоги задаються у вигляді правил та умов в спеціальному діалоговому вікні *PCB Rules and Constraints Editor* (рис. 34), яке можна відкрити, обравши у меню *Design* пункт *Rules*.

У лівій частині вікна обирається правило із деякої категорії, а у правій частині вікна відображаються налаштування обраного правила, включаючи область його

²⁷ Детальніше про правила та їх налаштування дивіться у [Додатку В](#).

дії та умови, що ним встановлюються. Початково усі правила або задані за замовчуванням, або визначені під час використання майстра створення ДП.

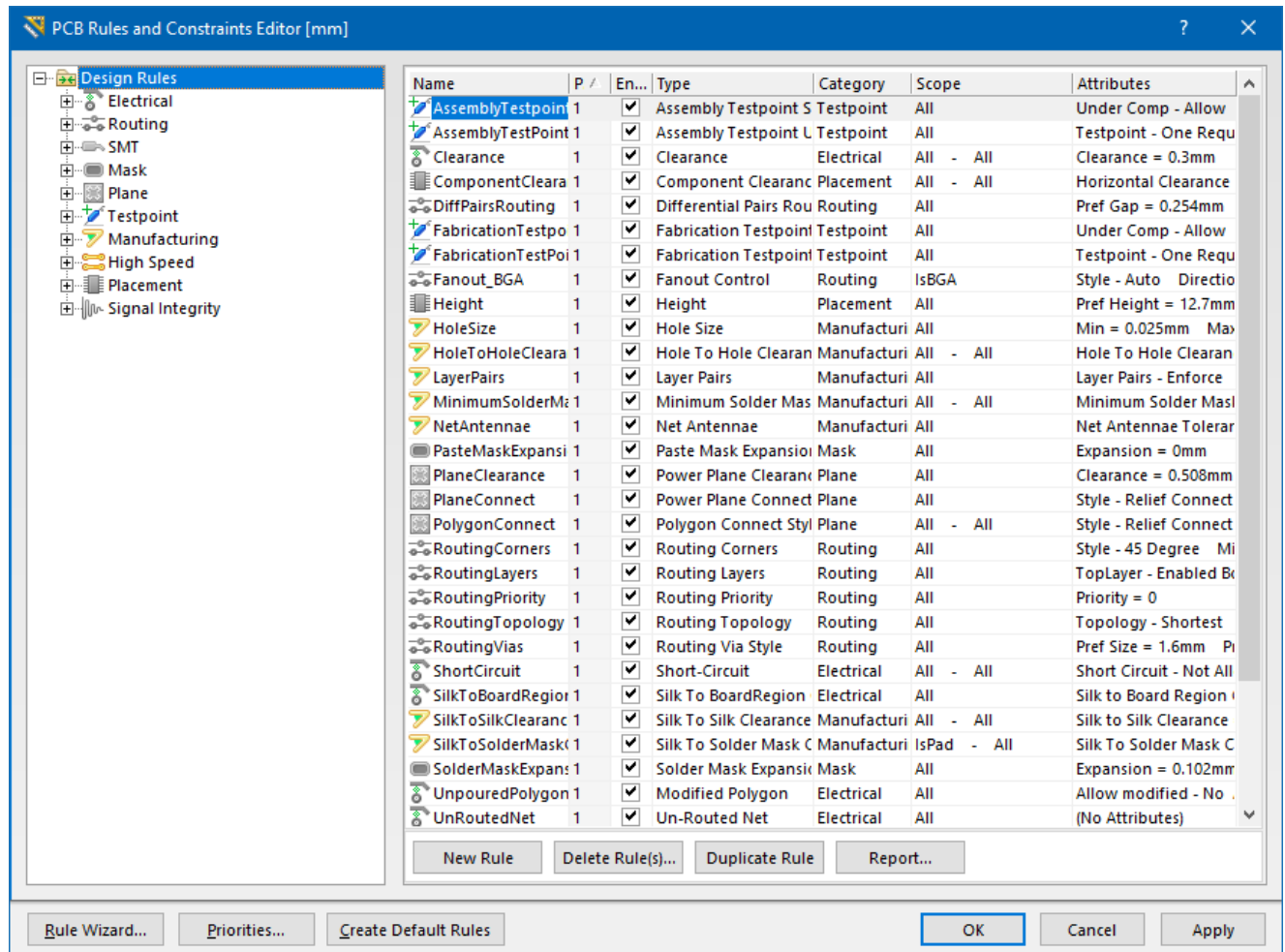


Рис. 34. Діалогове вікно *PCB Rules and Constraints Editor*

До першої групи правил **Electrical** входять: зазори (**Clearance**), короткозамкнені ланцюги (**Short-Circuit**), з'єднання, що не були трасовані (**Un-Routed Net**), і не приєднані контакти (**Un-Connected Pin**).

До другої групи правил **Routing** входять: ширина друкованих провідників (**Width**), алгоритм трасування (**Routing Topology**), пріоритет трасування (**Routing Priority**), дозволені шари трасування (**Routing Layers**), варіант побудови кутів трасування (**Routing Corners**), розміри перехідних отворів (**Routing Via Style**), правила підключення до КМ (**Fanout Control**), трасування диференціальних пар (**Differential Pairs Routing**).

Третя група правил **SMT** пов'язана із налаштуваннями розмірів провідника, що підключається до планарних КМ: налаштування відстані від КМ до повороту траси (**SMD To Corner**), налаштування мінімальної відстані до перехідного отвору (**SMD To Plane**), налаштування звуження ширини провідника (**SMD Neck-Down**).

До четвертої групи правил **Mask** входять налаштування зазорів для паяльної маски (**Solder Mask Expansion**) та налаштування зазорів для паяльної пасти (**Paste Mask Expansion**).

П'ята група правил **Plane** визначає налаштування приєднання полігонів та екранних шарів: стиль з'єднання перехідних отворів та КМ із шаром живлення (**Power Plane Connect Style**), зазори у шарах живлення (**Power Plane Clearance**), стиль з'єднання перехідних отворів та КМ з полігоном (**Polygon Connect Style**).

Шоста група правил **Testpoint** описує різні види контрольних точок.

Сьома група правил **Manufacturing** дає змогу налаштувати технологічні умови виробництва: мінімальний розмір КМ (**Minimum Annular Ring**), обмеження на розмір гострих кутів (**Acute Angle**), діаметр отворів (**Hole Size**), пари шарів (**Layer Pairs**), співпадіння та перетин отворів (**Hole to Hole Clearance**), мінімальна ширина ділянки в захисній масці (**Minimum Solder Mask Silver**), зазор між маркуванням та розкритими від маски металізованими КМ (**Silkscreen Over Component Pads**), перевірка зазорів на шарі шовкографії (**Silk to Silk Clearance**), аналіз ділянок металізації, які можуть або не можуть бути сформовані на етапі виробництва ДП (**Minimum Copper Width**), визначення не підключених ділянок трас та дуг на сигнальних шарах (**Net Antennae**).

Восьма група правил **High Speed** встановлює правила трасування для високошвидкісних ланцюгів: довжина паралельних трас (**Parallel Segment**), довжина певного ланцюга (**Length**), налаштування різниці в довжині певних ланцюгів або диференціальних пар (**Matched Net Lengths**), налаштування Т-подібного з'єднання до КМ (**Daisy Chain Stub Length**), дозвіл перехідного отвору під КМ (**Vias Under SMD**), максимальна кількість перехідних отворів (**Maximum Via Count**).

Дев'ята група правил **Placement** визначає правила розміщення елементів на платі: область розміщення (**Room Definition**), відстань між компонентами (**Component Clearance**), орієнтація компонентів (**Component Orientations**), розміщені шари (**Permitted Layers**), ланцюги, що ігноруються (**Nets to Ignore**), висота (**Height**).

Остання, десята група правил **Signal Integrity**, визначає правила для моделювання електричних схем ДП.

Наявних за замовчуванням правил достатньо для переважної більшості проектів. Однак користувачу доступна можливість створення власних правил за допомогою спеціального майстра *Design Rule Wizard*. Для того, щоб його

викликати, необхідно натиснути кнопку **Rule Wizard** у лівій нижній частині вікна *PCB Rules and Constraints Editor* (див. рис. 34).

12. Тривимірні моделі корпусів електронних компонентів

Будь-яка ДП є складовою частиною електронного пристрою, який загалом містить не тільки ДП, а має корпус, зовнішні роз'єми тощо. Це означає, що розробник повинен розуміти, яким чином плата буде розміщуватись у корпусі приладу. Для цього йому потрібно чітко знати усі геометричні розміри ДП, зокрема: довжину, ширину та висоту. Якщо говорити про довжину і ширину, то вони визначаються конструкцією основи плати і чітко відомі на етапі розробки. А от висота визначається набором електронних компонентів, які будуть змонтовані на платі. Фактично, вона стане достеменно відомою тільки після виготовлення плати і розташування усіх компонентів. А що, якщо висота плати з компонентами виявиться більшою за висоту корпусу приладу? Тоді потрібно буде або міняти компоненту базу або збільшувати висоту корпусу. Добре, якщо не буде проблем зі зміною висоти корпусу, однак у деяких випадках до корпусу можуть бути жорсткі вимоги за висотою, тому доведеться вносити зміни у проєкт плати. Очевидно, що це призведе до значних втрат часу та ресурсів.

Тому можливість контролю загальних геометричних параметрів (зокрема, висоти) ДП у тривимірному форматі є вкрай важливою для будь-якої САПР електронних пристроїв.

Найпростішим способом такого контролю в **Altium Designer** є додавання атрибуту висоти кожного компонента. Для цього необхідно у Редакторі посадкових місць двічі клацнути ЛКМ на назві ПМ у списку *Components* панелі *PCB Library*. У вікні, що відкриється, буде поле **Height**, яке і визначає загальну висоту електронного компонента, якому належить це ПМ. Якщо ж після цього налаштувати правила проєктування, які визначають мінімальну та максимальну висоту, то у випадку їх порушення деяким компонентом буде генеруватись відповідне повідомлення.

Налаштування висоти окремого компонента дійсно зручно використовувати під час перевірки на відповідність правилам, проте воно не впливає на відображення компонента у тривимірному режимі, який є зручним для розробника. Тому в **Altium Designer** передбачені більш потужні інструменти, які дозволяють візуалізувати у

тривимірному режимі увесь корпус електронного компонента. Причому [усі його геометричні характеристики можуть бути враховані під час аналізу](#).

Тривимірну модель можна створити як вбудованими засобами **Altium Designer**, так й імпортувати готову модель, що створена у іншій спеціалізованій системі проектування механічних пристроїв (SolidWorks, AutoCAD, CATIA тощо). Розглянемо докладніше обидва способи. Необхідно зауважити, що в обох випадках для створення і/або розміщення тривимірної моделі корпусу використовується вікно **3D Body**²⁸ (рис. 35), доступ до якого у Редакторі посадкових місць виконується із меню **Place** викликом однойменної команди.

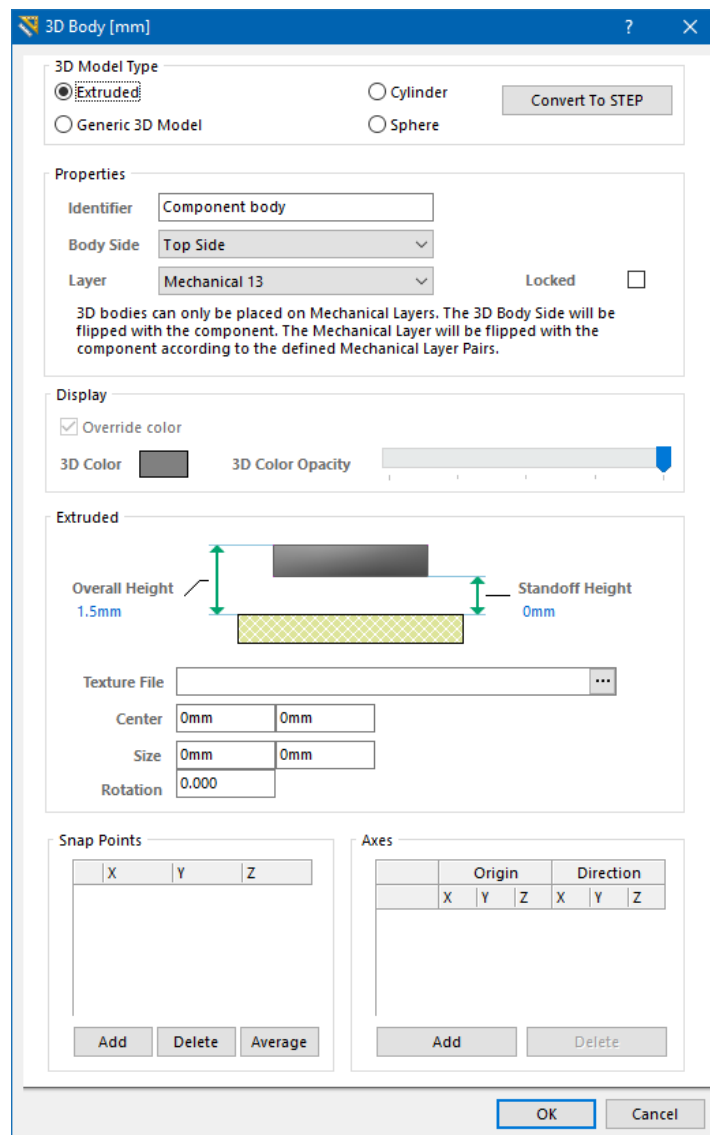


Рис. 35. Вікно **3D Body** із обраною формою **Extruded**

²⁸ До речі, так само називається і сам об'єкт, який міститиме інформацію про тривимірну модель корпусу електронного компоненту.

Важливо!

Тривимірна модель корпусу електронного компонента обов'язково має розміщуватись на механічному шарі, причому вибір номеру такого шару є довільним.

Для того, щоб створити тривимірну модель корпусу засобами **Altium Designer** передбачено три простих об'ємних геометричних форми (або простих об'єктів **3D Body**): екструдований багатокутник (**Extruded**), циліндр (**Cylinder**) та сфера (**Sphere**). Для вибору необхідної форми потрібно встановити відповідний прапорець в області **3D Model Type**. Кожна із форм має власні геометричні параметри, які можна налаштовувати. Зокрема, на рис. 35 показане вікно із обраною формою **Extruded**, для якої можна налаштувати висоту від поверхні плати, максимальну висоту, колір, текстуру тощо.

Після натискання кнопки **ОК** вікно закриється і вам стане доступним режим розміщення багатокутника, вершини якого формуються натисканням ЛКМ у заданій точці²⁹. Для виходу із режиму розміщення багатокутника потрібно натиснути клавішу **<Esc>** або ПКМ. Після цього автоматично знову відкриється вікно **3D Body**, що дасть вам змогу додати нову геометричну форму. Якщо ви не хочете більше додавати нових форм, то просто закрийте це вікно.

Зауваження!

Ви можете застосовувати будь-яку кількість простих форм (об'єктів **3D Body**), комбінуючи їх у просторі таким чином, щоб створити складну геометричну просторову фігуру, яка визначатиме корпус електронного компонента. У випадку використання великої кількості простих форм рекомендується кожній з них привласнювати окрему назву (поле **Identifier**), що дасть змогу легше ідентифікувати кожен елемент моделі у майбутньому під час редагування.

Розміщену просту геометричну форму можна переміщувати на площині, виділивши її та утримуючи ЛКМ, змінювати форму зміщенням опорних точок (для

²⁹ Це стосується лише випадку, коли обрана форма **Extruded**. Якщо будуть обрані форми **Cylinder** або **Sphere**, то після натискання кнопки **ОК** ви перейдете у режим розміщення безпосереднього самого об'єкту.

багатокутника – переміщенням його вершин) та у будь-який момент відкрити подвійним клацанням ЛКМ вікно властивостей для редагування. Також об'єкти **3D Body** можна копіювати з метою дублювання, що є вкрай корисним, зокрема, коли створюються виводи корпусу електронного компоненту, які зазвичай є однаковими.

Об'єкти **3D Body** можуть бути використані перевіркою правил проєктування для контролю перетинів компонентів один з одним і зазорів між ними, а також для візуалізації ДП у тривимірному режимі.

Також існує ще один спосіб створення об'єкту **3D Body** засобами **Altium Designer**. Для цього необхідно використовувати менеджер створення корпусів компонентів, який можна викликати із меню *Tools*, обравши команду *Manage 3D Bodies for Current Components*³⁰. Внаслідок виконання цієї команди відкриється нове вікно *Component Body Manager for component: <ім'я_компоненту>* (рис. 36).

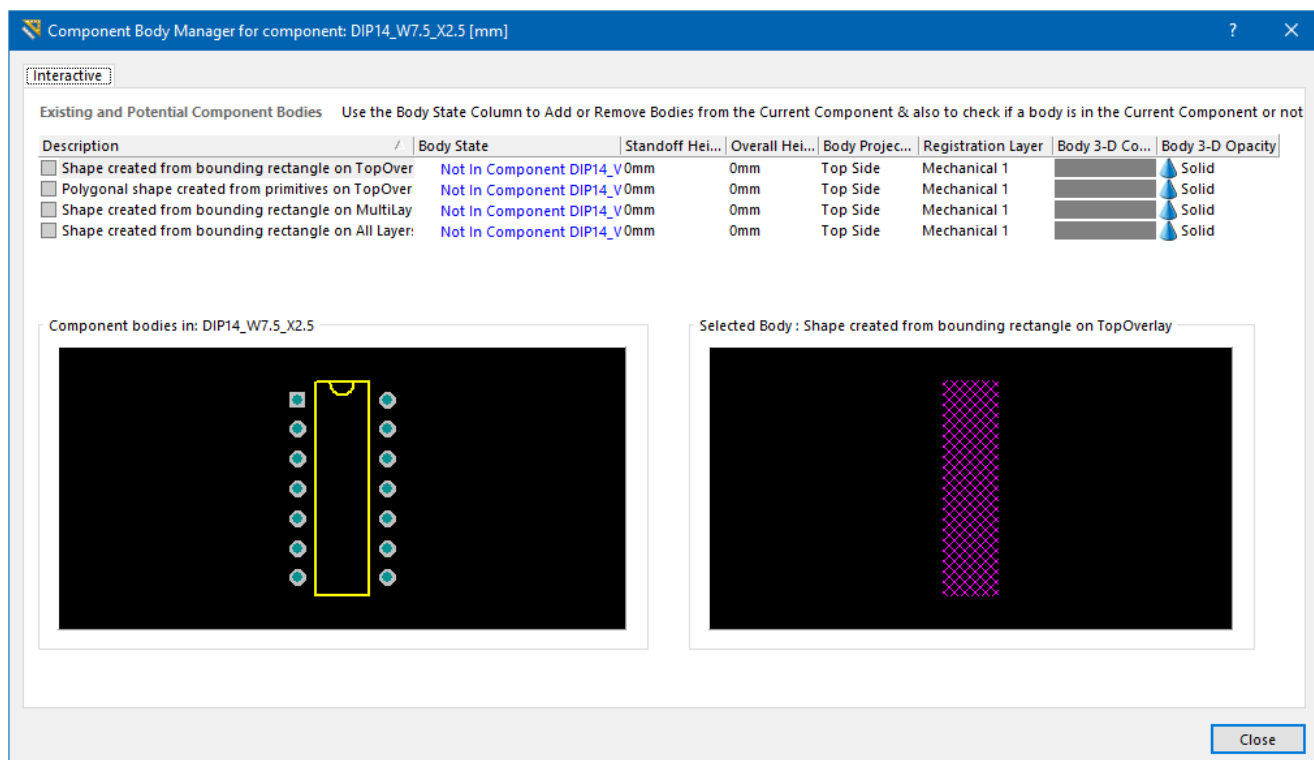


Рис. 36. Вікно *Component Body Manager for component*

³⁰ Окрім запуску менеджера створення корпусів компонентів для окремого компонента, можна також виконати дещо іншу команду: *Tools* → *Manage 3D Bodies for Library*. У цьому випадку відкриється менеджер із переліком усіх компонентів у поточній бібліотеці із можливістю навігації та автоматичним створенням **3D Body** на основі замкнутих багатокутних обрисів існуючих примітивів.

У цьому вікні [можна автоматично створювати об'єкти 3D Body на основі прямокутного контуру або замкнутих багатокутних обрисів примітивів, які вже є в ПМ](#). Наприклад, для того, щоб створити просту екструдовану форму на основі лінії контуру, визначеного для компонента на шарі шовкографії, необхідно обрати опцію *Polygonal shape created from primitives on Top Overlay* (Створити багатокутну форму з примітивів на шарі *Top Overlay*). Водночас у стовпчику **Body State** потрібно встановити опцію у положення **In Component** <ім'я_компоненту>, у стовпчику **Registration Layer** вибрати механічний шар, на який буде додано об'єкт **3D Body**, а у стовпчиках **Overall Height** і **Standoff Height** встановити відповідно максимальну висоту компонента та зазор між нижньою площиною корпусу та поверхнею плати.

Очевидно, що наявні в **Altium Designer** інструменти для створення тривимірних моделей корпусів є достатніми для роботи і дають змогу за наявності необхідних навичок створювати досить непогані моделі. Проте вони є незручними і до них не так легко звикнути. Для того, щоб створювати реалістичні моделі корпусів необхідно все ж використовувати спеціалізовані системи проєктування механічних пристроїв, які надають більш широкий спектр можливостей для моделювання.

Altium Designer 17.1 підтримує імпортування із таких форматів, як STEP (*.Step), SolidWorks (*.SldPrt) та Parasolid (*.x_t, *.x_b). Отже, тривимірну модель, яка створена у SolidWorks можна безпосередньо імпортувати у **Altium Designer**, це ж стосується і САПР механічних пристроїв, які використовують для моделювання геометричне ядро Parasolid. Але як бути, якщо тривимірна модель створена у іншій САПР? Її також можна буде імпортувати, якщо спочатку конвертувати у формат STEP³¹ (звичайно, САПР, у якій створювалась тривимірна модель, має підтримувати конвертування у цей формат). Фактично, у цьому випадку STEP є своєрідним проміжним форматом, який використовується для обміну даними між різними програмами.

³¹ [STEP](#) (Standard for Exchange of Product data) – універсальний формат подання даних про виріб. Основне призначення – представлення 3D-моделей конструктивних елементів у єдиному форматі (єдиній інформаційній моделі), незалежно від САПР, у якій зрештою було створено модель. [STEP-файли](#) мають модульну структуру, кожен модуль називається «класом». Класи містять архітектуру реалізації, тестові процедури відповідності, ресурсні інформаційні моделі та протоколи застосунків. Це означає, що STEP-файл є більше, ніж файлом геометричних даних. Більшість сучасних САПР підтримують експортування 3D-геометрії у цей формат.

Таким чином, ви можете власноруч створити³² тривимірну модель корпусу електронного компонента у будь-якій зручній для вас САПР механічних пристроїв, після чого конвертувати модель у формат STEP і, врешті-решт, підключити цю модель до компонентного модуля в **Altium Designer**.

Рекомендація!

Для зручності краще створити у головній папці проєкту бібліотеки додаткову вкладену папку, назвавши її, наприклад, «3D models». У цій папці можна зберігати моделі у форматі STEP усіх електронних компонентів, що належать до бібліотеки, і вже із неї виконувати імпортування.

Для того, щоб підключити зовнішню модель, також використовується вікно **3D Body** (див. рис. 35), у якому необхідно обрати опцію **Generic 3D Model**. Після цього потрібно натиснути кнопку **Load from file** та завантажити файл моделі. За потреби розміщену модель можна обертати та переміщувати таким чином, щоб виводи співпали із КМ.

Для порівняння на рис. 37 представлені різні варіанти тривимірних моделей корпусу DIP14.

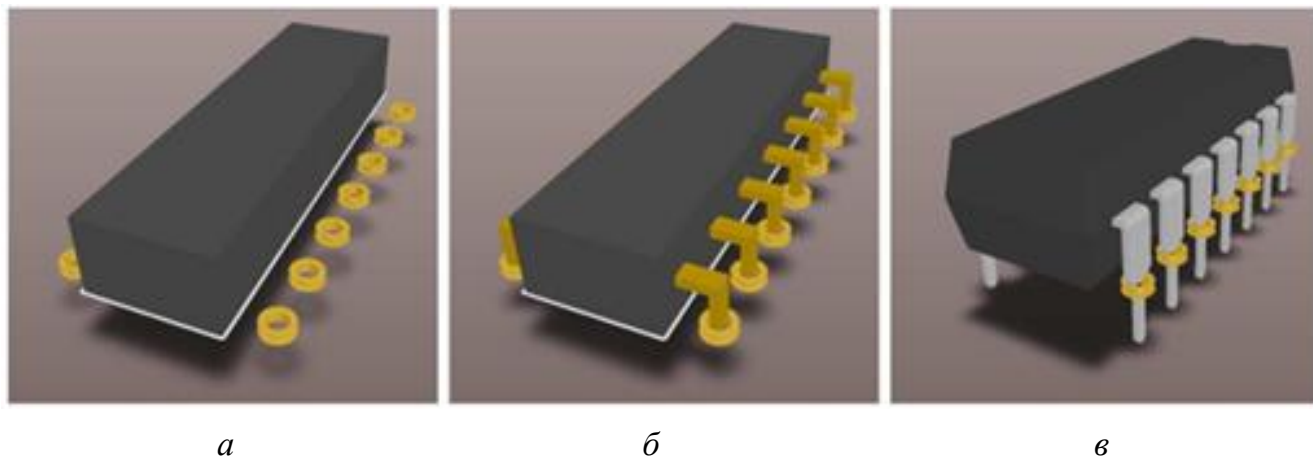


Рис. 37. Корпус DIP14: *а* – у вигляді екструдованої прямокутної форми, *б* – кілька різних екструдованих прямокутних форм, що моделюють сам корпус та виводи, *в* – імпортована тривимірна модель

³² Іншим варіантом є використання сторонніх моделей корпусів електронних компонентів, які є у вільному доступі в інтернеті та розміщені на спеціалізованих ресурсах, наприклад, www.3dcontentcentral.com або www.ultralibrarian.com.

Зауваження!

Об'єкти **3D Body**, що створені вручну із найпростіших форм (екструдованої прямокутної, циліндричної, сферичної), або за допомогою **Component Body Manager**, або імпортовані із зовнішніх редакторів, можна комбінувати між собою.

13. Робота із платою у тривимірному режимі

У зв'язку зі збільшенням рівня складності та щільності монтажу сучасних електронних пристроїв розробнику ДП необхідно не тільки контролювати розташування компонентів в площині плати, а й враховувати вимоги до їх розташування у просторі. Можливість перегляду конструкції плати у тривимірному режимі дає змогу виконати віртуальну збірку пристрою і перевірити повне компонування вже на стадії розробки.

Редактор друкованих плат може відображати моделі плат як у двовимірному, так і у тривимірному вигляді. Для перемикання між **2D** і **3D** режимами можна скористатися командами меню **View** → **Switch To 3D** та **View** → **Switch To 2D** (або клавіші <3> та <2> відповідно).

Для роботи в 3D режимі **Altium Designer** потрібно мати графічний адаптер з підтримкою **DirectX 9.0c** і вище та **Shader Model 3** і вище.

У тривимірному режимі відображення можна:

- масштабувати плату, для чого необхідно переміщувати мишку, утримуючи натиснутими клавішу <Ctrl> і ПКМ, або прокручувати колесо мишки з утриманням <Ctrl>, або натискати клавіші <Page Up> / <Page Down>;
- переміщувати плату, для чого необхідно переміщувати мишку із затисненою ПКМ;
- обертати плату у просторі, для чого необхідно переміщувати мишку, утримуючи клавішу <Shift> і ПКМ.

Щодо обертання плати у просторі необхідно дещо пояснити, оскільки новачки в **Altium Designer** не одразу розуміють, як це робити. Спочатку необхідно натиснути клавішу <Shift> у деякій точці робочої області Редактора плат, після чого в позиції курсору з'явиться сфера з напрямками (рис. 38). Далі затиснути ПКМ

і почати плавно переміщувати курсор мишки. Необхідно пам'ятати, що поворот моделі здійснюється відносно центру сфери.

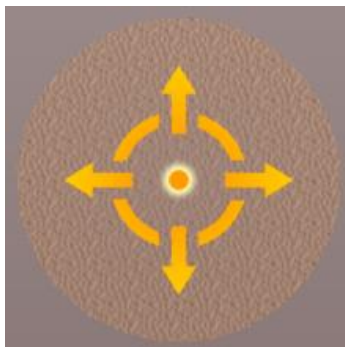


Рис. 38. Сфера для обертання плати у тривимірному режимі

Налаштування параметрів відображення у тривимірному режимі відбувається у вікні *View Configurations* (див. рис. 21). Можна обрати колір поверхні і робочого простору. Для деяких поверхонь можна задавати властивість прозорості, причому, чим вище значення параметра, тим менше «світла» пропускає поверхня, приховуючи об'єкти за нею. Також можна обрати кольори шарів для відображення корпусів електронних компонентів або 3D-об'єктів.

Тривимірну модель ДП із розміщеними на ній компонентами можна експортувати у формат **STEP** або **DWG / DXF**, для чого необхідно скористатись командою *Export* із меню *File*.

Також, перебуваючи у тривимірному режимі, ви можете скопіювати знімок моделі ДП у буфер обміну, використовуючи комбінацію клавіш <Ctrl + C>. Зображення поміщається у буфер обміну ОС Windows у растровому форматі, після чого ви можете вставити його у будь-який графічний редактор.

14. Перехресне виділення та перехресний перехід

Altium Designer містить багато зручних можливостей для одночасної роботи як із принциповою схемою, так і платою. Особливо, коли вони одночасно відображаються на одному екрані у різних вкладках або взагалі на різних моніторах.

Часто виникає ситуація, коли необхідно знайти ПМ на платі, що відповідає деякому компоненту на схемі (або навпаки). Для цього використовується перехресне виділення, коли можна просто виділити компонент у одному із редакторів, після чого він автоматично виділиться у іншому. Так само можна

зробити з групою елементів: якщо виділити групу елементів у одному редакторі, то уся вона автоматично буде «підсвічена» у іншому. Для того, щоб працювало перехресне виділення, необхідно у меню **Tools** активувати пункт **Cross Select Mode**.

Також як у Редакторі схем, так і у Редакторі плат можна використовувати інструмент **Cross Probe** із меню **Tools**. Завдяки цьому інструменту можна легко обрати в одному та відобразити у іншому редакторі компоненти, шини, з'єднання, виводи та КМ. Цей інструмент особливо зручно використовувати, коли плата відображається у тривимірному режимі. За замовчуванням після вибору деякого елементу в одному із редакторів на короткий час відображається відповідний йому елемент у іншому, після чого відбувається повернення у вихідний редактор. Для того, щоб не повертатись, необхідно утримувати клавішу <Ctrl> під час вибору.

15. Налаштування вихідних даних проєкту

Вихідні дані проєкту (наприклад, складальні кресленики, дані для виробництва та звіти) можна налаштувати за допомогою меню **File** і **Reports**. Ці дані також містяться у файлі проєкту і тому завжди доступні для поточного проєкту. Крім того, ви можете задати параметри вихідних даних у файлі **Output Job** (меню **File** → **New** → **Output Job File**).

16. Питання для самоконтролю

1. Поясніть призначення та особливості САПР **Altium Designer**.
2. Опишіть структуру САПР **Altium Designer**.
3. Поясніть призначення основних програмних модулів, що входять до складу **Altium Designer**.
4. Поясніть, що таке електронний компонент?
5. Що собою представляє компонентний модуль?
6. Поясніть, в чому різниця між поняттями «електронний компонент» та «компонентний модуль».
7. Що представляє собою бібліотека в програмі **Altium Designer**?
8. Яким чином створюється УГП та які атрибути має містити?
9. Яким чином створюється посадкове місце електронного компонента?
10. Що таке «проєкт» в **Altium Designer**?

11. Які проекти можна створити **Altium Designer**?
12. Опишіть типову структуру каталогу проекту плати.
13. Опишіть типову структуру каталогу проекту бібліотеки.
14. Яким чином виконується пошук у бібліотеці?
15. Опишіть структуру Редактора схем.
16. Опишіть структуру Редактора плат.
17. У чому полягають недоліки створення та трасування плати, яка не належить до проекту?
18. Яким чином виконується автоматичне трасування плати?
19. Що таке стратегія автоматичного трасування плати і як її можна змінити?
20. Що таке інтерактивне ручне трасування плати?
21. Опишіть основні режими інтерактивного трасування.
22. Яким чином обирається ширина траси на платі?
23. Для чого потрібна область металізації та яким чином вона створюється?
24. Що таке шар плати?
25. Які існують шари плати у проекті **Altium Designer**?
26. Що таке парні шари і як вони використовуються?
27. Опишіть способи створення контуру плати в **Altium Designer**.
28. Яким чином налаштовується проект та правила перевірки і пошуку помилок?
29. Яким чином можна перенести проектні дані із Редактора схем у Редактор плат?
30. Яким чином відбувається синхронізація між Редактором схем та Редактором плат?
31. Що таке правила проектування та як вони налаштовуються?
32. Яким чином в **Altium Designer** створити тривимірну модель корпусу електронного компоненту?
33. Яким чином в **Altium Designer** імпортувати тривимірну модель корпусу електронного компоненту, що створена у САПР механічних пристроїв?
34. Опишіть основні прийоми роботи із платою у тривимірному режимі.
35. Поясніть, що таке перехресне виділення та перехресний перехід в **Altium Designer**?
36. Поясніть, яким чином налаштовуються вихідні дані проекту?

Комп'ютерний практикум № 1.

ОСНОВИ РОБОТИ В СЕРЕДОВИЩІ ALTIUM DESIGNER

***Мета роботи:** ознайомитись із системою автоматизованого проєктування Altium Designer; ознайомитись з меню та основними командами; створити власну інтегровану бібліотеку та задати початкові налаштування програми.*

1. Теоретичні відомості

САПР **Altium Designer** надає гнучкі можливості з налаштування, що дає змогу користувачеві визначити індивідуальні параметри проєкту. Налаштування **Altium Designer** умовно можна розділити на глобальні, які стосуються самої програми та застосовуються до усіх нових створюваних документів, та локальні, які налаштовуються тільки для окремих документів і не впливають на інші.

Важливими складовими будь-якого проєкту у програмі **Altium Designer** є бібліотеки електронних компонентів. Усі компонентні модулі, які є віртуальними реалізаціями електронних компонентів, повинні належати до тієї чи іншої бібліотеки (вони не існують окремо). Це означає, що перш ніж перейти до розробки компонентних модулів, необхідно створити бібліотеку.

В **Altium Designer** можна виділити 4 основних типи бібліотек: символів (*.SchLib), посадкових місць (*.PcbLib), інтегрована (*.IntLib) та база бібліотек даних (*.DBLib).

У бібліотеці символів зберігаються УГП електронних компонентів, які використовуються під час розробки принципової схеми, а у бібліотеці посадкових місць зберігаються як самі ПМ, так і тривимірні моделі корпусів електронних компонентів, які використовуються під час створення ДП. Саме ці дві бібліотеки є мінімальним набором, який є основою проєкту бібліотеки користувача. Для того, щоб врахувати усі зв'язки між бібліотекою символів та ПМ, в папці проєкту також міститься спеціальний файл *.LibPkg (Library Package). Насправді такий набір розрізнених файлів не є зручним для використання. Тому на основі файлу *.LibPkg компілюють інтегровану бібліотеку *.IntLib, яка є своєрідним архівом, що у вже єдиному файлі зберігає УГП, ПМ та їх асоціативні моделі.

Бібліотека символів є ключовою бібліотекою, оскільки вона містить не тільки УГП компонентів, але й дозволяє об'єднати графічні позначення із відповідними ПМ. Також завдяки такому інструменту цієї бібліотеки, як властивості компонента, можна задавати потрібні користувачу параметри, які згодом будуть використані під час формування технічної документації проєкту плати.

Зовнішній вигляд вікна властивостей *Library Component Properties* нового компонентного модуля за замовчуванням виглядає так, як по казано на рис. 1.1. Це вікно розділене на такі області: *Properties*, *Parameters*, *Models*, *Library Link* та *Graphical*.

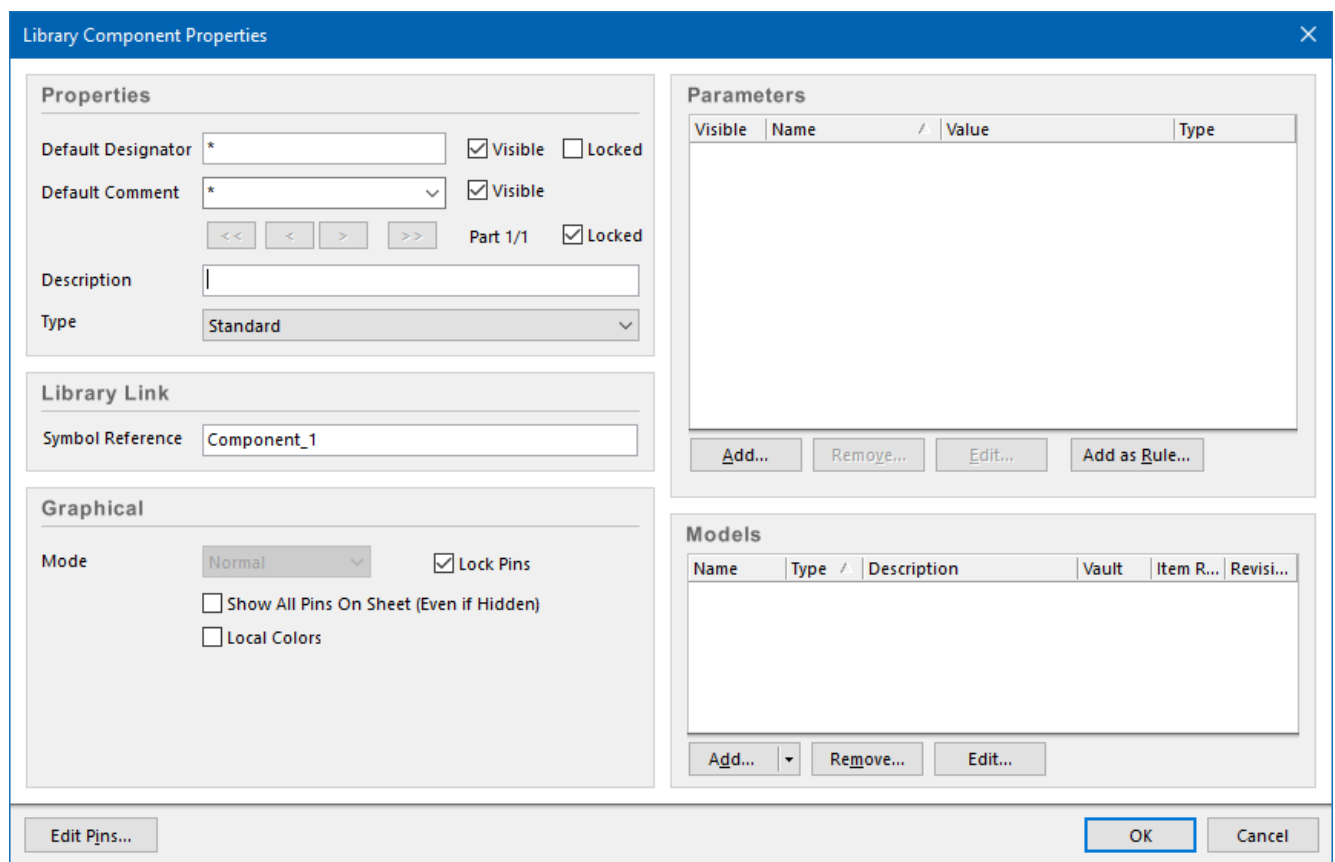


Рис. 1.1. Вікно властивостей *Library Component Properties* нового компонентного модуля бібліотеки символів

За допомогою елементів керування в області *Models* можна додати до графічного зображення електронного компонента його ПМ, тривимірну модель корпусу тощо.

У полі *Symbol Reference* області *Library Link* задають назву компонента у бібліотеці.

Область *Graphical* використовується для керування візуальним відображенням компонента на схемі.

Область **Properties** містить так звані системні атрибути компонента, а **Parameters** – користувацькі.

Розглянемо спочатку системні атрибути, основними з яких є: **Default Designator**, **Default Comment**, **Description** та **Type**.

У полі **Default Designator** вказують позиційне позначення компонента на схемі (наприклад, для резистора – це R, для аналогової мікросхеми – DA тощо). Цей атрибут може бути відображений на схемі (якщо активована опція **Visible**). Також атрибут має опцію **Locked**, активація якої фіксує введене позиційне позначення та запобігає його зміні.

Поле **Default Comment** (коментар за замовчуванням) використовується для введення деякого опису компонента. Тут можна написати назву компонента, або фізичну величину (наприклад, величину опору, ємності, індуктивності тощо). Дані із цього поля відображаються в атрибуті **Comment** посадкового місця електронного компонента на платі, якщо схема та ДП синхронізовані. Окрім того, що в полі **Default Comment** можна напряду записати деякий текст, існує можливість обрати для відображення із випадного списку один із користувацьких параметрів, якщо такі були створені. Для відображення атрибуту **Default Comment** на схемі має бути активована його опція **Visible**.

У полі **Description** вказують короткий опис³³, який має бути лаконічним, але чітко характеризувати сам електронний компонент, для якого створюється бібліотечний компонентний модуль. В описі можна вказати, наприклад, що електронний компонент є резистором, логічною схемою, операційним підсилювачем тощо.

За допомогою параметру **Type** вказують тип бібліотечного компонента (обирається із випадного списку):

- **Standard** – ці компоненти мають стандартні електричні властивості та завжди синхронізовані (тобто присутні як на схемі, так і на платі);
- **Mechanical** – ці компоненти є на платі, але відсутні на схемі, оскільки не мають електричних властивостей (наприклад, радіатор охолодження);
- **Graphical** – ці компоненти є на схемі, проте не мають електричних властивостей і можуть бути відсутніми на платі, внаслідок чого вони не використовуються під час синхронізації і не перевіряються на наявність електричних помилок (наприклад, це можуть бути логотипи компанії);

³³ Зазвичай це не повинна бути назва компонента, оскільки вона відображається у полі **Symbol Reference**.

- **Net Tie (in BOM)** – це компоненти, всередині яких допустиме об'єднання ланцюгів (наприклад, НВЧ-структури, об'єднання земель), причому вони відображаються у специфікації (in BOM);
- **Net Tie** – те саме, що й у попередньому пункті, тільки без відображення у специфікації;
- **Standard (No BOM)** – те саме, що і *Standard*, тільки без відображення у специфікації;
- **Jumper** – це компоненти-перемички, що є на платі, але не відображаються на схемі (здебільшого використовуються на односторонніх платах).

Описані вище системні атрибути компонента є чітко визначеними, проте часто виникає необхідність використовувати додаткові атрибути, які може створити сам користувач. Для цього використовується область вікна *Library Component Properties*, яка називається *Parameters*. У ній розробник може індивідуально налаштувати необхідні параметри компонентів, які знадобляться під час роботи над тим чи іншим проєктом. Завдяки цим параметрам компонент легко ідентифікувати у базі даних, а також виконувати пошук. Більше того, дані із цієї області використовуються під час формування переліку елементів схеми (**Bill of Materials**), конструкторської документації тощо.

Нові параметри створюються шляхом натискання кнопки **Add**, яка розташована безпосередньо внизу області *Parameters*.

Кожен розробник сам вирішує, які параметри компонента йому необхідні надалі під час роботи над проєктом. Назви параметрів і їх значення обираються самостійно, хоча варто дотримуватись певних угод. З переліком рекомендованих користувацьких параметрів ви зможете ознайомитись під час виконання цього комп'ютерного практикуму.

2. Завдання на комп'ютерний практикум

1. Створити власну інтегровану бібліотеку.
2. Задати і зберегти початкові налаштування програми для використання у наступних проєктах.
3. Налаштувати базові параметри компонентних модулів, які будуть використовуватись у майбутніх проєктах.

3. Порядок виконання комп'ютерного практикуму

3.1. Створення інтегрованої бібліотеки

Для створення нового інтегрованого пакета бібліотек необхідно спочатку виконати команду **File** → **New** → **Project** → **Integrated Library**. У вікні [New Project](#) (рис. 1.2) потрібно обрати пункт **Integrated Library**. За замовчуванням буде запропоновано створити новий файл пакету бібліотек з іменем **Integrated_Library.LibPkg** у спеціальному каталозі **Projects**, який використовується для збереження різних проєктів.

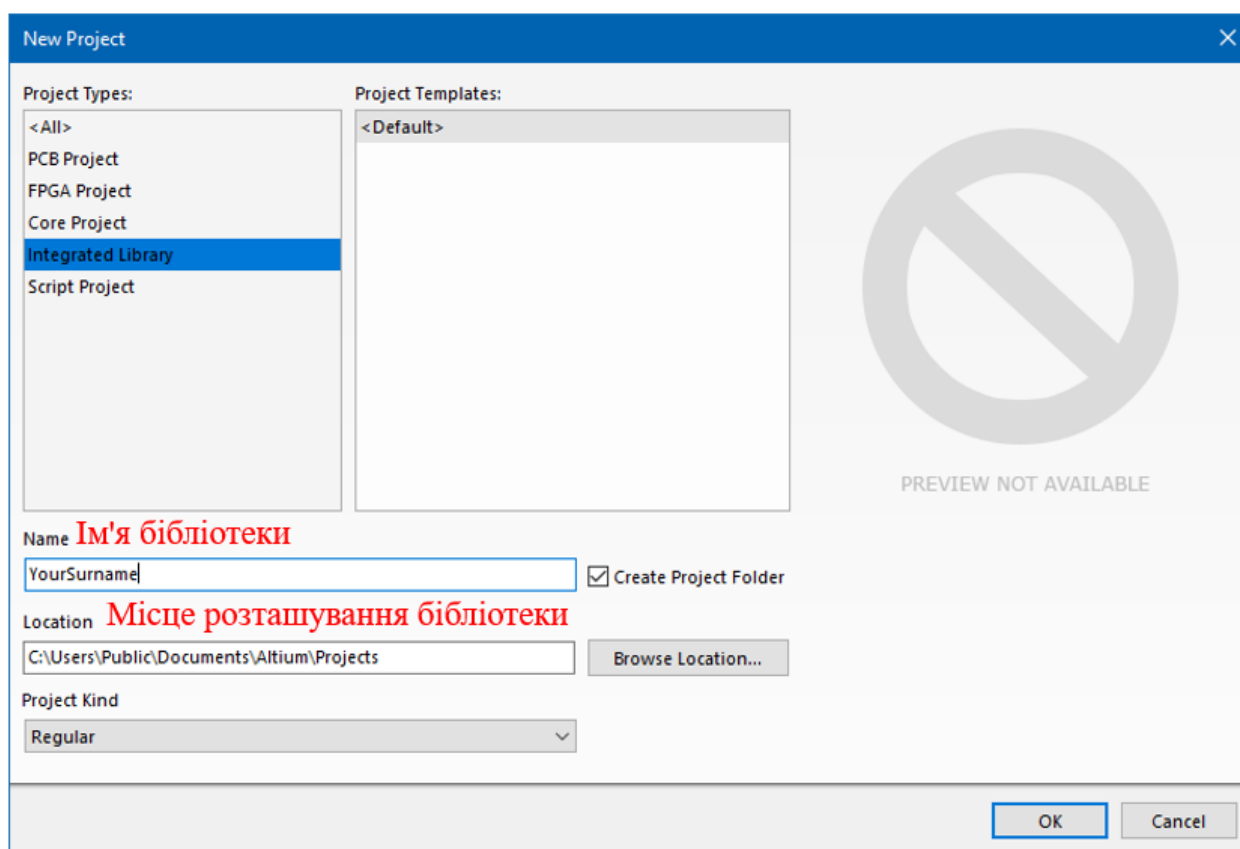
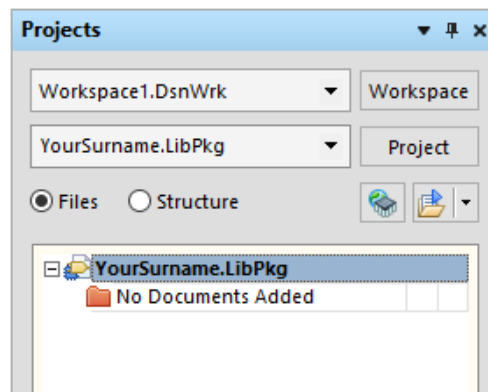


Рис. 1.2. Вікно створення нового проєкту бібліотеки

У полі **Name** задайте нове осмислене ім'я для власної бібліотеки³⁴. Врахуйте, що розширення ***.LibPkg** буде додано автоматично, якщо ви не введете його явно. Після створення нова бібліотека буде відображатись у панелі **Projects** (рис. 1.3).

³⁴ Ім'я бібліотеки може бути будь-яким, проте для цього циклу комп'ютерних практикумів рекомендовано як ім'я створюваної бібліотеки задати своє прізвище англійською мовою. Зверніть увагу, що на рис. 1.2 ім'я бібліотеки задане як **YourSurname**, що можна перекласти як «ваше прізвище».

Рис. 1.3. Панель **Projects**, у якій відображається новостворений проект бібліотеки

3.1.1. Додавання бібліотеки умовних графічних позначень

Щоб додати до проекту порожню бібліотеку УГП, натисніть ПКМ на імені пакету бібліотек у панелі **Projects** та у розкритому меню оберіть пункт **Schematic Library**³⁵ (рис. 1.4). Після виконання цих дій до проекту буде додана нова бібліотека з ім'ям **Schlib1.SchLib**.

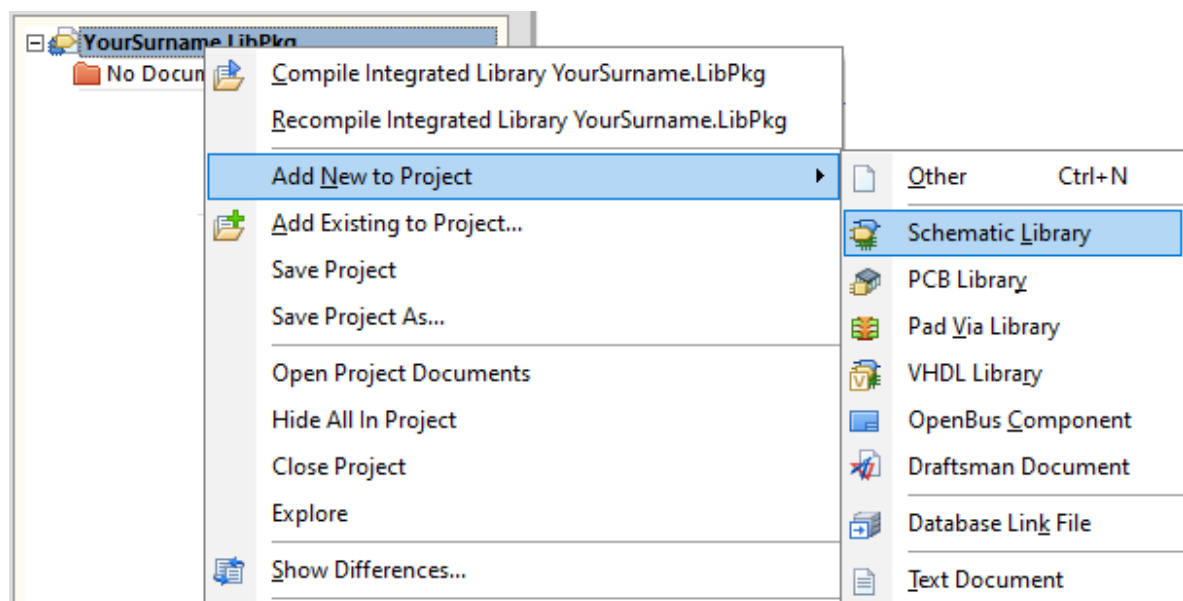


Рис. 1.4. Додавання бібліотеки УГП до проекту

Виділіть у панелі **Projects** додану бібліотеку УГП, натисніть ПКМ і оберіть команду **Save As**. За допомогою відповідного діалогу збережіть бібліотеку у каталозі проекту під іменем **Schematic_Component.Schlib**. У списку компонентів цієї бібліотеки буде присутній поки що єдиний компонент із назвою **Component_1**, який створюється за замовчуванням (очевидно, що у будь-якій бібліотеці має бути

³⁵ Або можна виконати команду **File** → **New** → **Library** → **Schematic Library**.

присутнім хоча б один компонент). Для того, щоб побачити перелік компонентів, що належать до бібліотеки, необхідно відкрити панель *SCH Library*, клацнувши ЛКМ на вкладці *SCH Library* у нижній частині бокової панелі *Projects* (рис. 1.5).

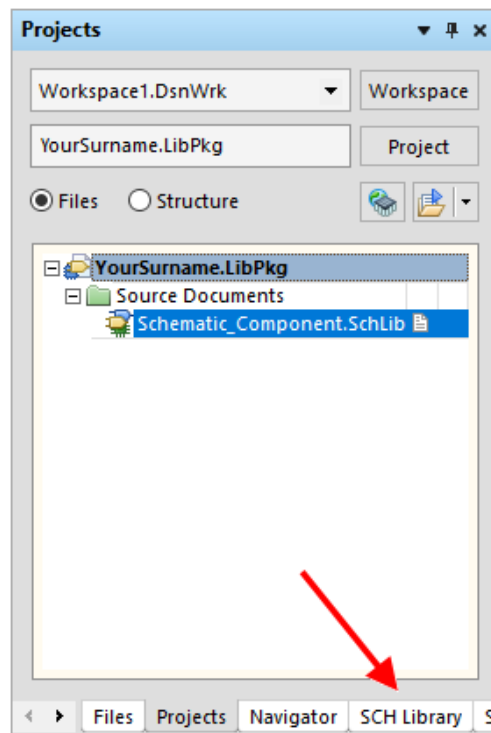


Рис. 1.5. Вкладка *SCH Library* у панелі *Projects*

Після того, як ви перейдете на вкладку *SCH Library*, буде відображена пуста робоча область компонента **Component_1**. Саме у цій графічній області надалі буде створюватись УГП компонента.

Важливо!

Зверніть увагу на сітку³⁶ робочої графічної області, у якій буде створюватись УГП компоненту. Можна помітити, що насправді сіток дві, а не одна: перша – пунктирна, друга – суцільна. Причому крок суцільної сітки вдвічі більший за крок пунктирної. Саме крок пунктирної сітки відповідає поточному значенню параметра *Grid*. Цей факт потрібно враховувати під час графічних побудов, оскільки якщо орієнтуватись на суцільну сітку, то розміри УГП врешті виявляться вдвічі більшими за необхідні.

³⁶ Спочатку ви можете і не бачити сітку, якщо ви налаштовуєте програму «з нуля». Це пов'язано із тим, що колір сітки за замовчуванням може бути таким самим, як колір фону, а отже, вона буде не помітною. Далі буде показано, яким чином змінити колір сітки.

3.1.2. Налаштування параметрів компонентів бібліотеки умовних графічних позначень

Відкрийте вікно властивостей **Library Component Properties** поки що єдиного компоненту бібліотеки **Component_1** за допомогою подвійного клацання ЛКМ на ньому. Це вікно розділене на кілька різних областей (рис. 1.6). Нас поки будуть цікавити дві: **Properties**, що містить системні атрибути компонента, та **Parameters**, у якій задають користувацькі параметри.

Library Component Properties

Properties

Default Designator: * ☐ Visible ☐ Locked

Default Comment: * ☐ Visible

Description:

Type: Standard

Library Link

Symbol Reference: Blank_Component

Graphical

Mode: Normal ☒ Lock Pins

☐ Show All Pins On Sheet (Even if Hidden)

☐ Local Colors

Parameters

Visible	Name	Value	Type
<input type="checkbox"/>	Family	*	STRING
<input type="checkbox"/>	Manufacturer	*	STRING
<input type="checkbox"/>	Model	*	STRING
<input type="checkbox"/>	PackageType	*	STRING
<input type="checkbox"/>	PurchaseURL	*	STRING
<input type="checkbox"/>	URLtoDatasheet	*	STRING
<input type="checkbox"/>	Value	*	STRING

Models

Name	Type	Description	Vault	Item R...	Revisi...
------	------	-------------	-------	-----------	-----------

Buttons: Add..., Remove..., Edit..., Add as Rule...

Buttons: Edit Pins..., OK, Cancel

Рис. 1.6. Вікно властивостей компоненту

Наразі в області **Properties** усі поля залиште без змін. Нас більше цікавитиме область **Parameters**. Можна помітити, що наразі вона є пустою. Кожен розробник сам вирішує, які параметри компонента будуть йому необхідні надалі під час роботи над проєктом. Назви параметрів і їх значення обираються самостійно. Можна порекомендувати використовувати такий перелік параметрів кожного окремого компоненту (див. рис. 1.6):

- **Family** – сімейство, до якого належить компонент (наприклад, резистор, мікросхема тощо);
- **Manufacturer** – виробник електронного компоненту;
- **Model** – назва компоненту;

- **PackageType** – тип корпусу;
- **URLtoDatasheet** – адреса інтернет-ресурсу, на якому розміщена інформація про цей компонент (datasheet);
- **PurchaseURL** – адреса інтернет-ресурсу, на якому можна купити цей компонент;
- **Value** – значення деякого фізичного параметру електронного компоненту (наприклад, для резистора може бути вказаний його опір, для конденсатора – ємність тощо).

Зверніть увагу, що у назвах параметрів не використовуються пробіли, а окремі слова пишуться разом. Це зроблено для того, щоб уникнути помилок. Наприклад, коли застосовується операція конкатенації різних параметрів в полі **Default Comment**, то зайві пробіли можуть обумовити помилку.

Зауваження!

В Altium Designer для компонентів передбачені також зарезервовані імена користувацьких параметрів (зокрема, *HelpURL*, *ComponentLink1URL* та *ComponentLink1Description*, які використовуються спільно). Це означає, що якщо ви саме так назвете параметр, то він буде оброблятися спеціальним чином. Наприклад, якщо ви створили параметр з іменем *ComponentLink1URL*, у якому задали посилання на деякий сайт в інтернеті або шлях до файлу на комп'ютері з інформацією про компонент, та параметр *ComponentLink1Description*, який містить назву цього посилання, то під час натискання ПКМ на компоненті у розкритому меню з'явиться пункт *Reference*, у якому буде відображена назва із параметру *ComponentLink1Description*. Натискання на цій назві призведе до відкриття самого посилання, що збережене в параметрі *ComponentLink1URL*.

Використовуючи кнопку **Add**, яка розташована безпосередньо внизу області **Parameters**, додайте той перелік параметрів, що вказаний на рис. 1.6. У стовпчику **Value** для усіх параметрів встановіть символ «*», який означає, що їх значення є пустими (поки що не задані).

Зверніть увагу, що після того, як були задані користувацькі параметри, у полі **Default Comment** з'явилась можливість із випадного списку обрати один із них (у форматі 'ім'я параметру'). Це означає, що в атрибуті **Default Comment** буде завжди

відображатись обраний параметр. До речі, для відображення кількох параметрів можна використовувати конкатенацію за допомогою знаку «+».

Важливо!

Усі нові компоненти, які створюються із використанням кнопки **Add** у субвікні *Components* вкладки *SCH Library*, завжди мають пусті користувацькі параметри. Але створювати нові пусті компоненти, параметри яких потім треба буде знову заповнювати вручну, не завжди зручно. Тому у цьому циклі комп'ютерних практикумів ми будемо використовувати такий підхід: одноразово створимо заготовку пустого компоненту із заздалегідь налаштованими ключовими користувацькими параметрами, після чого не будемо створювати у бібліотеці символів нові пусті компоненти, а будемо копіювати цей існуючий компонент, задаючи йому нове ім'я. Таким чином будуть копіюватись усі користувацькі параметри в новий компонент, а нам потрібно буде лише намалювати нове умовне графічне позначення і задати значення для існуючих параметрів. Таким чином у нашій бібліотеці усі компоненти будуть мати однаковий набір параметрів, однак із різними значеннями. Це зручно, зокрема, тому, що під час формування файлу специфікації параметри з однаковою назвою будуть відображатися в одному стовпчику.

Після того, як ви налаштували усі параметри, перейменуйте існуючий компонент, задавши нове ім'я **Blank_Component**. Для цього у полі *Symbol Reference* напишіть **Blank_Component**. Саме на основі цього базового компоненту надалі ми будемо створювати копії.

3.1.3. Додавання бібліотеки посадкових місць

Для того, щоб додати до проєкту бібліотеку посадкових місць, натисніть ПКМ на імені пакету бібліотек у панелі *Projects* та у розкритому меню оберіть пункт *PCB Library*³⁷ (див. рис. 1.4). Бібліотека завжди має містити хоча б одну модель ПМ, тому після створення у новій бібліотеці буде присутній порожній компонент з ім'ям **PCBCOMPONENT_1** (рис. 1.7).

³⁷ Або виконайте команду *File* → *New* → *Library* → *PCB Library*.

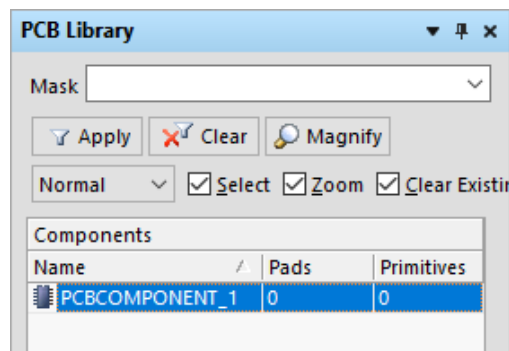


Рис. 1.7. Список компонентів у бібліотеці посадкових місць

Виділіть у панелі **Projects** додану бібліотеку посадкових місць, натисніть ПКМ і за допомогою команди **Save As** збережіть її у папці проєкту під іменем **PCB_Footprints.PcbLib**. Після цього у пакеті бібліотек має бути два вихідних файли (рис. 1.8).

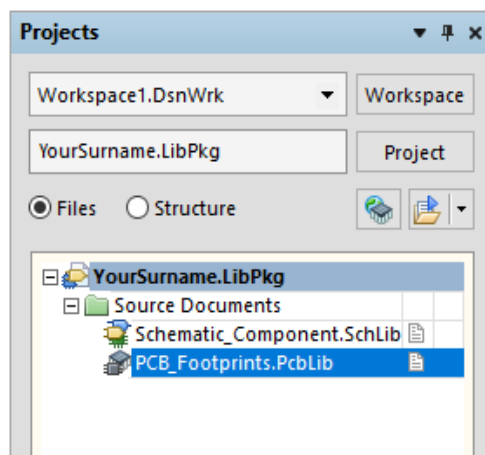


Рис. 1.8. Дерево документів нового проєкту

3.1.4. Компіляція бібліотеки

Для компіляції пакета бібліотек виконайте команду **Project** → **Compile Integrated Library <ім'я_бібліотеки>**. Це дозволить створити на основі вихідних бібліотек і моделей, які входять у пакет бібліотек, єдиний файл інтегрованої бібліотеки³⁸. Якщо під час компіляції будуть виявлені помилки, то попередження відображатимуться в панелі **Messages** (**View** → **Workspace Panels** → **System** → **Messages**). Щоб отримати більш детальну інформацію про помилку або перейти до

³⁸ Створити інтегровану бібліотеку також можна і на основі існуючого проєкту плати. Якщо вже була створена принципова схема деякого електронного пристрою, то у редакторі **Schematic** необхідно виконати команду **Design** → **Make Integrated Library**, причому спочатку будуть створені вихідні бібліотеки, а потім інтегрована.

компоненту, необхідно двічі клацнути на повідомленні у панелі **Messages**. Після виправлень необхідно перекомпілювати інтегровану бібліотеку.

Нова інтегрована бібліотека автоматично буде додана в список встановлених бібліотек і стане доступною для використання через бічну панель **Libraries**.

Якщо усе зроблено правильно, то у папці проєкту, яка була вказана під час збереження, буде присутній перелік папок і файлів, як показано на рис. 1.9. Це зручно, коли усі складові проєкту бібліотеки зберігаються в одному місці.

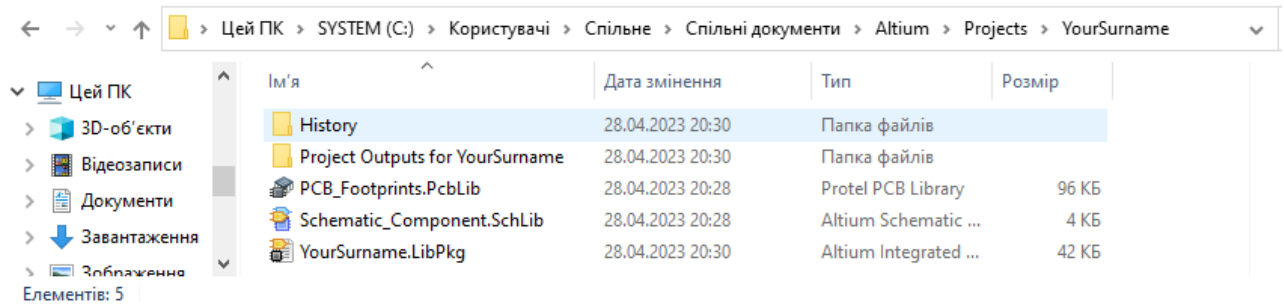


Рис. 1.9. Перелік папок і файлів на диску після створення власного проєкту

Перевагами використання інтегрованих бібліотек є те, що під час їх компіляції або перекомпіляції завжди відбувається перевірка узгодженості виводів УГП та контактних майданчиків ПМ електронного компонента, а уся інформація компактно зберігається в одному файлі і є наочною.

3.2. Встановлення налаштувань програми

3.2.1. Встановлення глобальних налаштувань Редактора схем та Редактора символів

Для встановлення глобальних налаштувань, які впливатимуть як на Редактор схем, так і на Редактор символів, який використовується під час створення УГП, необхідно викликати вікно налаштувань **Preferences (DXP → Preferences)**.

Спочатку налаштуємо сітку. За замовчуванням сітка в Редакторі схем та Редакторі символів відображається білим кольором, що не є зручним, оскільки її не видно на білому фоні робочої області. У вікні **Preferences** знайдіть розділ **Schematic → Grids**. У полі **Grid Options** для параметру **Grid Color** встановіть бажаний колір сітки (рекомендується встановити синій колір). Також у цьому розділі можна встановити додаткові значення кроку сітки, які послідовно перемикатимуться за допомогою гарячої клавіші <G>. Для цього необхідно клацнути ПКМ на будь-якій вільній частині поля для налаштування кроку сітки і у розкритому меню обрати

Add GridSetting → ... (за потреби можна відредагувати додане значення кроку). Встановіть такі значення для метричної сітки, як показано на рис. 1.10.

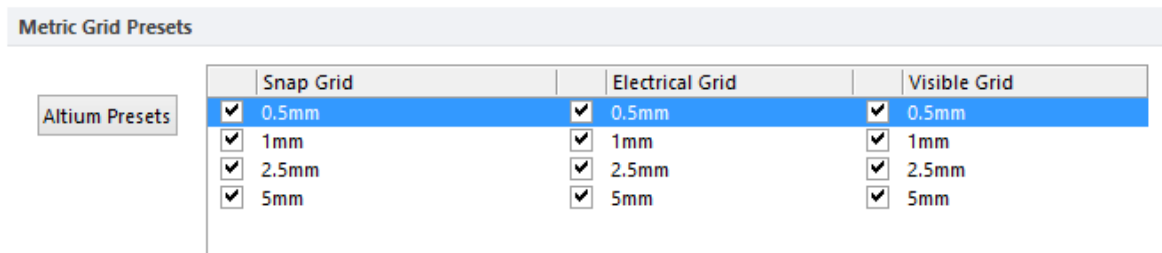


Рис. 1.10. Налаштування кроку сітки

Важливо!

Крок сітки *Snap Grid*, який налаштовується в області *Metric Grid Presets*, не може бути меншим 0,254 мм (0,01 дюйм), тоді як *Electrical Grid* та *Visible Grid* можуть бути меншими 0,254 мм. Водночас, якщо ви будете використовувати команду *Set Snap Grid* у Редакторі символів або Редакторі схем, то програма дозволить вам задати практично будь-який крок сітки *Snap Grid* і не генеруватиме помилку. Сітка *Snap Grid* – це сітка прив'язки курсора, тобто під час створення різних графічних примітивів (наприклад, ліній) курсор може бути встановлений тільки у вузлах цієї сітки.

За замовчуванням в **Altium Designer** використовується дюймова система вимірювання³⁹. Для того, щоб увімкнути метричну систему вимірювань, потрібно перейти в розділ *Schematic* → *Default Units* (рис. 1.11) і встановити галочку *Use Metric Unit System*. Для вибору міліметрів як одиниць вимірювання у випадному списку *Metric unit used* слід обрати рядок *Millimeters*.

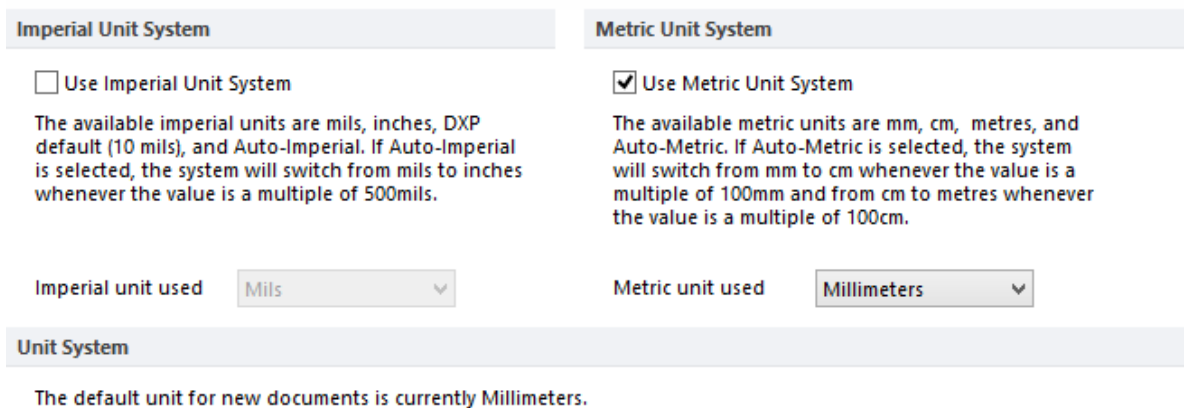


Рис. 1.11. Налаштування метричної системи вимірювань редакторів схем та символів

³⁹ Основними одиницями дюймової системи вимірювань в **Altium Designer** є дюйм (inch) та міл (mil): 1 дюйм = 1000 міл = 25,4 мм.

Далі налаштуємо відображення розділового знаку у позиційному позначенні компоненту на принциповій схемі. Варто відзначити, що закордонні стандарти регламентують як розділовий знак у позиційному позначенні компоненту, що складається із декількох частин, використовувати двокрапку та літеру, тоді як українські стандарти – крапку і цифру. Щоб використовувати у власних проєктах як розділовий знак крапку і цифру, необхідно перейти в розділ ***Schematic*** → ***General***. У випадному списку ***Alpha Numeric Suffix*** потрібно обрати рядок ***Numeric, separated by a dot '.'***.

Перед початком роботи зі Редактором схем та Редактором символів необхідно також налаштувати «примітиви». В **Altium Designer** цей термін позначає всі графічні елементи, з яких складаються схеми, плати і кресленики: лінії, кола, текстові рядки, виводи УГП тощо. Однією з властивостей примітивів (наприклад, ліній, дуг тощо) є товщина. Важливо зауважити, що у Редакторі схем та Редакторі символів товщина не задається чисельним значенням. Замість цього користувачам пропонується чотири попередньо налаштованих і незмінних значення: ***smallest***, ***small***, ***medium*** і ***large***. Наприклад, товщина виводу (Pin) має значення ***small***, причому для виводу це значення змінити неможливо. Для ліній (Line) рекомендується також встановити значення ***small***. Тоді як для шини (Bus) встановлюють значення ***medium***.

Для налаштування примітивів необхідно перейти у розділ ***Schematic*** → ***Default Primitives*** (рис. 1.12). У вікні цього розділу внизу розташовані вкладки ***Mils*** і ***MMs***. Оскільки ми працюємо в метричній системі, тому спочатку потрібно зробити активною вкладку ***MMs***.

Далі у списку ***Primitive List*** необхідно вибрати пункт ***Drawing Objects***. Для примітивів ***Arc***, ***Bezier***, ***Ellipse***, ***Elliptical Arc***, ***Line***, ***Pie***, ***Polygon***, ***Rectangle*** і ***Round Rectangle*** перевизначте товщину ліній значенням ***small*** (для того, щоб відкрити властивості заданого примітиву, необхідно подвійно клацнути ЛКМ на ньому) та встановіть чорний колір. Після цього у списку ***Primitive List*** оберіть пункт ***Library Objects***, відкрийте вікно властивостей примітиву ***Pin*** та встановіть довжину виводу 5 мм (параметр ***Length*** у полі ***Graphical***).

Зверніть увагу, що **Altium Designer** підтримує багаторівневу операцію скасування, дозволяючи скасувати кілька попередніх дій. Кількість дій, які можна скасувати командою ***Undo***, задається користувачем і обмежується лише об'ємом оперативної пам'яті комп'ютера. Відповідний параметр можна налаштувати у розділі ***Schematic*** → ***Graphical Editing***.

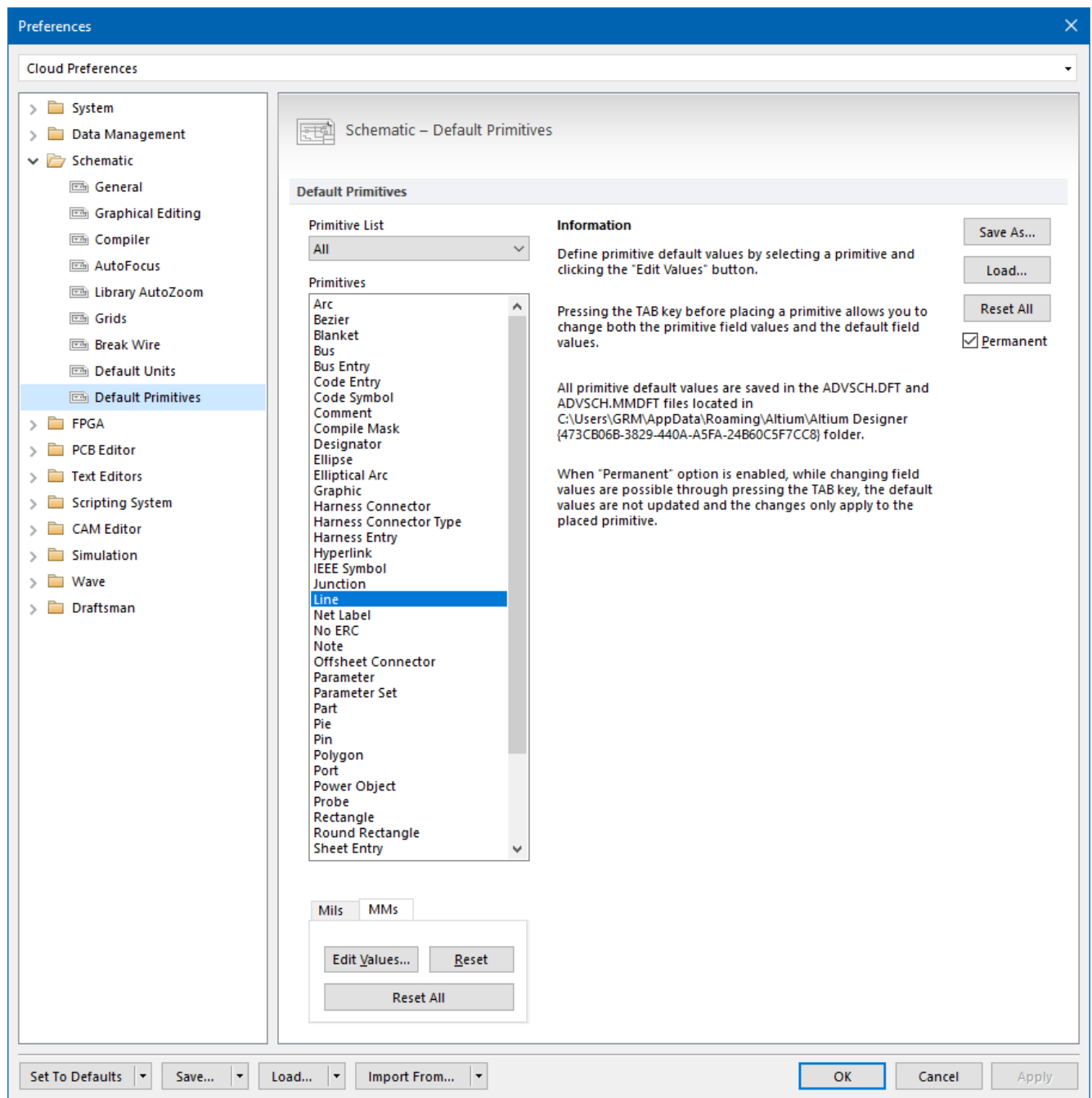


Рис. 1.12. Вікно налаштувань примітивів редакторів схем та символів

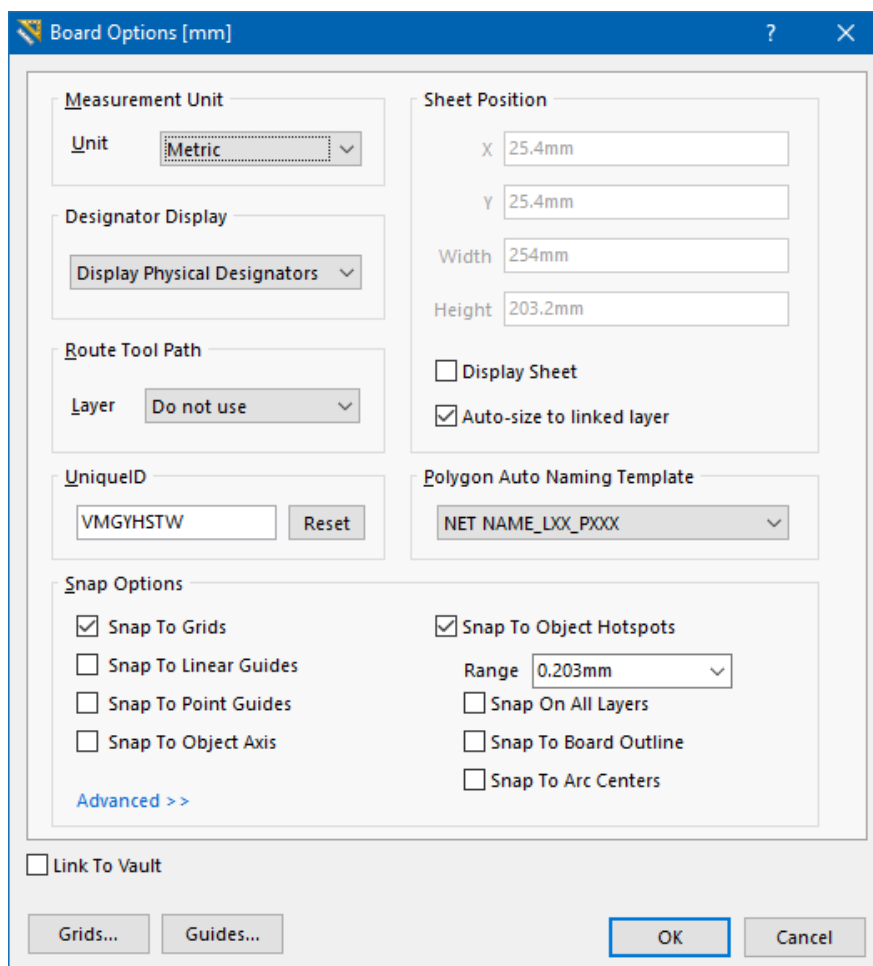
3.2.2. Встановлення глобальних налаштувань Редактора друкованих плат та Редактора посадкових місць

Для того, щоб встановити глобальні налаштування Редактора друкованих плат та Редактора посадкових місць, виконують команду ***DXP*** → ***Preferences***, після чого у вікні ***Preferences*** у списку ліворуч обирають пункт ***PCB Editor*** (див. рис. 1.12). Під час налаштування цих редакторів необхідно враховувати таку особливість: система одиниць вимірювань є властивістю не редактора, а самого документа (або плати, або посадкового місця).

Зауваження!

Оскільки система одиниць вимірювань є властивістю не Редактора друкованих плат / Редактора посадкових місць, а документа, тому рекомендується виконувати налаштування у пункті *PCB Editor* вікна *Preferences* при відкритому РСВ-документі (або відкритому документі бібліотеки посадкових місць) з виставленою в останньому метричною системою вимірювань.

Тому спочатку активуйте (або відкрийте) створений раніше файл бібліотеки посадкових місць **PCB_Footprints.PcbLib**, подвійно клацнувши на ньому ЛКМ у боковій панелі *Projects*. Для того, щоб встановити метричні одиниці вимірювань, необхідно зайти в меню *Tools* → *Library Options* і у області *Measurement Unit* із випадного списку *Unit* вибрати *Metric* (рис. 1.13).

Рис. 1.13. Вікно *Board Options*

Після того, як ви налаштували метричну систему вимірювання у документі бібліотеки посадкових місць, знову відкрийте розділ *PCB Editor* у вікні *Preferences*.

За замовчуванням у Редакторі друкованих плат / Редакторі посадкових місць для малювання контурів ПМ використовується лінія товщиною 0,254 мм (10 mil). Призначимо для ліній контурів нову товщину 0,2 мм. Для цього у розділі **PCB Editor** оберіть **Defaults**, після чого в правій частині у списку **Primitive Type** оберіть примітив **Track** (траса / лінія контурів) та активуйте вікно його параметрів подвійним натисканням ЛКМ (рис. 1.14). У вікні **Track** встановіть товщину лінії **Width** рівною 0,2 мм.

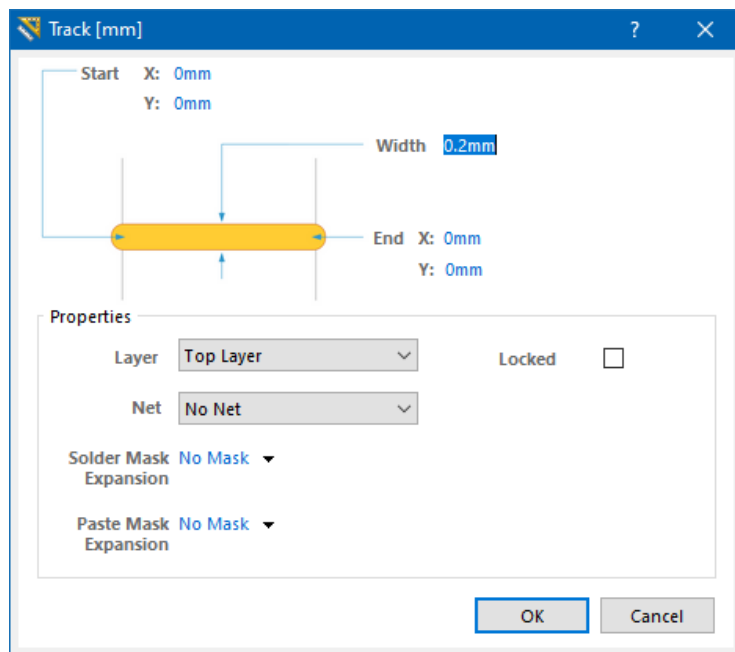


Рис. 1.14. Вікно властивостей примітиву **Track**

Зауваження!

Зверніть увагу, що серед примітивів у списку **Primitive Type** немає **Line** (лінії), хоча лінії використовуються в Редакторі посадкових місць для малювання контурів електронних компонентів. Так от, параметри примітиву **Track**, який тут налаштовується, будуть використовуватись для ліній у Редакторі посадкових місць, а також для ліній та трас під час ручного трасування у Редакторі плат.

3.2.3. Встановлення локальних налаштувань Редактора символів

Відкрийте створений раніше документ бібліотеки символів **Schematic_Component.Schlib**. Для налаштування його властивостей необхідно виконати команду **Tools** → **Document Options**. У вікні **Schematic Library Options**, що з'явиться, у вкладці **Library Editor Options** можна задати орієнтацію документа, колір фону, розмір робочої області тощо. А у вкладці **Units** можна змінити одиниці системи вимірювання.

Зауваження!

У вкладці *Units* вікна *Schematic Library Options* має бути активною та ж система вимірювань (у нашому випадку – метрична), що налаштована в глобальних параметрах. Зазвичай після завдання метричної системи в глобальних налаштуваннях усі нові документи автоматично будуть мати як основну метричну систему. Якщо ж з деяких причин так не відбулось, то встановіть у налаштуваннях документу *Schematic_Component.Schlib* також метричну систему. Тільки у такому випадку працюватиме перемикання кроку сітки за допомогою гарячої клавіші <G>. Якщо ж глобальна система одиниць вимірювань відрізнятиметься від локальної, то крок сітки в документі перемикатись не буде.

3.2.4. Встановлення локальних налаштувань Редактора посадкових місць

Знову активуйте документ бібліотеки посадкових місць *PCB_Footprints.PcbLib*.

Наразі ми налаштуємо тільки один параметр – це режим відображення сітки. Для цього виконайте команду *Tools* → *Grid Manager*. У вікні, що з'явиться, натисніть кнопку **Menu** у лівому нижньому куті та виберіть пункт *Properties*. Відкриється нове вікно, представлене на рис. 1.15.

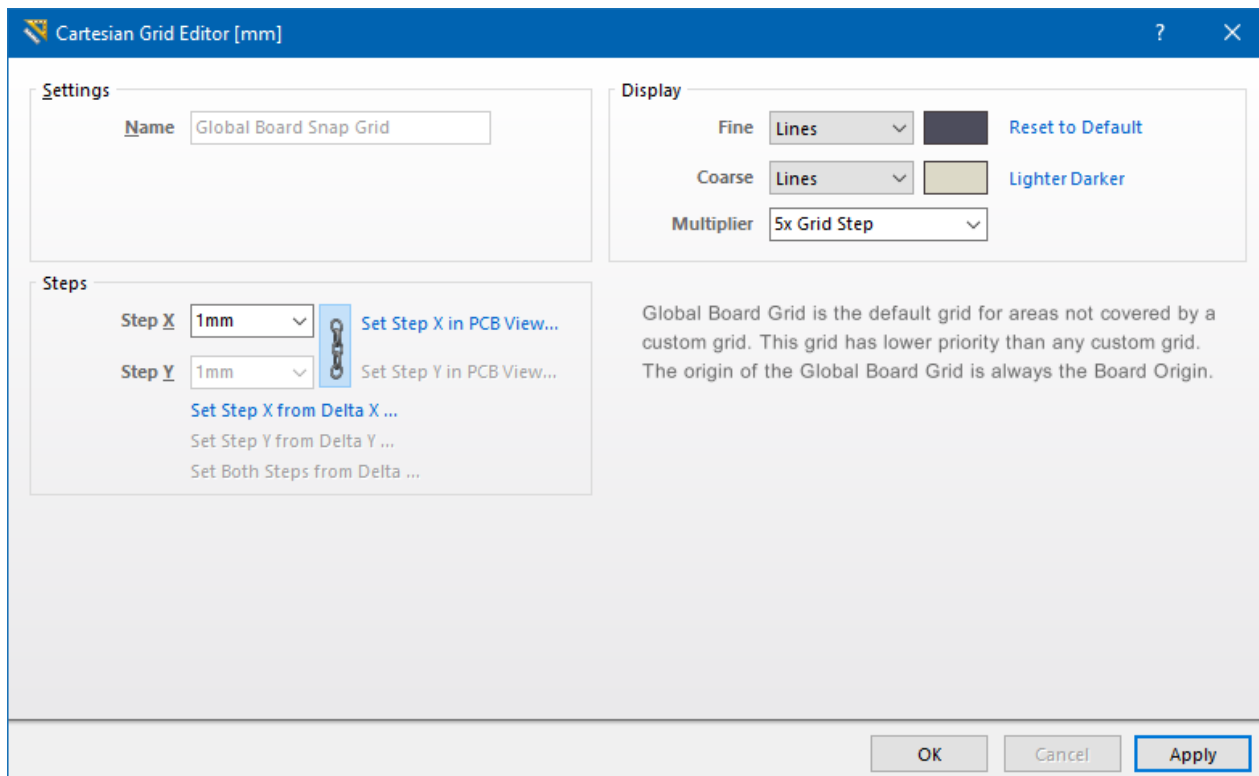



Рис. 1.15. Налаштування відображення сітки в Редакторі посадкових місць

Колір відображення для дрібної сітки встановлюється у пункті *Fine*, для збільшеної сітки – *Coarse*. Зверніть увагу на пункт *Multiplier*, який визначає, у скільки разів сітка *Coarse* буде більшою за *Fine*. Встановіть такі параметри сітки (крок, колір та множник), як показано на рис. 1.15.

Важливо!

Поточний крок сітки відповідає сітці *Fine* і відображається кольором, визначеним під час налаштувань (в нашому випадку було обрано темно-сірий колір). Тоді як сітка *Coarse* завжди буде більшою у N разів (задається множником *Multiplier*) від поточної сітки і відображатиметься в нашому випадку кольором, близьким до бежевого. Тому, якщо на екрані ви спостерігаєте тільки світлу сітку, це означає, що вона більша за основну. Якщо малювати контури ПМ, орієнтуючись на цю сітку, то вони будуть більшими за необхідні у N разів. Тому для малювання завжди необхідно орієнтуватись на основну сітку, яка згідно заданих налаштувань має відображатись темно-сірим кольором. Щоб її побачити (у випадку, якщо відображається тільки збільшена сітка), наблизьте область малювання, затиснувши клавішу <Ctrl> і обертаючи колесо мишки.

Поточний крок сітки можна встановити, використовуючи наступну піктограму на панелі інструментів , або за допомогою комбінації клавіш <Shift + Ctrl + G>. Поточний крок сітки завжди відображається у лівому верхньому куті робочої області (параметр *Snap*) та додатково у нижньому рядку вікна програми, якщо у меню *View* увімкнений *Status Bar* (рис. 1.16).

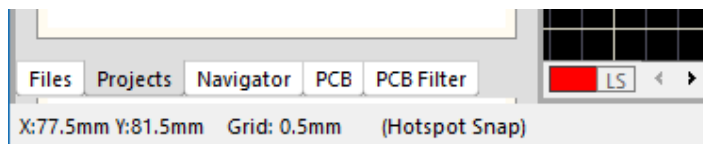


Рис. 1.16. Відображення поточного кроку сітки

Зауваження!

Комбінації клавіш і способи встановлення сітки для Редактора посадкових місць та Редактора друкованих плат відрізняються від тих, що застосовуються у Редакторі символів та Редакторі схем. Для більш детального ознайомлення з переліком гарячих клавіш дивіться [Додаток Г](#).

4. Питання для самоконтролю

1. Поясніть, яким чином налаштувати глобальні параметри **Altium Designer**?
2. Поясніть, як змінювати і налаштовувати сітку в Редакторі схем та Редакторі символів?
3. Поясніть, яким чином налаштовується товщина ліній примітивів у Редакторі схем та Редакторі символів?
4. Поясніть, яким чином можна змінити колір фону в Редакторі схем та Редакторі символів?
5. Поясніть, за що відповідає параметр *Alpha Numeric Suffix*?
6. Поясніть, які існують види бібліотек в **Altium Designer**?
7. Опишіть, як створити власну бібліотеку в **Altium Designer**?
8. Що собою представляє інтегрована бібліотека?
9. З якою метою виконують компіляцію бібліотеки?
10. Опишіть основні системні атрибути компонентного модуля.
11. Поясніть, яким чином створюють користувацькі параметри компонентного модуля?
12. Поясніть, яким чином налаштувати локальні параметри Редактора плат або Редактора посадкових місць?

Комп'ютерний практикум № 2.

РОЗРОБКА КОМПОНЕНТНИХ МОДУЛІВ ПАСИВНИХ ДИСКРЕТНИХ ЕЛЕКТРОННИХ КОМПОНЕНТІВ

Мета роботи: створити умовні графічні позначення та посадкові місця різного типу для резисторів, конденсаторів та котушок індуктивності; ознайомитись із типами корпусів пасивних дискретних електронних компонентів.

1. Теоретичні відомості

Пасивним називається електронний компонент, який не має внутрішніх джерел енергії і виконує або накопичення енергії (конденсатор, індуктивність), або її розсіювання (резистор). Пасивні електронні компоненти випускаються як у корпусах для монтажу в отвори, так і для поверхневого монтажу.

Резистор – це структурний елемент електричного кола, основне функціональне призначення якого чинити опір електричному струму з метою регулювання сили струму і напруги. Резистори бувають постійними (опір незмінний) та змінними (опір можна регулювати у певних межах).

Кілька десятиліть тому в Україні найбільш поширеними видами постійних резисторів загального призначення були радянські резистори МЛТ та ОМЛТ в корпусах для монтажу в отвори. Сьогодні ж найчастіше застосовуються такі резистори для монтажу в отвори: металооксидні плівкові (metal-oxide, серія МО), металоплівкові (metal film, серія MF, C2-33, C2-33H), вуглецеві (carbon film, серії CF, CFR) та дровові (wire wound, серія KNP). Завдяки розвитку технологій наразі користувачам (а не тільки виробникам плат) також стала доступною можливість використовувати у власних проєктах і резистори в корпусах для поверхневого монтажу.

Конденсатор – елемент з певним значенням ємності та малої омичної провідності, що застосовується для накопичення енергії електричного поля. Конструктивно конденсатор є системою, яка складається із двох провідників, що розділені діелектриком. Серед найбільш поширених типів конденсаторів можна виділити керамічні та електролітичні.

Котушка індуктивності – елемент, що володіє значною індуктивністю за відносно малої ємності та малого активного опору. Такий елемент здатен накопичувати магнітну енергію під час протікання електричного струму.

Пасивні компоненти є найпоширенішими виробами електронної промисловості та можуть складати більше половини всіх елементів принципової схеми.

2. Завдання на комп'ютерний практикум

Створити компонентні модулі для наступних електронних компонентів: постійний резистор для монтажу в отвори серії **MF** потужністю 0,125 та 0,25 Вт; постійний резистор для поверхневого монтажу в корпусі **0805**; змінні резистори **CA9V10** та **WH148**; керамічний конденсатор серії **K10-17Б** для монтажу в отвори; керамічний конденсатор для поверхневого монтажу в корпусі **0805**; електролітичний конденсатор **Hitano ECR**; котушка індуктивності серії **KIG**.

Спочатку в Редакторі символів намалювати УГП електронних компонентів, що зображені на рис. 2.1, використовуючи вказані геометричні розміри. Врахуйте, що довжина виводів кожного із представлених УГП має становити 5 мм.

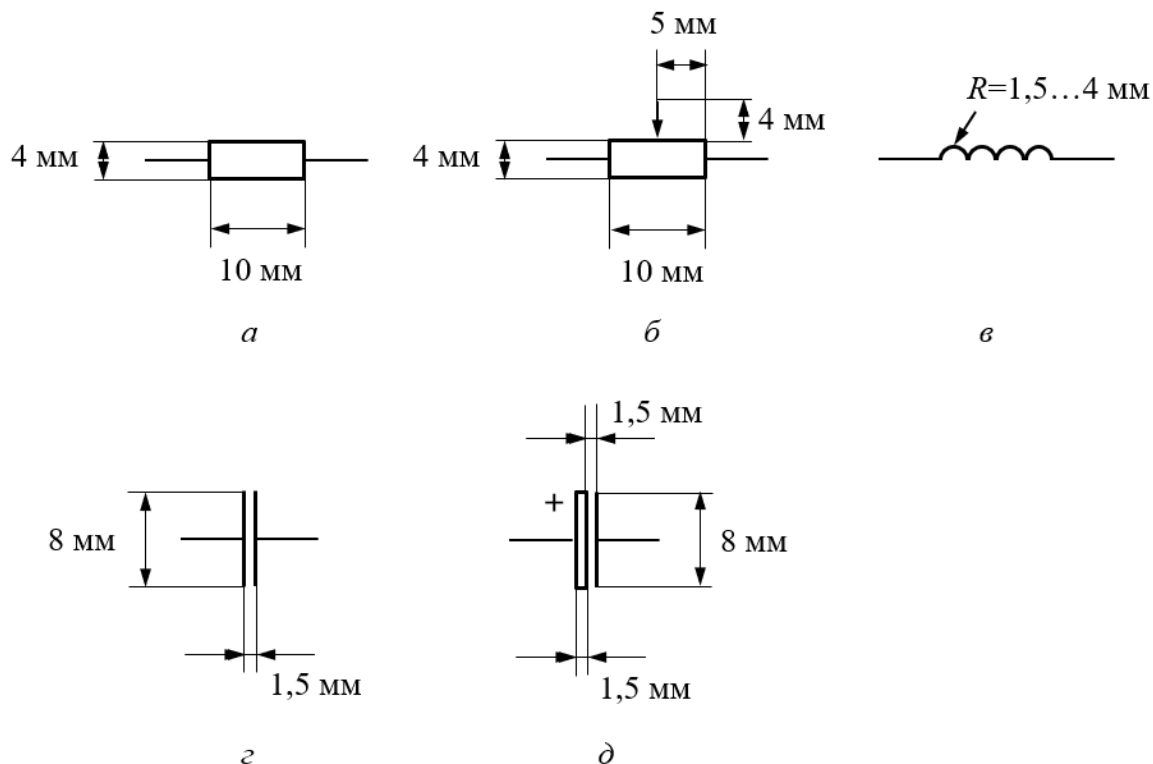


Рис. 2.1. УГП електронних компонентів із вказаними геометричними розмірами:
 а – постійний резистор, б – змінний резистор, в – котушка індуктивності, г – постійний конденсатор, д – полярний електролітичний конденсатор

Після цього створити ПМ для заданих електронних компонентів, використовуючи вказані на рис.2.2-2.8 геометричні розміри.

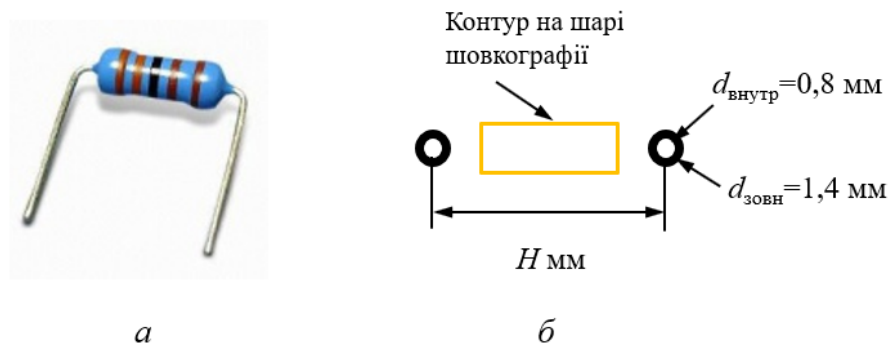


Рис. 2.2. Постійний резистор серії MF для монтажу в отвори: *а* – реальне зображення, *б* – геометричні розміри посадкового місця (відстань між контактними майданчиками H задають на основі даних із технічної документації⁴⁰)

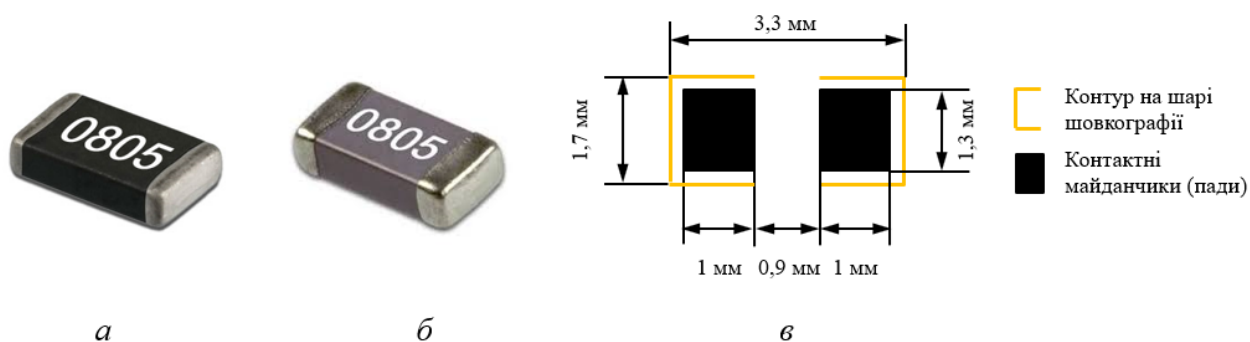


Рис. 2.3. Електронні компоненти в корпусі 0805 для поверхневого монтажу: *а* – реальне зображення SMD резистора, *б* – реальне зображення SMD конденсатора, *в* – геометричні розміри посадкового місця

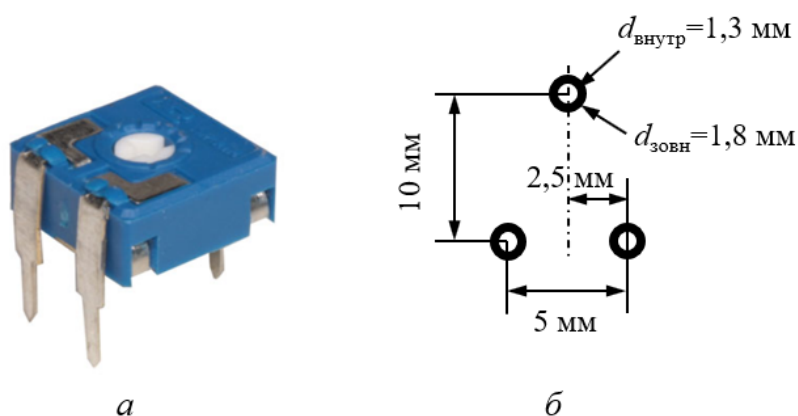


Рис. 2.4. Підлаштовувальний резистор CA9V10 для монтажу в отвори: *а* – реальне зображення, *б* – геометричні розміри посадкового місця

⁴⁰ Зазвичай ця відстань залежить від довжини корпусу резистора, який у свою чергу залежить від допустимої розсіюваної потужності.

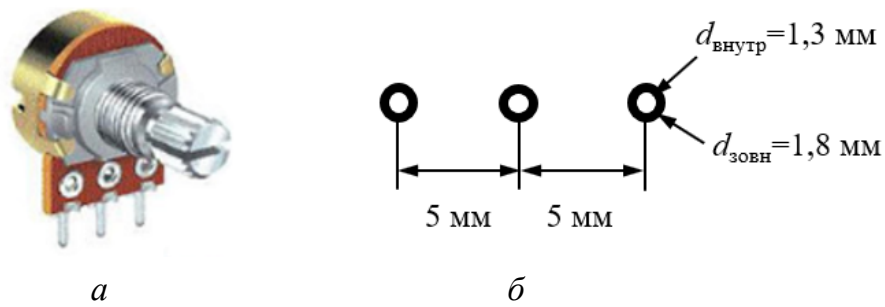


Рис. 2.5. Регулювальний резистор серії WH148 для монтажу в отвори:
a – реальне зображення, *б* – геометричні розміри посадкового місця

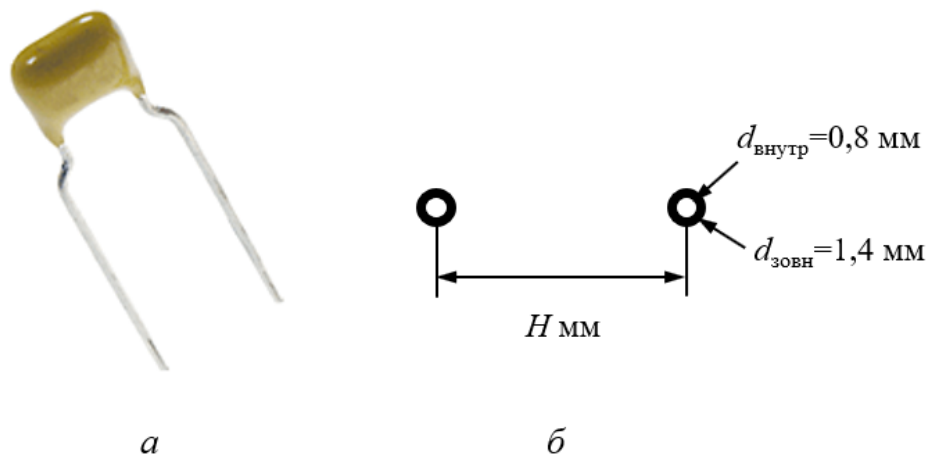


Рис. 2.6. Керамічний конденсатор серії K10-17Б для монтажу в отвори: *a* – реальне зображення, *б* – геометричні розміри посадкового місця (відстань між контактними майданчиками *H* задають на основі даних із технічної документації)

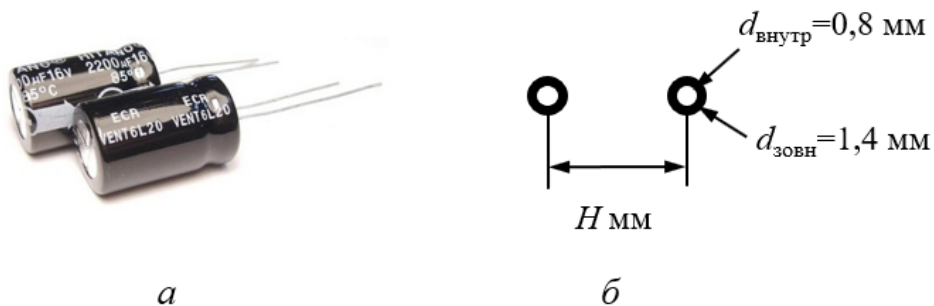


Рис. 2.7. Електролітичний конденсатор Hitano ECR для монтажу в отвори:
a – реальне зображення, *б* – геометричні розміри посадкового місця (відстань між контактними майданчиками *H* задають на основі даних із технічної документації)

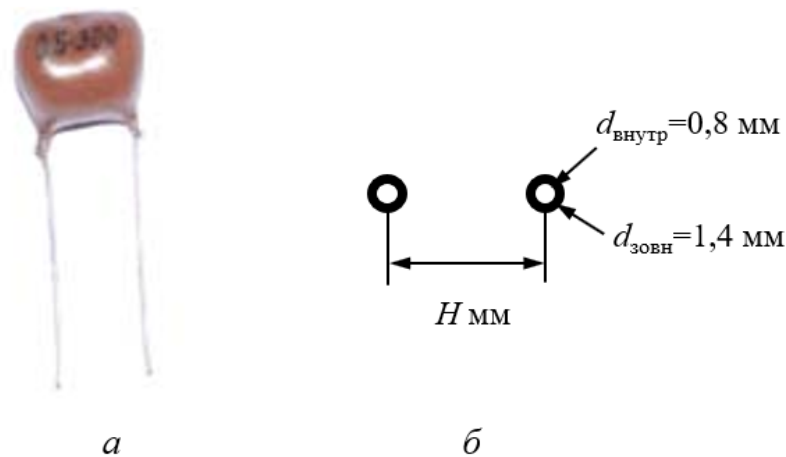


Рис. 2.8. Котушка індуктивності серії КІГ для монтажу в отвори: *а* – реальне зображення, *б* – геометричні розміри посадкового місця (відстань між контактними майданчиками H задають на основі даних із технічної документації)


3. Порядок виконання комп'ютерного практикуму

3.1. Створення УГП резисторів.

3.1.1. Створення УГП постійного резистора

Відкрийте програму **Altium Designer** і завантажте, якщо потрібно, файл власної бібліотеки з розширенням **.LibPkg**. У вкладці **Projects**, що знаходиться в лівій бічній панелі головного вікна програми, оберіть бібліотеку УГП **Schematic_Component.SchLib** та у випадку, якщо вона ще не активна, відкрийте її подвійним клацанням ЛКМ.

Перш ніж розпочати розробку нового символу, необхідно налаштувати робочу область, тобто обрати необхідні одиниці вимірювань і крок сітки. Ці установки виконуються у вікні **Library Editor Workspace**, яке викликається командою **Tools** → **Document Options**. Переконайтесь, що у вкладці **Units** активними є метричні одиниці вимірювання, а у вкладці **Library Editor Options** встановлені галочки навпроти пунктів **Snap** (сітка переміщення курсору в режимі графічної команди) та **Visible** (сітка, яка відображається на екрані).

Після завершення налаштувань необхідно на панелі інструментів програми натиснути піктограму , обрати пункт **Set Snap Grid** та встановити значення сітки основного кроку 1 мм (рис. 2.9).

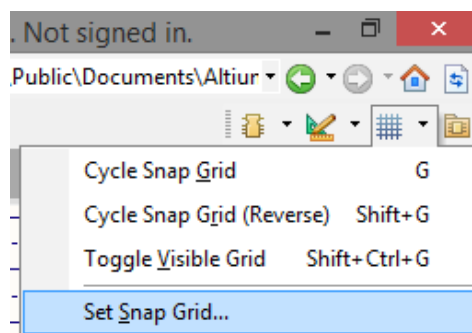


Рис. 2.9. Встановлення кроку сітки

Створіть копію компонента **Blank_Component**, яка за замовчуванням матиме назву **Blank_Component_1**.

Зауваження!

У цьому та наступних комп'ютерних практикумах ми будемо створювати нові компоненти методом копіювання та редагування заготовки **Blank_Component**, а не за рахунок використання команди *Tools* → *New Component*.

Далі у головній графічній області Редактора символів намалюйте контур резистора за допомогою команди меню *Place* → *Line*.

Зауваження!

Товщина і властивості лінії можуть бути змінені під час розміщення. Для цього необхідно натиснути клавішу <Tab>, щоб відкрилось вікно властивостей. Причому товщина лінії може бути заданою лише як *smallest*, *small*, *medium* та *large* (наприклад, *small* відповідає товщині 0,254 мм). Задати іншу товщину ліній в налаштуваннях неможливо.

Виводи резистора створіть за допомогою команди *Place* → *Pin*. Нумерація виводів має починатись з 1 і послідовно збільшуватись з кроком 1. Для лівого виводу задайте номер 1, а для правого – номер 2.

Важливо!

У межах одного елемента номери виводів не повинні повторюватись. Номери виводів задаються в полі *Designator*.

Зверніть увагу, що під час додавання виводів кінець, до якого на принциповій схемі приєднуватиметься електричний ланцюг, розташовується за замовчуванням праворуч і позначається ледь помітним хрестиком (рис. 2.10). Над виводом розміщується атрибут *Designator*, ліворуч – *Display Name*.

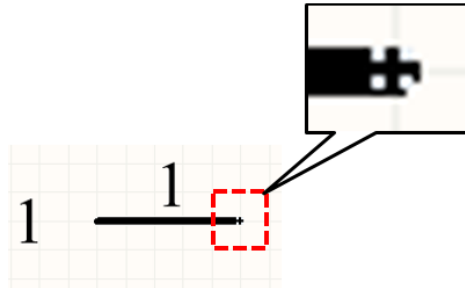


Рис. 2.10. Зображення виводу

Якщо вивід знаходитиметься праворуч від контуру компонента, то таке положення кінця виводу, до якого приєднується електричний ланцюг, є правильним, якщо ж вивід має знаходитись ліворуч від контуру, то його необхідно розвернути на 180° (рис. 2.11). Тобто, атрибут *Display Name* має бути завжди направленим у напрямку контуру.

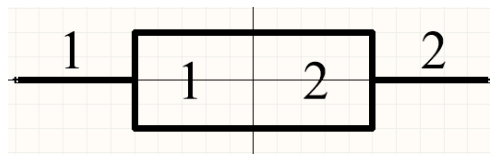


Рис. 2.11. Розташування виводів відносно контуру компонента

Рекомендація!

Якщо вивід уже встановлений, то для повороту на 90° потрібно спочатку його виділити, після чого натиснути клавішу <Пробіл>.

На УГП резисторів не прийнято відображати імена і номери виводів. Для того, щоб відключити видимість на схемі імен та номерів виводів, у діалоговому вікні властивостей виводів *Properties* для параметрів *Display Name* та *Designator* необхідно вимкнути опції *Visible* (рис. 2.12). Також у властивостях виводів перевірте, що їх довжина дорівнює 5 мм⁴¹.

⁴¹ У першому комп'ютерному практикумі ми задали за замовчуванням довжину виводу 5 мм у глобальних налаштуваннях. Тому наразі у властивостях виводів у полі *Graphical Length* має дорівнювати 5 mm. Якщо раптом це не так, то просто введіть у полі *Length* необхідне значення.

Display Name	<input type="text"/>	<input type="checkbox"/> Visible
Designator	<input type="text" value="1"/>	<input type="checkbox"/> Visible

Рис. 2.12. Опції видимості імені та номеру виводу

Після створення контуру та додавання виводів можна переходити до редагування властивостей усього компоненту. Кожен елемент на принциповій схемі повинен мати свої атрибути, до яких належить позиційне позначення, тип та номінал. Зокрема, над УГП резистора відображається латинська літера R, яка вказує на приналежність до класу резисторів, і порядковий номер, а під УГП – номінальний опір резистора.

Для налаштування атрибутів символу необхідно спочатку обрати компонент у списку **Components** панелі **SCH Library** і натиснути кнопку **Edit**. Відкриється вікно **Library Component Properties** (рис. 2.13).

Library Component Properties

Properties

Default Designator: ☒ Visible ☐ Locked

Default Comment: ☒ Visible

Part 1/1 ☒ Locked

Description:

Type:

Library Link

Symbol Reference:

Graphical

Mode: ☒ Lock Pins

☐ Show All Pins On Sheet (Even if Hidden)

☐ Local Colors

Parameters

Visible	Name	Value	Type
<input type="checkbox"/>	Family	*	STRING
<input type="checkbox"/>	Manufacturer	*	STRING
<input type="checkbox"/>	Model	*	STRING
<input type="checkbox"/>	PackageType	*	STRING
<input type="checkbox"/>	PurchaseURL	*	STRING
<input type="checkbox"/>	URLtoDatasheet	*	STRING
<input type="checkbox"/>	Value	1 кОм	STRING

Models

Name	Type	Description	Vault	Item R...	Revisi...
------	------	-------------	-------	-----------	-----------

Рис. 2.13. Вікно **Library Component Properties**

Введіть у полі **Default Designator** для резистору літеру **R** зі знаком запитання. Під час розміщення на принциповій схемі та процесу автоматичної нумерації знак


запитання після символу позиційного позначення буде замінюватись порядковим номером компонента (наприклад, R1, R2, R3 тощо).

У полі **Default Comment** вводиться коментар, який буде відображатися поруч з УГП електронного компонента, коли воно буде розміщене на схемі. Якщо залишити поле **Default Comment** незаповненим, то під час додавання компонента у схему там буде відображена його бібліотечна назва (параметр **Symbol Reference**). Нас це не влаштовує, оскільки на схемі поруч із УГП резистора має бути вказаний його номінал. Тому із випадного списку у полі **Default Comment** оберіть рядок **'=Value'**. Водночас у вікні **Parameters** у стовпчику **Value** для параметру **Value** задайте значення 1 кОм.

Переконайтеся, що налаштування видимості **Visible** полів **Default Designator** та **Default Comment** увімкнені.

У полі **Description** вводиться опис компонента. Цей текст можна використовувати під час пошуку в бібліотеці. Також він відображається в панелі **Libraries** і є підказкою про призначення та особливості компонента. Для створеного УГП введіть текст **Resistor**.

Введіть у полі **Symbol Reference** текст **Res MF 0.25** (фактично, таким чином ми перейменовуємо компонент).

Збережіть поточні результати, натиснувши піктограму  на панелі інструментів. Зображення УГП постійного резистору в бібліотеці повинно мати вигляд, як на рис. 2.14a.

Рекомендація!

Не забувайте зберігати поточні результати після створення кожного нового компонента.

Для створення компонентного модуля резистора MF потужністю 0,125 Вт можна виділити у вікні **Components** панелі **SCH Library** вже існуючий компонент **Res MF 0.25**, натиснути комбінацію клавіш **<Ctrl + C>**, а потім **<Ctrl + V>**. Буде створено копію першого компонента **Res MF 0.25_1**. Так зручно робити, щоб з самого початку не малювати УГП резистора для нового компонента, оскільки УГП постійного резистора є типовим. Тепер необхідно тільки змінити назву **Res MF 0.25_1** на **Res MF 0.125**, причому інші налаштування змінювати не потрібно.

Аналогічним чином методом копіювання створіть новий компонентний модуль резистора для поверхневого монтажу, задавши назву **Res SMD0805**.

Рекомендація!

Коли створюється нова бібліотека компонентів, то корисним підходом є максимальне застосовування принципу копіювання. Найбільш очевидним цей принцип є під час роботи із дискретними електронними компонентами (як пасивними, так і активними): резисторами, конденсаторами, транзисторами, діодами тощо. Кожна з груп дискретних компонентів має однотипні УГП, тому немає сенсу малювати їх кожен раз під час створення нового компонентного модуля. Набагато простіше і швидше скопіювати існуючий та відредагувати у ньому один або кілька параметрів (а також, можливо, замінити ПМ), щоб отримати новий бібліотечний компонент.

3.1.2. Створення УГП змінного резистора

Створіть копію компонента **Blank_Component**, після чого перейменуйте його на **Res CA9V10**. УГП змінного резистору побудувати аналогічно описаному вище (див. рис. 2.1б). Довжина виводу над прямокутником – 4 мм. На верхньому виводі намалюйте лінію зі стрілкою, вістря якої створіть за допомогою команди **Place** → **Polygon** або **Place** → **Bezier**. Нумерація виводів: лівий – 1, правий – 2, верхній – 3.

Основні параметри вікна **Library Component Properties** заповніть подібно як для постійних резисторів. У полі **Description** введіть текст **Resistor Variable**.

Далі створіть копію компонента **Res CA9V10**, після чого перейменуйте її на **Res WH148**. Окрім цього для компонента **Res WH148** змініть нумерацію виводів на наступну: лівий – 1, правий – 3, верхній – 2.

Зауваження!

Різна нумерація виводів на УГП змінних резисторів CA9V10 та WH148 зроблена не просто так. Це пов'язано із тим, як розташовується середній вивід, що з'єднаний із рухомим контактом всередині корпусу кожного із цих резисторів. В корпусі змінного резистора CA9V10 середній вивід має номер 3 та знаходиться не на одній прямій із виводами 1 та 2, тоді як в корпусі змінного резистора WH148 середній вивід має номер 2 і розташовується на одній лінії посередині між виводами 1 та 3. Неврахування цього факту в майбутньому може призвести до того, що компонент буде неправильно підключений до електронних ланцюгів на платі, а отже, плата не функціонуватиме або працюватиме неправильно.

Остаточний вигляд УГП постійного і змінного резисторів зображений на рис. 2.14.

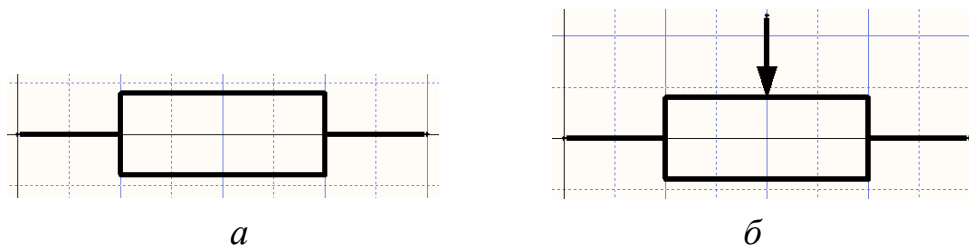


Рис. 2.14. Побудовані УГП резисторів: *а* – постійного, *б* – змінного

3.2. Створення УГП конденсаторів

3.2.1. Створення УГП керамічного конденсатора постійної ємності

Створіть новий компонент з іменем **Cap K10-17Б**. За допомогою команди **Place** → **Line** накресліть контур конденсатора – дві паралельні лінії довжиною 8 мм на відстані 1,5 мм одна від одної (див. рис. 2.12). За допомогою команди **Place** → **Pin** додайте виводи довжиною 5 мм. Нумерація виводів: лівий – 1, правий – 2. Кінцевий результат зображений на рис. 2.15а.

У полі **Default Designator** введіть атрибут типу (літера **C** зі знаком запитання), а в **Default Comment** оберіть рядок **'=Value'**. Водночас у вікні **Parameters** у стовпчику **Value** для параметру **Value** напишіть 1 нФ. У полі **Description** введіть текст **Capacitor**.

Створіть копію розробленого компонента керамічного конденсатора постійної ємності та змініть його ім'я на **Cap SMD0805**.

3.2.2. Створення УГП полярного електролітичного конденсатора

Додайте до бібліотеки новий компонент з іменем **Cap Hitano ECR**. УГП полярного електролітичного конденсатора побудуйте аналогічно описаному вище, використовуючи геометричні розміри, наведені на рис. 2.14. Результат зображений на рис. 2.15б.

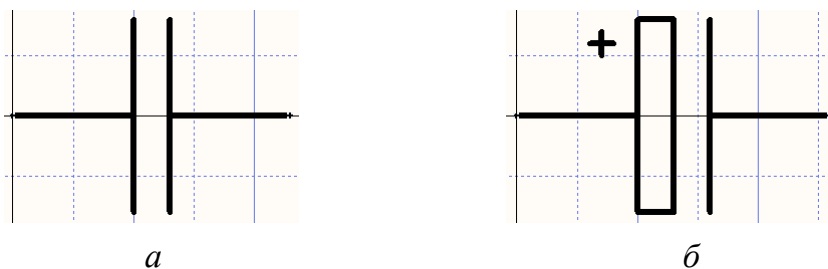


Рис. 2.15. Побудовані УГП конденсаторів: *а* – постійного, *б* – полярного електролітичного

Основні атрибути електролітичного конденсатора мають бути заповнені аналогічно до того, як це було зроблено для керамічного конденсатора. Тільки у полі **Description** введіть текст **Capacitor Polar**.

3.3. Створення УГП котушки індуктивності

Створіть новий компонент з іменем **Ind KIG**. Намалюйте чотири півкола радіусом 1,5 мм, для чого використовуйте команду **Place** → **Arc**. За допомогою команди **Place** → **Pin** додайте виводи довжиною 5 мм.

У полі **Default Designator** введіть атрибут типу (літера **L** зі знаком запитання), а в **Default Comment** оберіть рядок **'=Value'**. Водночас у вікні **Parameters** у стовпчику **Value** для параметру **Value** напишіть 1 мкГн. У полі **Description** введіть текст **Inductor**.

Остаточний вигляд УГП котушки індуктивності представлений на рис. 2.16.

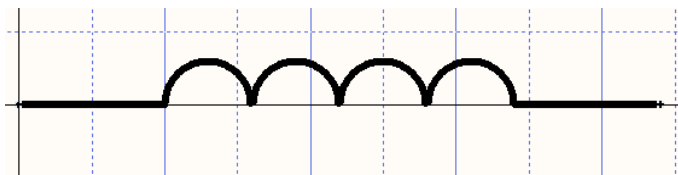


Рис. 2.16. Побудоване УГП котушки індуктивності без осердя

3.4. Створення посадкових місць резисторів для монтажу в отвори

3.4.1. Створення посадкового місця постійного резистору для монтажу в отвори

Створення та редагування ПМ виконується у Редакторі посадкових місць **PCB Library Editor** (рис. 2.17).

У вкладці **Projects** лівої бічної панелі головного вікна програми оберіть бібліотеку посадкових місць **PCB_Footprints.PcbLib** і якщо вона ще не активна, то відкрийте її подвійним клацанням ЛКМ. Після цього в лівій бічній панелі перейдіть на вкладку **PCB Library**.

Виконайте команду **Tools** → **Library Options**. У вікні, що відкриється, в полі **Unit** оберіть метричні одиниці вимірювання.

Переконайтеся, що панель шарів плати, яка знаходиться у нижній частині робочої області, містить перелік основних шарів: **Top Layer**, **Bottom Layer**, **Top Overlay**, **Bottom Overlay**, **Top Solder**, **Bottom Solder**, **Top Paste**, **Bottom Paste** тощо.

Якщо перелік неповний, то натисніть на цій панелі ПКМ, після чого у контекстному меню оберіть пункт **Layer Sets** і активуйте опцію **All Layers**.

Після встановлення усіх налаштувань можна переходити до створення ПМ, яке зазвичай складається із схематичного контуру компонента, що малюється за допомогою різноманітних примітивів (ліній, дуг тощо), та КМ, які пов'язані із фізичними виводами електронного компонента. У ПМ може зберігатись будь-яка інформація, включаючи габаритні розміри для встановлення, позначки кутів, спеціальні маркери для позиціонування корпусів мікросхем тощо.

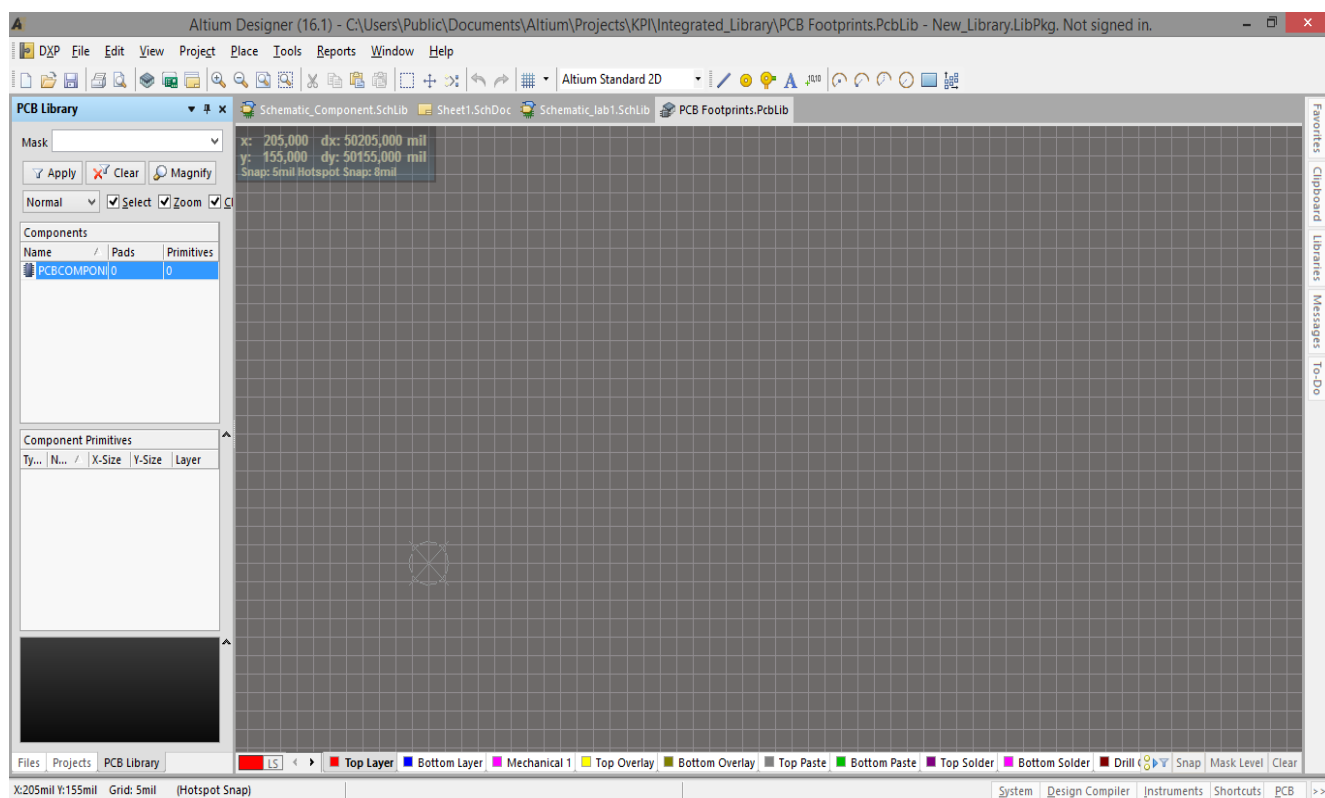


Рис. 2.17. Головне вікно Редактора посадкових місць


ПМ переважно створюються на верхніх шарах плати у Редакторі посадкових місць, незалежно від того, на якій стороні ДП компонент буде розміщений у майбутньому (хоча бувають і винятки). Верхні шари позначаються як **Top**, а нижні – **Bottom**. Зауважте, що більшість верхніх шарів мають відповідні протилежні шари: сигнальні (**Top Layer** та **Bottom Layer**), шовкографія (**Top Overlay** та **Bottom Overlay**), паяльна маска (**Top Solder** та **Bottom Solder**), апертура для пасти (**Top Paste** та **Bottom Paste**) тощо.

Об'єкти, що розташовуються на певних шарах, такі як КМ для компонентів поверхневого монтажу, паяльна маска, апертура для пасти, шовкографія тощо, під

час перенесення компонента в Редакторі друкованих плат на протилежну сторону плати, автоматично переносяться на відповідні протилежні шари.

Зауваження!

Зверніть увагу, що кожен шар має свій колір⁴². Цим кольором відображатимуться лінії, дуги та інші примітиви, що належать до того чи іншого шару. Наприклад, шар шовкографії має жовтий колір за замовчуванням. Це означає, що лінії контуру ПМ на цьому шарі будуть саме жовтого кольору.

У Редакторі посадкових місць оберіть єдиний поки що компонент з іменем **PCVCOMPONENT_1**. Спочатку встановимо КМ, після чого намалюємо контур посадкового місця резистора (хоча ніщо не заважає зробити навпаки, це справа зручності). Оберіть активним шаром **Top Layer**⁴³. Для розміщення КМ виконайте команду **Place** → **Pad** (послідовність гарячих клавіш: <P, P>) або натисніть кнопку  на панелі інструментів. Поруч з курсором з'явиться КМ. Перед встановленням першого КМ натисніть клавішу <Tab>, у результаті чого з'явиться вікно **Pad** (рис. 2.18), в якому можна налаштувати властивості КМ.

Перемикачі в області **Hole Information** використовуються для того, щоб задати форму отвору всередині КМ: **Round** – круглий, **Rect** – квадратний, **Slot** – у вигляді прорізи. Налаштування **Plated** в області **Properties** визначає, чи буде отвір металізованим. Ця інформація враховується під час підготовки плати до виробництва. Зокрема, для металізованих і неметалізованих отворів можуть створюватися різні файли для станка з ЧПУ (NC Drill).

Введіть налаштування, що показані на рис. 2.18, щоб створити круглий КМ з отвором: внутрішній діаметр 0,8 мм, зовнішній – 1,4 мм, номер – 1.

⁴² Загалом, кольорова палітра відіграє важливу роль під час проектування плати. Вона допомагає розробнику візуально аналізувати правильність розміщення різних примітивів на заданому шарі, оскільки кожен шар має власний колір. Колір будь-якого шару можна змінити у налаштуваннях.

⁴³ Насправді КМ електронних компонентів для монтажу в отвори можуть бути намальованими при будь-якому активному шарі, оскільки у їх властивостях усе одно буде задано параметр **Multi-Layer**, який означає, що ці КМ є наскрізними (проходять через усі шари плати). Проте візьміть собі за правило усе ж створювати КМ при активному шарі **Top Layer**. Така звичка буде особливо корисною під час створення планарних КМ, оскільки вони вже повинні бути намальованими на одному із сигнальних шарів.

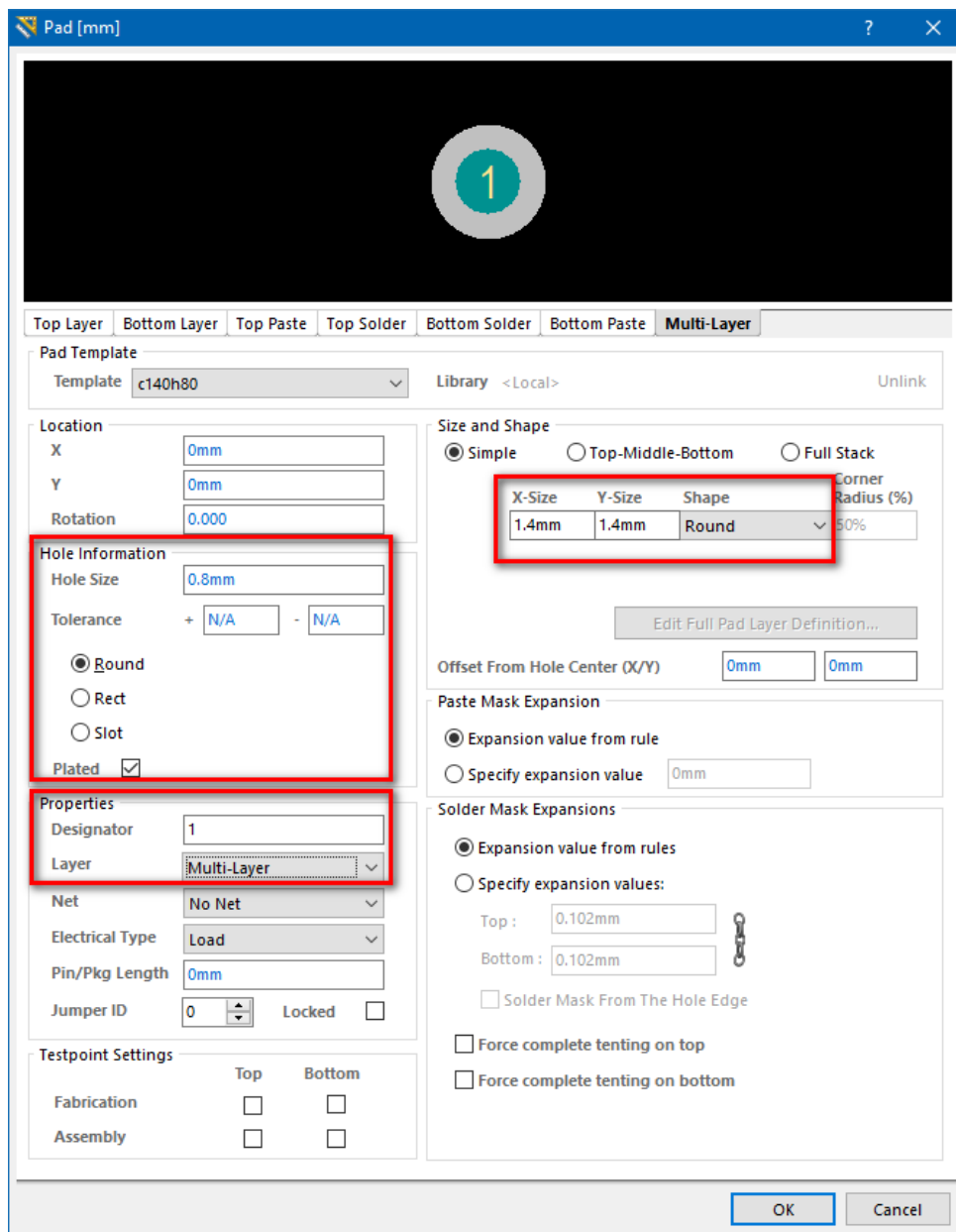


Рис. 2.18. Діалогове вікно налаштування КМ

Встановіть перший КМ прямо по центру робочої області, яка позначена колом із перехрестям.

Важливо!

ПМ в графічній області Редактора друкованих плат згодом під час розміщення буде позиціонуватись саме відносно цієї центральної точки. У нашому випадку це означатиме, по-перше, що центр першого КМ резистора буде завжди потрапляти у вузли координатної сітки графічної області, а, по-друге, центр КМ буде слугувати віссю обертання ПМ.

Після розміщення першого КМ поруч з курсором з'явиться наступний. Розмістіть другий КМ правіше від першого на відстані 10 мм по горизонталі.

Примітка!

Номер КМ збільшується автоматично.

Клацніть ПКМ або натисніть <Esc> для виходу з режиму розміщення КМ. У результаті в робочій області будуть розміщені два КМ, як на рис. 2.19, причому той, що ліворуч, повинен мати номер 1, а той, що праворуч, – 2.

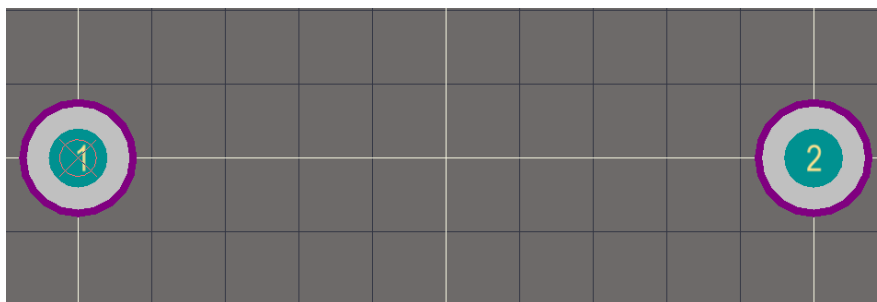


Рис. 2.19. Розташування КМ резистора у робочій області

Зауваження!

Під час створення ПМ вкрай важливо правильно і точно розміщувати КМ, оскільки внаслідок помилки, що пов'язана із неправильним позиціонуванням, згодом можуть виникнути проблеми з монтажем реального компонента на плату.

Після розміщення обох КМ намалюємо контур ПМ резистора для монтажу в отвори. Примітиви, з яких складається контур, зазвичай малюються на шарі шовкографії *Top Overlay*.

Встановіть поточним шар шовкографії *Top Overlay*. Накресліть посередині між КМ прямокутний контур резистору (див. рис. 2.21a), обравши команду *Place* → *Line*. Контур повинен мати розміри 2x4 мм, а лінія, якою він малюється, – товщину 0,2 мм.

Зауваження!

Товщина лінії може бути змінена під час малювання, якщо натиснути клавішу <Tab>.

Важливо!

На шарі шовкографії *Top Overlay* малюють те, що з точки зору того чи іншого розробника плати має бути нанесене спеціальною фарбою у вигляді непровідного рисунку на платі, наприклад, контури елементів⁴⁴, полярності конденсаторів і діодів, позиційні позначення елементів тощо. Перевантажувати шар шовкографії різноманітною графікою не рекомендується. Зазвичай на шарі шовкографії малюють тільки те, що необхідно під час монтажу і налаштування. Все інше малюють на графічних шарах. До таких шарів належать механічні шари, на яких малюють те, що необхідно відобразити на складальному кресленнику. Причому таких шарів може бути кілька: на одних можуть бути накреслені контури, що відповідають реальним геометричним розмірам електронних компонентів, елементи кріплень, позиційні позначення тощо; на інших – розміри і допоміжна графіка. За такого підходу, вмикаючи і вимикаючи окремі шари, можна отримувати різні варіанти креслеників, наприклад, складальні або кресленики самих плат. Також можна створити кілька варіантів графіки, зокрема, для різних масштабів.

Перейменуйте компонент **PCBCOMPONENT_1**, двічі клацнувши на ньому ЛКМ у вікні *Components*. У новому вікні, що відкриється, задайте для створеного ПМ ім'я **RES_CONST_L10.0D0.8** (рис. 2.20).

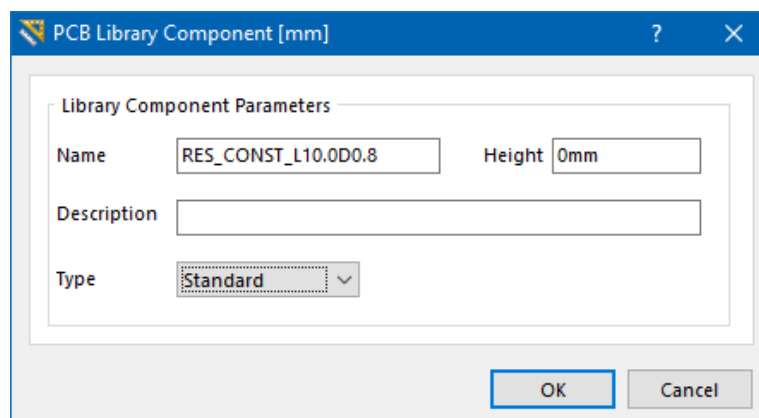


Рис. 2.20. Вікно перейменування ПМ

⁴⁴ У цих вказівках до виконання комп'ютерних практикумів контури ПМ завжди малюються на шарі *Top Overlay*. Причому під контуром мається на увазі набір ліній або геометричних фігур, що роблять малюнок ПМ на платі більш зручним для читання та розуміння. У такому випадку контур не обов'язково співпадає з геометричними характеристиками реального корпусу компоненту (найчастіше, взагалі не співпадає).

Аналогічно описаному вище створіть ПМ для резистора MF потужністю 0,125 Вт. Спочатку зробіть дублікат попереднього ПМ: для цього виділіть його у вкладці **PCB Library**, натисніть ПКМ та виберіть пункт **Copy** зі спливаючого меню. Далі у будь-якому пустому місці вікна **Components** натисніть ПКМ та виберіть пункт **Paste 1 Components**. З'явиться новий компонент з іменем **RES_CONST_L10.0D0.8-DUPLICATE**. Необхідно змінити його ім'я на **RES_CONST_L6.0D0.8**. Також для нового ПМ змініть відстань між центрами КМ із 10 мм на 6 мм та змістіть контур так, щоб він розташовувався рівно посередині між КМ.

Остаточний вигляд ПМ постійних резисторів зображений на рис. 2.21.

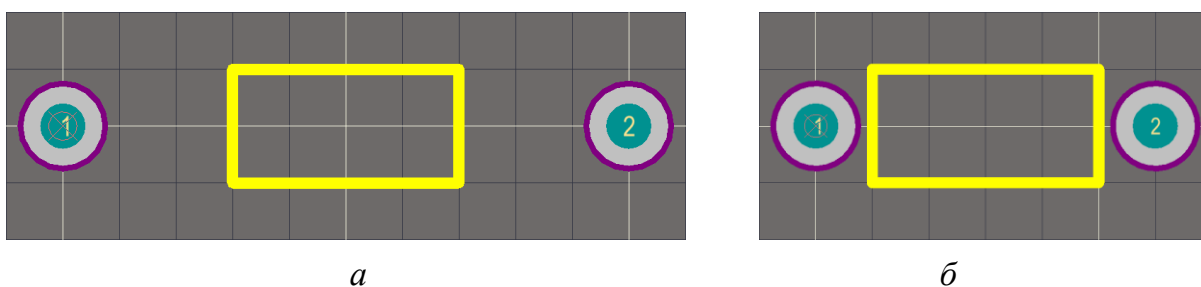


Рис. 2.21. Посадкове місце постійного резистору: *a* – MF 0,25, *б* – MF 0,125

3.4.2. Створення посадкового місця підлаштовувального резистору для монтажу в отвори

Створіть нову заготовку ПМ. Для цього у будь-якому пустому місці вікна **Components** натисніть ПКМ та у спливаючому меню виберіть пункт **New Blank Component**⁴⁵. У новій робочій області Редактора посадкових місць оберіть поточним шаром **Top Layer**, після чого встановіть три КМ згідно заданих геометричних розмірів (див. рис. 2.4): внутрішній діаметр – 1,3 мм, зовнішній – 1,8 мм. Нумерація КМ має відповідати тій, що представлена на рис. 2.22.

Далі активуйте шар **Top Overlay** та намалюйте 1) прямокутну область розмірами 11x10 мм, що охоплює КМ (однак врахуйте, що лінії прямокутника перериваються на КМ), 2) коло діаметром 4 мм в центрі ПМ (за допомогою команди **Place** → **Full circle**) та 3) вертикальну лінію, що ділить коло навпіл. Оберіть товщину ліній 0,2 мм. Остаточний результат зображений на рис. 2.22.

Збережіть ПМ під іменем **RES_VAR_CA9V10**.

⁴⁵ Створити нову заготовку ПМ також можна й у інший спосіб: із меню **Tools** виконати команду **New Blank Component**.

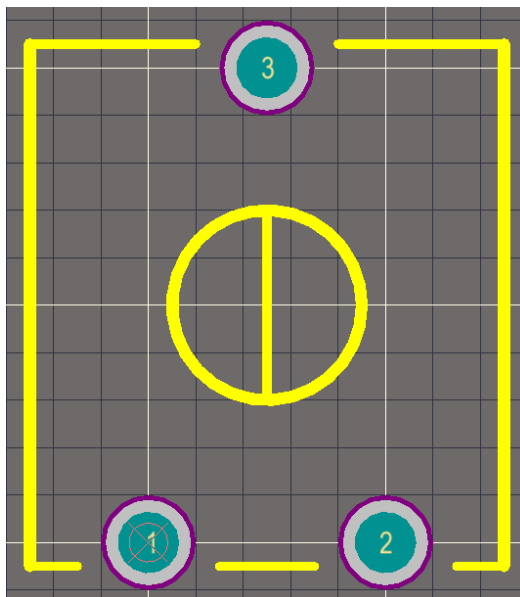


Рис. 2.22. Посадкове місце підлаштовувального резистору CA9V10

3.4.3. Створення посадкового місця регулювального резистору для монтажу в отвори

Створіть нову заготовку ПМ. Аналогічно пункту 3.4.2 розробіть ПМ для регулювального резистору, використовуючи геометричні розміри та нумерацію, що показані на рис. 2.5б: відстань між КМ – 5 мм, внутрішній діаметр – 1,3 мм, зовнішній – 1,8 мм. У властивостях першого КМ оберіть для параметру *Shape* значення *Rectangular*. Нумерація КМ має відповідати тій, що представлена на рис. 2.23.

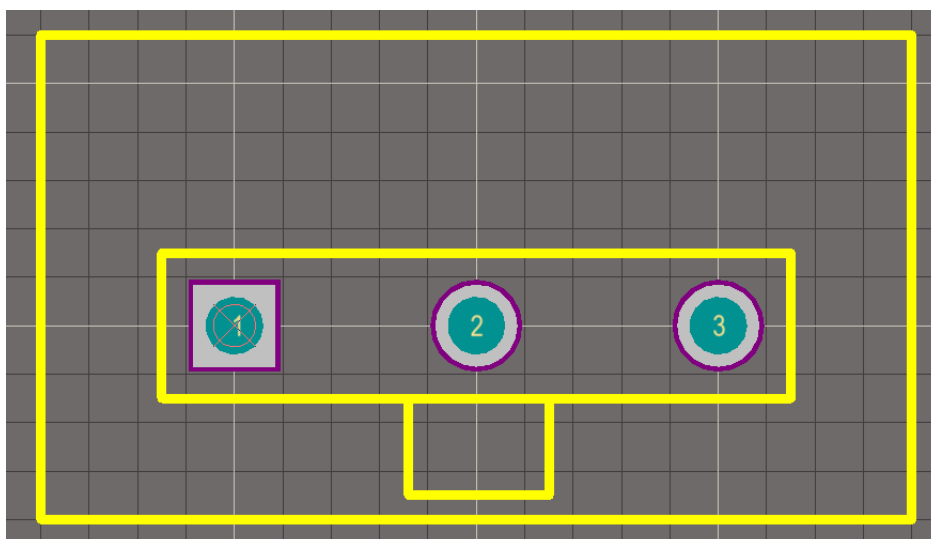


Рис. 2.23. Посадкове місце регулювального резистору WH148

Намалюйте контур ПМ розмірами 18x10 мм. Зауважте, що нижня лінія контуру має бути зміщена на 4 мм відносно центрів отворів (відповідно верхня лінія контуру

буде зміщена на 6 мм). Всередині головного контуру довільно намалюйте ще один, який схематично вказуватиме, з якої сторони знаходиться вал регулювального резистору. У результаті має бути створене ПМ, яке зображене на рис. 2.23.

Збережіть ПМ під іменем **RES_VAR_WH148**.

Зауваження!

Зверніть увагу, що для цього ПМ ми встановили прямокутну форму тільки першого КМ⁴⁶. Так зроблено спеціально, оскільки існує правило, що тільки перший КМ посадкового місця будь-якого електронного компонента, що призначений для монтажу в отвори, може мати прямокутну форму. Це дає змогу завдяки візуальному огляду зрозуміти, де перший КМ того чи іншого ПМ, адже фізично на платі номерів КМ видно не буде. Завдяки цьому з'являється можливість правильно встановити електронний компонент у отвори, щоб номери виводів співпадали із номерами отворів. Для ПМ регулювального резистору WH148 було б достатньо вказати тільки прямокутну форму першого КМ, але додатково ми на шарі шовкографії вказали сторону, де знаходиться вал резистора, що підвищить вірогідність його встановлення правильним чином.

3.5. Створення посадкових місць конденсаторів для монтажу в отвори

3.5.1. Створення посадкового місця керамічного конденсатора для монтажу в отвори

Створіть нову заготовку ПМ. За допомогою команди *Place* → *Pad* розмістіть два КМ (внутрішній діаметр – 0,8 мм, зовнішній – 1,4 мм), відстань між центрами яких має становити 5 мм. Задайте номери КМ: лівий – 1, правий – 2.

Накресліть прямокутний контур із закругленими кряями, використовуючи лінії та дуги товщиною 0,2 мм. Контур повинен мати габаритні розміри 7,2x2,2 мм та бути намальованим на шарі шовкографії *Top Overlay*. Остаточний результат представлено на рис. 2.24.

Збережіть створене ПМ у власну бібліотеку під іменем **CAP_CERAM_L5.0D0.8**.

⁴⁶ Так ми могли б зробити й для усіх ПМ, що були створені раніше.

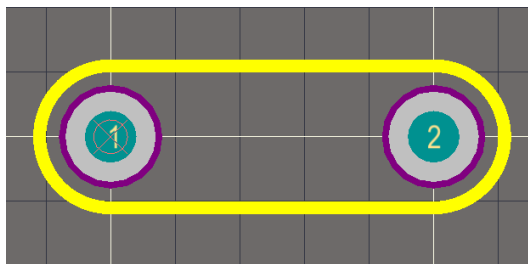


Рис. 2.24. Посадкове місце керамічного конденсатору серії K10-17Б

3.5.2. Створення посадкового місця полярного електролітичного конденсатора для монтажу в отвори

Створіть нову заготовку ПМ. За допомогою команди **Place** → **Pad** розмістіть у робочій області два КМ (внутрішній діаметр – 0,8 мм, зовнішній – 1,4 мм), відстань між центрами яких має становити 2,5 мм. Задайте номери КМ наступним чином: лівий – 1, правий – 2.

Накресліть на шарі **Top Overlay** коло діаметром 6 мм, що охоплює КМ (товщина лінії 0,2 мм). За допомогою команди **Place** → **String** встановіть маркер, що вказуватиме полярність (це знак «+» біля КМ з номером 1).

Остаточний результат зображений на рис. 2.25.

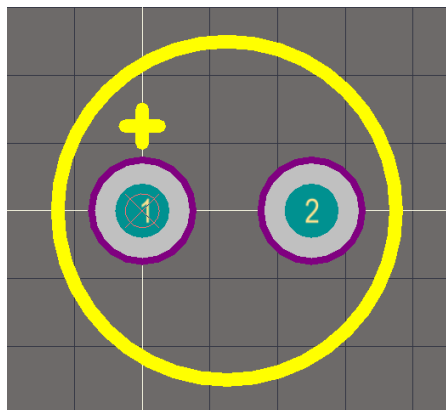


Рис. 2.25. Посадкове місце електролітичного конденсатору Hitano ECR

Збережіть створене ПМ у власну бібліотеку під іменем **CAP_POLAR_L2.5D0.8**.

3.6. Створення посадкового місця котушки індуктивності

Створіть нову заготовку ПМ. За допомогою команди **Place** → **Pad** розмістіть у робочій області два КМ (внутрішній діаметр – 0,8 мм, зовнішній – 1,4 мм), відстань

між центрами яких має становити 6 мм. Задайте номери КМ наступним чином: лівий – 1, правий – 2.

Далі зробіть поточним (активуйте) шар **Top Overlay** та накресліть контур, який складається з двох рядів по чотири півкола радіусом 0,5 мм (ряди мають бути встановлені між КМ на відстані 1,6 мм один від одного по вертикалі). Товщина ліній півкіл має становити 0,2 мм.

Остаточний результат зображений на рис. 2.26.

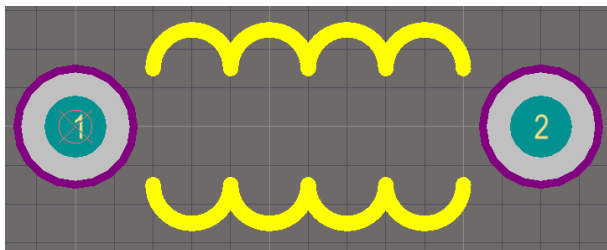


Рис. 2.26. Посадкове місце котушки індуктивності герметичної (КІГ), діапазон індуктивностей якої лежить в межах 1-500 мкГн

Збережіть створене ПМ у свою бібліотеку під іменем **IND_L6.0D0.8**.

3.7. Створення посадкового місця для корпусів пасивних SMD компонентів

Посадкове місце для корпусів таких SMD компонентів, як резистори та конденсатори (а також котушки індуктивності, світлодіоди тощо), є стандартним і не залежить від типу компонента. У [Додатку Д](#) наведені геометричні розміри найбільш популярних типів корпусів пасивних SMD компонентів, а у [Додатку Е](#) – рекомендовані розміри ПМ під різні типорозміри корпусів SMD резисторів.

Таким чином, створивши одне ПМ із визначеними розмірами, у майбутньому його можна буде використовувати як для SMD резисторів, так і для SMD конденсаторів тощо.

Створіть нову заготовку ПМ. Зробіть активним шар **Top Layer**. Встановіть два КМ, використовуючи налаштування, які показані на рис. 2.27. Зауважте, що необхідні геометричні розміри КМ задаються в полях **X-Size** та **Y-Size**. Оскільки КМ компонентів для поверхневого монтажу розміщуються на шарі **Top Layer**, то переконайтеся, що у полі **Properties** параметр **Layer** має значення **Top Layer**. Відстань між центрами КМ має становити 1,9 мм. Задайте такі номери КМ: лівий – 1, правий – 2.

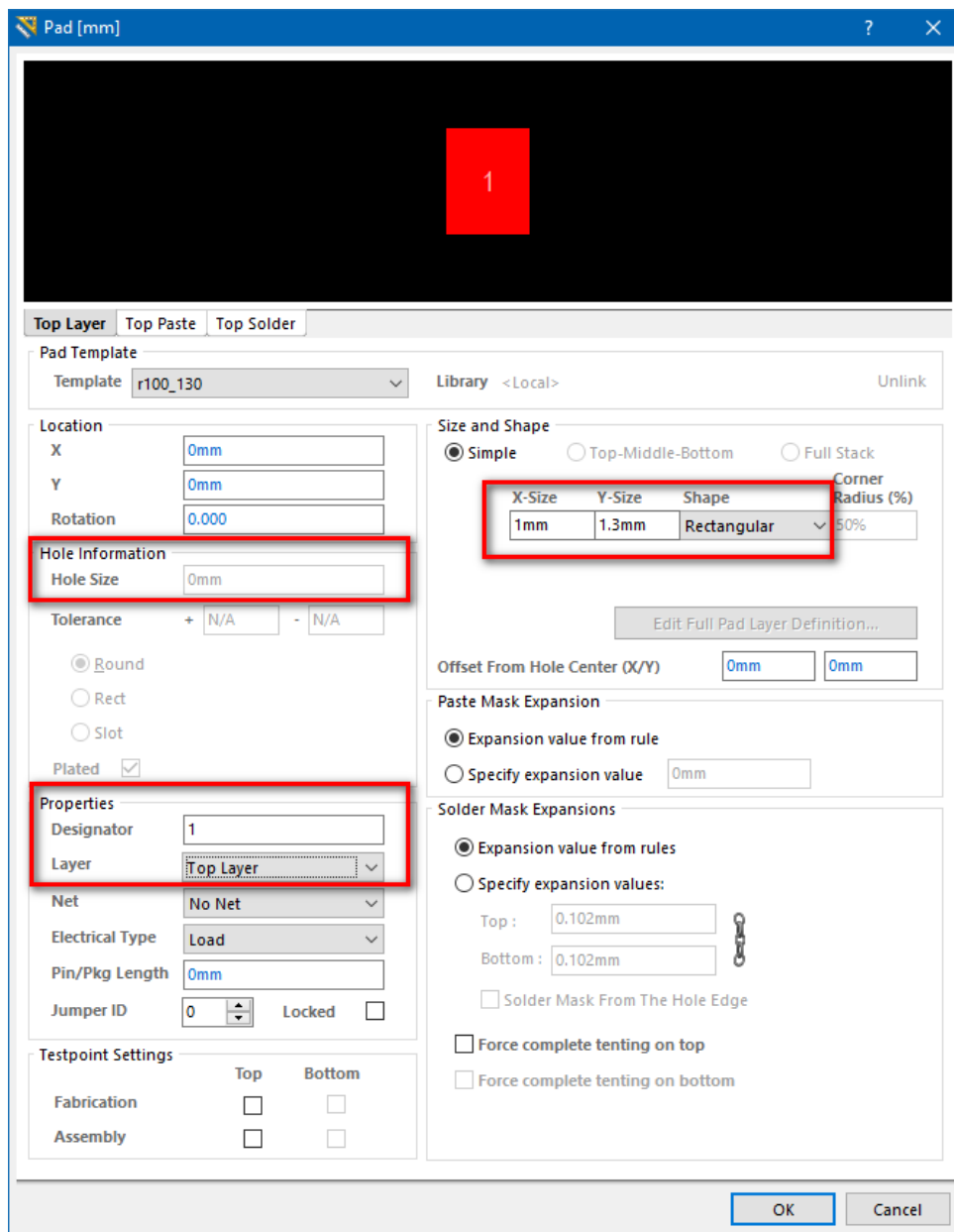


Рис. 2.27. Налаштування КМ планарного типу

Намалюйте контур ПМ згідно заданих геометричних розмірів (див. рис. 2.3) на шарі **Top Overlay** лінією товщиною 0,1 мм.

Остаточний результат зображений на рис. 2.28.

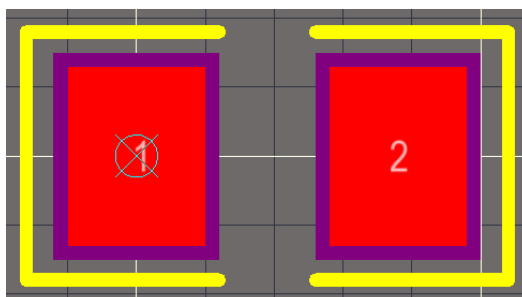


Рис. 2.28. Посадкове місце для корпусу SMD компоненту типорозміру 0805

Збережіть ПМ під іменем **RES_SMD0805**. Після цього створіть копію розробленого ПМ, задавши нове ім'я **CAP_SMD0805**⁴⁷.

3.8. Додавання посадкового місця до схемного компоненту

Після того, як були створені усі ПМ, поверніться у вікно **Projects** лівої бічної панелі, оберіть бібліотеку **Schematic_Component.SchLib** та перейдіть у вкладку **SCH Library**. Виберіть компонент **Res MF 0.125** і відкрийте вікно його властивостей. Натисніть на маленьку стрілку праворуч від кнопки **Add** у розділі **Models** вікна **Library Component Properties** і виберіть зі випадного списку рядок **Footprint** (рис. 2.29).

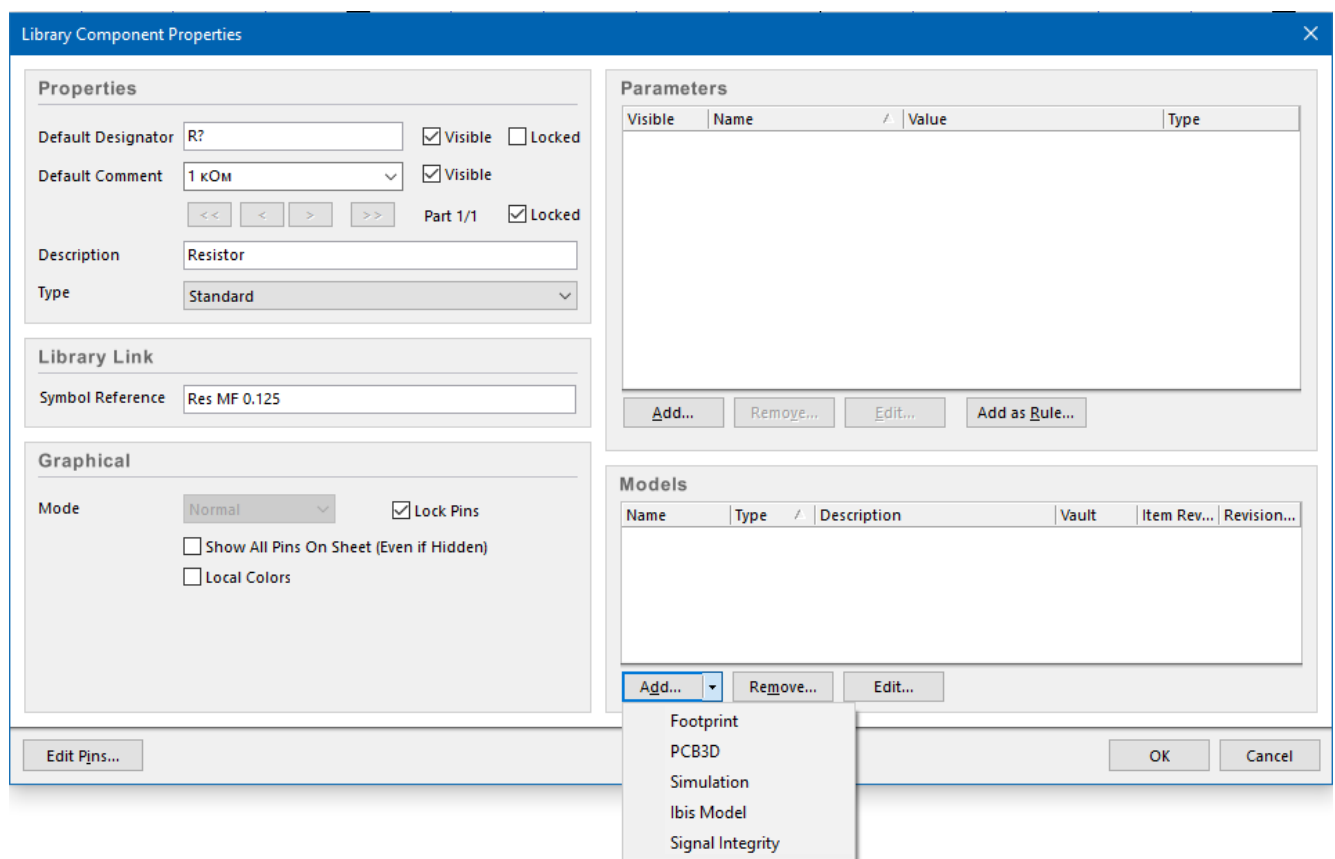


Рис. 2.29. Додавання моделі посадкового місця до компоненту

⁴⁷ Чому ми створили два окремих (хоча й ідентичних) ПМ для електронних компонентів у корпусі 0805, якщо раніше зазначали, що ПМ як для резисторів, так і для конденсаторів, будуть однаковими? Так зроблено лише тому, що надалі ми додамо тривимірні моделі корпусів резисторів та конденсаторів до відповідних ПМ. А оскільки за таким параметром, як висота корпусу, вони різні, то ми не зможемо використовувати одну тривимірну модель як для резисторів, так і для конденсаторів. Саме тому було створено два окремих ПМ.

Відкриється нове вікно *PCB Model* (рис. 2.30), яке використовується для додавання ПМ до бібліотечного компонента.

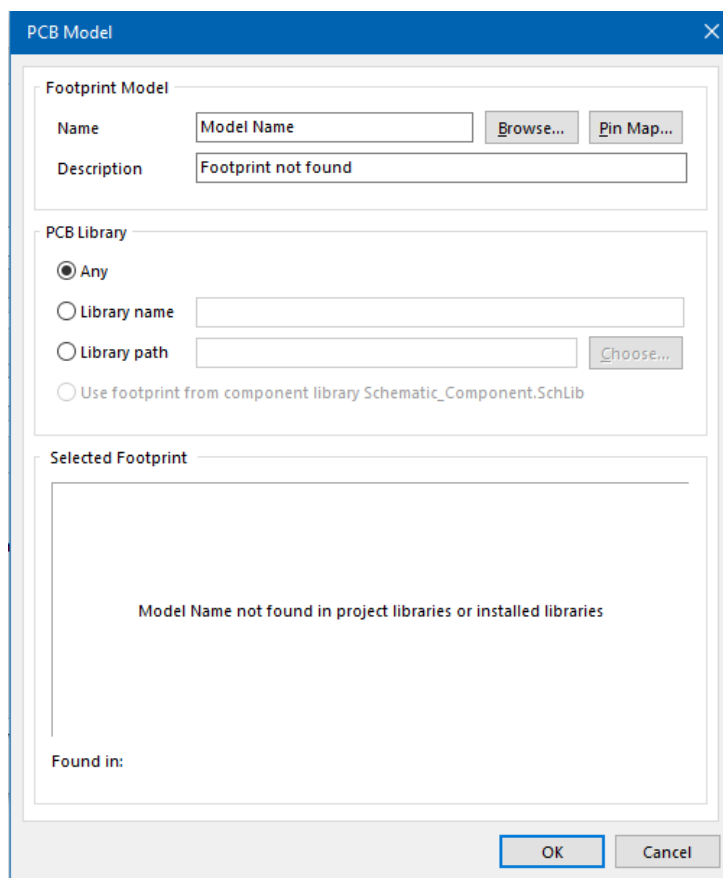


Рис. 2.30. Вікно *PCB Model*

Важливо!

Вибір перемикача *Any* в полі *PCB Library* означає, що модель ПМ береться з будь-якої бібліотеки. У такому випадку, якщо в проєкті будуть присутні кілька моделей з однаковими іменами, то невідомо, яка саме модель буде обрана програмою для проєкту. Перемикач *Library name* вказує, що модель буде взята із конкретної бібліотеки, завдяки чому конфлікту імен моделей не виникатиме.

Натисніть кнопку **Browse**, щоб відкрити діалог *Browse Libraries*, завдяки якому можна переглянути список доступних бібліотек ПМ (як доданих у проєкт, так і встановлених) та обрати відповідне ПМ.

Виберіть модель ПМ **RES_CONST_L6.0D0.8** та натисніть кнопку **OK**, щоб її додати. Інформація про додану модель з'явиться у розділі *Model* у нижній частині робочого простору (рис. 2.31).

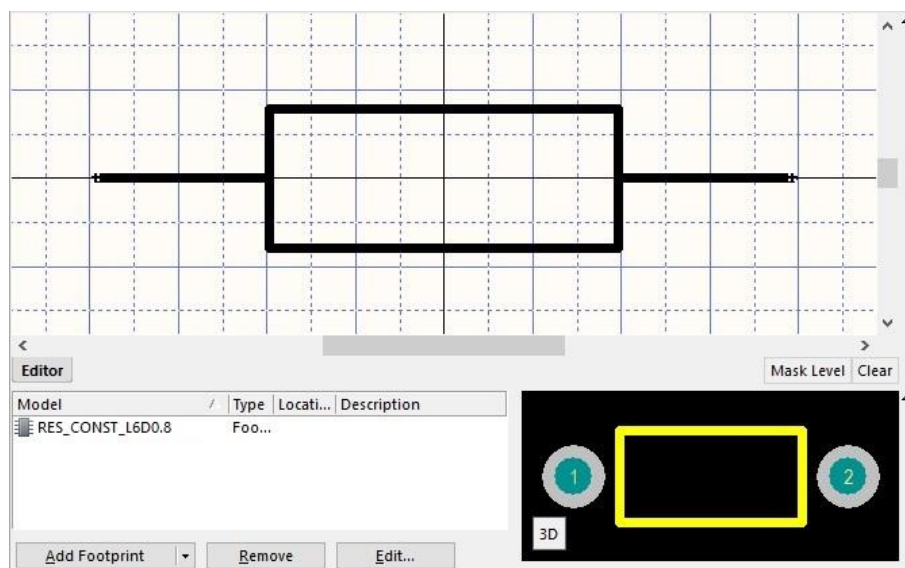


Рис. 2.31. Символ компонента із доданою моделлю посадкового місця

Зауваження!

Зверніть увагу, що модель ПМ завжди повинна повністю відображатись у нижній частині робочого простору після того, як вона підключена до символу. Якщо ж ви не бачите ПМ, а лише пусту чорну (або сіру) область, то однією із причин може бути наступне: коли ви створювали ПМ, то могли деякий примітив (наприклад, лінію, дугу, КМ тощо) випадково розмістити далеко від основного малюнка ПМ і забути про це. Програма вважає, що усі графічні елементи в межах робочої області належать до моделі і, відповідно, масштабує область таким чином, щоб вмістити усі елементи. Саме за рахунок такого масштабування й видно тільки темну область і не видно самого ПМ⁴⁸. Щоб виправити ситуацію, необхідно перейти у Редактор посадкових місць і видалити усі зайві елементи, які знаходяться далеко за межами моделі ПМ.

Після додавання моделі бажано перевірити відповідність нумерації виводів УГП номерам КМ посадкового місця (*Pin Mapping*). Відкрийте діалог *PCB Model* (див. рис. 2.30). Натисніть кнопку **Pin Map**, щоб відкрити додаткове вікно *Model Map* (рис. 2.32). Перевірте і виправте, якщо необхідно, відповідність номерів виводів компонента (*Component Pin Designator*) номерам КМ моделі ПМ (*Model Pin Designator*).

⁴⁸ Час від часу можуть проявлятися нестандартні ситуації під час роботи із програмою **Altium Designer**. Деякі із таких ситуацій описані у [Додатку Ж](#).

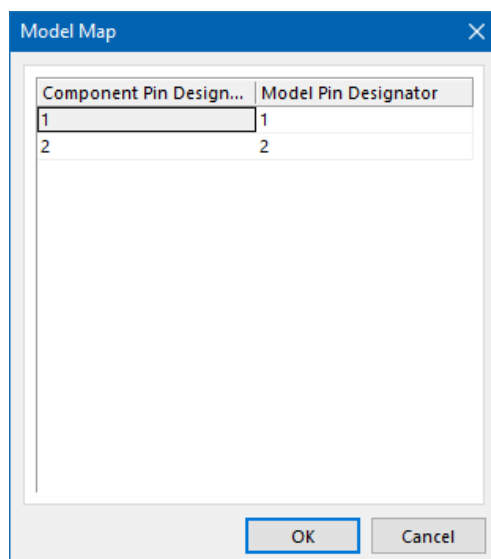


Рис. 2.32. Вікно перевірки відповідності виводів компонента контактним майданчикам моделі ПМ

Подібним чином об'єднайте наступні розроблені УГП та ПМ:

- **Res MF 0.25 + RES_CONST_L10.0D0.8;**
- **Res SMD0805 + RES_SMD0805;**
- **Res CA9V10 + RES_VAR_CA9V10;**
- **Res WH148 + RES_VAR_WH148;**
- **Cap K10-17Б + CAP_CERAM_L5.0D0.8;**
- **Cap Hitano ECR + CAP_POLAR_L2.5D0.8;**
- **Cap SMD0805 + CAP_SMD0805;**
- **Ind KIG + IND_L6.0D0.8.**

4. Питання для самоконтролю

1. Поясніть, що таке пасивні електронні компоненти?
2. Опишіть деякі види пасивних електронних компонентів та їх характеристики.
3. Опишіть УГП резисторів та вимоги до їх створення.
4. Поясніть, чим УГП постійного резистора відрізняється від УГП змінного резистора?
5. Що необхідно враховувати під час нумерації виводів на УГП змінного резистора?
6. Опишіть УГП конденсаторів та вимоги до їх створення.

7. Опишіть УГП котушок індуктивності та вимоги до їх створення.
8. Поясніть, яким чином створюється контур елемента в Редакторі символів?
9. Поясніть, яким чином додати виводи елемента? Які властивості виводу можна налаштовувати у програмі?
10. Опишіть, які атрибути електронного компонента можна редагувати у програмі?
11. Поясніть, яким чином створюються посадкові місця для електронного компонента?
12. Поясніть, навіщо перший КМ деяких електронних компонентів для монтажу в отвори роблять прямокутним?
13. Яким чином можна об'єднати УГП електронного компонента та його ПМ?

Комп'ютерний практикум № 3.

РОЗРОБКА КОМПОНЕНТНИХ МОДУЛІВ АКТИВНИХ ДИСКРЕТНИХ ЕЛЕКТРОННИХ КОМПОНЕНТІВ

Мета роботи: створити умовні графічні позначення та посадкові місця для діоду, біполярного та польового транзисторів; ознайомитись із типами корпусів активних дискретних електронних компонентів.

1. Теоретичні відомості

До активних дискретних електронних компонентів належать діоди, транзистори, тиристори, стабілітрони, фотоелектричні електронні компоненти тощо. Основою активних дискретних компонентів є **напівпровідники**.

Діод – це електронний компонент, що має два електроди і володіє різною провідністю залежно від напрямку електричного струму. Електроди діода називаються анодом і катодом. Принцип дії напівпровідникового діода ґрунтується на явищі *p-n*-переходу. Якщо до діода прикладено пряму напругу (тобто анод має додатний потенціал щодо катода), то діод відкритий (опір діода малий, через нього тече прямий струм). Якщо ж до діода прикладено зворотну напругу (катод має додатний потенціал щодо анода), то діод закритий (опір діода великий, зворотний струм малий і може вважатися рівним нулю у багатьох випадках).

Одним із поширених корпусів діодів є DO-41, геометричні характеристики якого наведені на рис. 3.1.

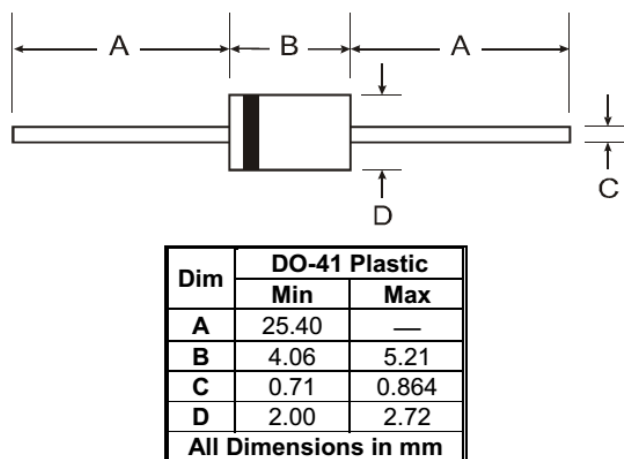


Рис. 3.1. Зовнішній вигляд та геометричні розміри корпусу діодів DO-41

Транзистор – це напівпровідниковий електронний компонент, що має три електроди і призначений для зміни параметрів електричного струму та керування ним. Польовий (уніполярний) транзистор – транзистор, у якому сила струму, що протікає крізь нього, керується зовнішнім електричним полем, тобто напругою. Біполярний транзистор – транзистор, у якому сила струму у вихідному колі регулюється струмом керування.

Транзистори випускаються у різних корпусах. Одним із поширених корпусів, у якому випускаються біполярні транзистори, є SOT-54, геометричні розміри якого наведені на рис. 3.2.

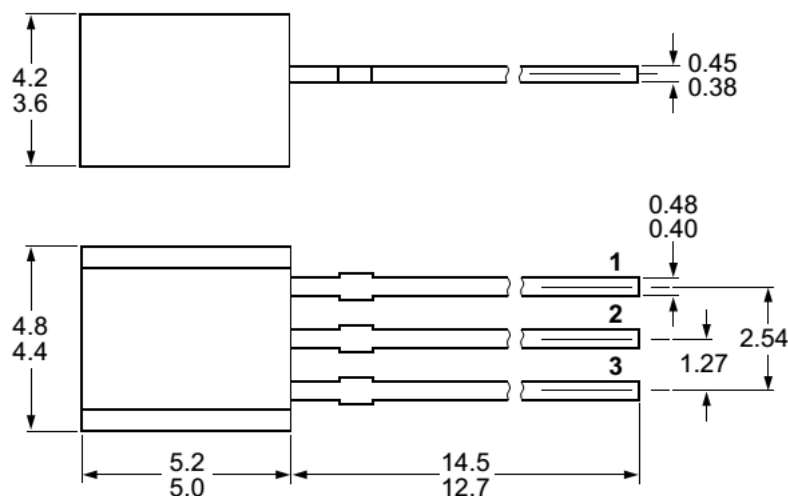


Рис. 3.2. Зовнішній вигляд та геометричні розміри корпусу SOT-54 (у міліметрах)

Польові транзистор випускаються здебільшого у корпусах SOT-23-3 та DPAK, геометричні розміри яких наведені на рис. 3.3 та рис. 3.4 відповідно.

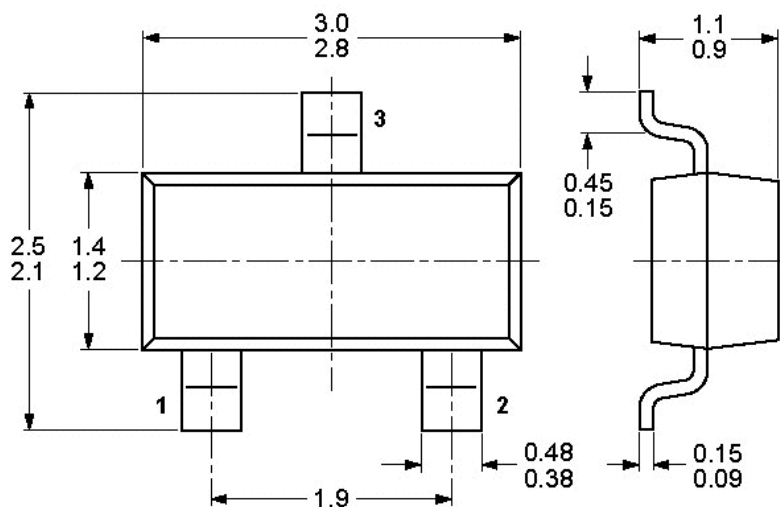


Рис. 3.3. Зовнішній вигляд та геометричні розміри корпусу SOT-23-3 (у міліметрах)

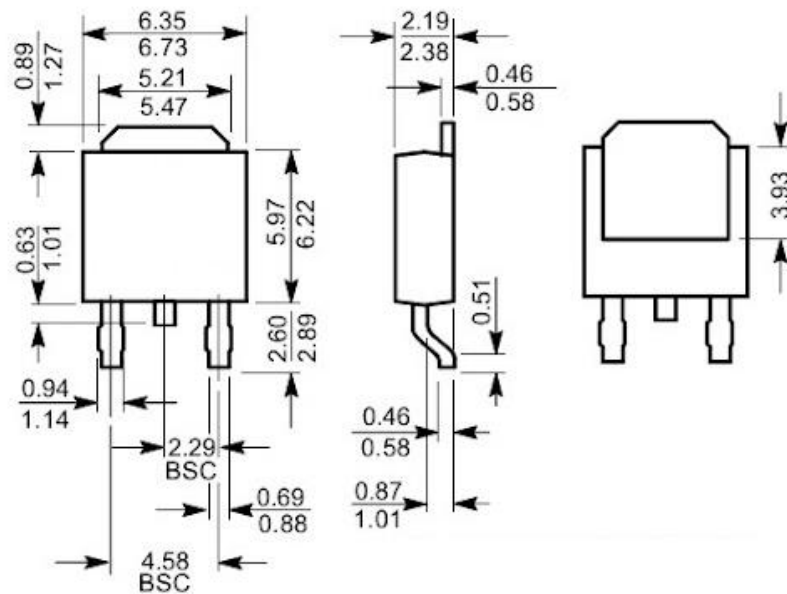


Рис. 3.4. Геометричні розміри корпусу DPAK (у міліметрах)

Загалом під час створення компонентних модулів транзисторів (та й деяких інших електронних компонентів) треба бути уважним, задаючи відповідність номерів виводів їх функціональному призначенню. Це пов'язано з тим, що для різних типів транзисторів різні фірми-виробники визначають свою відповідність номерів виводів їх функціональному призначенню. Розглянемо це на прикладі корпусу SOT-23-3. В цьому корпусі можуть випускатись як біполярні, так і польові транзистори. Фізична нумерація виводів такого корпусу строго визначена. Проте з функціональної точки зору, наприклад, для транзистору JFET BFT46 (виробник NXP Semiconductors) та JFET MMBFJ111 (виробник Fairchild Semiconductor) перший вивід – сток, другий – джерело, третій – затвор; а для транзистору JFET PMBFJ308 (виробник Philips) та JFET MMBF4091 (виробник Fairchild Semiconductor) перший вивід – джерело, другий – сток, третій – затвор. Для MOSFET транзисторів відповідність номерів виводів в корпусі SOT-23-3 їх функціональному призначенню може значно відрізнятись від розглянутої для JFET. Наприклад, на рис. 3.5 представлена нумерація виводів та їх функціональне призначення для MOSFET IRLML2502 (виробник International Rectifier), де перший вивід – затвор, другий – джерело, третій – сток.

Таким чином, функціональне призначення виводів 1, 2 та 3 корпусу SOT-23-3 відрізняється як у різних виробників, так і у різних типів напівпровідникових приладів одного й того ж виробника. Отже, якщо це не враховувати, то можна неправильно встановити транзистор на плату, що призведе до неможливості роботи

схеми. Тому важливою задачею є встановлення правильної відповідності функціонального призначення виводів та їх номерів в корпусі мікросхеми. Для цього обов'язково попередньо необхідно ознайомитись зі специфікацією (datasheet) на той чи інший електронний компонент від виробника.

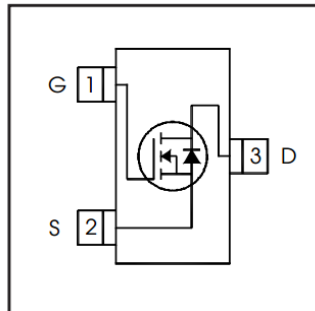


Рис. 3.5. Виводи транзистору MOSFET IRLML2502

Розглянемо ще один приклад. Біполярні транзистори серії BC547 (структури $n-p-n$) і BC557 (структури $p-n-p$) випускаються в корпусах SOT-54. Зверніть увагу, що транзистор BC547, який випускається фірмою Philips Semiconductors, має наступну відповідність функціонального призначення виводів їх номерам: емітер – 1, база – 2, колектор – 3. Тоді як транзистор BC557, що випускається фірмою ON Semiconductors, має таку відповідність: емітер – 3, база – 2, колектор – 1 (рис. 3.6).

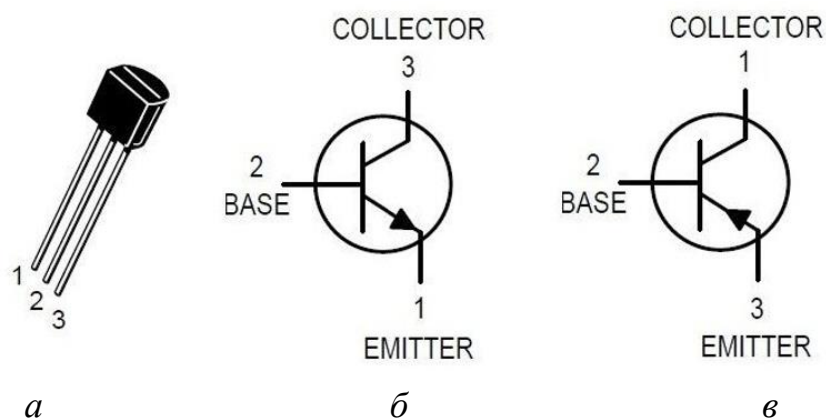


Рис. 3.6. Транзистори корпусі SOT-54: *а* – зовнішній вигляд корпусу SOT-54, *б* – BC547 (Philips Semiconductors), *в* – BC557 (ON Semiconductors)

2. Завдання на комп'ютерний практикум

Створити типові компонентні модулі для наступних електронних компонентів: діод **1N4001**, біполярні транзистори **BC547** (Philips Semiconductors) та **BC557** (ON Semiconductors), JFET транзистор **PMBFJ308** (Philips Semiconductors) та **MMBFJ176** (ON Semiconductors), MOSFET транзистор

IRFR9310 (Vishay Siliconix) та **NTD24N06** (ON Semiconductors). Для цього спочатку намалювати УГП, що зображені на рис. 3.7-3.9, використовуючи вказані геометричні розміри.

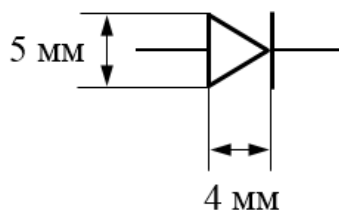


Рис. 3.7. УГП напівпровідникового діоду

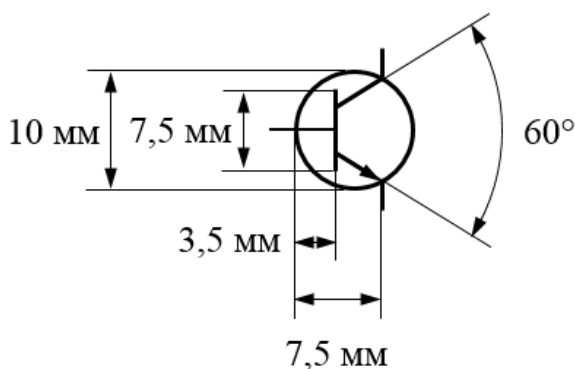


Рис. 3.8. УГП біполярного транзистора

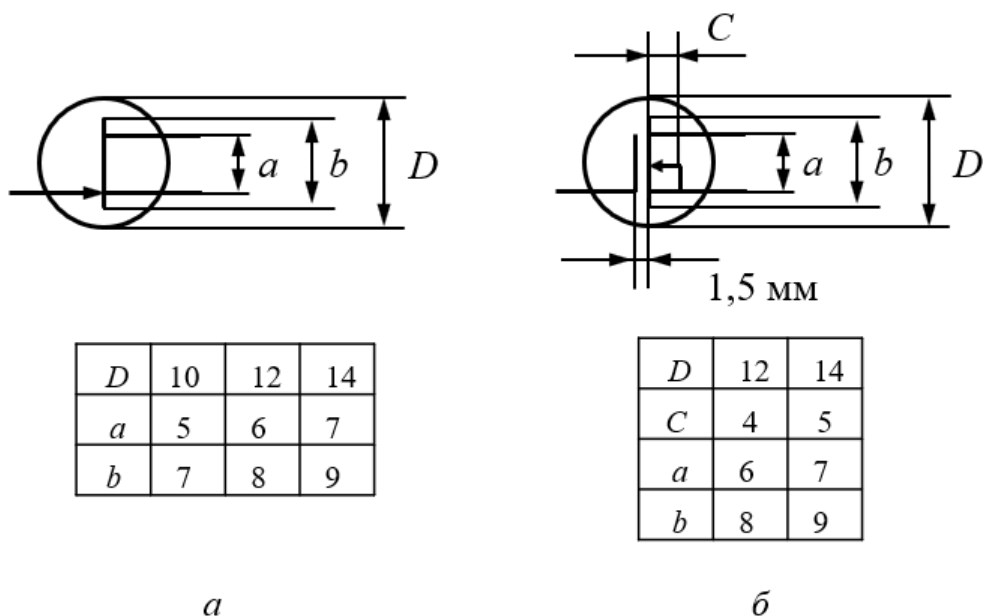


Рис. 3.9. УГП електронних компонентів: *a* – польовий транзистор, *б* – MOSFET

Далі створити ПМ кожного електронного компонента, що зображені на рис. 3.10-3.13.

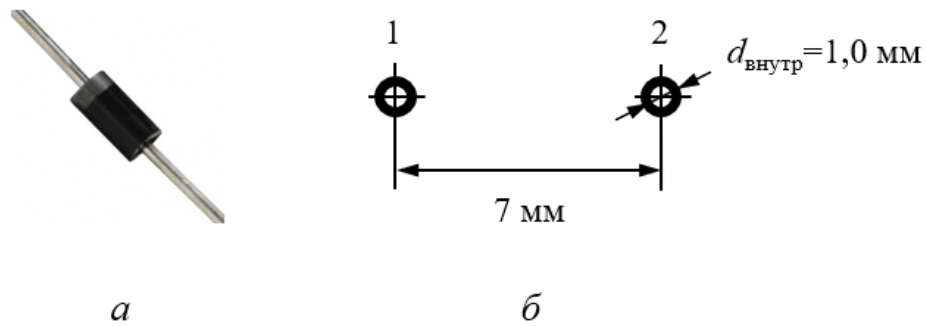


Рис. 3.10. Діод в корпусі DO-41: *а* – зображення корпусу, *б* – геометричні розміри посадкового місця для монтажу в отвори

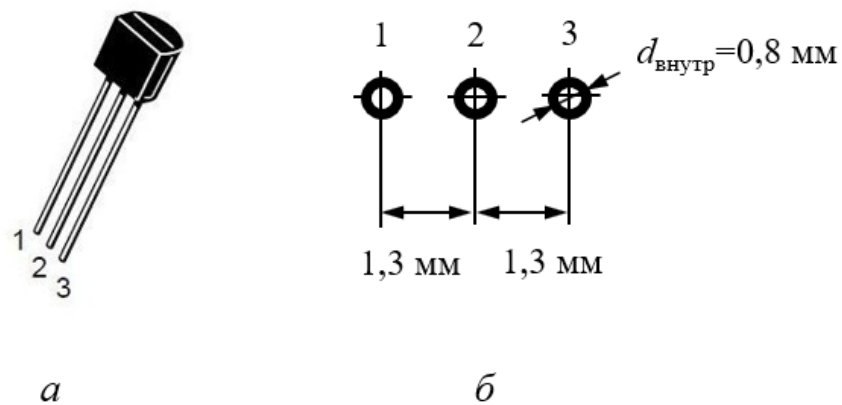


Рис. 3.11. Біполярний транзистор в корпусі SOT-54: *а* – зображення корпусу, *б* – геометричні розміри посадкового місця для монтажу в отвори

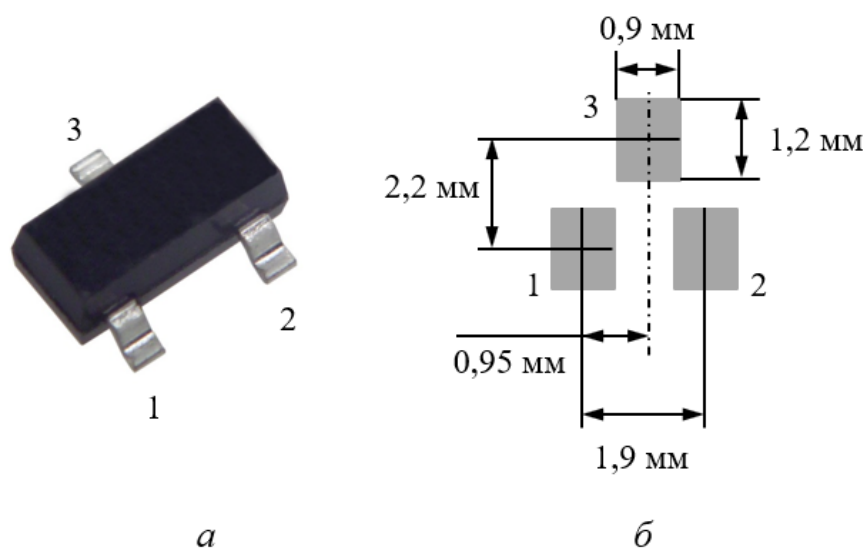


Рис. 3.12. JFET транзистор в корпусі SOT-23-3: *а* – зображення корпусу, *б* – рекомендовані геометричні розміри посадкового місця для поверхневого монтажу

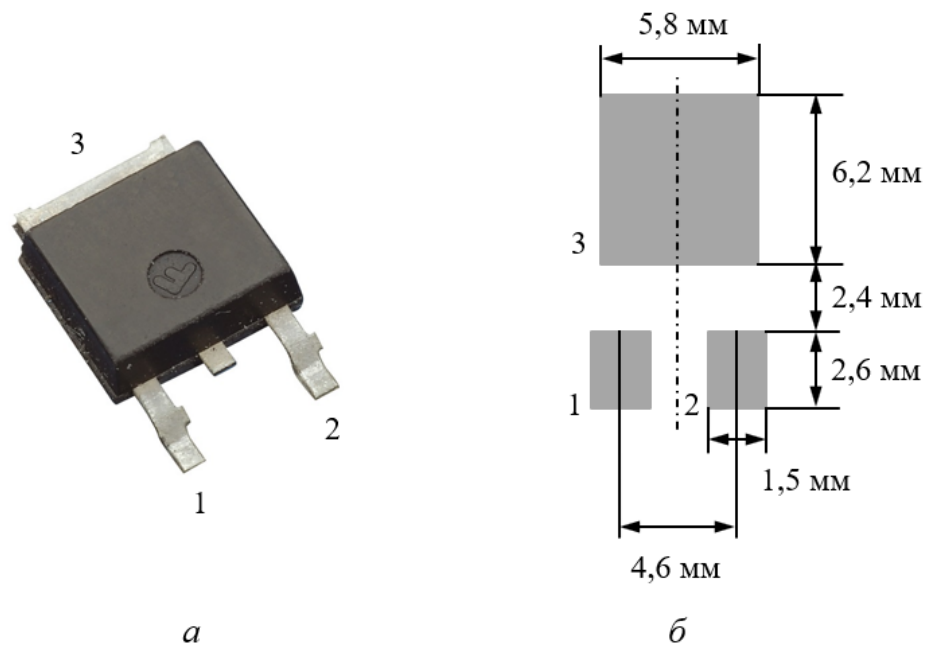


Рис. 3.13. MOSFET транзистор в корпусі DPAK: *a* – зображення корпусу, *б* – рекомендовані геометричні розміри посадкового місця для поверхневого монтажу

3. Порядок виконання комп'ютерного практикуму

3.1. Створення УГП діоду

Відкрийте програму **Altium Designer**. Перейдіть у бібліотеку символів **Schematic_Component.SchLib**. Створіть копію компонента **Blank_Component**, яку перейменуйте на **1N4001**. У робочому полі накресліть контур елемента за допомогою команди **Place** → **Line**. Для цього потрібно накреслити рівнобедрений трикутник, одна сторона якого (основа) складає 5 мм, а висота складає 4 мм. Проведіть вертикальну лінію довжиною 5 мм, яка проходить через вершину трикутника паралельно основі (рис. 3.14).

Рекомендація!

За замовчуванням у Редакторі символів лінії можуть розміщуватись тільки вертикально або горизонтально. Якщо вам необхідно намалювати лінію під довільним кутом, то під час малювання натисніть послідовно декілька разів клавішу <Пробіл> (у такому випадку будуть послідовно перемикатись режими) до тих пір, поки не стане активним режим із можливістю розміщення лінії під довільним кутом.

Встановіть виводи елементу довжиною 5 мм за допомогою команди **Place** → **Pin**. Для першого виводу (який розміщується ліворуч від символу) у діалоговому вікні **Place Pin** в полі **Display Name** введіть знак «+», а в полі **Designator** – цифру 1 (водночас опція **Visible** в обох випадках має бути вимкненою). Аналогічно створіть другий вивід, встановивши у полі **Display Name** знак «-», а в полі **Designator** – цифру 2.

Остаточний результат наведено на рис. 3.14.

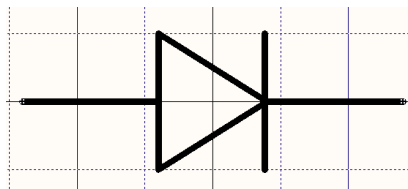


Рис. 3.14. УГП діоду

Відкрийте вікно властивостей розробленого компоненту і в рядку **Default Designator** вкажіть префікс позиційного позначення діода **VD**, після якого поставте знак запитання⁴⁹. У полі **Description** введіть текст **Diode**, в рядку **Default Comment** приберіть галочку **Visible**.

3.2. Створення УГП біполярного транзистору

Створіть новий компонент з ім'ям **BC547**. Викличте команду **Place** → **Full circle** та накресліть коло довільного радіусу у довільній точці. Потім, натиснувши ПКМ на контурі кола, у діалоговому меню оберіть пункт **Properties**. У вікні **Arc Properties**, що відкриється, в полі **Radius** введіть 5 мм. Відповідно до розмірів, що наведені на рис. 3.8, намалюйте необхідні лінії всередині кола (рис. 3.16a).

Лінія, що позначає емітерний вивід, повинна мати стрілку (для транзистору структури *n-p-n* стрілка має напрямок від бази, для *p-n-p* – до бази). Для створення стрілки можна використати два підходи. Перший полягає у тому, що необхідно викликати властивості лінії та у вкладці **Graphical** для параметра **Start Line Shape** (або **End Line Shape**) встановити значення **SolidArrow** (рис. 3.15).

Інший підхід полягає у використанні команди **Place** → **Polygon**, за допомогою якої окремо рисують трикутну стрілку на кінці лінії. Для більш точного позиціонування стрілки рекомендується використовувати крок сітки 0,1 мм і менше.

⁴⁹ **VD** – це стандартне позиційне позначення діода на принциповій схемі, а от замість знаку «?» в майбутньому під час виконання автоматичної нумерації елементів на схемі **Altium Designer** буде підставляти порядковий номер діода.

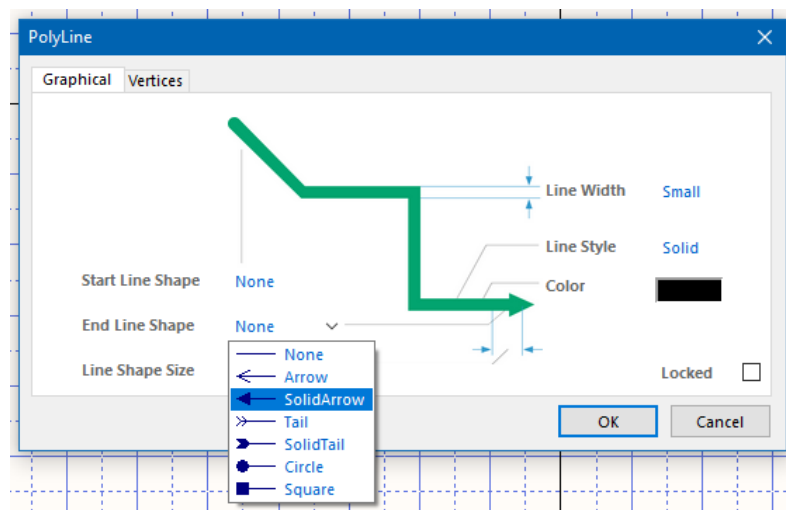


Рис. 3.15. Вікно властивостей лінії

Створіть виводи транзистору за допомогою команди **Place** → **Pin**. У діалоговому вікні **Place Pin** в полі **Length** задайте довжину виводу 2,5 мм. У вікні властивостей виводу у полі **Display Name** введіть англійську⁵⁰ літеру **E** (емітер), а у полі **Designator** – значення 1 (відображення імені та номеру виводу має бути відключене). Розмістіть вивід знизу під контуром елементу (там, де стрілочка). Аналогічно створіть виводи для бази **B** (у полі **Designator** введіть 2) та колектору **C** (у полі **Designator** введіть 3) і розмістіть їх відповідно зліва та зверху над елементом.

Зауваження!

В принципі, нумерація виводів символу транзистора може бути довільною. У даному випадку вибір саме такого порядку нумерації виводів пов'язаний з тим, що так вказано у документації біполярного транзистора BC547 фірми Philips Semiconductors, що випускається у корпусі SOT-54. Оскільки сам символ біполярного n-p-n-транзистора є стандартним, то його можна використовувати (методом копіювання) під час створення компонентних модулів біполярних транзисторів інших серій. Однак необхідно пам'ятати, що нумерація виводів та їх відповідність функціональному призначенню можуть відрізнитись залежно від серії транзистора. У такому випадку необхідно буде або змінити нумерацію виводів символу згідно документації відповідного транзистора, або потрібно буде відредагувати дані у вікні **Pin Map** під час об'єднання УГП та ПМ і задати необхідний зв'язок між номерами виводів транзистору та номерами КМ посадкового місця.

⁵⁰ Рекомендується вводити імена виводів саме англійською мовою.

Аналогічно описаному вище створіть компонент **BC557** (рис. 3.16б). Однак врахуйте, що біполярний транзистор **BC557** фірми ON Semiconductors має іншу нумерацію виводів, ніж **BC547**, а саме: колектор – 1, база – 2, емітер – 3.

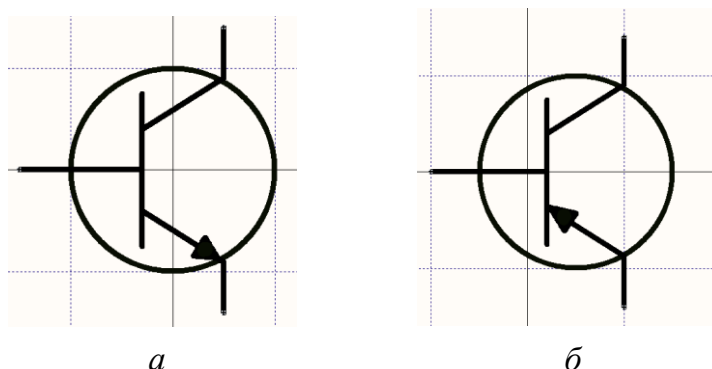


Рис. 3.16. УГП біполярного транзистору: *а* – тип n-p-n, *б* – тип p-n-p

Відкрийте вікно властивостей кожного транзистора і в полі **Description** для транзистора **BC547** введіть текст **Transistor, NPN Generic**, а для **BC557** – текст **Transistor, PNP Generic**. В рядку **Default Designator** вкажіть префікс позиційного позначення компонента **VT?**. Приберіть галочку **Visible** в рядку **Default Comment**.

3.3. Створення УГП польових транзисторів JFET

Створіть новий компонент з ім'ям **PMBFJ308**. Намалюйте коло радіусом 5 мм. Відповідно до рис. 3.9а оберіть значення $a=5$ мм, $b=7$ мм. Намалюйте усі необхідні лінії всередині кола (рис. 3.17а).

Створіть виводи транзистору за допомогою команди **Place** → **Pin** довжиною 2,5 мм. У діалоговому вікні **Place Pin** в полі **Display Name** та **Designator** вимкніть опції **Visible**. У полі **Display Name** введіть для джерела (sources) літеру **S** (в полі **Designator** – цифру 1), для стоку (drain) – **D** (в полі **Designator** – цифру 2), для затвору (gate) – **G** (в полі **Designator** – цифру 3). Розмістіть виводи відповідно їх призначенню на УГП польового транзистора: затвор – ліворуч від символу, джерело – знизу, сток – зверху⁵¹.

⁵¹ В документації на деякі польові транзистори виробники вказують, що виводи джерела та стоку є взаємозамінними. Це справедливо тільки у випадку, коли канал між джерелом і стоком, по якому рухаються носії заряду, є геометрично симетричним. Однак внаслідок недосконалості процесу виробництва ця умова не завжди виконується, тому краще не змінювати місцями джерело та сток. Це необхідно враховувати, коли до символу буде додаватись модель ПМ і необхідно буде встановити відповідність номерів виводів польового транзистору та номерів КМ посадкового місця.

Аналогічно описаному вище створіть компонент **MMBFJ176** (рис. 3.17б). Однак врахуйте, що польовий транзистор **MMBFJ176** має іншу нумерацію виводів, ніж **PMBFJ308**, а саме: сток – 1, джерело – 2, затвор – 3.

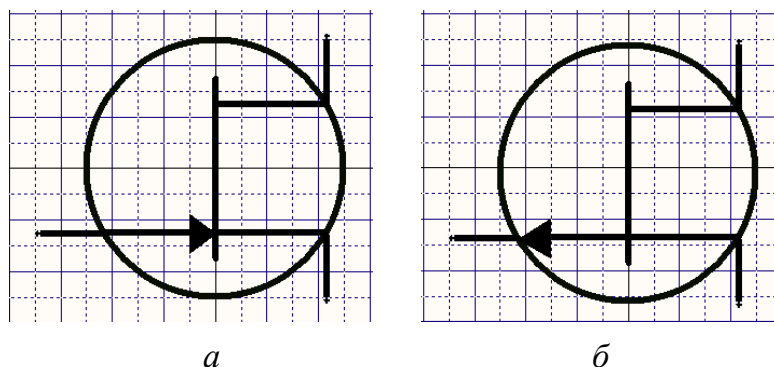


Рис. 3.17. УГП польових транзисторів JFET: *a* – n-канал, *б* – p-канал

Викличте властивості кожного транзистора і в полі **Description** для транзистора **PMBFJ308** введіть текст **JFET transistor, n-channel**, а для **MMBFJ176** – текст **JFET transistor, p-channel**. В рядку **Default Designator** вкажіть префікс позиційного позначення компонента **VT?**. Приберіть галочку **Visible** у рядку **Default Comment**.

3.4. Створення УГП MOSFET-транзисторів з індукованим каналом

Створіть новий компонент з ім'ям **IRFR9310**. Намалюйте коло радіусом 6 мм. Відповідно до рис. 3.9б використовуйте значення $a=6$ мм, $b=8$ мм, $c=4$ мм. Намалюйте усі необхідні лінії всередині кола (рис. 3.18а).

Створіть виводи транзистору за допомогою команди **Place** → **Pin** довжиною 2,5 мм. У діалоговому вікні **Place Pin** в полі **Display Name** та **Designator** вимкніть опції **Visible**. У полі **Display Name** введіть для затвору літеру **G** (в полі **Designator** – цифру 1), для джерела – **S** (в полі **Designator** – цифру 2), для стоку – **D** (в полі **Designator** – цифру 3). Розмістіть виводи відповідно їх призначенню на УГП MOSFET транзистора: затвор – ліворуч від символу, джерело – знизу, сток – зверху.

Аналогічно описаному вище створіть новий компонент **NTD24N06**, що є n-канальним MOSFETом (рис. 3.18б). Нумерація виводів має бути такою самою, як і для **IRFR9310**.

Відредагуйте властивості обох створених компонентів. У полі **Description** для транзистора **IRFR9310** введіть текст **MOSFET transistor, p-channel**, а для **NTD24N06** – текст **MOSFET transistor, n-channel**. В рядку **Default Designator**

вкажіть префікс позиційного позначення компонента **VT?**. Приберіть галочку **Visible** в рядку **Default Comment**.

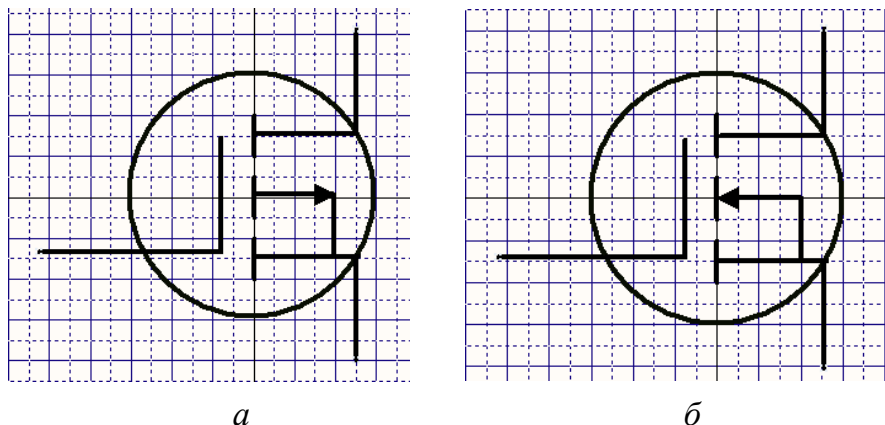


Рис. 3.18. MOSFET з індукованим каналом: *а* – р-канал, *б* – n-канал

3.5. Створення посадкового місця діода

Перейдіть у бібліотеку ПМ. Створіть нову заготовку, виконавши команду **Tools** → **New Blank Component**. За допомогою команди **Place** → **Pad** розмістіть у робочій області два КМ (внутрішній діаметр – 1,0 мм, зовнішній – 1,4 мм), відстань між центрами яких має становити 7 мм. Нумерація КМ згідно рис. 3.19.

Накресліть контур ПМ діода та символ діода за допомогою команд **Place** → **Line** та **Place** → **Arc (Center)**. Контур повинен мати вигляд округленого прямокутника (основа 2,8x2,0 мм та півкола радіусом 1,4 мм) і бути намальованим лінією товщиною 0,2 мм на шарі **Top Overlay**. В межах контуру схематично намалюйте зображення діода, як показано на рис. 3.19.

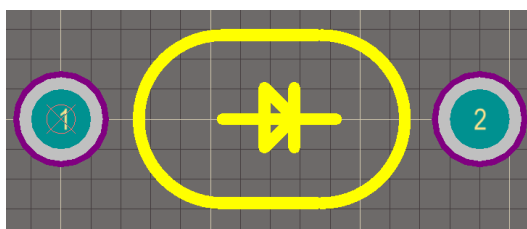


Рис. 3.19. Посадкове місце для діода в корпусі DO-41

Збережіть створене ПМ у свою бібліотеку під іменем **DO41**.

3.6. Створення посадкового місця біполярного транзистору

Створіть наступну заготовку ПМ для транзистору в корпусі SOT-54. Необхідно враховувати, що виводи цього типу корпусу мають прямокутний перетин та

розташовуються близько один до одного, що не дозволяє задати достатній діаметр КМ круглої форми для якісного припаювання. Незважаючи на те, що виводи корпусу SOT-54 мають прямокутну форму, внутрішній отвір КМ під вивід все одно матиме круглу форму. А для того, щоб забезпечити надійне припаювання елементу у випадку, коли виводи розташовані близько один до одного, зовнішня форма КМ повинна мати вигляд овалу.

За допомогою команди **Place** → **Pad** розмістіть вздовж горизонталі три КМ з отворами, відстань між центрами яких має становити 1,3 мм. Задайте наступні параметри кожного КМ:

- в графічній області **Size and Shape**: форма (**Shape**) – кругла (**Round**), ширина **X-Size** – 1 мм; висота **Y-Size** – 1,8 мм;
- в графічній області **Hole Information**: діаметр внутрішнього отвору (**Hole Size**) – 0,8 мм.

Нумерація КМ має відповідати тій, яка представлена на рис. 3.20.

Накресліть контур транзистора за допомогою команд **Place** → **Line** та **Place** → **Arc (Center)**. Контур повинен бути намальованим на шарі **Top Overlay** та мати вигляд обрізаного кола (у вікні властивостей дуги **Arc Properties** встановіть **Radius** – 2,5 мм, **Start Angle** – 325 deg, **End Angle** – 215 deg), центр якого співпадає із центром другого КМ. Товщина ліній контуру має становити 0,2 мм.

Остаточний результат зображений на рис. 3.20.

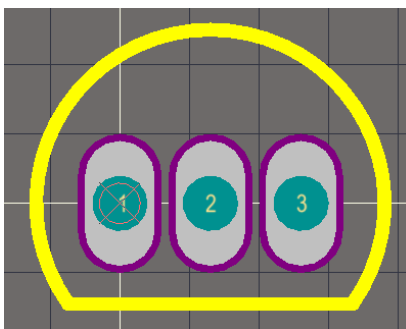


Рис. 3.20. Посадкове місце для біполярного транзистора в корпусі SOT-54

Збережіть створене ПМ у свою бібліотеку під іменем **SOT54**.

3.7. Створення посадкового місця JFET транзистору

Створіть наступну заготовку ПМ. За допомогою команди **Place** → **Pad** розмістіть три планарні КМ на шарі **Top Layer**, використовуючи геометричні розміри та нумерацію, які наведені на рис. 3.12.

Схематично накресліть контур ПМ за допомогою команди *Place* → *Line* (товщина лінії 0,1 мм) на шарі *Top Overlay*. Результат представлений на рис. 3.21.

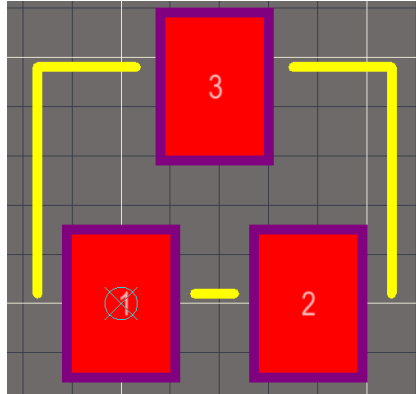


Рис. 3.21. Посадкове місце для польового транзистору JFET у корпусі SOT-23-3

Збережіть створене ПМ місце у свою бібліотеку під іменем **SOT23-3**.

3.8. Створення посадкового місця MOSFET транзистору

Створіть наступну заготовку ПМ. За допомогою команди *Place* → *Pad* розмістіть три планарні КМ на шарі *Top Layer*, використовуючи геометричні розміри та нумерацію, які наведені на рис. 3.13.

Схематично накресліть контур ПМ (рис. 3.22) лініями товщиною 0,1 мм за допомогою команди *Place* → *Line*. Контур має бути намальованим на шарі шовкографії *Top Overlay*.

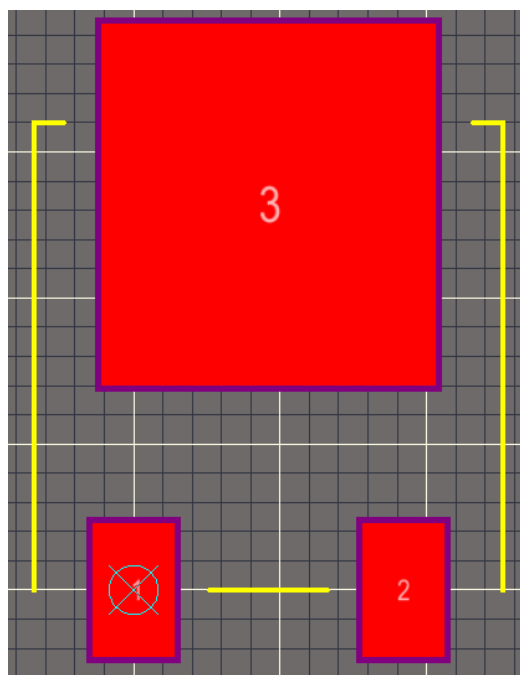


Рис. 3.22. Посадкове місце для польового транзистору MOSFET у корпусі DPAK

Збережіть створене ПМ у свою бібліотеку під іменем **DPAK**.

3.9. Об'єднання УГП та посадкових місць розроблюваних компонентних модулів

Розглянемо об'єднання УГП та ПМ на прикладі діода. Після чого аналогічні дії необхідно буде повторити для усіх інших компонентів.

Зробіть активним Редактор бібліотеки символів. Перейдіть у вкладку **SCH Library** лівої бічної панелі **Projects**. Оберіть компонент **1N4001** і відкрийте його властивості. Натисніть на маленьку стрілку праворуч від кнопки **Add** у розділі **Model** діалогу **Library Component Properties** і виберіть зі списку **Footprint**. Відкриється діалогове вікно **PCB Model**. Натисніть кнопку **Browse**, щоб відкрити нове вікно пошуку **Browse Libraries**, за допомогою якого можна переглянути список доступних бібліотек і вибрати відповідне ПМ.

Оберіть потрібну модель ПМ та натисніть кнопку **OK**. Інформація про додану модель з'явиться у розділі **Model** в нижній частині робочого простору (рис. 3.23).

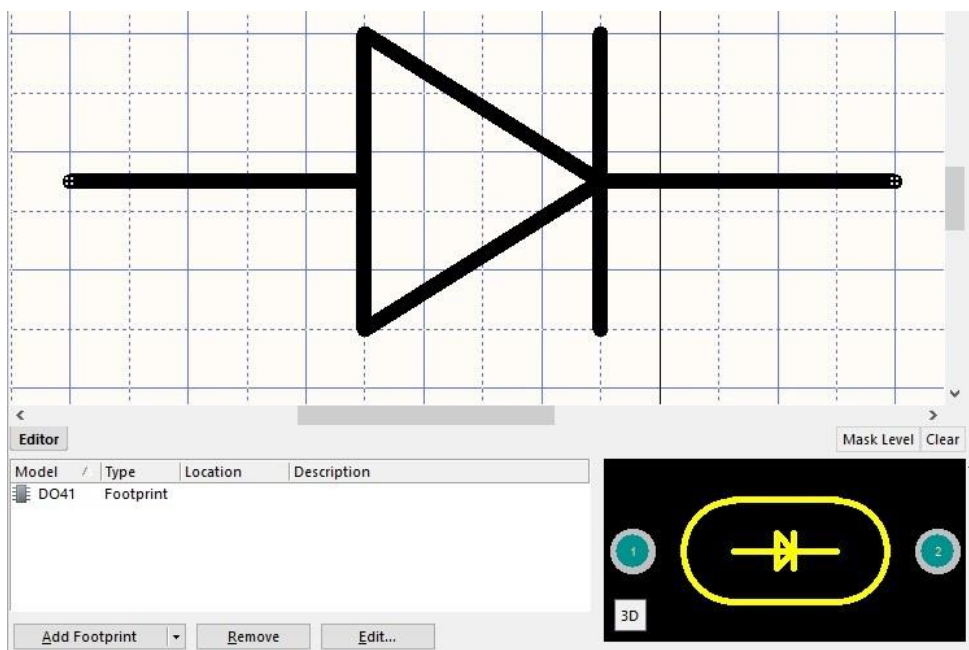


Рис. 3.23. УГП діода з доданим посадковим місцем

Після додавання моделі ПМ перевірте відповідність нумерації виводів електронного компонента номерам КМ за допомогою діалогу **Model Map** (рис. 3.24). Якщо відповідності не буде, то з'ясуйте причину цього та усуньте помилки.

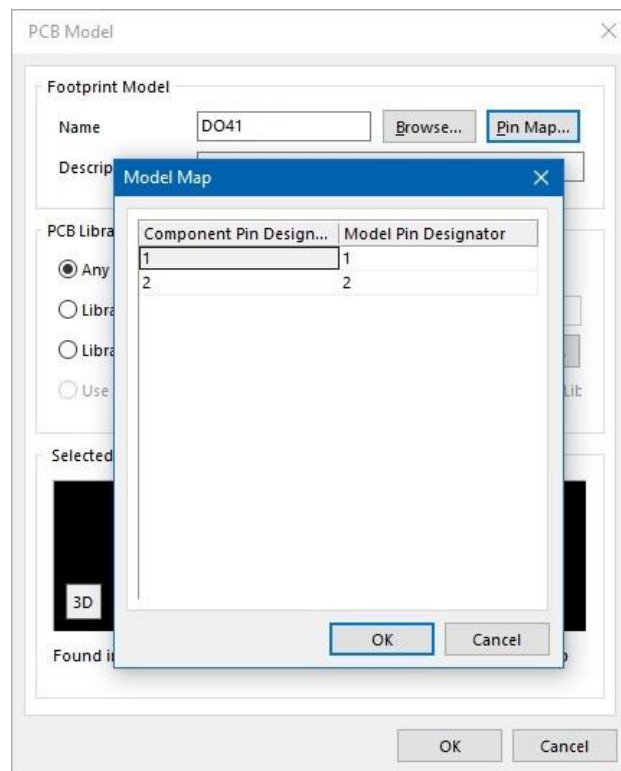


Рис. 3.24. Діалог *Model Map*

Аналогічно описаному вище об'єднайте УГП **BC547** та **BC557** з посадковим місцем **SOT54**, УГП **PMBFJ308** та **MMBFJ176** – з посадковим місцем **SOT23-3**, УГП **IRFR9310** та **NTD24N06** – з посадковим місцем **DPAK**.

4. Питання для самоконтролю

1. Поясніть, що таке активні дискретні електронні компоненти?
2. Поясніть, які чинники необхідно враховувати при завданні нумерації виводів транзисторів та встановленні відповідності номерів виводів їх функціональному призначенню.
3. Яким чином за УГП діода визначити, у якому напрямку він пропускає струм?
4. Яким чином за корпусом діода визначити, де катод, а де – анод?
5. Поясніть, яким чином створити контур КМ не круглої форми, а, наприклад, овальної?
6. Опишіть процес розробки ПМ для електронного компоненту у корпусі SOT23-3.
7. Поясніть, яким чином об'єднати УГП та ПМ на прикладі одного із транзисторів.

Комп'ютерний практикум № 4.

РОЗРОБКА КОМПОНЕНТНОГО МОДУЛЯ ЦИФРОВОЇ МІКРОСХЕМИ K155ЛА3

Мета роботи: ознайомитись з конструкцією та призначенням мікросхеми K155ЛА3; створити умовне графічне позначення структурного елементу мікросхеми K155ЛА3, яким є логічний елемент І-НЕ; розробити посадкове місце для мікросхеми K155ЛА3.

1. Теоретичні відомості

Мікросхема K155ЛА3 належить до сімейства цифрових мікросхем і містить чотири логічних елементи І-НЕ з двома входами (рис. 4.1), що виготовлені за технологією транзисторно-транзисторної логіки⁵² (ТТЛ). Живлення подається на мікросхему через виводи, позначені як GND (вивід 7) і Vcc (вивід 14). Усі інші виводи є входами та виходами чотирьох ТТЛ-елементів.

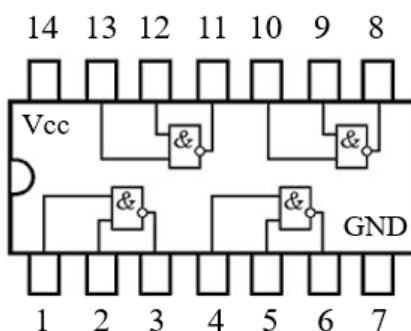


Рис. 4.1. Внутрішня структура мікросхеми K155ЛА3

Мікросхема K155ЛА3 була радянським аналогом (а точніше, копією) мікросхеми SN7400, що виготовлялась американською фірмою Texas Instruments із середини 1960-х років. Наразі мікросхема K155ЛА3 не виготовляється, а її повністю витіснили варіанти SN7400, які тепер випускають безліч різних компаній. Проте мікросхема K155ЛА3 свого часу була настільки популярною, що її виготовляли такими партіями, залишки яких досі можна знайти на ринку електронних компонентів.

⁵² Транзисторно-транзисторна логіка – це різновид цифрових логічних мікросхем, побудованих на основі біполярних транзисторів і резисторів.

Мікросхема K155ЛА3 випускалась тільки у корпусі DIP, тоді як SN7400 могла мати корпус або DIP, або SOP. Ще однією особливістю було те, що K155ЛА3 мала дещо відмінні від SN7400 геометричні характеристики корпусу, які представлені на рис. 4.2.

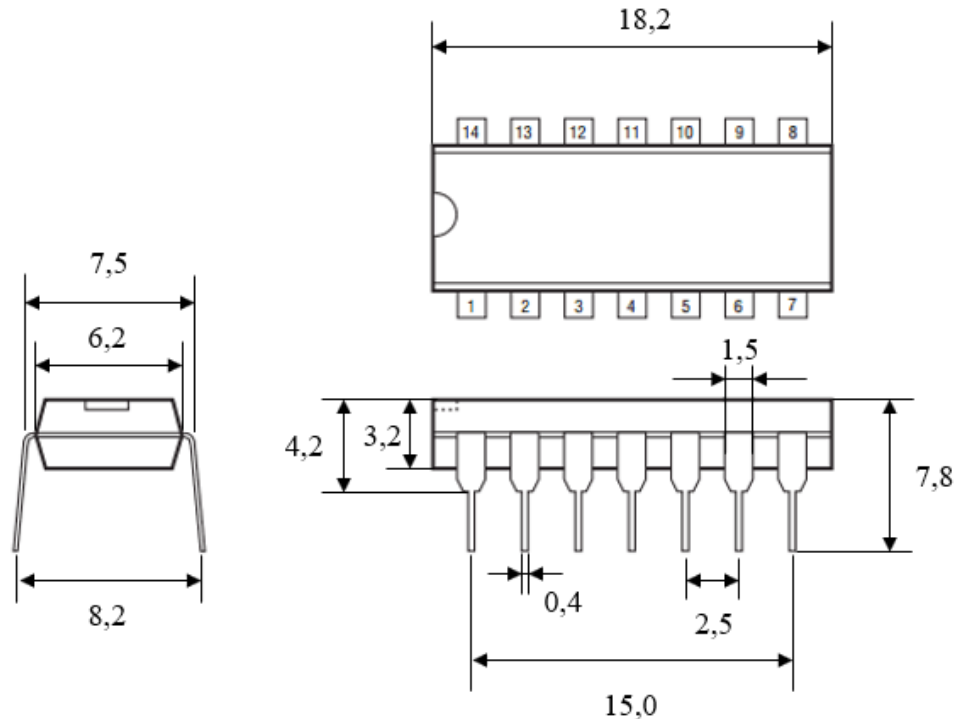


Рис. 4.2. Геометричні характеристики⁵³ корпусу мікросхеми K155ЛА3 (усі розміри дані в міліметрах)

2. Завдання на комп'ютерний практикум

1. Створити УГП логічного елементу 2І-НЕ та зберегти його у власній бібліотеці.
2. Створити ПМ для мікросхеми **K155ЛА3** у корпусі DIP.
3. Об'єднати графічне зображення логічного елементу 2І-НЕ та ПМ мікросхеми **K155ЛА3** для створення завершеного компонентного модуля.

⁵³ Треба враховувати, що під час виготовлення партії мікросхем допустимий певний розкид геометричних характеристик корпусу. Також корпуси, що виготовлені різними виробниками можуть мати дещо інші габаритні розміри. Також стандарти на корпус регламентують певні допуски і за такими параметрами як: товщина і ширина корпусу, довжина ніжок тощо. У цьому комп'ютерному практикумі буде створене посадкове місце для мікросхеми K155ЛА3, реальна довжина корпусу якої становить 18,2 мм (допустимий максимальний розмір згідно стандартів не повинен перевищувати 19,5 мм).

3. Порядок виконання комп'ютерного практикуму

3.1. Створення УГП логічного елементу 2І-НЕ, що є структурним елементом мікросхеми K155ЛА3

Відкрийте **Altium Designer**. Активуйте бібліотеку символів і перейдіть на вкладку **SCH Library**. Створіть копію компонента **Blank_Component**, яку перейменуйте на **K155ЛА3**.

В графічній області накресліть контур елементу у вигляді прямокутника розміром 8x10 мм (ширина - висота), використовуючи команду **Place** → **Line**. Відкрийте властивості лінії та встановіть колір зі списку під номером 221.

Додайте виводи до контуру елемента, використовуючи команду **Place** → **Pin**. У діалоговому вікні **Place Pin** у розділі **Designator** увімкніть опцію **Visible** (активувати видимість на схемі номерів виводів), а опцію **Display Name** вимкніть. Переконайтесь, що довжина виводу **Length** має значення 5 мм. Натисніть кнопку **ОК** та розмістіть перший вивід елементу на лівій бічній грані на відстані 2,5 мм від верхнього краю контуру. Аналогічно додайте другий вивід (в полі **Designator** введіть цифру 2) на лівій бічній грані на відстані 2,5 мм від її нижнього краю контуру. А третій вивід розмістіть по центру правої бічної грані (у полі **Designator** введіть цифру 3), змінивши для нього опцію у переліку **Outside Edge** на **Dot**.

Викличте команду **Place** → **Text**. У діалоговому вікні в полі **Text** наберіть символ **&** (шрифт Times New Roman, 14 pt) та розмістіть його приблизно посередині прямокутника, що позначає контур логічного елементу 2І-НЕ (рис. 4.3).

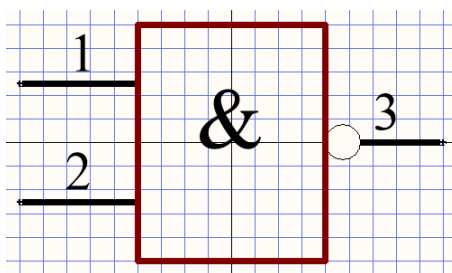


Рис. 4.3. УГП окремого логічного елементу мікросхеми K155ЛА3

Після створення першого елементу 2І-НЕ мікросхеми **K155ЛА3** необхідно створити ще три, при цьому позиційні позначення виводів кожного елементу повинні збігатися з реальними номерами виводів мікросхеми. Для створення нового елементу всередині одного компонентного модуля виконується команда **Tools** → **New Part**. Для інших елементів мікросхеми створіть такий самий УГП, як

і для першого (можна використовувати копіювання символу), але нумерація виводів має відповідати тій, що наведена на рис. 4.1.

Виводи 7 і 14 мікросхеми **K155ЛА3** належать до ланцюгів землі та живлення і не повинні бути відображені на схемі під час додавання символу логічного елемента 2І-НЕ. Для того, щоб приховати вивід, необхідно спочатку його створити, після чого викликати вікно властивостей⁵⁴ та увімкнути опцію **Hide**. Також у тому ж вікні властивостей у полі **Connect To** необхідно задати назву ланцюга, до якого за замовчуванням буде підключатися цей вивід. Крім цього, для таких виводів зазвичай ставлять значення 0 в полі **Part Number**, що говорить про відсутність їх приналежності до якогось конкретного структурного елемента мікросхеми (рис. 4.4). Виводи, через які мікросхема з'єднується із землею або живленням, можуть бути додані до УГП у будь-якому вільному місці робочої області будь-якого структурного елемента, оскільки вони все одно будуть невидимі (але краще встановлювати їх ближче до контуру елемента). Під час розробки компонентного модуля мікросхеми **K155ЛА3** рекомендується вивід 7 встановити і налаштувати у робочій області другого структурного елемента 2І-НЕ (в полі **Connect To** для цього виводу написати **GND**), а вивід 14 – у робочій області четвертого (в полі **Connect To** для цього виводу написати **Vcc**).

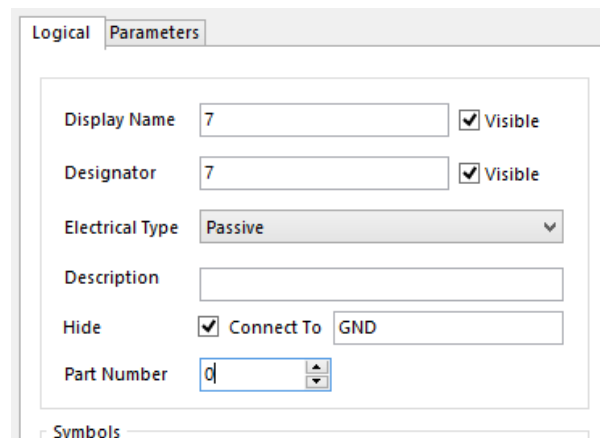


Рис. 4.4. Налаштування прихованого виводу 7 (GND) мікросхеми K155ЛА3

Відкрийте вікно властивостей компоненту **K155ЛА3** та у полі **Default Designator** введіть текст **DD?**, у полі **Description** – текст **TTL Quad 2-Input NAND**, у полі **Default Comment** встановіть символ «*». Для полів **Default Comment** та **Default Designator** активуйте властивість **Visible**.

⁵⁴ Відкрити вікно властивостей виводу можна безпосередньо під час його створення. Для цього, під час виконання команди **Place** → **Pin**, необхідно натиснути клавішу <Tab>, після чого автоматично відкриється вікно властивостей, які можна одразу ж редагувати.

Зауваження!

Зверніть увагу, що вивід живлення було позначено як Vcc. Взагалі, ланцюги живлення можуть бути позначені будь-яким зрозумілим розробнику чином, наприклад, V+, +V, Uж, U₀ тощо. Головна вимога до іменування виводів живлення полягає у тому, щоб було зрозуміло, до якого джерела підключатимуться виводи (причому важливо враховувати, що до одного джерела можуть бути підключені кілька мікросхем). Однак, щоб не виникало непорозумінь між розробником схеми, та тим, хто читає схему, або тим, хто на її основі проєктує плату, дотримуються деяких угод щодо іменування ланцюгів живлення. Зокрема, обране нами позначення Vcc розшифровується наступним чином: «V» – означає voltage (напруга), «с» – означає collector (колектор⁵⁵), а подвійне «сс» є даниною деякій традиції, яка вказує на те, що це вхідна напруга.

3.2. Створення посадкового місця типу DIP для мікросхеми K155ЛА3

Використовуючи геометричні характеристики, що наведені на рис. 4.5, створіть ПМ для мікросхеми K155ЛА3.

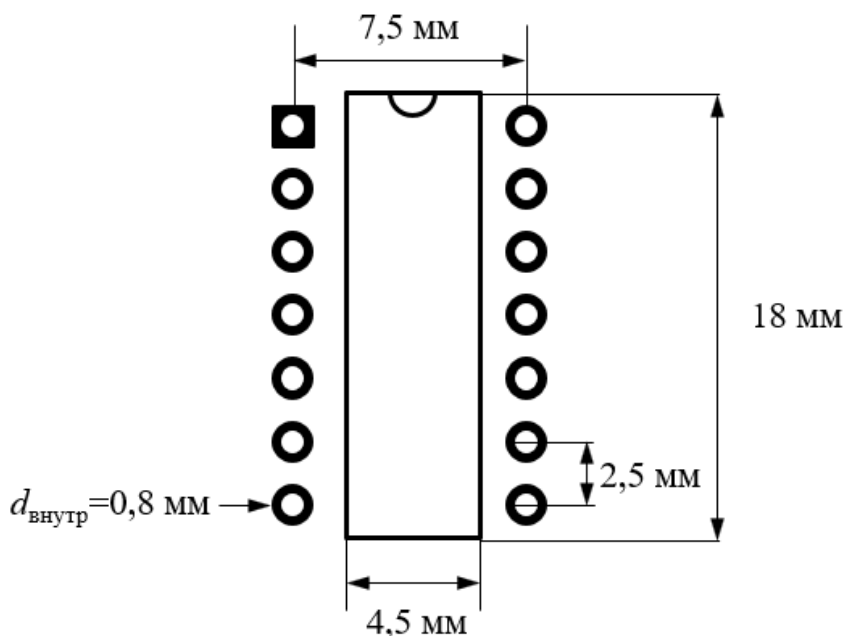


Рис. 4.5. Геометричні характеристики посадкового місця мікросхеми K155ЛА3

⁵⁵ Колектор – це один з виводів біполярного транзистора. Саме на цей вивід зазвичай подається живлення.

Перейдіть у бібліотеку посадкових місць. Створіть нову заготовку ПМ, виконавши команду **Tools** → **New Blank Component**. Встановіть основним кроком сітки 2,5 мм. Викличте команду **Place** → **Pad** та натисніть клавішу <Tab>. Задайте наступні параметри КМ: форма – кругла; ширина (**X-Size**) – 1,4 мм; висота (**Y-Size**) – 1,4 мм; діаметр (**Hole Size**) – 0,8 мм; номер виводу (**Designator**) – 1.

Згідно зі специфікацією мікросхеми встановіть чотирнадцять КМ за допомогою команди **Place** → **Pad** у два ряди по сім КМ у кожному: відстань між рядами – 7,5 мм, відстань між окремими КМ в ряді – 2,5 мм (див. рис. 4.6). Після встановлення усіх КМ змініть контур тільки першого із них на прямокутний.

Зауваження!

Інколи ми можемо створити кілька об'єктів, що є частиною ПМ (наприклад, тих же КМ), а потім помітити, що вони мають не ті характеристики, які мали б бути (наприклад, діаметр внутрішнього отвору прохідного КМ мав би бути більшим). Що робити у такому випадку? Можна видалити усі об'єкти і створити заново із потрібними характеристиками. Або можна відкривати по черзі властивості кожного із них і змінювати їх окремо. Обидва підходи не можна назвати раціональними. У випадку необхідності зміни деяких параметрів групи однотипних об'єктів, що є частиною ПМ, краще скористатись інструментом **PCBLIB Inspector**⁵⁶. Якщо спочатку викликати **PCBLIB Inspector**, а потім виділити групу необхідних об'єктів, то у вікні **PCBLIB Inspector** будуть відображені загальні параметри, редагування яких призведе до однакової зміни в усіх об'єктах.

Встановіть основним кроком сітки 0,5 мм. У рядку стану відкрийте список шарів і оберіть поточним шар **Top Overlay**. Намалюйте контур ПМ посередині між двома рядами КМ у вигляді прямокутника розмірами 4,5х18 мм (переконайтесь, що товщина поточної лінії **Line Width** становить 0,2 мм). У верхній частині прямокутника намалюйте «ключ» у вигляді півкола радіусом 1 мм (для кращого позиціонування півкола по центру верхньої частини прямокутника необхідно використовувати крок сітки 0,25 мм). Остаточний результат показано на рис. 4.6.

Збережіть у власну бібліотеку розроблене ПМ під іменем **DIP14_W7.5_X2.5**.

⁵⁶ Необхідно у меню **View** обрати пункт **Workspace Panels** → **PCB** → **PCBLIB Inspector**.

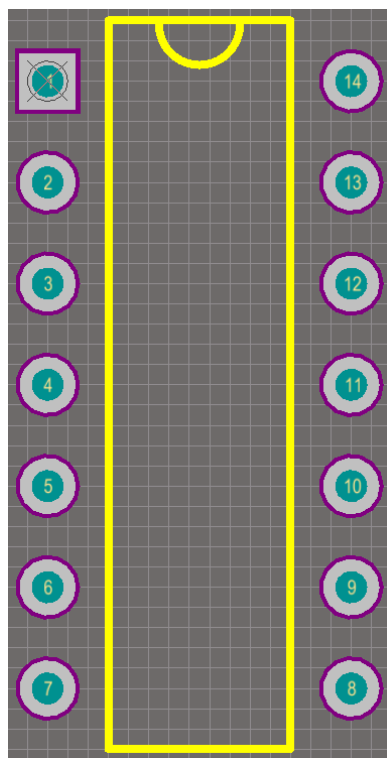


Рис. 4.6. Посадкове місце мікросхеми K155JA3

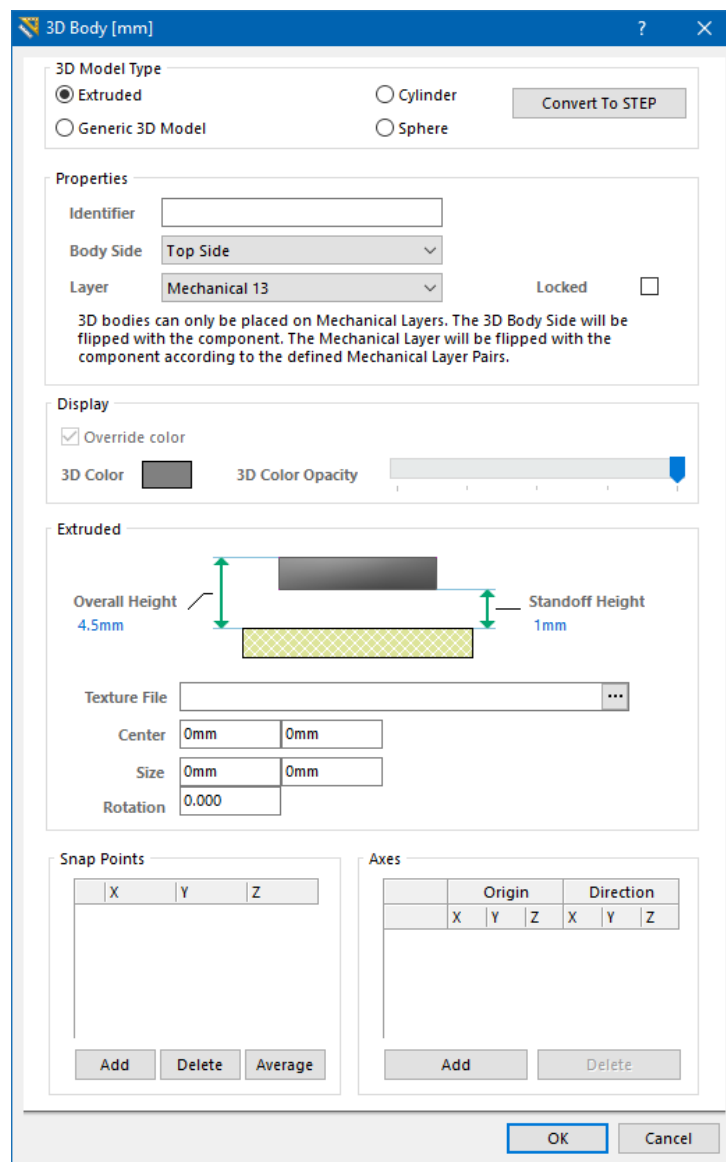
3.3. Створення тривимірної моделі корпусу мікросхеми K155JA3

Для створення простої тривимірної моделі корпусу мікросхеми необхідно додати об'єкт **3D Body** до моделі ПМ.

Важливо!

Об'єкти **3D Body** повинні розміщуватись на механічних шарах (*mechanical layer*) плати, причому можна обрати будь-який механічний шар. Тому, перш ніж додавати такий об'єкт, необхідно встановити поточним який-небудь із наявних механічних шарів. Однак, обираючи той чи інший механічний шар для розміщення тривимірної моделі, необхідно враховувати, що для нормального відображення на такому шарі не повинні міститись ніякі інші графічні примітиви. У цьому і наступних комп'ютерних практикумах ми будемо використовувати для розміщення тривимірних моделей шар *Mechanical 13*. Вибір механічного шару під номером 13 пов'язаний лише із тим, що саме на цьому шарі розміщуються тривимірні моделі компонентів із стандартної бібліотеки та компонентів, які створюються за допомогою майстра *IPC Compliant Footprint Wizard*.

Щоб створити тривимірну модель, необхідно активувати заданий механічний шар та виконати команду **Place** → **3D Body** (або послідовно натиснути клавіші <P, B>). Відкриється вікно **3D Body** (рис. 4.7). Об'єкти **3D Body** можуть бути створені з найпростіших форм (екструдованої прямокутної, циліндричної та сферичної) або за допомогою імпортування моделі STEP. За необхідності можна використовувати комбінацію різних методів.

Рис. 4.7. Вікно **3D Body**

У цьому комп'ютерному практикумі буде створена спрощена тривимірна модель корпусу мікросхеми **K155ЛA3** у вигляді прямокутного паралелепіпеда. У вікні **3D Body** оберіть тип моделі **Extruded**, встановивши відповідний прапорець, та введіть наступні параметри: **Overall Height** (загальна висота) – 4,5 мм, **Standoff Height** (мінімальний зазор між корпусом та поверхнею ДП) – 1 мм. Із випадного

списку **Layer** оберіть 13-й механічний шар⁵⁷. Після натискання кнопки **ОК** вікно закриється і розпочнеться процес розміщення багатокутного об'єкта в робочій області. Намалюйте прямокутник розмірами 19,5х6,5 мм та розмістіть⁵⁸ його між рядами отворів ПМ (рис. 4.8а), причому контур буде частково перекривати отвори. Щоб переглянути, яким чином виглядатиме створена тривимірна модель (рис. 4.8б), необхідно виконати команду **View** → **3D Layout Mode** (або натиснути клавішу <3>).

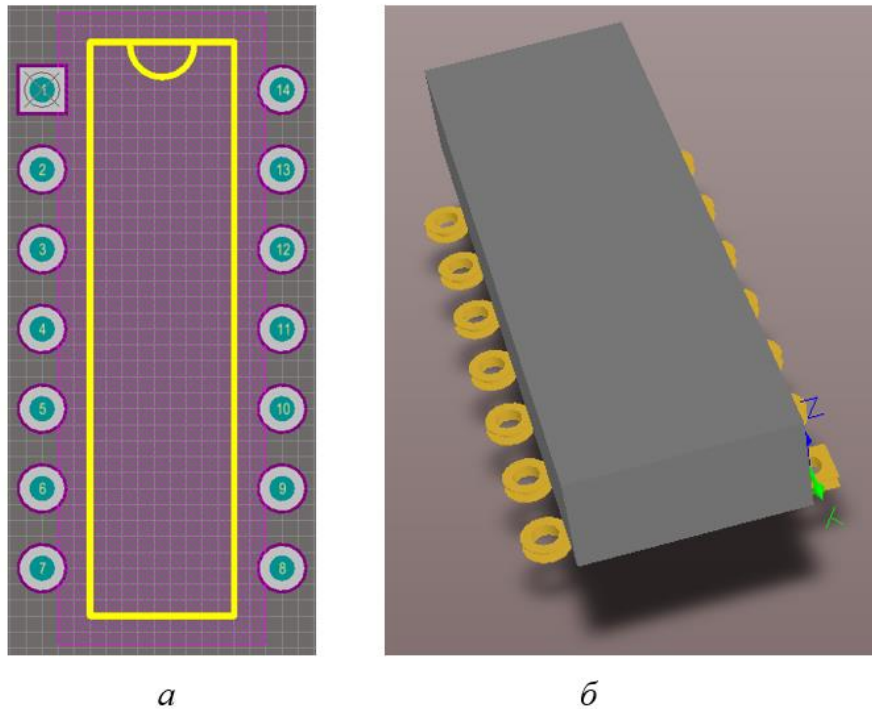


Рис. 4.8. Мікросхема К155ЛА3: а – контур корпусу мікросхеми у режимі 2D, б – вигляд корпусу мікросхеми у режимі 3D

⁵⁷ Загалом, якщо ви попередньо зробили активним шар **Mechanical 13**, то він автоматично буде відображатись у випадному списку **Layer**. Проте, якщо 13-й механічний шар не було додано взагалі до проєкту, то, по-перше, ви не зможете зробити його активним, а, по-друге, його не буде серед переліку шарів у випадному списку **Layer**. Для того, щоб додати до проєкту шар **Mechanical 13**, потрібно зайти у меню **Tools** та обрати пункт **Layers & Colors**. У вікні **View Configurations** в області, де розташований перелік тільки механічних шарів, приберіть галочку з пункту **Only show enabled mechanical layers**. Тоді стануть доступні для редагування усі існуючі механічні шари. Для того, щоб деякий шар став активним і доступним, необхідно встановити галочки в пунктах **Show** та **Enable**.

⁵⁸ Процес розміщення контуру мікросхеми в режимі 2D представляє собою стандартний процес розміщення багатокутного об'єкта. Для того, щоб змінити активний кут розміщення, використовується клавіша <Пробіл>, а комбінація <Shift + Пробіл> дозволяє перемикає стиль кута.

Зауваження!

Для обертання тривимірної моделі корпусу електронного компонента необхідно рухати мишею, утримуючи натиснутою клавішу <Shift> і ПКМ. Під час натискання клавіші <Shift> в позиції курсору з'являється сфера з напрямками. Поворот моделі здійснюється відносно центру сфери.

3.4. Створення компонентного модуля мікросхеми K155ЛА3

Перейдіть у бібліотеку УГП. Відкрийте вікно властивостей компоненту **K155ЛА3**. Натисніть на маленьку стрілку праворуч від кнопки **Add** у розділі **Model** діалогу **Library Component Properties** і оберіть зі списку рядок **Footprint** (рис. 4.9).

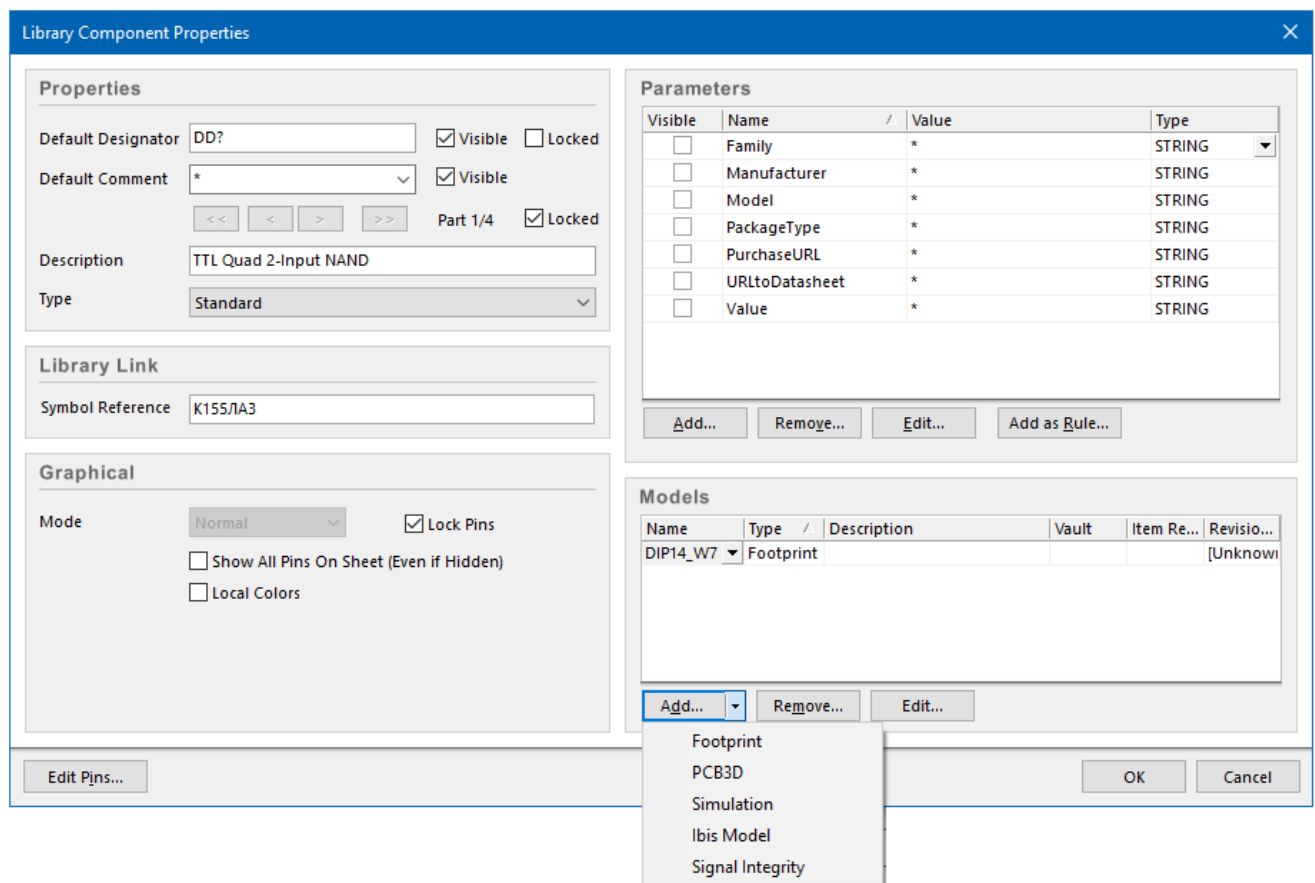


Рис. 4.9. Додавання моделі посадкового місця до символу компонента

У новому вікні **PCB Model**, що відкриється, натисніть кнопку **Browse**, щоб відкрити діалог **Browse Libraries**. Оберіть потрібну модель ПМ та натисніть кнопку **OK**, щоб її додати. Інформація про додану модель з'явиться у розділі **Model** в нижній частині робочого простору (рис. 4.10).

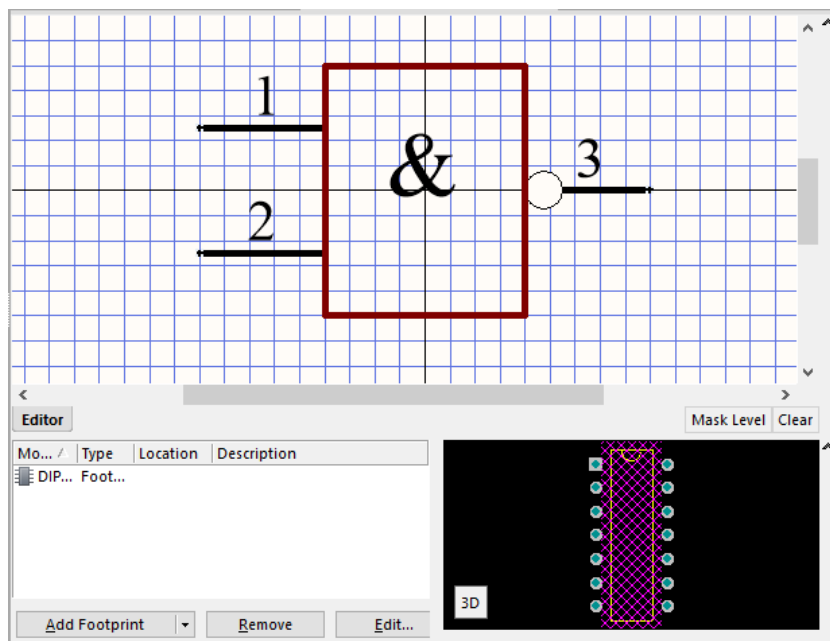


Рис. 4.10. Компонент разом із доданою моделлю посадкового місця

Після додавання моделі необхідно перевірити відповідність нумерації виводів електронного компонента номерам КМ (*Pin Mapping*). Для цього знову відкрийте діалог *PCB Model*, у якому натисніть кнопку **Pin Map**, щоб активувати інформаційне вікно *Model Map* (рис. 4.11).

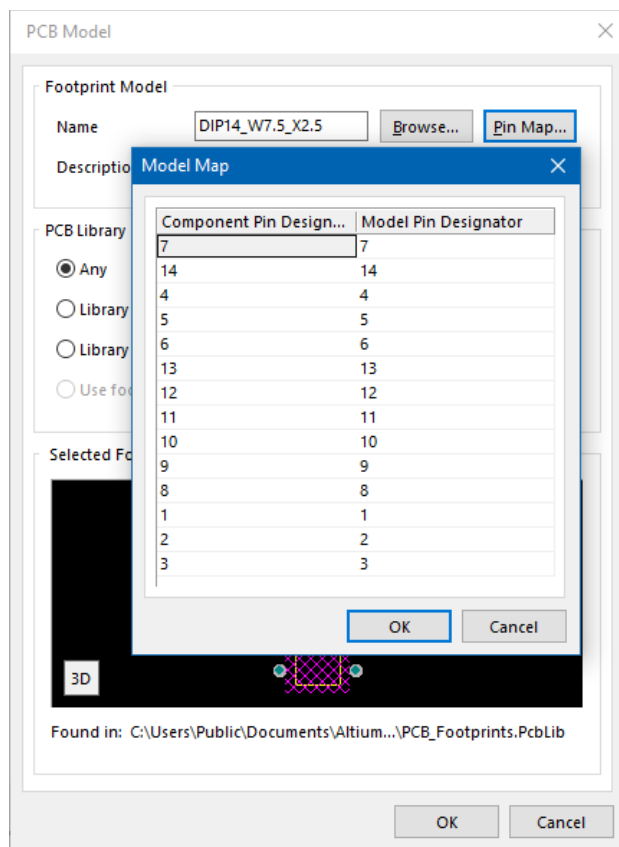


Рис. 4.11. Остаточний вигляд таблиці відповідності номерів виводів мікросхеми номерам контактних майданчиків посадкового місця

Перевірте і виправте, якщо необхідно, відповідність номерів виводів компонента (*Component Pin Designator*) номерам КМ моделі ПМ (*Model Pin Designator*).

Після створення компоненту необхідно його перевірити на відсутність помилок. Для цього виконайте команду **Reports** → **Component Rule Check** як у Редакторі символів, так і у Редакторі посадкових місць. Така перевірка фіксує найпростіші помилки: наявність дубльованих виводів, атрибутів тощо.

Зауваження!

Загальна ж перевірка бібліотеки компонентів виконується під час її компіляції.

4. Питання для самоконтролю

1. Поясніть логічну функцію, яку виконує елемент 2I-HE.
2. Що таке транзисторно-транзисторна логіка?
3. Опишіть характеристики мікросхеми K155ЛА3.
4. Яким чином формується позначення ланцюгів живлення мікросхем?
5. Яким чином додаються до УГП компонента виводи живлення та землі?
6. Опишіть процес створення посадкового місця мікросхеми K155ЛА3.
7. Яким чином створюється найпростіша тривимірна модель корпусу мікросхеми засобами **Altium Designer**?
8. З яких міркувань і яким чином обирається механічний шар для розміщення тривимірної моделі корпусу мікросхеми?
9. Яким чином виконується перевірка розробленого компоненту на наявність помилок?

Комп'ютерний практикум № 5.

РОЗРОБКА КОМПОНЕНТНОГО МОДУЛЯ ТАЙМЕРА 555

Мета роботи: ознайомитись з конструкцією та призначенням мікросхеми таймера 555; створити умовне графічне позначення та посадкове місце для таймера 555.

1. Теоретичні відомості

Timer 555 – аналогова інтегральна схема, що призначена для формування (генерації) одиночних і повторюваних імпульсів зі стабільними часовими характеристиками. Застосовується для побудови різних генераторів, модуляторів, реле часу, порогових пристроїв та інших вузлів електронної апаратури. Вперше цей таймер був випущений у 1971 році компанією Signetics та мав назву NE555. Функціональні аналоги оригінального NE555 випускаються в безлічі варіантів (як із використанням технології TTL, так і з використанням КМОН-технології). Здвоєна версія таймеру 555 випускається під позначенням 556, а мікросхема, в якій розміщені 4 таймери, – під позначенням 558.

Таймер 555 може або одноразово видавати імпульси певної довжини, або видавати імпульси постійно, через задані проміжки часу. Режим роботи і параметри вихідного сигналу залежать від підключеної до виводів мікросхеми «обв'язки», яка будується з конденсаторів і резисторів.

На схемах таймер 555 зазвичай позначається у вигляді прямокутника з написом «G» або «GN», що розшифровується як спеціалізований генератор, який використовується для формування одиночних імпульсів або серій імпульсів.

Мікросхема таймера 555 складається з дільника напруги із двома опорними напругами для порівняння, двох прецизійних компараторів (низького і високого рівнів), RS-тригера з додатковим входом скидання, транзисторного ключа з відкритим колектором і вихідного підсилювача потужності для збільшення навантажувальної здатності. На рис. 5.1 наведена внутрішня будова мікросхеми NE555, яка випускається фірмою Fairchild Semiconductor, із вказівкою нумерації виводів та їх найменуванням.

Таймер 555 найчастіше випускається у корпусах для монтажу в отвори DIP8 та поверхневого монтажу SOIC8 або SOP8, але зустрічаються й інші варіанти корпусів.

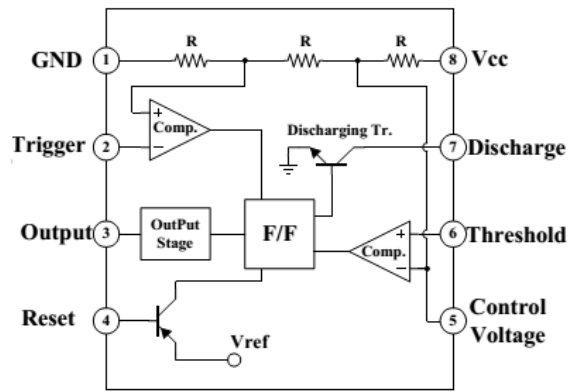


Рис. 5.1. Внутрішня будова мікросхеми NE555

2. Завдання на комп'ютерний практикум

1. Створити УГП мікросхеми таймера 555.
2. Створити ПМ для мікросхеми таймера **NE555** (виробник Fairchild Semiconductor) в корпусах DIP та SOP згідно геометричних характеристик, наведених на рис. 5.2 та рис. 5.3.
3. Об'єднати УГП та ПМ таймера 555 для створення завершених бібліотечних компонентних модулів.

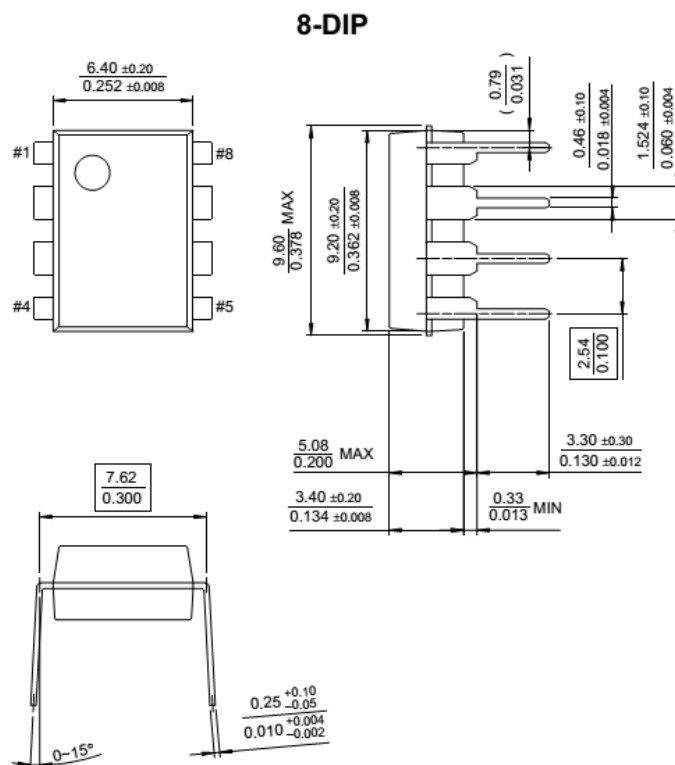


Рис. 5.2. Геометричні характеристики корпусу мікросхеми NE555 (Fairchild Semiconductor) у корпусі DIP: розміри вказані у міліметрах (над символом дробу) та у дюймах (під символом дробу)

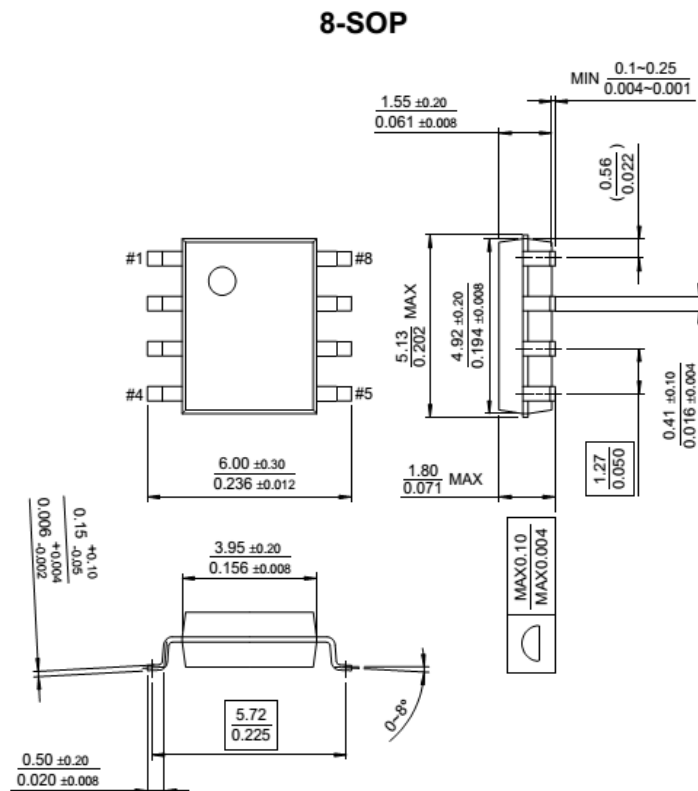


Рис. 5.3. Геометричні характеристики корпусу мікросхеми NE555 (Fairchild Semiconductor) у корпусі SOP: розміри вказані у міліметрах (над символом дробу) та у дюймах (під символом дробу)

Зауваження!

Мікросхема таймеру 555 від Fairchild Semiconductor у корпусі DIP має назву NE555N, а у корпусі SOP – NE555D. Оскільки УГП у них однакові, то спочатку буде створено УГП бібліотечного компоненту NE555N, після чого розроблене УГП буде скопійовано у компонент NE555D.

3. Порядок виконання комп'ютерного практикуму

3.1. Створення УГП таймеру NE555

Відкрийте **Altium Designer**. Активуйте вкладку **SCH Library** і створіть новий компонент з іменем **NE555N**.

В графічній області накресліть контур елементу за допомогою команди **Place** → **Line** у вигляді прямокутника розміром 20x25 мм (ширина - висота). Всередині прямокутника намалюйте дві вертикальні лінії на відстані 5 мм від кожного краю. Відкрийте властивості ліній та встановіть колір зі списку під номером 221.

Додайте виводи елементу довжиною 5 мм за допомогою команди **Place** → **Pin**. Налаштуйте відображення імен виводів (у полі **Display Name** та **Designator** увімкніть мітки **Visible**). Розмістіть чотири виводи з лівого боку корпусу таймера через кожні 5 мм (відстань від верхнього/нижнього краю становить 5 мм) та два – з правого боку на відстані 10 мм один від одного (відстань від верхнього/нижнього краю становить 7,5 мм). Для кожного виводу відкрийте меню **Pin Properties** та змініть назву відповідно до табл. 5.1.

Таблиця 5.1. Нумерація та назви виводів таймеру

Designator	2	3	4	5	6	7
Display Name	TRIG	OUT	RST	CON	THR	DIS

Для назв виводів підберіть шрифт таким чином, щоб усі написи вмістились по ширині в додаткових полях УГП. Для цього у властивостях кожного виводу активуйте прапорці **Customize Position** і **Use local font setting**, після чого з'явиться можливість їх редагування.

Виконайте команду **Place** → **Text**. У діалоговому вікні в полі **Text** надрукуйте літеру **G**, яку потім розташуйте у верхній частині основного поля УГП. Остаточний результат представлено на рис. 5.4.

Вивід 1 та 8 розмістіть у будь-якій частині робочої області, в якій намальоване УГП таймеру. У властивостях цих виводів після встановлення відповідного позначення і назви увімкніть опцію **Hide**, а в полі **Connect To** для виводу 1 напишіть **GND**, для виводу 8 – **Vcc**.

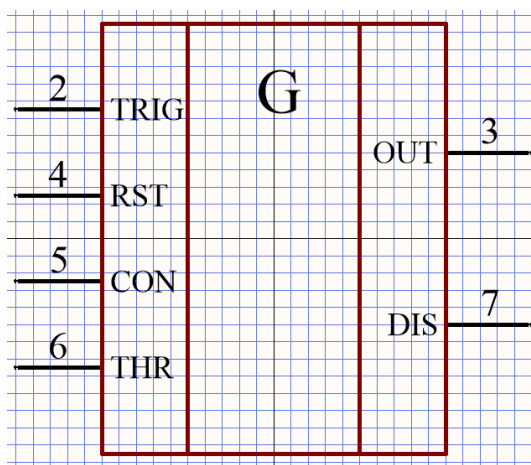


Рис. 5.4. УГП таймера NE555

Відкрийте вікно властивостей компоненту **NE555N** та у полі **Default Designator** введіть текст **DD?**, у полі **Description** – текст **Timer 555**, у полі **Default**

Comment встановіть символ «*». Для полів **Default Comment** та **Default Designator** активуйте властивість **Visible**.

Для створення компоненту **NE555D** виділіть у вікні **Components** вкладки **SCH Library** новостворений компонент **NE555N**, натисніть комбінацію клавіш **<Ctrl + C>**, потім **<Ctrl + V>**. Внаслідок цих дій з'явиться копія **NE555N_1**, яку необхідно перейменувати на **NE555D**.

3.2. Створення посадкових місць для мікросхеми NE555 у корпусі DIP та SOP із використанням майстра *Component Wizard*

Використовуючи геометричні характеристики, що наведені на рис. 5.5, створіть ПМ для мікросхеми **NE555**.

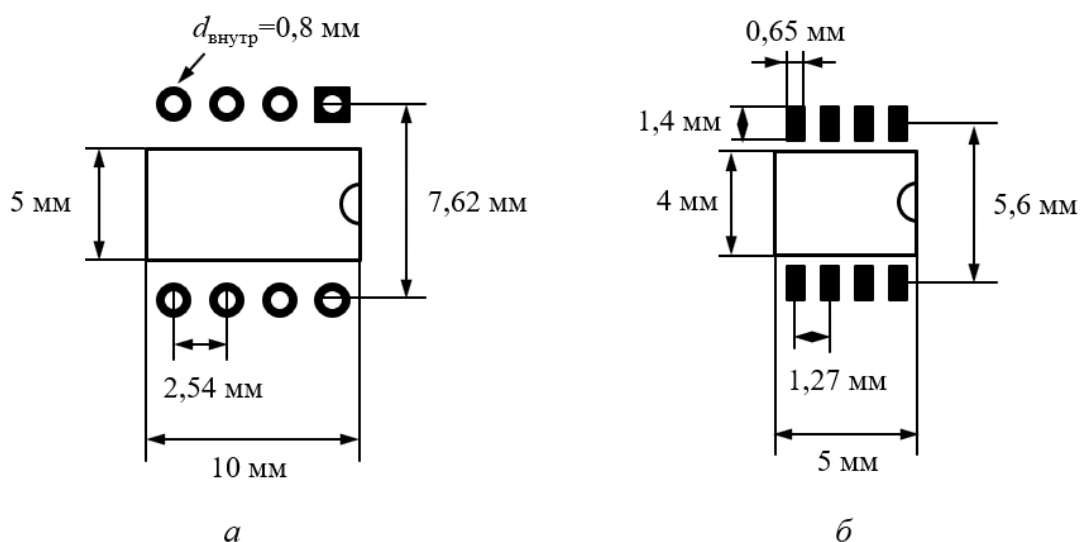


Рис. 5.5. Геометричні характеристики посадкового місця мікросхеми NE555:

а – у корпусі DIP8, б – у корпусі SOP8

3.2.1. Створення посадкового місця для мікросхеми таймеру у корпусі DIP

Перейдіть у бібліотеку посадкових місць. Запустіть [майстер створення посадкового місця *Component Wizard*](#), виконавши команду **Component Wizard...**⁵⁹ із меню **Tools**. Відкриється вікно із коротким описом призначення цього майстра. Натисніть **Next** для продовження.

У новому вікні, що відкриється (рис. 5.6), оберіть метричні одиниці вимірювань та тип корпусу DIP.

⁵⁹ Якщо використовується цей майстер, то не потрібно заздалегідь створювати нову заготовку посадкового місця за допомогою команди **Tools** → **New Blank Component**.

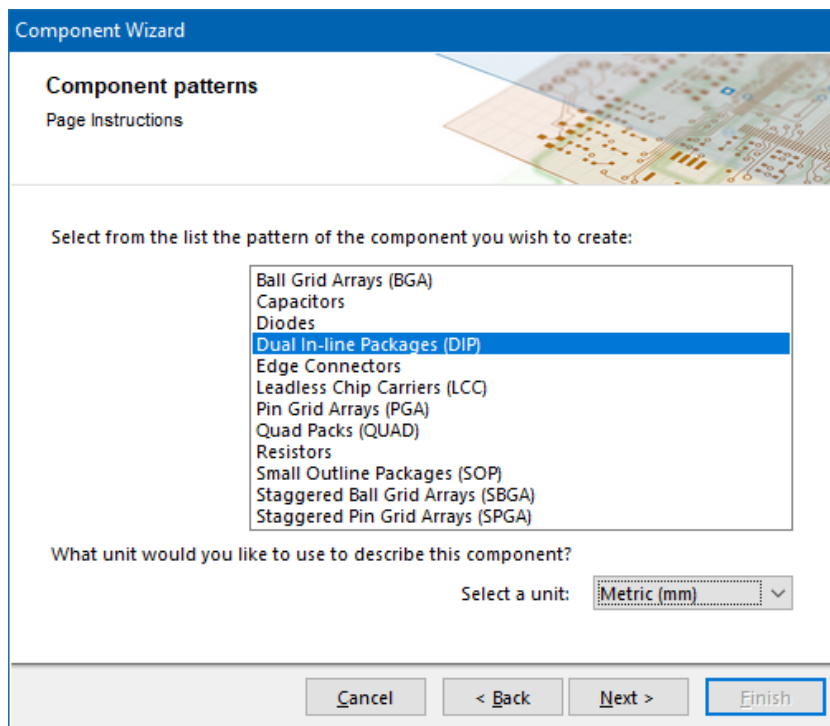


Рис. 5.6. Початкове вікно *Component Wizard*

Натисніть кнопку **Next** для переходу в наступне вікно (рис. 5.7), в якому задаються геометричні розміри отворів. Введіть значення, що вказані на рис. 5.7.

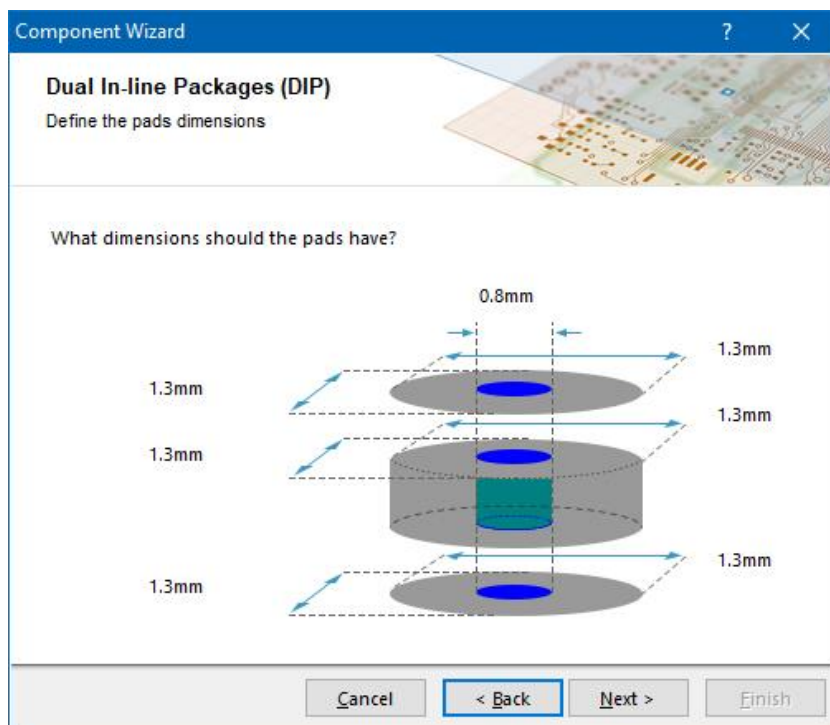


Рис. 5.7. Вікно вибору розмірів отворів для контактного майданчика

Натисніть кнопку **Next** для переходу в наступне вікно (рис. 5.8), в якому вказують відстань між центрами КМ вздовж кожного ряду виводів (введіть

значення 2,54 мм) та відстань між рядами КМ (введіть значення 7,62 мм). Зверніть увагу, що перший КМ буде мати прямокутну форму.

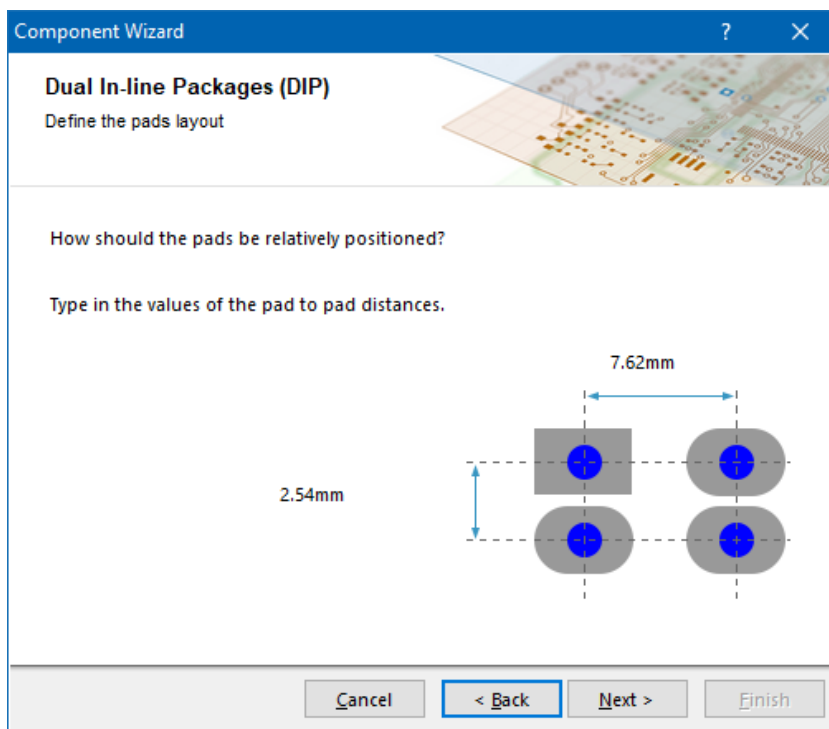


Рис. 5.8. Вибір відстані між контактними майданчиками по горизонталі та вертикалі

Натисніть кнопку **Next**, щоб перейти у наступне вікно (рис. 5.9), в якому вказують товщину ліній контуру ПМ на шарі **Top Overlay**. Встановіть товщину лінії рівною 0,2 мм.

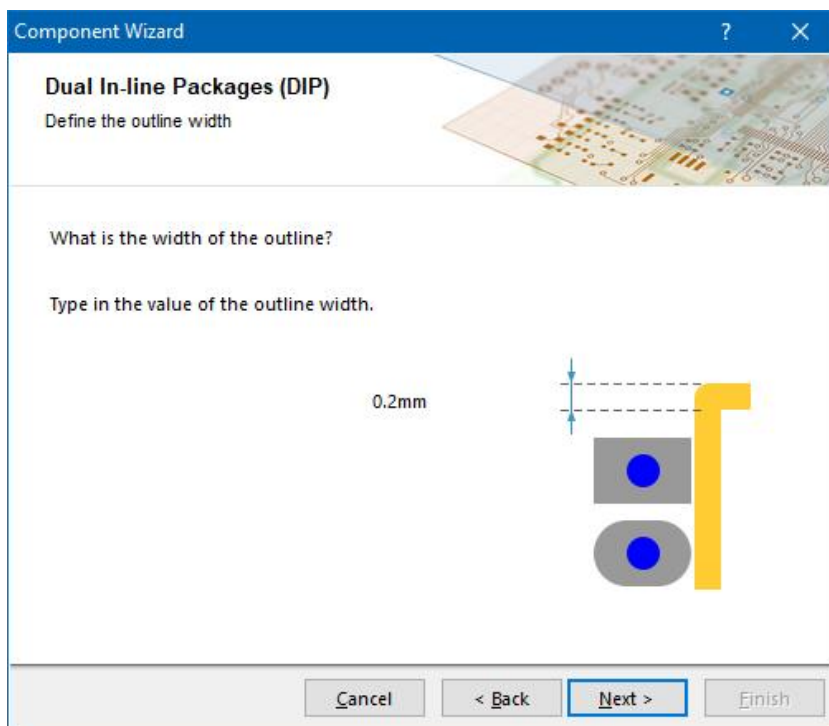


Рис. 5.9. Вибір товщини лінії контуру компонента

Натисніть кнопку **Next** для переходу в наступне вікно (рис. 5.10), в якому вказують загальну кількість виводів. Оберіть значення **8**.

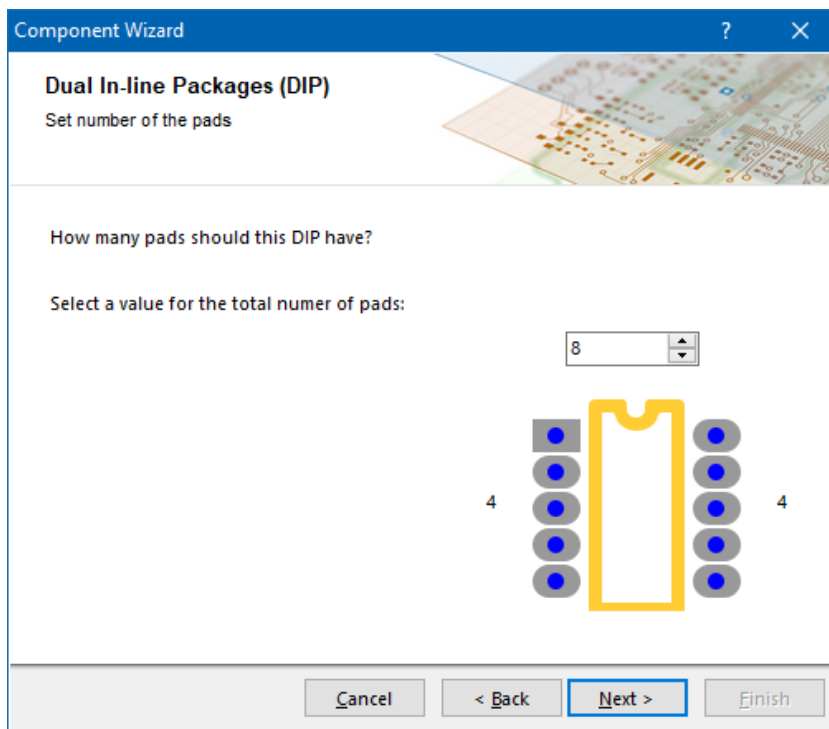


Рис. 5.10. Вікно вибору кількості виводів

Натисніть кнопку **Next** для переходу в останнє вікно (рис. 5.11), в якому задається ім'я ПМ. Задайте ім'я **DIP8_W7.62_X2.54**.

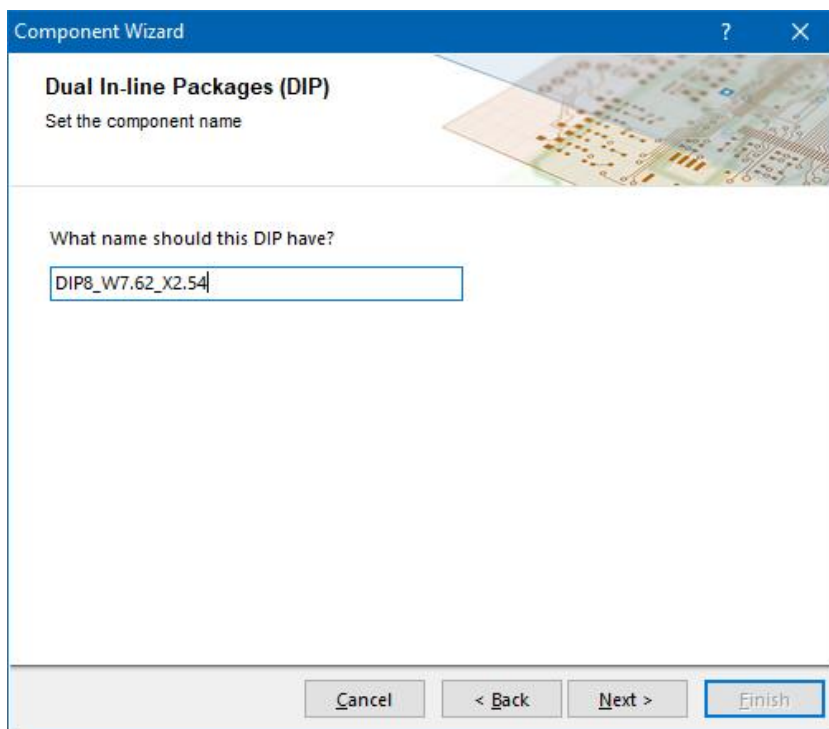


Рис. 5.11. Задання імені посадкового місця

Завершіть створення ПМ натисканням кнопки **Finish**. Остаточний результат представлений на рис. 5.12.

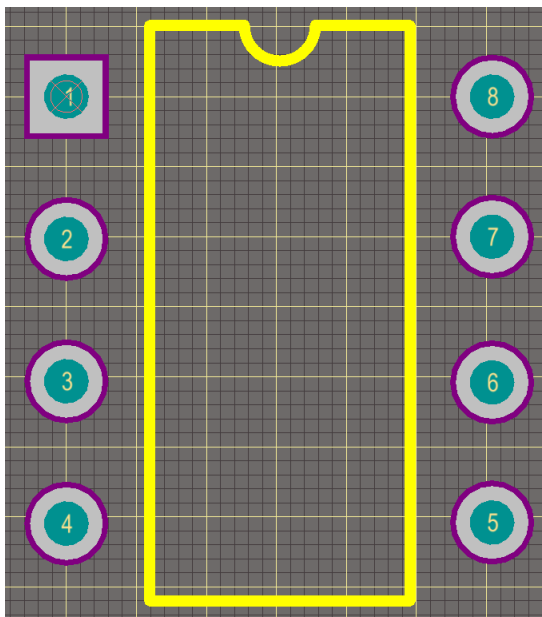


Рис. 5.12. Посадкове місце для таймеру в корпусі DIP8

Зауваження!

Доопрацювання корпусу здійснюється стандартними засобами графічного редактора Altium Designer.

3.2.2. Створення посадкового місця для мікросхеми таймеру у корпусі SOP

Знову запусіть майстер створення посадкового місця **Component Wizard**. Послідовність дій зі створення ПМ для мікросхеми у корпусі SOP буде аналогічною до того, як створювалось ПМ для корпусу DIP. Тому тут не буде описано докладно, як це робити. Самостійно, орієнтуючись на вже здобуті знання, створіть нове ПМ. Під час роботи із майстром використовуйте наступні параметри:

- **Pattern Type** – *Small Outline Packages (SOP)*;
- **Measurement Unit** – міліметри;
- **Pads Dimensions** – 1,4 мм x 0,65 мм;
- **Pad Distances** – 1,270 мм x 5,6 мм;
- **Outline Width** – 0,1 мм;
- **Number of Pads** – 8.

Збережіть створене ПМ під іменем **SOP8_W5.6_X1.27**. Остаточний вигляд ПМ, на який необхідно орієнтуватись, наведено на рис. 5.13.

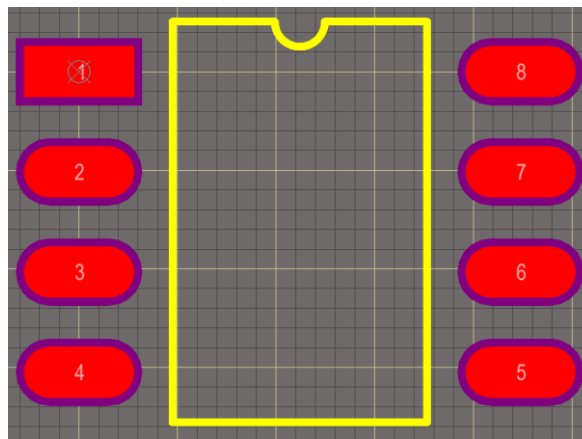


Рис. 5.13. Посадкове місце для мікросхеми таймеру в корпусі SOP8

3.3. Створення тривимірних моделей корпусів мікросхеми NE555

На основі даних про геометричні розміри корпусу DIP та SOP, що наведені на рис. 5.2 та рис. 5.3, створіть на 13-му механічному шарі прості тривимірні моделі корпусів засобами **Altium Designer**.

3.4. Створення посадкового місця для мікросхеми NE555 у корпусі SOP із використанням майстра *IPC Compliant Footprint Wizard*

В **Altium Designer** є спеціальний майстер [*IPC Compliant Footprint Wizard*](#), призначений для створення ПМ за стандартами IPC⁶⁰. Ідея використання цього майстра полягає у тому, що розробник вводить не безпосередньо геометричні розміри ПМ та КМ, а задає за допомогою відповідних форм фактичні розміри корпусу та виводів електронного компонента. Далі, на основі формул, розроблених для стандарту IPC-7351, майстер сам генерує ПМ, набір КМ та тривимірну модель корпусу електронного компонента.

Стандарт IPC-7351B «Generic Requirements for Surface Mount Design and Land Pattern Standard» встановлює вимоги до конструкції КМ основних типів пасивних і активних компонентів, включаючи резистори, конденсатори, мікросхеми в корпусах MELF, SSOP, TSSOP, BGA, QFP, QFN і SON. Стандарт передбачає кілька варіантів конструкції КМ для всіх типів корпусів компонентів залежно від

⁶⁰ IPC (The Institute for Interconnecting and Packaging Electronic Circuits) – міжнародна асоціація виробників електроніки, яка випускає стандарти для проєктування та виготовлення друкованих плат і електронних збірок.

щільності монтажу, містить рекомендації стосовно КМ для безсвинцевої пайки, а також до нових типів корпусів з сімейства QFN і SON.

Запустіть майстер *IPC Compliant Footprint Wizard*⁶¹, виконавши однойменну команду із меню *Tools*. Спочатку відкриється вікно привітання (рис. 5.14).

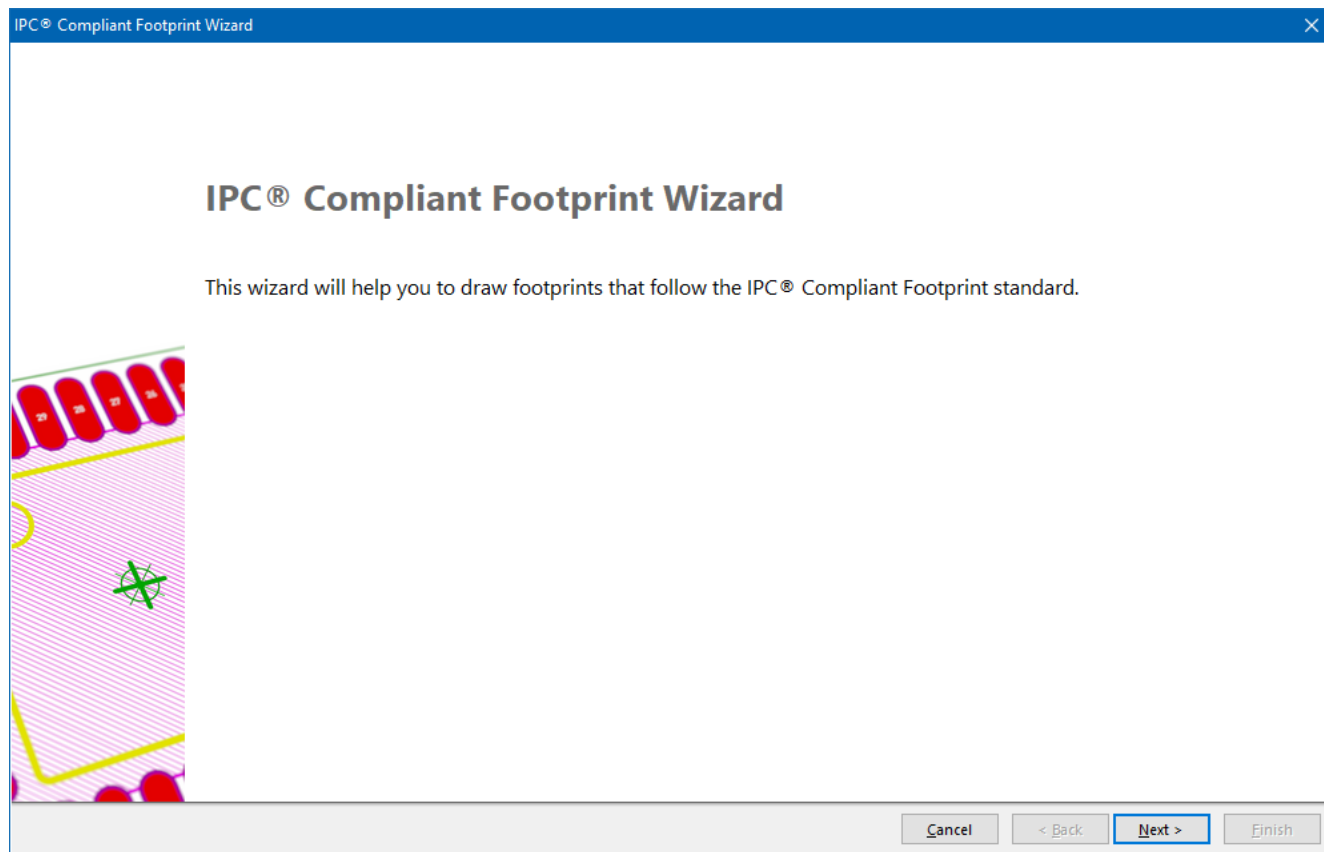


Рис. 5.14. Головне вікно майстра *IPC Compliant Footprint Wizard*

Після натискання кнопки **Next** відкриється нове вікно, у якому необхідно вибрати тип компоненту зі списку. Оберіть тип **SOP/TSOP (Small Outline Package - Gullwing Leads)**, після чого натисніть кнопку **Next**.

У новому вікні встановіть геометричні розміри корпусу мікросхеми та КМ таким чином, як це показано на рис. 5.15 (порівняйте ці дані зі значеннями, що наведені на рис. 5.3).

У цьому ж вікні (рис. 5.15) ви можете встановити прапорець **Generate STEP Model Preview** у нижньому лівому куті вікна, завдяки чому буде автоматично створена тривимірна модель корпусу за введеними даними. Якщо ви хочете побачити зображення 3D-моделі корпусу, то натисніть кнопку **3D** в області *Preview*.

⁶¹ У цьому випадку, як і під час використання майстра *Component Wizard*, немає необхідності створювати нову заготовку ПМ за допомогою команди *Tools* → *New Blank Component*.

Встановивши усі необхідні параметри у вікні *SOP/TSOP Package Dimensions*, натисніть кнопку **Next**.

IPC® Compliant Footprint Wizard

SOP/TSOP Package Dimensions
Enter the required package values.

Overall Dimensions

Width Range (H)	Minimum	5.7mm
	Maximum	6.3mm
Maximum Height (A)		1.8mm
Minimum standoff height (A1)		0.1mm
Body width range (E)	Minimum	3.75mm
	Maximum	4.15mm
Body length range (D)	Minimum	4.72mm
	Maximum	5.12mm

Pin Information

Number of pins	8	
Lead Width Range (B)	Minimum	0.31mm
	Maximum	0.51mm
Lead Length Range (L)	Minimum	0.3mm
	Maximum	0.7mm
Pitch (e)	1.27mm	

☒ Generate STEP Model Preview

Buttons: Cancel, < Back, Next >, Finish

Рис. 5.15. Встановлення геометричних параметрів корпусу мікросхеми та виводів

У наступному вікні *SOP/TSOP Package Thermal Pad Dimensions* переконайтеся, що прапорець *Add Thermal Pad* не встановлено, оскільки для корпусу SOP8 мікросхеми **NE555D** не використовуються тепловідвідні КМ⁶². Натисніть кнопку **Next**.

У вікні *SOP/TSOP Package Heel Spacing*, що з'явиться, має бути встановлений прапорець *Use calculated values*. Завдяки цьому майстер автоматично розраховуватиме відстань між рядами КМ згідно стандартів IPC. Натисніть кнопку **Next**.

У новому вікні *SOP/TSOP Solder Fillets* задається форма паяльного з'єднання (фактично, з'єднання виводу компонента із КМ), від якої залежить як міцність, так і надійність з'єднання. Параметри форми визначені стандартом IPC-7351 на основі випробувань надійності та галузевих емпіричних знань. Відповідно до цього

⁶² Термовідвідні (або терморельєфні) КМ – спеціальні КМ, які з'єднані із полігоном зовнішньої мідної заливки лише тонкими містками (у вигляді своєрідних «спиць»). Це запобігає витоку тепла на полігон міді під час припаювання компонента, а отже і негативним наслідкам, що можуть супроводжувати цей процес.

стандарту існує три рівні щільності плати, які для корпусів SOP мають наступні параметри форми паяльного з'єднання:

а) **Level A – Low density** (suggests higher minimal values for Toe, Heel and Side Fillets):

- Toe Fillet (JT Min) = 0.55 mm;
- Heel Fillet (JH Min) = 0.45 mm;
- Side Fillet (JS Min) = 0.05 mm.

б) **Level B – Medium density** (suggests medium minimal values for Toe, Heel and Side Fillets):

- Toe Fillet (JT Min) = 0.35 mm;
- Heel Fillet (JH Min) = 0.35 mm;
- Side Fillet (JS Min) = 0.03 mm.

в) **Level C – High density** (suggests lower minimal values for Toe, Heel and Side Fillets):

- Toe Fillet (JT Min) = 0.15 mm;
- Heel Fillet (JH Min) = 0.25 mm;
- Side Fillet (JS Min) = 0.01 mm.

Переконайтеся, що прапорець *Use default values* встановлено. Водночас для корпусу SOP8 мікросхеми **NE555D** у випадному списку *Board density Level* оберіть рядок **Level A – Low density**.

Зверніть увагу, що майстер має ще багато наступних вікон. Натискаючи кнопку **Next**, ви можете переглянути, які параметри можна встановити за їх допомогою. Проте зверніть увагу, що для більшості параметрів надалі використовуються прапорці на зразок *Use default values*, *Use calculated values* тощо. Тому вже після необхідних налаштувань у вікні *SOP/TSOP Solder Fillets* можна натиснути кнопку **Finish**, а не **Next**. Тоді майстер завершить роботу, застосувавши значення параметрів за замовчуванням, які налаштовуються у наступних вікнах.

Усе ж ми налаштуємо ще один параметр ПМ після завдання форми паяльного з'єднання. Для цього натискайте кнопку **Next** до тих пір, поки не з'явиться вікно *SOP/TSOP Silkscreen Dimensions*. У цьому вікні встановіть товщину лінії на шарі шовкографії рівною 0,1 мм (параметр *Silkscreen Line Width*). І вже у цьому вікні натисніть кнопку **Finish**. Остаточний вигляд створеного ПМ представлений на рис. 5.16.

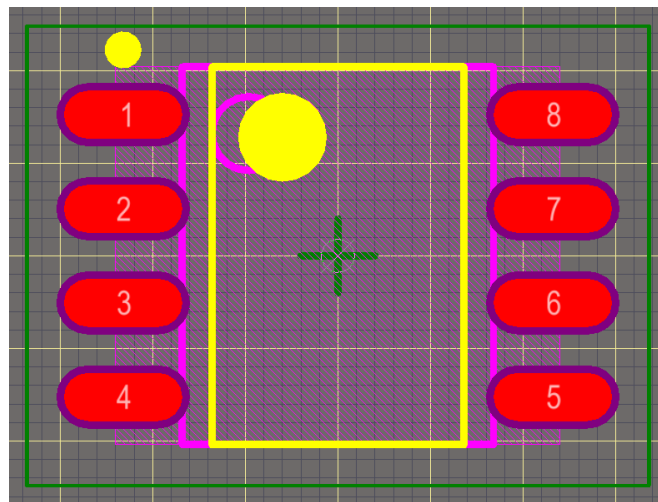


Рис. 5.16. Посадкове місце SOP8, створене за допомогою майстра
IPC Compliant Footprint Wizard

Зверніть увагу, що ім'я цього ПМ було задане автоматично програмою. Крім того, автоматично було додано опис, а у вікні властивостей відображені деякі геометричні характеристики (рис. 5.17).

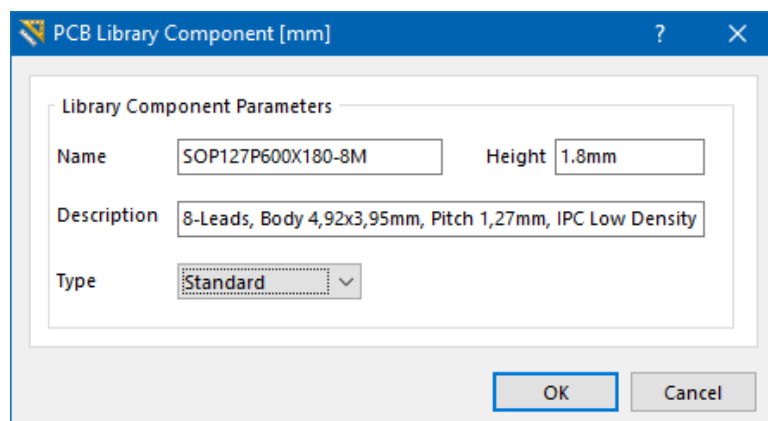


Рис. 5.17. Вікно властивостей створеного посадкового місця

3.5. Створення компонентного модуля мікросхеми NE555

Перейдіть у бібліотеку УГП. Аналогічно описаному в попередніх комп'ютерних практикумах до УГП компонентного модуля **NE555N** додайте ПМ під корпус для монтажу в отвори DIP, а до **NE555D** – обидва⁶³ ПМ під корпус для поверхневого монтажу SOP, що були створені за допомогою різних майстрів.

⁶³ Варто зауважити, що моделей корпусу та ПМ для однієї мікросхеми може бути декілька. В **Altium Designer** передбачено можливість додати одразу декілька різних моделей. Вибір конкретного корпусу і ПМ відбувається під час створення схеми.

Після додавання моделі ПМ необхідно перевірити відповідність номерів виводів мікросхеми номерам КМ, використовуючи діалог ***Pin Mapping***.

Також виконайте перевірку кожного компоненту за допомогою команди ***Reports → Component Rule Check***.

4. Питання для самоконтролю

1. Опишіть призначення таймеру 555.
2. Поясніть принцип роботи таймеру 555.
3. Наведіть приклади застосування таймеру 555.
4. Поясніть, що собою представляє тип корпусу SOP?
5. Поясніть, що собою представляє тип корпусу DIP?
6. Поясніть процес створення посадкового місця за допомогою майстра ***Pattern Wizard***.
7. Поясніть, для чого призначений майстер ***IPC Compliant Footprint Wizard***?
8. Поясніть процес створення посадкового місця за допомогою майстра ***IPC Compliant Footprint Wizard***.
9. Поясніть відмінності між майстрами створення посадкових місць ***Pattern Wizard*** та ***IPC Compliant Footprint Wizard***.

Комп'ютерний практикум № 6.

РОЗРОБКА КОМПОНЕНТНИХ МОДУЛІВ ОПЕРАЦІЙНИХ ПІДСИЛЮВАЧІВ LMH6714MF ТА AD642

Мета роботи: ознайомитись з теорією операційних підсилювачів; створити умовне графічне позначення операційного підсилювача; ознайомитись з конструкцією та призначенням мікросхем операційних підсилювачів LMH6714MF та AD642; створити посадкові місця для зазначених мікросхем.

1. Теоретичні відомості

Операційний підсилювач – це електронний підсилювач напруги з високим коефіцієнтом підсилення, що має диференціальний вхід і зазвичай один вихід. Напруга на виході може перевищувати різницю напруг на входах у сотні або навіть тисячі разів.

Основне призначення ОП – побудова схем з точно синтезованою передавальною функцією, яка залежить практично тільки від властивостей ланцюга зворотного зв'язку. На основі ОП створюються прецизійні масштабувальні підсилювачі, генератори сигналів, генератори функцій, стабілізатори напруги і струму, активні частотні фільтри, логарифмувальні підсилювачі, інтегратори та диференціатори тощо.

На рис. 6.1 показано схематичне зображення ОП. Зверніть увагу, що вигляд (або форма) УГП за українським стандартом відрізняється від західних стандартів. Також врахуйте, що позначення виводів на рис. 6.1 не є вичерпними і можуть іменуватись інакше; тут головне, щоб імена виводів були зрозумілими і відповідали їх функціональному призначенню. Ще один нюанс, який необхідно розуміти, полягає у тому, що виводи, які підключаються до джерел живлення, можуть бути як показані на УГП підсилювача (рис. 6.1б), так і приховані⁶⁴ (рис. 6.1а).

Для функціонування будь-якого ОП абсолютно необхідними є п'ять виводів: інвертуючий та неінвертуючий входи, вихід, входи додатного та від'ємного джерел

⁶⁴ На рис. 6.1 для УГП в українському стандарті приховані виводи живлення, а для УГП в західному стандарті – показані. Однак ви можете зробити і навпаки. Чи показувати, чи приховувати виводи живлення, залежить від розробника і правил, якими він керується під час створення принципової схеми приладу.

живлення. Крім цього, деякі ОП можуть мати додаткові виводи, призначені для: встановлення струму спокою, частотної корекції, балансування (корекції зсуву) та ряду інших функцій.

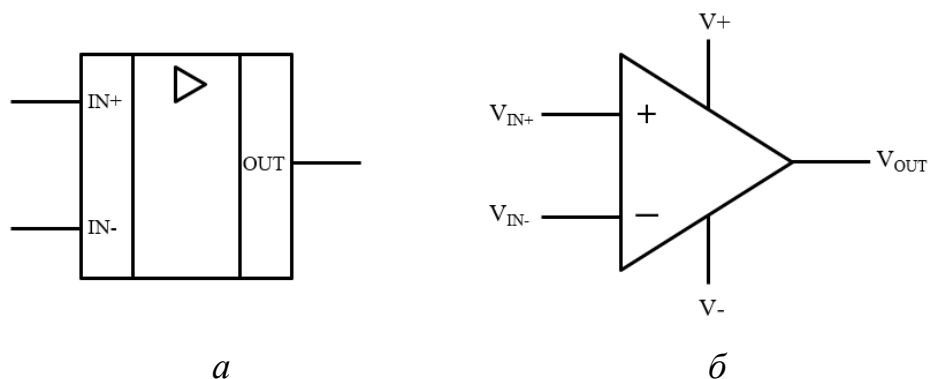


Рис. 6.1. Символ операційного підсилювача: *а* – український стандарт ($IN+$ – неінвертуючий вхід, $IN-$ – інвертуючий вхід, OUT – вихід), *б* – західний стандарт (V_{IN+} – напруга на неінвертуючому вході (+), V_{IN-} – напруга на інвертуючому вході (-), V_{out} – напруга на виході, $V+$ – плюс джерела живлення (також може позначатися як V_{S+} , $+V_S$, V_{DD} , V_{CC} або V_{CC+}), $V-$ – мінус джерела живлення (також може позначатися як V_{S-} , $-V_S$, V_{SS} , V_{EE} або V_{CC-}))

ОП зазвичай живляться від симетричних джерел, що забезпечують однакові за величиною додатну (V_{S+} , $+U_{ж}$) і від'ємну (V_{S-} , $-U_{ж}$) напруги щодо нульового проводу («землі»). Для більшості сучасних ОП напругу живлення можна змінювати у досить широких межах від ± 3 В до ± 18 В.

ОП випускаються як у вигляді окремих інтегральних мікросхем, так і можуть бути складовою частиною багатьох спеціалізованих аналогових мікросхем.

ОП випускаються у найрізноманітніших корпусах. Приклади деяких типів корпусів представлені на рис. 6.2.

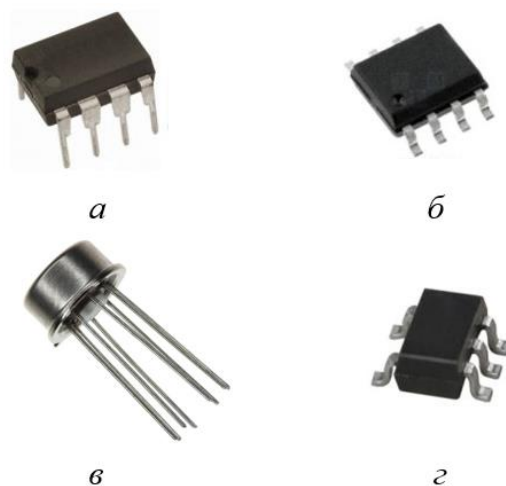


Рис. 6.2. Корпуси мікросхем ОП: *а* – DIP-8, *б* – SOIC-8, *в* – TO-99, *г* – SOT23-5

В технічній документації (datasheet) для кожної мікросхеми завжди є схема з конфігурацією виводів ОП та поясненням їх призначення. На рис. 6.3 показаний приклад схеми конфігурації виводів для мікросхеми BA4558N в корпусі SIP, яка містить два ОП.

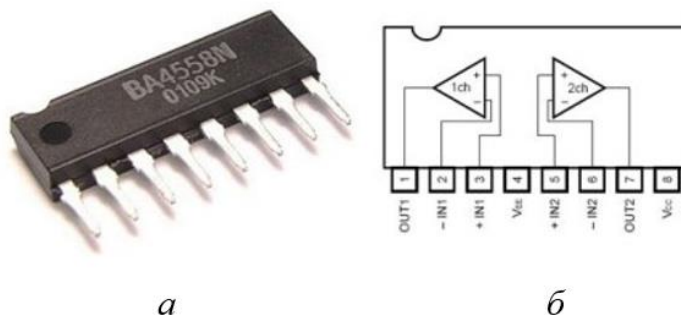


Рис. 6.3. ОП в корпусі SIP: *а* – зовнішній вигляд, *б* – діаграма, що показує конфігурацію виводів

2. Завдання на комп'ютерний практикум

1. Створити загальне УГП операційного підсилювача.
2. Створити ПМ для мікросхеми **AD642**, яка містить в одному корпусі два операційних підсилювачі. На рис. 6.4 показана конфігурація виводів мікросхеми **AD642** в корпусі TO-99 згідно її специфікації.

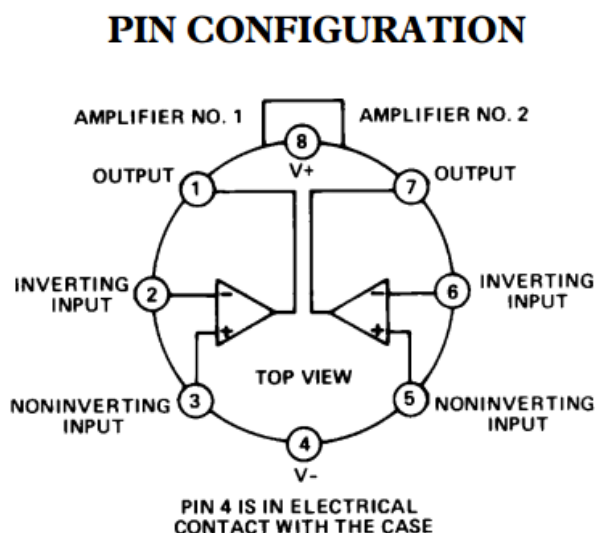


Рис. 6.4. Конфігурація виводів (pin configuration) мікросхеми AD642 у корпусі TO-99

3. Створити ПМ для мікросхеми **LMH6714MF** в корпусі SOT-23-5. На рис. 6.5 показана конфігурація виводів мікросхеми **LMH6714MF**.

Connection Diagrams

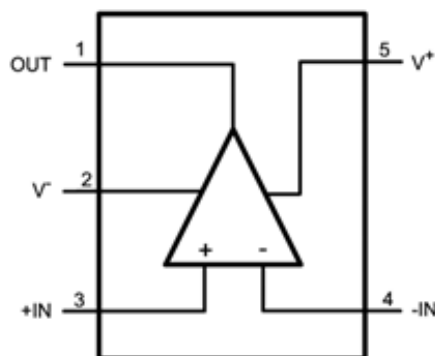


Рис. 6.5. Конфігурація виводів (connection diagrams) мікросхеми LMH6714MF у корпусі SOT-23-5

4. Об'єднати УГП і відповідні ПМ мікросхем **LMH6714MF** та **AD642** для створення завершених бібліотечних компонентних модулів.

3. Порядок виконання комп'ютерного практикуму

3.1. Створення УГП операційного підсилювача

Запустіть програму **Altium Designer**. Активуйте вкладку **SCH Library** і створіть новий компонент з іменем **AD642**.

У графічній області накресліть контур елемента за допомогою команди **Place** → **Line** у вигляді прямокутника розміром 15x20 мм (ширина-висота). Всередині прямокутника намалюйте дві вертикальні лінії на відстані 5 мм від кожного краю. Відкрийте властивості ліній та встановіть колір зі списку під номером 221.

Додайте виводи довжиною 5 мм за допомогою команди **Place** → **Pin**. Налаштуйте відображення імен виводів (у полі **Display Name** та **Designator** увімкніть мітки **Visible**). Розмістіть два виводи з лівого боку контуру ОП на відстані 10 мм один від одного та один вивід – по центру контуру справа (див. рис. 6.6). Для кожного виводу першого ОП мікросхеми **AD642** відкрийте вікно властивостей **Pin Properties** та змініть назву відповідно до табл. 6.1, яка створена на основі рис. 6.4. Вивід живлення 4 встановіть у будь-якій частині робочої області, в якій намальоване УГП. У властивостях цього виводу після встановлення відповідного позначення і назви увімкніть опцію **Hide**, а у полі **Connect To** напишіть **V-**. Крім

цього, у полі **Part Number** встановіть значення 0 (поле стане активним для зміни тільки після створення другого елементу мікросхеми).

Для назв виводів підберіть шрифт таким чином, щоб всі написи вмістились по ширині в додаткових полях УГП. Для цього у властивостях кожного виводу активуйте прапорці **Customize Position** і **Use local font setting**, після чого з'явиться можливість їх редагування.

Таблиця 6.1. Нумерація та назви виводів мікросхеми **AD642**

Designator	1	2	3	4	5	6	7	8
Pin Description	OUT	IN-	IN+	V-	IN+	IN-	OUT	V+

Намалюйте символ підсилювача, що має вигляд трикутника, у верхній частині основного поля УГП за допомогою команди **Place** → **Line**. Остаточний результат наведено на рис. 6.6.

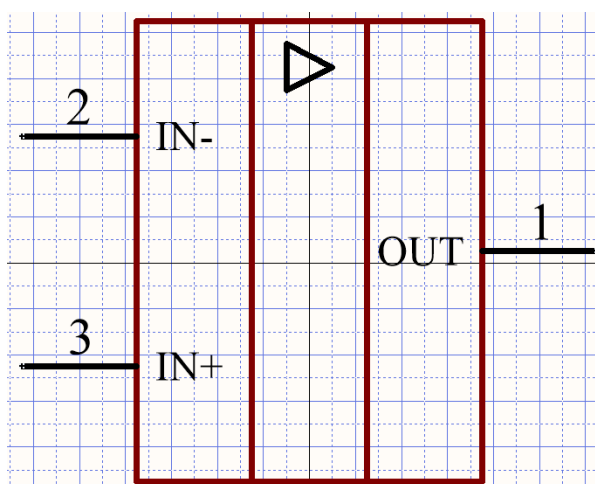


Рис. 6.6. УГП операційного підсилювача

Створіть другу частину (part) компонентного модуля **AD642** за допомогою команди **Tools** → **New Part**. Скопіюйте намальоване для першого елементу УГП у графічну область другого, змінивши нумерацію виводів згідно табл. 6.1 та рис. 6.4. Водночас у властивостях виводу 8 необхідно активувати опцію **Hide**, у полі **Connect To** ввести текст **V+**, у полі **Part Number** – встановити значення 0.

Відкрийте вікно властивостей компоненту **AD642** та у полі **Default Designator** введіть текст **DA?**, у полі **Description** – текст **OpAmp**, у полі **Default Comment** встановіть символ «*». Для полів **Default Comment** та **Default Designator** активуйте властивість **Visible**.

Створіть новий компонент з іменем **LMH6714MF**. Скопіюйте УГП ОП, яке було намальоване для компоненту **AD642**, змінивши водночас нумерацію виводів згідно рис. 6.5. Зауважте, що в корпусі мікросхеми **LMH6714MF** міститься тільки один підсилювач, тому використовуватиметься тільки один елемент, в якому повинні бути встановлені виводи як для додатного живлення, так і від'ємного. Вікно властивостей цього компоненту має бути так само оформлене, як і вікно властивостей **AD642**.

3.2. Створення посадкового місця для мікросхеми в корпусі TO-99

Перейдіть у Редактор посадкових місць. Використовуючи геометричні характеристики, що наведені на рис. 6.7, створіть⁶⁵ ПМ для мікросхеми **AD642**.

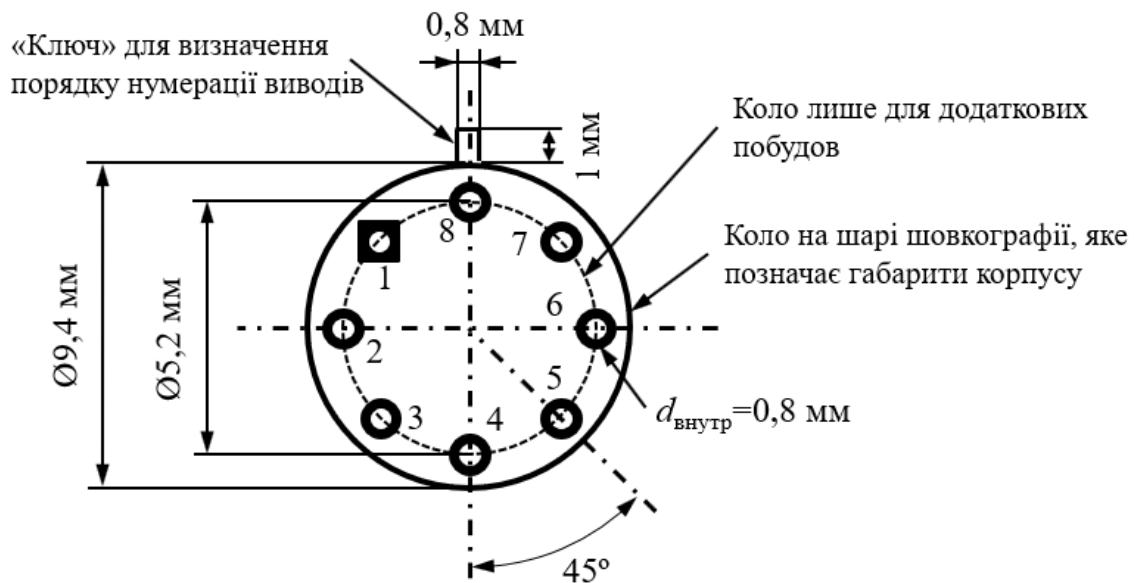


Рис. 6.7. Геометричні характеристики посадкового місця мікросхеми AD642

Встановіть основним кроком сітки 0,1 мм. За допомогою команди **Place** → **Arc** (товщина лінії 0,2 мм) намалюйте коло радіусом 2,6 мм⁶⁶, яке

⁶⁵ Наразі у **Altium Designer 17.1** неможливо створити посадкове місце для мікросхеми у корпусі TO-99 за допомогою **Component Wizard** або **IPC Compliant Footprint Wizard**.

⁶⁶ Насправді виводи безпосередньо у корпусі TO-99 розташовані по колу радіусом 2,54 мм (відповідно діаметр становить 5,08 мм). Проте КМ посадкового місця ми будемо розмішувати по колу радіусом 2,6 мм. Так робити допустимо (ми навіть могли б встановити КМ і на більшому радіусі), оскільки виводи із корпусу TO-99 є гнучкими, а, отже, під час паяння ми можемо їх вигнути таким чином, щоб вони без проблем увійшли в отвори КМ, що розташовані на більшому

використовуватиметься для позиціонування⁶⁷ КМ (оскільки це коло виконує лише допоміжну роль і після усіх побудов буде видалене, то наразі немає ніякого значення, на якому поточному шарі воно буде намальоване). Згідно геометричних характеристик, показаних на рис. 6.7, за допомогою команди **Place** → **Pad** встановіть 8 КМ, внутрішній діаметр яких 0,8 мм, а зовнішній – 1,4 мм. Після встановлення КМ допоміжне коло необхідно видалити. Пронумеруйте КМ так, як показано на рис. 6.7.

Оберіть поточним шар **Top Overlay**. Намалюйте контур ПМ у вигляді кола радіусом 4,7 мм, який ззовні охоплює усі КМ. Також навпроти восьмого КМ за допомогою команди **Place** → **Line** (товщина лінії 0,1 мм) намалюйте «ключ» у вигляді прямокутника розмірами 1x0,8 мм. Остаточний результат показано на рис. 6.8.

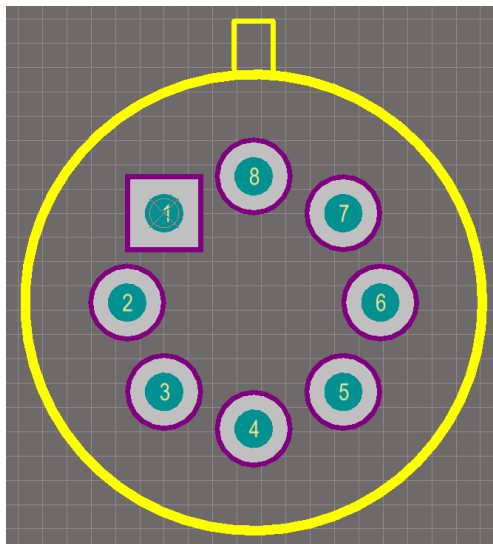


Рис. 6.8. Посадкове місце мікросхеми в корпусі ТО-99

Збережіть у власну бібліотеку розроблене ПМ під іменем **TO99**.

Створіть просту тривимірну модель корпусу ТО-99, використовуючи стандартні інструменти **Altium Designer**. Геометричні характеристики корпусу ТО-99 наведені на рис. 6.9.

радіусі. Однак не варто і занадто збільшувати радіус уявного кола, за яким будуть встановлюватись КМ.

⁶⁷ Рекомендується під час створення цього ПМ виконувати усі побудови відносно центру робочого поля. Це виявиться зручним під час розміщення КМ. Однак після виконання усіх побудов необхідно виділити усі елементи посадкового місця і змістити їх так, щоб центр робочого поля співпав із центром першого КМ.

OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

TO-99

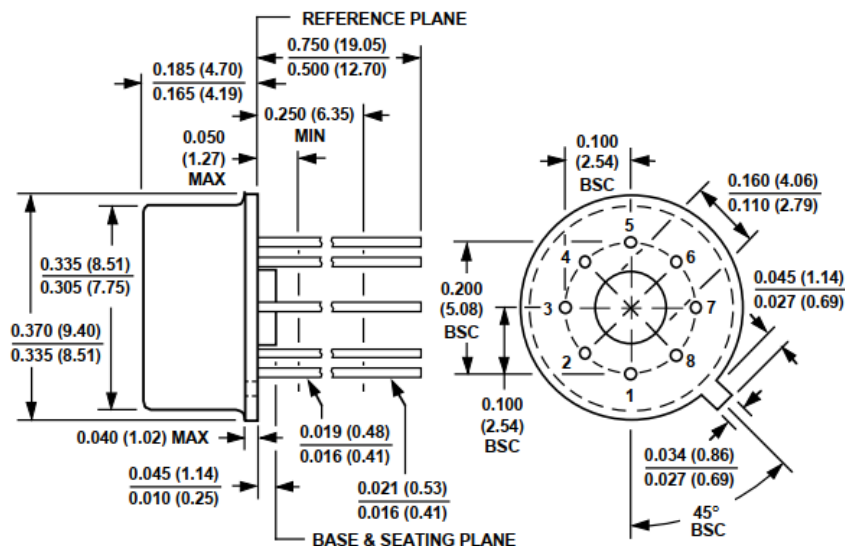


Рис. 6.9. Геометричні характеристики корпусу TO-99 згідно технічної документації на мікросхему AD642

Зверніть увагу, що розміри на рис. 6.9 вказані як у дюймах, так і міліметрах. Ми використовуватимемо значення у міліметрах⁶⁸.

3.3. Створення посадкового місця для мікросхеми в корпусі SOT-23-5

Використовуючи набуті під час виконання попередніх комп'ютерних практикумів знання розробіть ПМ для мікросхеми в корпусі SOT-23-5 згідно рекомендованих геометричних розмірів, що наведені на рис. 6.10.

На рис. 6.10 (як і на рис. 6.9) розміри вказані як у дюймах, так і міліметрах, тому прочитайте пояснювальний текст, щоб зрозуміти, як вказуються ті чи інші величини. Також врахуйте, що деякі величини містять другу і навіть третю цифру після коми. Зазвичай така точність для величин, які наведені у міліметрах, є надмірною під час проєктування ПМ, тому дозволяється виконувати округлення до першої цифри після коми (наприклад, якщо відстань між центрами рядів виводів вказана як 2,59 мм, то її можна округлити до 2,6 мм).

⁶⁸ Тому уважно прочитайте пояснювальний текст на рис. 6.9, щоб зрозуміти, де позначення величин у дюймах, а де – у міліметрах.

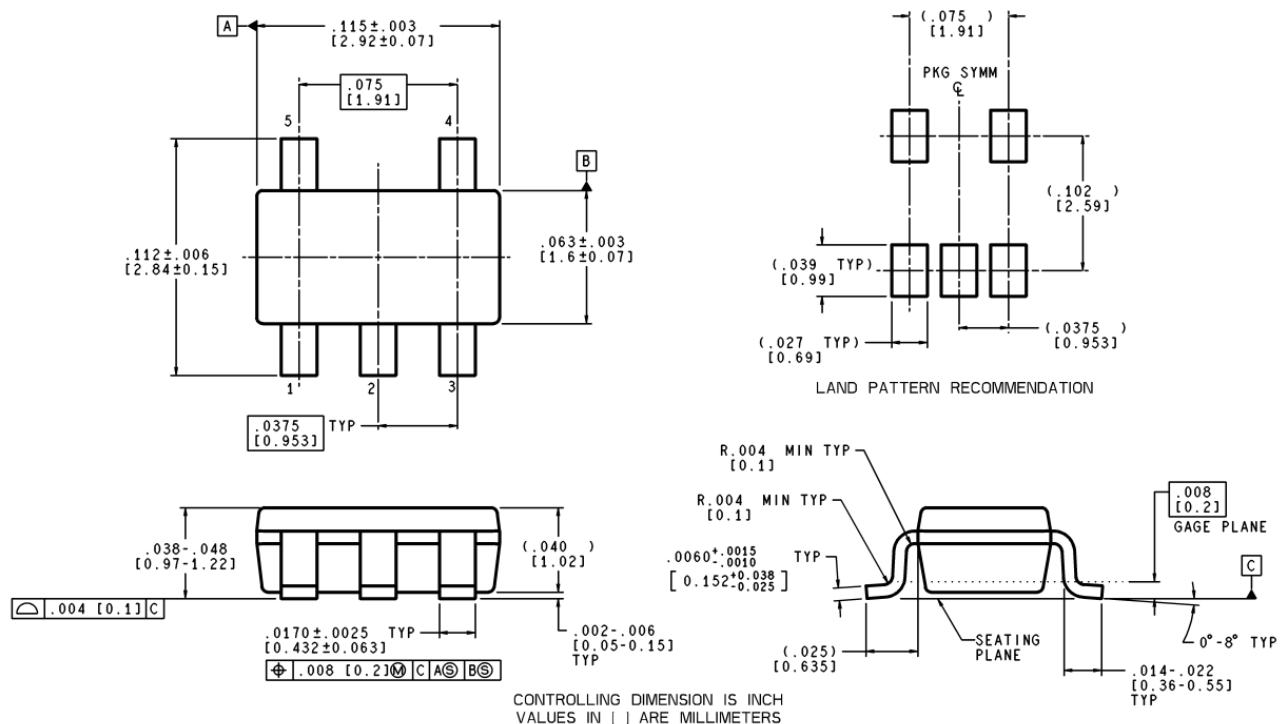


Рис. 6.10. Розміри мікросхеми та рекомендовані геометричні розміри посадкового місця SOT-23-5 для мікросхеми LMH6714MF згідно технічної документації

Якщо ПМ буде створюватись вручну, то використовуйте одразу розміри, які позначені на рис. 6.10 як **LAND PATTERN RECOMMENDATION**. Якщо буде використовуватись *IPC Compliant Footprint Wizard*⁶⁹, то вказуйте геометричні розміри корпусу мікросхеми із урахуванням максимальних та мінімальних значень (наприклад, якщо довжина корпусу вказана як $2,92 \pm 0,07$, то мінімальне значення буде 2,85, а максимальне 2,99).

Збережіть розроблене ПМ у власну бібліотеку під іменем **SOT23-5**.

Якщо це ПМ було намальоване вручну, то створіть просту тривимірну модель цього корпусу, використовуючи стандартні інструменти **Altium Designer**. Якщо ж використовувався *IPC Compliant Footprint Wizard* і була активована опція *Generate STEP Model Preview*, то тривимірна модель буде створена автоматично.

3.4. Створення компонентних модулів мікросхем LMH6714MF та AD642

Об'єднайте УГП кожного підсилювача із відповідним йому ПМ. Під час додавання моделі ПМ перевірте відповідність нумерації виводів на УГП номерам КМ.

⁶⁹ Зверніть увагу, що за допомогою майстра *Component Wizard* корпус SOT-23-5 створити неможливо.

Також виконайте перевірку розроблених компонентних модулів на наявність помилок.

4. Питання для самоконтролю

1. Поясніть, що таке операційний підсилювач?
2. Що таке зворотній зв'язок в операційних підсилювачах і яких типів він буває?
3. З якою метою в підсилювачах використовують зворотній зв'язок?
4. Які схеми можна побудувати на основі операційних підсилювачів?
5. Опишіть, в яких корпусах випускаються операційні підсилювачі?
6. Опишіть процес розробки посадкового місця операційного підсилювача AD642.

Комп'ютерний практикум № 7.

РОЗРОБКА КОМПОНЕНТНИХ МОДУЛІВ З'ЄДНУВАЧІВ ТА РОЗ'ЄМІВ

Мета роботи: ознайомитись з різними типами з'єднувачів та роз'ємів; створити УГП та посадкові місця роз'ємів PLS, BNC, DS-210 та клеми KF301-2P.

1. Теоретичні відомості

Для підключення зовнішніх джерел живлення до плати та для з'єднання різних плат між собою з метою передачі сигналів використовуються різноманітні з'єднувачі.

Електричний з'єднувач (connector) – це електротехнічний пристрій, призначений для механічного з'єднування та роз'єднування електричних провідників. Зазвичай з'єднувач складається з двох частин (або роз'ємів⁷⁰), які містять *контактні елементи*, завдяки яким, власне, і створюється електричний контакт.

Згідно міжнародного стандарту IEC 60050-151:2001 «International Electrotechnical Vocabulary (IEV) - Part 151: Electrical and magnetic devices» контактні елементи поділяються на *гніздові* та *штирові*. Гніздовий контакт (socket contact, female contact) призначений створювати електричний контакт на його внутрішній поверхні для з'єднання із зовнішньою поверхнею іншого контактного елемента. Штировий контакт (pin contact, male contact) призначений створювати електричний контакт на його зовнішній поверхні для з'єднання з внутрішньою поверхнею іншого контактного елемента. В побуті штировий контакт часто називають «тато», а гніздовий – «мама».

Самі ж частини з'єднувача називаються вилкою (або штекером, від нім. stecker – вилка) та розеткою. Відповідно до того ж стандарту IEC 60050-151:2001 вилка (plug) – це з'єднувач, що приєднаний до кабелю, а розетка (socket) – з'єднувач, що приєднаний до апарату або конструктивного елемента, або до чогось подібного. Водночас у примітці зазначається, що контактні елементи розетки можуть бути

⁷⁰ Роз'єм (або рознім) – це окрема складова з'єднувача, проте роз'ємом можуть називати і сам з'єднувач.

гніздовими контактами, штировими контактами або тими й іншими⁷¹. Якщо говорити в загальному, то розеткою вважають менш рухому частину з'єднувача, яка монтується на платі пристрою, колодці чи панелі, а вилкою – більш рухому, яка монтується на проводі, кабелі або іншій пристрої, який має більше можливостей переміщуватись порівняно до головного пристрою.

Роз'ємні з'єднувачі на принципових схемах зазвичай першим символом у позиційному позначенні містять літеру **X** (після якої можуть бути ще деякі літери, які уточнюють тип роз'єму). Також можуть бути ще такі варіанти позначень: розетка позначається літерою **J** (від англ. jack), а вилка – літерою **P** (від англ. plug).

Існує величезна кількість видів роз'ємів, що обумовлюється їх різним функціональним призначенням. Їх конструкція визначається числом одночасно комутованих електричних ланцюгів, частотами сигналів тощо. Розглянемо деякі види роз'ємів і з'єднувачів.

Для того, щоб подати напругу від джерела живлення на плату, використовують різноманітні роз'єми живлення, які можуть випускатись у варіантах, що призначені для монтажу на плату, з кріпленням під гвинт на корпус тощо. На рис. 7.1а представлений досить поширений роз'єм живлення DS-210⁷² (розетка із штировим контактним елементом), який припаюється до плати.



Рис. 7.1. Частина з'єднувача для підключення живлення до плати: *а* – роз'єм живлення для монтажу на плату DS-210, *б* – штекер для монтажу на проводі

Зверніть увагу, що роз'єм DS-210 є гібридним роз'ємом, оскільки він містить як штировий контактний елемент («тато»), так і гніздовий («мама»). Оскільки штекер такого

⁷¹ Часто розетку називають «гніздом», маючи на увазі, що вона містить гніздові контакти. Так склалось історично, оскільки розетки спочатку містили тільки гніздові контакти. Тому для розетки із гніздовими контактами використання терміну «гніздо» не буде чимось неправильним. Однак згідно останніх стандартів поняття «розетка» дещо ширше, ніж «гніздо».

⁷² Подібний роз'єм випускають різні виробники, у яких він може називатись не DS-210, а інакше, наприклад, [DC-005](#), [PJ-002AH](#) тощо. Водночас можуть незначним чином відрізнятись геометричні характеристики корпусу та виводів, тоді як конструктивно вони подібні.

з'єднувача (див. рис. 7.1б) входить в розетку DS-210, то її також називають гніздом живлення (незважаючи на те, що вона має штировий контактний елемент).

Для з'єднання різноманітних пристроїв за допомогою окремих проводів використовують клемні колодки (наприклад, таким чином можна підключати джерело живлення до плати). Клемна колодка являє собою механічний затискач (кліть) в пластиковому корпусі або без нього. Залежно від типу затиску, клема може бути оснащена гвинтами, металевою пластиною або пружиною для фіксації провідника. Провідник встановлюється в затискну кліть та фіксується всередині. На рис. 7.2 представлена клема KF301-2P, яка монтується на плату та використовується для підключення двох проводів. Особливістю клем цього виду є те, що збоку вони мають спеціальні пази, за допомогою яких можна міцно з'єднати кілька окремих клем у одну довгу клемну колодку.



Рис. 7.2. Клема KF301-2P

Ще одним способом з'єднання двох пристроїв за допомогою окремих проводів є використання роз'єму PLS⁷³, що являє собою N-контактну дворядну штирову лінійку, призначену для припаювання до плати (рис. 7.3). Кількість контактів N варіюється від 1 до 40. За необхідності лінійка PLS може бути розділена на відрізки меншого розміру, причому це дуже легко зробити вручну шляхом відламування.

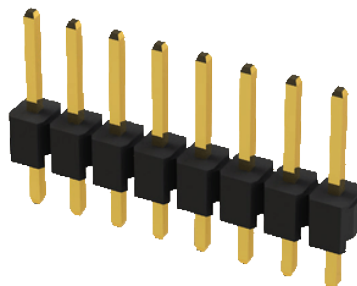


Рис. 7.3. Роз'єм типу PLS

⁷³ Дуже часто цей вид роз'ємів використовується під час прототипування та макетування. Особливо популярний у проєктах Arduino, зокрема, застосовується під час підключення різноманітних плат розширення.

Для передачі височастотних сигналів і з'єднання різноманітних радіоелектронних пристроїв (генераторів, осцилографів тощо) використовуються електричні роз'єми з байонетною фіксацією BNC (аббревіатура від англ. Bayonet Neill-Concelman⁷⁴). Зазвичай роз'єми BNC застосовуються для підключення коаксіального кабелю з хвильовим опором 50 Ом (або 75 Ом) і діаметром до 8 мм.

Кабельному роз'єму-штекеру відповідає приладовий роз'єм-гніздо, що встановлюється або у корпусі приладу та фіксується гайкою, або монтується на плату з додатковою можливістю фіксації гайкою до корпусу приладу з метою підвищення міцності кріплення (рис. 7.4).



Рис. 7.4. Роз'єм BNC (гніздо) для монтажу на плату

Зауваження!

Деякі електронні компоненти, що монтуються на ДП (наприклад, роз'єм BNC), потребують для кращої фіксації, окрім монтажних (металізованих), ще й додаткові отвори. Це можуть бути кріпильні неметалізовані отвори або металізовані отвори, що зазвичай з'єднуються із загальною шиною землі на платі (в такі отвори запаюються виводи, що з'єднані з екраном самого електронного компонента). Іноді в якості кріпильних використовуються металізовані отвори, які електрично ні з чим не з'єднані.

2. Завдання на комп'ютерний практикум

Створити бібліотечні компонентні модулі роз'ємів **PLS-2**, **PLS-4**, **DS-210**, **BNC** та клеми **KF301-2P**. Під час розробки ПМ використовувати геометричні характеристики, що наведені на рис. 7.5-7.8.

⁷⁴ Названий на честь розробників: Пола Нейла (англ. Paul Neill) з лабораторії «Bell Labs» і Карла Концельмана (англ. Carl Concelman) з фірми «Amphenol».

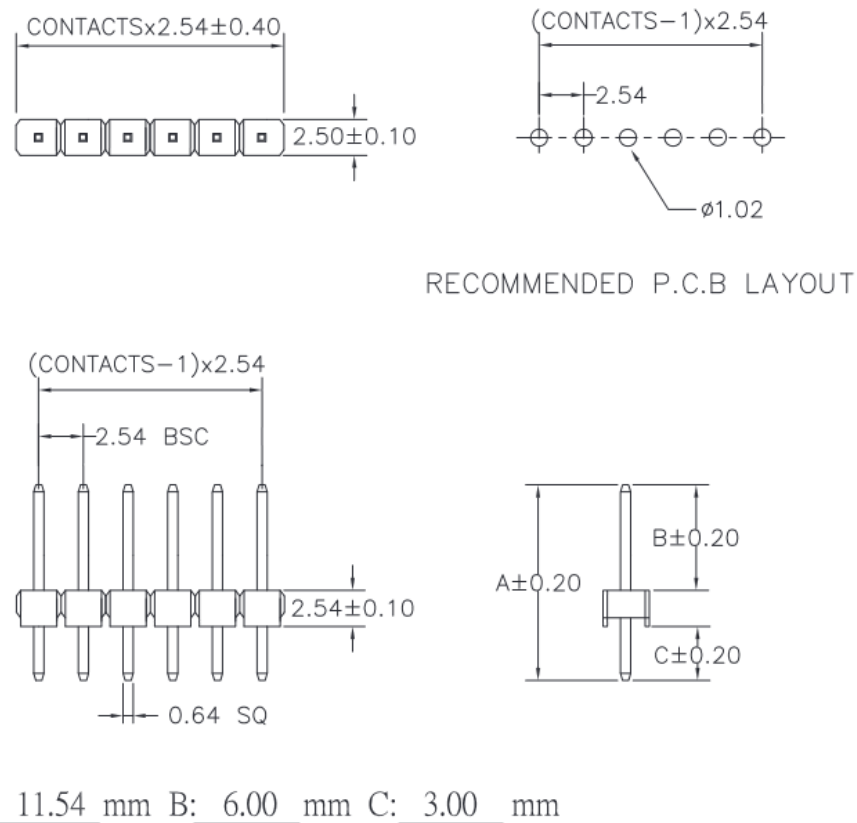


Рис. 7.5. Геометричні розміри та розміри посадкового місця роз'єму типу PLS

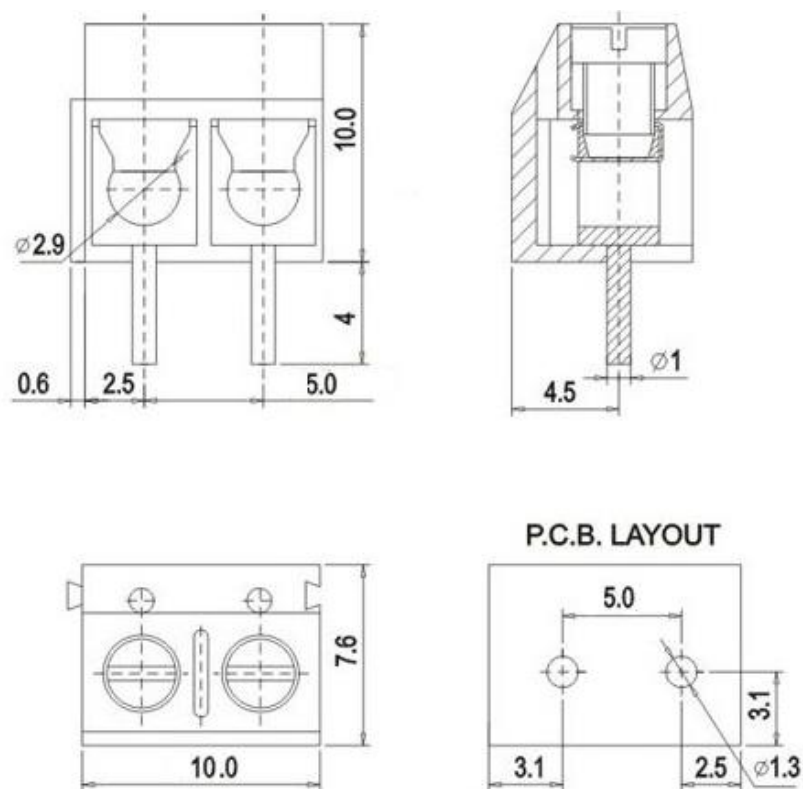


Рис. 7.6. Геометричні розміри та розміри посадкового місця клеми KF301-2P

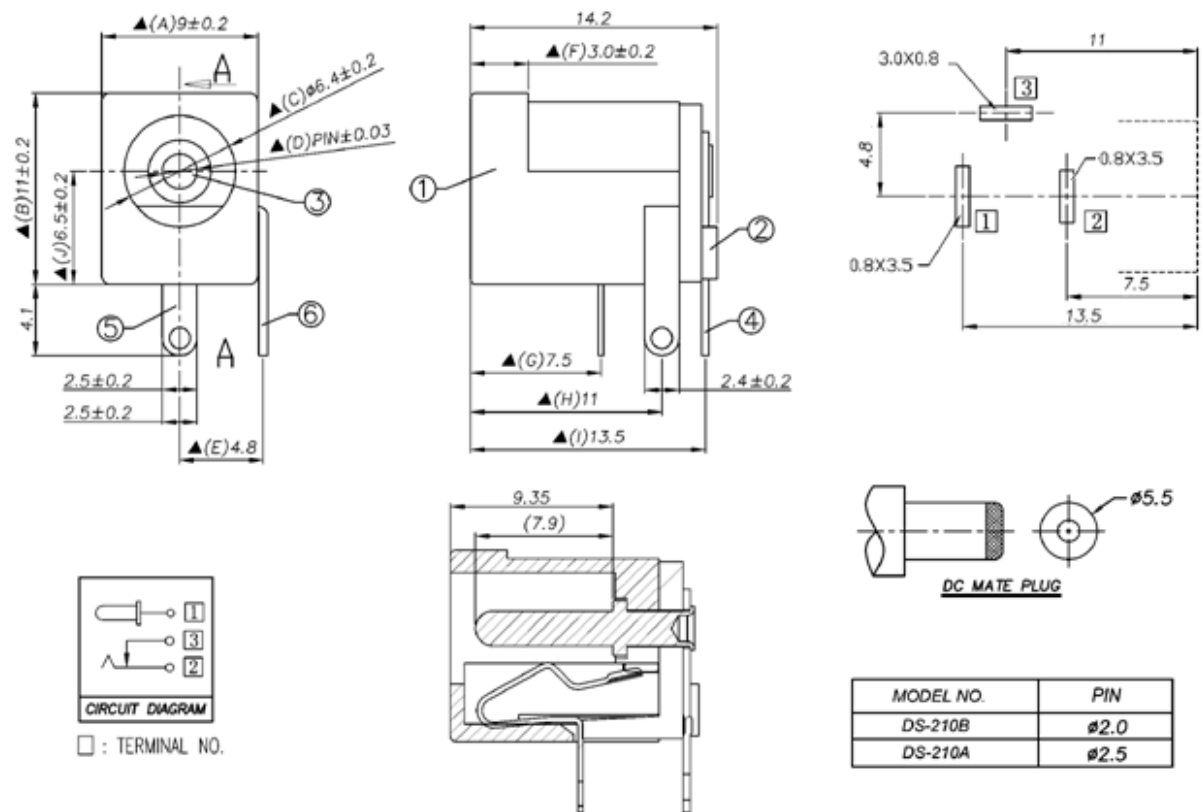


Рис. 7.7. Геометричні розміри та розміри посадкового місця роз'єму DS-210

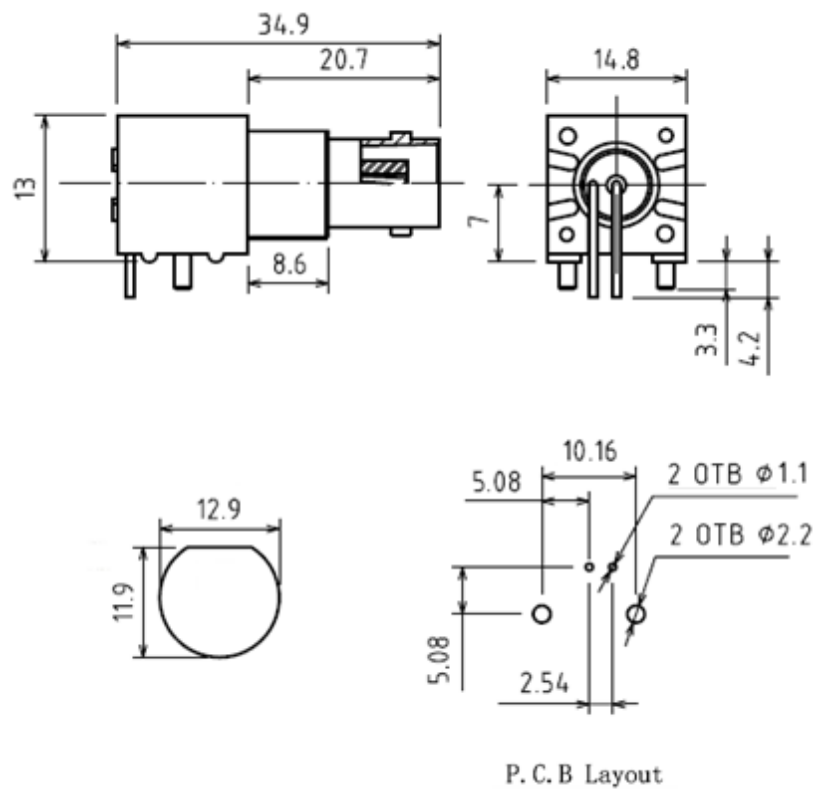


Рис. 7.8. Геометричні розміри та розміри посадкового місця роз'єму BNC для монтажу на плату

3. Порядок виконання комп'ютерного практикуму

3.1. Створення компонентних модулів роз'ємів PLS-N

Активуйте вкладку *SCH Library* і створіть новий компонент з іменем **PLS-2**. Намалюйте УГП роз'єму **PLS-2** згідно геометричних характеристик, що наведені на рис. 7.9. Довжину виводів встановіть 5 мм, нумерація виводів: верхній – 1, нижній – 2. У вікні властивостей кожного виводу в полі *Display Name* увімкніть мітку *Visible*, а в полі *Designator* – вимкніть.

Над УГП цього роз'єму намалюйте лінію зі стрілкою, причому сама стрілка складається з двох ліній довжиною 2,8 мм із кутом 90° між ними⁷⁵.

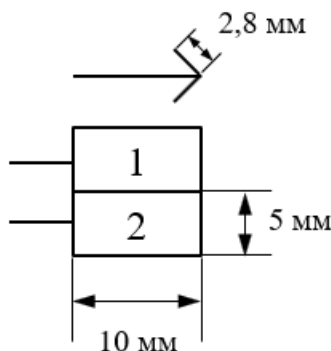


Рис. 7.9. УГП роз'єму PLS-2

Відкрийте вікно властивостей створеного компоненту **PLS-2** і в полі *Default Designator* введіть текст **XP?**, а в полі *Description* – текст **Connector**. Для полів *Default Designator* та *Default Comment* активуйте властивість *Visible*.

ПМ для цього виду коннекторів можна або використати із вбудованої бібліотеки (рис. 7.10a), або накреслити власне, яке буде більш детально відображати їх геометричну форму (рис. 7.10б).

В **Altium Designer** коннектори PLS знаходяться у бібліотеці **Miscellaneous Connectors.IntLib**, але мають назву **Header** (їх ПМ називаються **HDR**). Щоб переглянути, як цей тип з'єднувачів представлений в **Altium Designer**, оберіть **Miscellaneous Connectors.IntLib** зі списку бібліотек на панелі *Libraries*. Для швидкого пошуку двоконтактного роз'єму введіть «*2» у поле фільтра на панелі *Libraries*. Виберіть **Header 2** зі списку елементів і ознайомтесь з його УГП та ПМ.

⁷⁵ Для можливості малювання лінії під кутом необхідно послідовно натискати клавішу <Пробіл> під час її розміщення.

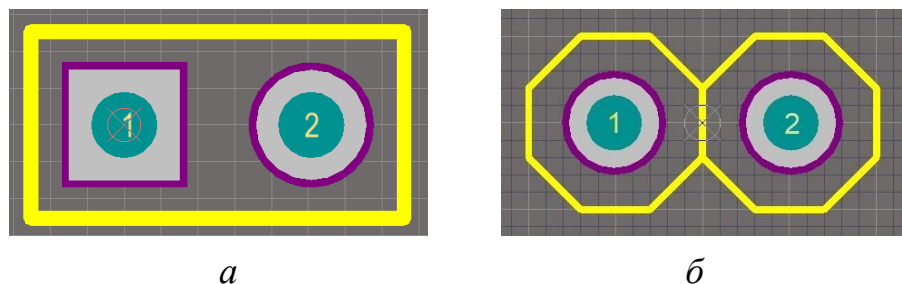


Рис. 7.10. Посадкове місце PLS коннекторів: *а* – посадкове місце із вбудованої бібліотеки **Miscellaneous Connectors**, *б* – один із варіантів власного посадкового місця

Розглянемо, яким чином можна скопіювати компонент (або тільки ПМ) із існуючої інтегрованої бібліотеки у власну. Цей процес не є простим. Це пов'язано з тим, що перш ніж щось скопіювати із інтегрованої бібліотеки, її спочатку необхідно розпакувати. Для цього необхідно виконати команду **File** → **Open**, після чого знайти бібліотеку **Miscellaneous Connectors.IntLib**⁷⁶ і натиснути кнопку **Open**. Внаслідок цих дій з'явиться вікно, що представлено на рис. 7.11.

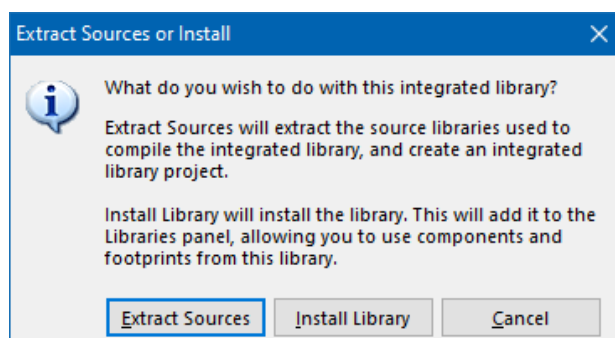
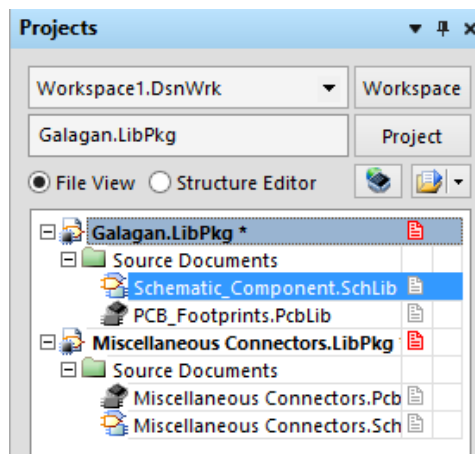


Рис. 7.11. Вікно **Extract Sources or Install**

Необхідно натиснути кнопку **Extract Sources**, що призведе до розпаковування зазначеної бібліотеки та її відображення у панелі **Projects** (рис. 7.12).

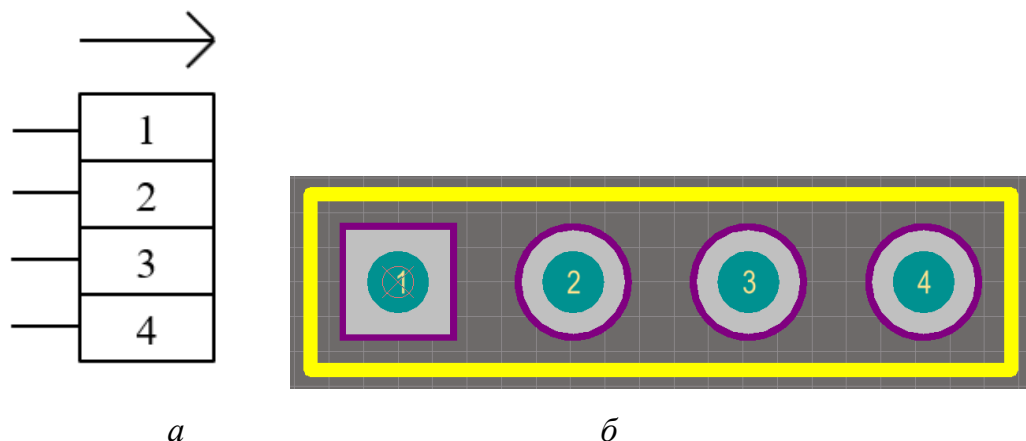
Після того, як бібліотека була розпакована, її можна редагувати: додавати нові компоненти, копіювати весь компонент із розпакованої інтегрованої бібліотеки у власну або копіювати тільки ПМ у власну бібліотеку з розширенням ***.PcbLib**. Для цього необхідно спочатку перейти в бібліотеку, з якої буде копіюватись елемент, знайти потрібний елемент (або ПМ), скопіювати (наприклад, використовуючи комбінацію клавіш **<Ctrl + C>**), після чого перейти у власну бібліотеку і вставити елемент (можна скористатись комбінацією клавіш **<Ctrl + V>**).

⁷⁶ Ця бібліотека за замовчуванням розташована на диску C за наступним шляхом: Користувачі \ Загальні \ Загальні документи \ Altium \ AD17 \ Library.

Рис. 7.12. Розпакована бібліотека **Miscellaneous Connectors.IntLib**

У нашому випадку потрібно скопіювати тільки ПМ **HDR1X2** із бібліотеки **Miscellaneous Connectors.PcbLib**. Після того, як ви скопіювали зазначене ПМ у власну бібліотеку, змініть його назву⁷⁷ на **PLS-X2** та додайте до власного розробленого УГП цього роз'єму, оскільки воно відрізняється від того УГП, що використовується в **Altium Designer**.

Аналогічно описаному вище створіть компонентний модуль роз'єму **PLS-4**, що має 4 контакти (рис. 7.13). ПМ для цього роз'єму можна також взяти із бібліотеки **Miscellaneous Connectors.PcbLib**. Змініть назву скопійованого ПМ на **PLS-X4**, після чого підключіть до УГП.

Рис. 7.13. Роз'єм PLS-4: *a* – УГП, *б* – ПМ

⁷⁷ Взагалі, змінювати назву не обов'язково. Проте інколи за наявності у різних інтегрованих бібліотеках однойменних ПМ можуть виникати помилки, пов'язані із тим, що замість одного ПМ програма під час автоматичного перенесення проєктних даних буде використовувати інше. Якщо ви не вносите ніяких змін у скопійоване ПМ, то проблем не буде. Але ж ви можете дещо відредагувати скопійоване ПМ і воно буде відрізнятися від оригіналу, тому вас зовсім не влаштуватиме, що програма використовуватиме вихідне ПМ.

3.2. Створення компонентного модуля роз'єму BNC

У бібліотеці символів створіть новий компонент з іменем **BNC-7044**. Намалюйте УГП роз'єму BNC, використовуючи геометричні характеристики, що наведені на рис. 7.14.

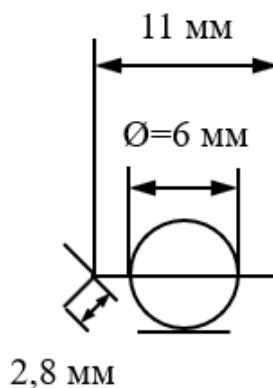


Рис. 7.14. Геометричні розміри УГП роз'єму BNC

Створіть коло за допомогою команди *Place* → *Full Circle*. Використовуючи команду *Place* → *Pin* додайте вивід елементу (із номером 1) довжиною 11 мм і розмістіть його горизонтально вздовж центру кола. Зверніть увагу, що кінець виводу, до якого приєднується електричний ланцюг, повинен знаходитись праворуч від УГП. У властивостях виводу у полі *Display Name* та *Designator* вимкніть мітки *Visible*.

У будь-якій частині робочої області розмістіть другий вивід, у властивостях якого встановіть номер 2, увімкніть опцію *Hide*, а в полі *Connect To* напишіть **GND**.

За допомогою команди *Place* → *Line* додайте дві короткі лінії (≈2,8 мм), що виходять з лівого кінця першого виводу і мають кут між собою 90°. Також додайте горизонтальну лінію довжиною 6 мм, що дотична до нижньої частини кола.

Відкрийте вікно властивостей компоненту **BNC-7044** і в полі *Default Designator* введіть текст **XW?**, а в полі *Description* – текст **Connector**. Для полів *Default Comment* та *Default Designator* активуйте властивість *Visible*.

Знайдіть у вже розархівованій бібліотеці **Miscellaneous Connectors** ПМ **BNC_RA CON**, скопіюйте його у власну бібліотеку, перейменуйте на **BNC_CONNECTOR** та підключіть до створеного УГП роз'єму BNC. На рис. 7.15 показано, як має виглядати робоча область компонентного модуля **BNC-7044** з підключеним ПМ.

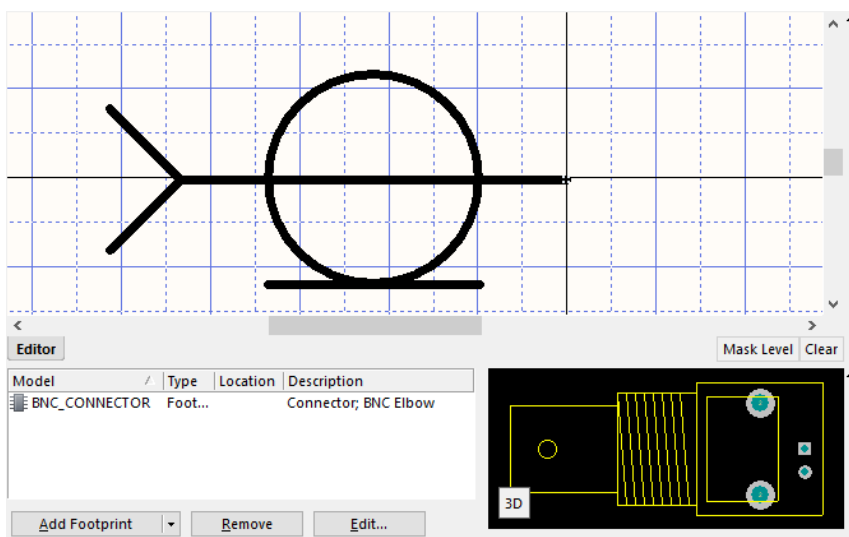


Рис. 7.15. Компонентний модуль роз'єму BNC-7044

3.3. Створення компонентного модуля клеми KF301-2P

Створіть новий компонент з ім'ям **KF301-2P**. Намалюйте УГП клеми (рис. 7.16), використовуючи наступні геометричні характеристики: діаметр кіл – 2 мм, довжина виводів – 5 мм, відстань між виводами – 5 мм. Верхній вивід повинен мати номер 1, нижній – 2. Зверніть увагу, що кінець, до якого на принциповій схемі приєднуватиметься електричний ланцюг (позначається ледь помітним хрестиком), має розташовуватись праворуч. У налаштуваннях виводів відключіть властивість **Visible** для поля **Display Name**, а для поля **Designator** – активуйте цю властивість. Зверніть увагу, що для того, щоб примітив **Designator** був зміщений ближче до початку виводу (як показано на рис. 7.16), то необхідно у вікні властивостей виводу в полі **Designator Position and Font** активувати опцію **Customize Position**, після чого правильно підібрати відступ (**Margin**).

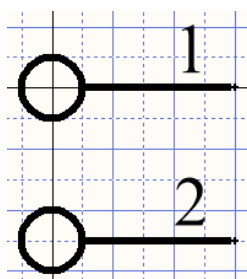


Рис. 7.16. УГП клеми

Відкрийте вікно властивостей компоненту **KF301-2P** і в полі **Default Designator** введіть текст **XT?**, а в полі **Description** – текст **Screw Type Terminal**

Block (2 Pole Counts). Для полів *Default Comment* та *Default Designator* активуйте властивість *Visible*.

Далі перейдіть у бібліотеку посадкових місць та створіть ПМ клеми, використовуючи геометричні розміри, що наведені на рис. 7.6. Товщина ліній контуру – 0,2 мм. КМ із прямокутним контуром повинен мати номер 1, а із круглим – 2. Кінцевий результат представлено на рис. 7.17.

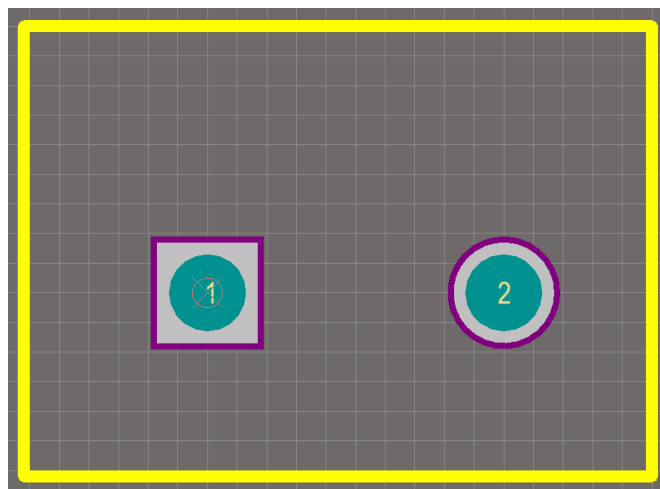


Рис. 7.17. Посадкове місце клеми KF301-2P

Збережіть ПМ під іменем **KF301-X2** та додайте його до УГП клеми.

3.4. Створення компонентного модуля роз'єму живлення DS-210

У теоретичній частині до цього комп'ютерного практикуму пояснювалось, що роз'єм DS-210 є гібридним роз'ємом, тобто містить і штирові контакти, і гніздові. На рис. 7.18 представлені кілька варіантів реалізації УГП такого роз'єму.

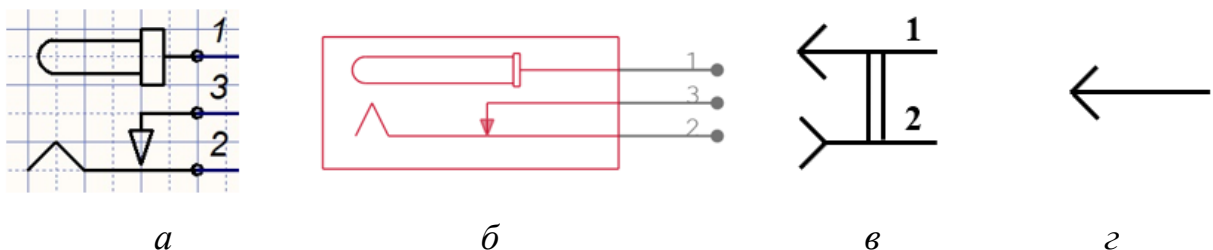


Рис. 7.18. Варіанти УГП роз'єму DS-210: а, б – символ створено у різних САПР так, як це вказано в технічній документації на цей роз'єм, в – символ, який враховує наявність штирового та гніздового контакту в роз'ємі, г – символ, який вказує, що сигнальний контакт є штировим (водночас другий гніздовий контакт прихований на УГП і за замовчуванням з'єднаний із землею)

Окрім представлених на рис. 7.18 варіантів УГП можуть бути створені й інші. Вибір того чи іншого варіанту УГП залежить від багатьох факторів, зокрема від того, на які стандарти орієнтується розробник. У цьому комп'ютерному практикумі ми розробимо УГП на основі рис. 7.18в.

Створіть новий компонент з іменем **DS-210**. Намалюйте УГП двоконтактного гібридного роз'єму (рис. 7.19). Верхній вивід повинен мати довжину 7 мм, а нижній – 5 мм, дві лінії, що виходять з лівого кінця кожного виводу, – 2,8 мм (при цьому кут між ними 90°). Відстань між выводами по висоті – 5 мм. Дві з'єднувальні вертикальні лінії розмістіть на відстані 2 мм та 3 мм від правого краю выводів. Для верхнього виводу задайте номер 1, для нижнього – 2. У налаштуваннях выводів відключіть властивість *Visible* для поля *Display Name*, а для поля *Designator* – активуйте цю властивість.

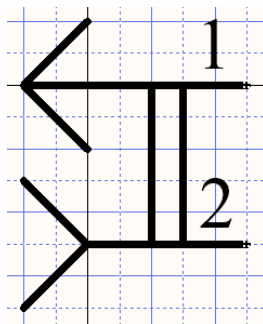


Рис. 7.19. УГП двоконтактного роз'єму DS-210

Відкрийте вікно властивостей компоненту **DS-210** і в полі *Default Designator* введіть текст **XS?**, а в полі *Description* – текст **Power Connector**. Для полів *Default Comment* та *Default Designator* активуйте властивість *Visible*.

Далі відкрийте бібліотеку посадкових місць та створіть ПМ роз'єму **DS-210**, використовуючи геометричні розміри, що наведені на рис. 7.7.

Зверніть увагу, що виводи цього типу корпусу мають прямокутний перетин, причому один з розмірів значно перевищує інший. У цьому випадку недоцільно створювати круглий отвір на платі (діаметр якого був би дещо більшим за розмір довшої сторони прямокутного перетину виводу), оскільки він виявився би занадто великим та незручним для припаювання. Тому для подібного типу выводів створюються КМ спеціальної форми.

Для першого та другого КМ посадкового місця гнізда живлення **DS-210** можна використовувати наступні параметри (рис. 7.20):

- в графічній області *Size and Shape* встановити: форма (*Shape*) – кругла (*Round*), ширина (*X-Size*) – 4,5 мм; висота (*Y-Size*) – 2 мм;

- в графічній області **Hole Information** встановити діаметр отвору (**Hole Size**) – 0,8 мм, активувати опцію **Slot**, після чого стане доступним для зміни параметр **Length**, який встановити рівним 3,5 мм.

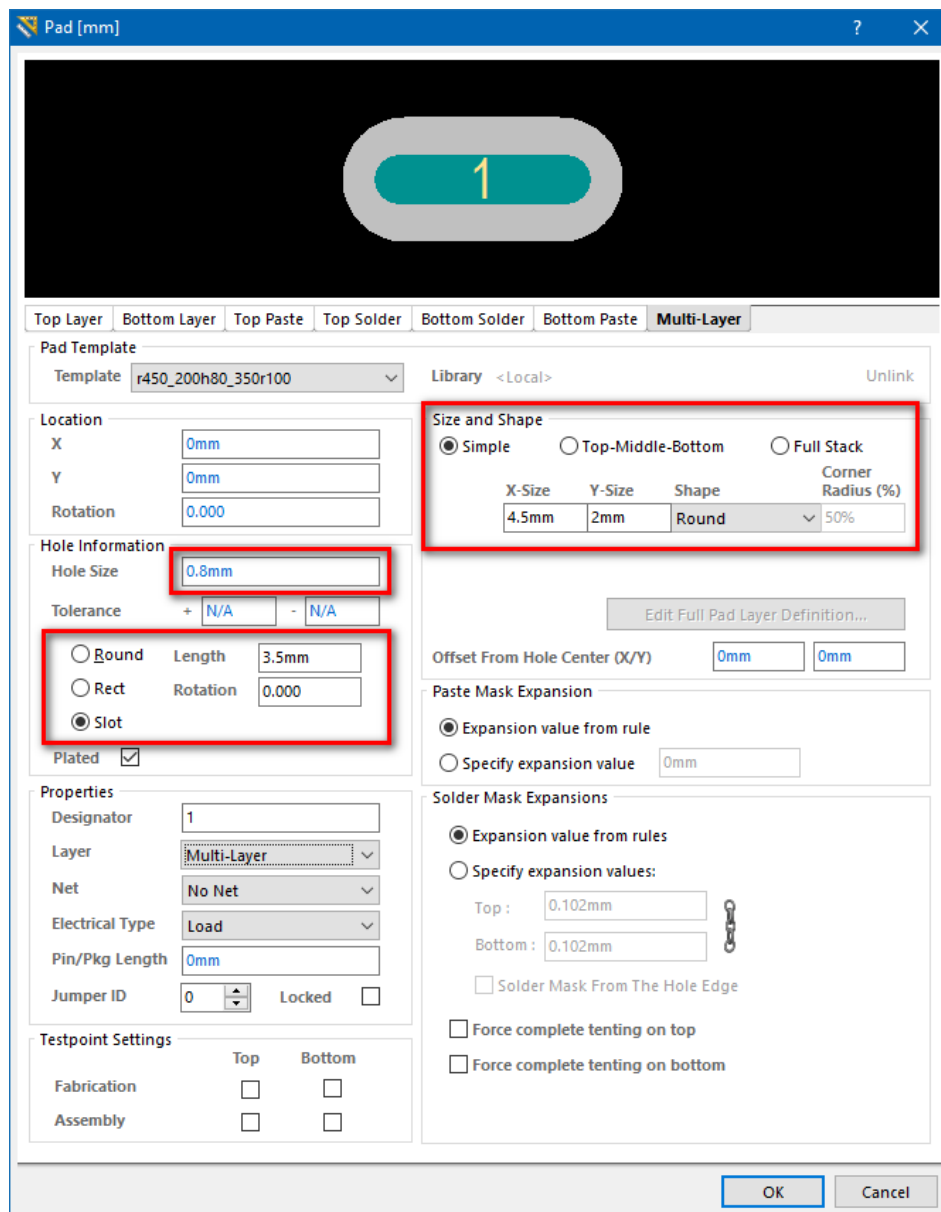


Рис. 7.20. Налаштування КМ посадкового місця роз'єму DS-210

Встановлення зазначених вище параметрів призведе до створення КМ, що розташований горизонтально. Для того, щоб розмістити КМ вертикально, необхідно повернути його на 90° (для цього виділіть потрібний елемент, натисніть і утримуйте на ньому ЛКМ та одночасно з цим натисніть клавішу <Пробіл>).

Врахуйте, що розмір третього КМ дещо відрізняється від перших двох. Для нього можна встановити наступні параметри: форма (**Shape**) – кругла (**Round**), ширина (**X-Size**) – 4 мм; висота (**Y-Size**) – 2 мм, діаметр отвору (**Hole Size**) – 0,8 мм, параметр **Length** із активованою опцією **Slot** – 3 мм.

Важливо!

У полі *Hole Information* кожного КМ не забудьте активувати опцію *Plated*. Ця опція визначає, чи буде отвір металізованим всередині. Здебільшого отвори, в які вставляються виводи електронних компонентів, мають бути металізованими всередині (за двостороннього трасування плати металізація отвору дозволяє електрично з'єднати протилежні шари⁷⁸). Однак металізація не використовується у випадку створення монтажних отворів, основним призначенням яких є кріплення плати до корпусу приладу.

Контур ПМ намалюйте на шарі *Top Overlay* із урахуванням максимальних габаритних розмірів роз'єму **DS-210**, що наведені рис. 7.7. Товщина ліній контуру – 0,2 мм. Кінцевий результат представлений на рис. 7.21.

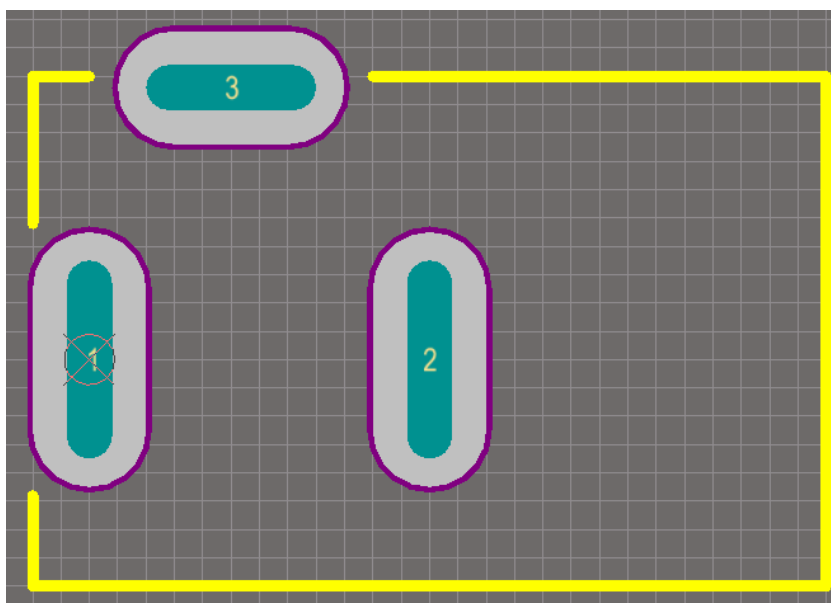


Рис. 7.21. Посадкове місце роз'єму DS-210

Зверніть увагу, що лінії контуру не повинні перетинати КМ⁷⁹ (у цьому місці вони розриваються). Збережіть ПМ під іменем **DS210**.

Об'єднайте УГП роз'єму **DS-210** та створене ПМ.

⁷⁸ Хоча здається, що функцію електричного з'єднання двох шарів плати через отвір може виконати сам вивід компонента, який буде припаяний на одній із сторін плати, а виходитиме із іншої, проте для надійності ліпше використовувати металізацію отворів.

⁷⁹ Чому саме так краще зробити, стане зрозуміло після виконання наступних комп'ютерних практикумів.

Зауваження!

На УГП роз'єму DS-210 зображено два виводи, але фізично сам компонент має 3 виводи (тому модель ПМ містить саме три КМ). Однак третій вивід використовується для зручної і надійної фіксації роз'єму на платі, до нього не повинен підключатись ніякий електричний ланцюг. Якщо залишити номер цього КМ як 3, то під час об'єднання УГП і ПМ буде встановлена відповідність номерів виводів електронного компоненту номерам КМ тільки для першого і другого виводу. Тоді під час трасування ДП до третього КМ не буде підключено ніякого ланцюга. З іншого боку, згідно datasheet на роз'єм DS-210, третій КМ на платі може бути з'єднаний із другим (який підключається до загального проводу). Тому третьому КМ можна привласнити номер 2 (тобто із номером 2 будуть два КМ). У такому випадку під час трасування ДП третій КМ буде з'єднаний із землею, як і другий.

3.5. Компіляція бібліотеки

Насамкінець перекомпілюйте власну бібліотеку, щоб нові компоненти відображались у вашій бібліотеці на панелі *Libraries*. Якщо під час компіляції виникнуть помилки, проаналізуйте їх та виправіть.

4. Питання для самоконтролю

1. Поясніть, для чого призначені роз'єми і з'єднувачі?
2. Поясніть, якими бувають контактні елементи роз'ємів?
3. Поясніть, чим вилка відрізняється від розетки?
4. Наведіть та охарактеризуйте деякі відомі вам типи роз'ємів і з'єднувачів.
5. Поясніть, що таке багатоконтактні роз'єми?
6. Яким чином розробляються УГП багатоконтактних роз'ємів?
7. Опишіть процес розробки компонентного модуля роз'єму DS-210.
8. Опишіть процес розробки компонентного модуля роз'єму BNC.
9. Опишіть процес розробки компонентного модуля клеми KF301-2P.
10. Опишіть процес розробки компонентного модуля роз'єму PLS.
11. Яким чином створити форму КМ, яка відрізняється від круглої або прямокутної?
12. Чи можуть бути у посадковому місці два контактних майданчики із однаковими номерами? Поясніть.

Комп'ютерний практикум № 8.

РОЗРОБКА КОМПОНЕНТНОГО МОДУЛЯ ДРАЙВЕРА СИЛОВИХ ТРАНЗИСТОРІВ MD1822

***Мета роботи:** ознайомитись із принципами роботи та областями застосування драйверів транзисторів; створити умовне графічне позначення драйвера транзистора із використанням міжнародних стандартів; побудувати посадкове місце для мікросхеми MD1822.*

1. Теоретичні відомості

Драйвери польових транзисторів (або драйвери затвора) – це пристрої для управління потужними напівпровідниковими приладами у вихідних каскадах перетворювачів електричної енергії. Вони використовуються як проміжна ланка між двома частинами будь-якого імпульсного перетворювача електричної енергії: схемою керування (мікроконтролером, цифровим сигнальним процесором або ПЛІС) і силовою частиною (зазвичай потужними виконавчими елементами у вигляді силових ключів).

На сьогоднішній день в якості силових ключів в імпульсних перетворювачах можуть використовуватися біполярні транзистори з ізольованим затвором (IGBT) або потужні польові (MOSFET) транзистори. MOSFET- та IGBT-транзистори є базовими елементами сучасної силової електроніки і використовуються як елементи комутації великих струмів і напруг. Для того, щоб перемикає силові ключі у відкритий або закритий стан, необхідно забезпечити задану потужність, а також швидкість наростання та/або спадання фронту керуючого сигналу на затворі транзистора. Низьковольтні логічні керуючі сигнали від, наприклад, мікроконтролерів, не можуть впоратись із цією задачею. Саме тому виникає необхідність у використанні спеціальних пристроїв узгодження, якими є драйвери транзисторів (рис. 8.1).

Під час вибору драйвера транзистора необхідно враховувати наступне:

- драйвер має забезпечувати надійне відкривання та закривання обраного ключа;
- необхідно дотриматися вимог щодо адекватної тривалості переднього і заднього фронтів під час комутації;
- драйвер сам не повинен перевантажуватися, працюючи у схемі.

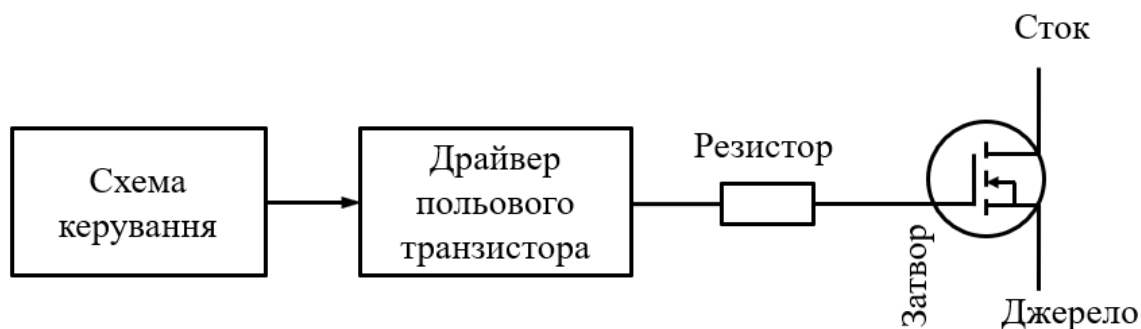


Рис. 8.1. Схематичне представлення взаємодії схеми керування і силового транзистора із використанням драйвера

Для кожного типу силових транзисторів необхідно використовувати спеціалізований тип драйверів, що враховує специфіку управління ними і дозволяє максимально повно використовувати їх позитивні властивості. Саме тому на ринку існує значна кількість мікросхем драйверів транзисторів. Однією із таких мікросхем є [MD1822](#).

MD1822 – це високошвидкісний чотириканальний драйвер, розроблений для керування високовольтними Р- і N-канальними MOSFET, які використовуються там, де потрібен високий вихідний струм для ємнісного навантаження (зокрема, може використовуватись для керування імпульсним збудженням ультразвукових датчиків). Високошвидкісний вхідний каскад MD1822 може працювати з логічним інтерфейсом від 1,8 до 5,0 В з оптимальним робочим діапазоном вхідного сигналу від 1,8 до 3,3 В. Вихідний каскад здатний досягати пікових струмів до $\pm 2,0$ А, залежно від використовуваної напруги живлення та наявного ємнісного навантаження.

2. Завдання на комп'ютерний практикум

1. Створити УГП драйвера, орієнтуючись на міжнародні стандарти⁸⁰ побудови УГП мікросхем, які прийняті в **Altium Designer**.
2. Створити ПМ для мікросхеми **MD1822**.
3. Об'єднати УГП драйвера та ПМ мікросхеми **MD1822** для створення завершеного бібліотечного компонентного модуля.

⁸⁰ Докладніше про стандарти можна прочитати у [Додатку Б](#).

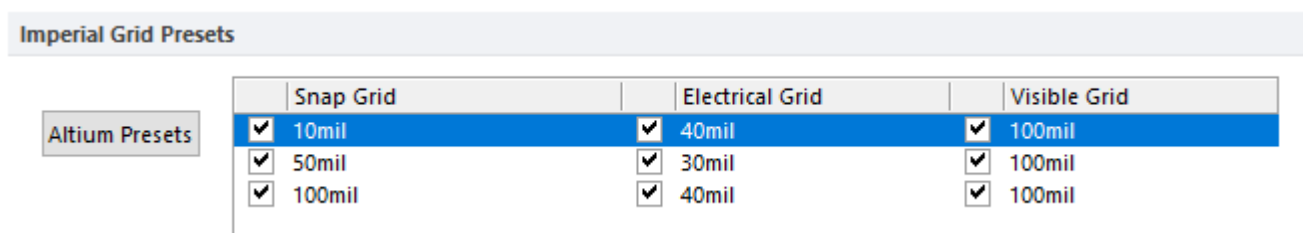
3. Порядок виконання комп'ютерного практикуму

3.1. Створення УГП драйвера

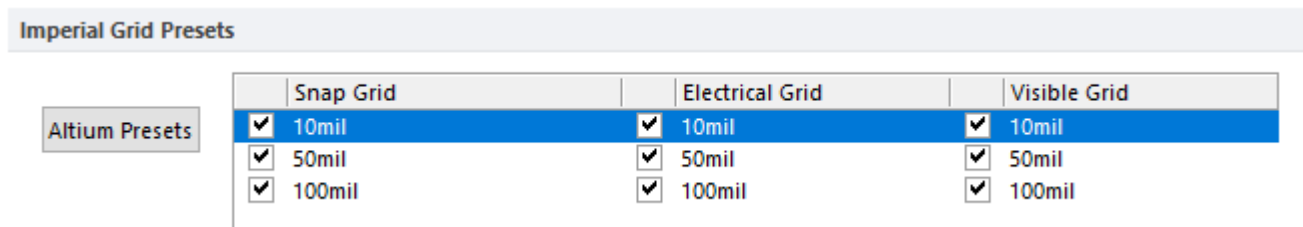
Активуйте вкладку *SCH Library* і створіть новий компонент з іменем **MD1822**.

Оскільки ми будемо створювати УГП, орієнтуючись на західні стандарти, то встановимо основними одиницями вимірювання дюймову систему. Для цього відкрийте меню *DXP* → *Preferences*. У вікні *Preferences* необхідно вибрати розділ *Schematic* → *Default Units* і встановити галочку *Use Imperial Unit System*.

Далі налаштуємо крок сітки. У тому ж розділі *Schematic* оберіть рядок *Grids*. В області *Imperial Grid Presets* можна задати необхідні кроки сітки. Якщо у вас в цьому розділі встановлені за замовчуванням такі параметри, як на рис. 8.2а, то змініть їх таким чином, щоб вони відповідали рис. 8.2б.



а



б

Рис. 8.2. Параметри кроку дюймової сітки:

а – за замовчування, б – які необхідно встановити

Після цього у Редакторі символів із меню *Tools* виконайте команду *Document Options*. У вікні *Schematic Library Options*, що відкриється, перейдіть на вкладку *Units* та встановіть галочку навпроти пункту *Use Imperial Unit System*. Насамкінець поверніться у вкладку *Library Editor Options* та у полі *Custom Size* встановіть значення геометричних розмірів робочої області 3000 mil за кожною координатою.

Після завершення усіх налаштувань для роботи із дюймовою системою встановіть поточним кроком сітки 100 mil.

У всіх попередніх комп'ютерних практикумах ми зазвичай спочатку створювали контур УГП, а вже потім додавали виводи. Наразі ми зробимо інакше – спочатку розмістимо виводи, а потім створимо контур УГП.

Перш за все у документації на мікросхему [MD1822](#) знаходимо таблицю із описом її виводів (рис. 8.3).

Pin Description

Pin #	Function	Description
1	INB	Logic input.
2	VDD	High side supply voltage.
3	VSS	Low side supply voltage. VSS is also connected to the IC substrate. It is required to connect to the most negative potential of voltage supplies.
4	INC	Logic input.
5	IND	
6	GND	Logic input ground reference.
7	VL	Supply voltage for N-channel output stage.
8	OUTC	Output drivers
9	OUTD	
10, 11	VH	Supply voltage for P-channel output stage.
12	OUTA	Output drivers
13	OUTB	
14	VL	Supply voltage for N-channel output stage.
15	PE	Power enable logic input. When PE is high, sets the input logic threshold. When PE is low, all outputs are at default state (see truth table) and IC in standby mode.
16	INA	Logic input.
Substrate		The IC substrate is internally connected to the thermal pad. Thermal pad and VSS must be connected externally.

Рис. 8.3. Опис виводів мікросхеми MD1822

Розмістіть послідовно один під одним (див. рис. 8.4a) виводи⁸¹ на робочому полі, використовуючи команду меню **Place** → **Pin**. Під час розміщення кожного виводу необхідно відкривати властивості (натискаючи клавішу <Tab>) і задавати нумерацію (поле **Designator**) та назву виводу (поле **Display Name**) відповідно до рис. 8.3. Переконайтесь, що довжина кожного виводу становить 200 mil.

Зверніть увагу, імена та номери виводів за замовчуванням відображаються шрифтом Times New Romans 10pt. Загалом, можна було б залишити усе, як є, проте інколи може виникнути необхідність зміни шрифту для групи виводів. Щоб не робити це для кожного виводу окремо, розглянемо, як у цьому випадку може допомогти інспектор властивостей.

⁸¹ Зверніть увагу, що мікросхема має 16 виводів згідно технічної документації. Проте ми додатково створимо 17 вивід із назвою TERMP, який потім буде асоційований із термopідкладкою.

Відкрийте вікно інспектора властивостей, виконавши команду **View** → **Workspace Panels** → **SCH** → **SCHLIB Inspector**. Після цього виділіть усі створені виводи і зверніть увагу, що дані у вікні інспектора оновляться для групи виділених контактів. Замінімо лише розмір шрифту⁸² назв та номерів виводів. Щоб це зробити, для параметрів **Pin Name Font Mode** та **Pin Designator Font Mode** встановіть режим **Custom** замість **Default**. Після цього з'явиться можливість редагувати зазначені параметри. Змініть тільки розмір шрифту на 9pt. Результат представлено на рис. 8.4б.

INB	1
VDD	2
VSS	3
INC	4
IND	5
GND	6
VL	7
OUTC	8
OUTD	9
VH	10
VH	11
OUTA	12
OUTB	13
VL	14
PE	15
INA	16
TERMP	17

а

INB	1
VDD	2
VSS	3
INC	4
IND	5
GND	6
VL	7
OUTC	8
OUTD	9
VH	10
VH	11
OUTA	12
OUTB	13
VL	14
PE	15
INA	16
TERMP	17

б

Рис. 8.4. Розташовані у робочій області виводи: а – із розміром шрифту за замовчуванням, б – із розміром шрифту 9pt

Тепер згрупуйте виводи за призначенням (вхідні, вихідні, живлення тощо). Один із варіантів групування представлено на рис. 8.5а. Відстань між рядами виводів становить 800 mil. Це не означає, що цей варіант найкращий, можна згрупувати виводи інакше – усе залежить від розробника і вимог. Також зверніть увагу, що цього разу ми не приховуємо виводи живлення та землі, як це було зроблено у попередніх комп'ютерних практикумах. У такому випадку на принциповій схемі необхідно буде безпосередньо створювати з'єднання між

⁸² Наприклад, за стандартом ISO для імен виводів встановлюється шрифт ISOCPEUR 14pt Italic, а для номерів виводів – ISOCPEUR 10 Italic. Проте зазвичай на персональних комп'ютерах ці шрифти не встановлені, їх потрібно додати в систему окремо, тільки після цього можна буде їх обрати зі списку шрифтів. А, можливо, за іншими стандартами необхідно використовувати інший шрифт. У будь-якому випадку, якщо його немає в системі, то його необхідно встановити окремо.

виводом джерела живлення та відповідним виводом мікросхеми. Однак один вивід все ж приховуємо у кінцевій реалізації УГП драйвера транзистора, оскільки він призначений для реалізації термopідкладки і його немає у технічній документації. Відкрийте властивості 17-го «віртуального» виводу **TERMP** та активуйте опцію **Hide**, а поле **Connect To** залиште пустим.

Після цього між рядами виводів встановіть контур УГП за допомогою команди **Place** → **Rectangle**. Доданий прямокутник буде перекривати назви виводів, оскільки він розміщений останнім і знаходитиметься на передньому плані. Для того, щоб деякий об'єкт перемістити на задній план, необхідно із меню **Edit** обрати команду **Move** → **Send To Back**, після чого клацнути ЛКМ на заданому об'єкті. Кінцевий результат представлено на рис. 8.5б.

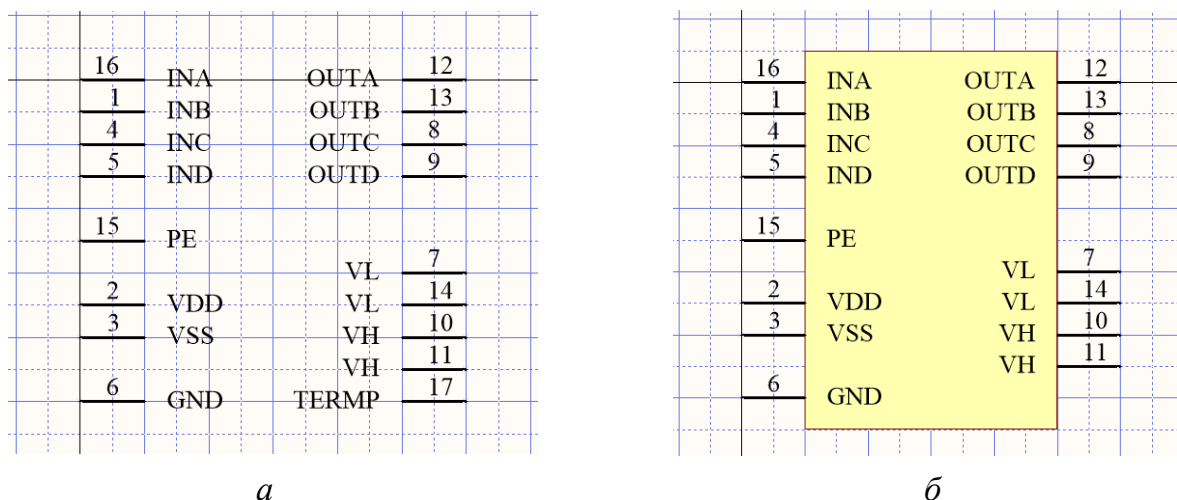


Рис. 8.5. Схематичне зображення драйвера MD1822: *а* – розташовані виводи, *б* – завершене УГП

Відкрийте вікно властивостей компоненту MD1822 і в полі **Default Designator** введіть текст **U?**, а в полі **Description** – текст **Transistor Driver**. Для полів **Default Comment** та **Default Designator** активуйте властивість **Visible**. У полі **Default Comment** із випадного списку оберіть рядок **'=Model'**.

Створіть опис мікросхеми, який може знадобитися згодом під час складання специфікації. Для цього у тому ж вікні властивостей у розділі **Parameters** для параметрів, які є в переліку, необхідно ввести конкретні значення.

Зауваження!

На першому комп'ютерному практикумі було вже створено перелік параметрів та пояснено їх призначення.

Для компоненту MD1822 введіть такі значення параметрів (рис. 8.6):

- **PackageType** – 16-Lead QFN;
- **URLtoDatasheet** –
<http://ww1.microchip.com/downloads/en/DeviceDoc/md1822.pdf>;
- **PurchaseURL** – <https://www.digikey.com/en/products/detail/microchip-technology/MD1822K6-G/4902884>;
- **Manufacturer** – Supertex;
- **Model** – MD1822K6-G;
- **Family** – Transistor driver.

Оскільки параметр *Value* потрібен для опису деякого фізичного параметру дискретних пасивних електронних компонентів, то у цьому випадку його можна видалити і не використовувати.

Library Component Properties

Properties

Default Designator: U? ☒ Visible ☐ Locked

Default Comment: =Model ☒ Visible

Part 1/1 ☒ Locked

Description: Transistor driver

Type: Standard

Library Link

Symbol Reference: MD1822

Graphical

Mode: Normal ☒ Lock Pins

☐ Show All Pins On Sheet (Even if Hidden)

☐ Local Colors

Parameters

Visible	Name	Value	Type
<input type="checkbox"/>	Family	Transistor driver	STRING
<input type="checkbox"/>	Manufacturer	Supertex	STRING
<input type="checkbox"/>	Model	MD1822K6-G	STRING
<input type="checkbox"/>	PackageType	16-Lead QFN	STRING
<input type="checkbox"/>	PurchaseURL	https://www.digikey.com/en/pro	STRING
<input type="checkbox"/>	URLtoDatasheet	http://ww1.microchip.com/down	STRING

Models

Name	Type	Description	Vault	Item Re...	Revisio...
------	------	-------------	-------	------------	------------

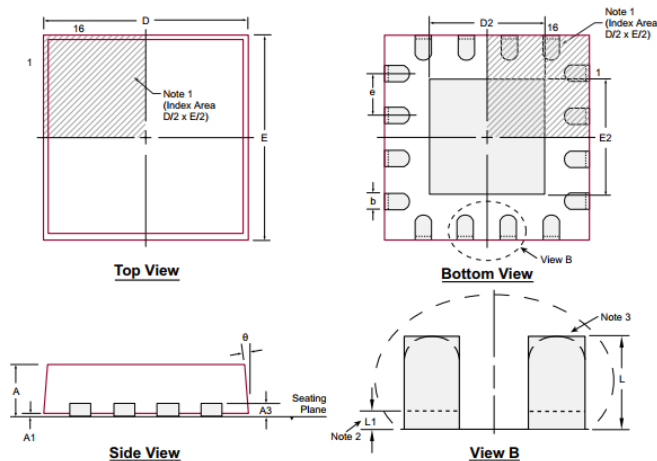
Buttons: Add..., Remove..., Edit..., Add as Rule...

Buttons: Edit Pins..., OK, Cancel

Рис. 8.6. Вікно параметрів компоненту MD1822

3.2. Створення посадкового місця для мікросхеми в корпусі QFN-16

Мікросхема **MD1822** випускається у корпусі QFN. Геометричні характеристики корпусу (рис. 8.7) беремо із технічної документації, яку можна знайти, зокрема, на офіційному сайті виробника цієї мікросхеми.

16-Lead QFN Package Outline (K6)**3.00x3.00mm body, 1.00mm height (max), 0.50mm pitch**

- Notes:
1. A Pin 1 Identifier must be located in the index area indicated. The Pin 1 Identifier can be: a molded mark/identifier; an embedded metal marker; or a printed indicator.
 2. Depending on the method of manufacturing, a maximum of 0.15mm pullback (L1) may be present.
 3. The inner tip of the lead may be either rounded or square.

Symbol	A	A1	A3	b	D	D2	E	E2	e	L	L1	θ
Dimension (mm)	MIN	0.80	0.00	0.20	0.18	2.85*	1.50	2.85*	1.50	0.20†	0.00	0°
	NOM	0.90	0.02	REF	0.25	3.00	1.65	3.00	1.65	0.30†	-	-
	MAX	1.00	0.05		0.30	3.15*	1.80	3.15*	1.80	0.45	0.15	14°

JEDEC Registration MO-220, Variation VEED-4, Issue K, June 2006.

* This dimension is not specified in the JEDEC drawing.

† This dimension differs from the JEDEC drawing.

Drawings not to scale.

Supertex Doc.#: DSPD-16QFNK63X3P050, Version A092909.

Рис. 8.7. Геометричні розміри корпусу мікросхеми MD1822

Перейдіть у Редактор посадкових місць. Запустіть інструмент *Component Wizard*, що автоматизує процес створення ПМ. Після того, як відкриється початкове вікно майстра, натисніть **Next**. У наступному вікні (рис. 8.8) оберіть тип корпусу *Quad Packs (QUAD)* та одиниці вимірювання *Metric (mm)*, після чого натисніть **Next**.

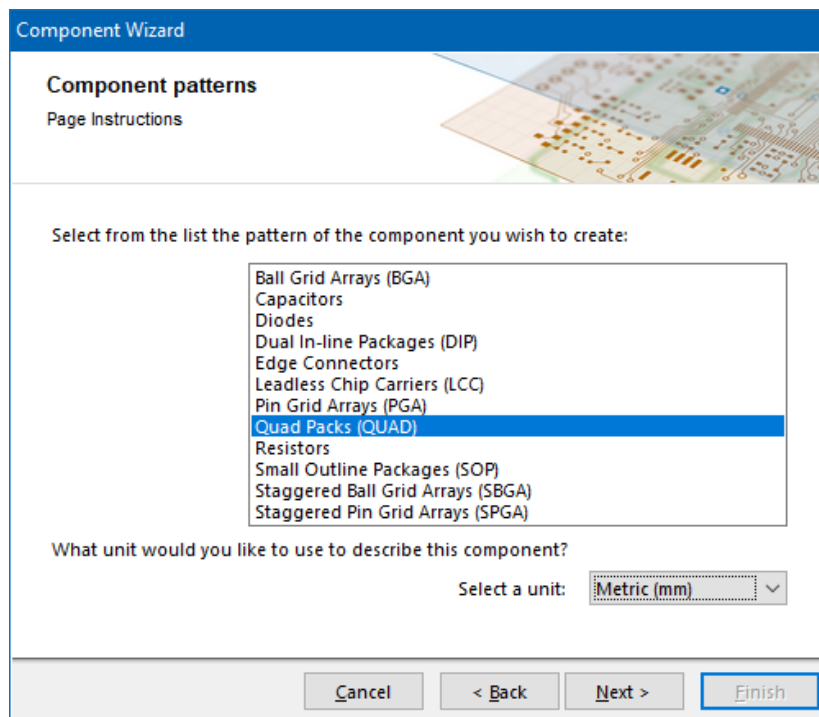


Рис. 8.8. Вікно вибору типу корпусу

У новому вікні (рис. 8.9) задаємо розміри КМ на основі даних із технічної документації. Ширину КМ обираємо як максимальне значення (рядок з написом MAX у таблиці на рис. 8.7) параметру **b**, таким чином ми нівелюємо допуски на ширину виводів мікросхеми. Довжину КМ варто робити більшою за максимальну довжину виводу **L**, що призведе до покращення припаювання паяльником та покращить контакт мікросхеми з платою. Після встановлення розмірів КМ натисніть **Next**.

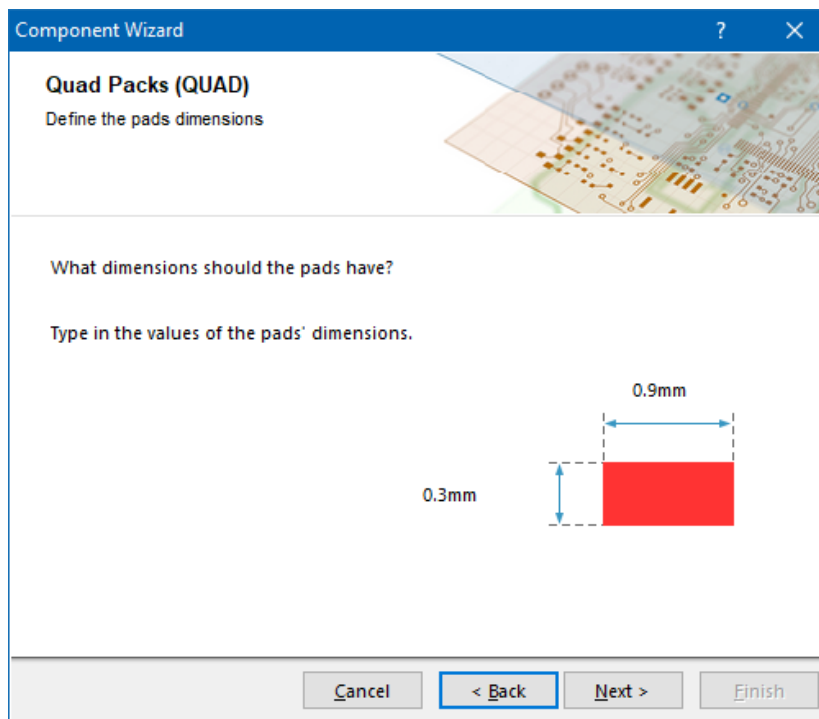


Рис. 8.9. Вікно налаштувань розмірів КМ

У наступному вікні майстер пропонує задати форму першого КМ у вигляді прямокутника із заокругленими кряями для визначення положення ключа мікросхеми, а усі інші КМ – зробити прямокутної форми. Але оскільки у нашому випадку КМ досить малі, тому різниці у формі КМ на платі ми не помітимо. Отже, робити різні форми КМ немає сенсу. Тому зробіть всі КМ однієї форми **Rectangular** та натисніть **Next**, щоб перейти до вікна вибору товщини лінії на шарі шовкографії. Тут товщину лінії залиште без змін і натисніть **Next**, щоб перейти у вікно налаштування взаємного розташування КМ (рис. 8.10). Щоби правильно ввести усі геометричні параметри у цьому вікні, необхідно знову звернутись до технічної документації. Відстань між КМ становить 0,5 мм (параметр **e**). Відстань між лініями центрів КМ, які розташовані вздовж перпендикулярних сторін мікросхеми, розраховується самостійно на основі геометричних розмірів корпусу та з урахуванням того, що довжина КМ на попередніх кроках була задана більшою за

максимальне значення. Встановіть значення, як на рис. 8.10, після чого натисніть кнопку **Next**.

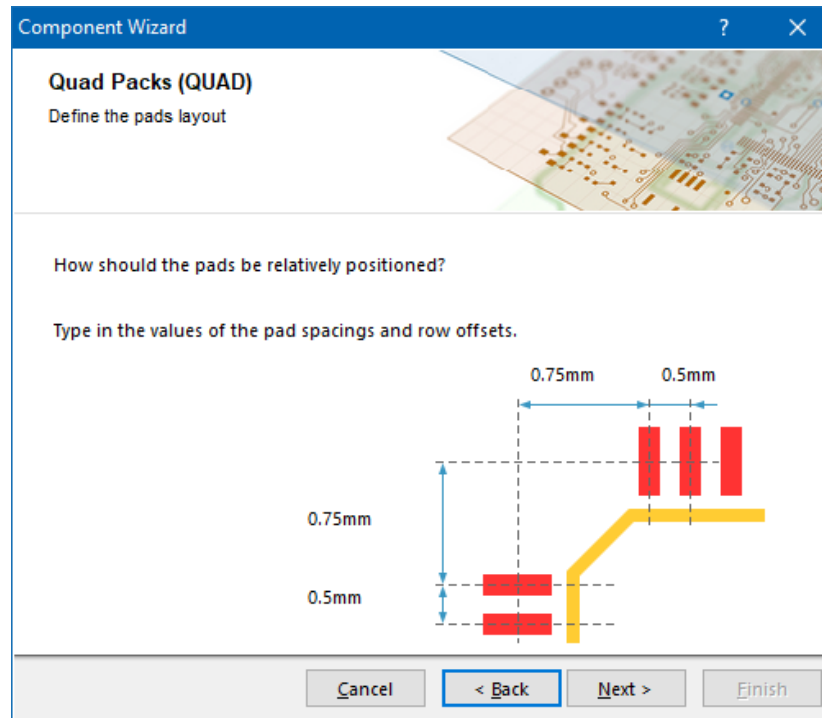


Рис. 8.10. Вікно налаштувань взаємного розташування КМ

У новому вікні майстер пропонує вибрати початкову точку відліку контактів та напрямок, в якому вони будуть нумеруватись. Встановіть відлік від ключа мікросхеми і напрямок нумерації проти годинникової стрілки (рис. 8.11), після чого натисніть **Next**.

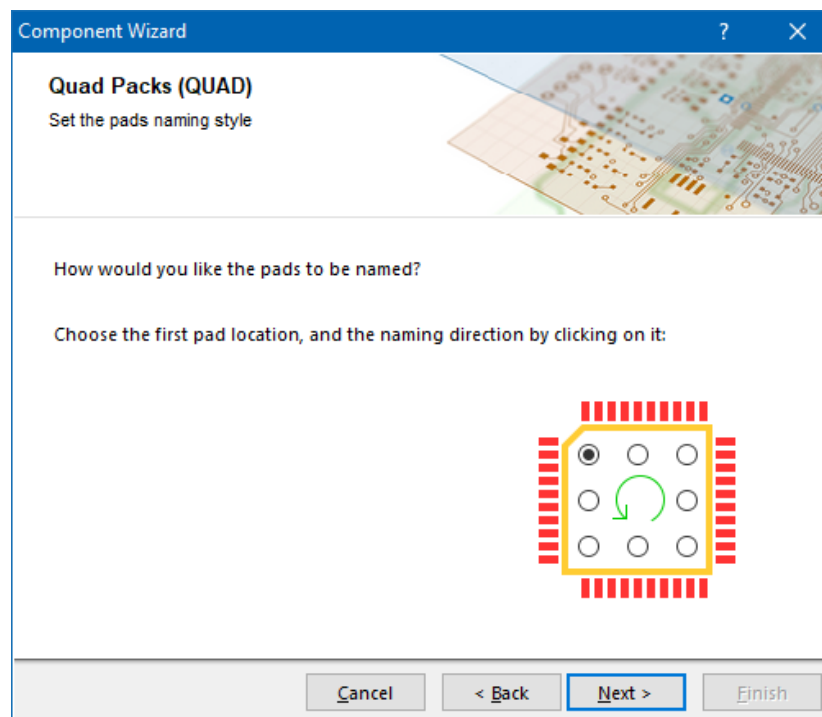


Рис. 8.11. Вікно налаштувань точки відліку та напрямку нумерації виводів

У наступному вікні вибору кількості виводів мікросхеми необхідно вказати, що вздовж кожної сторони мають розміщуватись 4 виводи. Після чого натисніть **Next**, щоб перейти до передостаннього вікна, у якому вводиться назва моделі ПМ – **QFN_16-LEAD(3x3)**. Після введення назви натисніть **Next**, а в останньому вікні – **Finish**. На рис. 8.12 представлено результат роботи майстра *Component Wizard*.

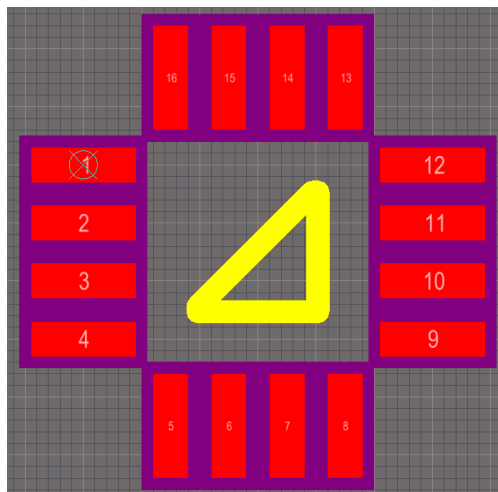


Рис. 8.12. Модель посадкового місця, що створена за допомогою *Component Wizard*

Розроблене ПМ потребує деякої модифікації для того, щоб на нього можна було встановлювати мікросхему **MD1822**. Перш за все видалимо автоматично створений контур ПМ у вигляді трикутника на шарі шовкографії – він явно не відповідає контуру мікросхеми і у такому вигляді несе мало смислової інформації.

У попередніх комп'ютерних практикумах ми суміщали точку прив'язки (центр робочої області) із першим КМ. Однак деякі стандарти регламентують встановлювати точку прив'язки у геометричний центр ПМ. Для того, щоб так зробити, виконайте команду **Edit** → **Set Reference** → **Center**. Врешті ми повинні отримати наступне зображення (рис. 8.13).

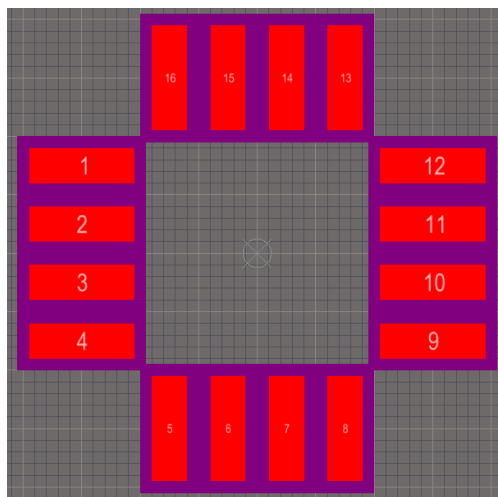


Рис. 8.13. Модель посадкового місця з точкою прив'язки у центрі

Далі необхідно створити термopідкладку та поставити її у відповідність 17-му «віртуальному» контакту мікросхеми. Для цього зробіть активним шар *Top Layer*, після чого створіть КМ із параметрами, що наведені на рис. 8.14.

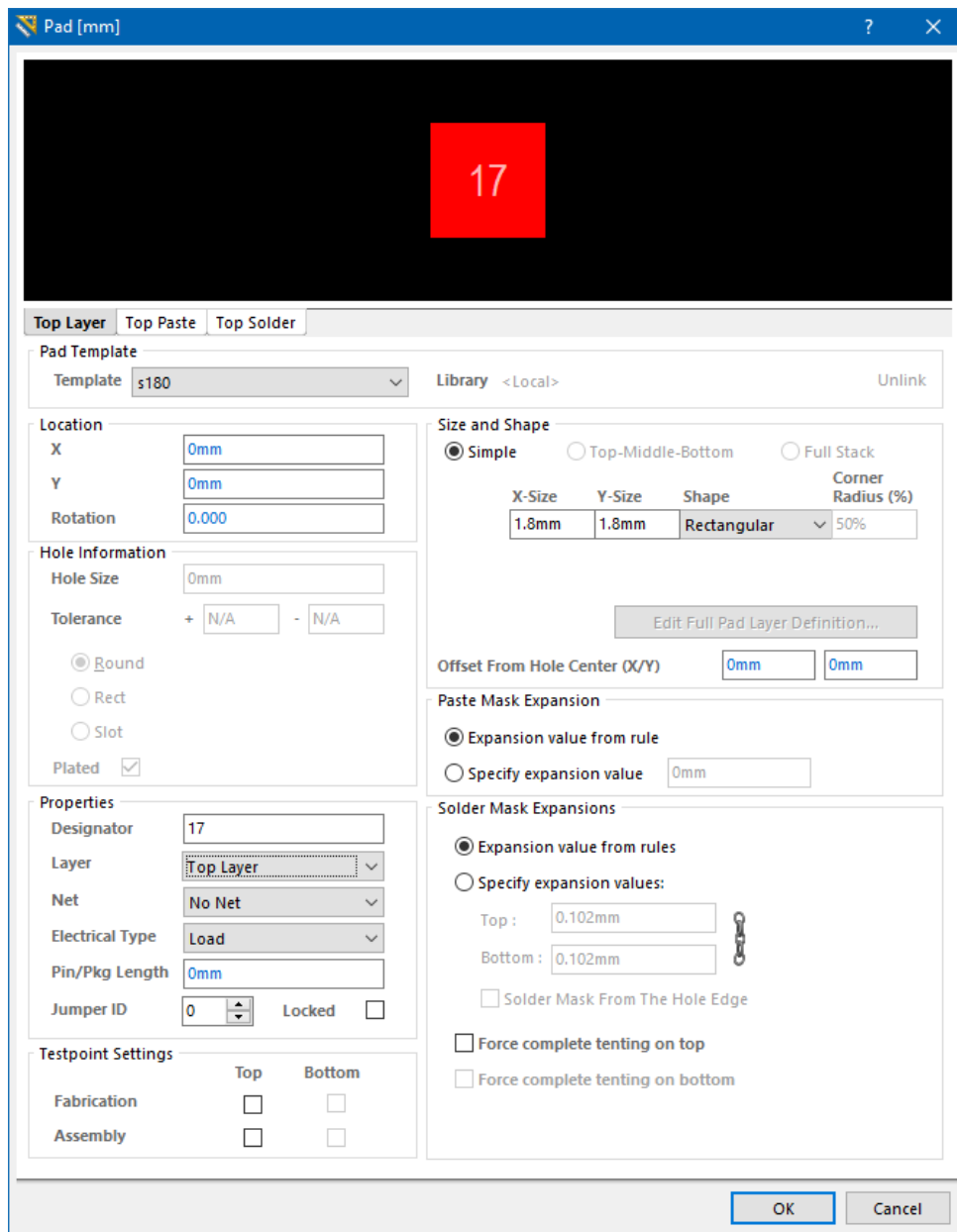


Рис. 8.14. Налаштування контактoного майданчика, що виконуватиме роль термopідкладки

Після завершення налаштувань встановіть цей КМ у центр ПМ (рис. 8.15).

Зверніть увагу на захисний шар *Top Solder*, який відображається фіолетовим кольором. Графічні примітиви на цьому шарі означають, що під час виготовлення плати на ці ділянки не буде нанесена захисна (або паяльна) маска. Як видно з рис. 8.15, фіолетовим кольором заповнені усі проміжки між КМ, тобто між ними не буде захисної маски. З одного боку, це не є проблемою, оскільки між виводами усе одно буде простір діелектрика, із якого виготовляється сама плата. А з іншого боку

– проблеми можуть виникнути. Захисна маска, по суті, також діелектрик, який наноситься на плату і захищає її від пошкоджень та зовнішніх впливів. Проте вважається, що захисна маска, на відміну від просто діелектричної основи плати, допомагає досягти кращих результатів пайки, зводячи до мінімуму ймовірність утворення паяних містків між близько розташованими КМ. Тому логічно, що паяльна маска повинна була б бути присутньою між КМ⁸³. Однак це не завжди можливо і причиною тому можуть бути обмежені технологічні можливості обладнання, на якому буде виготовлятися ДП.

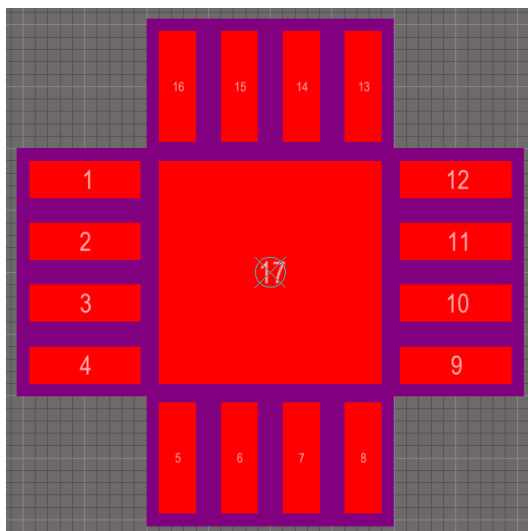


Рис. 8.15. Посадкове місце із термопідкладкою

Розглянемо на прикладі розробленого нами ПМ, які проблеми із-за неправильно обраних зазорів можуть виникати. У технічній документації (див. рис. 8.7) вказано, що відстань між КМ корпусу QFN становить 0,5 мм, тоді як максимальна ширина КМ – 0,3 мм. Це означає, що проміжок між КМ становить 0,2 мм (0,5-0,15-0,15). Зазвичай технологічне обладнання дозволяє створити захисну маску такої ширини, але це означає, що проміжок між КМ зовсім не повинен бути заповнений фіолетовим кольором, тобто не буде ніякого зазору між КМ та захисною маскою. Так можна зробити, але тоді виникає інша проблема під час виробництва – внаслідок недосконалості технології виготовлення, якщо не передбачено зазору між КМ та захисною маскою, то вона може частково покрити

⁸³ Цю ділянку плати, яка розташована між КМ, називають масковим містком. Зазвичай відсутність маскового містка порушує правила проєктування, які за замовчуванням встановлені у **Altium Designer**. Яким чином уникнути проблем, що виникають внаслідок порушення правил проєктування, буде описано у наступних комп'ютерних практикумах, які присвячені розробці принципів схем та плат.

сам КМ. А це вже буде заважати нормальному припаюванню виводів мікросхеми до КМ. Якщо ж припустити, що ширина зазору між КМ та захисною маскою становитиме 0,05 мм⁸⁴, то у нашому випадку ширина захисної маски становитиме 0,1 мм (0,5-0,15-0,15-0,05-0,05). А створити захисну маску такої ширини на платі вже може бути проблемою. Що робити у подібній ситуації? Відмовитись⁸⁵ від захисної маски у цьому місці плати. Саме це ми і спостерігаємо на прикладі розробленого нами ПМ.

Насамкінець створимо графічні примітиви у вигляді контуру корпусу мікросхеми та ще одного додаткового зовнішнього контуру, що позначає межі ПМ на шарі шовкографії. Зазвичай цей шар слугує лише для візуального відображення контурів електронних компонентів і не впливає на працездатність виробу, а на деяких платах його взагалі не використовують з міркувань економії. Але, коли мова йде про SMD-компоненти, цей шар використовують як допоміжний інструмент позиціонування та розмежування електронних компонентів. Приблизно створіть такі графічні елементи на шарі шовкографії, як це показано на рис. 8.16. Товщина ліній має бути 0,1 мм, а мінімальна відстань від ліній до КМ має бути не меншою 0,1 – 0,15 мм. Зверніть увагу, що лівий верхній кут контуру відрізняється від інших кутів – так позначено ключ мікросхеми (створити його можна за допомогою команди *Place* → *Solid Region*).

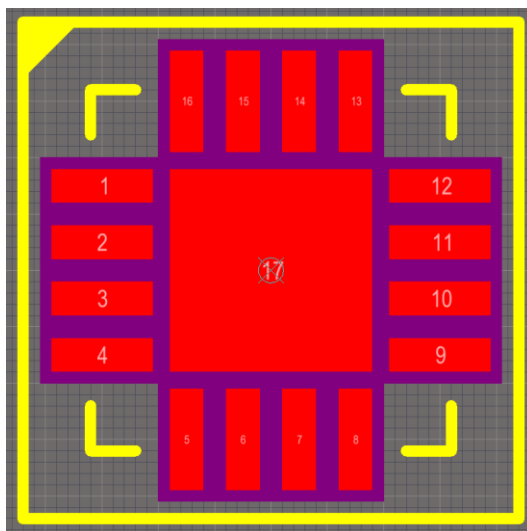


Рис. 8.16. Остаточний вигляд моделі посадкового місця

⁸⁴ За замовчуванням у **Altium Designer** ширина зазору між КМ та паяльною маскою становить 0,102 мм. Цей зазор називається *Solder Mask Expansion*.

⁸⁵ Насправді спочатку з'ясовують технологічні можливості виробництва. І вже відштовхуються від цього, вирішуючи, які зазори можуть бути між КМ та захисною маскою і яка взагалі може бути мінімальна ширина паяльної маски.

3.3. Створення компонентного модуля мікросхеми MD1822

Перейдіть у бібліотеку символів. Додайте до УГП драйвера відповідне йому ПМ. Під час додавання моделі ПМ перевірте відповідність нумерації виводів на УГП номерам КМ.

Також перевірте розроблений компонентний модуль на наявність помилок.

4. Питання для самоконтролю

1. Поясніть, що таке драйвер транзистору? Для чого він призначений?
2. Поясніть, яким чином змінити властивості групи контактів?
3. Поясніть, як розробити УГП драйвера транзистора MD1822?
4. Поясніть, яким чином створити посадкове місце для мікросхеми у корпусі QFN?
5. Поясніть, для чого призначена паяльна (або захисна) маска?
6. Поясніть, що таке масковий місток?
7. Поясніть, для чого використовується параметр *Solder Mask Expansion*?

Комп'ютерний практикум № 9.

РОЗРОБКА 3D-МОДЕЛЕЙ КОРПУСІВ ЕЛЕКТРОННИХ КОМПОНЕНТІВ ТА ЇХ ДОДАВАННЯ ДО БІБЛІОТЕЧНИХ КОМПОНЕНТНИХ МОДУЛІВ

***Мета роботи:** навчитись створювати тривимірні моделі різних електронних компонентів та мікросхем за допомогою середовища SolidWorks; підключити тривимірні моделі до компонентних модулів в Altium Designer.*

1. Теоретичні відомості

3D-проекування – це процес створення тривимірної (об'ємної) комп'ютерної моделі бажаного об'єкта. Графічне зображення тривимірних об'єктів відрізняється тим, що включає побудову геометричної проекції тривимірної моделі сцени на площину (наприклад, екран комп'ютера, ноутбука, планшета тощо) за допомогою спеціалізованих програм.

Головною перевагою 3D-проекування є наочність кінцевого результату, коли проєкт представлено не у вигляді безлічі різних малозрозумілих двовимірних креслень, а як реалістичну тривимірну модель, яку можна обертати та розглядати з наближенням та під різними кутами. Завдяки тривимірним моделям можна попередньо ознайомитися з майбутнім проєктом, уточнити деякі деталі, внести правки, додаючи щось нове та видаляючи зайве.

На сьогоднішній день існує багато спеціальних програм, які дають змогу швидко створювати високодеталізовані тривимірні моделі, що максимально наближені до справжніх об'єктів. Поширеними програмами тривимірного проєкування, які застосовуються в технічних галузях, є SolidWorks, AutoCAD, КОМПАС-3D тощо.

Тривимірне моделювання сьогодні є невід'ємною частиною будь-якого проєкту ДП, оскільки завдяки ньому можна забезпечити коректність конструкції ще на етапі проєкування, коли плата існує лише у віртуальному просторі.

Altium Designer була однією із перших систем проєкування плат, у якій з'явився режим тривимірного конструювання. В **Altium Designer** реалізована можливість додавання тривимірних моделей корпусів електронних компонентів як на рівні бібліотечних компонентів, так і безпосередньо під час розробки плати.

Причому додати тривимірну модель можна одним із двох основних способів. Перший спосіб полягає у використанні вбудованих інструментів тривимірного проектування, що дуже зручно. Проте ці інструменти надають досить обмежені можливості⁸⁶ і не дозволяють створювати реалістичні моделі⁸⁷. Другий спосіб полягає у розробці тривимірної моделі у деякій спеціалізованій програмі тривимірного проектування, конвертуванні її у проміжний формат (наприклад, STEP) та наступного підключення до бібліотечного компонента або до проекту плати. Перевагою цього способу є можливість створення реалістичних моделей корпусів електронних компонентів (рис. 9.1).

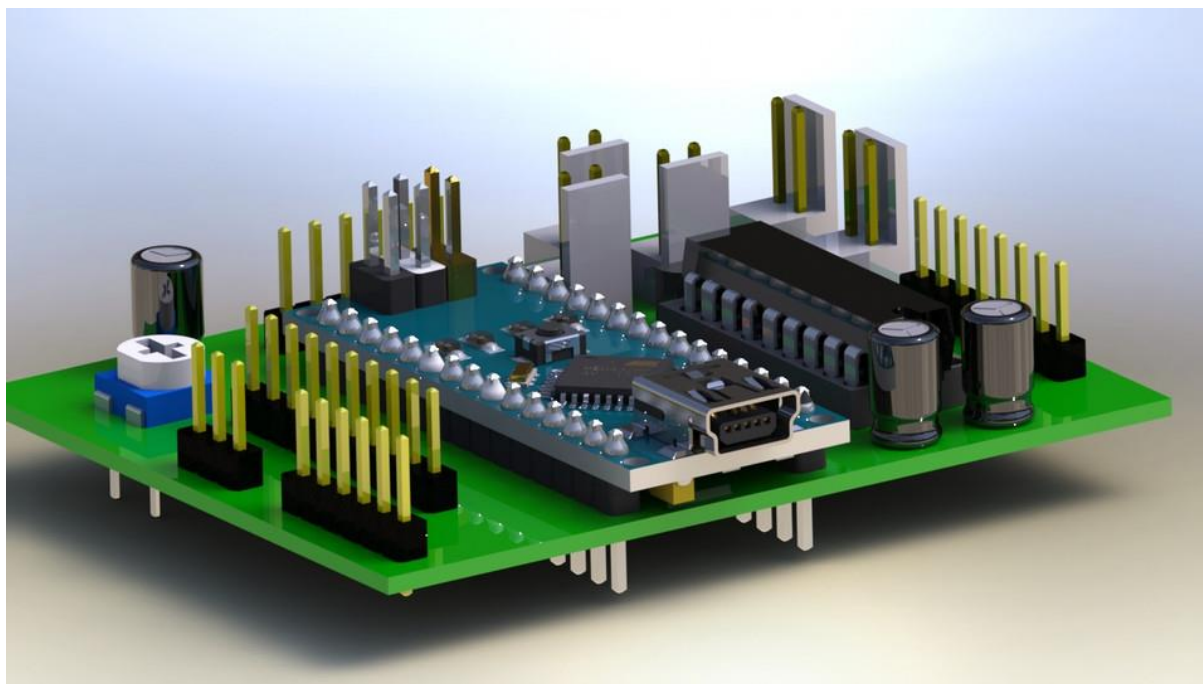


Рис. 9.1. [Тривимірна модель плати](#)

⁸⁶ У цьому ви могли переконатись, виконуючи попередні комп'ютерні практикуми. На деяких із них ми створювали моделі корпусів мікросхем у вигляді простого прямокутного паралелепіпеда. Звичайно, якщо докласти зусиль, то і за допомогою наявних можливостей можна створити досить непогану тривимірну модель шляхом об'єднання доступних простих форм тривимірних об'єктів: об'ємний багатогранник (окремим випадком якого є прямокутний паралелепіпед), сфера та циліндр. Таким чином, будь-яка загальна тривимірна модель може складатись із безлічі простих форм. Однак збільшення простих форм, із яких складатиметься модель, призводитиме до значного ускладнення процесу її редагування за допомогою наявних в **Altium Designer** можливостей.

⁸⁷ Тут ми не беремо до уваги майстер *IPC Compliant Footprint Wizard*, за результатами роботи якого створюється досить реалістична модель корпусу. Під час роботи цього модуля модель створюється автоматично. А також необхідно враховувати, що перелік електронних компонентів, для яких можна створити ПМ та тривимірну модель корпусу за допомогою цього майстра, є невеликим.

Зазвичай у проєктах **Altium Designer** використовують другий спосіб, додатковою перевагою якого є те, що, якщо ви самі не вмієте створювати тривимірні моделі в жодній зі спеціалізованих програм, ви можете використовувати моделі, що створені іншими користувачами.

Також варто відзначити, що наразі активно розвивається технологія двонаправленого обміну даними між **Altium Designer** та САПР механічних пристроїв, що ще більше полегшує процес розробки та аналізу.

Окрім того, що завдяки тривимірному відображенню зручно виконувати візуальний аналіз, **Altium Designer** дозволяє виконувати автоматизовані перевірки перетину моделей корпусів електронних компонентів і сигналізувати про конфліктні ситуації. За це відповідає правило ***Component Clearance***, яке можна налаштувати у редакторі правил ***PCB Rules and Constraint Editor***. Також у тривимірному режимі розробнику доступні такі інструменти, як вимірювання відстаней у межах однієї моделі (***Tools** → **3D Body Placement** → **Measure Distance***), так і вимірювання відстаней між різними моделями (***Reports** → **Measure 3D Objects***).

У цьому комп'ютерному практикумі для створення тривимірних моделей корпусів електронних компонентів буде використовуватись САПР SolidWorks, яка призначена для автоматизованого елементно-орієнтованого конструювання твердотільних моделей. Перевагою розробки моделі в САПР SolidWorks є то, що для того, щоб її підключити до бібліотечного компоненту, не потрібно конвертувати створену модель у формат STEP. Це тому, що **Altium Designer** підтримує пряме імпортування як у форматах STEP (*.step, *.stp), Parasolid (*.x_t, *.x_b), так і у форматі файлу деталі SolidWorks (*.sldprt).

Насамкінець, варто зауважити, що для того, щоб правильно розробити тривимірну модель корпусу електронного компоненту, необхідно спочатку ознайомитись із технічною документацією та знайти в ній його габаритні розміри. У комп'ютерному практикумі під час побудови моделей використовуватимуться саме ті розміри, які взяті із технічної документації для конкретного електронного компоненту.

2. Завдання на комп'ютерний практикум



1. Створити в САПР **SolidWorks** тривимірні моделі резистора для монтажу в отвори серії **MF**, електролітичного конденсатора **Hitano ECR**, котушки індуктивності **KIG**, мікросхеми у корпусі **SOT-23-5** та мікросхеми у корпусі **DIP14**, що має метричний крок виводів 2,5 мм.

2. Конвертувати тривимірні моделі у формат STEP⁸⁸.
3. Підключити власні створені STEP-моделі до ПМ в **Altium Designer**.
4. Отримати у викладача (або знайти в інтернеті самостійно) STEP-моделі інших електронних компонентів, які були розроблені на попередніх комп'ютерних практикумах, та підключити ці моделі до бібліотечних компонентних модулів.

3. Порядок виконання комп'ютерного практикуму

3.1. Проєктування моделі резистора для монтажу в отвори

Запустіть програму **SolidWorks**⁸⁹. Створіть новий проєкт для побудови тривимірної моделі (деталі). Для цього у меню **File** оберіть команду **New**. У вікні **New SOLIDWORKS Document**, що відкриється, оберіть пункт **Part** та натисніть **OK**.

У новому (і поки що пустому вікні, призначеному для створення тривимірної моделі) спочатку у лівій бічній панелі на вкладці **FeatureManager Design Tree** клацніть ПКМ на рядку **Front Plane** (передня площина) і у спливаючому меню оберіть пункт **Sketch** (піктограма ). Після цього на панелі інструментів активуйте інструмент **Circle**. Встановіть курсор в точку початку координат і намалюйте коло довільного радіусу. Для завершення операції натисніть <Esc>. Далі, натиснувши ЛКМ на колі, у лівій бічній панелі на вкладці **Property Manager (Менеджер властивостей)** вказуємо точне значення радіусу 1,15 мм. Для того, щоб зафіксувати потрібне значення радіусу, натисніть зелену галочку на вкладці **Property Manager**, а для того, щоб завершити формування поточного ескізу (sketch), натисніть напівпрозору піктограму  у верхньому правому куті робочої області (рис. 9.2).

⁸⁸ Хоча раніше було написано, що **Altium Designer** може безпосередньо завантажувати моделі у форматі *.sldprt, проте необхідно враховувати деякі нюанси. А саме те, що версія **Altium Designer** має бути новішою за **SolidWorks** для забезпечення сумісності. Проте, оскільки у навчальному посібнику використовується **Altium Designer** 17.1, а для розробки моделей буде використовуватись **SolidWorks** 2021, то, на жаль, **Altium Designer** 17.1 не зможе прочитати новий формат *.sldprt. Саме тому ми будемо конвертувати тривимірні моделі корпусу у формат STEP, а вже потім підключати до компонентного модуля.

⁸⁹ Для моделювання та написання протоколу комп'ютерного практикуму використовувалась англomовна версія програми 2021 року. Враховуйте, що у новіших версіях програми можуть змінюватись назви елементів меню, тому будьте уважні та шукайте подібні команди самостійно, якщо це буде необхідно.

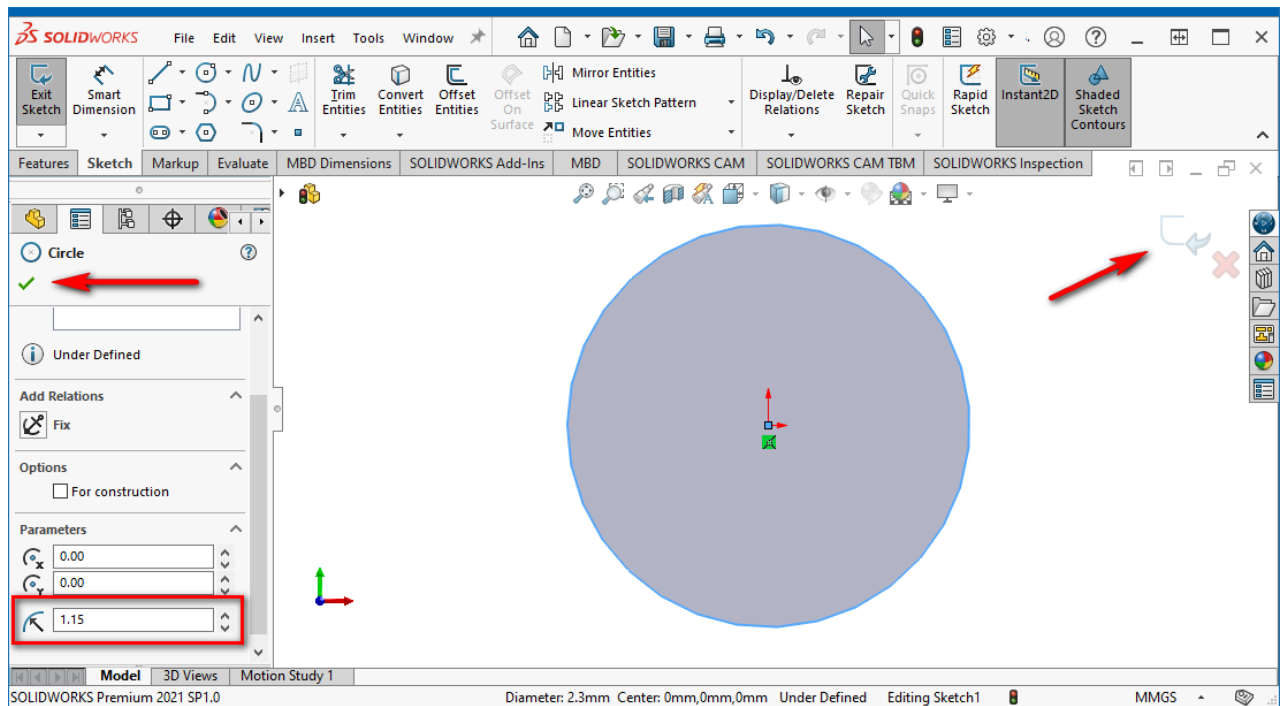


Рис. 9.2. Початковий ескіз, що містить коло

Наступними кроком буде створення витягнутого елемента із наявного ескізу. Відкрийте панель **Features** та оберіть команду **Extruded Boss/Base**. У **Менеджері властивостей** задаємо довжину моделі $D1=3,25$ мм (у напрямку 1). Для завершення операції натисніть ЛКМ на зеленій галочці в правому верхньому куті (рис. 9.3). Після цієї операції у робочій області буде зображена аксонометрична проекція циліндра, яка визначатиме корпус резистора.

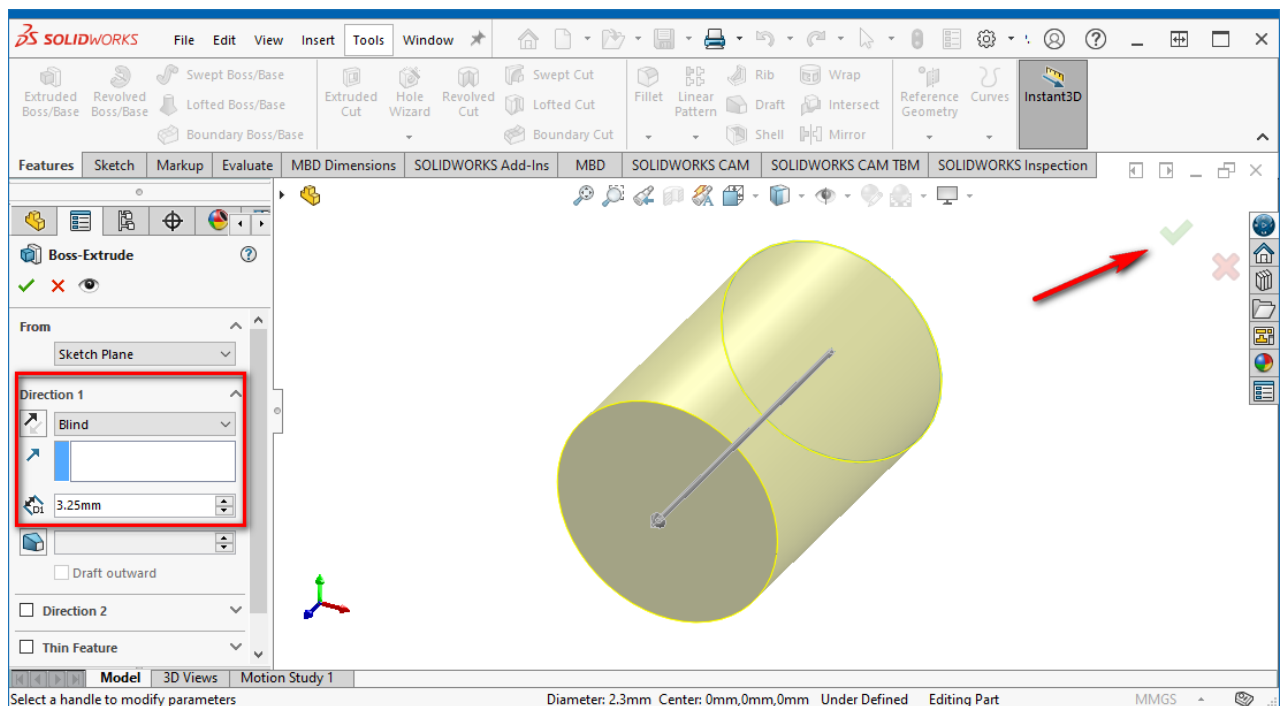


Рис. 9.3. Створення циліндра, основою якого є ескіз кола

Тепер перейдемо до побудови виводів (ніжок) резистора. Використовуючи елементи меню (наприклад, **View Orientation** у верхній частині робочої області) знову встановіть відображення передньої площини. Далі у **FeatureManager Design Tree** обираємо **Front Plane**, а у спливаючому меню – **Sketch**. Після чого у панелі інструментів активуйте інструмент **Circle** та намалуйте ще одне коло радіусом 0,3 мм з центром у точці початку координат (рис. 9.4). Завершіть формування ескізу натисканням відповідної піктограми.

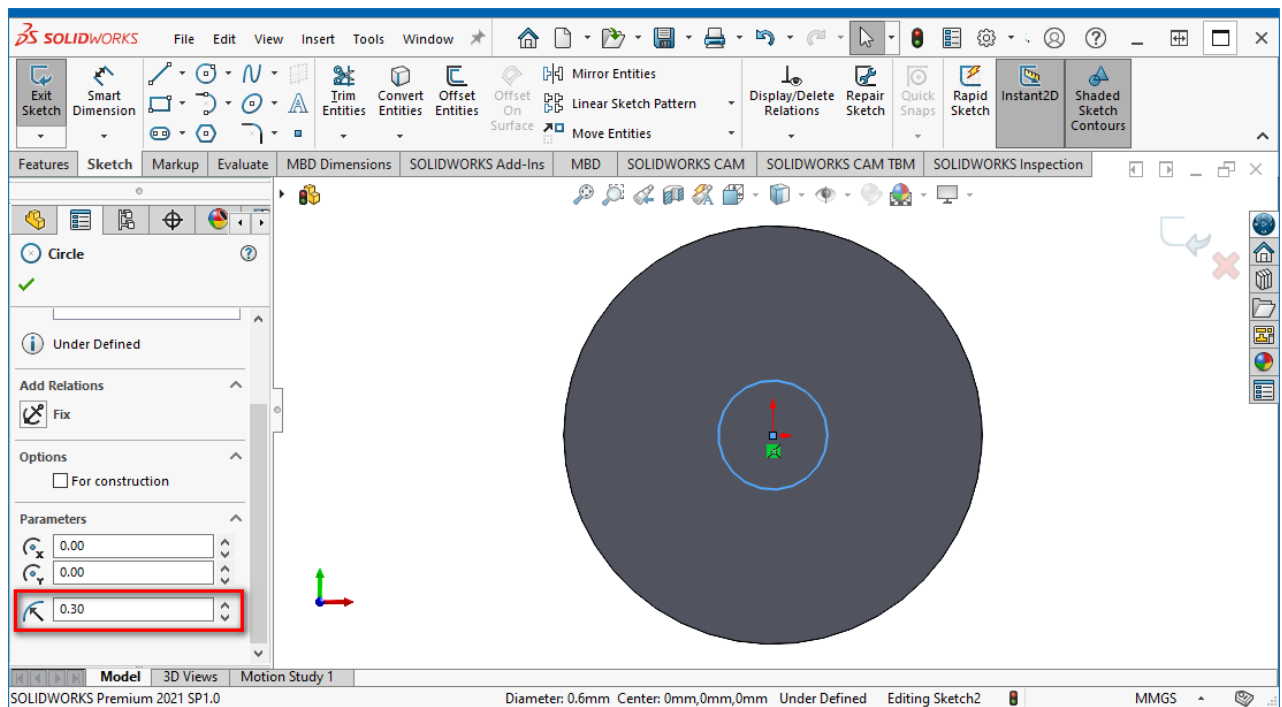


Рис. 9.4. Ескіз кола, яке буде основою виводу резистора

Для створення вигнутої форми ніжок резистору буде використовуватись наступний підхід: спочатку ми встановимо три контрольні точки, після чого через них проведемо необхідні лінії⁹⁰. Встановіть відображення моделі з правої сторони. Після чого у **FeatureManager Design Tree** обираємо площину **Right Plane**, клацаємо на ній ПКМ та у спливаючому меню активуємо **Sketch**. Далі оберіть інструмент **Point**. Розмістіть три точки (рис. 9.5) за наступними координатами (X; Y)⁹¹: перша точка – (1,25; 0), друга – (1,75; -0,5), третя – (1,75; -5). Завершіть формування ескізу натисканням відповідної піктограми.

⁹⁰ Взагалі, можна не використовувати контрольні точки, а намалювати одразу необхідні лінії. Проте це потребує більшого досвіду роботи із **SolidWorks**.

⁹¹ Спочатку ви можете їх поставити довільно, а потім через **Менеджер властивостей** задати точні координати.

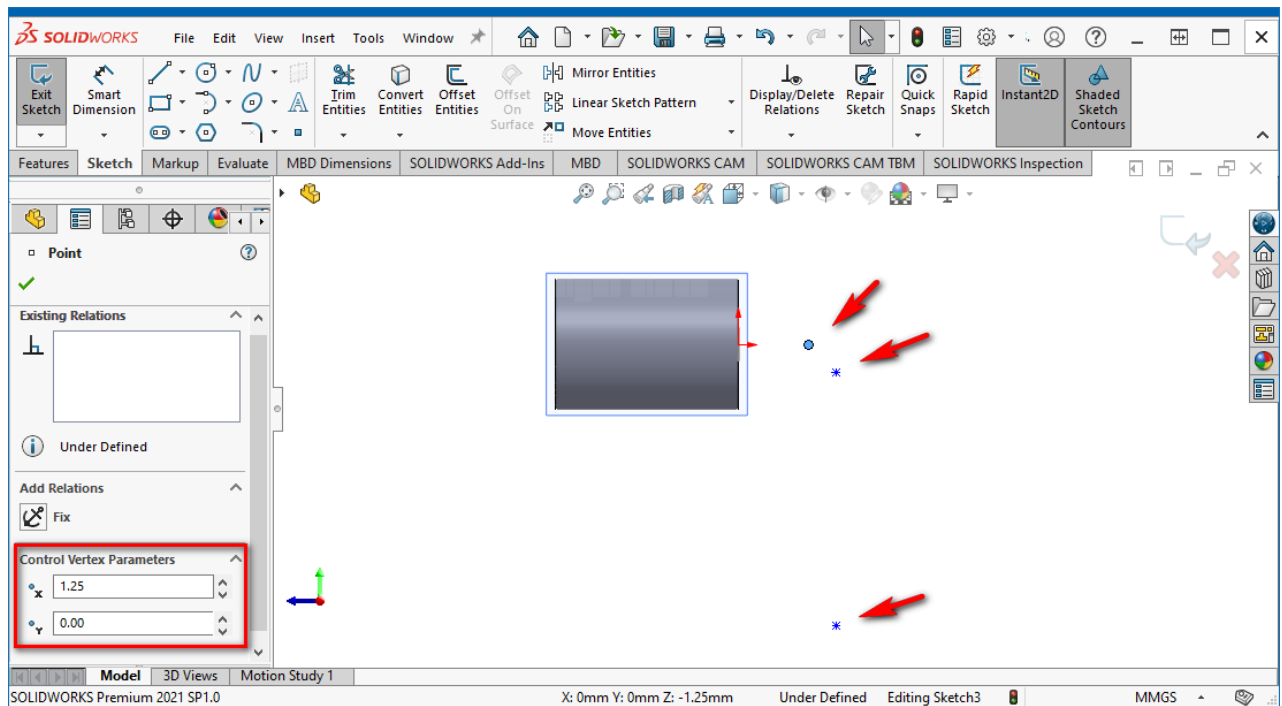


Рис. 9.5. Ескіз із контрольними точками

Наступним кроком з'єднаємо початок координат і першу точку, а також другу і третю точки прямими лініями, використовуючи інструмент **Line** із панелі **Sketch**. Далі першу і другу точки з'єднаємо дугою, причому необхідно використовувати інструмент **Tangent Arc** із все тієї ж панелі інструментів **Sketch**. Результат виконання цих дій представлено на рис. 9.6. Завершіть формування ескізу натисканням відповідної піктограми.

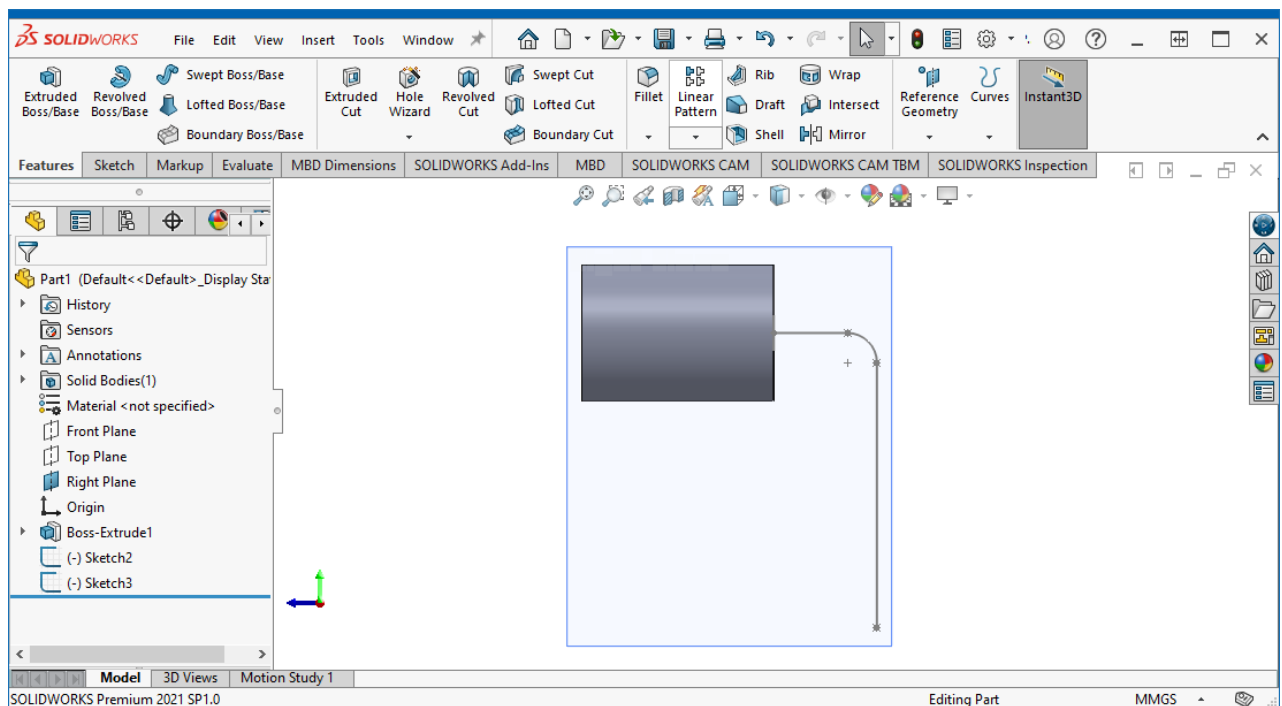


Рис. 9.6. Ескіз траєкторії, яка використовуватиметься для формування виводу резистора

Для створення об'ємної фігури за деякою траєкторією необхідно відкрити панель **Features** та обрати команду **Swept Boss/Base**. Після вибору команди необхідно вказати основу (профіль) та маршрут. Як основу вказуєте коло радіусом 0,3 мм шляхом натискання ЛКМ на ньому⁹², а як маршрут – виділяєте усі лінії та криві, що визначають форму ніжки. Водночас зверніть увагу, що у **Менеджері властивостей** елементи, які належать до профіля, мають відображатись у рядку, який позначено синім кольором, а елементи маршруту – у рядку, який позначено рожевим кольором (рис. 9.7). Для застосування змін натисніть зелену галочку в правому верхньому куті.

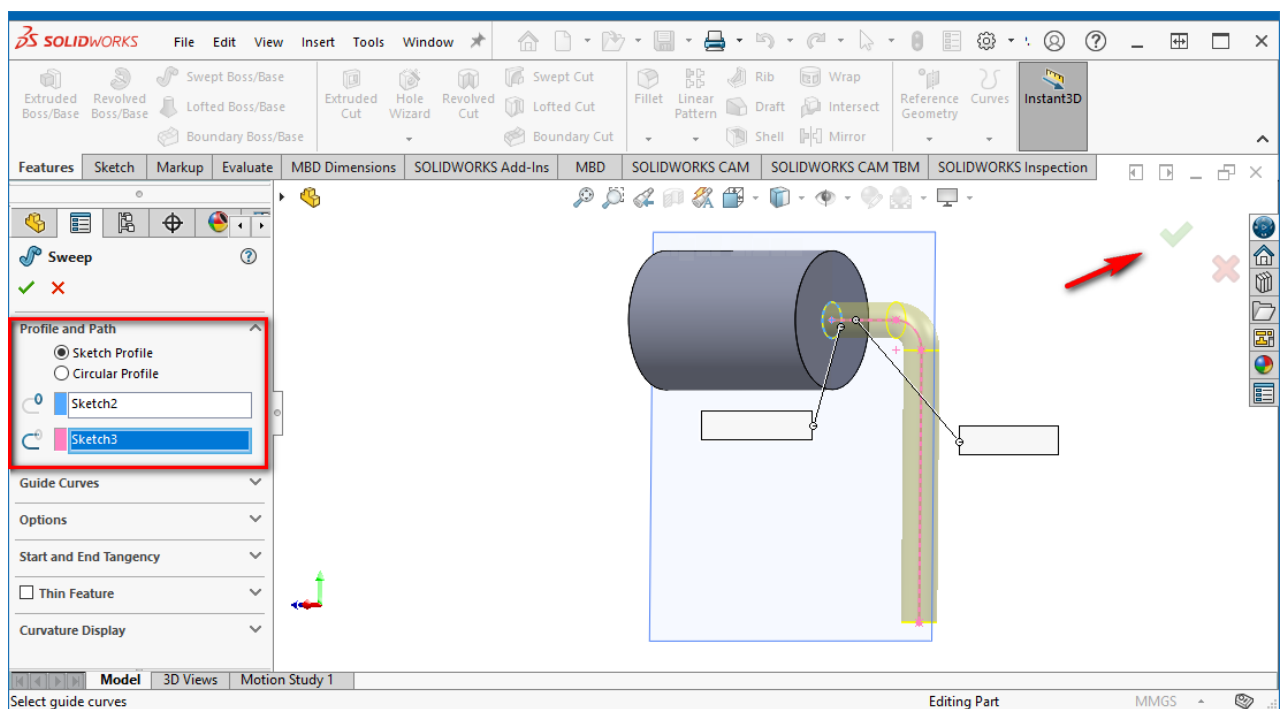


Рис. 9.7. Створення об'ємної фігури за траєкторією

Наразі ми створили тільки половину резистора. Другу половину зробимо, використовуючи інструмент дзеркального відображення. Для початку поверніть модель таким чином, щоб основа циліндра, яка не містить ніжки резистора, була спереду. В панелі **Features** оберіть команду **Mirror**, після чого оберіть грань (**Mirror Face/Plane**), відносно якої відбуватиметься віддзеркалення, – у нашому випадку це буде основа циліндру, яка не містить ніжки резистора. А також оберіть елементи, які необхідно віддзеркалити, – це корпус резистора та ніжка (можна обрати тільки один із елементів, що формує траєкторію ніжки, головне, щоб у **Менеджері властивостей** в області **Features to Mirror** з'явився пункт **Sweep**). Для застосування змін натисніть зелену галочку в правому верхньому куті (рис. 9.8).

⁹² Для полегшення вибору рекомендується спочатку дещо повернути загальний вид моделі.

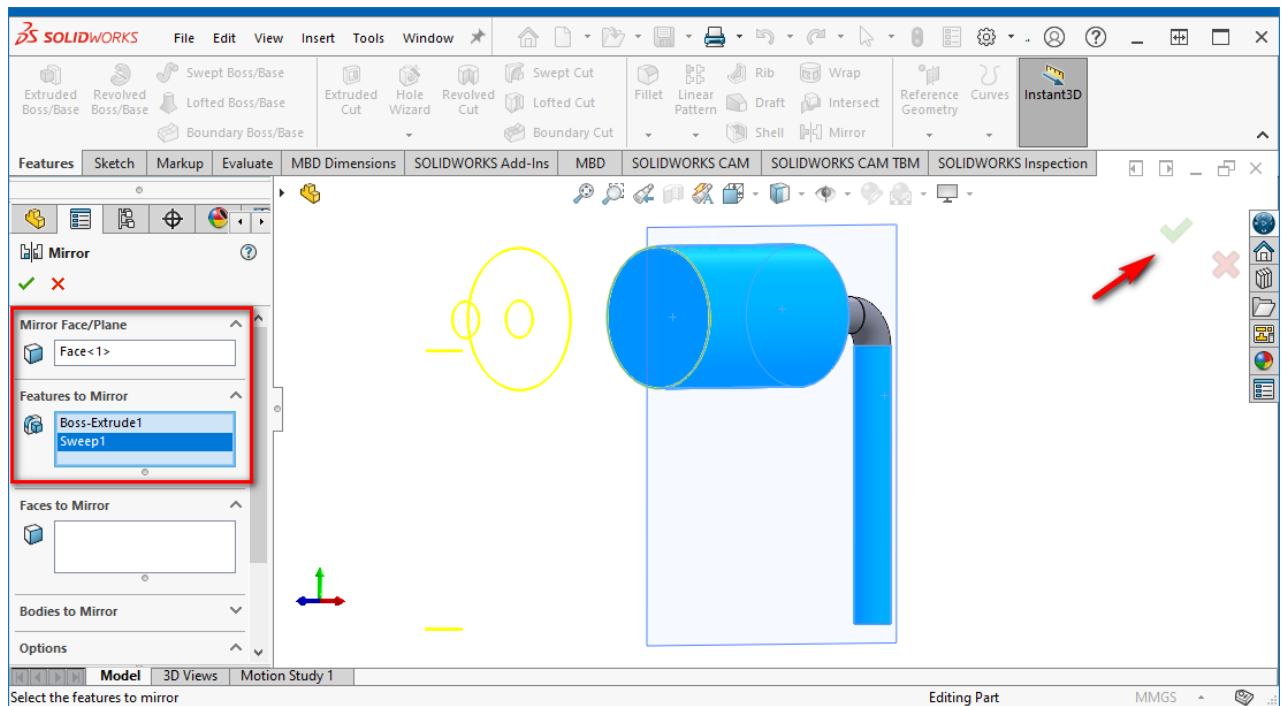



Рис. 9.8. Дзеркальне відображення деталі

Далі зафарбуємо корпус резистору та виводи у потрібні нам кольори. Натисніть піктограму  (**Edit Appearance**) у верхній частині робочого простору, після чого у лівій бічній панелі стануть доступними інструменти для вибору граней та елементів моделі, до яких можна застосувати кольори. Оберіть дві поверхні, що визначають основи циліндру, та його бічну грань (рис. 9.9) і задайте коричневий колір. Після цього подібним чином змініть колір виводів резистора на світло-сірий.

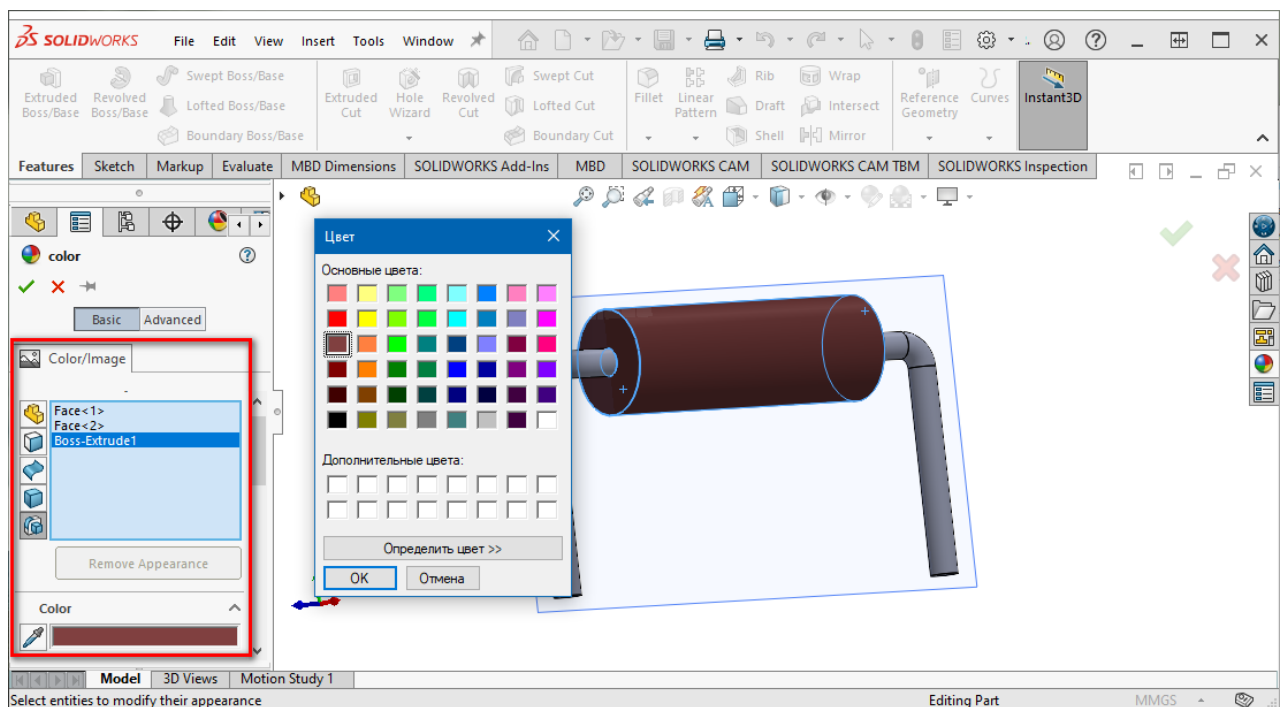


Рис. 9.9. Встановлення кольору корпусу резистора

Кінцевий варіант розробленої моделі резистору представлено на рис. 9.10. Додатково можете самостійно налаштувати параметри сцени для більш реалістичнішого відображення.

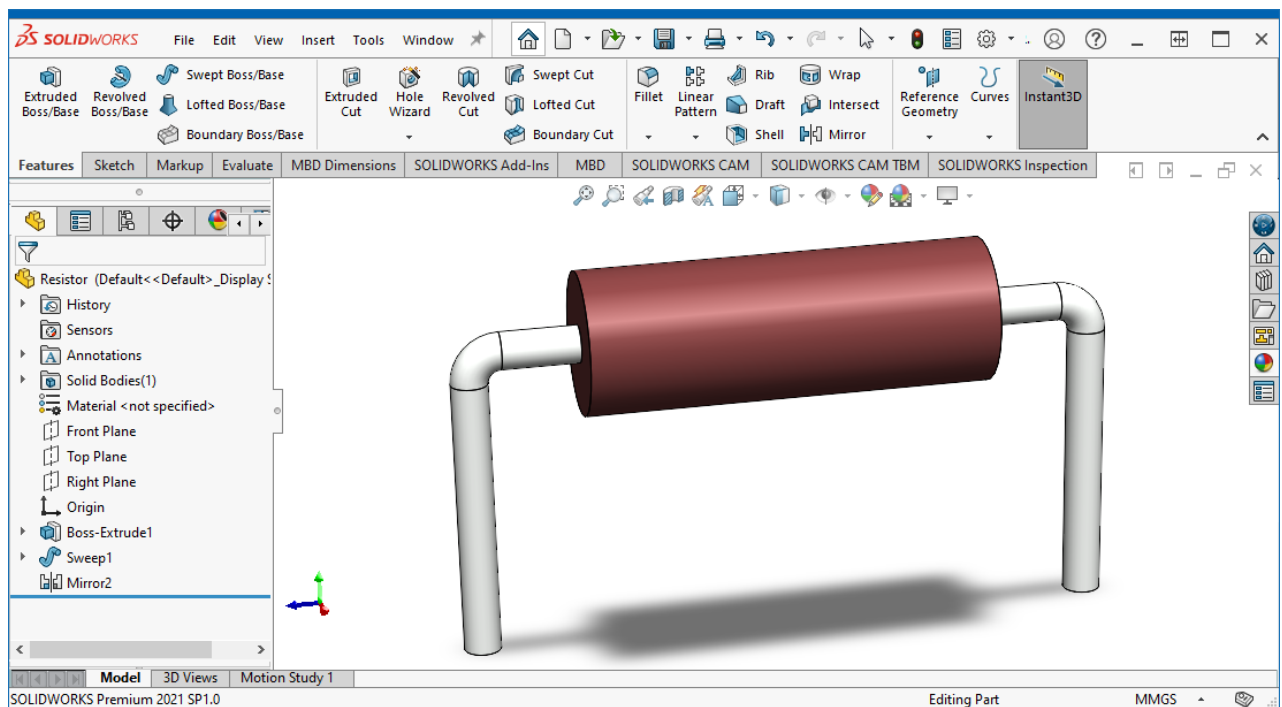


Рис. 9.10. Кінцевий варіант тривимірної моделі резистора для монтажу у отвори

Збережіть результат, задавши ім'я **ResMF025**, спочатку у форматі ***.sldprt**, а потім, обравши команду **Save As** – у форматі ***.step**, причому оберіть тип **STEP AP214** (рис. 9.11).

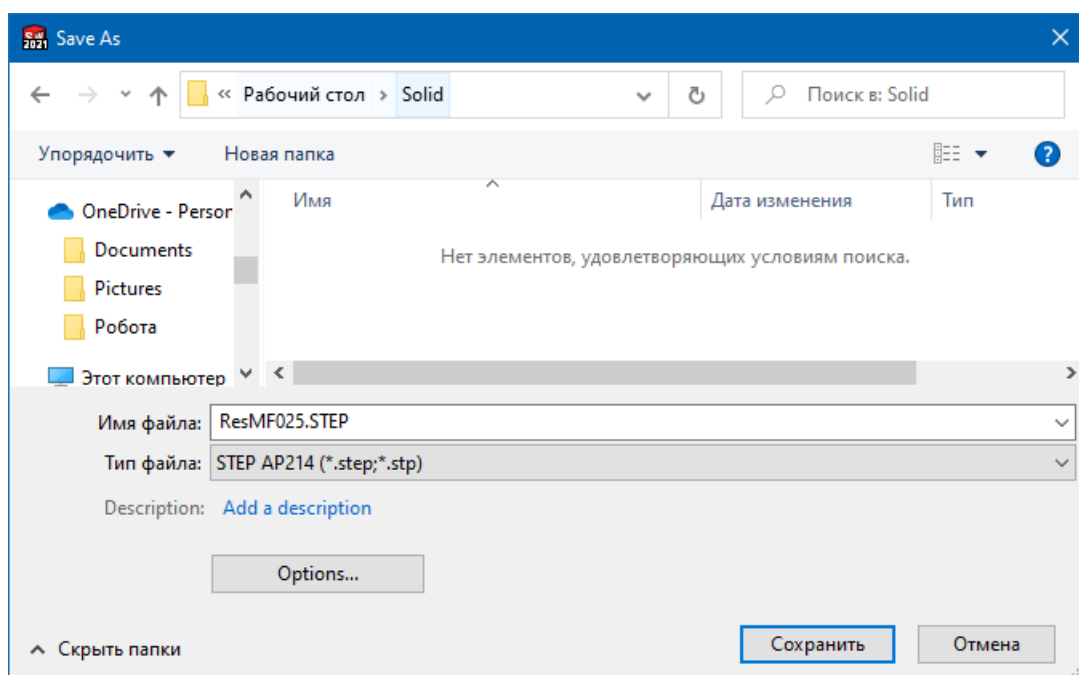


Рис. 9.11. Діалог збереження файлу моделі у форматі STEP

Розроблена нами модель корпусу є досить спрощеною, а її колір не відповідає кольорам, у яких випускаються резистори серії MF. Проте і цього може бути достатньо, щоб розуміти загальний вигляд плати, коли буде створюватись її тривимірна модель.

За бажанням ви можете самостійно спробувати розробити більш реалістичну модель, використовуючи дані, що наведені на рис. 9.12.

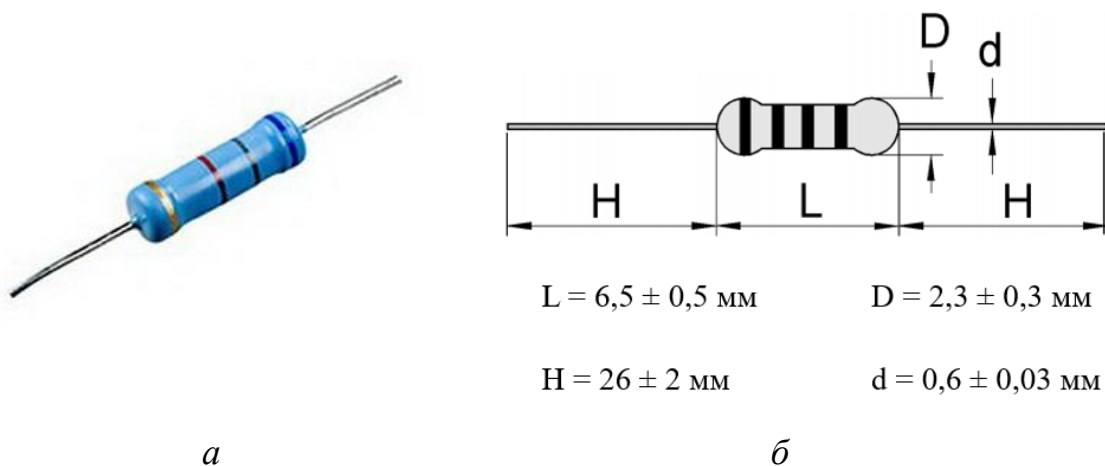


Рис. 9.12. Резистор серії MF: *a* – зовнішній вигляд⁹³, *б* – геометричні розміри⁹⁴ резистора потужністю 0,25 Вт

3.2. Проєктування моделі електролітичного конденсатора

Створіть нову заготовку моделі. У лівій бічній панелі на вкладці ***FeatureManager Design Tree*** клацніть ПКМ на рядку ***Front Plane*** і у спливаючому меню оберіть пункт ***Sketch***.

Після цього на панелі інструментів активуйте інструмент ***Corner Rectangle***. Встановіть курсор у точку початку координат і намалюйте прямокутник розміром 3 x 11 мм (причому довша сторона має бути вздовж горизонталі). Ви можете спочатку намалювати прямокутник довільних розмірів, а вже потім у ***Менеджері властивостей*** вказати точні значення довжин сторін, попередньо виділивши необхідну сторону (рис. 9.13).

⁹³ Смушками на корпусі кодується значення опору. Під час розробки загальної тривимірної моделі смужки можна не ставити.

⁹⁴ Розміри вказані саме для моделі резистора потужністю 0,25 Вт, оскільки для інших потужностей розміри відрізнятимуться, хоча сам зовнішній вигляд і форма корпусу зберігатимуться.

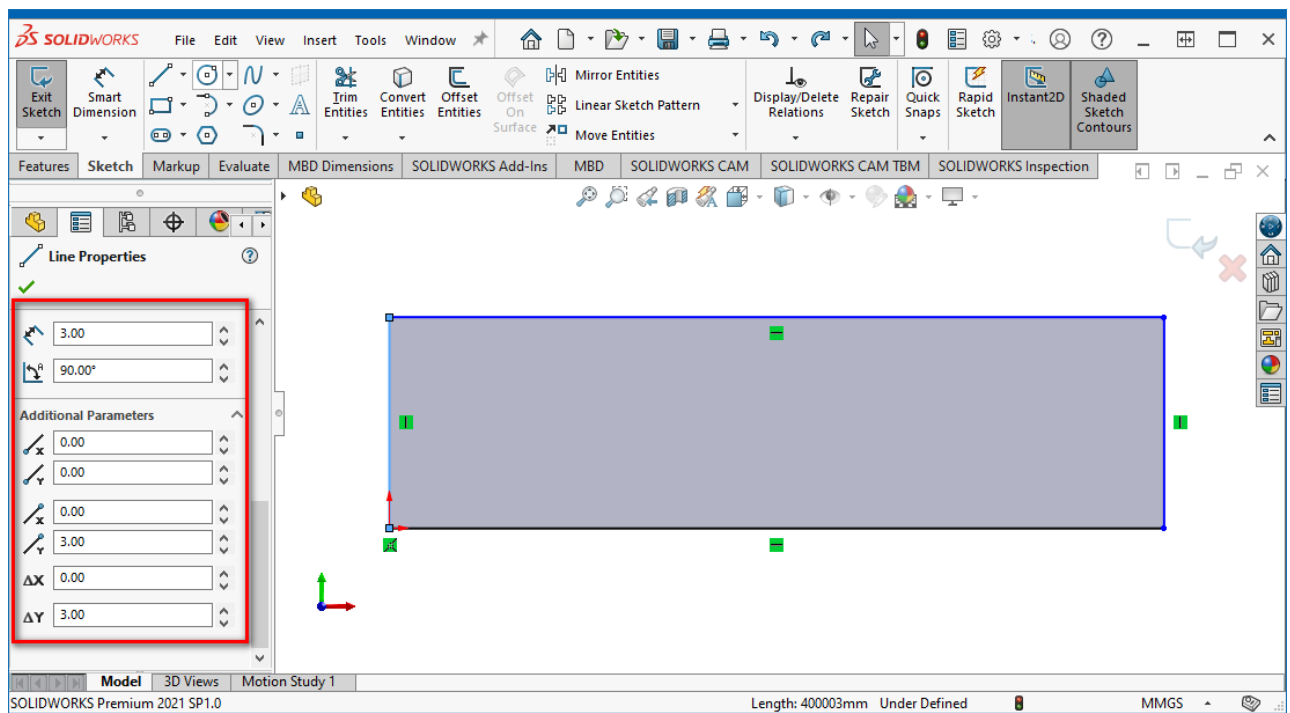


Рис. 9.13. Ескіз прямокутника

Намалюємо півколо, на основі якого надалі буде сформоване заглиблення в корпусі конденсатора. Переходимо в *Sketch* → *3 Point Arc* та малюємо довільним чином півколо. Після цього відкриваємо *Менеджер властивостей* та встановлюємо наступні значення координат трьох точок дуги (рис. 9.14): центральна – (1,5; 3), бокові – (1,1; 3) та (1,9; 3).

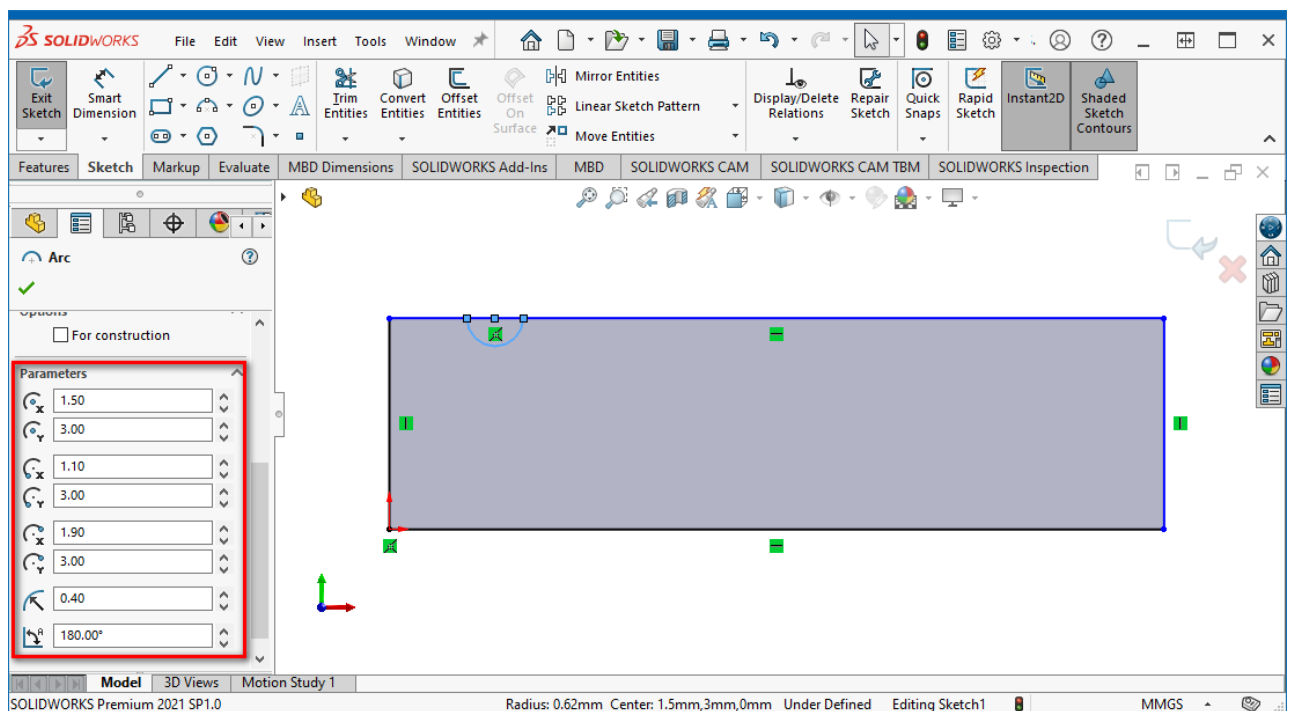


Рис. 9.14. Формування дуги на одній із сторін прямокутника

Відсікаємо непотрібні частини для отримання правильного контуру корпусу конденсатора, використовуючи команду *Sketch* → *Trim Entities* із увімкненою опцією *Power Trim*. Для того, щоб залишилась западина у формі дуги, необхідно після вибору команди *Trim Entities* затиснути ЛКМ та провести лінію (відображається сірим кольором), яка перетинатиме грань прямокутника в межах півкола (рис. 9.15). Для виходу із режиму відсікання натисніть <Esc>, а для застосування змін – зелену галочку.

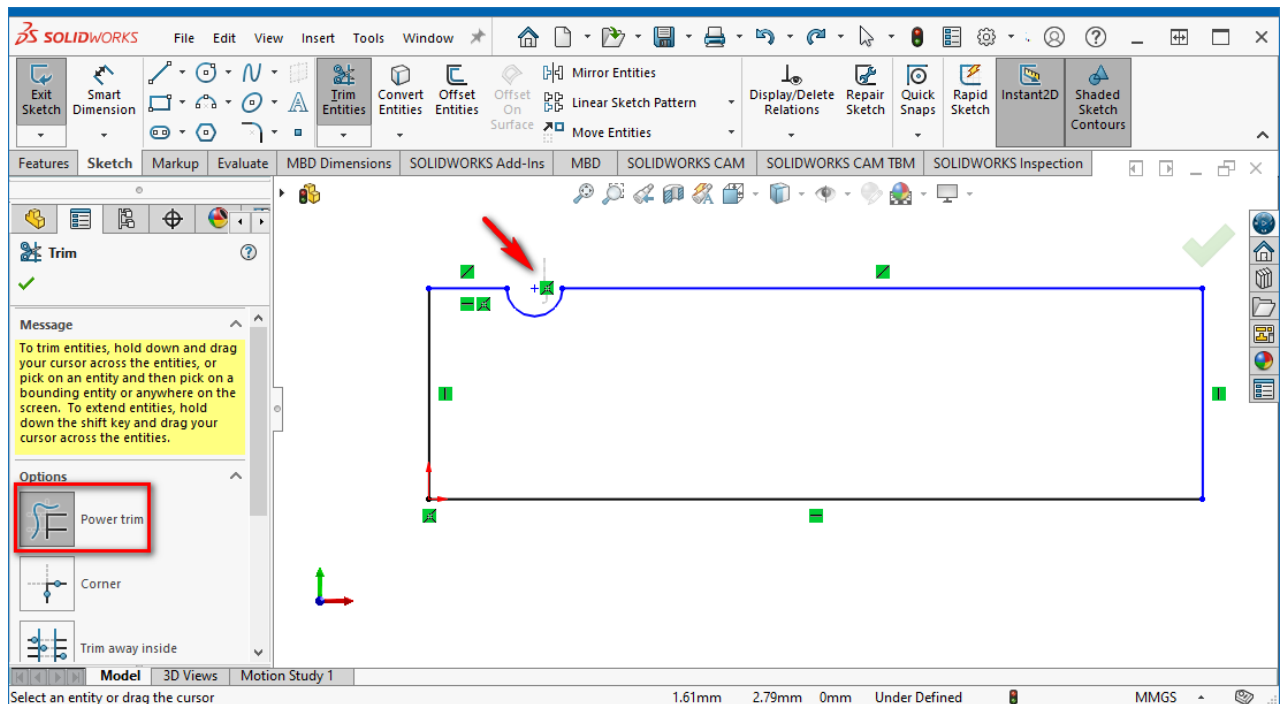


Рис. 9.15. Відсікання частини контуру

Далі із наявної плоскої фігури методом обертання створимо тривимірну модель корпусу. Перейдіть в *Features* → *Revolved Boss/Bases*, після чого клацніть ЛКМ на нижній грані прямокутника. Перевірте у *Менеджері властивостей*, що в області *Axis of Revolution* активною є нижня грань прямокутника⁹⁵, а в області *Direction1* – встановлений кут обертання 360° (рис. 9.16). Завершіть операцію натисканням ЛКМ на зеленій галочці в правому верхньому куті робочої області.

Використовуючи елементи меню встановіть відображення моделі таким чином, щоб площа основи циліндра, яка ближче до заглиблення на корпусі

⁹⁵ Зверніть увагу, що на рис. 9.16 ця грань позначена як *Line1@Sketch1*, що можна прочитати як «перша лінія першого ескізу». Це стандартна форма запису в *SolidWorks*, завдяки якій можна чітко розуміти приналежність графічних об'єктів до ескізів. У вас може бути дещо інший запис в сенсі нумерації лінії та ескізу. Наприклад, якщо ви створили перший ескіз з прямокутником, а потім з деяких причин його видалили і створили новий, то він матиме номер 2.

конденсатора, була направлена на вас (якщо ви робили усе за вказівками, то потрібно обрати площину *Left*).

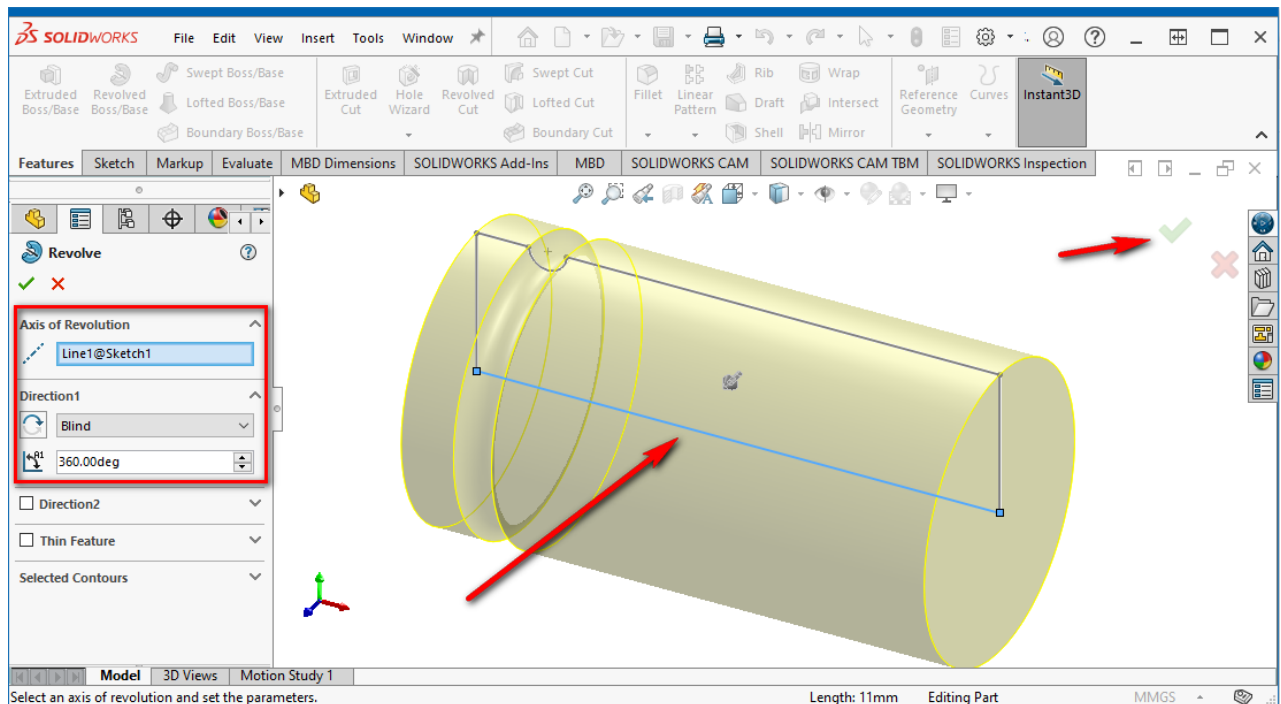


Рис. 9.16. Створення об'ємної фігури методом обертання

Далі у лівій бічній панелі на вкладці *FeatureManager Design Tree* клацніть ПКМ на рядку *Right Plane* і у спливаючому меню оберіть пункт *Sketch*. Після цього активуйте інструмент *Circle*. Намалуйте коло (рис. 9.17) радіусом 0,25 мм, центр якого знаходиться в координатах (0; 1,25).

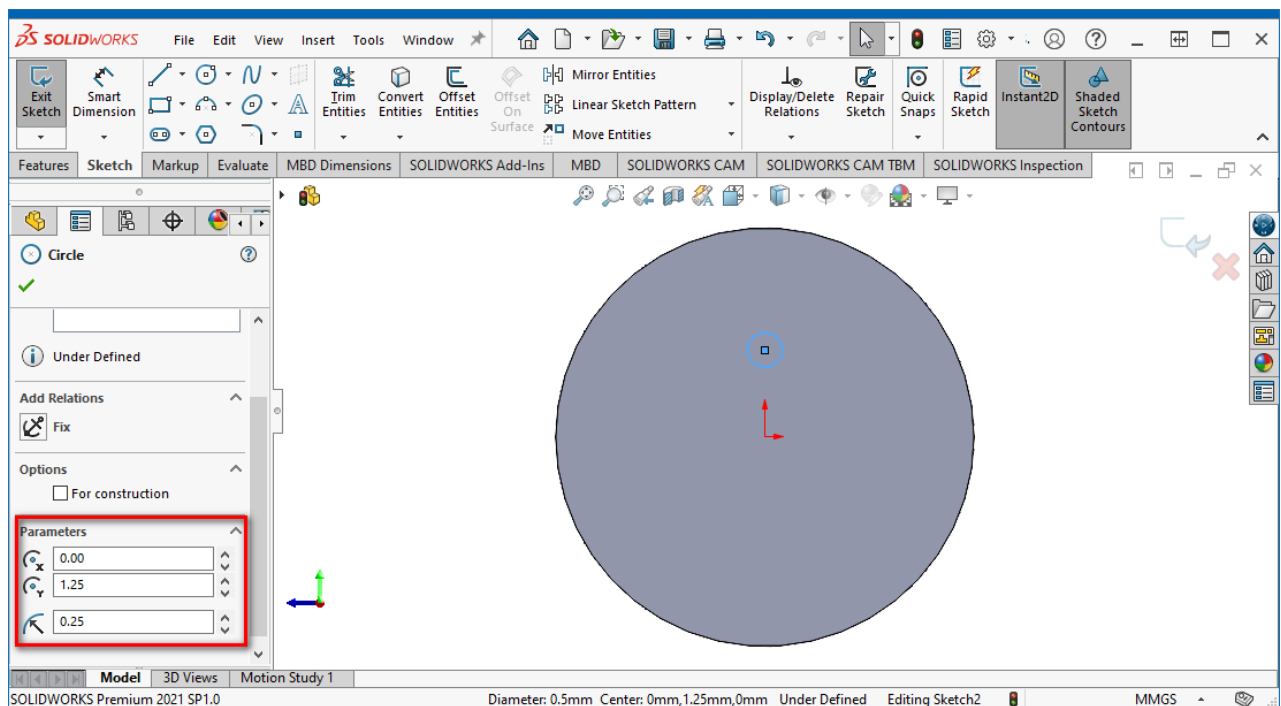



Рис. 9.17. Ескіз малого кола, на основі якого формуватиметься вивід конденсатора

Із нового ескізу робимо витягнутий елемент, для чого переходимо у *Features* → *Extruded Boss/Bases*, обираємо створене коло і у *Менеджері властивостей* задаємо довжину витягування $D1=5$ мм у напрямку 1. На цей раз не забудьте активувати опцію інвертування напрямку шляхом натискання піктограми . Інакше виводи конденсатора будуть фактично побудовані всередині корпусу, а не виходити із нього, як потрібно. Завершіть операцію натисканням ЛКМ на зеленій галочці в правому верхньому куті (рис. 9.18).

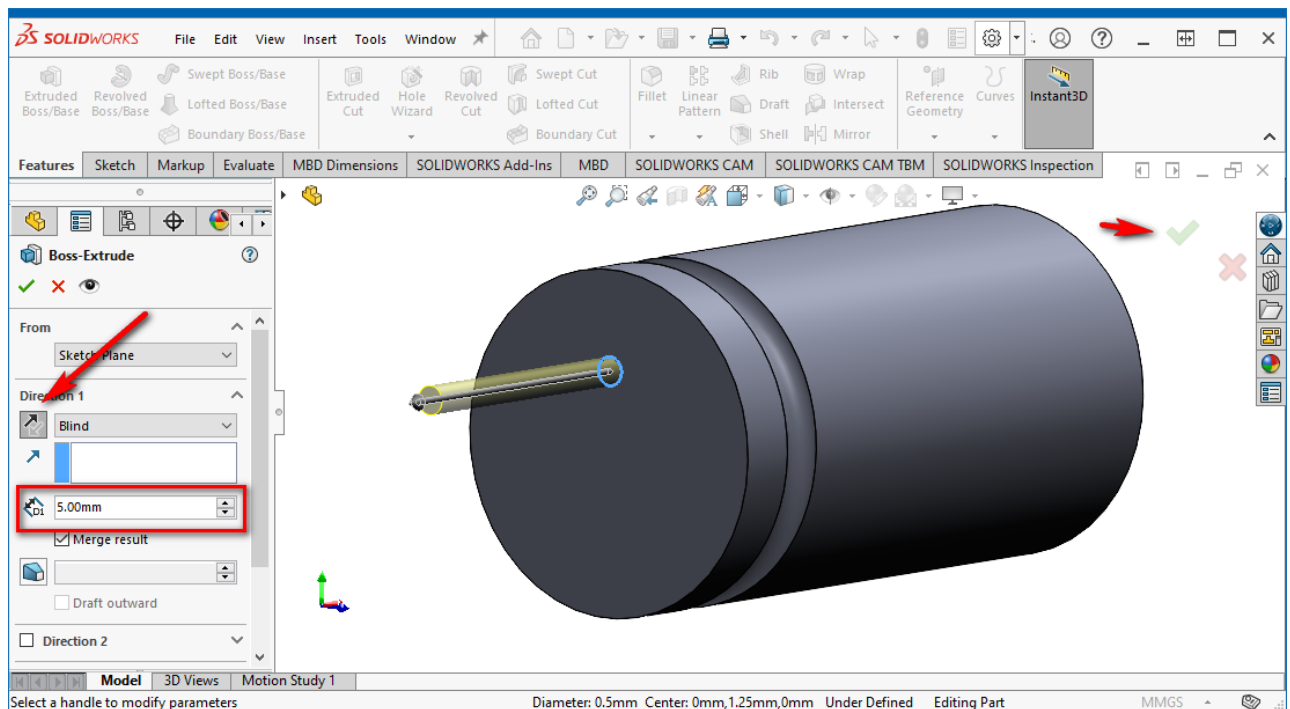

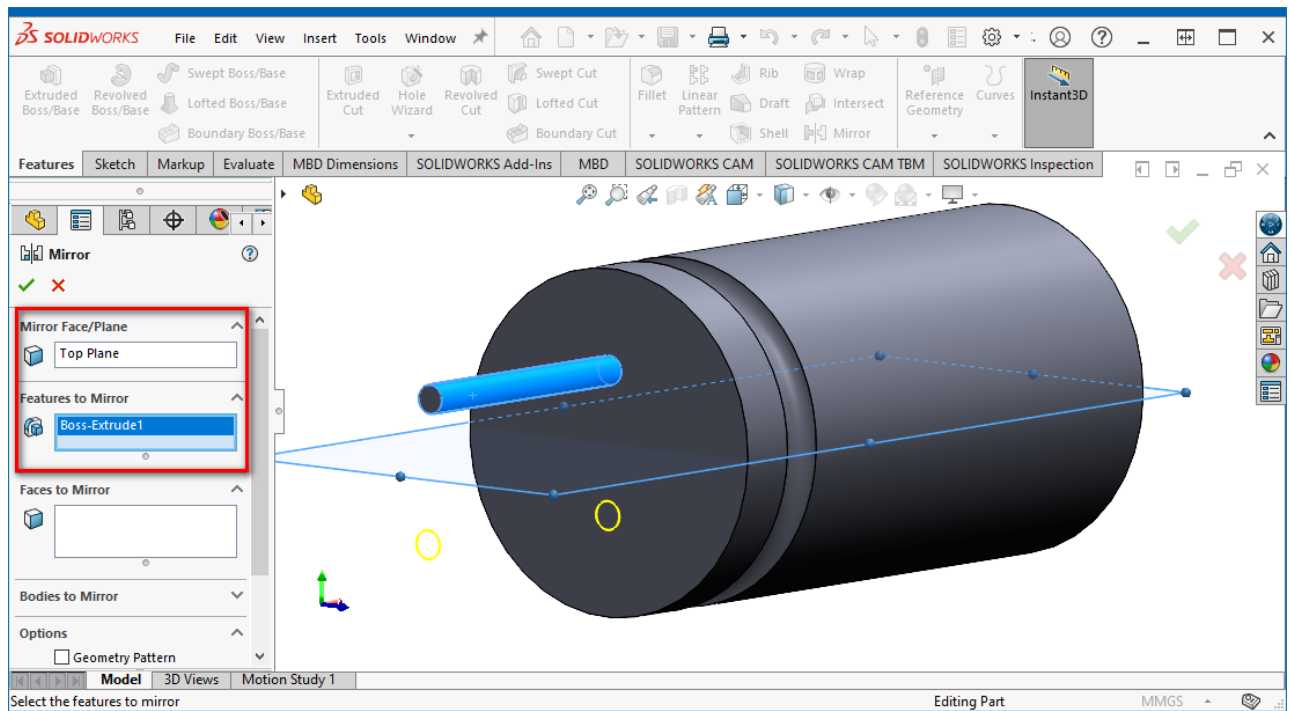



Рис. 9.18. Створення виводу конденсатора

Перша ніжка конденсатора готова, тепер виконаємо функцію віддзеркалення⁹⁶. Попередньо відобразимо допоміжну площину *Top*. Для цього у *FeatureManager Design Tree* клацніть ПКМ на рядку *Top Plane* і у спливаючому меню оберіть пункт *Show* (піктограма ).

Робимо віддзеркалення створеної ніжки конденсатора наступним чином. В панелі *Features* оберіть команду *Mirror*, після чого оберіть площину *Top Plane*, відносно якої відбуватиметься віддзеркалення, та ніжку конденсатора, яка має бути віддзеркаленою відносно площини *Top Plane*. Натисніть зелену галочку в правому верхньому куті для завершення процесу віддзеркалення (рис. 9.19).

⁹⁶ Насправді ми могли б одразу створити на попередньому кроці два виводи, якби в ескізі намалювали два кола. Такий підхід буде використано під час розробки наступної моделі. А спосіб із віддзеркаленням використовується більше з навчальною метою.

Рис. 9.19. Використання інструменту **Mirror** для віддзеркалення виводу

Далі зафарбуємо корпус конденсатора у чорний колір, а виводи – у світло-сірий. Натисніть піктограму  (**Edit Appearance**) у верхній частині робочого простору, після чого у лівій бічній панелі стане доступним інструмент **Selected Geometry** для вибору граней та елементів моделі, до яких можна застосувати кольори (рис. 9.20). Причому колір можна обрати як із деякого стандартного набору, так й з кольорової палітри, а можна навіть вказати у форматі RGB.

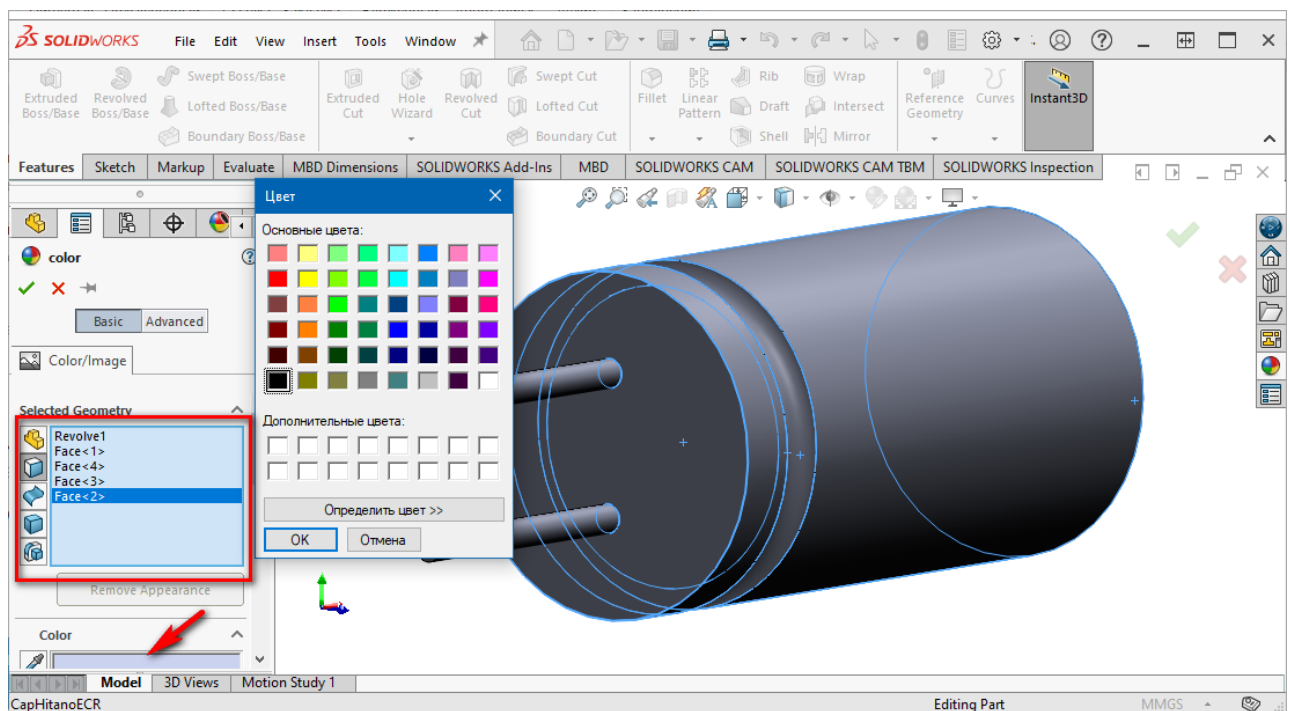


Рис. 9.20. Встановлення кольорів для елементів моделі

Тепер потрібно вказати ніжку конденсатора, яка має підключатись до від'ємного виводу джерела живлення. Зазвичай така ніжка позначається білою смугою на корпусі, на якій зображено знак мінус⁹⁷. Створити білу смугу можна різними способами, наприклад, використовуючи текстури, або додавши у модель окрему деталь, яка буде її імітувати, або зробити невеликий зріз на корпусі, який потім зафарбувати білим тощо. Наразі для імітації білої смуги ми використаємо підхід із створенням невеликого аксіального зрізу корпусу.

Спочатку встановіть вид з лівої сторони (*Left*). У *FeatureManager Design Tree* клацніть ПКМ на рядку *Right Plane* і у спливаючому меню оберіть пункт *Sketch*. Далі активуйте інструмент *Line* та намалуйте лінію довільної довжини (але так, щоб вона перетинала усю частину кола) на висоті 2,9 мм (рис. 9.21).

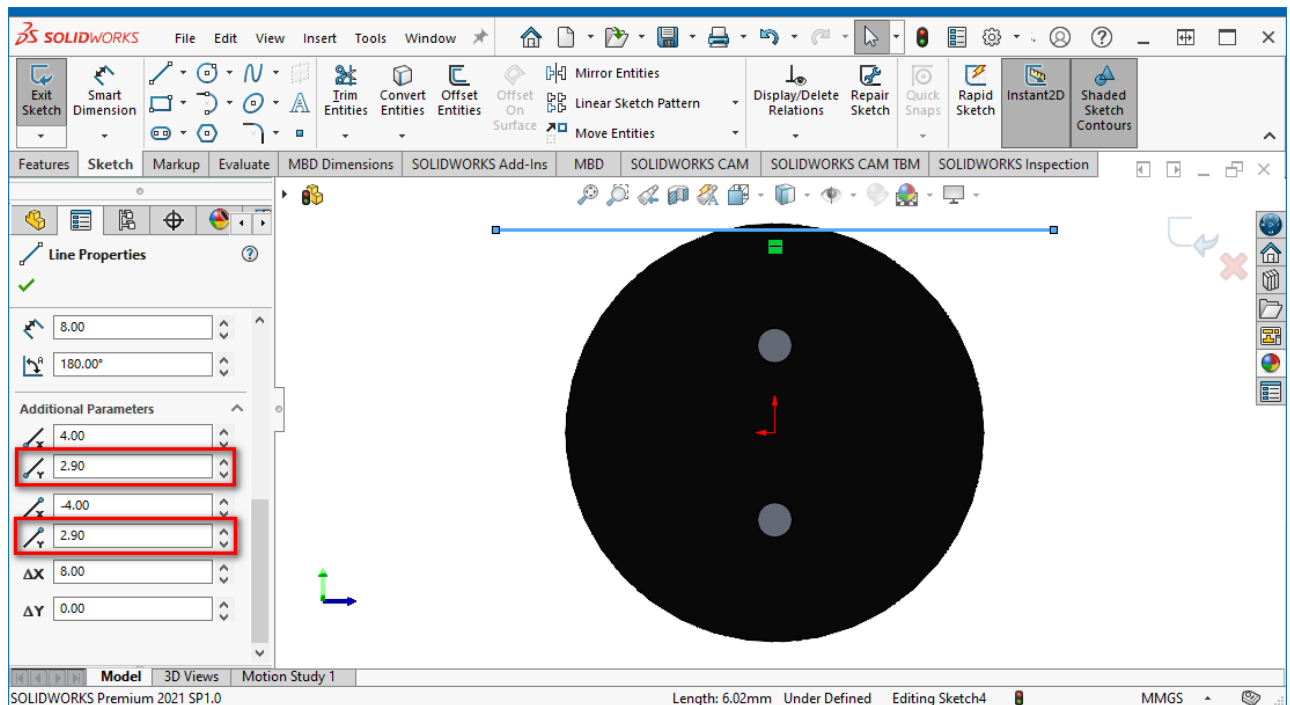


Рис. 9.21. Ескіз із лінією, яка буде використана для зрізу частини моделі

В панелі *Features* оберіть команду *Extruded Cut*, після чого виконайте необхідні налаштування у *Менеджері властивостей*. По-перше, обов'язково встановіть галочку для опції *Flip side to cut*, а, по-друге, для параметру *Direction 1* у випадному списку оберіть опцію *Through All* (усі інші параметри у *Менеджері властивостей* залиште без змін). На рис. 9.22 представлено, як буде виглядати робоча область в процесі відтинання частини моделі. Для завершення процедури натисніть зелену галочку у верхньому правому куті робочої області.

⁹⁷ Цікавий факт: на ПМ полярного конденсатора позначається не від'ємний вивід знаком мінус, а додатний вивід знаком плюс. Це необхідно враховувати під час монтажу такого конденсатора на ДП.

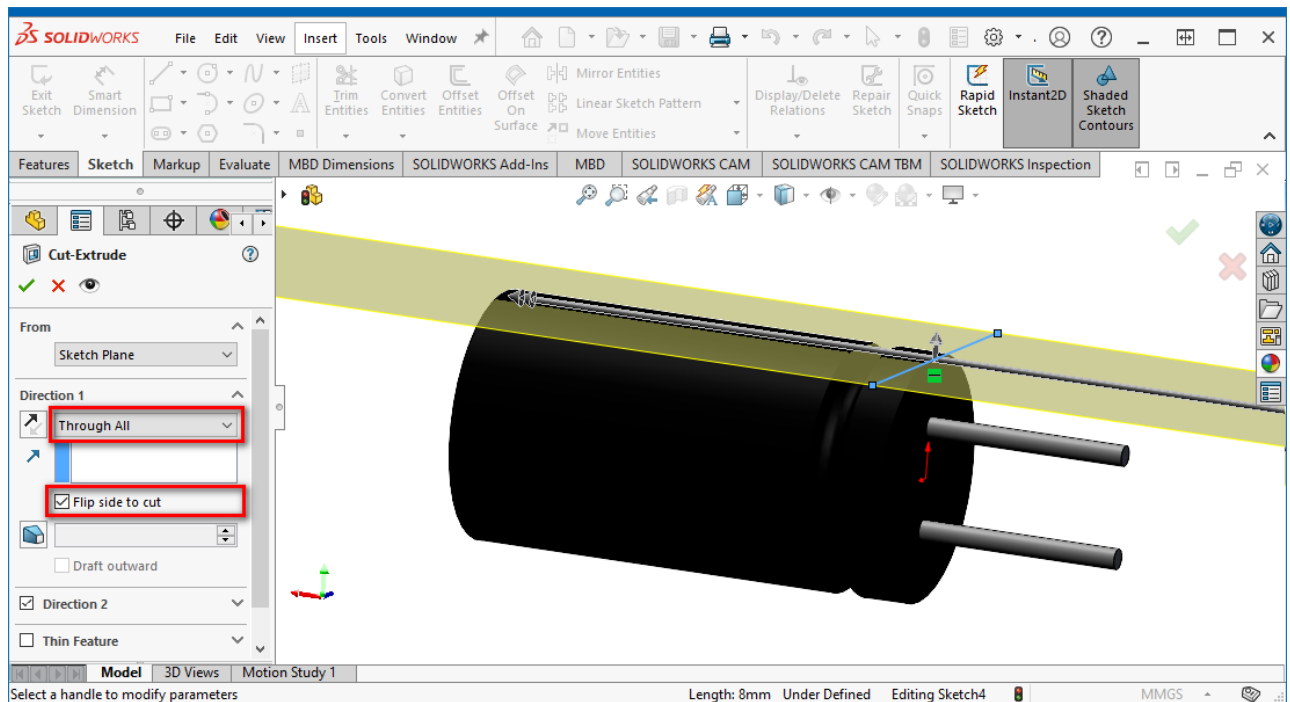


Рис. 9.22. Відтинання частини моделі

Також необхідно додати ще імітацію захисного клапану у верхній частині корпусу конденсатора. Для цього встановіть вид з правої сторони (*Right*). Клацніть ПКМ на верхній грані корпусу та у спливаючому меню оберіть *Sketch*. За допомогою інструменту *Circle* намалуйте коло радіусом 2,6 мм із центром в початку координат. Після цього у панелі *Features* оберіть команду *Extruded Cut* та зробіть виріз глибиною 0,1 мм всередину корпусу на основі ескізу кола (рис. 9.23).

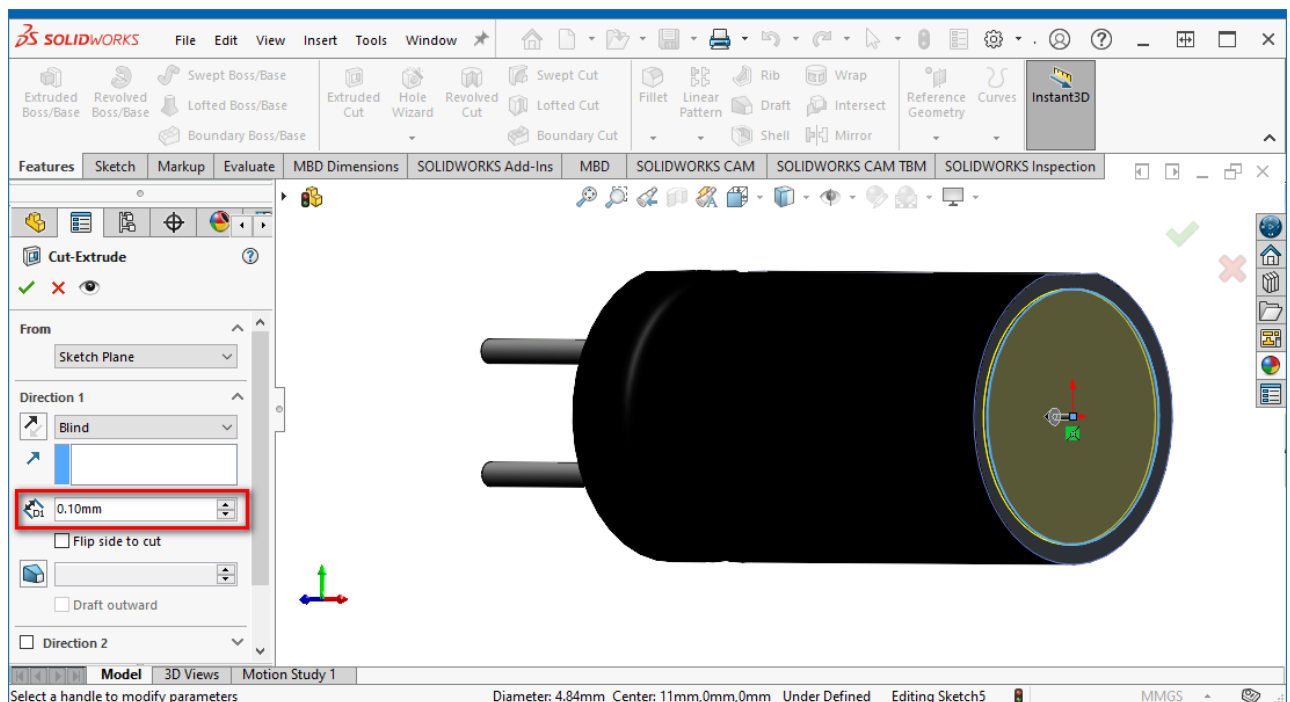


Рис. 9.23. Створення вирізу, що імітує захисний клапан на корпусі конденсатора

Насамкінець допрацюйте модель конденсатора, зафарбувавши білим боковий зріз на корпусі та світло-сірим – заглиблення у верхній частині корпусу. Кінцевий результат має виглядати так, як це показано на рис. 9.24.

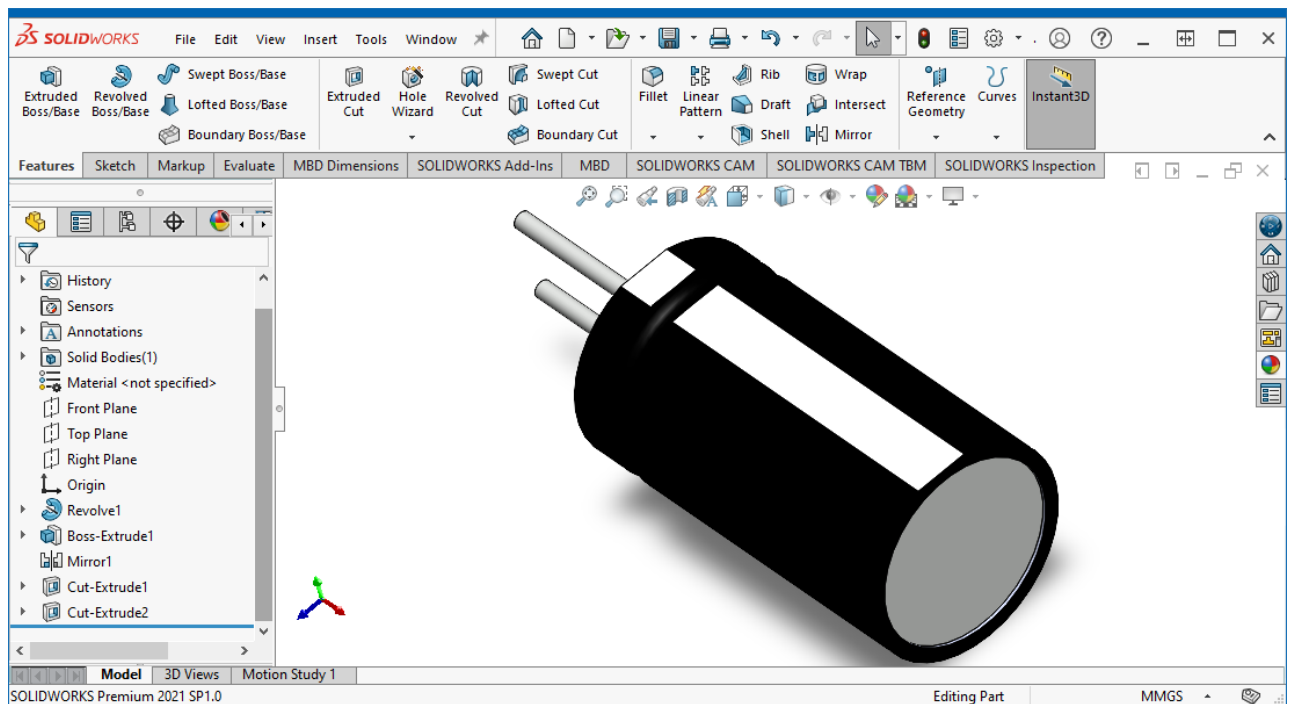


Рис. 9.24. Кінцевий вигляд моделі електролітичного конденсатора

Збережіть результат, задавши ім'я **CapHitanoECR**, як у форматі ***.sldprt**, так і ***.step**.

За бажанням можете самостійно розробити більш реалістичні моделі корпусу електролітичного конденсатора, наприклад, як на рис. 9.25.

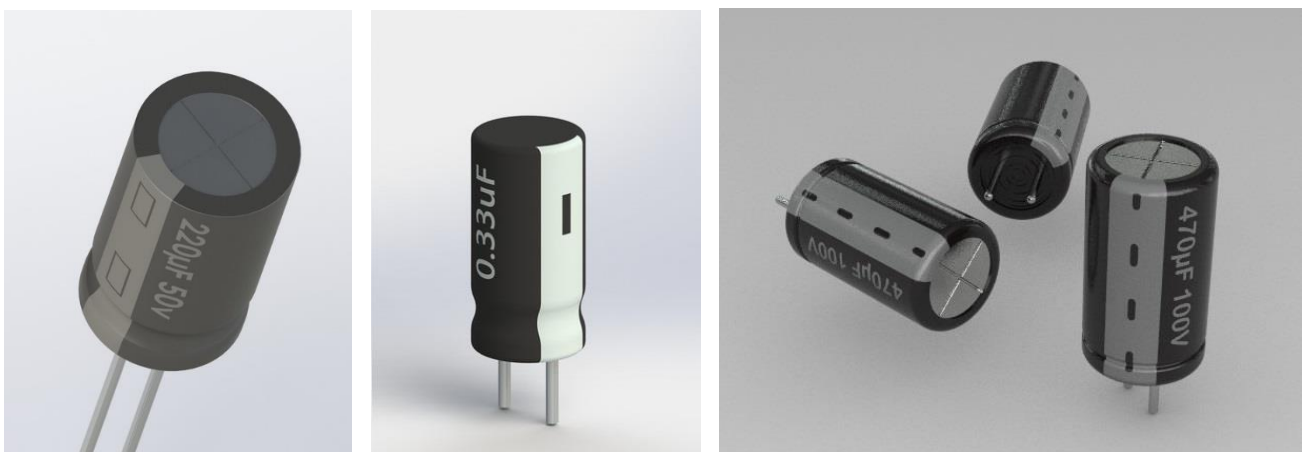


Рис. 9.25. Приклади тривимірних моделей електролітичних конденсаторів

Представлені на рис. 9.25 зображення конденсаторів взяті із сайту <https://grabcad.com/>. На ньому ви також зможете знайти і вихідні файли моделей у форматі ***.sldprt**, що може бути корисним для ознайомлення та аналізу.

3.3. Проектування моделі котушки індуктивності

Створіть нову заготовку моделі. У лівій бічній панелі на вкладці **FeatureManager Design Tree** клацніть ПКМ на рядку **Front Plane** і у спливаючому меню оберіть пункт **Sketch**. Після цього на панелі інструментів активуйте інструмент **Line**.

Встановіть курсор в точку початку координат і намалюйте пряму вертикальну лінію від початку координат у додатному напрямку довжиною 9,8 мм (рис. 9.26). Поки можете не завершувати формування ескізу натисканням відповідної піктограми, оскільки намальована лінія є частиною більш складного контуру, на основі якого надалі буде створена об'ємна форма корпусу котушки індуктивності.

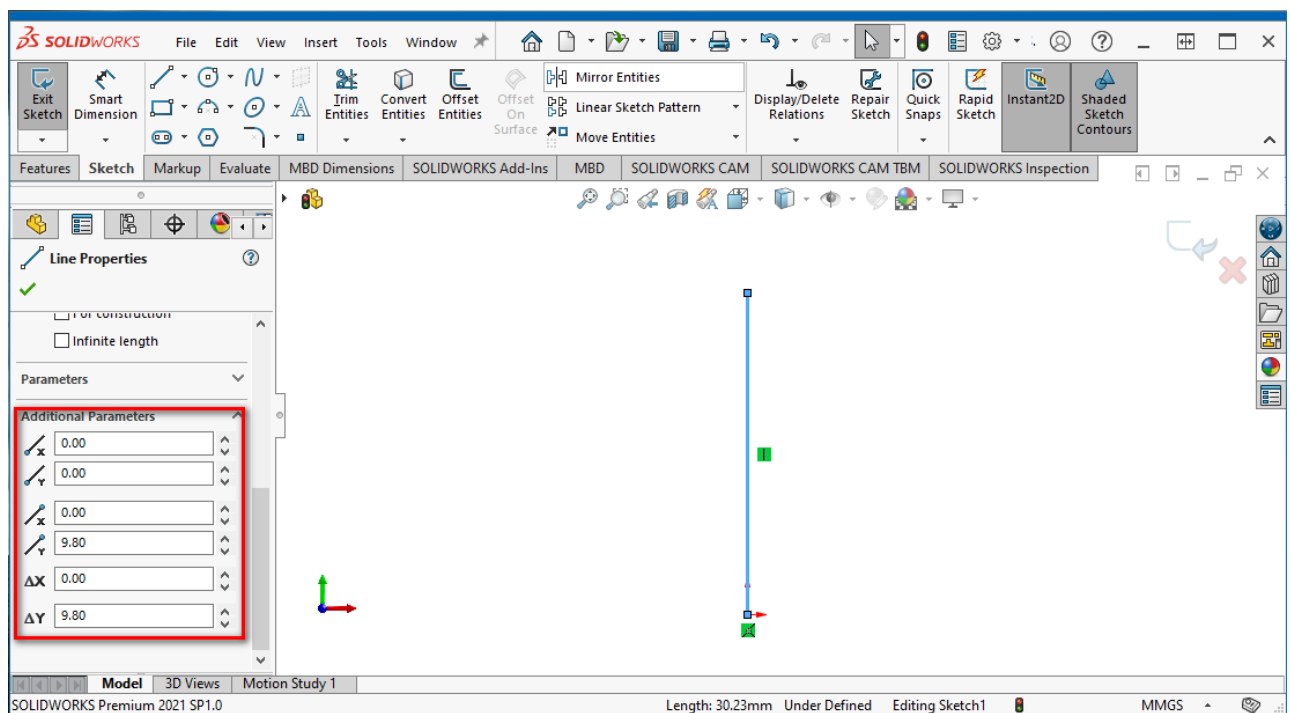


Рис. 9.26. Ескіз лінії

Далі будемо складний криволінійний контур елемента за допомогою сплайну, що проходить через 4 точки. Для цього у панелі **Sketch** активуйте інструмент **Spline** та намалюйте криву приблизно так, як показано на рис. 9.27. Після чого у **Менеджері властивостей** відредагуйте координати кожної точки наступним чином: перша – (0; 9,8), друга – (2,5; 8,7), третя – (3; 6,5), четверта – (1,2; 0). Насамкінець з'єднайте лінією початок координат та четверту точку, щоб вийшов замкнений контур. Виконавши усі описані дії, завершіть формування ескізу натисканням відповідної піктограми.

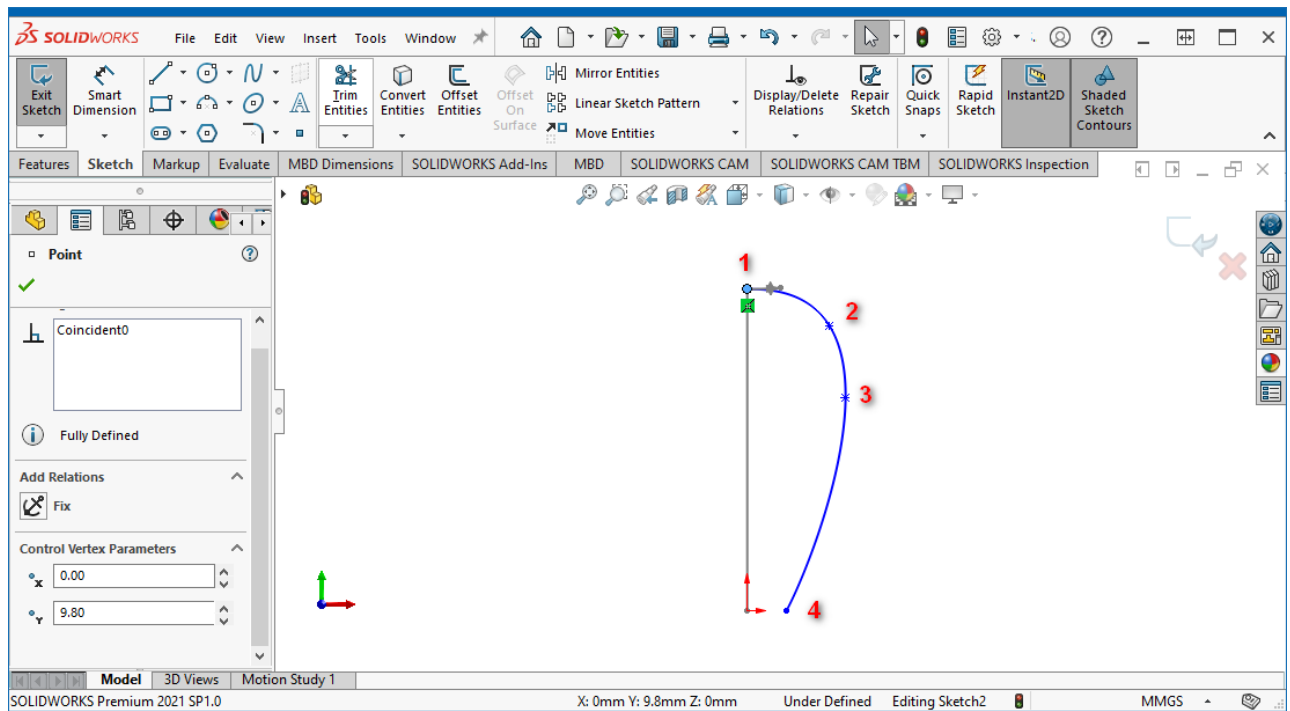


Рис. 9.27. Ескіз, який визначає контур корпусу котушки індуктивності

Наступними кроком буде створення витягнутого елемента з ескізу. Спочатку відкрийте панель **Features** та оберіть команду **Extruded Boss/Base**. Після цього у **Менеджері властивостей** задаємо довжину моделі $D1=7,00$ мм (у напрямку 1). Завершіть операцію натисканням ЛКМ на зеленій галочці у правому верхньому куті робочої області (рис. 9.28).

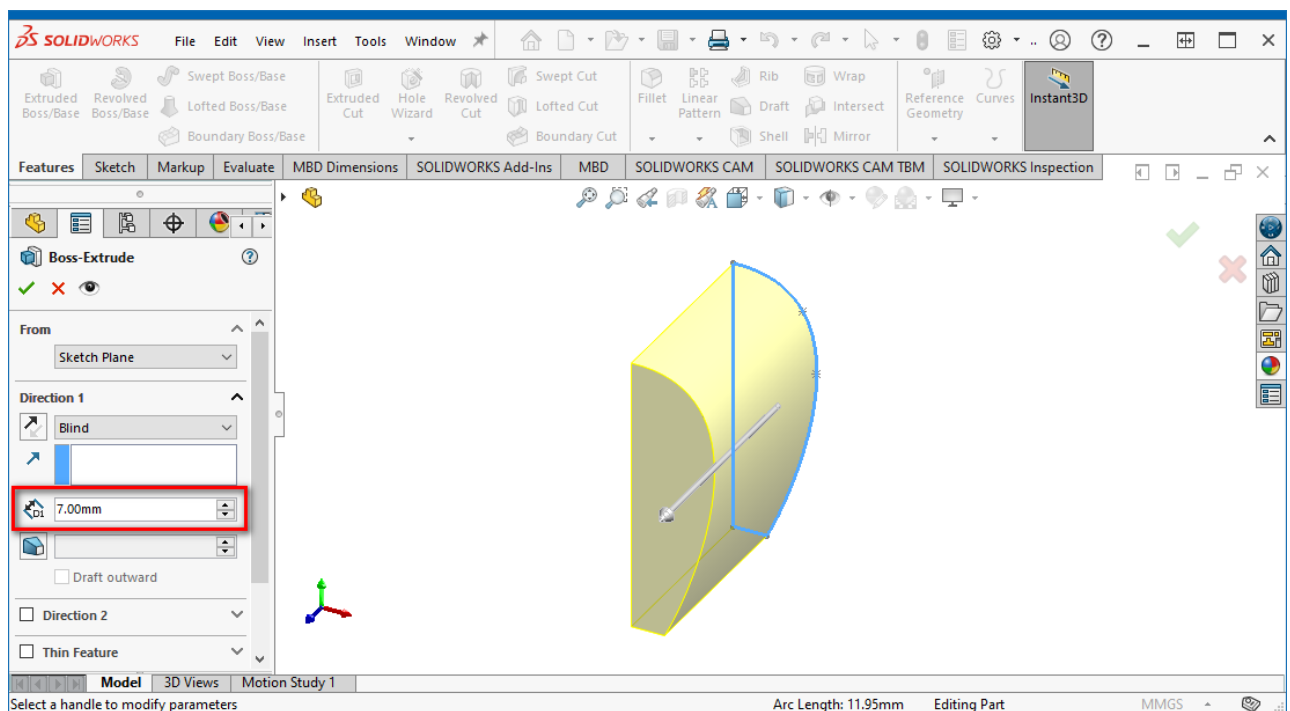


Рис. 9.28. Витягнутий елемент

Таким чином була створена лише половина корпусу. Виконаємо її віддзеркалення, щоб отримати повну модель. Для початку поверніть модель таким чином, щоб плоска сторона моделі була спереду. В панелі **Features** оберіть команду **Mirror**, після чого оберіть плоску грань, відносно якої відбуватиметься віддзеркалення, та елементи, які необхідно віддзеркалити, – це бічна грань (рис. 9.29). Завершіть операцію відповідним чином.

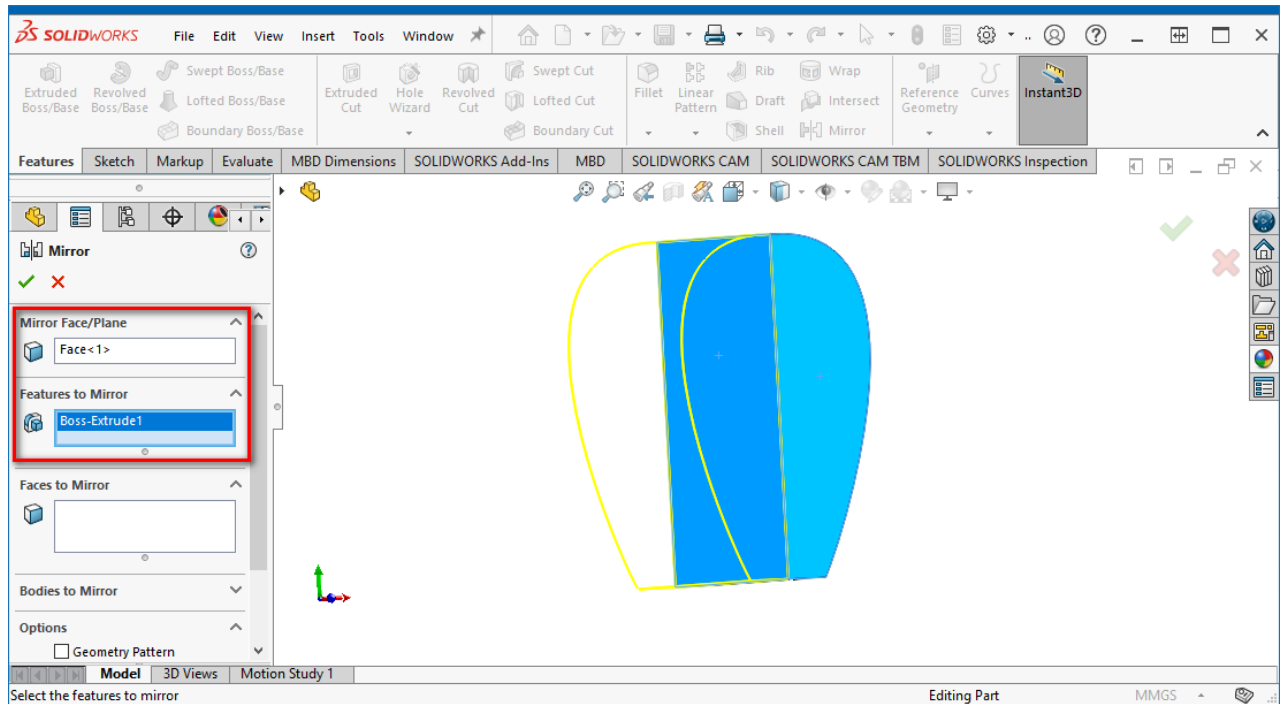


Рис. 9.29. Віддзеркалення моделі

Насамкінець виконаємо ще деякі модифікації форми створеної моделі корпусу. Встановіть спочатку вид справа. Потім на вкладці **FeatureManager Design Tree** клацніть ПКМ на рядку **Right Plane** і у спливаючому меню оберіть пункт **Sketch**. Після цього на панелі інструментів активуйте інструмент **Spline**. Намалуйте криву за п'ятьма точками, що мають наступні координатами: 1 – (-7; 7), 2 – (-5,5; 9,8), 3 – (-3,5; 10,5), 4 – (-1,5; 9,8), 5 – (0; 7). Завершивши створення сплайну, оберіть інструмент **Line** та з'єднайте 1, 6, 7 і 5 точки лініями таким чином, щоб вийшов замкнений контур. За координатою X точка 6 матиме таке саме значення, як і 1, а точка 7 – як 5. За висотою (координата Y) потрібно зробити так, щоб точка 6 і 7 просто були вищі за точку 3. Після цього, використовуючи все той же інструмент **Line**, створіть замкнену трапецію у нижній частині корпусу. Координати точок вершин трапеції наступні: 8 – (-5,8; 0), 9 – (-4,5; 1,8), 10 – (-2,5; 1,8), 11 – (-1,2; 0). Результат представлено на рис. 9.30. Завершіть операцію створення ескізу натисканням відповідної піктограми.

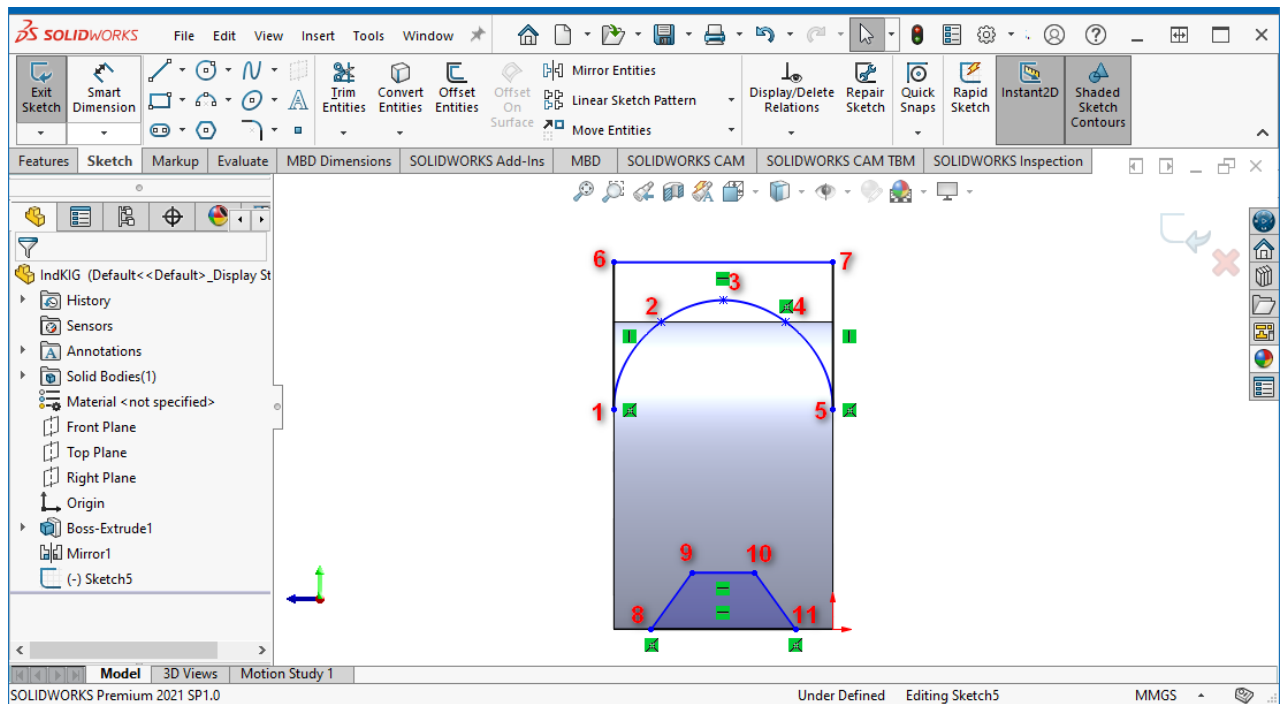


Рис. 9.30. Ескізи, які використовуватимуться для створення вирізів

Далі потрібно виконати виріз за створеними траєкторіями. Для цього у панелі **Features** оберіть команду **Extruded Cut**⁹⁸. У **Менеджері властивостей** задаємо довжину вирізу D1=7 мм у напрямку 1 та напрямку 2 (рис. 9.31). Завершіть операцію натисканням ЛКМ на зеленій галочці в правому верхньому куті.

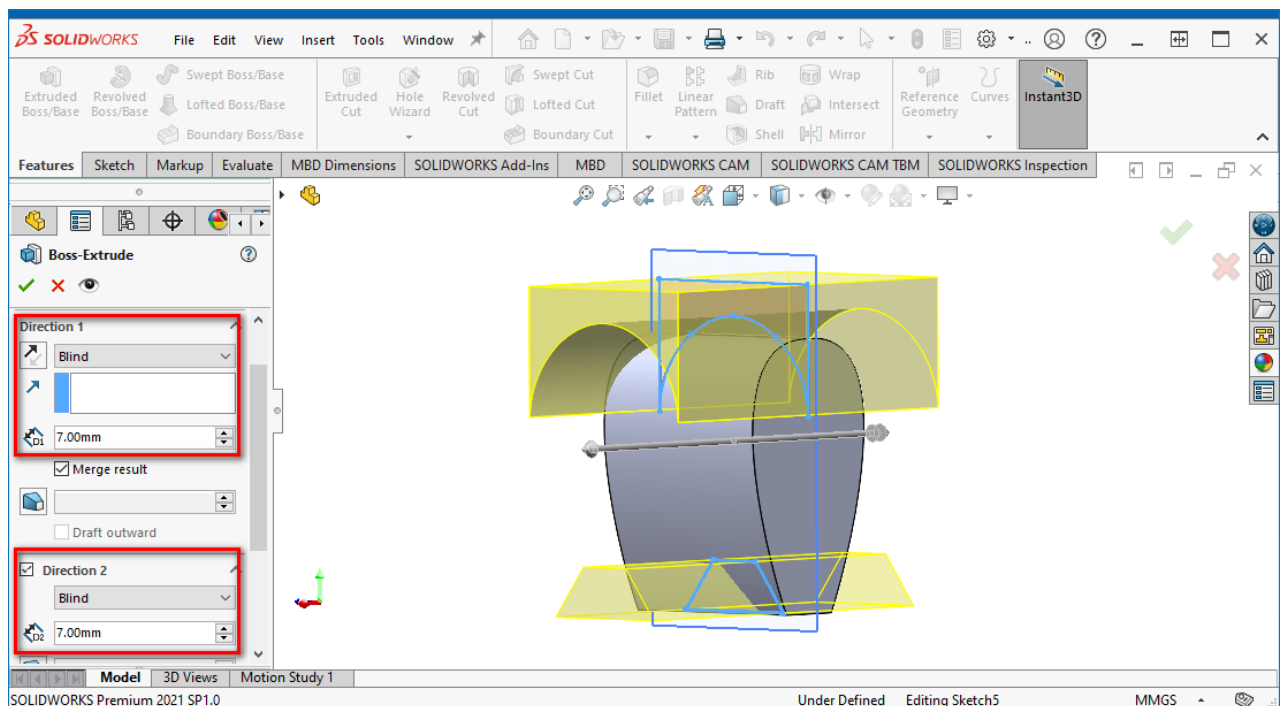


Рис. 9.31. Створення вирізів

⁹⁸ Якщо контури для вирізу не будуть розпізнані автоматично, то для додавання контурів необхідно у **Менеджері властивостей** скористатись формою **Selected Contours**.

Далі нам потрібно додати виводи котушки індуктивності. Спочатку поверніть модель таким чином, щоб нижня її грань була направлена на вас. На вкладці **FeatureManager Design Tree** клацніть ПКМ на рядку **Top Plane** і у спливаючому меню оберіть пункт **Sketch**. Активуйте інструмент **Circle** та побудуйте два кола (рис. 9.32) радіусом 0,3 мм з наступними координатами: перше – (0; -0,5), друге – (0; -6,5).

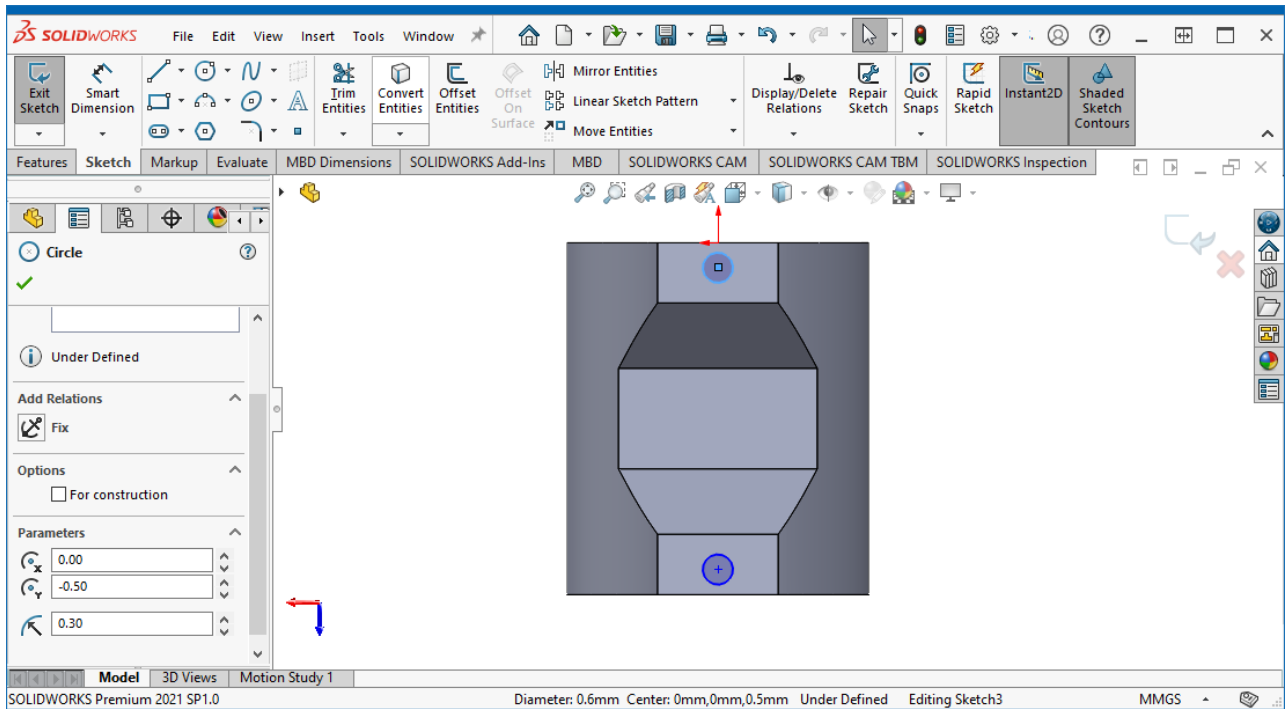


Рис. 9.32. Ескізи кіл, на основі яких будуть створені виводи

Тепер зі створених кіл сформуємо виводи котушки індуктивності (рис. 9.33).

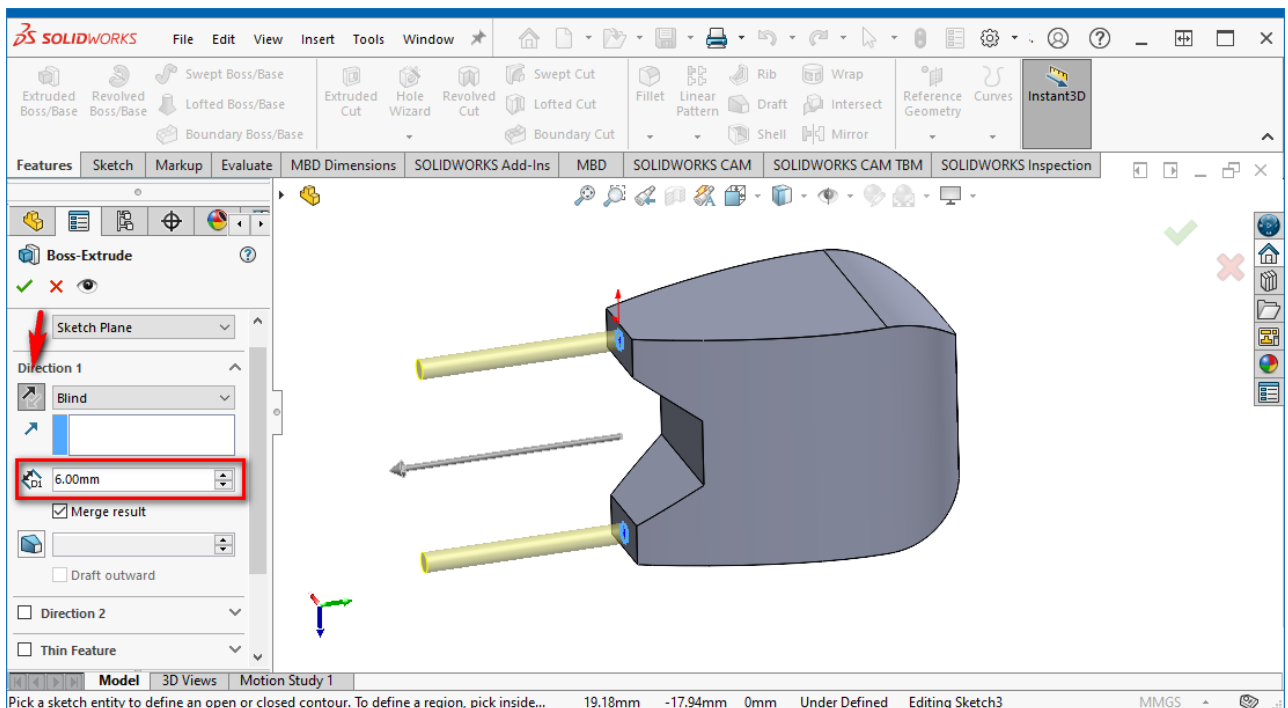


Рис. 9.33. Формування виводів котушки індуктивності

Відкрийте панель **Features** та оберіть команду **Extruded Boss/Base**. Виділіть обидва кола та у **Менеджері властивостей** задайте довжину D1=6,00 мм (у напрямку 1) та активуйте піктограму інверсії напрямку. Вигляд робочої області під час створення виводів представлено на рис. 9.33.

Насамкінець зафарбуйте корпус коричневим кольором, а виводи – світло-сірим. Кінцевий результат представлено на рис. 9.34.

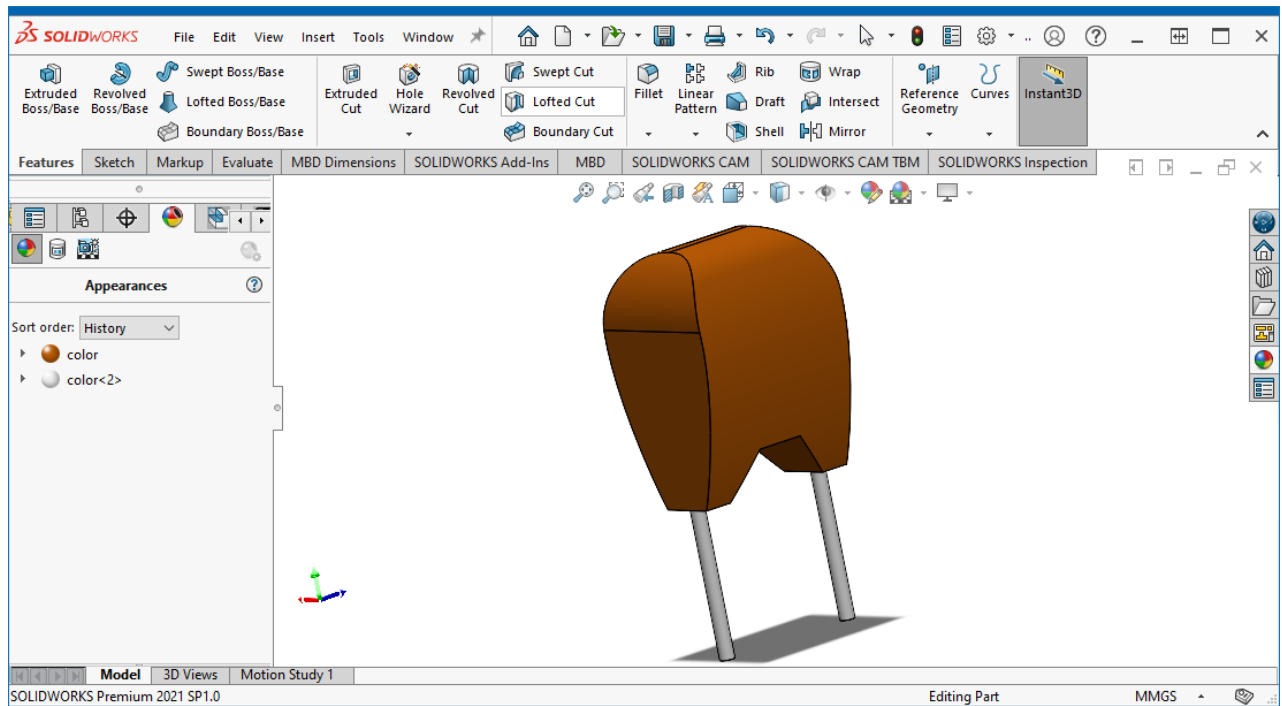


Рис. 9.34. Створена модель котушки індуктивності КІГ

Збережіть результат, задавши ім'я **IndKIG**, як у форматі ***.sldprt**, так і ***.step**.

3.4. Проєктування моделі мікросхеми у корпусі SOT-23-5

Для посадкового місця мікросхеми в корпусі SOT-23-5 побудуємо тривимірну модель на основі геометричних розмірів, які наведені на рис. 6.10 у комп'ютерному практикумі №6.

Створіть нову заготовку моделі. У лівій бічній панелі на вкладці **FeatureManager Design Tree** клацніть ПКМ на рядку **Front Plane** і у спливаючому меню оберіть пункт **Sketch**. Після цього на панелі інструментів активуйте інструмент **Corner Rectangle**. Встановіть курсор у точку початку координат і намалюйте довільний прямокутник. Після цього у **Менеджері властивостей** встановіть наступні точні параметри (рис. 9.35): висота (вздовж осі Y) – 2,9 мм, ширина (вздовж осі X) – 0,8 мм.

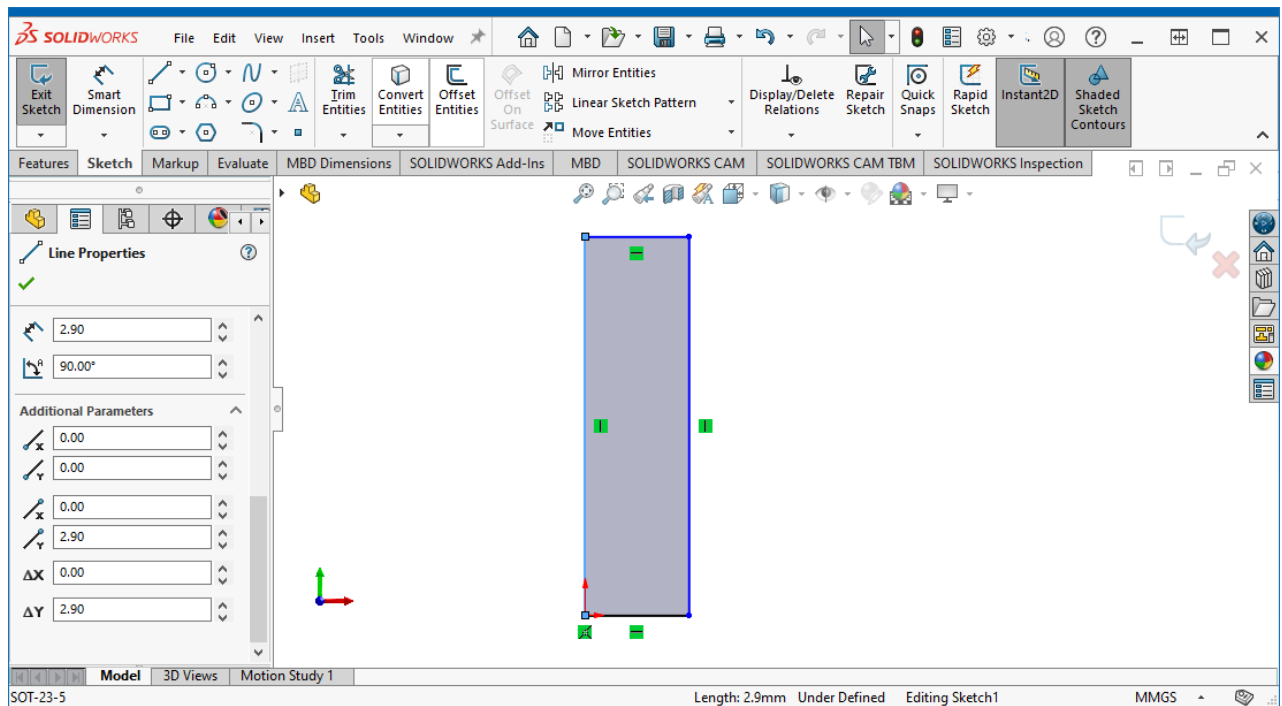


Рис. 9.35. Ескіз прямокутника

Наступними кроком буде створення екструдованого прямокутника з ескізу (рис. 9.36). Відкрийте панель **Features** та оберіть команду **Extruded Boss/Base**. У **Менеджері властивостей** задаємо довжину моделі D1=1,00 мм (у напрямку 1).

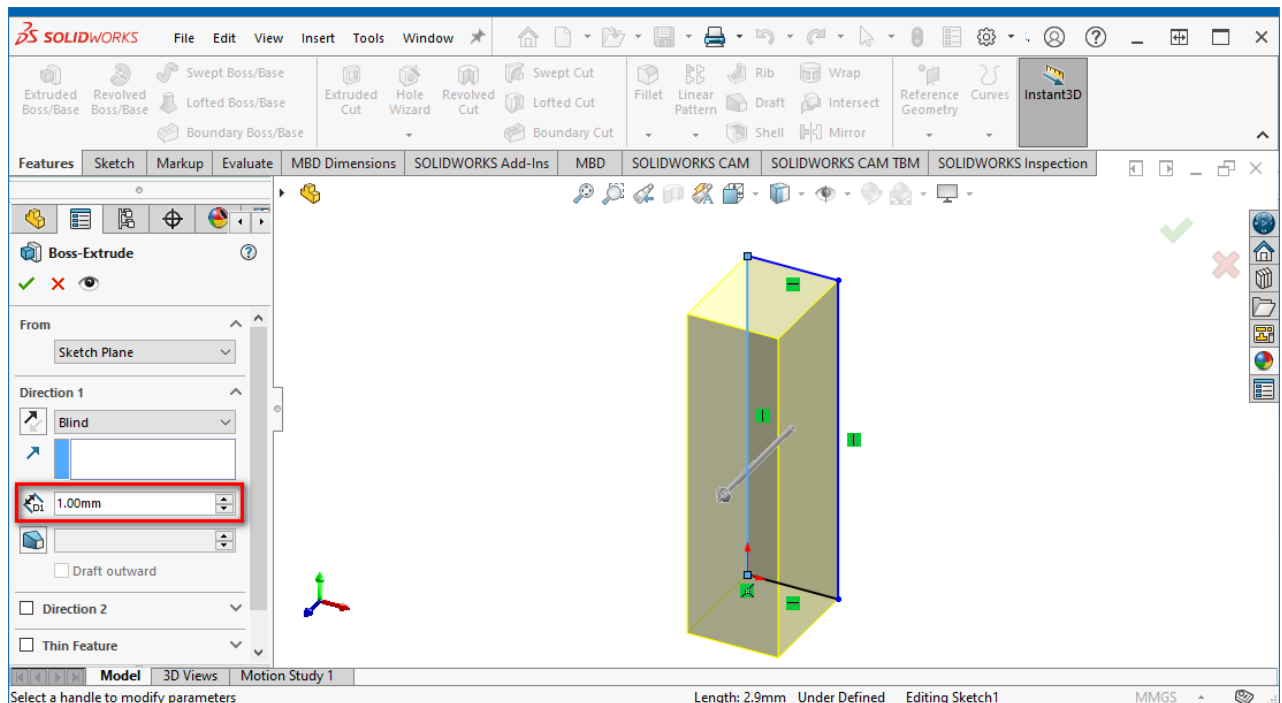


Рис. 9.36. Створення екструдованого прямокутника

Поверніть модель таким чином, щоб відображався вид зліва (**Left**). Натисніть ПКМ на лівій грані моделі та у спливаючому меню оберіть **Sketch** (рис. 9.37).

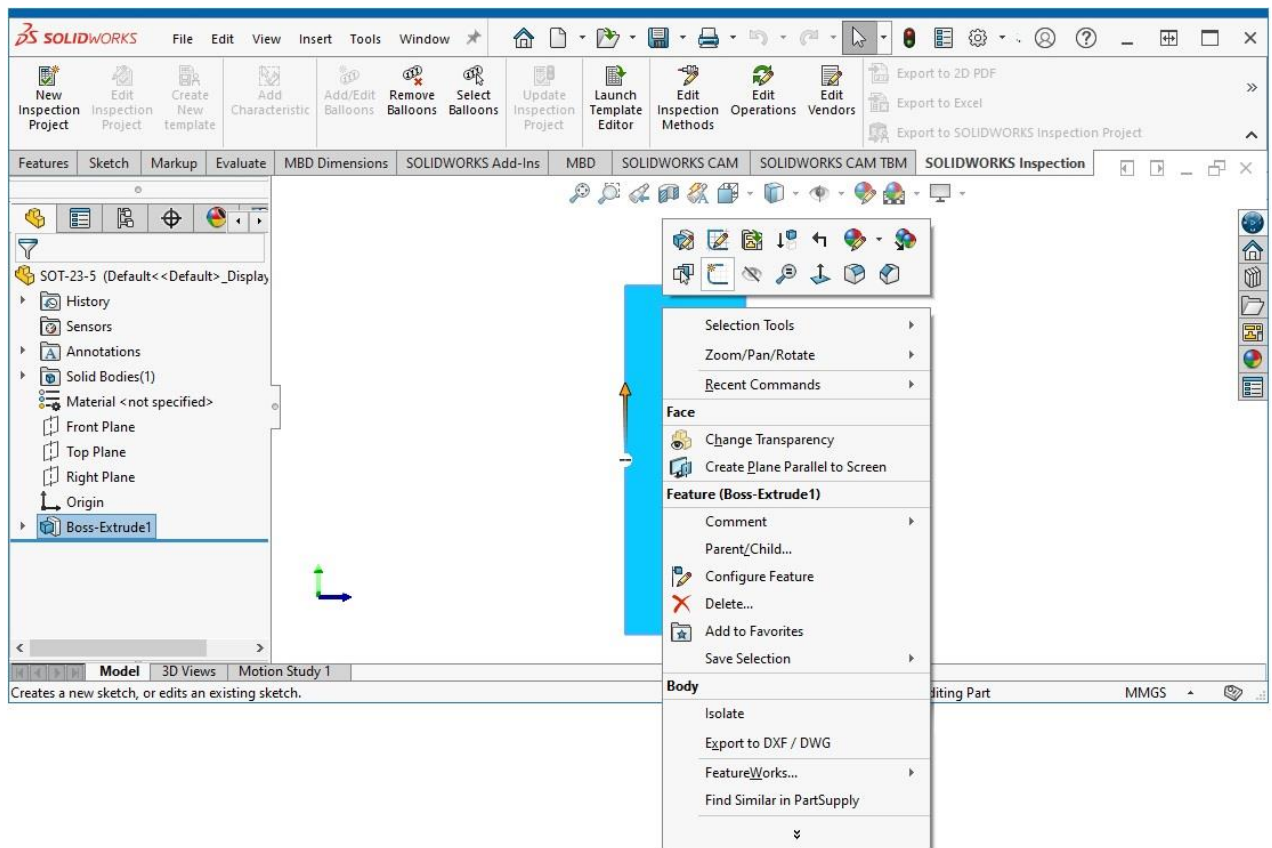


Рис. 9.37. Створення нового ескізу на лівій грані екструдованого прямокутника

Після цього на панелі інструментів активуйте інструмент **Corner Rectangle** та намалуйте прямокутник із вершинами у наступних координатах (рис. 9.38): 1 – (0,6; 2,615), 2 – (0,6; 2,185), 3 – (0,75; 2,185), 4 – (0,75; 2,615). Завершіть формування скетчу натисканням відповідної піктограми.

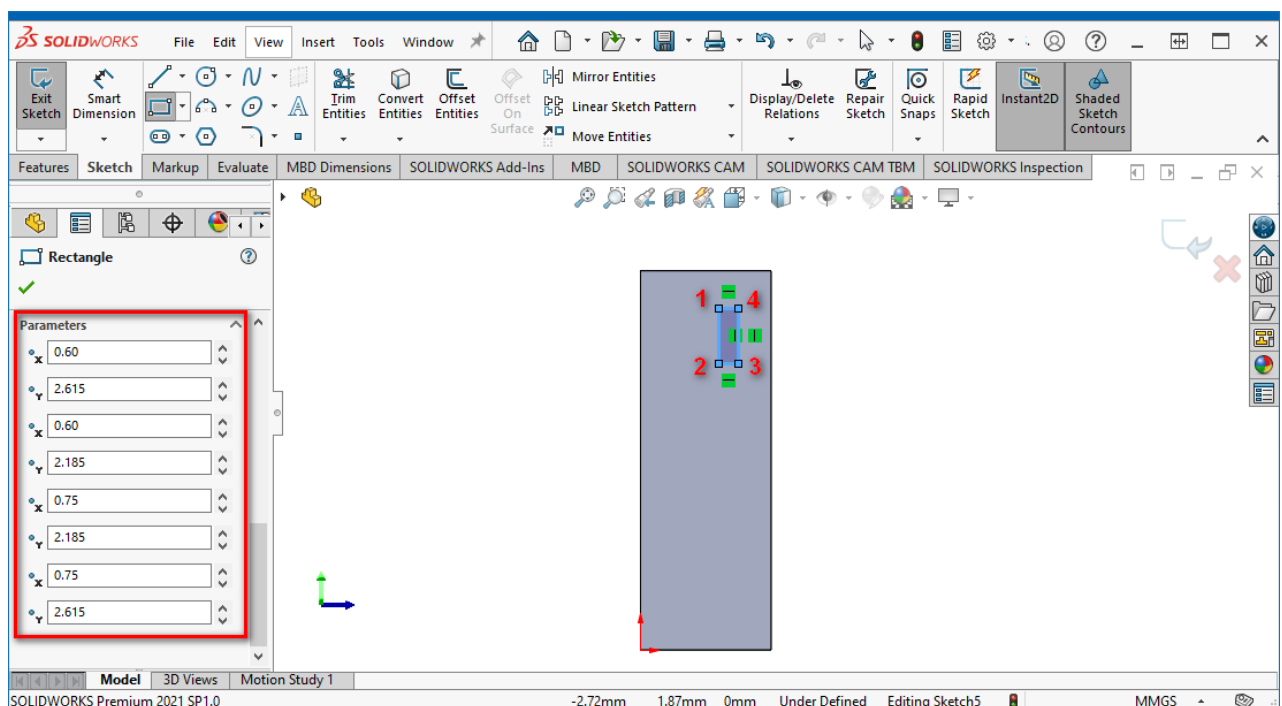



Рис. 9.38. Створення малого прямокутника на лівій грані екструдованого прямокутника

Тепер нам необхідно створити допоміжну площину, яка надалі використовуватиметься для побудови траєкторії, що визначатиме форму виводу мікросхеми. У панелі **Features** активуйте команду **Reference Geometry** → **Plane**. Після цього оберіть площину **Top Plane** як опорну, паралельно якій буде розташовуватись допоміжна площина. Для того, щоб її обрати, натисніть у верхньому лівому куті робочої області піктограму , після чого у випадному переліку оберіть **Top Plane** (якщо все зроблено правильно, то у **Менеджері властивостей** в області **First Reference** має відобразитись напис **Top Plane** (рис. 9.39). Далі задайте відстань $D1=2,4$ мм та завершіть формування площини натисканням зеленої галочки у верхньому правому куті робочої області (усі інші параметри у **Менеджері властивостей** залиште без змін).

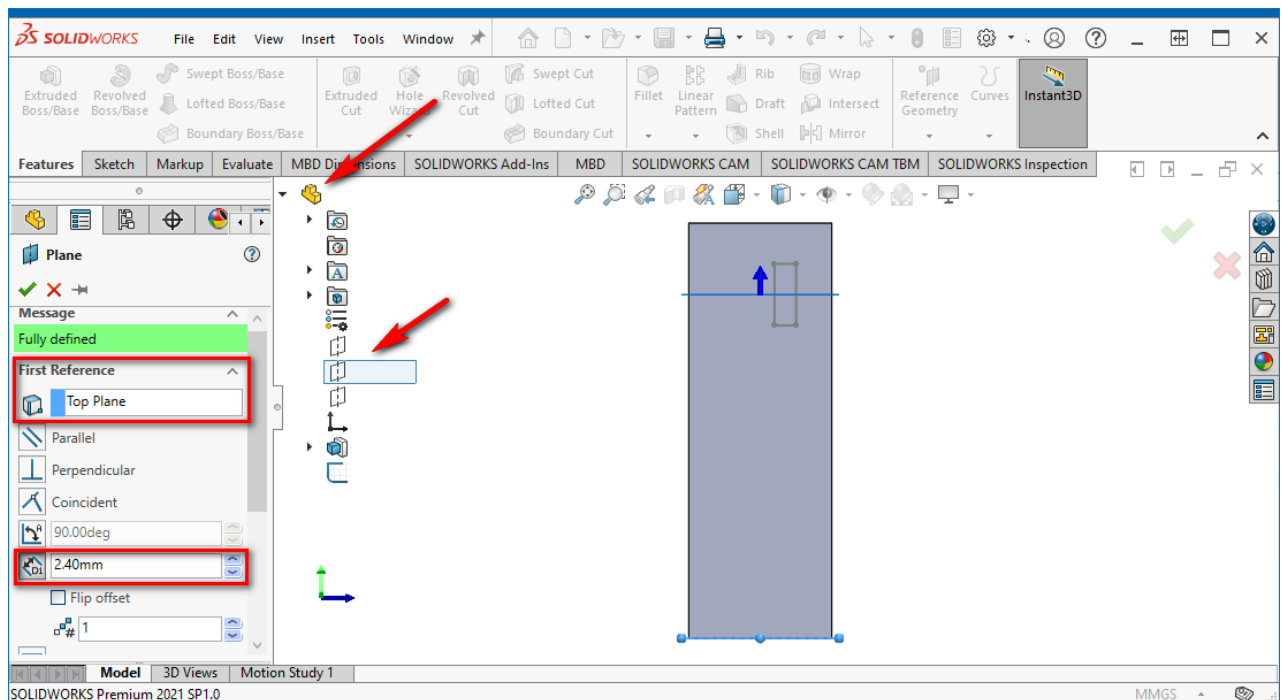


Рис. 9.39. Створення допоміжної площини

Встановіть вид відображення моделі знизу (**Bottom**). У лівій бічній панелі на вкладці **FeatureManager Design Tree** клацніть ПКМ на рядку **Plane1** (так за замовчуванням називається створена нами допоміжна площина) і у спливаючому меню оберіть пункт **Sketch**. Після цього на панелі інструментів активуйте інструмент **Point** та встановіть шість точок (рис. 9.40) з наступними координатами: 1 – (0; -0,675), 2 – (-0,175; -0,675), 3 – (-0,275; -0,575), 4 – (-0,275; -0,125), 5 – (-0,375; -0,025), 6 – (-0,63; -0,025).

На цьому етапі можете не завершувати формування ескізу, оскільки він повинен містити не самі точки, а лінії та криві, які їх будуть з'єднувати (хоча, якщо

ви навіть зараз і збережете ескіз тільки із точками, то згодом зможете без проблем його відредагувати).

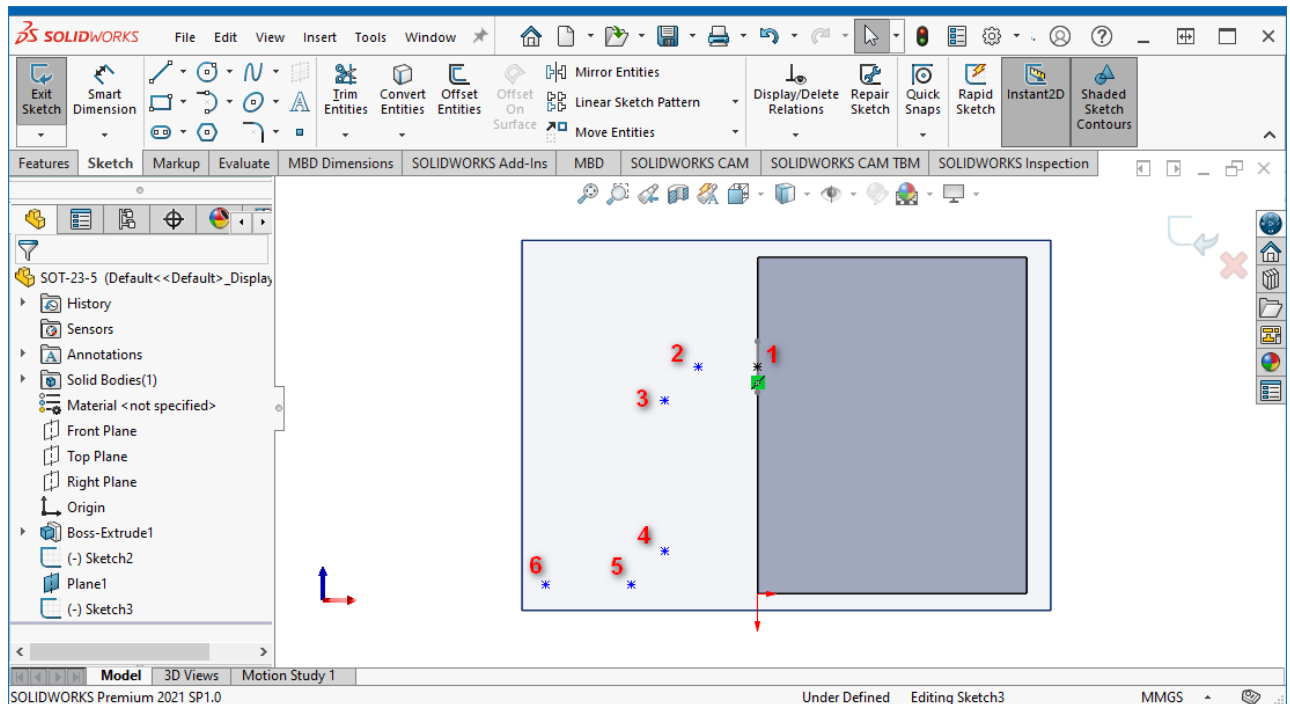


Рис. 9.40. Створення точок, які визначають траєкторію виводу

Далі з'єднайте ці точки лініями (*Line*) і дугами (*Tangent Arc*) так, як це показано на рис. 9.41. Тільки після цього завершіть формування ескізу натисканням відповідної піктограми.

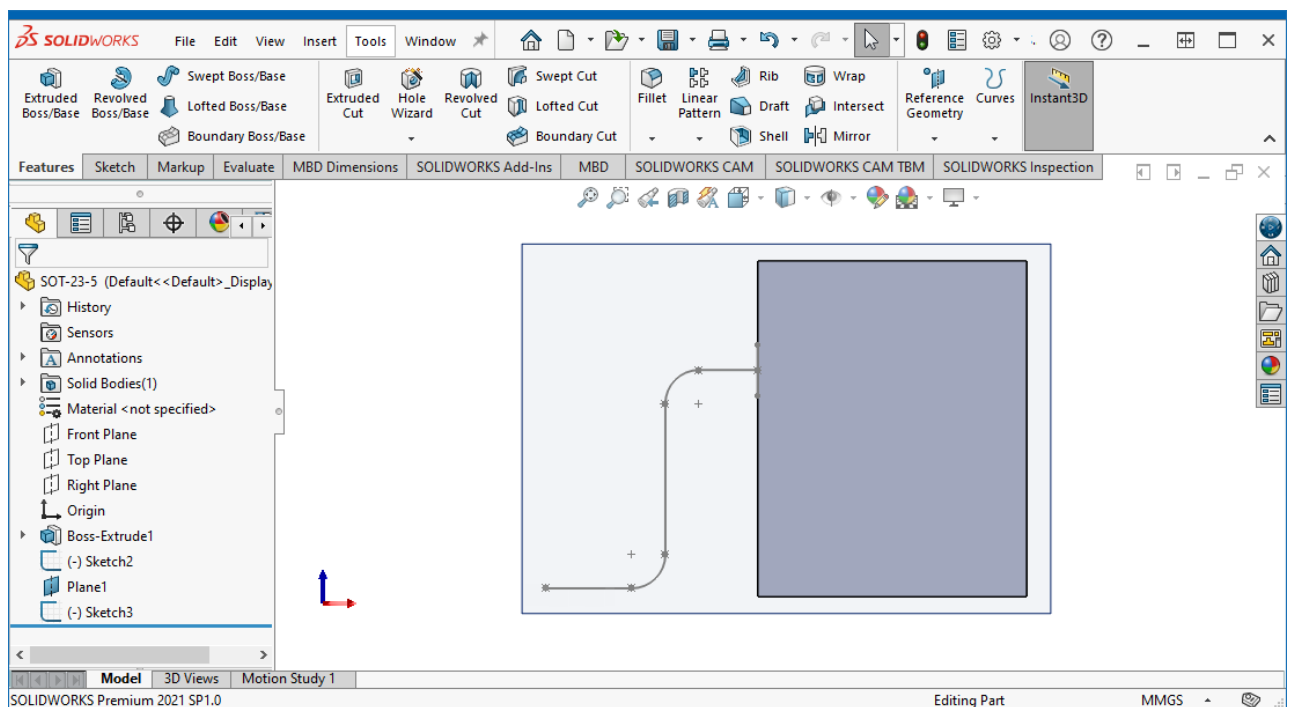


Рис. 9.41. Точки ескізу, що з'єднані лініями

Наступним кроком створимо об'ємну фігуру за траєкторією. Відкрийте панель **Features** та оберіть команду **Swept Boss/Base**. Після вибору команди вкажіть профіль (це має бути прямокутник на лівій боковій грані корпусу мікросхеми) та маршрут (це мають бути усі лінії та дуги, які були побудовані на допоміжній площині). Результат представлено на рис. 9.42.

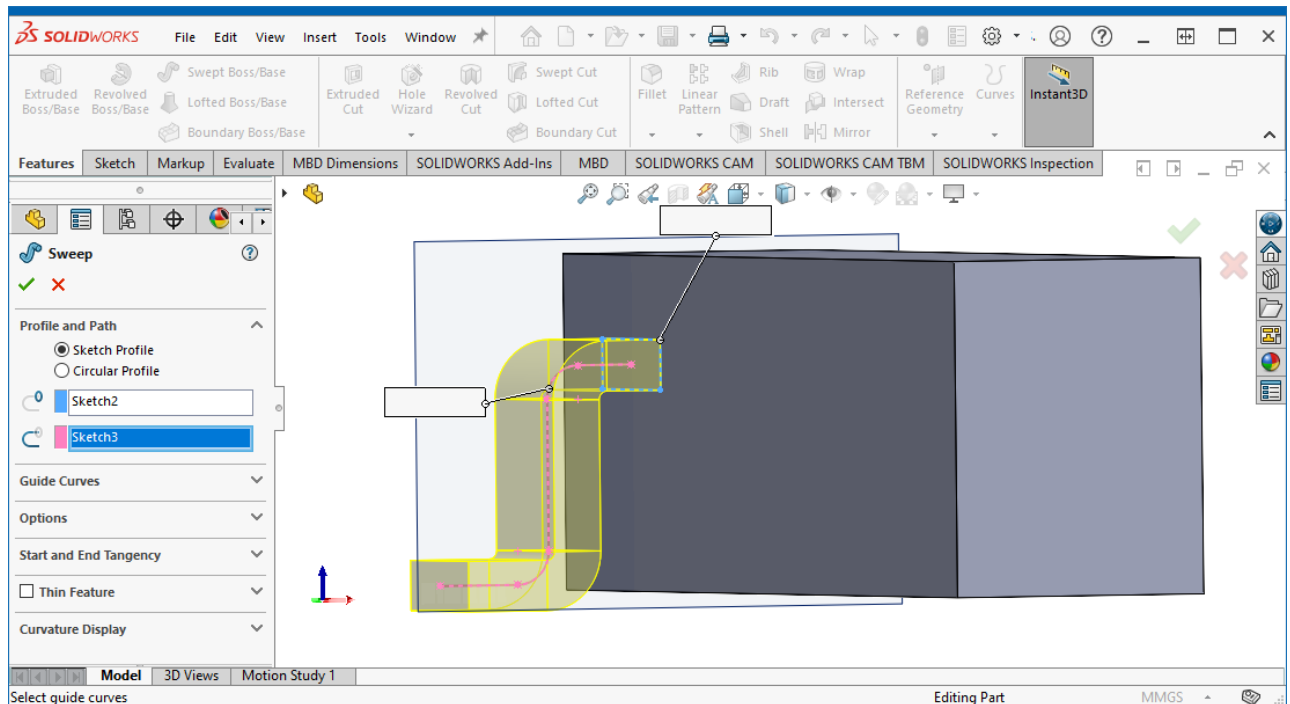


Рис. 9.42. Формування виводу мікросхеми за траєкторією

Оскільки на одній зі сторін мікросхеми виводів має бути три, тому використаємо лінійний масив для створення додаткових ніжок (але створювати будемо по чергово, щоб потім можна було правильно дзеркально відобразити потрібні елементи). Відкрийте панель **Features** та оберіть команду **Linear Pattern**. У **Менеджері властивостей** вказуємо наступні параметри (рис. 9.43):

- **Direction 1** – передня грань корпусу (**Face<1>**);
- Інтервал **D1** – 0,95 мм;
- Кількість елементів – 2;
- **Features and Faces** – вказати контур ніжки, що отриманий внаслідок створення об'ємної фігури за траєкторією.

Повторюємо процедуру побудови додатково для третьої ніжки з такими самими параметрами, окрім **D1**. Цього разу інтервал вкажіть як 1,9 мм.

Таким чином, половина мікросхеми готова. Далі виконаємо функцію віддзеркалення.

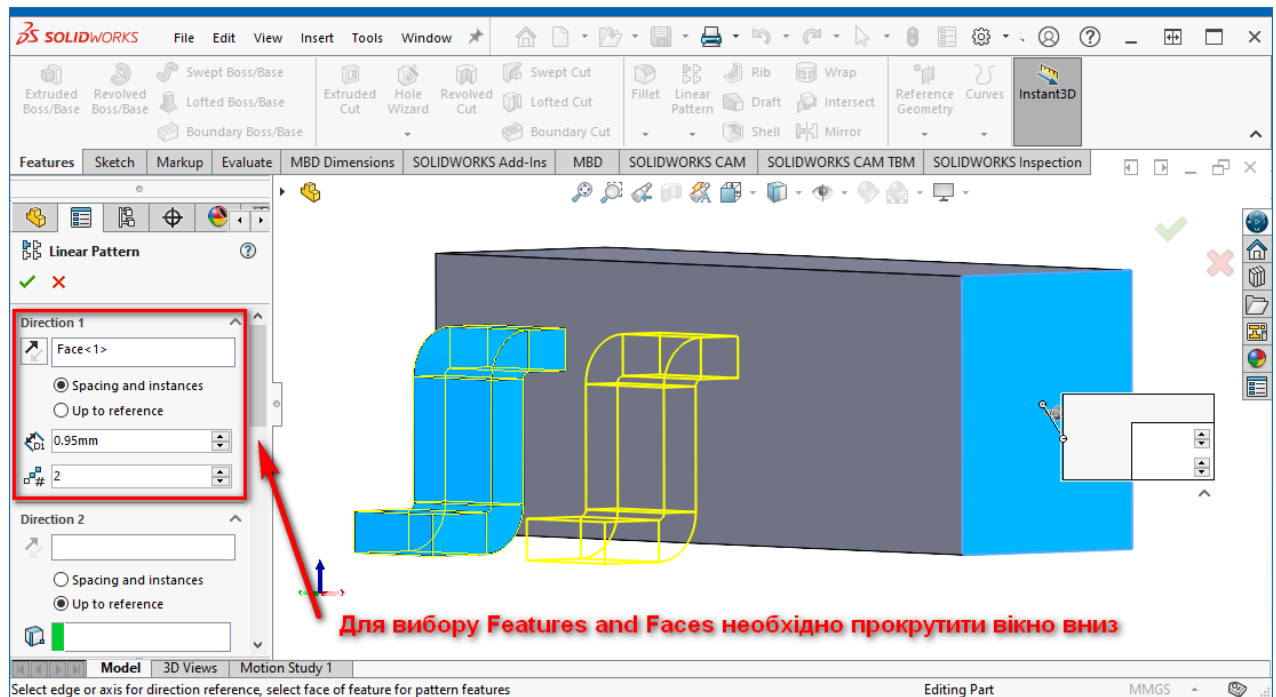


Рис. 9.43. Дублювання виводу за допомогою *Linear Pattern*

У панелі *Features* оберіть команду *Mirror*. У *Менеджері властивостей* для віддзеркалення вкажіть наступні значення (рис. 9.44): *Mirror Face/Plane* – вкажіть протилежну грань до тієї, із якої виходять наразі виводи мікросхеми; *Features to Mirror* – оберіть корпус та крайні ніжки мікросхеми.

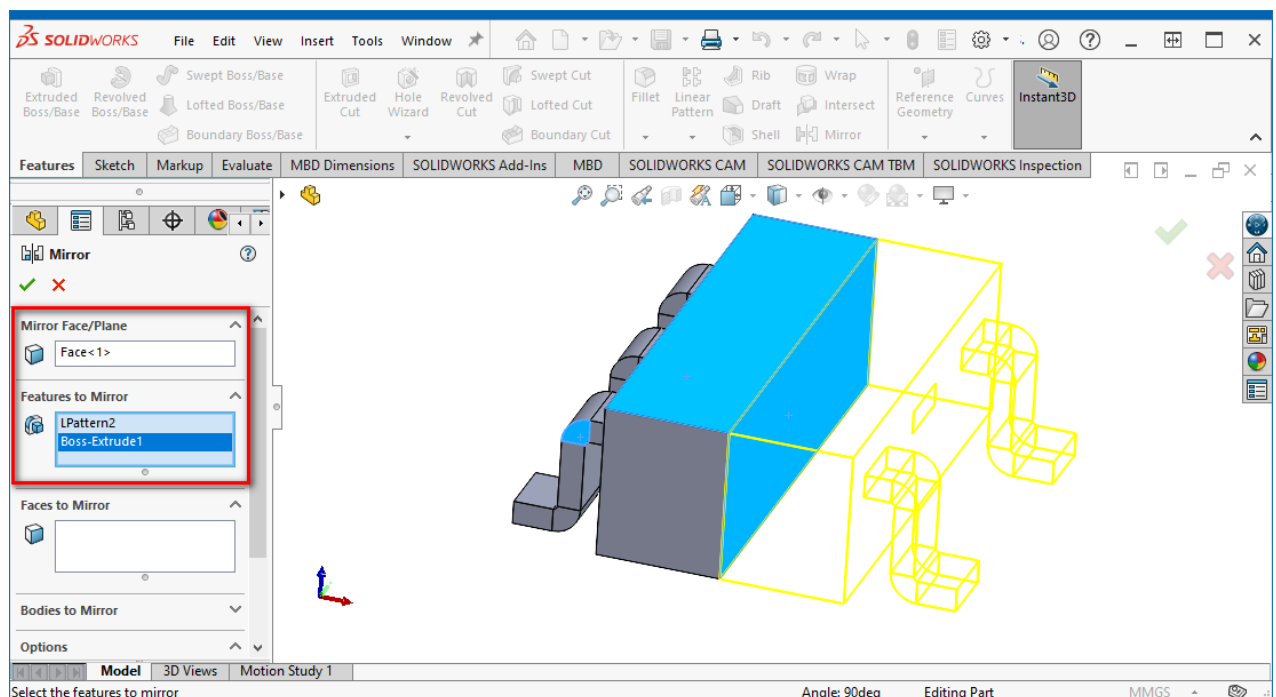


Рис. 9.44. Віддзеркалення моделі

Модель майже готова. Залишилось тільки додати «ключ» на корпусі та зафарбувати сам корпус, ключ і виводи у відповідні кольори.

Для додавання «ключа» виконайте наступні дії. Виділіть верхню грань корпусу мікросхеми, натисніть ПКМ та у спливаючому меню оберіть **Sketch**. Потім оберіть інструмент **Circle** та намалуйте коло радіусом 0,15 мм (рис. 9.45). Координати центра кола мають бути наступними: X – 0,25 мм, Y – 2,65 мм.

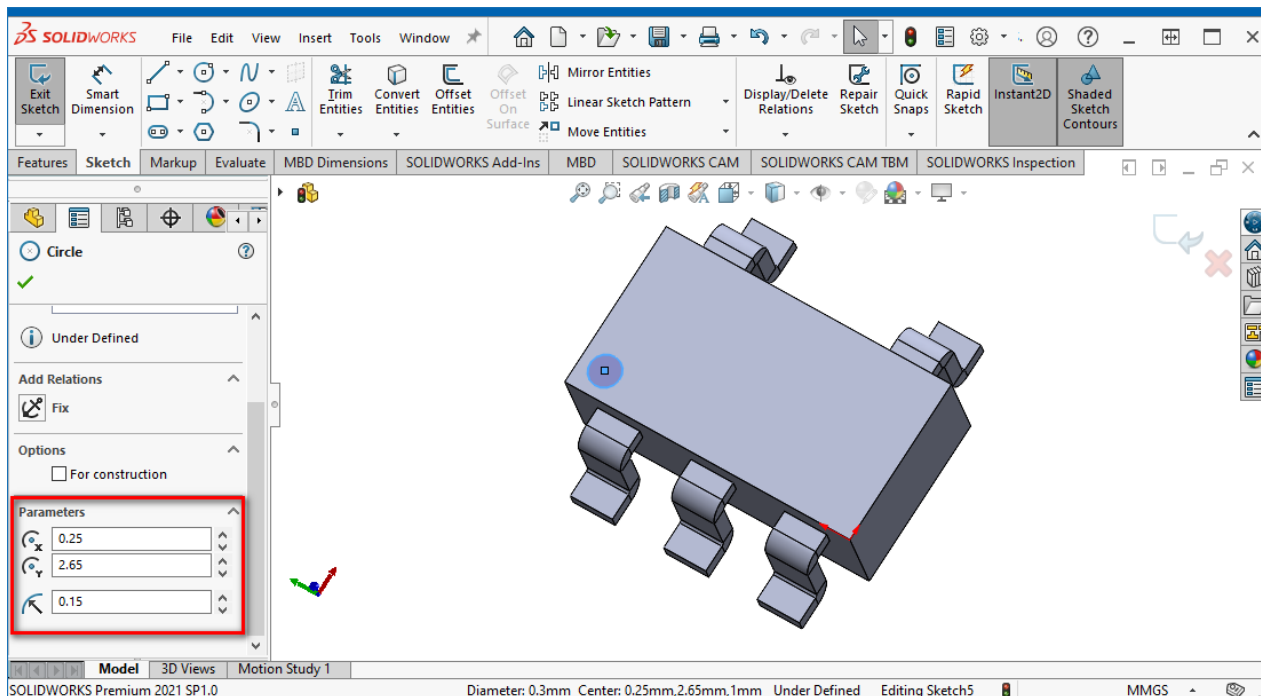


Рис. 9.45. Створення ескізу кола на верхній грані корпусу мікросхеми

Далі робимо невелике заглиблення на основі контуру кола (рис. 9.46). У панелі **Features** оберіть команду **Extruded Cut**. У **Менеджері властивостей** встановіть глибину вирізу D1=0,05 мм у напрямку всередину мікросхеми (напрямок 1).

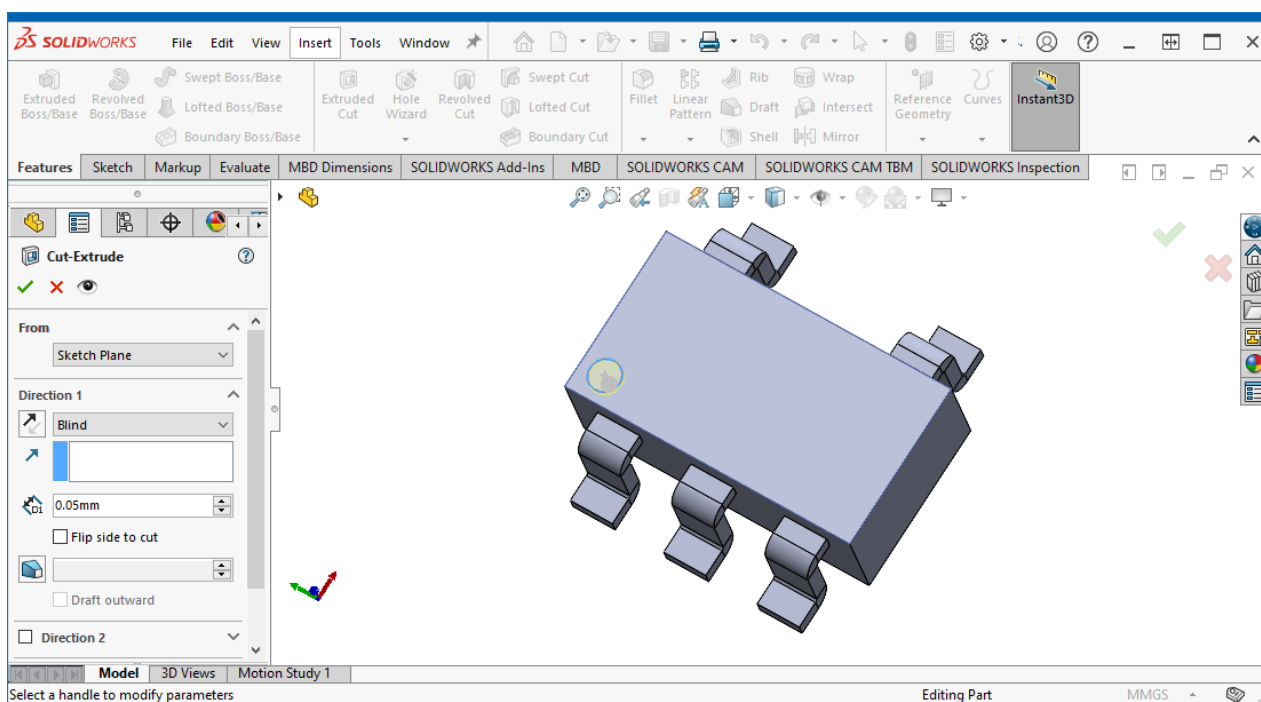


Рис. 9.46. Створення заглиблення у корпусі на основі ескізу кола

Насамкінець виконайте фарбування наступним чином: корпус – чорний, «ключ» – білий, виводи – світло-сірі. Кінцевий результат представлено на рис. 9.47.

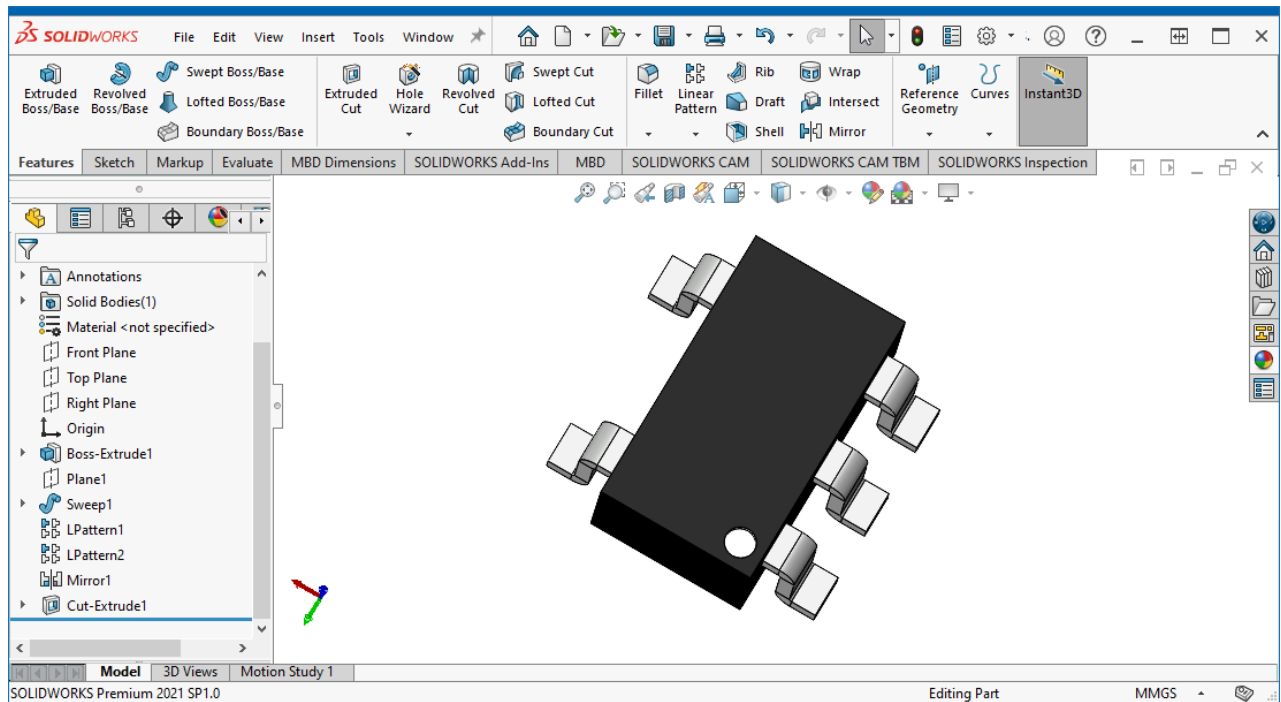


Рис. 9.47. Розроблена модель корпусу SOT-23-5

Збережіть результат, задавши ім'я **SOT-23-5**, як у форматі ***.sldprt**, так і ***.step**.

3.5. Проектування моделі мікросхеми у корпусі DIP14 із кроком виводів 2,5 мм та відстанню між рядами виводів 7,5 мм

У комп'ютерному практикумі №4 було розроблено бібліотечний компонентний модуль мікросхеми K155ЛА3, яка була радянським аналогом мікросхеми SN7400. Відмітною особливістю мікросхеми K155ЛА3 є те, що її геометричні розміри відповідали метричній системі вимірювань, тоді як розміри SN7400 – дюймовій. Зокрема, якщо крок виводів мікросхеми SN7400 становив 0,1 дюйма (тобто 2,54 мм), то крок K155ЛА3 – 2,5 мм; якщо відстань між рядами виводів SN7400 – 0,3 дюйми (тобто 7,62 мм), то у K155ЛА3 ця відстань становить 7,5 мм. Ці нюанси необхідно враховувати під час розробки тривимірної моделі корпусу мікросхеми K155ЛА3.

Якщо ви захочете знайти готову модель корпусу мікросхеми K155ЛА3 на спеціалізованих інтернет-ресурсах, то навряд чи вам це вдасться, оскільки усі корпуси будуть зроблені за стандартами мікросхеми SN7400. Це тому, що мікросхема SN7400 випускається досі, відповідає міжнародним стандартам і

поширена в усьому світі, тоді як K155ЛА3 вже не випускається, а територія її використання була локальною, тому з нею знайома мала кількість розробників. Якщо ж ви спробуєте модель корпусу мікросхеми SN7400 додати до розробленого компонентного модуля K155ЛА3, то помітите, що модель не узгоджується із ПМ.

Тому модель корпусу мікросхеми K155ЛА3 створіть самостійно. Для того, щоб розробити тривимірну модель, скористайтесь геометричними розмірами, що наведені на рис. 4.2 у комп'ютерному практикумі №4. Додатково ознайомтеся із [цим відео](#) на YouTube, в якому пояснюється, як розробляти мікросхеми у корпусі DIP14 (створюючи модель корпусу за відео, звичайно, враховуйте ті геометричні характеристики, які потрібні саме вам).

Збережіть результат, задавши ім'я **DIP14(K155LA3)**, як у форматі ***.sldprt**, так і ***.step**.

3.6. Підключення тривимірних моделей до бібліотечних компонентів

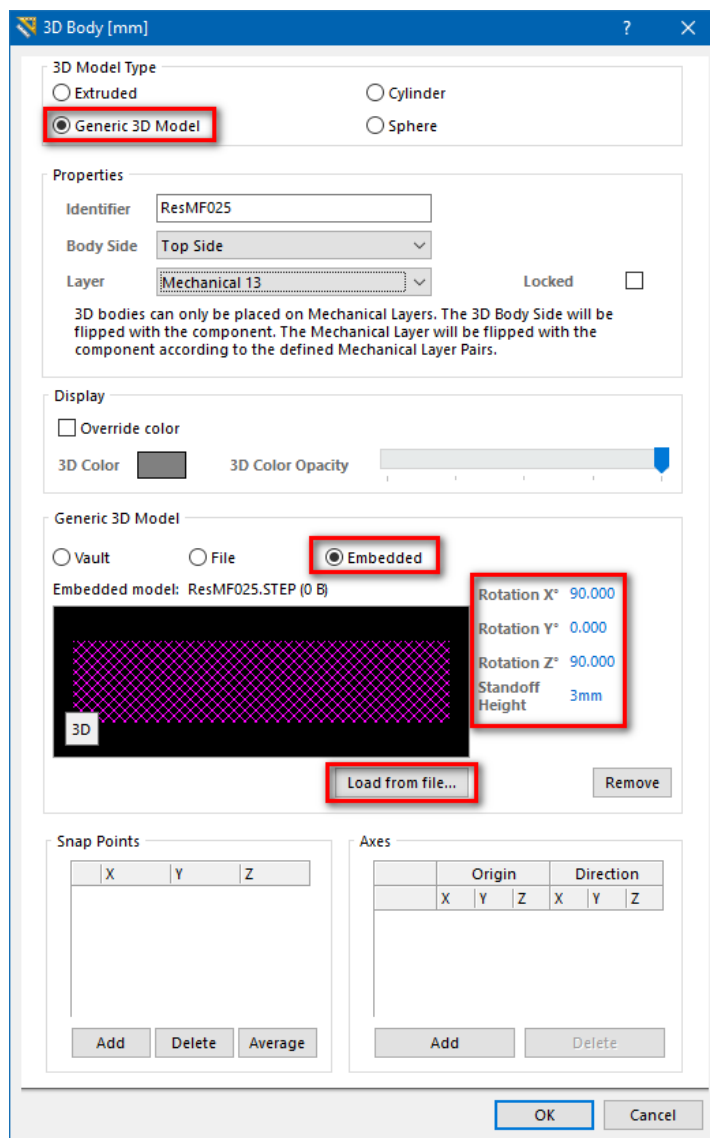
Для того, щоб додати тривимірну модель корпусу у форматі STEP до ПМ, необхідно виконати наступні дії (рис. 9.48):

1. Перейти у бібліотеку посадкових місць та обрати те, до якого буде підключатись модель корпусу.
2. Виконати команду **Place** → **3D Body** та у вікні, що відкриється, обрати опцію **Generic STEP Model**.
3. Натиснути кнопку **Load from file** та за допомогою стандартного менеджера пошуку файлів знайти і підключити файл моделі⁹⁹.
4. За потреби вказати кутові зміщення моделі вздовж кожної з координат та зміщення моделі за висотою¹⁰⁰.
5. Натиснути кнопку **OK** у вікні **3D Body** для розміщення моделі.

⁹⁹ Водночас автоматично буде заповнене поле **Identifier**, в якому відображатиметься назва підключеного файлу моделі.

¹⁰⁰ Така необхідність з'являється тому, що моделі корпусів створюються у САПР механічних пристроїв відносно певної площини та можуть мати різну орієнтацію. І ця площина та орієнтація моделі не завжди співпадає із площиною та орієнтацією ПМ в **Altium Designer** (здебільшого не співпадає). Тому доводиться її «крутити» таким чином, щоб площина та орієнтація моделі корпусу співпала із площиною ПМ. А змінювати висоту корисно, наприклад, для електронних компонентів, що призначені для монтажу в отвори, оскільки вони мають зазвичай довгі виводи і можуть бути по різному встановлені на платі за висотою.

6. За потреби знову відкрити вікно **3D Body** для встановленого корпусу (подвійно клацнувши на ньому ЛКМ) та змінити кутові положення і висоту.
7. Також, виділивши та утримуючи корпус за допомогою ЛКМ, змістити його так, щоб виводи співпали із центрами КМ посадкового місця. Зручніше це робити у тривимірному режимі.

Рис. 9.48. Вікно **3D Body**

3.6.1. Підключення моделі корпусу резистора до посадкового місця.

Перейдіть у бібліотеку посадкових місць та оберіть **RES_CONST_L10.0D0.8**. Виконайте команду **Place** → **3D Body**. У вікні **3D Body** активуйте усі необхідні опції та завантажте розроблену STEP модель резистора **ResMF025**. Для позиціонування та обертання моделі резистора встановіть такі значення кутів обертання (**Rotation**)

та висоти (*Standoff Height*), як показано на рис. 9.48. Після чого натисніть кнопку **ОК** для виходу із режиму розміщення та виконайте позиціонування (шляхом перетягування мишкою) моделі таким чином, щоб її виводи точно співпали із центрами КМ. Кінцевий результат представлено на рис. 9.49.

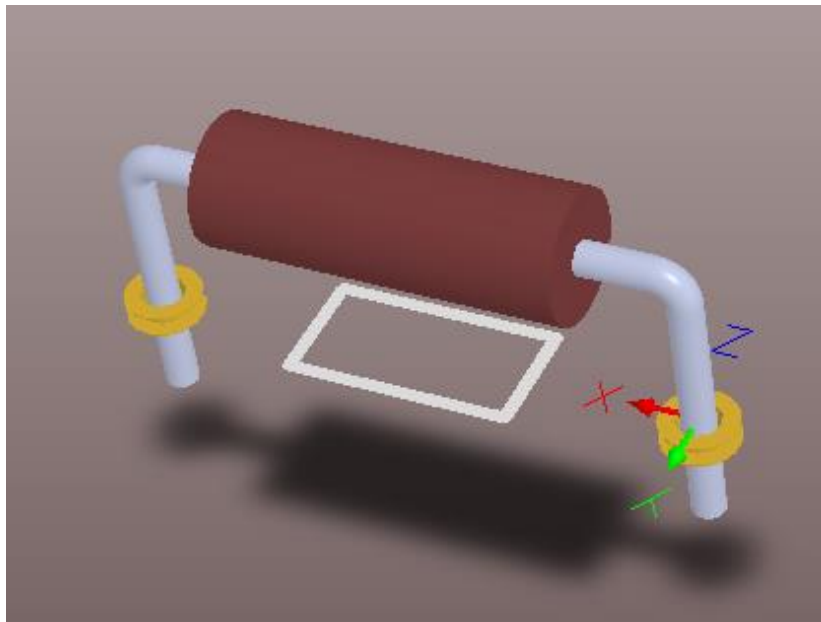


Рис. 9.49. Підключена до ПМ тривимірна модель резистору

3.6.2. Підключення інших моделей, що були розроблені власноруч.

Аналогічно до описаного вище підключіть до ПМ **CAP_POLAR_L2.5D0.8** модель корпусу **CapHitanoECR**, до ПМ **IND_L6.0D0.8** – модель **IndKIG**, до ПМ **SOT23-5** – модель **SOT-23-5**, до ПМ **DIP14_W7.5_X2.5** – модель **DIP14(K155LA3)**.

Якщо деякі із ПМ вже містили прості моделі корпусів, зроблені засобами **Altium Designer**, то попередньо видаліть їх. Усі моделі корпусів мають бути розміщені на 13-му механічному шарі.

3.6.3. Підключення інших моделей, що були завантажені із відкритих джерел.

У цьому комп'ютерному практикумі ми створили лише кілька моделей корпусів, а компонентних модулів у нас значно більше. Наразі розглянемо, яким чином можна знайти необхідну модель корпусу електронного компоненту на такому відкритому та безкоштовному ресурсі як [3D Content Central](https://www.3dcontentcentral.com/).

Для прикладу, знайдемо тривимірну модель корпусу для мікросхеми драйвера транзисторів MD1822, яка випускається в корпусі QFN16. Перейдіть на сайт [3D Content Central](#) та у рядку пошуку введіть назву корпусу мікросхеми «qfn 16»¹⁰¹ (рис. 9.50).

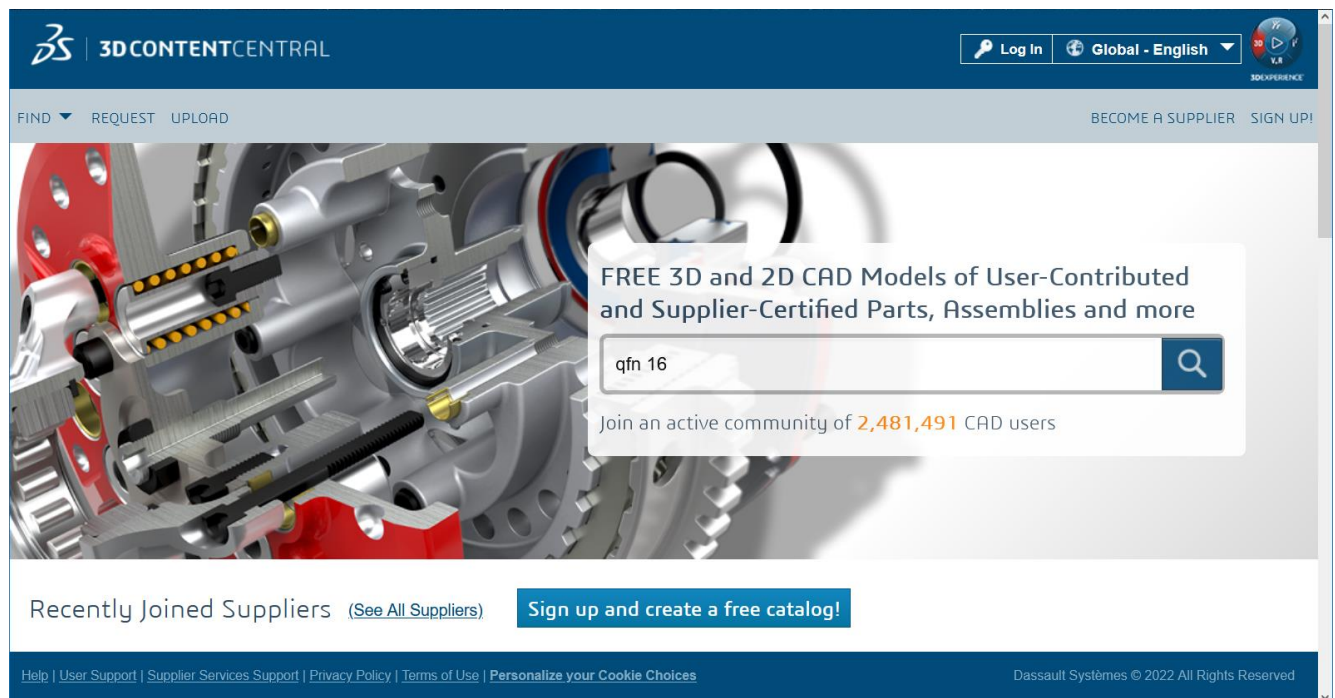


Рис. 9.50. Титульна сторінка сайту [3D Content Central](#) та рядок пошуку

Після цього з'явиться перелік усіх наявних моделей корпусу QFN16 (рис. 9.51). Зверніть увагу, що в деяких назвах міститься напис типу «4x4», «5x5», «3x3» тощо. Цей напис позначає розміри мікросхеми, зокрема, 5x5 означає, що розміри мікросхеми становлять 5 на 5 міліметрів. Який необхідно обрати? Якщо звернути увагу на рис. 8.7, де представлені геометричні розміри корпусу, що взяті із технічної документації мікросхеми MD1822, то стає очевидно, що нам потрібно обрати корпус розмірами 3 на 3 міліметри. Ви можете обрати і завантажити¹⁰² будь-яку із придатних моделей корпусу QFN16. Краще обрати кілька моделей від різних авторів, а потім подивитись, як ці моделі виглядають у тривимірному режимі на платі, та залишити кращу. Адже один автор може підійти до створення моделі більш серйозно, ніж інший, тому його модель виглядатиме якісніше.

¹⁰¹ Краще шукати саме за назвою корпусу, ніж за назвою мікросхеми. Це пов'язано із тим, що корпус є загальним для багатьох різних мікросхем, тому розробник тривимірної моделі зазвичай буде її іменувати саме за назвою корпусу, а не за назвою самої мікросхеми.

¹⁰² Для того, щоб мати можливість щось завантажувати із сайту [3D Content Central](#), спочатку потрібно зареєструватись.

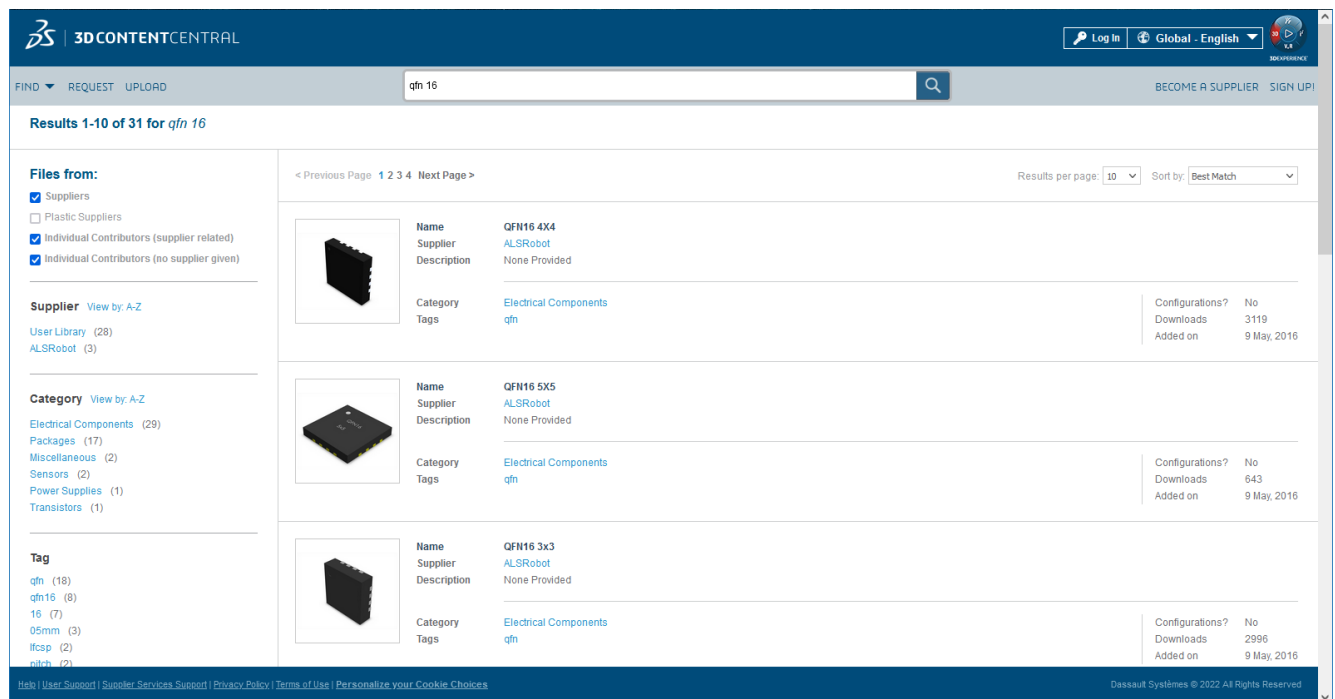


Рис. 9.51. Результат пошуку моделей корпусу QFN16

Після завантаження обраної моделі у форматі STEP (до речі, для більшості моделей на сайті доступні найрізноманітніші формати завантаження, такі як: *.step, *.sldprt, *.sat, *.cutproduct, *.pdf, *.jpg, *.x_t тощо) підключіть її до розробленого ПМ мікросхеми драйвера транзисторів MD1822.

Результат представлено на рис. 9.52.

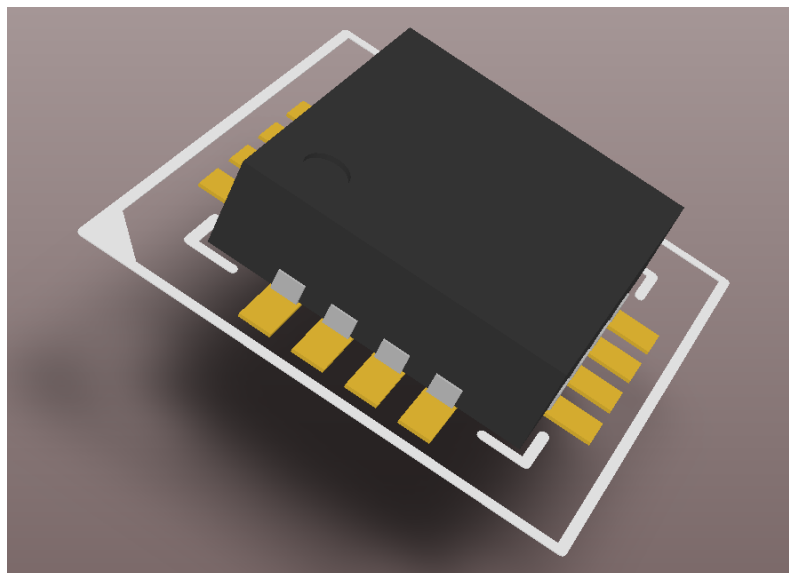


Рис. 9.52. Підключена модель корпусу мікросхеми MD1822

Подібним чином знайдіть тривимірні моделі для усіх інших електронних компонентів, для яких на попередніх комп'ютерних практикумах були створені компонентні модулі, та додайте знайдені моделі до ПМ.

4. Питання для самоконтролю

1. Що таке 3D-проєктування та які воно надає переваги під час проєктування ДП?
2. Яким чином створюються прості тривимірні моделі корпусів електронних компонентів засобами **Altium Designer**?
3. Яким чином додати тривимірну модель, що створена у **SolidWorks**, до бібліотечного компонентного модуля в **Altium Designer**?
4. Опишіть вікно **3D Body** та поясніть, які налаштування потрібно у ньому виконати під час додавання тривимірного об'єкту.
5. Яким чином створити тривимірну модель у **SolidWorks**? Поясніть на прикладі.
6. Де і яким чином можна шукати готові тривимірні моделі корпусів електронних компонентів?

Комп'ютерний практикум № 10.

РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ ТА ДРУКОВАНОЇ ПЛАТИ ПІДСИЛЮВАЧА НА ОДНОМУ ТРАНЗИСТОРІ

Мета роботи: розробити принципову схему підсилювача на одному транзисторі, використовуючи електронні компоненти із вбудованих бібліотек; на основі розробленої принципової схеми створити друковану плату; виконати ручне трасування плати та виправити помилки, які можуть виникнути при цьому.

1. Теоретичні відомості

Електронний підсилювач – це підсилювач, основне завдання якого полягає у тому, щоб збільшити сигнал за потужністю, водночас зберігши його початкову форму. Обов'язковою умовою роботи підсилювача є наявність зовнішнього джерела живлення.

Основним елементом підсилювачів є транзистор. Якщо використовується біполярний транзистор, то він може бути включений за трьома схемами: із загальним емітером, загальною базою або загальним колектором. Найчастіше застосовується схема із загальним емітером (ЗЕ).

Назва схеми «із загальним емітером» означає, що вивід емітера є загальним для вхідного і вихідного ланцюга. При включенні біполярного транзистора за схемою із ЗЕ вхідний сигнал подається на базу відносно емітера, а вихідними величинами є колекторний струм і напруга на колекторі відносно емітера. Особливістю цієї схеми є те, що вихідний сигнал інвертується щодо вхідного, тобто між вихідною і вхідною напругами є фазовий зсув 180° .

Основною перевагою включення транзистора за схемою підсилювача із ЗЕ є здатність отримувати найбільше підсилення за потужністю, що пов'язано із властивістю такого каскаду підсилювати як струм, так і напругу.

Найпростіша схема підсилювального каскаду із ЗЕ представлена на рис. 10.1. Він містить мінімальну кількість елементів, проте має істотний недолік у вигляді малої ефективності термостабілізації.

Конденсатор С1 використовується для відокремлення постійної складової вхідного сигналу від змінної (тобто, конденсатор пропускає тільки змінний сигнал). Таку саму функцію виконує конденсатор С2, тільки на виході.

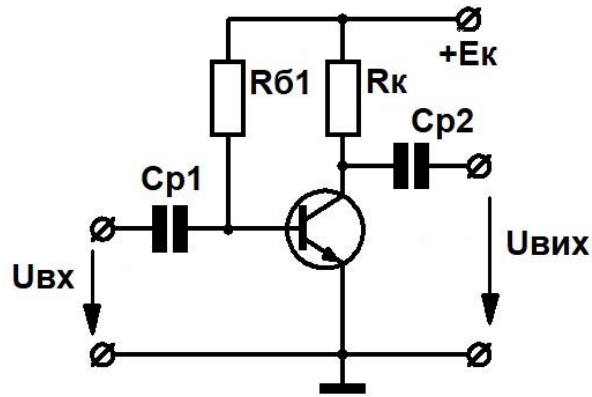
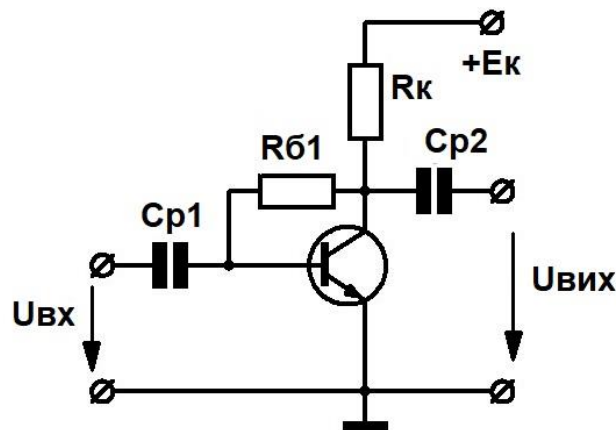


Рис. 10.1. Найпростіша схема підсилювача із ЗЕ

Більш високу термостабільність має каскад із ЗЕ, схема якого наведена на рис. 10.2. Термостабілізація у цій схемі здійснюється за рахунок від'ємного зворотного зв'язку, який вводиться у каскад за допомогою встановлення резистора $R_{б1}$ між базою і колектором транзистора.

Рис. 10.2 Схема із термостабілізацією за рахунок встановлення резистора $R_{б1}$ між базою і колектором транзистора

Все ж найкращі властивості серед базових схем із ЗЕ має схема, що наведена на рис. 10.3¹⁰³. В цій схемі ефект термостабілізації досягається фіксацією напруги на базі за допомогою резистивного дільника напруги ($R_{б1}$ і $R_{б2}$) і введенням від'ємного зворотного зв'язку на постійному струмі за допомогою встановлення резистора R_e .

¹⁰³ Більш докладно про роботу цієї схеми підсилювача та принципи термостабілізації ви можете прочитати у методичних вказівках «[Розрахунок двокаскадного підсилювача низької частоти](#)».

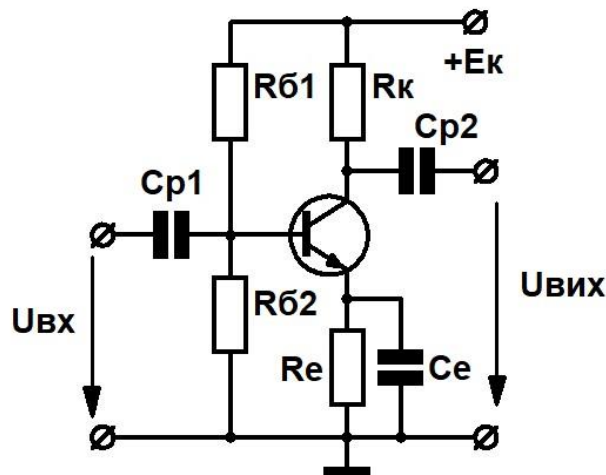


Рис. 10.3. Схема із термостабілізацією за рахунок використання резистивного дільника напруги на вході та резистора на емітерному виводі транзистора

Дільник напруги у схемі на рис. 10.3 дозволяє трохи привідкрити транзистор за рахунок того, що струм, який протікає через R_{61} і R_{62} , надходить на базу транзистора, після чого йде через емітер. Це називається базове зміщення транзистора, тобто його відкриття. Напруга зміщення визначає робочу точку.

Завдяки тому, що транзистор виявляється відкритим за рахунок дільника напруги, то вхідний сигнал не витрачає свою енергію на його відкриття і тим самим не спотворюється.

Протікання струму через транзистор призводить до його нагрівання. Зазвичай уся потужність розсіюється на колекторі. Однак підвищення температури різко змінює характеристики транзистора, зокрема опір його р-п переходу суттєво знижується при підвищенні температури.

Для того, щоб стабілізувати роботу транзистора, необхідно збалансувати його опір іншим джерелом. Це можна зробити з допомогою додаткового опору R_e на емітерному виході. Коли опір транзистора зменшується, резистор R_e забирає частину напруги на себе і не дозволяє збільшити струм в ланцюзі. Завдяки цьому транзистор: 1) не закривається; 2) не переходить у режим насичення; 3) не спотворює сигнал; 4) не перегрівається (а отже, його термостабілізація покращується).

А для того, щоб у нормальному режимі роботи, коли транзистор не нагрівається, резистор R_e не зменшував потужність схеми, у ланцюг додано шунтувальний електролітичний конденсатор C_e . Завдяки наявності цього конденсатора виключається від'ємний зворотний зв'язок за змінним струмом.

2. Завдання на комп'ютерний практикум

Розробити принципову схему та друковану плату підсилювача на одному транзисторі із ЗЕ (рис. 10.4), використовуючи електронні компоненти із вбудованих бібліотек.

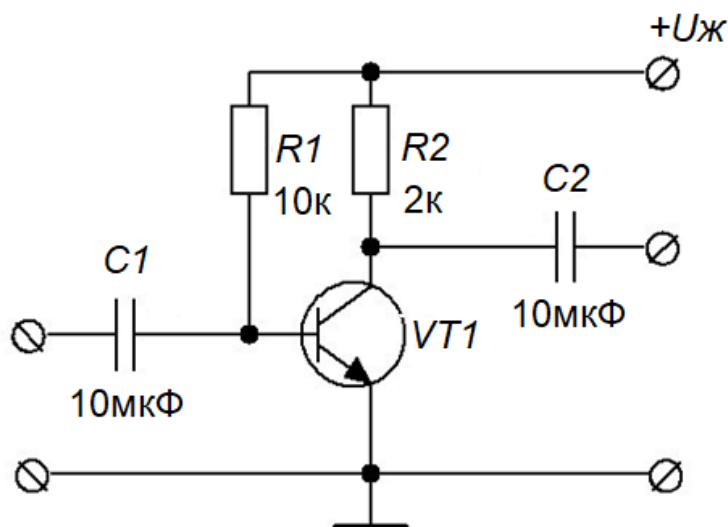


Рис. 10.4. Принципова схема підсилювача, яку необхідно розробити

Перелік елементів, що використовуються у схемі: біполярний транзистор 2N3904, керамічні конденсатори $C1$ та $C2$ у корпусі із радіальними виводами для монтажу в отвори, резистори $R1$ та $R2$ у корпусі із аксіальними виводами для монтажу в отвори, три роз'єми типу PLS-2 (перший буде використовуватись для підключення вхідного сигналу, другий – для живлення, третій – для зняття сигналу із виходу плати).

3. Порядок виконання комп'ютерного практикуму

Проєкт **Altium Designer** містить набір документів та параметрів, які із ним пов'язані. Файл проєкту (має розширення **.PrjPCB**) є текстовим файлом у стандарті **ASCII**, в якому перераховані документи, що використовуються у проєкті¹⁰⁴, та параметри, які використовуються для формування вихідних даних (наприклад, для друку або передачі в системи управління виробництвом CAM). У проєкт додаються посилання на аркуші з принциповими схемами, друкованими платами,

¹⁰⁴ Документи, що не пов'язані з яким-небудь проєктом, називаються вільними (**Free Documents**).

бібліотеками тощо. Після компіляції проєкту можна виконувати такі операції, як перевірка проєкту, синхронізація і порівняння. Взагалі, процес компіляції проєкту є важливим етапом, оскільки під час компіляції проєкт оновлюється з урахуванням усіх змін, внесених у вихідну принципову схему або ДП.

3.1. Створення нового проєкту

Спочатку створимо новий проєкт ДП. Існує кілька способів, як це зробити.

Перший спосіб полягає у використанні меню **File**, за допомогою якого виконують команду **New** → **Project**. У вікні, що після цього відкриється, в області **Project Types** оберіть **PCB Project**, а в області **Project Templates** оберіть тип документу **Default**. У полі **Name** задайте ім'я проєкту **AmplifierOnTransistor**, а також переконайтесь, що активовано прапорець навпроти пункту **Create Project Folder** (це дозволить автоматично створити однойменну папку проєкту). Натисніть кнопку **OK**. На панелі **Projects** має з'явитись новий файл проєкту **AmplifierOnTransistor.PrjPCB** (без доданих документів) (рис. 10.5).

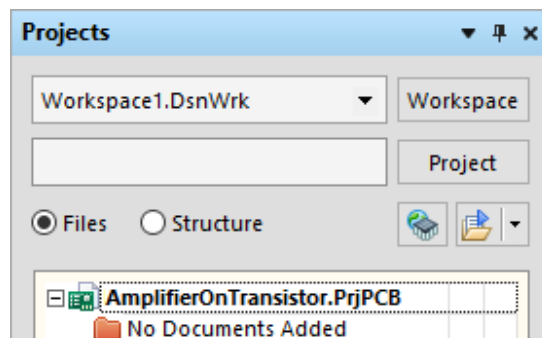


Рис. 10.5. Відображення нового проєкту плати на панелі **Projects**

Інший спосіб¹⁰⁵ створити новий проєкт ДП полягає у використанні розділу **New** панелі **Files**¹⁰⁶. Якщо у цьому розділі обрати пункт **Blank Project (PCB)**, то автоматично буде створений новий проєкт із іменем **PCB_Project1.PrjPCB** та розміщений на панелі **Projects**. Для цього проєкту не буде створено каталог автоматично, а ім'я, як бачимо, задане за замовчуванням. Тому спочатку необхідно

¹⁰⁵ Якщо ви вже створили проєкт плати попереднім способом, то наразі немає потреби використовувати цей.

¹⁰⁶ Якщо панель не відображається, то натисніть кнопку **System** в правому нижньому куті основного вікна та виберіть **Files**.

створити каталог для збереження цього проєкту, після чого із меню **File** необхідно обрати пункт **Save Project As** для того, щоб зберегти проєкт під заданим іменем. У вікні збереження файлу проєкту необхідно перейти у створену папку, у полі **File Name** ввести ім'я **AmplifierOnTransistor.PrjPCB** і натиснути кнопку **Save**.

3.2. Додавання до проєкту принципової схеми та встановлення її налаштувань

Після створення проєкту у нього необхідно додати аркуш принципової схеми. Для цього клацніть ПКМ на імені проєкту у панелі **Projects** і виберіть **Add New to Project** → **Schematic**. У головному вікні відкриється новий аркуш принципової схеми з ім'ям **Sheet1.SchDoc**, а у панелі **Projects** в каталозі **Source Documents** з'явиться значок схеми, що пов'язана з проєктом.

Збережіть нову схему (з розширенням **.SchDoc**), вибравши меню **File** → **Save As**. Перейдіть у каталог на жорсткому диску, в якому зберігається сам проєкт, у полі **File Name** введіть ім'я **AmplifierOnTransistor.SchDoc** і натисніть кнопку **Save**.

Важливо!

Документи, які збережені у тому ж каталозі, що і файл проєкту (або у вкладених каталогах), зв'язуються з проєктом із використанням відносних шляхів, а файли, збережені в іншому місці, – з використанням абсолютних шляхів.

Оскільки ми додали у проєкт принципову схему, то файл проєкту змінився. Клацніть ПКМ на імені файлу проєкту в панелі **Projects** і виберіть **Save Project**.

Відкрийте вікно властивостей документу принципової схеми **Design** → **Document Options** та на вкладці **Sheet Options** в полі **Standard Styles** встановіть формат аркуша A4.

Перейдіть у меню **Tools** та виберіть пункт **Schematic Preferences** (послідовність клавіш: <T, P>), щоб відкрити вікно налаштувань **Preferences**. У цьому діалоговому вікні перейдіть у розділ **Schematic**, виберіть пункт **Default Primitives** та активуйте опцію **Permanent** (у правій частині вікна).

Виконайте команду **View** → **Fit Document** (послідовність клавіш: <V, D>), щоб повністю вписати документ у вікно перегляду.

3.3. Розміщення компонентів на принциповій схемі

Відкрийте панель ***Libraries***, клацнувши на її вкладці в правій частині робочого простору. Із випадного списку у верхній частині панелі ***Libraries*** оберіть вбудовану бібліотеку **Miscellaneous Devices**.

Виберіть у списку транзистор **2N3904** та перетягніть його у робочу область принципової схеми¹⁰⁷. Перед розміщенням компонента на принциповій схемі можна натиснути клавішу <Tab>, щоб відкрити діалогове вікно ***Component Properties*** та за потреби змінити його властивості. Для транзистора **2N3904** змініть його властивість ***Designator***, написавши у відповідному полі **VT1**. Для завершення розміщення компонента на схемі клацніть ЛКМ.

У режимі редагування або розміщення компонентів (курсор миші має форму перехрестя) переміщення курсору за межі вікна документа призведе до автоматичного панорамування. Якщо ви випадково пересунули курсор далеко від схеми, натисніть послідовно клавіші <V, F> (або виконайте із меню ***View*** команду ***Fit All Objects***), щоб перемалювати вміст вікна. Цю функцію можна використовувати навіть під час розміщення об'єкта.

Далі знайдіть у бібліотеці **Miscellaneous Devices** конденсатор **Cap**. Скористайтесь фільтром, щоб швидко знайти необхідний компонент. За замовчуванням у полі фільтра встановлено груповий символ «*», що означає відображення всіх компонентів, що містяться у бібліотеці. Введіть текст «cap» (без лапок) у полі фільтра, щоб відобразити список компонентів, що мають у своїй назві набір символів «cap». Виберіть конденсатор **Cap** та розмістіть дві його копії на аркуші принципової схеми. Під час розміщення кожного з конденсаторів натисніть <Tab>, щоб відкрити їх властивості. У вікні ***Component Properties*** у полі ***Designator*** для першого конденсатора введіть **C1**, у полі ***Comment*** із випадного списку оберіть **'=Value'**, після чого в області ***Parameters*** для параметра ***Value*** введіть значення

¹⁰⁷ Якщо необхідно розмістити багато копій одного й того ж компонента, то краще скористатись альтернативним способом розміщення – двічі клацнути на імені компонента, після чого курсор приймає вигляд перехрестя, до якого «прикріплений» контур символу компонента. Це означає, що ви знаходитесь у режимі розміщення компонентів. При переміщенні курсору контур компонента переміщується разом з ним. Якщо натиснути ЛКМ або <Enter>, то у поточному місці документа буде вставлений компонент, однак режим розміщення продовжуватиметься (символ цього ж компоненту буде прикріплений до курсору) і ви зможете встановити копію цього ж компоненту у іншій частині схеми. Для виходу із цього режиму необхідно або натиснути ПКМ, або <Esc>.

10uF та відключіть опцію **Visible** (рис. 10.6). Для другого конденсатора зробіть усе те саме, тільки у полі **Designator** введіть **C2**.

Properties for Schematic Component in Sheet [AmplifierOnTransistor.SchDoc]

Properties

Designator: C1 ☒ Visible ☐ Locked

Comment: =Value ☒ Visible

Description: Capacitor

Unique Id: FKCFQQED

Type: Standard

Link to Library Component ☐ Use Vault Component

Design Item ID: Cap

☒ Library Name: Miscellaneous Devices.IntLib

☒ Table Name:

Graphical

Location X: 157.42mm Y: 110.04mm

Orientation: 0 Degrees ☐ Locked

Mode: Normal ☒ Lock Pins ☐ Mirrored

☐ Show All Pins On Sheet (Even if Hidden)

☐ Local Colors

Parameters

Visible	Name	Value	Type
<input type="checkbox"/>	LatestRevisionDate	17-Jul-2002	STRING
<input type="checkbox"/>	LatestRevisionNote	Re-released for DXP Platform.	STRING
<input type="checkbox"/>	PackageDescription	Radial Cap, Thru-Hole; 2 Leads; 0.3	STRING
<input type="checkbox"/>	PackageReference	RAD-0.3	STRING
<input type="checkbox"/>	Published	8-Jun-2000	STRING
<input type="checkbox"/>	Publisher	Altium Limited	STRING
<input checked="" type="checkbox"/>	Value	10uF	STRING

Models

Name	Type	Description	Vault	Item Re...	Revisio...
RAD-0.3	Footprint	Radial Cap, Thru-Hole; 2 Leads; 0.3			[Unknowr]
Cap	Signal Inte				[Unknowr]
CAP	Simulation	Capacitor			[Unknowr]

Рис. 10.6. Вікно редагування властивостей конденсатора

Аналогічно описаному вище розмістіть два резистори типу **Res2** із бібліотеки **Miscellaneous Devices**. У полі **Designator** першого резистора введіть **R1**, другого – **R2**, у полі **Comment** оберіть '=Value', параметр **Value** першого резистору 10k, другого – 2k.

Насамкінець розмістіть три роз'єми. Ці компоненти знаходяться у бібліотеці **Miscellaneous Connectors**. Виберіть цю бібліотеку і у полі фільтра введіть «*2». Виберіть **Header 2** зі списку елементів і перетягніть його у робочу область принципової схеми. У вікні властивостей цього компонента у полі **Designator** введіть **XP1**, у полі **Comment** введіть **PLS 2**. У випадку другого роз'єму у полі **Designator** введіть **XP2**, для третього – **XP3**.

Після розміщення усіх компонентів принципова схема має приблизно виглядати так, як показано на рис. 10.7¹⁰⁸.

¹⁰⁸ Рекомендується під час додавання компонентів на схему залишати достатньо простору між ними, щоб забезпечити їх зручне з'єднання у майбутньому. Це важливо, оскільки з'єднання не може перетинати вивід якого-небудь компонента на шляху до наступного виводу ланцюга. У

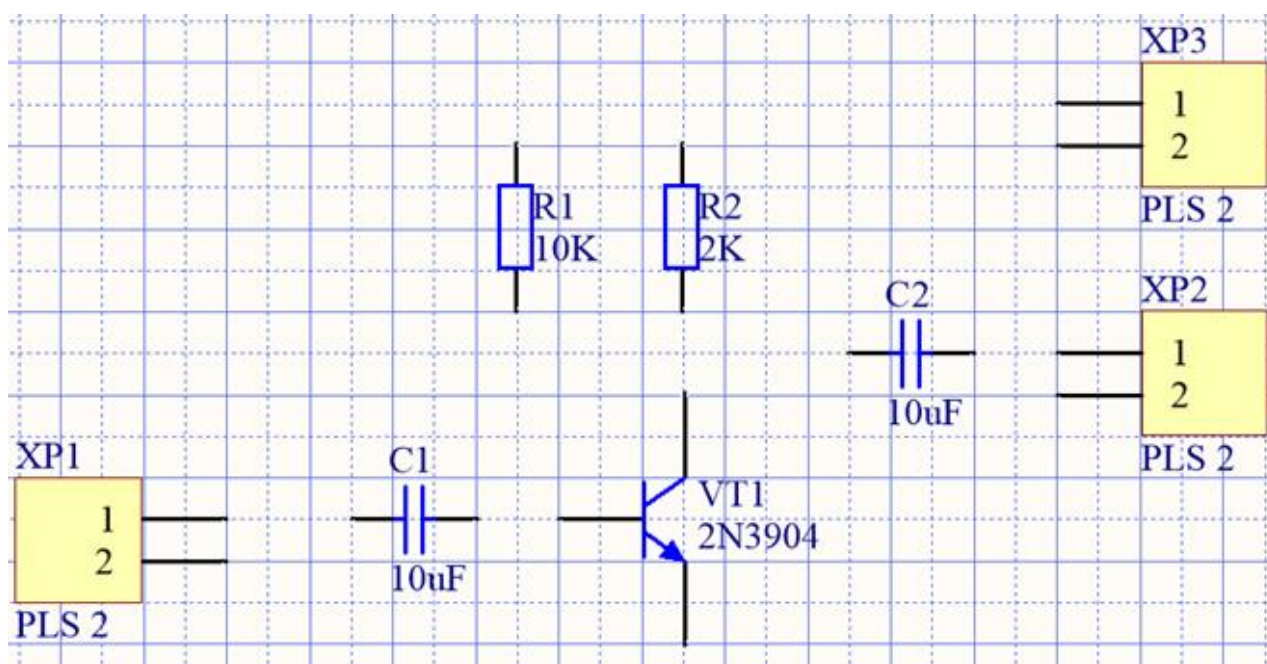



Рис. 10.7. Компоненти, що розміщені на схемі, але не з'єднані між собою

Нагадування!

Для того, щоб обертати компонент на принциповій схемі проти годинникової стрілки, його спочатку необхідно виділити, після чого натиснути клавішу <Пробіл>. Якщо ж необхідно дзеркально відобразити компонент по горизонталі, то необхідно його виділити, почати переміщувати, після чого натиснути клавішу X (щоб зробити те саме, але по вертикалі, натискають клавішу Y).

Збережіть принципову схему, обравши у меню **File** пункт **Save** (або комбінація клавіш: <F, S>).

3.4. З'єднання компонентів принципової схеми

З'єднання компонентів – це процес створення зв'язків між ними. Для того, щоб з'єднати потрібні виводи компонентів, використовують команду **Place** → **Wire** (або піктограму  на панелі інструментів), яка дозволяє увійти в режим розміщення з'єднань. Курсор миші прийме форму великого перехрестя чорного кольору, що суміщене із малим перехрестям сірого кольору, яке повернуте відносно більшого

цьому випадку всі виводи будуть підключені до цього ланцюга. Компонент завжди можна перемістити на нове місце, виділивши його і, утримуючи ЛКМ, рухати мишкою.

на 45°. Якщо у цьому режимі підвести курсор до виводу компоненту і навести його безпосередньо на точку електричного з'єднання, то мале перехрестя стане червоним. Це означає, що у цій точці можна з'єднати вивід та провід. Натисніть ЛКМ або натисніть **<Enter>**, щоб закріпити з'єднання в цій точці. Перемістивши курсор, ви побачите, що з'єднання починає тягнутися від точки закріплення до курсору. Підведіть курсор до виводу іншого компоненту і у момент, коли мале перехрестя знову стане червоним, натисніть ЛКМ або натисніть **<Enter>**. Таким чином буде встановлене з'єднання між двома виводами. Зверніть увагу, що курсор як і раніше має форму перехрестя. Це означає, що ви можете продовжити прокладання з'єднань. Для виходу із цього режиму слід натиснути ПКМ або клавішу **<Esc>**.

Подібно описаному вище можна з'єднати не тільки виводи компонентів, а й виводи та вже існуючі лінії з'єднань.

Створіть електричні з'єднання між компонентами схеми відповідно до рис. 10.8. Для зручності прокладання з'єднань використовуйте клавіші **<Page Up>** та **<Page Down>** для налаштування масштабу.

Якщо ви хочете перемістити деякий компонент на схемі разом із усіма з'єднаннями, то перетягніть його за допомогою мишки, утримуючи тільки ЛКМ. Якщо ж ви хочете перемістити компонент без з'єднань, то додатково утримуйте клавішу **<Ctrl>**.

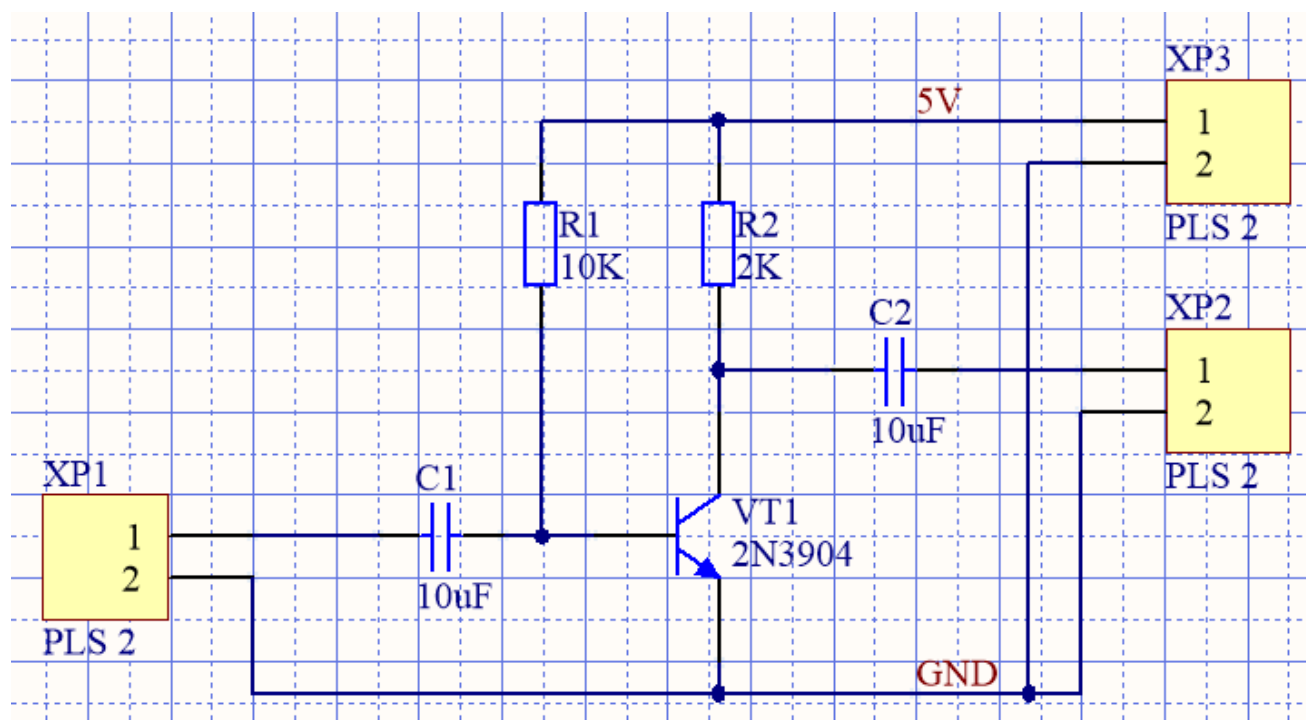


Рис. 10.8 Кінцевий вигляд принципової схеми підсилювача

Зверніть увагу, що виводи колектора та емітера транзистора **2N3904** можуть не розташовуватись у вузлах обраної координатної сітки, внаслідок чого лінії з'єднання із цими виводами можуть бути ламаними. Це означає, що необхідно зменшувати крок і підбирати його таким чином, щоб лінії з'єднань були прямими.

3.5. Ланцюги та мітки ланцюгів

Кожен набір з'єднаних між собою виводів компонентів утворює так званий ланцюг. Полегшити пошук важливих ланцюгів у схемі можна завдяки міткам¹⁰⁹, що їм привласнюються.

Додамо мітки для ланцюгів живлення та землі. У меню **Place** виберіть **Net Label** (послідовність клавіш: <P, N>). Поруч з курсором з'явиться прикріплена мітка ланцюга. Для редагування параметрів мітки перед її розміщенням натисніть клавішу <Tab>. Відкривається діалогове вікно **Net Label**. У полі **Net** введіть текст **5V** і натисніть кнопку **OK**, щоб закрити діалогове вікно.

Розташуйте мітку на провіднику схеми, який підключено до джерела живлення. Коли мітка торкається провідника, курсор приймає форму червоного перехрестя (рис. 10.9). Якщо перехрестя має сірий колір, то це означає, що ви намагаєтесь розташувати мітку на виводі, а не на лінії з'єднання.

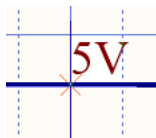


Рис. 10.9. Додавання мітки для ланцюга живлення

Після розміщення першої мітки ви все ще будете знаходитись у режимі розміщення міток, тому знову натисніть клавішу <Tab> для редагування параметрів наступної мітки. У полі **Net** введіть текст **GND** і натисніть кнопку **OK**, щоб закрити діалогове вікно. Розташуйте мітку на провіднику схеми, який підключено до землі. Клацніть ПКМ або натисніть клавішу <Esc>, щоб вийти з режиму розміщення міток.

Збережіть поточну версію схеми та сам проєкт. Перед тим як розпочати проєктування ДП на основі цієї схеми, необхідно налаштувати параметри проєкту і перевірити його на наявність помилок.

¹⁰⁹ Мітки – це, фактично, ярлики, які містять назву з'єднання. Важливо навчитись ефективно ними користуватись.

3.6. Налаштування параметрів проєкту

Принципова схема в **Altium Designer** містить усю необхідну інформацію про електричні з'єднання компонентів, яку можна використовувати для перевірки проєкту на наявність помилок.

Для того, щоб відкрити вікно налаштувань проєкту, потрібно із меню **Project** викликати команду **Project Options** (рис. 10.10).

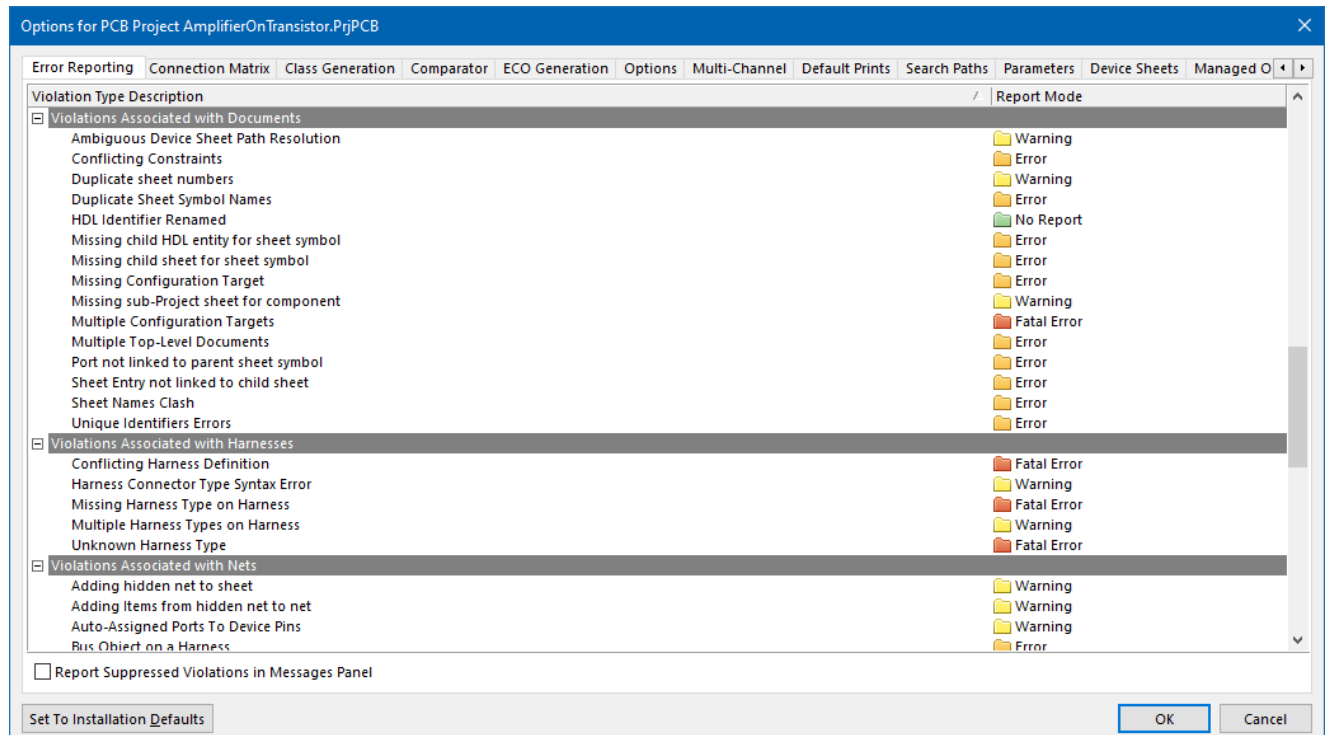


Рис. 10.10. Вікно налаштувань параметрів проєкту з відкритою вкладкою **Error Reporting**

3.6.1. Перевірка електричних властивостей принципової схеми

У цьому комп'ютерному практикумі ми не будемо змінювати ніяких правил у вкладці **Error Reporting**. Проте рекомендується самостійно ознайомитись із інформацією у цій вкладці та зрозуміти суть і призначення кожного правила.

3.6.2. Налаштування матриці з'єднань

У вкладці **Connection Matrix** діалогового вікна **Options for Project AmplifierOnTransistor.PrjPCB** (рис. 10.11) можна вказати, виводи яких типів можна з'єднувати один з одним. Під час компіляції визначається тип кожного виводу (зокрема: вхідний, вихідний, пасивний тощо), після чого кожен ланцюг перевіряється на наявність неправильно з'єднаних виводів, коли, наприклад,

вихідний вивід з'єднаний з вихідним. Після компіляції проєкту список виводів для кожного ланцюга міститься у пам'яті **Altium Designer**.

Колір на перетині рядків і стовпчиків у вкладці **Connection Matrix** є дуже важливим і визначає, як реагуватиме **Altium Designer** у тому випадку, коли відповідні виводи будуть з'єднані у проєкті. Наприклад, на перетині рядка **Output Pin** та стовпчика **Open Collector Pin** встановлено квадрат помаранчевого кольору (рис. 10.11). Цей колір вказує на те, що з'єднання вихідного виводу з відкритим колектором на принциповій схемі зумовить виникнення помилки під час компіляції проєкту.

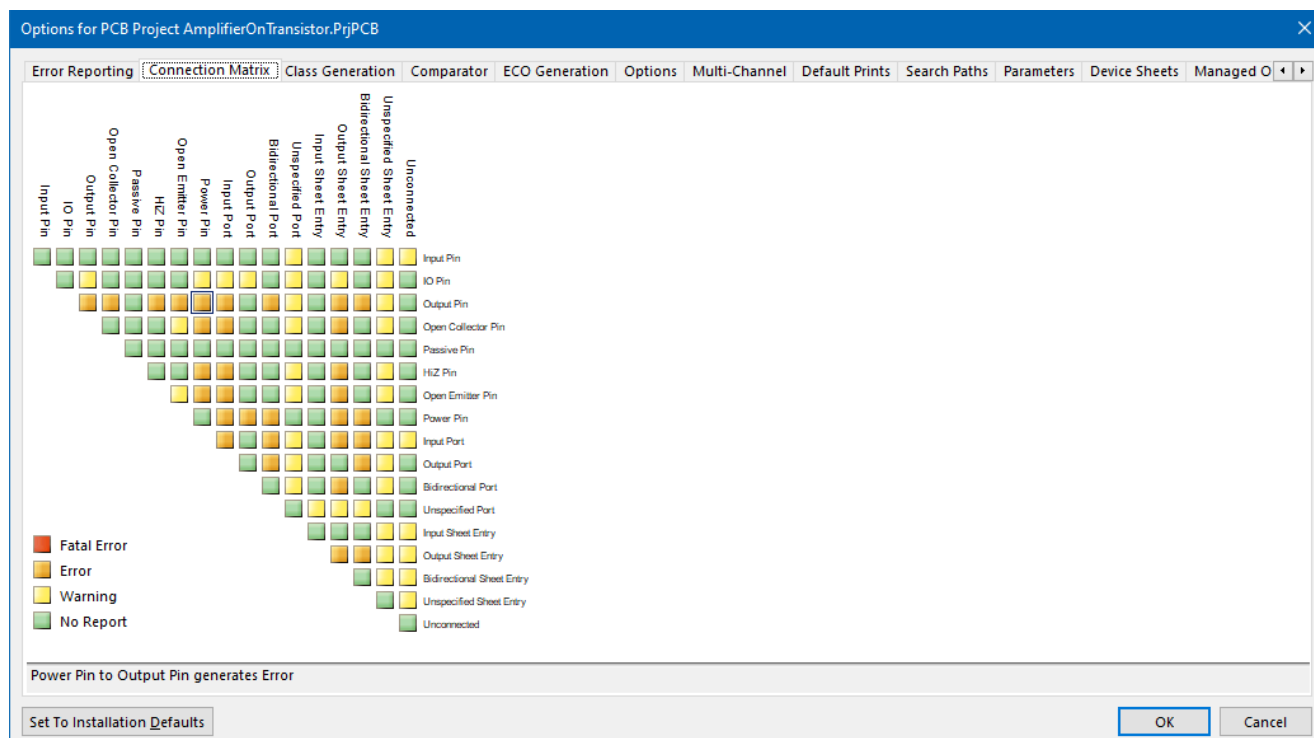


Рис. 10.11. Вкладка **Connection Matrix**

Для кожного типу помилок ви можете вибрати рівень: від ігнорування до критичної помилки (Fatal Error, Error, Warning, No Report). Для внесення змін у матрицю з'єднань необхідно клацнути ЛКМ на кольоровому квадраті (при цьому відбуватиметься послідовний перебір чотирьох можливих значень). Якщо ж натиснути ПКМ у діалоговому вікні, то буде відображене меню, в якому можна задати всі параметри одночасно, а також повернути значення за замовчуванням.

Для поточного проєкту налаштуємо, яким чином буде реагувати **Altium Designer** на непідключені пасивні виводи (наприклад, резисторів, конденсаторів та роз'ємів). За замовчуванням на перетині рядку **Passive Pin** та стовпчику **Unconnected** встановлено квадрат зеленого кольору. Це означає, що помилка не буде занесена в звіт. Тому клацніть ЛКМ на цьому квадраті, щоб він став жовтим.

Тепер у випадку виявлення непідключених пасивних виводів під час компіляції проєкту буде відображатися попередження.

3.7. Компіляція проєкту та перевірка на наявність помилок

Для того, щоб скомпілювати проєкт **AmplifierOnTransistor**, у меню **Project** виберіть пункт **Compile PCB Project AmplifierOnTransistor.PrjPcb**.

Під час компіляції проєкту виконується перевірка проєктних та електричних правил в усіх документах відповідно до налаштувань, що були задані у вкладках **Error Reporting** і **Connection Matrix** діалогового вікна **Options for Project AmplifierOnTransistor.PrjPCB** (див. рис. 10.10 та 10.11). Попередження та повідомлення про знайдені помилки відображаються на панелі **Messages**. Панель з'являється автоматично тільки за наявності помилок. Щоб відкрити її вручну, натисніть кнопку **System** у правому нижньому кутку робочого простору та у меню, що з'явиться, виберіть **Messages**. Двічі клацніть на елементі списку на панелі, щоб проаналізувати повідомлення про помилку.

Якщо принципова схема побудована правильно, на панелі **Messages** не повинно відображатися ніяких повідомлень. За наявності помилок вивчіть кожну з них і перевірте правильність з'єднань на схемі.

3.8. Створення друкованої плати

Для створення документу ДП клацніть ПКМ на імені проєкту у панелі **Projects** і виберіть **Add New to Project → PCB**. Новий документ плати з ім'ям **PCB1.PcbDoc** буде одразу відкритий та доданий до проєкту (відображатиметься у панелі **Projects** у каталозі **AmplifierOnTransistor**).

Збережіть документ ДП, обравши у меню **File** команду **Save As**. У вікні вибору каталогу для збереження файлу перейдіть у той каталог на жорсткому диску, в якому зберігається сам проєкт, у полі **File Name** введіть ім'я **AmplifierOnTransistor.PcbDoc** та натисніть кнопку **Save**.

Після цього необхідно виконати деякі початкові налаштування Редактора друкованих плат. Насамперед необхідно змінити систему одиниць вимірювань із дюймової на метричну. Для цього виконайте команду **Design → Board Options**. У вікні, що з'явиться, у полі **Measurement Unit** оберіть **Metric**.

В **Altium Designer** використовується [спеціальний об'єкт Редактора плат](#), який називається **Board Shape** або **Board Outline** (Контур плати). Цей об'єкт є замкненим полігоном, який визначає межі плати і який не належить до жодного з шарів плати. Саме цей об'єкт у вигляді чорного¹¹⁰ прямокутника ви бачите після створення нового документу плати (рис. 10.12). Початкові розміри цієї області визначаються глобальними налаштуваннями.



Рис. 10.12. Початкова область плати **Board Shape**

Однак кожен конкретний проєкт має власні розміри і форму контуру плати. Тому створений за замовчуванням об'єкт **Board Shape** має бути перевизначений таким чином, щоб він відповідав формі майбутньої плати¹¹¹. У розділі «Короткі

¹¹⁰ Насправді у цьому об'єкті відображається також сітка, проте зазвичай крок сітки значно менший за геометричні розміри області, тому її просто не видно. Якщо ж наблизити цю область, то ви обов'язково побачите сітку.

¹¹¹ Контур плати не обов'язково повинен створюватись саме на цьому етапі. Можна спочатку розташувати електронні компоненти у деякій локальній області у межах того розміру поля плати, який створюється автоматично і зазвичай є значно більшим, ніж потрібно для розміщення компонентів. Після розташування компонентів можна зрозуміти, які розміри повинна мати плата, після чого задати нові контури плати таким чином, щоб вони охоплювали усі компоненти.

теоретичні відомості про систему автоматизованого проєктування схем та друкованих плат Altium Designer» (підзаголовок 7.6) були описані декілька способів створення контуру плати. У цьому комп'ютерному практикумі ми будемо застосовувати підхід, що пов'язаний із використанням команди **Board Shape** → **Define from selected object** із меню **Design**.

Під час створення контуру плати необхідно виділити та використовувати окремий механічний шар, на якому не має бути ніяких примітивів, окрім ліній контуру. Ця умова є вкрай важливою! Оскільки новий контур плати має бути замкненим і відповідати формі плати, то наявність інших примітивів може призводити до виникнення помилок під час передавання даних на виробництво.

Що необхідно враховувати, обираючи механічний шар для розміщення ліній контуру плати?

Необхідно враховувати, які механічні шари вже використовуються у проєкті для тих чи інших потреб, на яких механічних шарах розміщуються тривимірні моделі корпусів, пояснювальні написи тощо. Наприклад, у попередніх комп'ютерних практикумах тривимірні моделі корпусів електронних компонентів ми розміщували на тринадцятому механічному шарі **Mechanical13**. Це означає, що якби ми у цьому комп'ютерному практикумі використовували електронні компоненти із власної бібліотеки, то вже не могли б обрати 13-й механічний шар для створення контуру плати. Ще важливо зауважити, що якщо для створення ПМ деякого електронного компоненту використовувався майстер **IPC Compliant Footprint Wizard**, то він автоматично розміщує тривимірну модель корпусу також на 13-му механічному шарі.

Якщо говорити про електронні компоненти із вбудованих бібліотек, то їх тривимірні моделі розміщуються на тому ж 13-му механічному шарі (а на 15-му шарі розміщується інформація про отвори КМ). Таким чином, оскільки у цьому комп'ютерному практикумі використовуються тільки електронні компоненти із вбудованих бібліотек, то ми можемо створити лінії контуру плати на будь-якому механічному шарі, окрім 13-го та 15-го.

Лінії контуру плати будемо розміщувати на механічному шарі **Mechanical4**. Виконайте команду **Design** → **Board Layers & Colors**. У вікні, що з'явиться, приберіть галочку із пункту **Only show enabled mechanical Layers** і у переліку шарів встановіть галочку для пункту **Enable** навпроти шару **Mechanical4**, обравши світло-зелений колір (рис. 10.13).

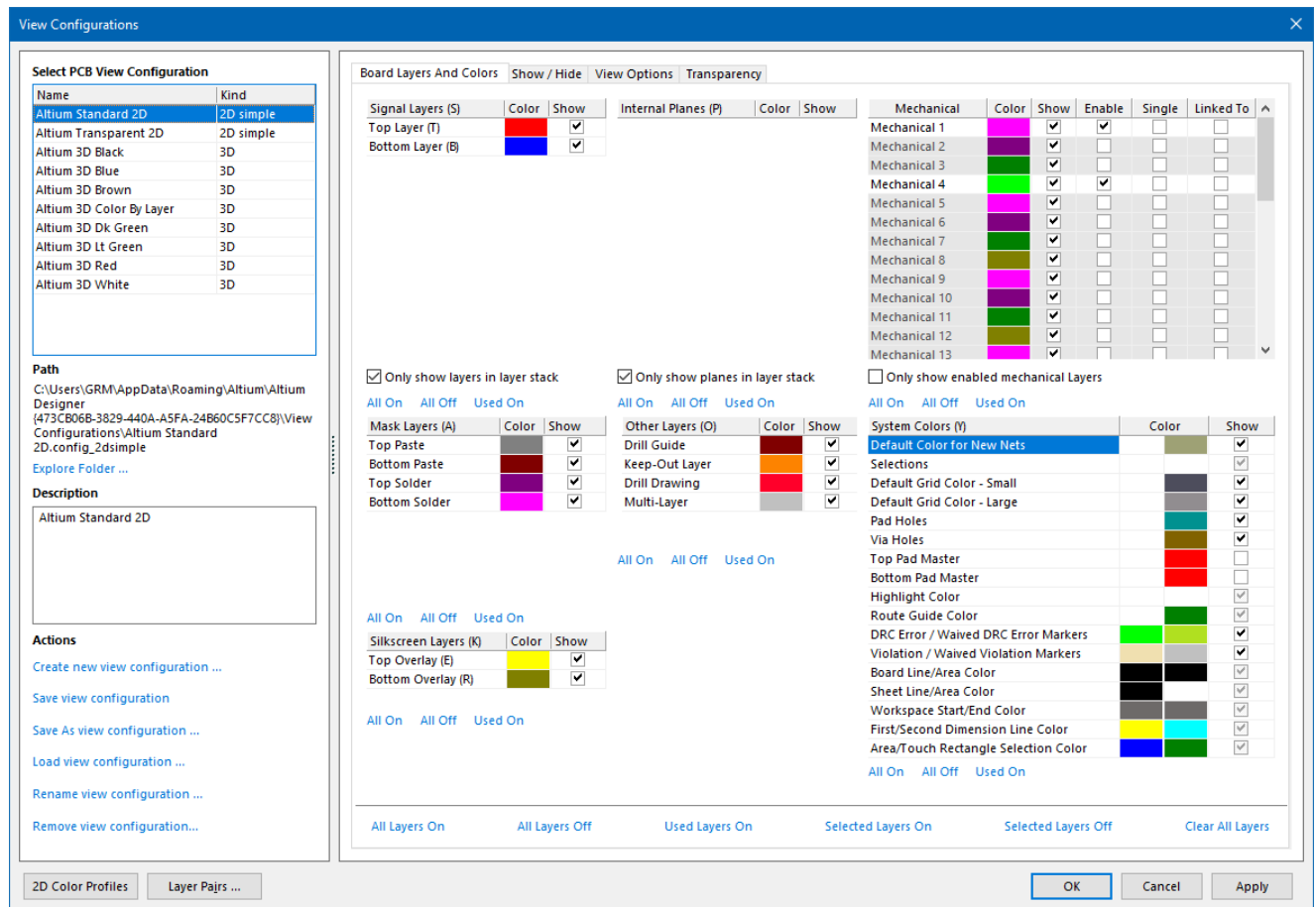



Рис. 10.13. Вікно налаштувань шарів плати

Тепер, встановивши активним шар **Mechanical4**, [створіть контур плати](#) у вигляді прямокутника довжиною 25 мм та висотою 20 мм. Спочатку задайте відповідний крок сітки (наприклад, 5 мм). Потім встановіть початок координат контуру плати. Для цього на панелі інструментів натисніть піктограму , виберіть пункт **Set Origin** та розмістіть точку початку координат ближче до нижнього лівого кута документу¹¹².

Після цього оберіть пункт **Place** → **Line** та починаючи від точки початку координат намалюйте прямокутний контур плати заданого розміру¹¹³. Товщина лінії контуру ДП може бути будь-якою.

¹¹² За замовчуванням для документів ДП нижній лівий кут аркушу є початком координат. Оскільки ми будемо із загальної площі аркушу використовувати лише деяку область для майбутньої плати, тому й встановлюємо новий початок координат у межах аркушу.

¹¹³ Контур можна малювати, орієнтуючись на координатну сітку і рахуючи кількість комірок сітки. Або можна під час малювання лінії натиснути клавішу <J>, після чого у меню, що з'явиться, обрати пункт **New Location**. У новому вікні **Jump To Location** можна задати координати наступної точки, до якої має бути проведена лінія. Після натискання кнопки **OK** лінія буде прокладена до поточної точки. Якщо продовжити вести лінію із цієї точки, то можна

Зауваження!

Використовуючи під час створення контуру комбінацію клавіш <Shift + Пробіл> можна змінювати кут малювання лінії.

Після створення контуру візуально переконайтесь, що він замкнений. Для цього важливо перевірити кутові та інші точки з'єднання, використовуючи інструменти для збільшення масштабу.

Далі за допомогою мишки виділіть створений контур плати, перейдіть у меню *Design* та оберіть пункт *Board Shape* → *Define from selected object*. Після цього робоча область для розміщення компонентів зменшиться до заданих розмірів¹¹⁴, які визначаються лініями контуру.

Важливо!

Обраний механічний шар не міститиме самого об'єкту *Board Shape*, на ньому будуть намальовані тільки лінії, із яких сформується новий контур в об'єкті *Board Shape*. Фактично, механічний шар використовується лише як проміжна ланка, оскільки десь же треба намалювати лінії. Після того, як ви за допомогою цих ліній сформували новий контур, то можете спокійно їх видалити і використовувати обраний механічний шар для інших потреб. Однак може бути принаймні дві причини залишити лінії контуру на механічному шарі. По-перше, деякі виробники вимагають, щоб контур плати був на окремому механічному шарі, який повинен імпортуватись у Gerber-файл¹¹⁵. По-друге, ви можете залишити лінії на обраному механічному шарі, щоб, наприклад, використовувати їх під час формування складального креслення.

знову натиснути клавішу <J> і виконати перелічені раніше дії, вказавши координати наступної точки тощо.

¹¹⁴ Якщо ж згодом виникне необхідність змінити контури плати, то робиться це методом перетягування поточних меж плати та повторного виконання команди *Board Shape* → *Define from selected object* із меню *Design*.

¹¹⁵ Інший варіант формування контуру плати для виробника, щоб він міг її виготовити, полягає у використанні файлу для програми свердління, який створюється за допомогою команди *File* → *Fabrication Outputs* → *NC Drill Files*.

3.9. Розміщення компонентів на друкованій платі

Перед тим, як передати дані принципової схеми у Редактор друкованих плат, спочатку відкрийте документ принципової схеми **AmplifierOnTransistor.SchDoc**. У меню **Design** виберіть **Update PCB Document AmplifierOnTransistor.PcbDoc**. Почнеться процес компіляції, після завершення якого відкриється діалогове вікно **Engineering Change Order** (рис. 10.14).

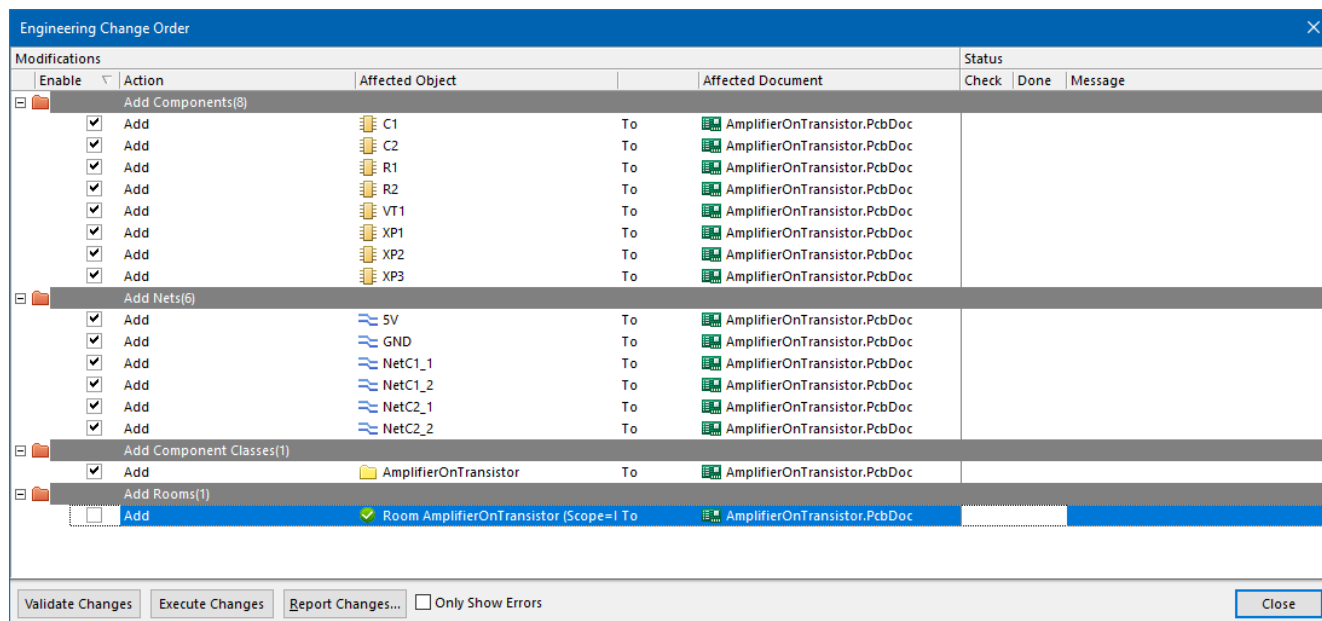


Рис. 10.14. Діалогове вікно **Engineering Change Order**

У цьому вікні приберіть галочку навпроти пункту **Add Room AmplifierOnTransistor**¹¹⁶.

Натисніть кнопку **Validate Changes**. Якщо усі зміни коректні, навпроти кожної зміни у графі **Status** має з'явитися зелений прапорець. Якщо зміни не пройшли перевірку, закрийте діалогове вікно, перегляньте вміст панелі **Messages** і усуньте помилки.

Натисніть **Execute Changes**, щоб внести зміни у ДП. Після завершення процесу всі рядки позначаються зеленим прапорцем в стовпчику **Done**. Після

¹¹⁶ Під час передачі інформації від редактора принципової схеми у редактор плат програма **Altium Designer** автоматично формує «кімнати» (**rooms**) на платі (**прямокутна область червоного кольору праворуч від контуру плати**), якщо це не відключено у налаштуваннях. «Кімнати» – це області плати, за якими закріплені деякі групи компонентів. «Кімнати» формуються відповідно до аркушів схеми, і найбільш доцільно застосовувати їх у тих випадках, коли є повторювані фрагменти схеми, тому що, зробивши трасування в одній «кімнаті», його можна продублювати в інших.

натискання кнопки **Close** відкриється документ ДП з компонентами, що будуть розташовані праворуч за межами плати. Натисніть послідовно клавіші <V, D> (*View* → *Fit Document*), якщо компоненти не відображаються повністю у робочій області (рис. 10.15).

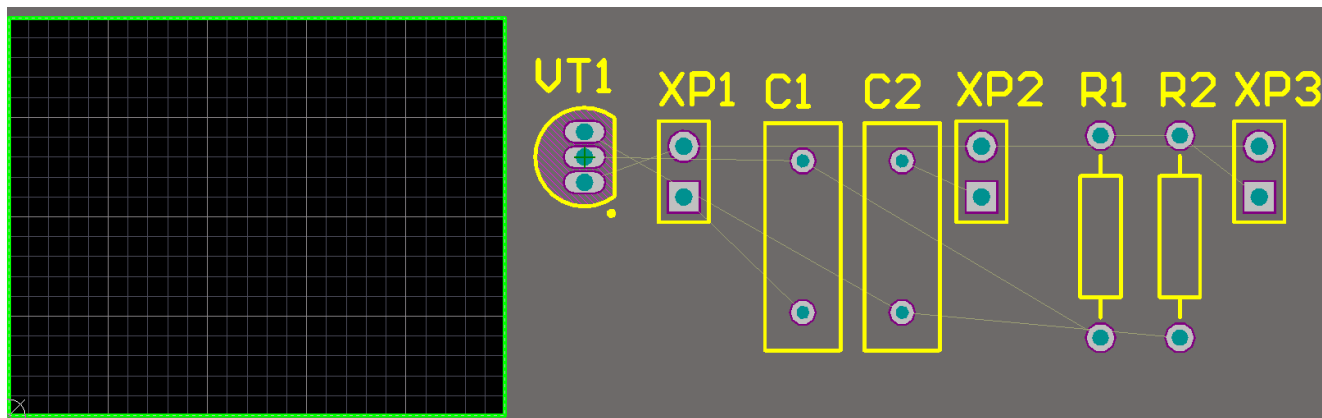


Рис. 10.15. Новостворений документ плати з нерозміщеними компонентами

Тепер необхідно розмістити усі компоненти у межах контуру плати. Це виконується проєктувальником на основі правил, рекомендацій, вимог, досвіду тощо. У деякому сенсі цей процес є творчим.

Розмістіть компоненти на платі приблизно так, як це показано на рис. 10.16.

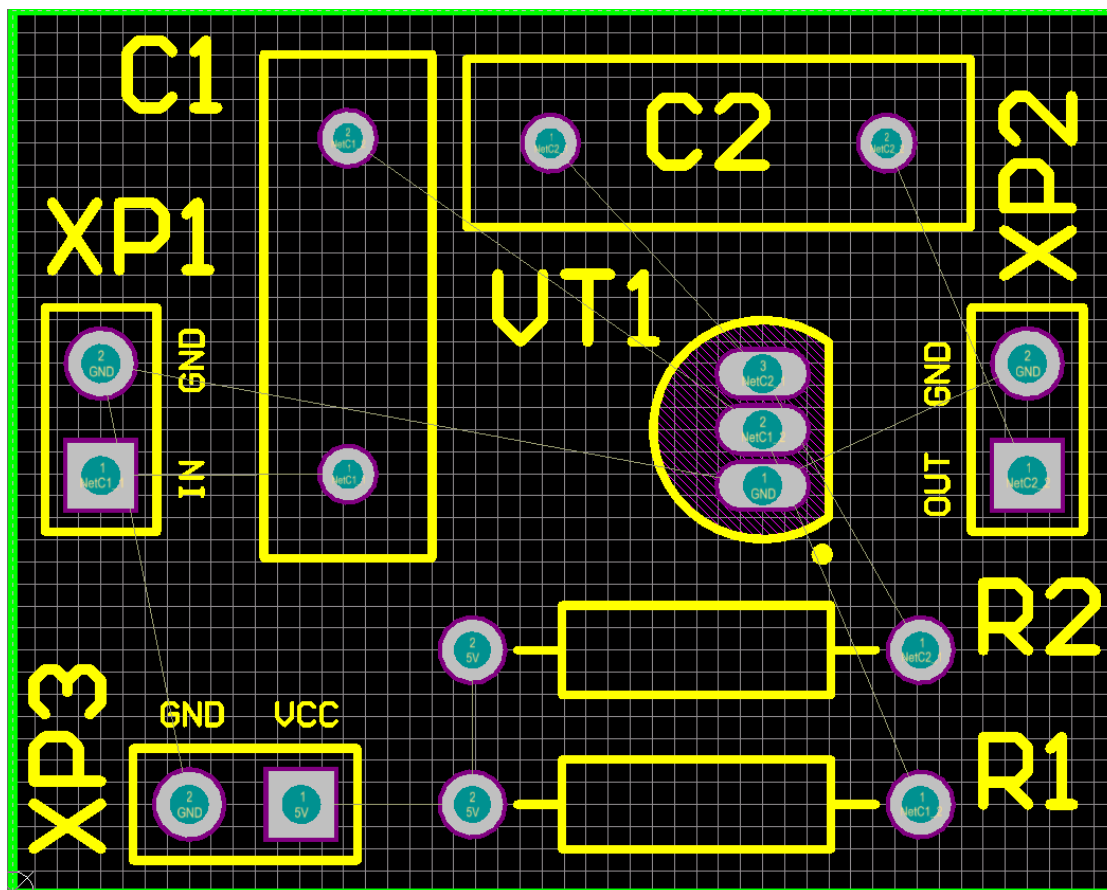


Рис. 10.16. Приклад розміщення компонентів на платі

Зверніть увагу, що символи позиційних позначень деяких електронних компонентів були зміщені вручну таким чином, щоб їх зручно було читати та щоб вони не накладались на контури ПМ і не виходили за контури плати¹¹⁷.

Ще один нюанс полягає у тому, що розміщені на платі пасивні компоненти не мають номіналів, а транзистор та роз'єми не мають назв. Це не є чимось неправильним, а залежить від розробника і налаштувань. Яким чином відображати номінали та назви електронних компонентів, ви можете ознайомитись у цьому [відео](#).

Також ви можете помітити, що були додані деякі спеціальні позначення на шарі шовкографії **Top Overlay** (зокрема, **IN**, **OUT**, **VCC**, **GND**). Зроблено це за допомогою команди **Place** → **String**. Під час розміщення цих написів були використані наступні геометричні характеристики (рис. 10.17): ширина символу – 0,1 мм, висота – 0,5 мм. Такі додаткові пояснювальні написи на шарі шовкографії є вкрай зручними, оскільки допомагають чітко зрозуміти призначення кожного роз'єму та окремого його виводу.

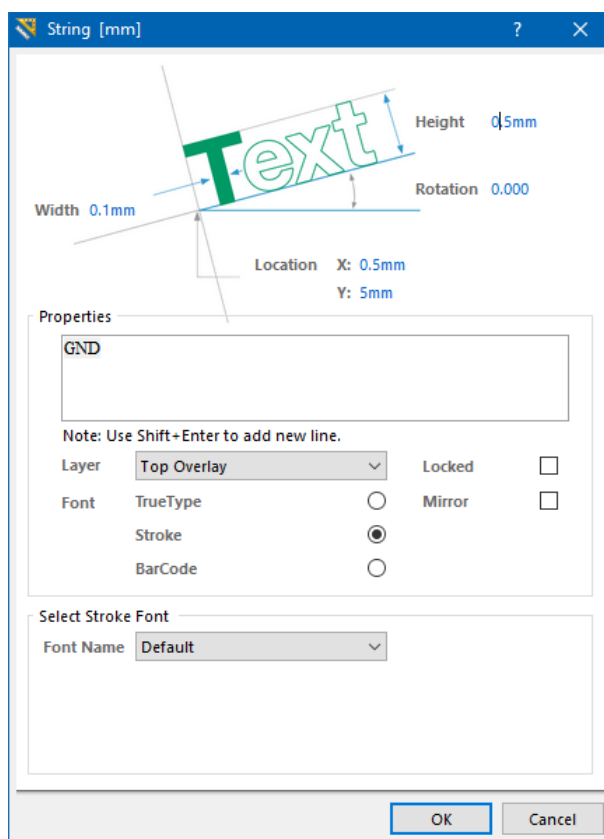


Рис. 10.17. Вікно властивостей текстових позначень

¹¹⁷ У цьому прикладі символи позиційних позначень мають розміри за замовчуванням, і можна помітити, що вони дещо звеликі відносно розмірів цієї плати. Розміри символів позиційних позначень можна у будь-який момент відрегувати таким чином, щоб вони займали оптимальне місце на платі. Для цього необхідно [подвійно клацнути ЛКМ на відповідному позиційному позначенні та у вікні властивостей, що відкриється, змінити його розміри](#).


3.10. Трасування плати в інтерактивному режимі

Трасування – це процес прокладання провідників і перехідних отворів на платі з метою електричного з'єднання компонентів. **Altium Designer** пропонує два режими трасування: ручне та автоматичне. У цьому комп'ютерному практикумі буде розглянуто режим інтерактивного ручного трасування плати.

Під час інтерактивного трасування необхідно навчитись використовувати команди і комбінації гарячих клавіш, що описані у розділі «**Короткі теоретичні відомості про систему автоматизованого проектування схем та друкованих плат Altium Designer**» (підзаголовок 7.1).

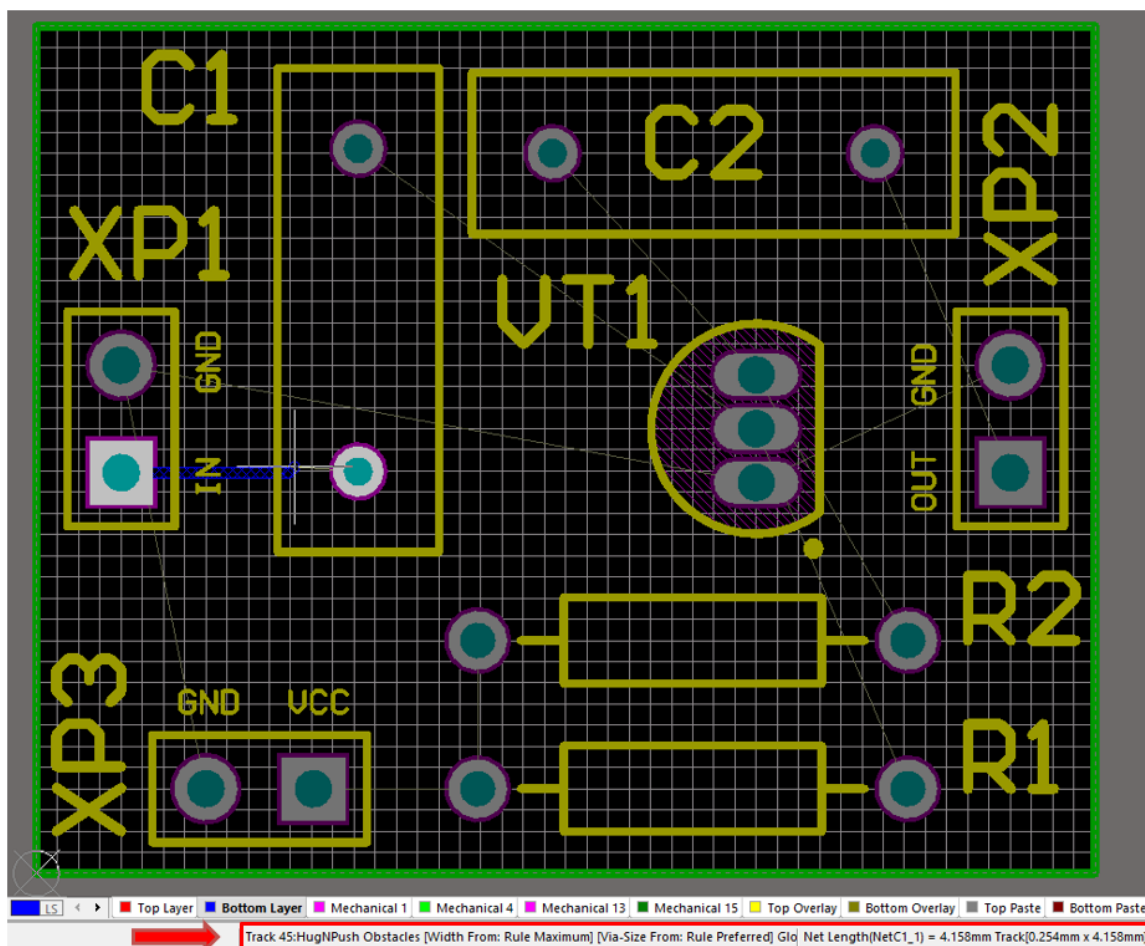
Важливо!

Траси прокладаються тільки на шарах *Top Layer*, *Bottom Layer* або *Internal Layer* (цей шар використовується тільки в проектах багат шарових плат). Тому спочатку необхідно зробити активним один із цих шарів, після чого починати інтерактивне трасування. Розміщений сегмент траси буде відображатись у кольорі відповідного шару.

Оскільки цей проєкт є досить невеликим і у ньому використовуються виключно компоненти для монтажу в отвори, то усе трасування можна виконати на одному шарі, наприклад, *Bottom Layer*. Тому зробіть саме цей шар активним, після чого виконайте команду *Route* → *Interactive Routing* або натисніть піктограму  на панелі інструментів. Виберіть початковим перший КМ (має прямокутну форму) роз'єму **XP1** та почніть вести трасу від нього до першого КМ конденсатора **C1** (рис. 10.18). Під час прокладання траси натисніть кілька разів комбінацію клавіш <Shift + R> до появи режиму *HugNPush Obstacle*. Назва режиму відображатиметься у рядку стану під вкладками шарів або у спливаючій інформації в лівому верхньому кутку робочої області (див. рис. 10.18).

Зауваження!

Неможливо випадково з'єднати трасою КМ, які не зв'язані між собою. Зв'язані між собою компоненти позначаються тонкою світлою прямою лінією на платі. **Altium Designer** постійно аналізує зв'язність компонентів плати і перешкоджає створенню невірних трас або їх перетину.

Рис. 10.18. Інтерактивне трасування плати у режимі *HugNPush Obstacle*

Далі з'єднайте усі виводи компонентів відповідно до того, як вони були з'єднані на принциповій схемі¹¹⁸. Остаточний результат зазвичай має відповідати тому, який представлений на рис. 10.19¹¹⁹.

Зверніть увагу, що траси на ДП формуються з послідовності прямих сегментів. Під час кожної зміни напрямку починається новий сегмент. Крім того, за замовчуванням в **Altium Designer** провідники можна розташовувати вертикально, горизонтально або під кутом в 45°.

Також ви можете помітити, що траси провідників живлення та землі мають більшу ширину, ніж інші провідники. Більша ширина траси призводить до зменшення її опору, що є важливим для провідників цього типу.

¹¹⁸ Рекомендується ознайомитись із відео «[Курс з Altium Designer. Частина 5](#)», щоб краще зрозуміти, як виконується ручне трасування.

¹¹⁹ Якщо ви розмістили компоненти на платі дещо інакше, то ви можете отримати інший провідний малюнок на платі. Це не є проблемою. Головне, щоб трасування плати не містило помилок.

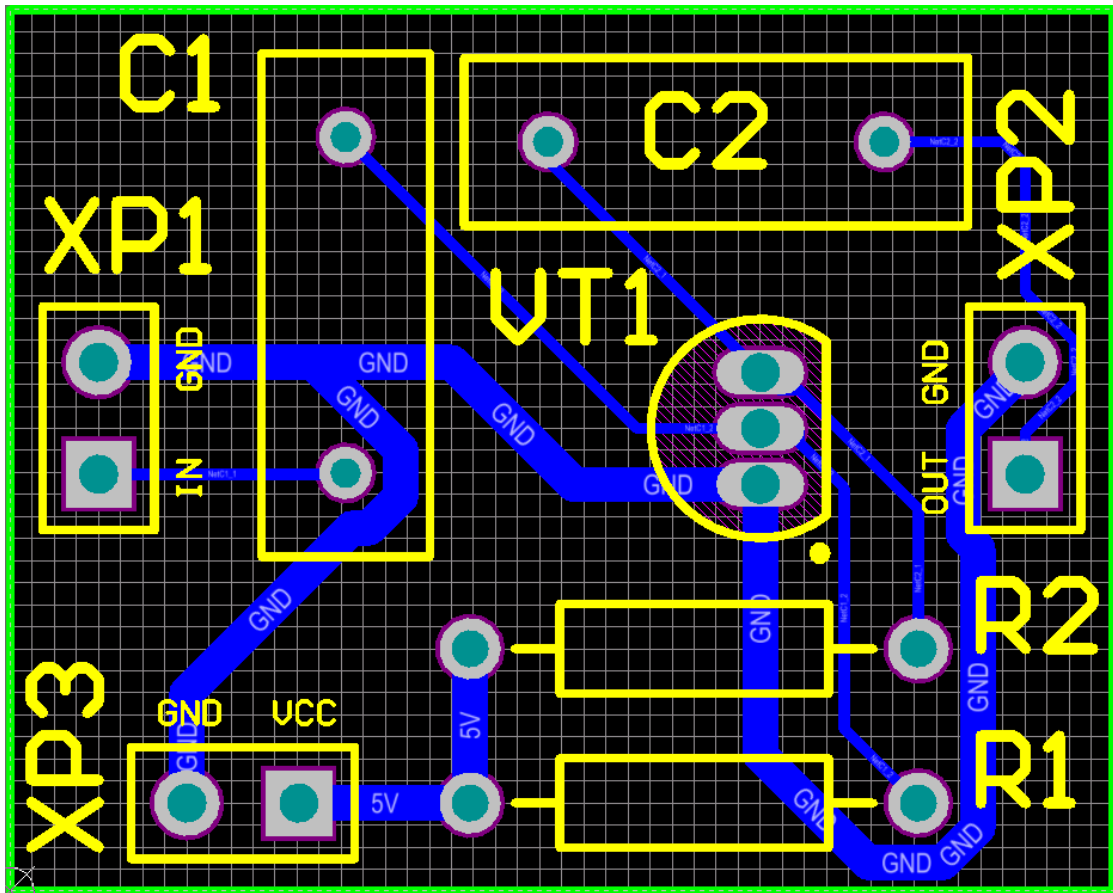


Рис. 10.19. Плата із розміщеними компонентами та прокладеними трасами

Для того, щоб змінити ширину доріжки під час інтерактивного трасування, необхідно натиснути клавішу <Tab>. Це призведе до відкриття вікна властивостей поточної траси (рис. 10.20).

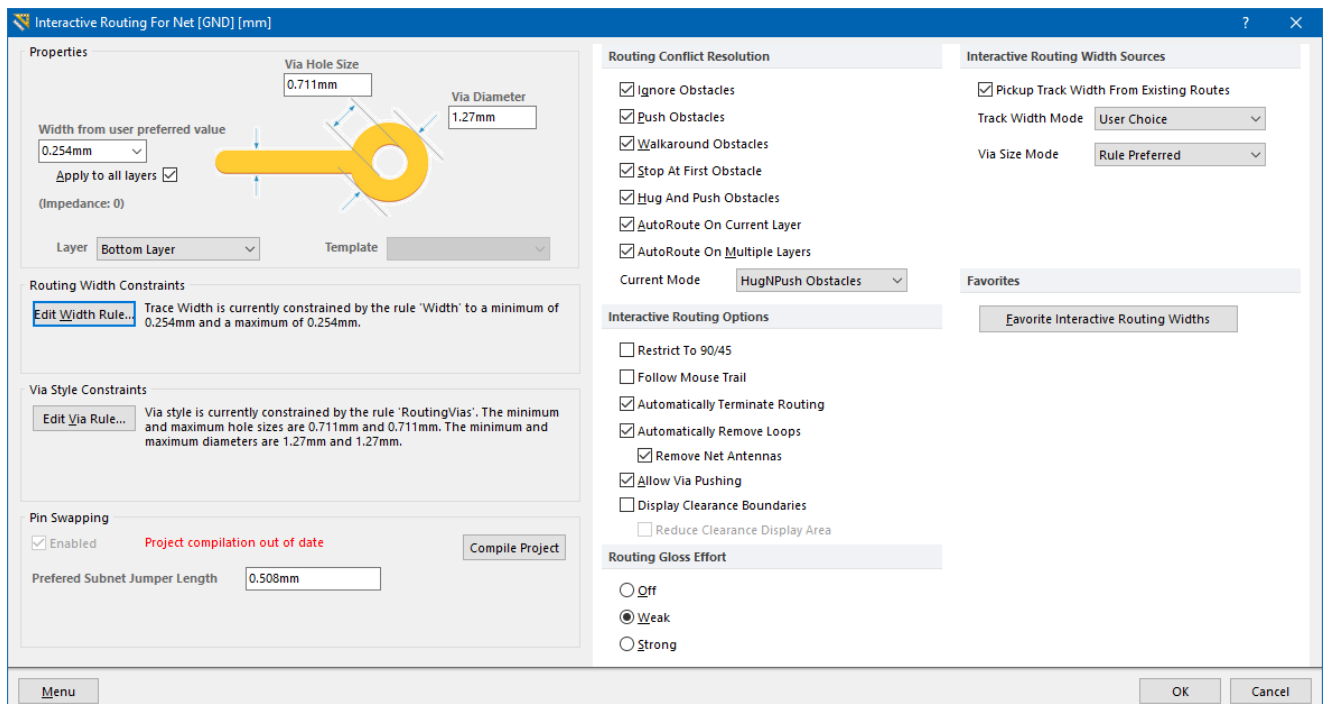


Рис. 10.20. Вікно властивостей траси

Спочатку в області **Routing Width Constraints** натисніть кнопку **Edit Width Rule**. У новому вікні, що відкриється, задайте параметр **Max Width** рівним 0,8 мм (рис. 10.21).

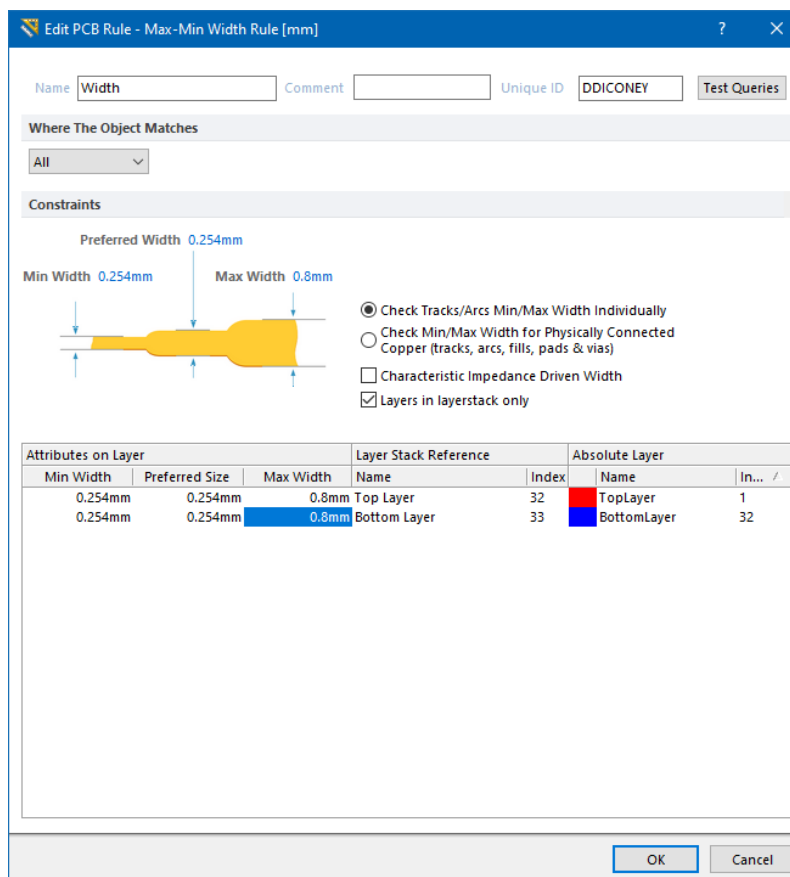


Рис. 10.21. Вікно властивостей, у якому задається діапазон зміни ширини трас

Після встановлення необхідних значень і закриття вікна **Edit Width Rule** ви знову повернетесь до вікна властивостей траси. Тепер у цьому вікні необхідно в області **Properties** задати значення параметру **Width from user preferred value** рівним 0,8 мм. У результаті ширина поточної траси зміниться.

Очевидно, що після того як буде прокладена траса заданої ширини, для наступної траси, якщо вона повинна мати іншу ширину, доведеться знову викликати вікно властивостей траси і встановлювати нове значення. Проте можна використовувати інший підхід до вибору ширини траси, із яким ви можете ознайомитись у цьому [відео](#) (дивитись із 3:40 до 4:20).

3.11. Візуальний аналіз плати у 2D режимі

Після того, як було виконане трасування плати, бажано візуально її проаналізувати. Для цього використовують режим збільшення / зменшення

масштабу, щоб можна було оцінити провідний малюнок загалом, а за потреби – переглянути окрему ділянку плати у збільшеному масштабі.

Зручним інструментом під час аналізу правильності прокладання трас є використання клавіші <Shift>. Якщо затиснути клавішу <Shift> і одночасно наводити курсор мишки на ту чи іншу трасу, то вона починає підсвічуватись. Це дає змогу чітко візуально виокремити цю трасу і подивитись, чи з'єднує вона усі необхідні КМ на платі.

3.12. Видалення сегментів трас та їх повторне прокладання

Після прокладання траси і її візуального аналізу може виникнути необхідність видалення деякого сегмента траси. Для цього натисніть ЛКМ на потрібному сегменті для того, щоб його виділити, після чого натисніть для видалення обраного сегменту траси. Тепер знову увімкніть режим інтерактивного трасування та заново створіть сегмент.

Іноколи може виникнути потреба повністю перетрасувати деяке з'єднання між КМ, а не окремі його сегменти. Якщо почати видаляти окремі сегменти деякого з'єднання, то це неефективно і займе багато часу. В **Altium Designer** прокладання нової траси замість старої робиться інакше. Необхідно увімкнути режим інтерактивного трасування і, не видаляючи попереднє з'єднання, натиснути ЛКМ у центрі першого КМ, від якого почнеться прокладання траси заново, потім прокласти трасу за новим маршрутом до наступного КМ. Після того, як буде натиснута ЛКМ на кінцевому КМ, старі сегменти траси автоматично повністю видаляться, а залишиться тільки нова траса.

3.13. Створення складального кресленика друкованої плати та креслеників окремих шарів

Під час підготовки [документації на вузол ДП](#) важливим завданням є створення складального кресленика та креслеників окремих шарів ДП.

Є кілька варіантів, яким чином можна створювати документацію на розроблену ДП. Розглянемо варіант із використанням файлу **Output Job** (або **OutJob**). По суті, цей файл є попередньо налаштованим набором команд, які зібрані в одному файлі та можуть бути виконані у пакетному режимі. Для роботи з цими файлами в **Altium Designer** є вбудований редактор під назвою **OutputJob**

Editor. Для того, щоб відкрити цей редактор, необхідно виконати команду **File** → **New** → **Output Job File**. Після цього файл **Output Job** буде доданий до проєкту (рис. 10.22) та відкритий у новій вкладці (рис. 10.23).

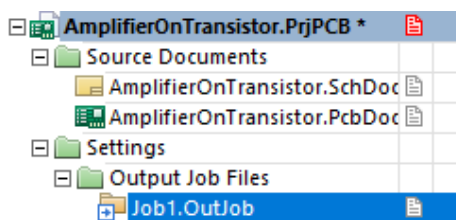


Рис. 10.22. Відображення файлу **Output Job** на панелі **Projects**

Натисніть на імені файлу **Out1.OutJob** у дереві проєктів та збережіть його під іменем **AmplifierOnTransistor.OutJob**.

Зверніть увагу, що редактор файлів Output Job розділено на три області (рис. 10.23):

- **Variant Choice** (вибір варіанту);
- **Outputs** (вихідні файли);
- **Output Containers** (контейнери вихідних файлів).

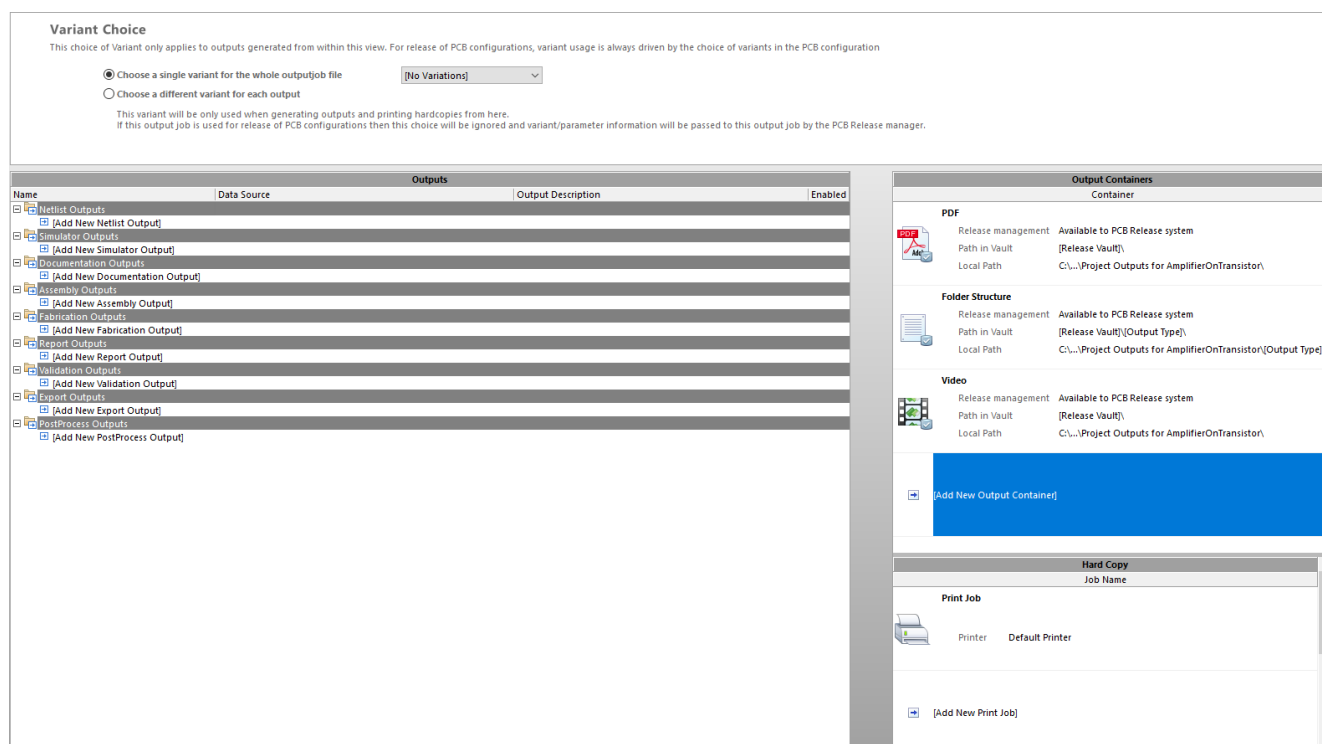


Рис. 10.23. Робоча область файлу **Output Job**

Вихідні файли, які можна додати до проєкту, відображаються у вікні **Outputs** та згруповані за категоріями, наприклад: **Netlist Outputs**, **Documentation Outputs**,

Assembly Outputs, Fabrication Outputs тощо. У правій частині редактора **OutputJob Editor** знаходяться контейнери вихідних файлів, які визначають розташування та формат вихідних документів, що будуть сформовані. Новий файл **Output Job** за замовчуванням містить три контейнери: PDF, Folder Structure та Video.

Перейдіть у робочу область файлу **Output Job** та знайдіть розділ **Documentation Outputs**, у якому натисніть на рядку **Add New Documentation Outputs** та у випадному меню оберіть **PCB Prints** → **AmplifierOnTransistor.PcbDoc** (рис. 10.24).

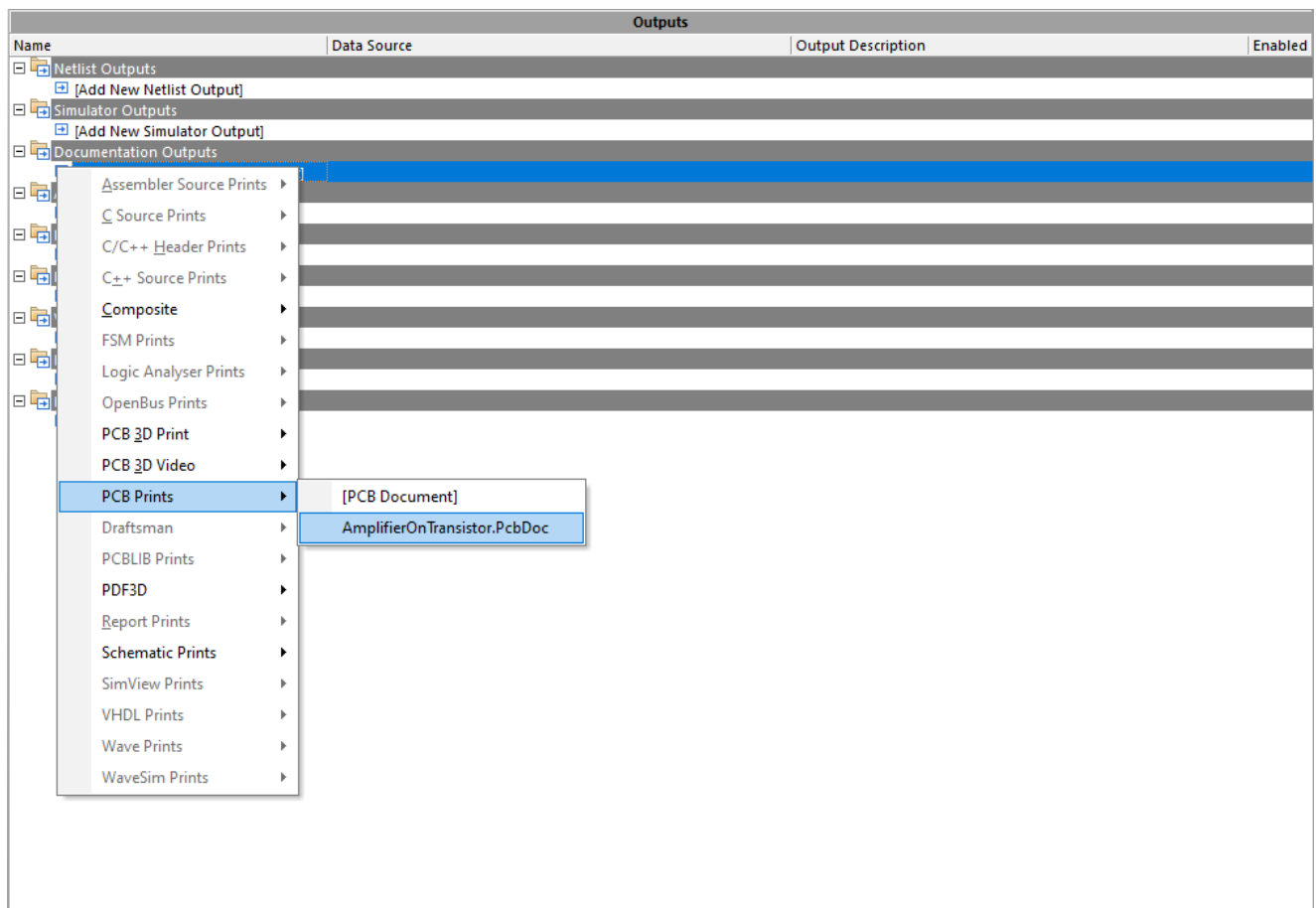


Рис. 10.24. Додавання нового елементу, який надалі буде використаний для формування документації

Після цього в області **Output Containers** оберіть PDF та активуйте прапорець у стовпчику **Enabled** для обраного файлу **AmplifierOnTransistor.PcbDoc** (рис. 10.25), що дозволить друкувати необхідні документи проєкту плати у форматі PDF.



Рис. 10.25. Завдання можливості друку документів у форматі PDF

Далі в області **Output Containers** у контейнері **PDF** натисніть напис **Generate Content**, після чого буде створений PDF-файл, у якому на одному аркуші різними кольорами будуть надруковані усі шари ДП.

Однак інколи потрібно друкувати окремі шари плати. Для цього викличте вікно налаштувань, подвійно клацнувши ЛКМ на рядку **PCB Prints AmplifierOnTransistor.PcbDoc**. У новому вікні натисніть ПКМ на рядку **Multilayer Composite Print** та у випадному меню оберіть **Create Final** (рис. 10.26). У результаті з'явиться можливість друкувати окремі шари плати на окремих аркушах PDF-файлу.

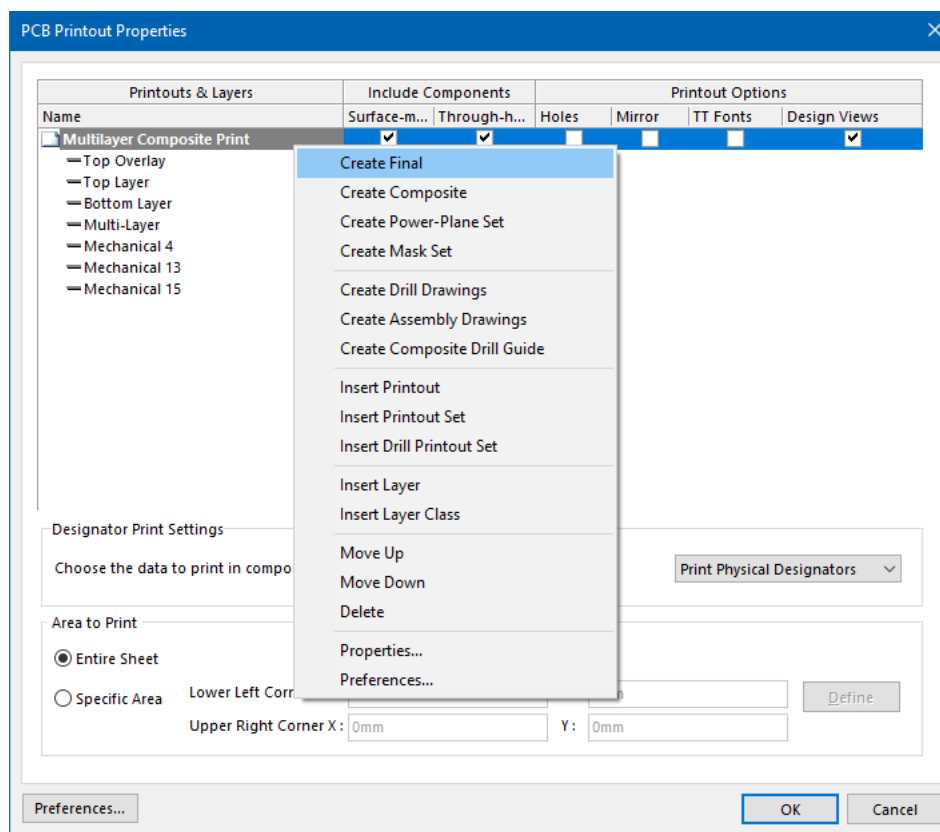


Рис. 10.26. Вікно налаштувань **PCB Printout Properties**

Якщо ж подвійно клацнути на назві того чи іншого шару, то відкриється додаткове вікно, у якому можна зазначити, що конкретно потрібно і не потрібно відображати на шарі під час роздруківки.

3.14. Синхронізація змін між принциповою схемою та друкованою платою

Важливо розуміти, що під час роботи над будь-яким проектом, у ньому можливі зміни. Причому зміни можуть вноситись як у схему, так і плату. У такому випадку виникає серйозне запитання: яким чином підтримувати синхронізацію даних між схемою та платою, щоб були враховані усі внесені зміни?

В Altium Designer для синхронізації використовується модуль порівняння (або компаратор). Синхронізація може бути як двосторонньою (одночасно між схемою і платою), так односторонньою (від схеми до плати або від плати до схеми).

Розглянемо, як це працює, на наступному прикладі. Нехай внаслідок певних причин у створеній нами принциповій схемі (див. рис. 10.8) потрібно, щоб перший вивід роз'єму **XP3** підключався до землі, а другий – до живлення (наразі усе навпаки). Також припустімо, що необхідно змінити нумерацію роз'ємів таким чином: **XP3** → **XP2**, **XP2** → **XP3**.

Внесіть необхідні виправлення у принципову схему методом видалення старих з'єднань і створенням нових, а також відредагуйте властивості роз'ємів, змінивши їх позиційне позначення. Результат представлено на рис. 10.27. Зверніть увагу, що ланцюг **5V**, який іде до виводу 2 роз'єму **XP2** перетинається із ланцюгом **GND**, який іде до виводу 1 того ж роз'єму, але тільки візуально (не фізично).

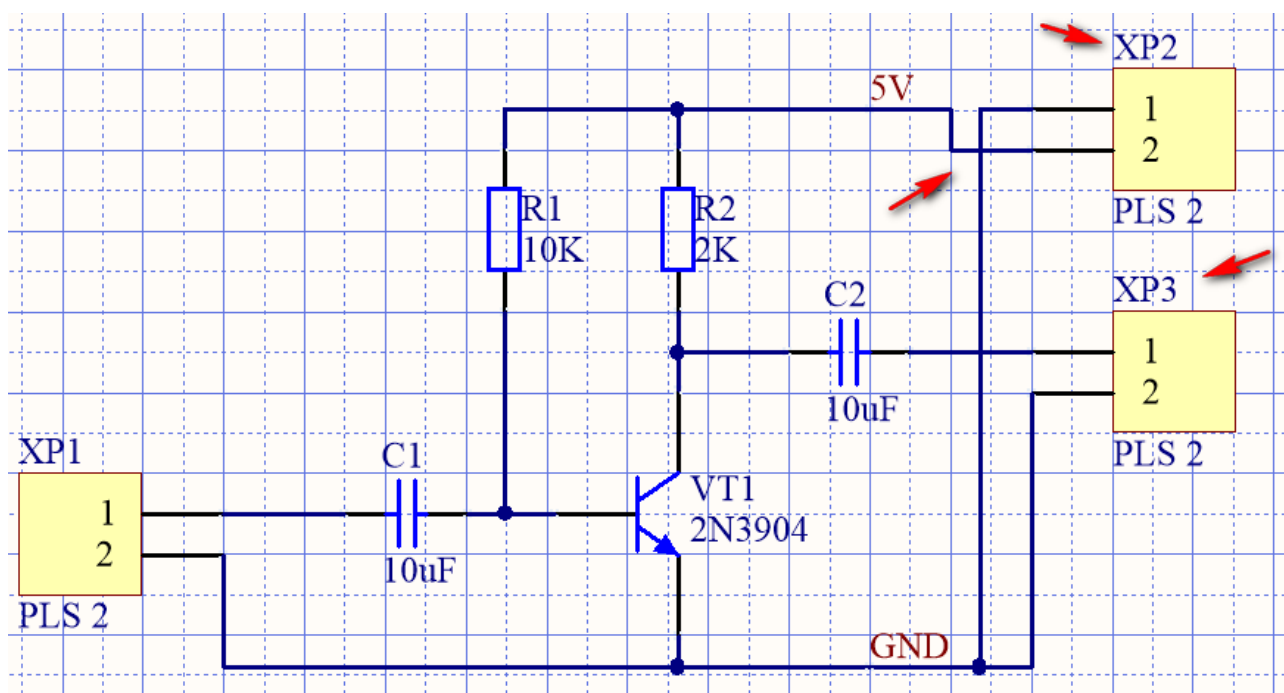
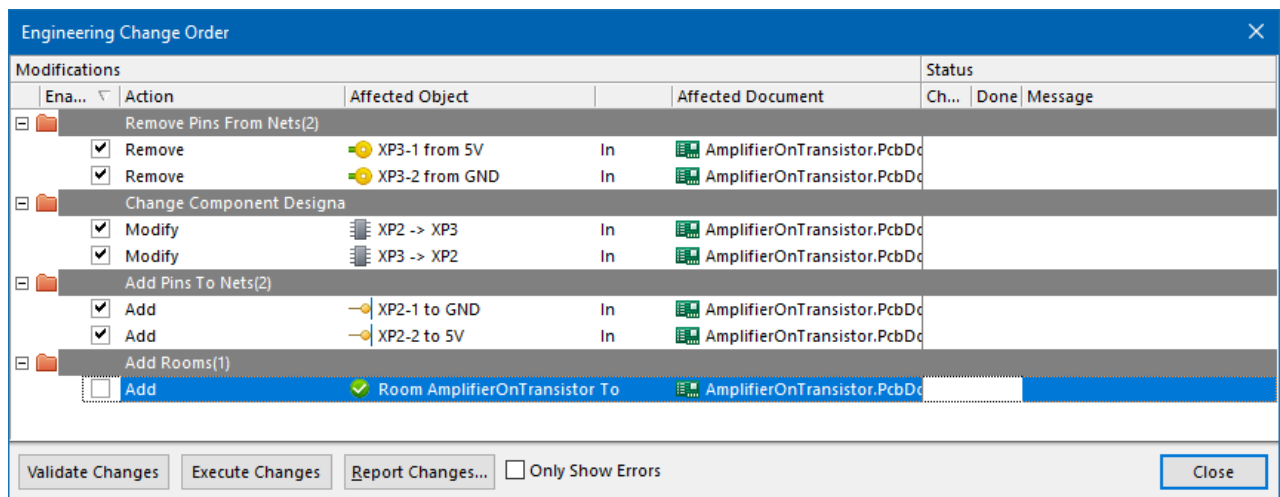


Рис. 10.27. Модифікована схема підсилювача (червоними стрілками показані зміни)

Тепер із меню **Design** виконайте команду **Update PCB Document AmplifierOnTransistor.PcbDoc**. З'явиться вікно **Engineering Change Order** (рис. 10.28), у якому приберіть галочку у рядку **Add Rooms**. Натисніть спочатку кнопку **Validate Changes** і після того, як у стовпчику **Check** з'являться усі зелені галочки, натисніть кнопку **Execute Changes**. Після цього закрийте вікно **Engineering Change Order**.

Рис. 10.28. Вікно *Engineering Change Order*

Після застосування усіх змін плата виглядатиме так, як показано на рис. 10.29.

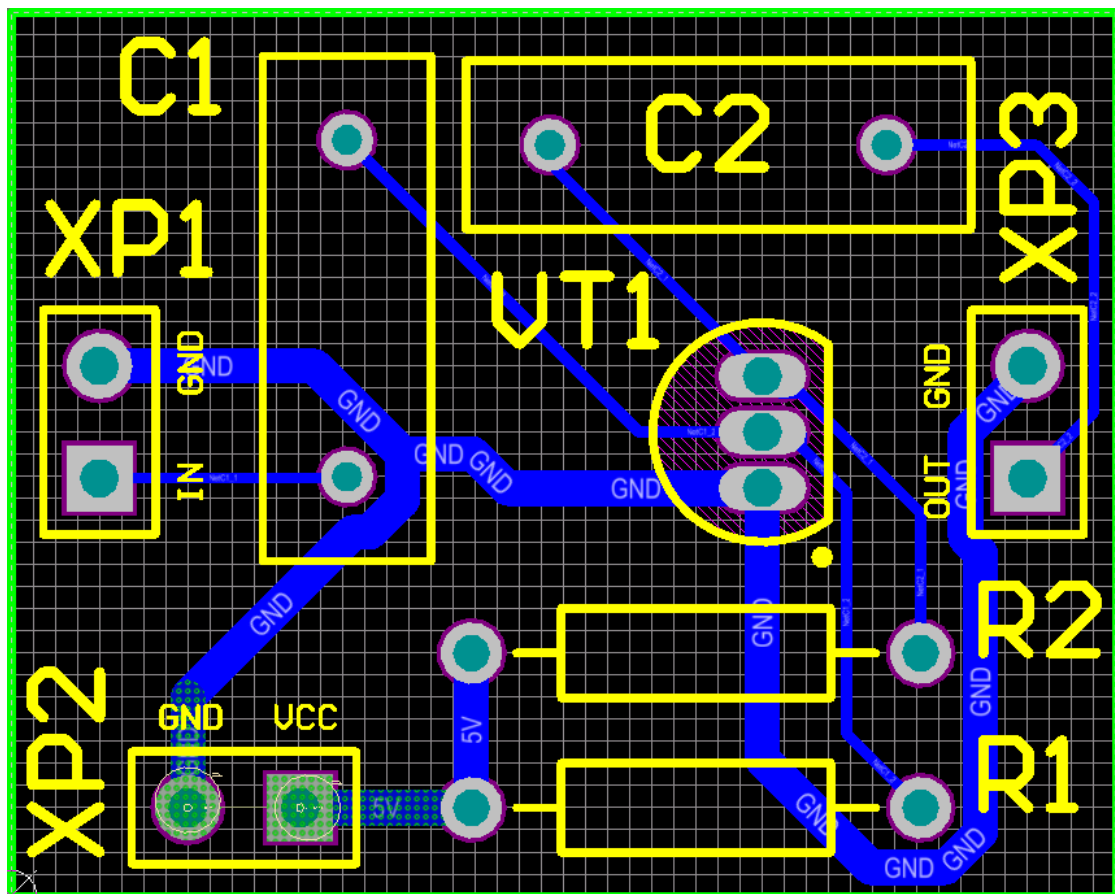


Рис. 10.29. Плата із внесеними змінами

Якщо докладно придивитись до ПМ роз'єму **XP2**, то можна помітити помилку, яка виникла із-за того, що на принциповій схемі були змінені місцями виводи, які підключались до ланцюгів із мітками **5V** та **GND**. Виправити це можна видаленням старих трас і прокладанням нових. А можна зробити інакше – роз'єм **XP2** просто дзеркально відобразити по горизонталі (рис. 10.30).

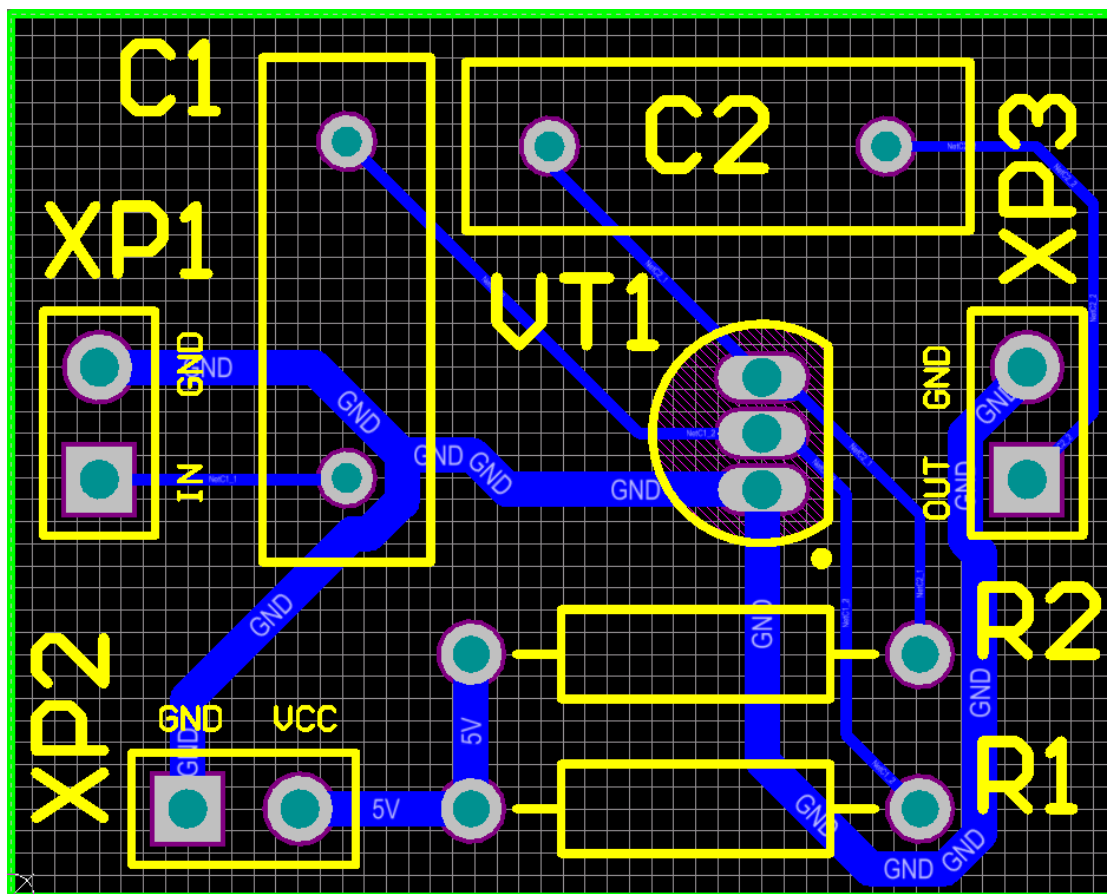


Рис. 10.30. Кінцевий варіант плати, що синхронізована зі змінами, які були внесені у принципову схему

Звичайно, тепер дещо нелогічно виглядає підключення трас землі до роз'ємів **XP1**, **XP3** та **XP2**, а саме: у першого та третього роз'єму земля підведена до КМ круглої форми, а у другого – до квадратної. Проте це обумовлено перепідключенням з'єднань на схемі. Цей приклад був більше демонстраційний, тому якихось узгоджень по формі КМ ми не дотримувались. Тим паче з електричної точки зору форма КМ не впливає на проходження сигналу.

4. Питання для самоконтролю

1. Поясніть принцип роботи схеми підсилювача із загальним емітером.
2. Яким чином забезпечується термостабілізація схеми підсилювача із загальним емітером?
3. Що містить у собі файл проєкту **Altium Designer**?
4. Яким чином створити новий проєкт у **Altium Designer**?
5. Яким чином налаштовуються параметри проєкту?

6. Поясніть, що таке матриця з'єднань?
7. Яким чином додати у створений проєкт принципову схему та друковану плату?
8. Яким чином знайти необхідний електронний компонент у бібліотеці та розмістити його на аркуші принципової схеми?
9. Яким чином виконується з'єднання виводів компонентів між собою? Що необхідно враховувати під час процесу з'єднання?
10. Для чого використовуються мітки на ланцюгах електричних з'єднань? Яким чином вони встановлюються?
11. Яким чином виконується компіляція проєкту та перевірка на наявність помилок?
12. Яким чином передати дані від Редактора принципової схеми у Редактор плат?
13. Поясніть, що таке об'єкт **Board Shape**?
14. Яким чином створити контур плати у Редакторі плат?
15. Поясніть, чи потрібно залишати лінії контуру плати на обраному механічному шарі?
16. Яким чином виконується інтерактивне ручне трасування у Редакторі плат? Опишіть режими ручного трасування.
17. Яким чином встановити необхідну ширину траси?
18. Яким чином видаляти сегменти траси? Яким чином прокласти трасу заново?
19. Яким чином створюється складальний кресленик друкованої плати та кресленики окремих шарів?
20. Яким чином відбувається синхронізація та обмін даними між Редактором принципової схеми та Редактором плати?

Комп'ютерний практикум № 11.

РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ ТА ДРУКОВАНОЇ ПЛАТИ ГЕНЕРАТОРА ТАКТОВИХ ІМПУЛЬСІВ НА БАЗІ МІКРОСХЕМИ ТАЙМЕРУ 555

Мета роботи: розробити принципову схему генератора тактових імпульсів на базі мікросхеми таймеру 555, використовуючи електронні компоненти із власної бібліотеки; на основі розробленої принципової схеми створити друковану плату; виконати автоматичне трасування плати та виправити помилки, які можуть виникнути при цьому.

1. Теоретичні відомості

Генератор тактових імпульсів (або генератор тактової частоти) призначений для синхронізації різних процесів в цифрових пристроях – починаючи від найпростіших електронних годинників і закінчуючи комп'ютерами. Генератор тактових імпульсів (ГТІ) виробляє електричні імпульси (зазвичай прямокутної форми) заданої частоти, яка часто використовується як еталонна – рахуючи кількість імпульсів, можна, наприклад, вимірювати часові інтервали.

Існує величезна кількість реалізацій схем ГТІ: генератори на основі RC-ланцюгів та логічних елементів, генератори на основі кварцових резонаторів та логічних елементів, генератори на основі спеціалізованих мікросхем тощо.

На рис. 11.1 представлена принципова схема ГТІ у режимі вільних коливань, в якій основним елементом є мікросхема таймеру 555.

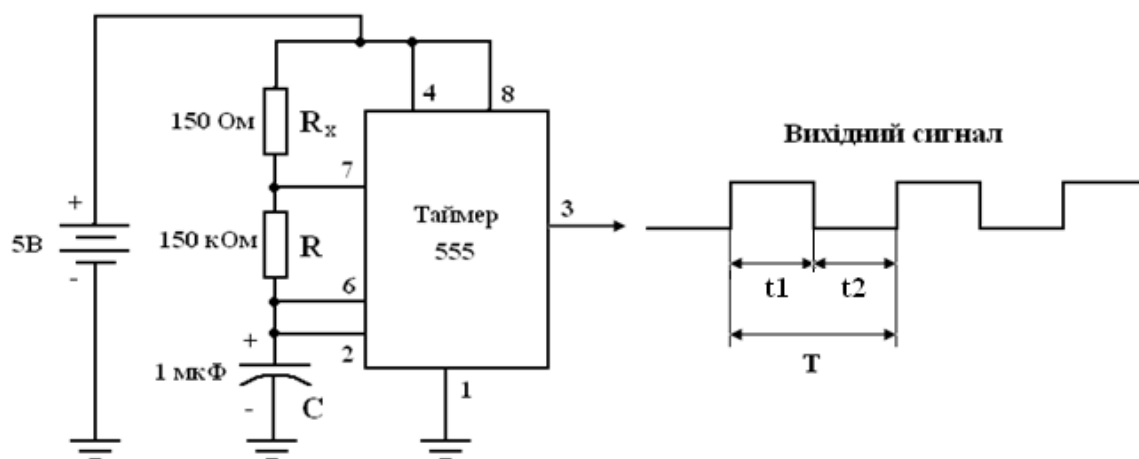


Рис. 11.1. Схема генератора тактових імпульсів

Період коливань залежить від номіналів резисторів R , R_x та конденсатора C :

$$t_1 = 0,693 \cdot (R_x + R) \cdot C;$$

$$t_2 = 0,693 \cdot R \cdot C.$$

Якщо $R_x \ll R$, то період коливань T пов'язаний з номіналами резистора R та конденсатора C співвідношенням $T \approx 1,4 \cdot R \cdot C$. В такому випадку генерується імпульс типу меандр, шпаруватість якого приблизно дорівнює 2.

2. Завдання на комп'ютерний практикум

Розробити принципову схему та друковану плату генератора тактових імпульсів на основі мікросхеми таймеру 555 (рис. 11.2), використовуючи електронні компоненти із власної бібліотеки.

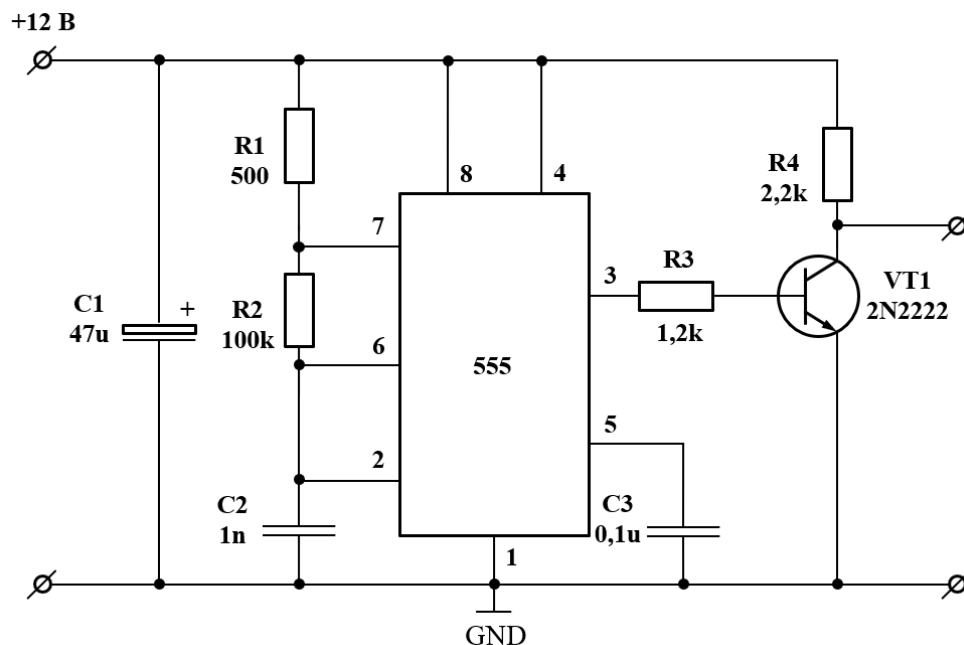


Рис. 11.2. Принципова схема генератора тактових імпульсів, яку необхідно розробити

Перелік елементів: мікросхема таймера NE555N у корпусі DIP, біполярний транзистор 2N2222, електролітичний конденсатор та керамічні конденсатори для монтажу в отвори, резистори у корпусі із аксіальними виводами для монтажу в отвори потужністю 0,25 Вт, роз'єм DS-210 та роз'єм BNC.

Зверніть увагу, що у вашій власній бібліотеці мають бути усі необхідні електронні компоненти, що перераховані у попередньому абзаці, окрім біполярного транзистора 2N2222. У цьому комп'ютерному практикумі компонентний модуль транзистора 2N2222 буде створено на основі розробленого раніше компонентного модуля транзистора BC547.

3. Порядок виконання комп'ютерного практикуму

3.1. Створення компонентного модуля транзистора 2N2222

Транзистор 2N2222 випускається в корпусі SOT-54 (інша назва TO-92) та має конфігурацію виводів, що представлена на рис. 11.3.

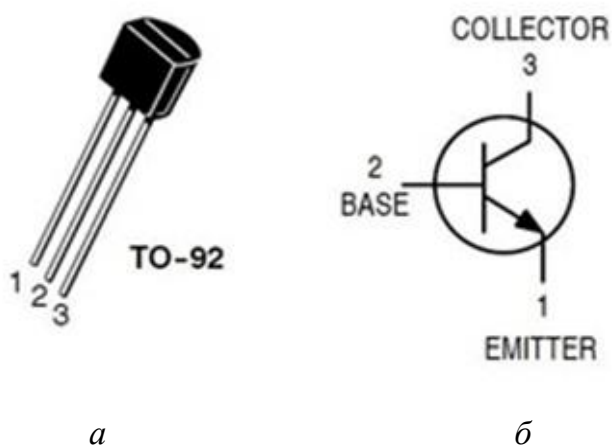


Рис. 11.3. Транзистор 2N2222: *а* – корпус TO-92, *б* – конфігурація та призначення виводів

Оскільки наразі у вашій бібліотеці немає компонентного модуля транзистора 2N2222, то його необхідно створити. Проте ми не будемо створювати його з нуля, а врахуємо той факт, що у бібліотеці вже є компонентний модуль біполярного транзистора BC547, який також має структуру NPN. Більше того, якщо порівняти транзистор 2N2222 та BC547, то можна помітити, що вони випускаються в однакових корпусах та мають однакову конфігурацію виводів, а відрізняються лише електричними характеристиками. Це означає, що ми можемо просто створити копію компонентного модуля BC547 у власній бібліотеці, після чого перейменувати її на 2N2222.

Тому, перш ніж переходити до виконання основної частини комп'ютерного практикуму, спочатку у власній бібліотеці створіть компонентний модуль транзистора 2N2222.

3.2. Створення нового проєкту

Створіть новий проєкт ДП із назвою **PulseGenerator** та збережіть його в окремій папці на диску.

3.3. Додавання до проєкту принципової схеми

Додайте у проєкт новий аркуш принципової схеми, який збережіть у папці проєкту під іменем **PulseGenerator.SchDoc**.



Відкрийте вікно властивостей документу принципової схеми за допомогою команди **Design** → **Document Options** та переконайтеся, що на вкладці **Sheet Options** в полі **Standard Styles** встановлено формат аркуша A4.

3.4. Розміщення компонентів на принциповій схемі

Відкрийте панель **Libraries**, клацнувши на її вкладці в правій частині робочого простору. З випадного списку у верхній частині панелі **Libraries** виберіть власну бібліотеку, щоб зробити її активною.

Оберіть у списку компонент **NE555N** та перетягніть його у робочу область принципової схеми. Для розміщення компонента на схемі клацніть ЛКМ або натисніть клавішу **<Enter>**.

Аналогічно описаному вище розмістіть усі необхідні компоненти на схемі (рис. 11.4), використовуючи власну бібліотеку. Причому ви можете або під час розміщення таких компонентів, як резистори та конденсатори, змінити їх номінали, або вже після розміщення. Окрім номіналів ніяких інших властивостей компонентів на схемі не змінюйте.

У цьому проєкті, на відміну від попереднього, ми не будемо використовувати мітки ланцюгів для позначення ланцюгів живлення та землі. Замість міток ми використаємо готові компоненти **GND Power Port** та **VCC Power Port**, які позначають відповідно ланцюги живлення та землі і є глобальними для усього проєкту. Для додавання цих компонентів на аркуш принципової схеми використовуються піктограми  та  із панелі інструментів.

Зауваження!

Фактично, компоненти **GND Power Port** та **VCC Power Port** замінюють мітки, які вказуються на ланцюгах. Використовувати ці компоненти або мітки – це вибір розробника схеми. Використання компонентів **GND Power Port** та **VCC Power Port** є зручним у тому сенсі, що візуально вказує на живлення та землю завдяки використанню відповідних УГП на схемі.

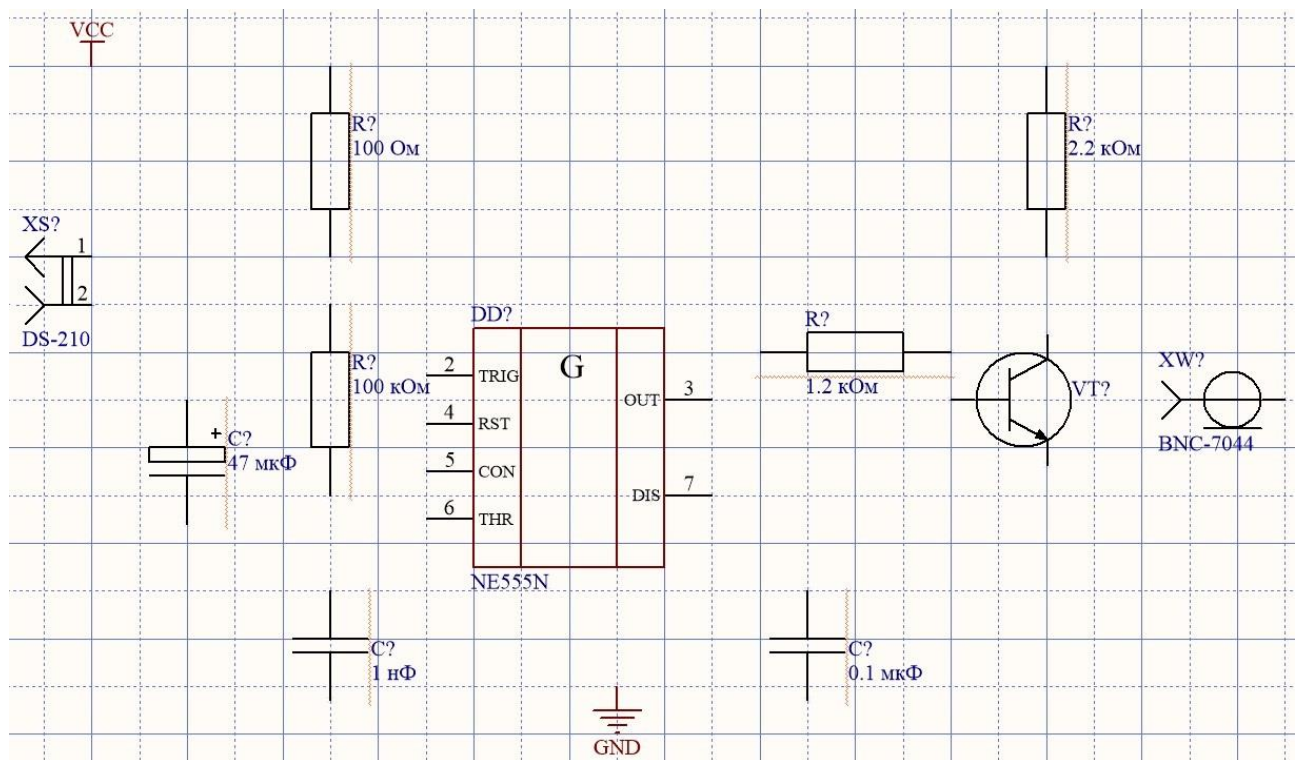



Рис. 11.4. Компоненти, що розміщені на схемі, але не з'єднані між собою

Зауваження!

На схемі вище можна помітити, що усі резистори та конденсатори підкреслені червоною хвилястою лінією. Якщо навести курсор, наприклад, на будь-який із резисторів, то з'явиться спливаюче повідомлення, у якому міститься опис помилки «Error: Duplicate Part Designators». Ця помилка виникає тоді, коли два або більше електронних компонентів на схемі мають однакове позиційне позначення (designator). Поки що це так і є, оскільки позиційне позначення усіх резисторів наразі «R?», а конденсаторів – «C?». Наразі це не заважає побудові принципової схеми, а згодом ця помилка буде виправлена із використанням механізму анотації принципової схеми (*Tools* → *Annotation* → *Annotate Schematic*).

3.5. З'єднання компонентів принципової схеми

Для з'єднання виводів компонентів використовуйте команду *Place* → *Wire* (або піктограму  на панелі інструментів), щоб увійти в режим розміщення з'єднань.

Встановіть електричні з'єднання між компонентами схеми відповідно до ескізу схеми, що представлена на рис. 11.5.

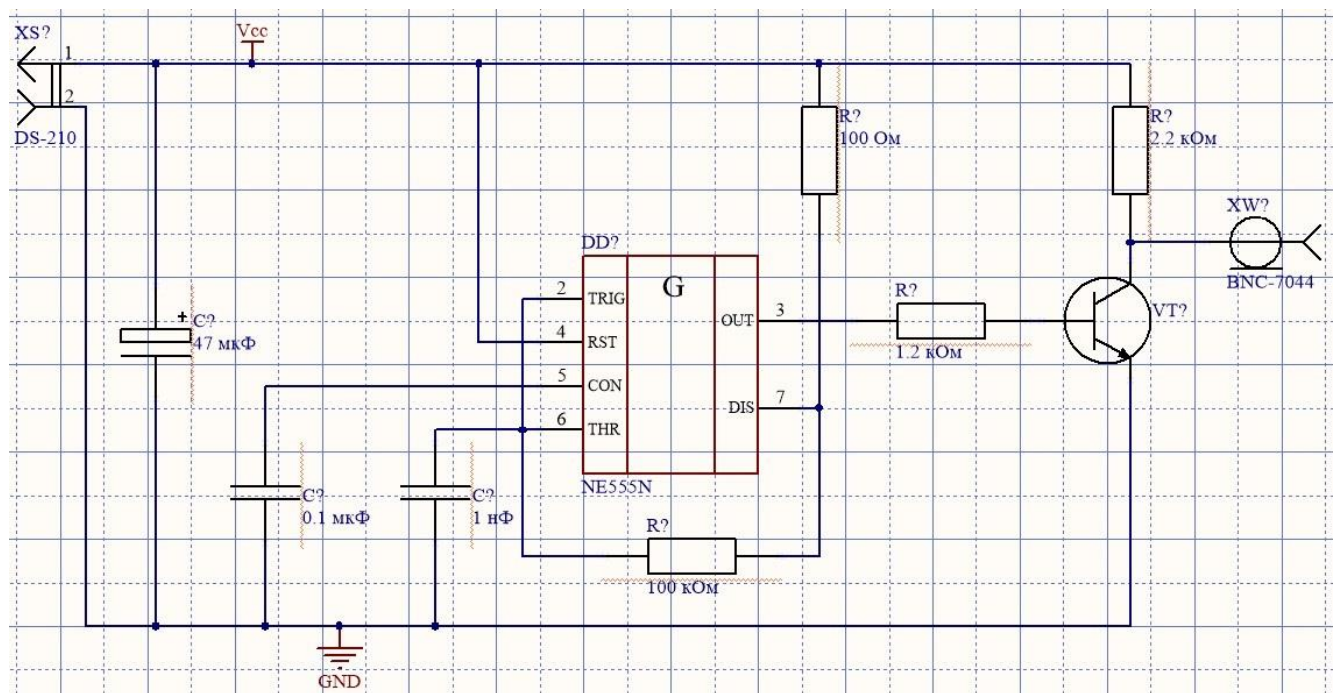


Рис. 11.5. Принципова схема із непронумерованими компонентами

Зверніть увагу, що усі видимі виводи мікросхеми таймера NE555N підключені до певних електричних ланцюгів. Однак на принциповій схемі не показані приховані виводи мікросхеми таймера, які підключаються до землі (вивід 1) та живлення (вивід 8). Варто пам'ятати, що під час створення бібліотечного компонентного модуля мікросхеми таймера NE555N для виводу, який має підключатись до живлення, у його властивостях у полі **Connect To** було написано текст **Vcc**, а для виводу землі – **GND**. Це було зроблено не просто так, а для того, щоб, якщо на принциповій схемі будуть використовуватись або компоненти **GND Power Port** та **VCC Power Port**, або мітки ланцюгів із іменами **Vcc** та **GND**, то приховані виводи мікросхеми, у властивостях яких у полі **Connect To** прописані такі самі назви, під час трасування автоматично з'єднувались із ланцюгами живлення та землі.

Якщо відкрити властивості компонента **VCC Power Port**, то можна помітити, що за замовчуванням у полі **Net** написано текст **VCC** (усі великі літери). Наразі нас це не влаштовує, оскільки, як ми зауважили у попередньому абзаці, у властивостях прихованого виводу живлення мікросхеми таймеру прописано підключення до ланцюга **Vcc** (перша літера велика, інші – малі). Щоби узгодити написи, можна вчинити двома способами: 1) відредагувати текст у вікні властивостей компонента **VCC Power Port** у полі **Net**; 2) відредагувати текст у властивостях виводу живлення мікросхеми у полі **Connect To**. Ми використаємо

перший спосіб. Тому у вікні властивостей компонента **VCC Power Port** у полі **Net** замість тексту **VCC** напишіть **Vcc** (рис. 11.6).

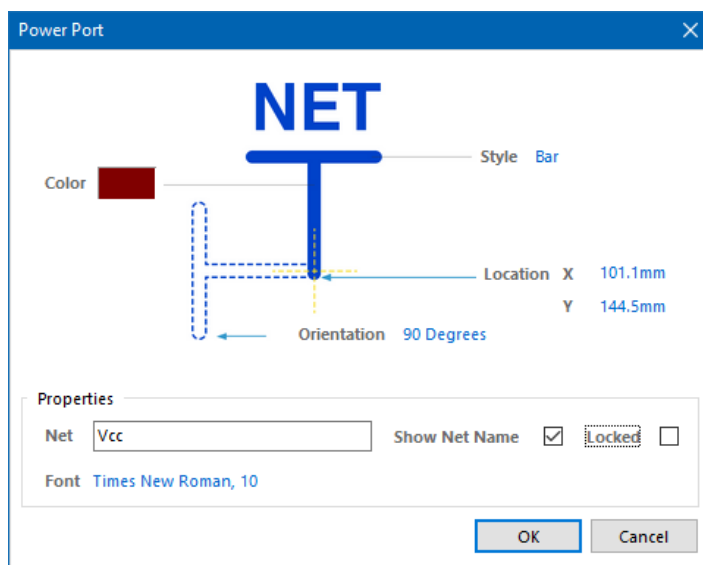


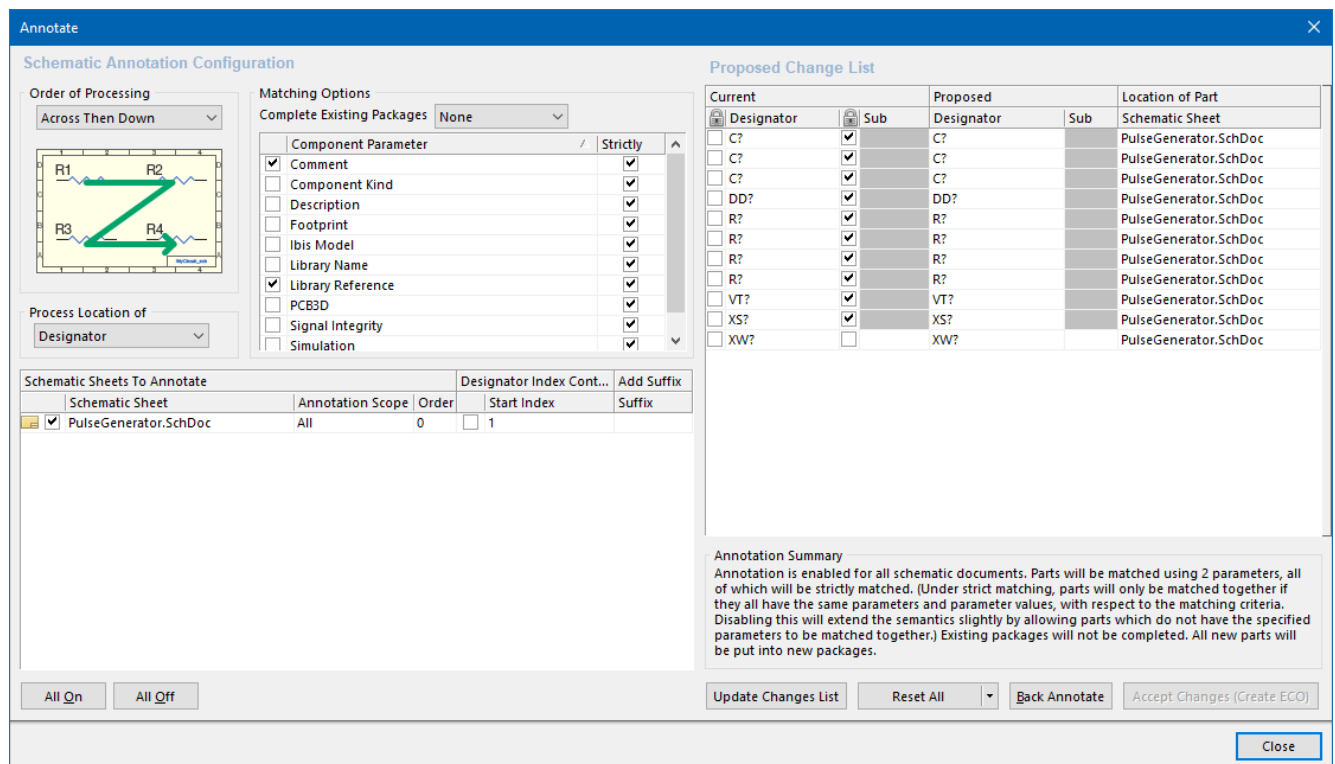
Рис. 11.6. Вікно властивостей компонента **VCC Power Port**

Властивості компонента **GND Power Port** редагувати не потрібно, оскільки напис **GND** співпадає із відповідним текстом у полі **Connect To** у властивостях відповідного виводу мікросхеми таймера.

3.6. Автоматична нумерація електронних компонентів на принциповій схемі

Поки що на схемі елементи не пронумеровані і містять у складі свого позиційного позначення знак запитання. Для того, щоб автоматично пронумерувати елементи схеми за заданими правилами, необхідно виконати команду **Tools** → **Annotation** → **Annotate Schematic**. Відкриється вікно налаштувань автоматичної нумерації елементів **Annotate** (рис. 11.7).

За допомогою цього вікна можна автоматично нумерувати позиційні позначення усіх або окремих УГП на принциповій схемі (причому, якщо проєкт містить кілька пов'язаних схем, то можна обрати й окремо схему, для якої виконуватимуться необхідні дії). Процес автонумерації гарантує, що позиційні позначення будуть унікальними та впорядкованими. Також у вікні **Annotate** можна налаштувати параметри для компонентів, що складаються з кількох частин, встановити параметри індексу та суфіксу тощо. А за потреби можна скинути нумерацію позиційних позначень.

Рис. 11.7. Вікно налаштувань *Annotate*

У випадковому списку *Order of Processing* обирають стратегію нумерування (наприклад, спочатку зверху вниз, а потім зліва направо, або спочатку зліва направо, а потім зверху вниз). У нашому випадку обираємо пункт *Down Then Across*. Перевірте, щоб в області *Proposed Change List* у першому стовпчику *Sub* (що визначає поточний стан нумерації *Current*) були встановлені галочки навпроти усіх елементів схеми. Далі натисніть кнопку **Update Changes List**, після чого з'явиться інформаційне вікно із повідомленням про те, для скількох елементів відбудуться зміни у їх позиційних позначеннях. Натисніть кнопку **OK**, щоб закрити інформаційне вікно, та подивіться, яким чином змінилися значення у другому стовпчику *Designator* (що належить до параметру *Proposed*). Якщо вас влаштовує, яким чином **Altium Designer** автоматично пронумерував усі елементи, то необхідно натиснути кнопку **Accept Change (Create ECO)**. У новому вікні *Engineering Change Order*, що відкриється, можна ще раз переглянути запропоновані зміни та у разі згоди із ними натиснути кнопку **Execute Changes**. Якщо все виконано правильно, то навпроти усіх елементів стоятимуть зелені прапорці. Для завершення натисніть кнопку **Close**.

Після виконання цих дій схема виглядатиме так, як це представлено на рис. 11.8. Можна помітити, що тепер резистори не підкреслюються червоною хвилястою лінією, що означає відсутність помилки «**Error: Duplicate Part Designators**».

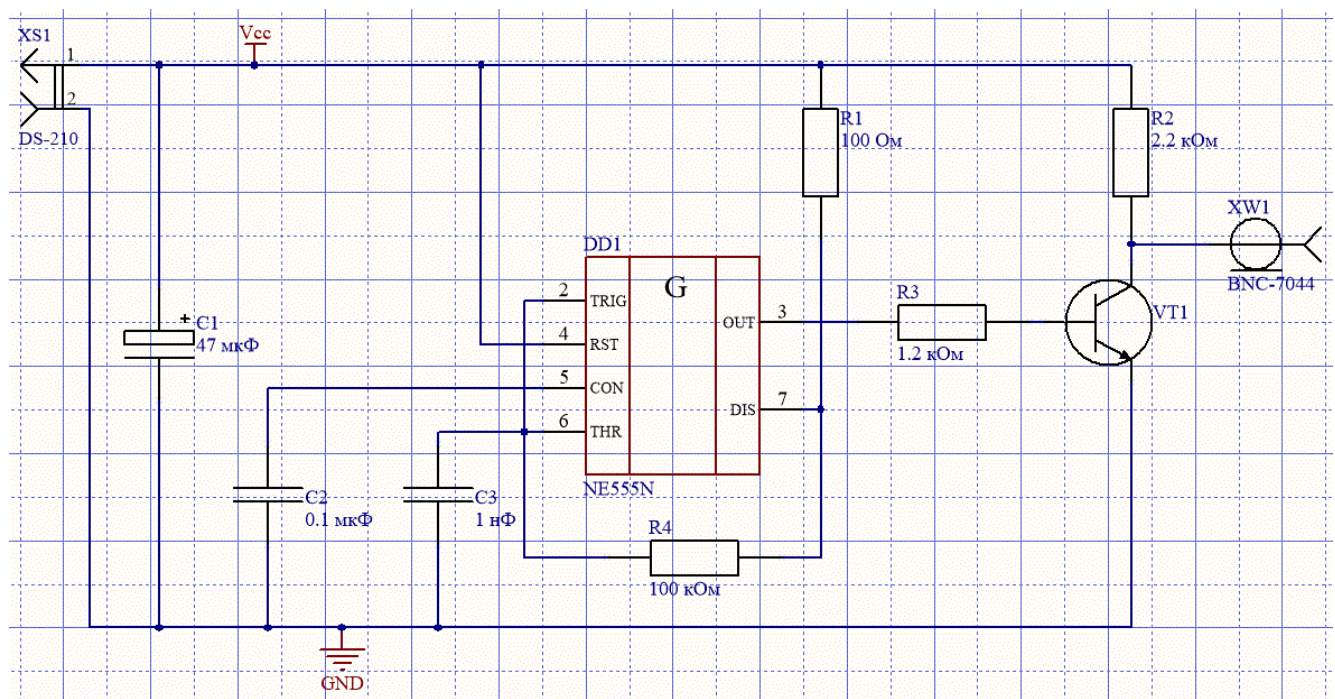


Рис. 11.8. Завершена принципова схема мультивібратора

Збережіть поточну версію схеми та сам проєкт.

Зауваження!

Зверніть увагу, що нумерація деяких елементів на принциповій схемі у Altium Designer відрізняється від тієї, що представлена на вихідній схемі (рис. 11.2). Це відбулось із-за того, що ми інакше розмістили в області аркуша принципової схеми компоненти, тому що так було зручніше. А внаслідок використання автоматичної нумерації ми отримали саме такий результат. У цьому немає проблеми – головне, щоб елементи по суті були правильно підключені. Якщо ж потрібно мати саме таку нумерацію, як на вихідній схемі, то необхідно або так само у просторі аркуша принципової схеми розміщувати компоненти, або використовувати нумерацію вручну.

Наприкінці цього розділу буде доцільним пояснити ще один момент, який пов'язаний із мітками на ланцюгах та підключенням прихованих виводів мікросхем до заданих ланцюгів. Зверніть увагу на те, яким чином на схемі (див. рис. 11.8) з'єднано виводи роз'єму живлення DS-210 із іншими компонентами. Можна помітити, що це безпосереднє з'єднання із ланцюгами на схемі. У цьому випадку так відбулось тому, що згідно рис. 11.2 на схемі присутні такі виводи компонентів, які не є прихованими виводами живлення, але на які безпосередньо необхідно

подати живлення (зокрема, це вивід згладжувального конденсатора на вході, вивід резистора, який знаходиться в ланцюзі колектора транзистора, вивід RST мікросхеми таймера тощо). А тепер уявіть, якщо б у схемі не були присутні усі ці підключення до ланцюгів живлення, а була б тільки мікросхема таймера із прихованими виводами живлення. Як у такому випадку з'єднати виводи роз'єму живлення DS-210 із прихованими виводами мікросхеми таймера? Напрямую – ніяк. Однак опосередковано через мітки ланцюгів це робиться елементарно.

Подивіться зовнішній вигляд схеми на рис. 11.9 та порівняйте її зі схемою на рис.11.8. Ці схеми в сенсі електричних з'єднань є абсолютно однаковими. Завдяки портам живлення **VCC Power Port** та **GND Power Port** на другій схемі роз'єм DS-210 є візуально відокремленим¹²⁰ від іншої частини схеми, хоча насправді через порти електрично з'єднаний із нею (а також з'єднаний із прихованими виводами живлення таймера).

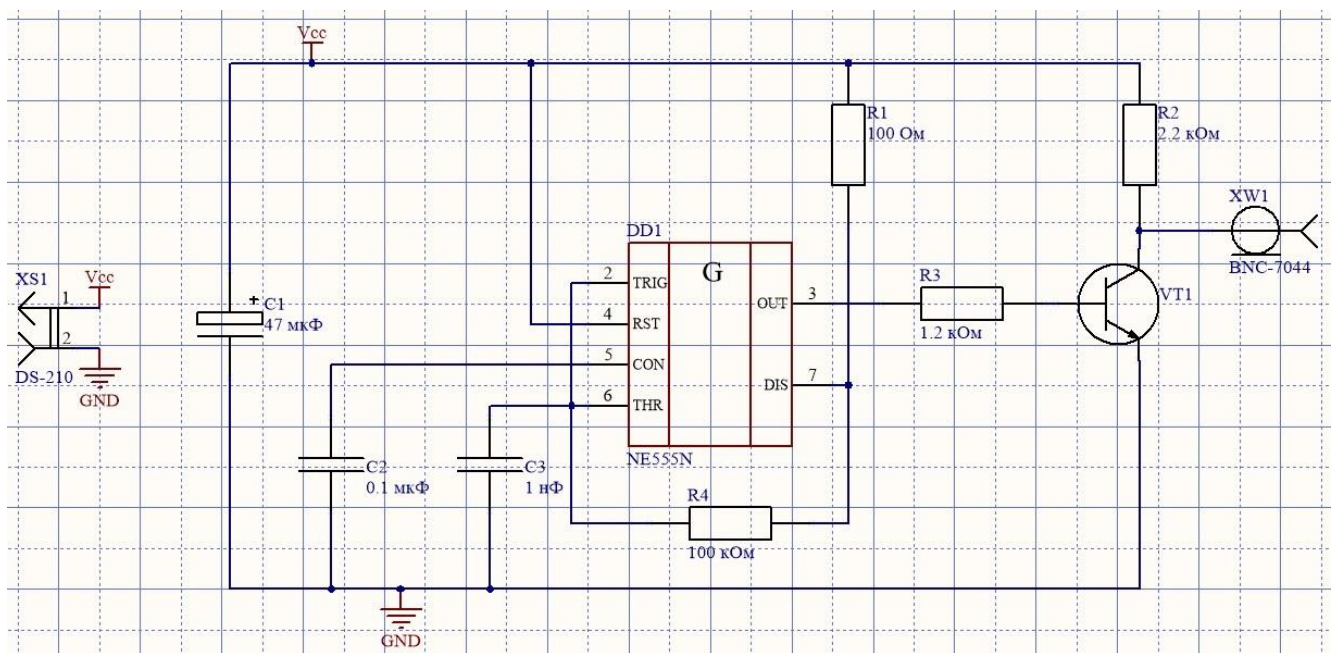


Рис. 11.9. Інший варіант візуального оформлення принципової схеми мультивібратора

3.7. Компіляція проєкту та перевірка на наявність помилок

Для того, щоб виконати компіляцію проєкту **PulseGenerator**, з меню **Project** виберіть пункт **Compile PCB Project PulseGenerator.PrjPcb**. Якщо принципова схема побудована правильно, то на панелі **Messages**, яка з'явиться після компіляції,

¹²⁰ І може бути розміщеним будь-де у вільній області схеми.

не повинно відображатися ніяких повідомлень про помилки. За наявності помилок вивчіть кожну з них і перевірте правильність з'єднань на схемі.

3.8. Створення друкованої плати

Перед тим як передати дані проєкту із Редактора принципів схем у Редактор друкованих плат, необхідно створити, як мінімум, порожню ДП. Під час виконання попереднього комп'ютерного практикуму нова заготовка ДП була створена методом натискання ПКМ на імені проєкту у панелі **Projects** і вибору пункту **Add New to Project** → **PCB**. У цьому комп'ютерному практикумі для створення друкованої плати буде використано спеціальний майстер **PCB Board Wizard**, який дозволяє вибрати один із стандартних контурів, а також задавати власні габарити плати. На будь-якому етапі роботи майстра, використовуючи кнопку **Back**, завжди можна повернутися до попередніх дій, щоб перевірити або змінити введені дані.

Спочатку відкрийте панель **Files**¹²¹. У розділі **New from Template** у нижній частині панелі **Files** клацніть на рядку **PCB Board Wizard** (якщо цей рядок не відображається, то згорніть деякі розділи панелі **Files**, клацнувши на значку зі стрілкою вгору в заголовку розділу). У результаті цих дій відкриється головна сторінка майстра. Натисніть **Next** для продовження. Відкриється нова сторінка, на якій оберіть метричну систему одиниць вимірювань (**Metric**) та натисніть **Next**, щоб відкрилась нова сторінка. На поточній сторінці майстра можна обрати шаблон плати із наперед заданими розмірами. Для того, щоб задати власні розміри плати, необхідно вибрати **Custom** зі списку конфігурацій плат і натиснути **Next**, щоб відкрилась наступна сторінка (рис. 11.10).

У лівій половині вікна можна вибрати форму плати: прямокутна (**Rectangular**), кругла (**Circular**) або довільна (**Custom**). Нижче в області **Board Size** встановлюються розміри. Для розроблюваної схеми генератора тактових імпульсів буде достатньо прямокутної плати розміром 60x30 мм¹²², тому для параметра

¹²¹ За замовчуванням ця панель прикріплена до лівої сторони вікна **Altium Designer**. Якщо панель не відображається, натисніть кнопку **System** у правому нижньому кутку робочого простору і виберіть пункт **Files** зі спливаючого меню.

¹²² Інколи встановлені за допомогою **PCB Board Wizard** розміри плати потребують змін, щоб оптимально розташувати компоненти. Є кілька способів змінити поточні розміри ДП. Один із них полягає у використанні режиму **Board Planning Mode** (гаряча клавіша <1>) із меню **View**.

Outline Shape виберіть значення **Rectangular**, а в області **Board Size** у полі **Width** введіть число 60, у полі **Height** – 30.

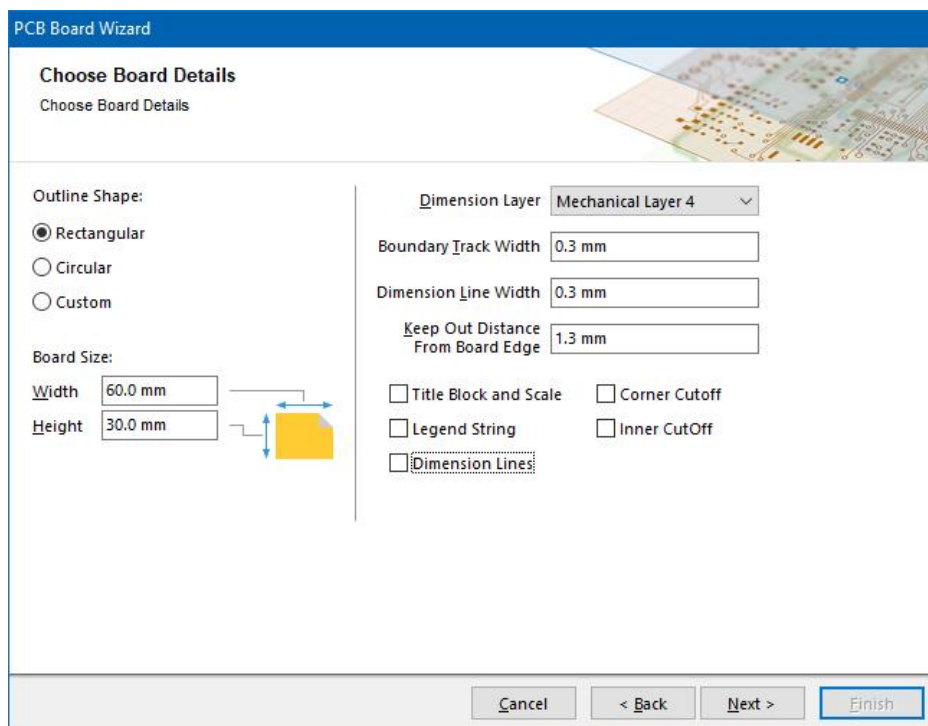


Рис. 11.10. Налаштування розмірів плати у майстрі **PCB Board Wizard**

У правій половині вікна налаштовуються такі параметри:

- У списку **Dimension Layer** вибирається шар для розміщення на ньому різних розмірних ліній (за замовчуванням – це **Mechanical Layer 1**);
- **Boundary Track Width** – товщина ліній, які окреслюють контур майбутньої плати;
- **Dimension Line Width** – товщина розмірних ліній;
- **Keep Out Distance From Board Edge** – задає відстань від країв плати до контуру, який не можуть перетинати компоненти та траси;
- **Title Block and Scale** – наявність на аркуші штампа з назвою та масштабом;
- **Legend String** – наявність опису плати;
- **Dimension Lines** – відображення розмірних ліній;
- **Corner Cutoff** – обрізка кутів плати;
- **Inner Cutoff** – створення прямокутних отворів (вирізів) у платі.

Після того, як обрано режим **Board Planning Mode**, необхідно перейти у меню **Design** і обрати пункт **Edit Board Shape**.

Важливо!

У випадковому списку *Dimension Layer* вибирається шар, на якому будуть розміщені графічні примітиви (стрілки, лінії, цифри), що лише вказують на розміри плати, а не шар, на якому розміщуватимуться лінії контуру плати. На відміну від способу створення нового контуру плати, який ми використовували під час виконання попереднього комп'ютерного практикуму, майстер *PCB Board Wizard* автоматично створить контур і перевизначить об'єкт *Board Shape* за тими розмірами, що були вказані в області *Board Size*.

Отже, якщо вам потрібні розмірні лінії контуру плати, то розміщуйте їх на будь-якому вільному механічному шарі¹²³. Наприклад, якщо ви виберете із випадного списку параметру *Dimension Layer* 4-й механічний шар, то розмірні лінії будуть розміщені на ньому. З іншого боку, оскільки у цьому комп'ютерному практикумі ми вимкнемо опції *Title Block and Scale*, *Legend String* та *Dimension Lines*, то це означатиме, що ніякі розмірні лінії не створюватимуться, а, отже, ніякий шар для них виділений не буде, щоб там не було вибрано у списку *Dimension Layer*.

Зауваження!

Параметр *Keep Out Distance From Board Edge* визначає відстань від країв плати до контуру, в межах якого можливе розміщення компонентів та прокладання трас. Якщо він не дорівнює нулю, то це означає, що автоматично на шарі *Keep Out Layer* будуть сформовані лінії, які визначатимуть своєрідний внутрішній контур, розміри якого будуть меншими за розміри плати. Наприклад, якщо параметр *Width* = 60 мм, *Height* = 30 мм, а *Keep Out Distance From Board Edge* = 1,3 мм, то значення ширини обмежувального контуру буде становити 57,4 мм (60-1,3-1,3), а висоти – 27,4 мм (30-1,3-1,3).

Перевірте, щоб налаштування усіх параметрів у вікні *Choose Board Details* відповідали тим, що представлені на рис. 11.10. Натисніть **Next** для продовження.

У наступному вікні можна вказати кількість шарів плати. Нам потрібно 2 сигнальних шари, внутрішні шари живлення та землі не знадобляться. Тому для

¹²³ Оскільки у попередніх комп'ютерних практикумах ми використовували 13-й механічний шар для розміщення на ньому тривимірних зображень корпусів, тому цей шар в нашому випадку не повинен використовуватись для розміщення розмірних ліній.

параметра **Signal Layers** встановить значення 2, а для **Power Planes** – 0. Натисніть **Next** для продовження.

З'явиться наступне вікно, у якому встановлюється стиль перехідних отворів (рис. 11.11): наскрізні (**Thruhole Vias only**) або глухі (**Blind and Buried Vias only**). Оскільки розроблювана нами плата не міститиме внутрішніх шарів, то виберіть **Thruhole Vias only** і натисніть **Next**.



Рис. 11.11. Стиль перехідних отворів: *а* – наскрізні; *б* – глухі

У новому вікні задається тип компонентів, що переважатимуть на платі: компоненти для поверхневого монтажу (**Surface-mount components**) або компоненти для монтажу в отвори (**Through-hole components**)¹²⁴. Виберіть тип компонентів **Through-hole components** та опцію **One Track**. Натисніть **Next**.

Відкриється передостання сторінка майстра, за допомогою якої можна задати деякі проєктні правила для ширини трас і розмірів перехідних отворів:

- **Minimum Track Size** – мінімальна ширина доріжки;
- **Minimum Via Width** і **Minimum Via HoleSize** – мінімальний зовнішній та внутрішній діаметр перехідного отвору;
- **Minimum Clearance** – мінімальна відстань між сусідніми провідниками.

У цьому вікні змініть лише один параметр **Minimum Track Size**, задавши нове значення 0,3 мм. Усі інші параметри залиште без змін. Натисніть **Next**.

Тепер майстер зібрав достатньо інформації для створення плати. Натисніть **Finish**. У Редакторі друкованих плат відкриється новий файл плати з іменем **PCB1.PcbDoc**.

Документ відображається у вигляді аркуша стандартного розміру, на якому розташована порожня плата (чорна область з сіткою). Щоб відключити відображення аркуша, у меню **Design** виберіть **Board Options** і зніміть прапорець **Display Sheet** в діалоговому вікні **Board Options**. За потреби ви можете вибрати інший формат аркуша, сітки і основного напису з набору шаблонів **Altium Designer**.

¹²⁴ Якщо буде обрано поверхневий монтаж, то необхідно додатково вказати, чи дозволене розміщення компонентів для поверхневого монтажу з двох сторін плати. Якщо ж буде обрано монтаж у отвори, то необхідно вказати, скільки доріжок може проходити між двома отворами – 1, 2 або 3.

Тепер, коли відображення креслярського аркуша відключене, виконайте у меню **View** команду **Fit Board**, щоб вписати плату в усю робочу область.

Зверніть увагу, що колір примітивів на шарі **Keep-Out Layer** та деяких інших механічних шарів (як от **Mechanical 1**, **Mechanical 5**, **Mechanical 9**, **Mechanical 13** тощо) є однаковим¹²⁵, що вводить в оману, оскільки може скластись враження, що контур, який обмежує область встановлення компонентів та трасування, побудований на одному із цих механічних шарів (рис. 11.12).

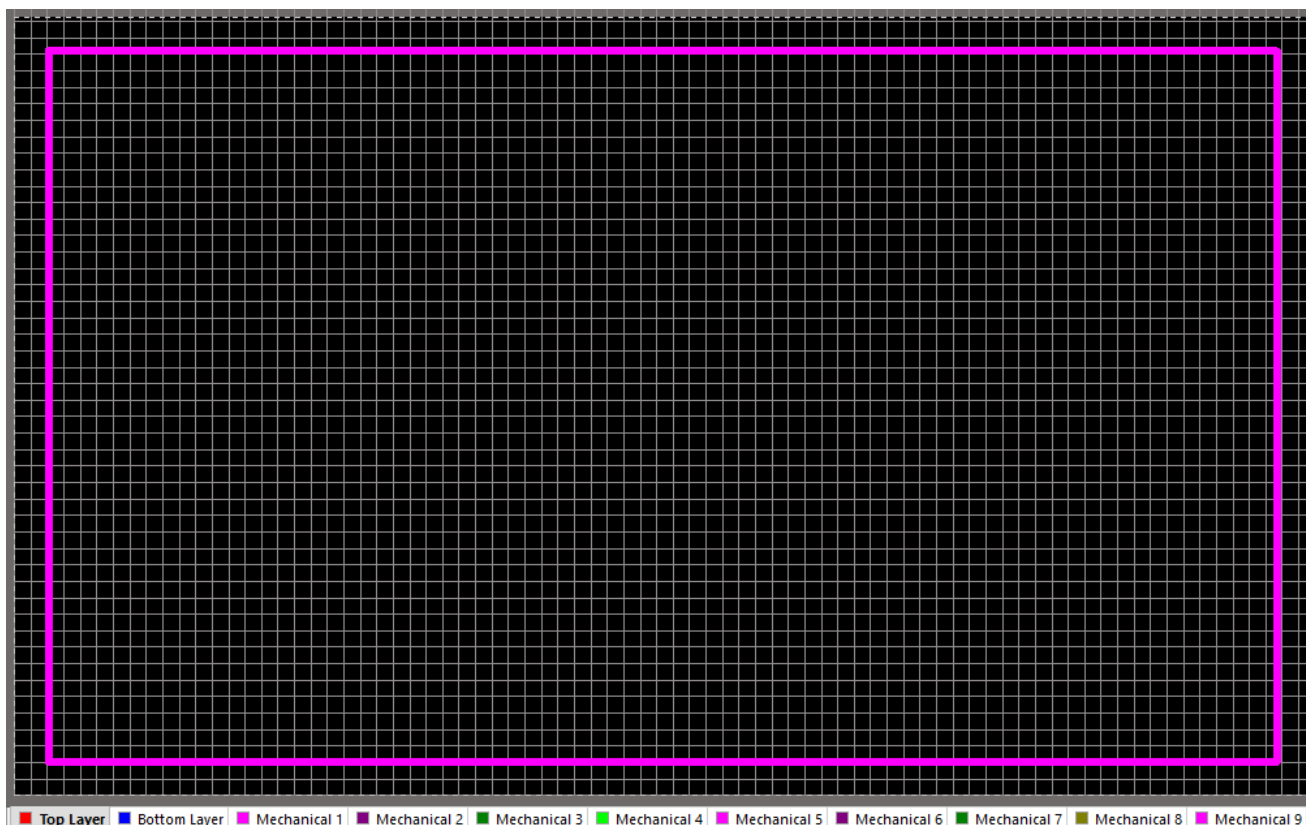


Рис. 11.12. Лінії обмежуючого контуру на шарі **Keep-Out Layer**, які відображаються таким самим кольором, як і деякі механічні шари

Щоб уникнути такої двозначності, змінимо колір відображення примітивів на шарі **Keep-Out Layer**. Для цього відкрийте вікно налаштувань параметрів шарів, виконавши команду **Design** → **Board Layers & Colors**. У вікні **View Configuration** знайдіть цей шар та встановіть такий колір, який би відрізнявся від усіх інших.

Ще одним нюансом використання майстра **PCB Board Wizard** є те, що він не створює ні на якому механічному шарі ліній контуру, які дублювали б контур

¹²⁵ Так буде лише у тому випадку, якщо ви використовуєте набір кольорів за замовчуванням. Якщо ж ви у попередніх власних проєктах налаштовували кольори шарів на власний розсуд і, наприклад, вже до виконання цього комп'ютерного практикуму змінили колір примітивів на шарі **Keep-Out Layer**, то у вас подібної ситуації не виникне.

плати, що зберігається в об'єкті **Board Shape**. У попередньому комп'ютерному практикумі зазначалось, що хоча дублюючі лінії контуру на одному із механічних шарів і не є обов'язковими, проте можуть існувати причини їх мати. Якщо вам потрібні лінії контуру на механічному шарі, то у цьому випадку можна скористатись командою **Board Shape** → **Create Primitives From Board Shape** із меню **Design**, яка дозволяє на основі форми об'єкту **Board Shape** створити лінії контуру на обраному механічному шарі. Після виконання цієї команди відкриється вікно **Line/Arc Primitives From Board Shape** (рис. 11.13). Хоча ми не будемо наразі створювати лінії контуру плати на окремому механічному шарі, проте ознайомтесь із рис. 11.13 та зверніть увагу на опції, які мали б бути активовані.

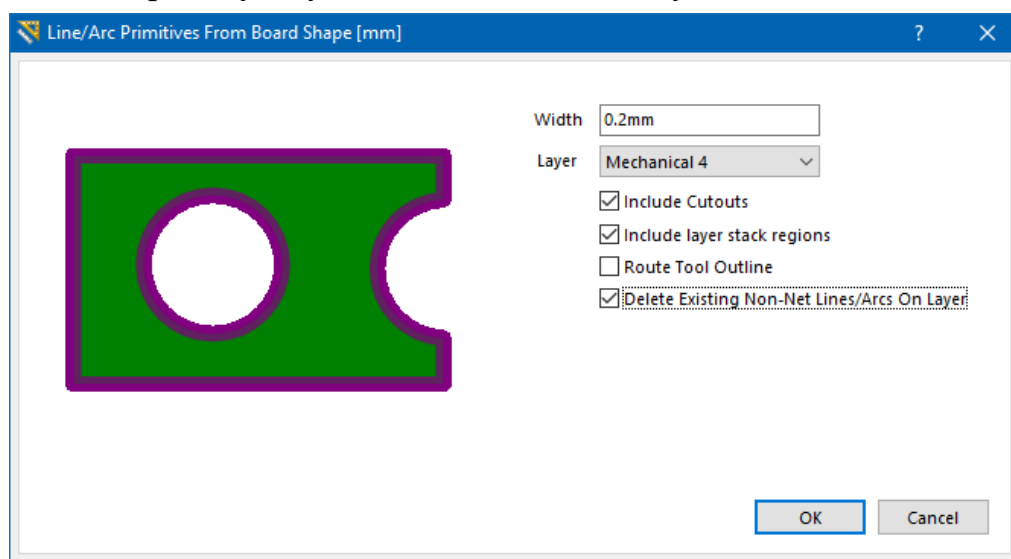
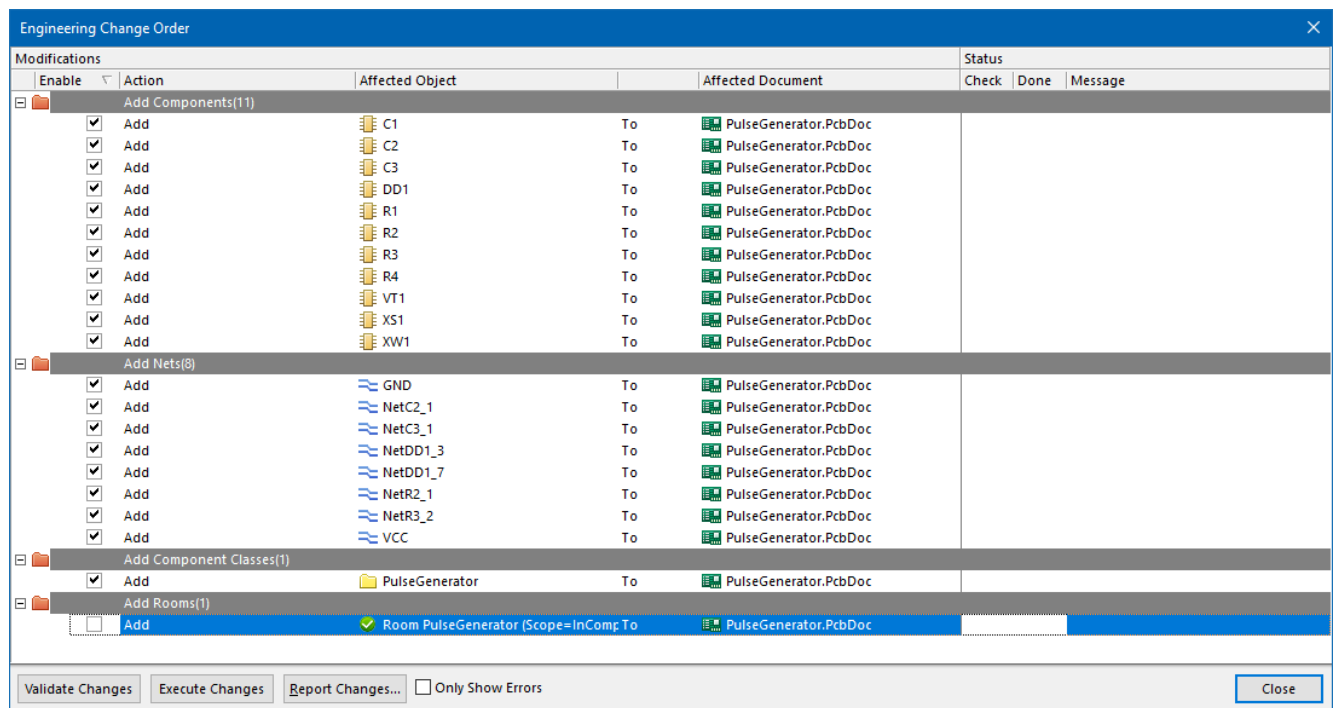


Рис. 11.13. Вікно **Line/Arc Primitives From Board Shape**

Після виконання усіх попередніх налаштувань для новоствореного документу ДП відкрийте панель **Projects**. Якщо файл плати не було автоматично додано до проєкту **PulseGenerator**, перетягніть його в деревоподібну структуру проєкту на панелі **Projects**. Клацніть ПКМ на новому файлі ДП у панелі **Projects** і виберіть **Save As** із контекстного меню. Переконайтеся, що ви зберігаєте файл у тій же папці, в якій знаходяться інші файли проєкту. Файл назвіть **PulseGenerator.PcbDoc**.

3.9. Перенесення проєктних даних

Відкрийте документ принципової схеми **PulseGenerator.SchDoc**. У меню **Design** виберіть **Update PCB Document PulseGenerator.PcbDoc**. Почнеться процес компіляції, після завершення якого відкриється діалогове вікно **Engineering Change Order** (рис. 11.14).

Рис. 11.14. Діалогове вікно *Engineering Change Order*

У цьому вікні приберіть галочку навпроти пункту *Add Room PulseGenerator*. Натисніть кнопку **Validate Changes**. Якщо всі зміни коректні, навпроти кожної зміни у графі *Status* має з'явитися зелений прапорець. Якщо ж зміни не пройшли перевірку, то закрийте діалогове вікно, перегляньте вміст панелі *Messages* і усуньте наявні помилки.

Натисніть *Execute Changes*, щоб внести зміни у ДП. Коли завершиться процес внесення змін, всі рядки будуть позначені зеленим прапорцем у стовпчику *Done*. Після натискання кнопки **Close** відкриється документ ДП з компонентами, що будуть розташовані праворуч за межами плати. Якщо відображається тільки плата, то просто віддаліть область відображення, щоб побачити усі компоненти (рис. 11.15).

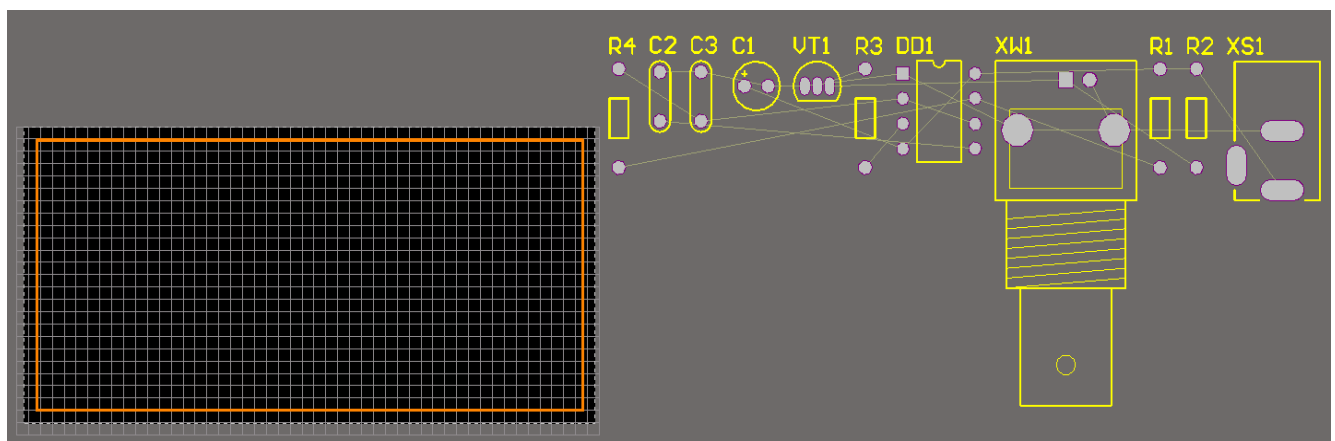


Рис. 11.15. Робоча область друкованої плати з нерозміщеними компонентами

3.10. Налаштування середовища проєктування друкованих плат

Перед тим, як почати розміщення компонентів на платі, необхідно налаштувати деякі параметри робочого середовища (наприклад, параметри сіток), а також задати параметри ДП, такі як шари і правила проєктування.

3.10.1. Сітки в середовищі проєктування друкованих плат

Всі об'єкти, що розміщені на ДП, прив'язуються до сітки. Щоб правильно налаштувати сітку, необхідно враховувати, яка технологія трасування буде використовуватись, які електронні компоненти будуть розміщуватись на платі, якою є ширина друкованих провідників та мінімальна відстань між ними тощо.

Для завдання кроку сітки натисніть ПКМ на вільному полі робочої області у Редакторі плат. У спливаючому меню наведіть курсор на пункт **Snap Grid**, після чого з'явиться додаткове меню із варіантами вибору кроку. Встановіть крок 0,1 мм. Якщо такого кроку немає у запропонованих варіантах за замовчуванням, то виберіть у додатковому меню пункт **Grid Properties**, за допомогою якого можна задати бажане значення (рис. 11.16).

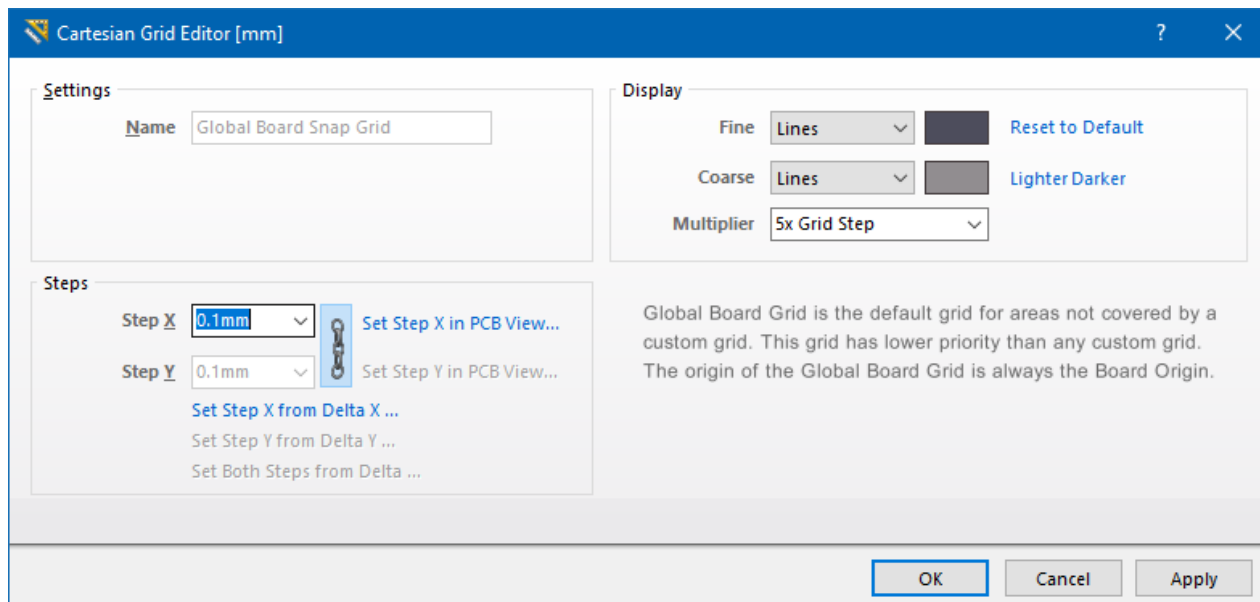


Рис. 11.16. Вікно налаштування кроку сітки

3.10.2. Завдання правил проєктування

Для поточного проєкту налаштуємо правила проєктування, які регламентують ширину ланцюгів живлення та землі. Переконайтеся, що ДП є активним документом, після чого у меню **Design** виберіть пункт **Rules**. У діалоговому вікні

PCB Rules and Constraints Editor у лівій його частині відображаються категорії правил. Відкрийте категорію **Routing**, що містить список правил трасування. Розгорніть підкатегорію **Width**, в якій відображаються наявні правила, що задають ширину провідників. Виберіть правило **Width**, щоб переглянути його налаштування (рис. 11.17). Це правило застосовується до всіх ланцюгів на ДП, оскільки область дії вказана як **All**.

Зверніть увагу, що за замовчуванням задано тільки одне правило ширини провідників, яке застосовується до всієї плати (ширина = 0,3 мм¹²⁶). Ми створимо ще одне правило, яке буде встановлювати окрему ширину трас для ланцюгів **Vcc** і **GND** (ширина = 0,6 мм).

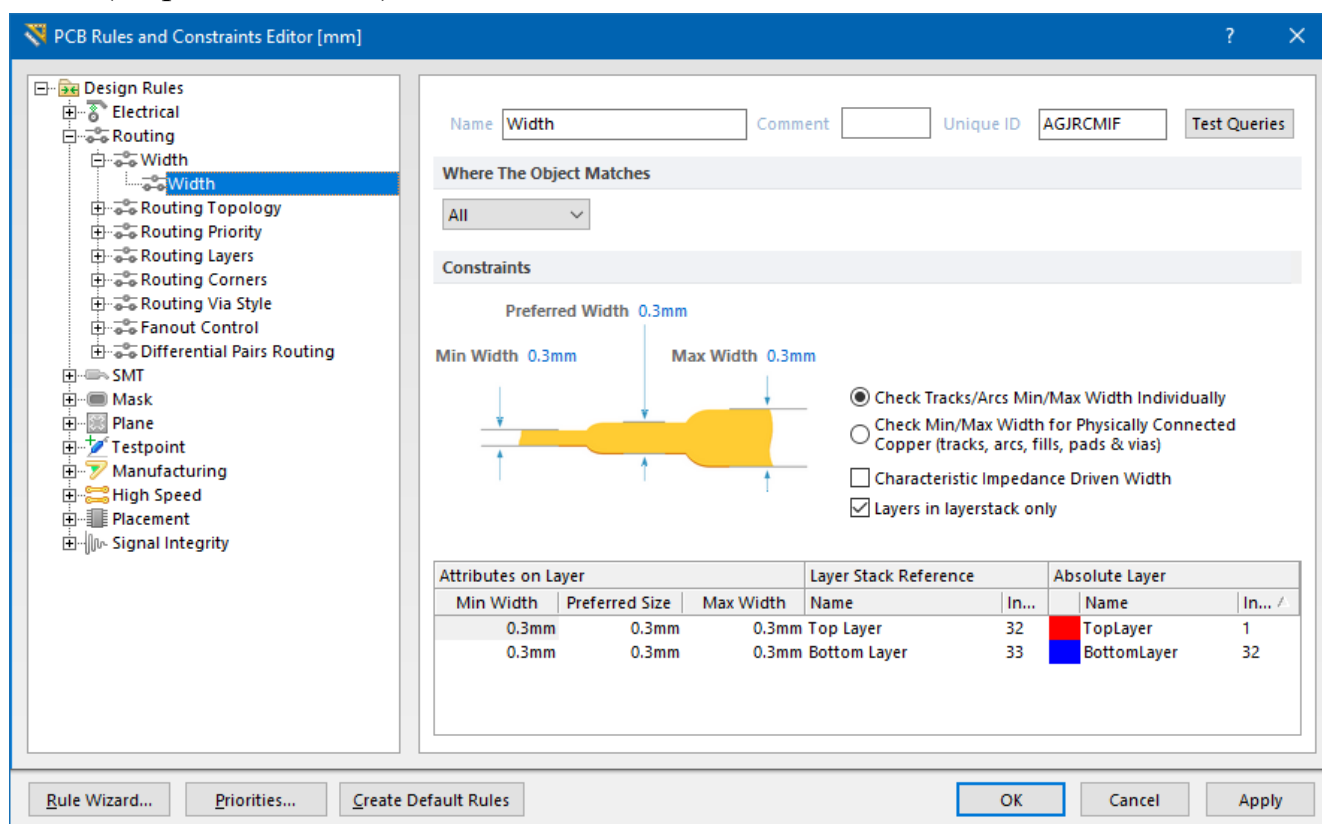


Рис. 11.17. Налаштування ширини провідників

Щоб додати нове правило, клацніть ПКМ на підкатегорії **Width** і виберіть пункт **New Rule** з контекстного меню. З'явиться нове правило з ім'ям **Width_1**. Виберіть його у списку правил, щоб показати його налаштування. У полі **Name** введіть **Width Power**. Для того, щоб задати область дії правила, необхідно із випадного списку в області **Where The Object Matches** вибрати опцію **Custom**

¹²⁶ Ширину провідника є саме такою тому, що під час використання майстра **PCB Board Wizard** ми встановили значення параметра **Minimum Track Size** рівним 0,3 мм.

Query. Далі потрібно натиснути кнопку **Query Builder**, щоб відкрити діалогове вікно **Building Query from Board** (рис. 11.18).

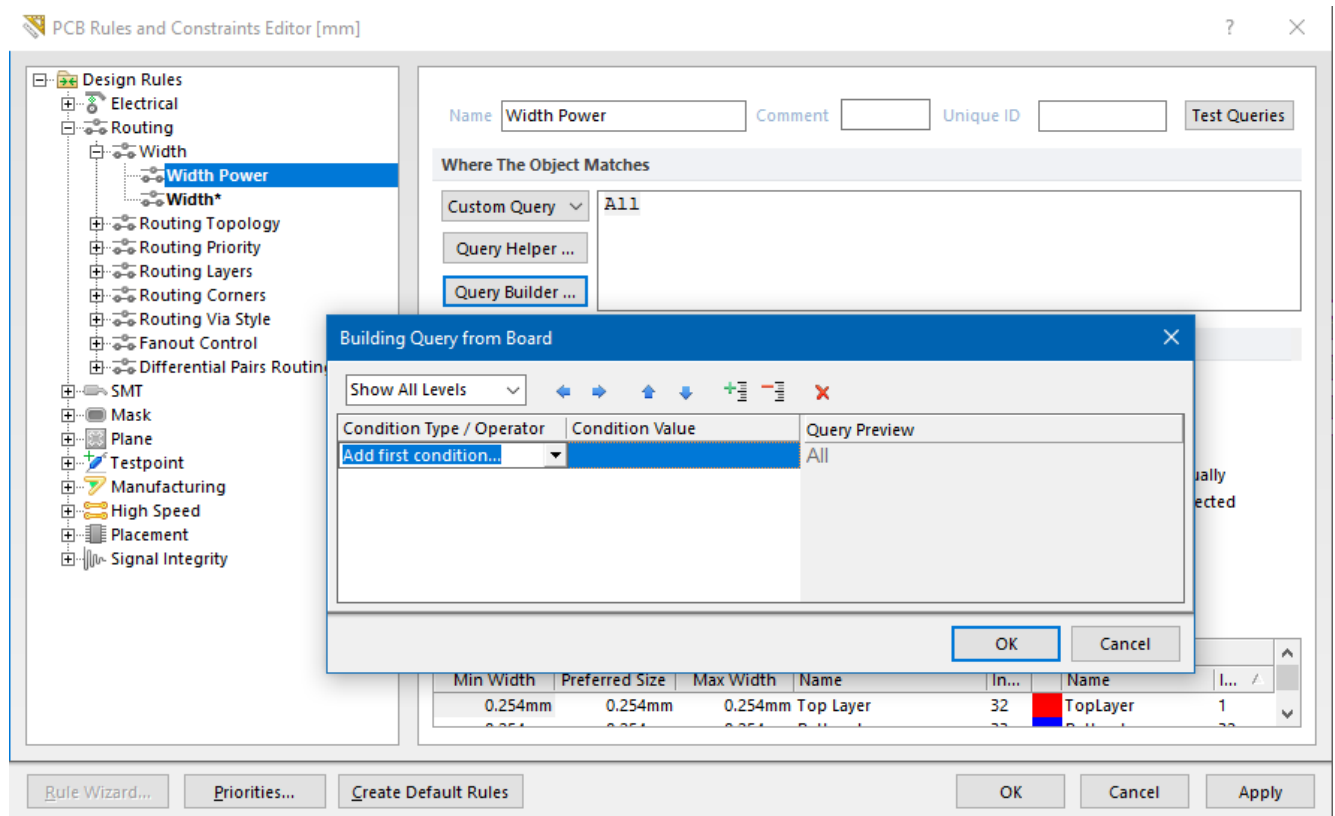


Рис. 11.18. Діалогове вікно **Building Query from Board**

Клацніть в полі **Add first condition** і оберіть **Belongs to Net** з випадного списку. Далі клацніть в полі **Condition Value** і оберіть зі списку **Vcc**. У правій частині вікна відображається попередній текст запиту **InNet ('Vcc')**. Клацніть на написі **Add another condition**, щоб додати в область дії нового правила також ланцюг **GND**. Як тип умови оберіть **Belongs to Net**, а як значення – **GND**. У лівій частині діалогового вікна клацніть на операторі **AND** (I) і виберіть **OR** (АБО) із випадного списку. Переконайтеся, що запит виглядає як **InNet ('Vcc') OR InNet ('GND')** (рис. 11.19).

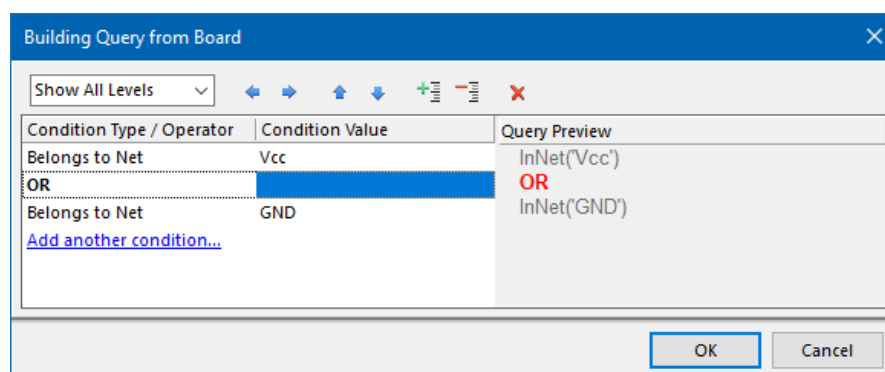


Рис. 11.19. Завдання області дії правила

Натисніть **OK**, щоб закрити діалогове вікно *Building Query from Board*. Тепер в області *Where The Object Matches* діалогового вікна *PCB Rules and Constraints Editor* відображається новий користувацький запис, що задає область дії правила. У нижній частині цього ж діалогового вікна встановіть нові значення для наступних параметрів (рис. 11.20):

- **Min Width** = 0,3 мм;
- **Preferred Width** = 0,6 мм;
- **Max Width** = 1,2 мм.

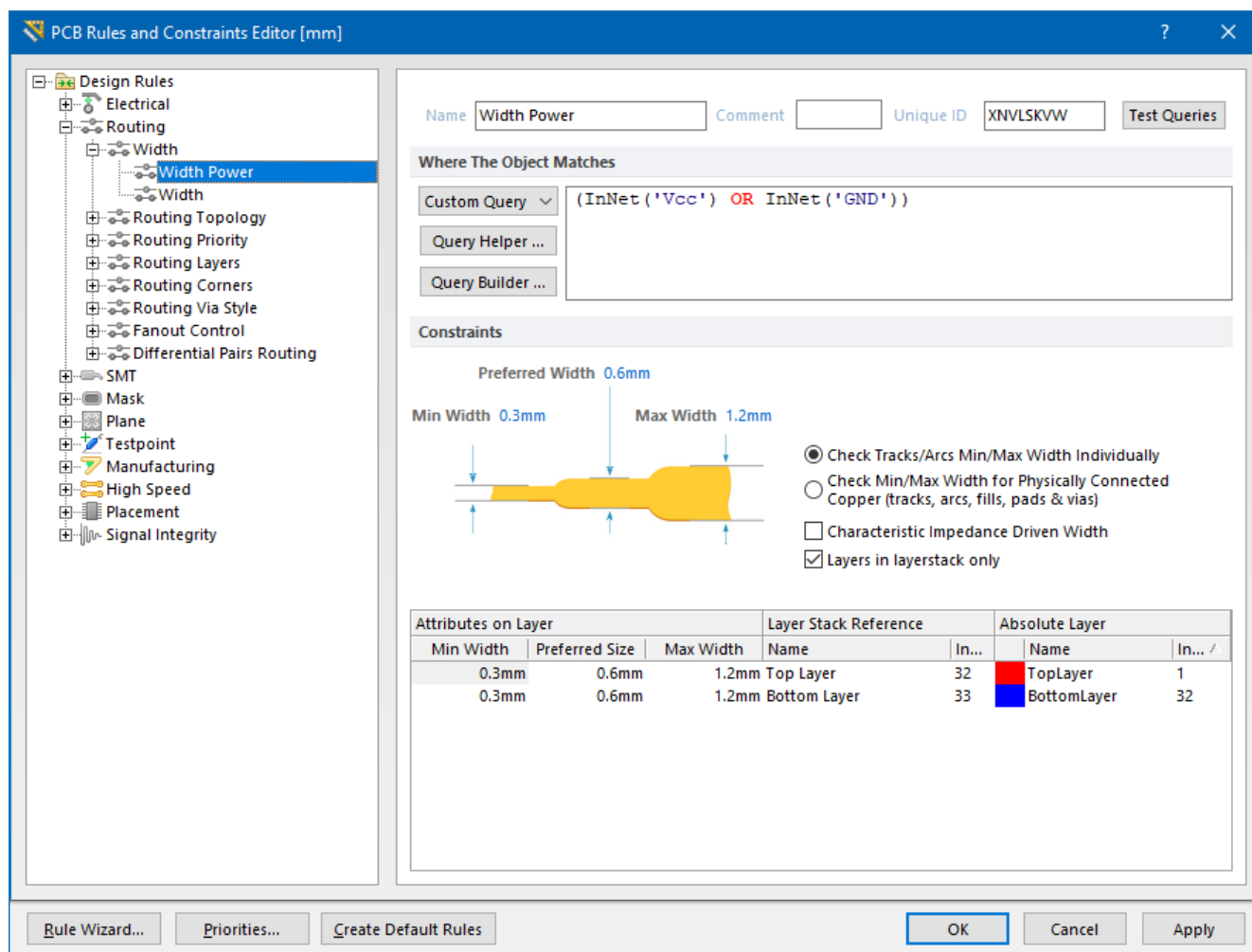


Рис. 11.20. Налаштування нового правила *Width Power*

Нове правило буде збережене під час закриття діалогового вікна або під час вибору іншого правила¹²⁷.

¹²⁷ Зверніть увагу, що в одному проєкті може бути задана будь-яка кількість правил, які визначають ширину траси. Головне, що потрібно розуміти: окремі правила застосовуються до ланцюгів із іменами. Коли виконується трасування, то система перевіряє, чи є для ланцюга із цим іменем окреме правило. Якщо є, то застосовується це окреме правило, а якщо немає – то загальне.

Тепер у нашому проєкті під час ручного або автоматичного трасування ДП усі доріжки будуть мати ширину 0,3 мм, за винятком ланцюгів землі (що мають мітку **GND**) і живлення (що мають мітку **Vcc**), ширина яких за новим правилом буде дорівнювати 0,6 мм.

3.11. Розміщення компонентів на друкованій платі

Для того, щоб перемістити компонент на плату, наведіть курсор на контур його ПМ, натисніть і утримуйте ЛКМ. Курсор прийме форму перехрестя і переміститься у точку прив'язки компонента. Продовжуючи утримувати ЛКМ, перетягніть компонент у задану позицію на платі. Щоб розмістити компонент в поточній позиції, відпустіть ЛКМ. Зверніть увагу, що лінії з'єднань переміщуються разом з компонентом.

Розмістіть компоненти на платі так, як показано на рис. 11.21.

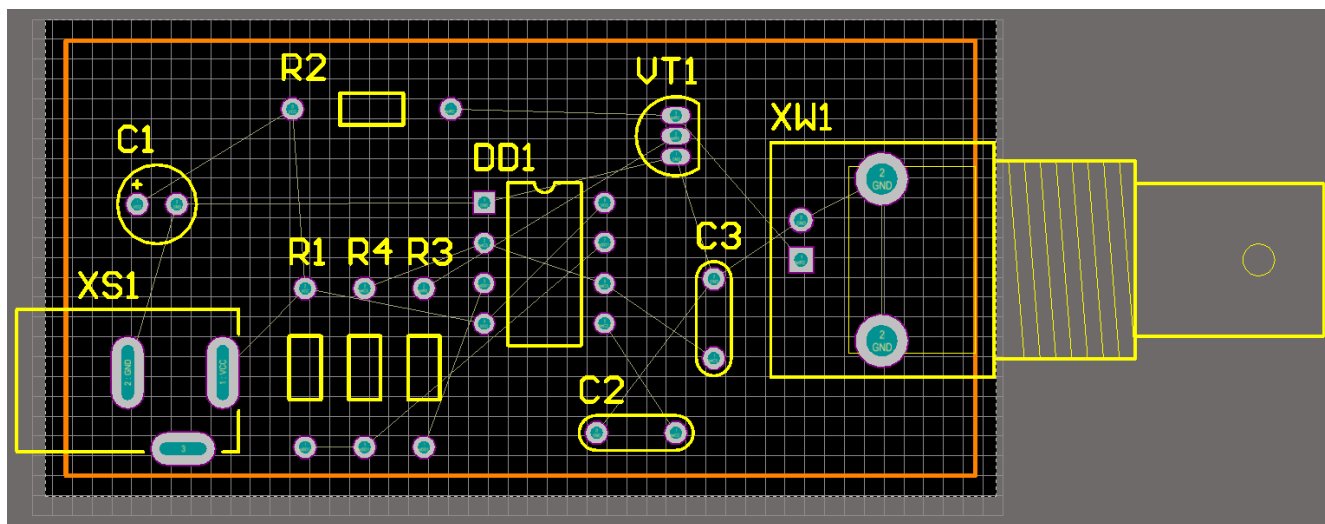


Рис. 11.21. Компоненти, що розміщені на платі

Нагадування!

Для повороту компонента на 90° проти годинникової стрілки необхідно натиснути клавішу <Пробіл> під час його переміщення. Подібним способом можна обернути і текстові мітки компонентів.

Altium Designer містить потужні інтерактивні засоби розміщення компонентів на платі. Скористаємося ними, щоб вирівняти на платі із заданим інтервалом ПМ резисторів R1, R4 та R3. Утримуючи клавішу <Shift>, клацніть ЛКМ на кожному ПМ цих резисторів, щоб вибрати їх. Кожен вибраний компонент

позначається зафарбованим прямокутником, колір якого задається системним параметром кольору *Selections*¹²⁸. Далі клацніть ПКМ на будь-якому із виділених ПМ резисторів і виберіть команду *Align* → *Align* з контекстного меню (або натисніть послідовність клавіш: <A, A>). У діалоговому вікні [Align Objects](#), що відкриється, встановіть опцію *Space Equally* в групі перемикачів *Horizontal* і опцію *Top* в групі перемикачів *Vertical* (рис. 11.22).

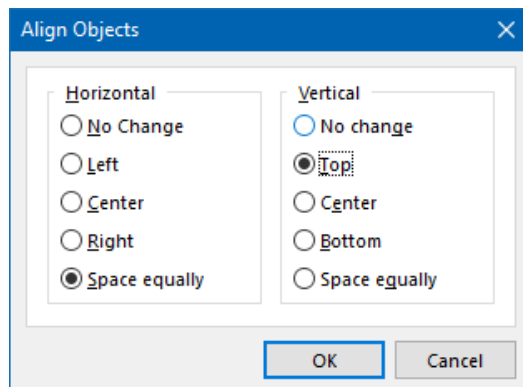


Рис. 11.22. Меню для вирівнювання ПМ резисторів

Натискання кнопки **OK** призведе до того, що ПМ резисторів будуть вирівняні по горизонталі з однаковим інтервалом в межах заданої локальної області ДП. Після цього клацніть мишкою у будь-якій точці робочої області, щоб зняти виділення з ПМ резисторів.

3.12. Автоматичне трасування плати

Для того, щоб запустити процес автоматичного трасування, необхідно виконати команду *Route* → *Auto Route* → *All*. Відкриється діалогове вікно [Situs Routing Strategies](#) (рис. 11.23), у верхній частині якого відображається налаштування звіту про трасування, а у нижній обирається стратегія трасування.

Зверніть увагу, що усі компоненти, які використовуються у цьому проєкті, мають корпуси, що призначені для монтажу в отвори. Таким чином можна спробувати виконати трасування лише на одному шарі *Bottom Layer*. Хоча в області *Routing Strategy* діалогового вікна *Situs Routing Strategies* немає у переліку стратегії для трасування на одному шарі, проте таке трасування можна виконати і на основі стратегії **Default 2 Layer Board**. Зробити це можна двома способами.

¹²⁸ Ви можете змінити цей колір у діалоговому вікні *View Configurations* (*Design* → *Board Layers & Colors* або клавіша <L>).

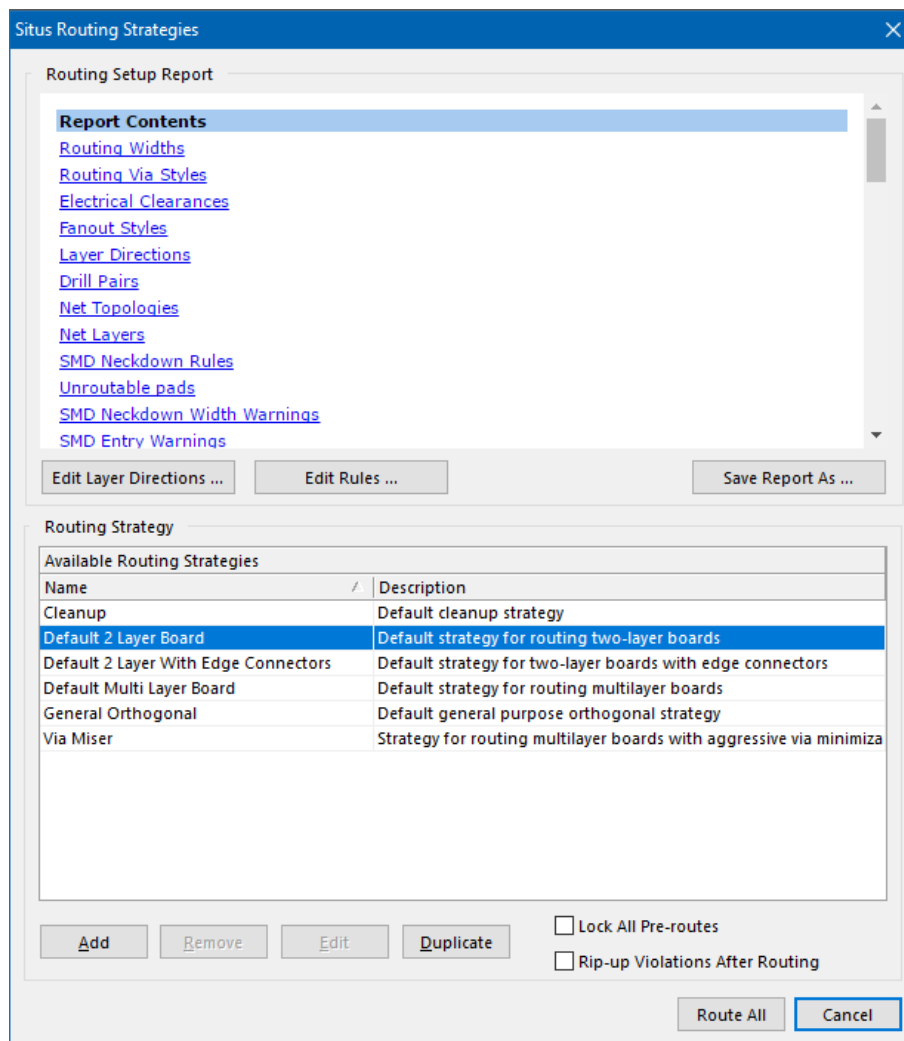


Рис. 11.23. Діалогове вікно *Situs Routing Strategies*

Перший спосіб. За обраної стратегії **Default 2 Layer Board** натисніть кнопку **Edit Layer Directions**. У вікні, що відкриється, для шару *Top Layer* у стовпчику *Current Settings* із випадного списку оберіть пункт *Not Used* (рис. 11.24).

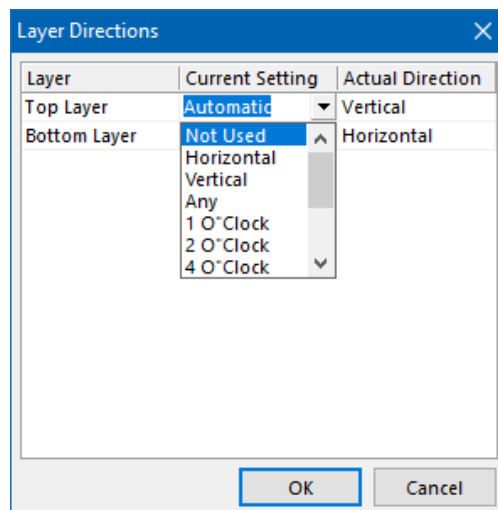


Рис. 11.24. Вікно налаштувань *Layer Directions*

Другий спосіб. За обраної стратегії **Default 2 Layer Board** натисніть кнопку **Edit Rules**. У цьому випадку ми можемо задати правило, яке визначатиме, на яких шарах плати має виконуватись трасування. У вікні, що відкриється, у категорії **Routing** оберіть підкатегорію **Routing Layers** та правило **RoutingLayers**. У правій частині вікна в області **Enabled Layers** приберіть прапорець із параметру **Allow Routing** для шару **Top Layer** (рис. 11.25).

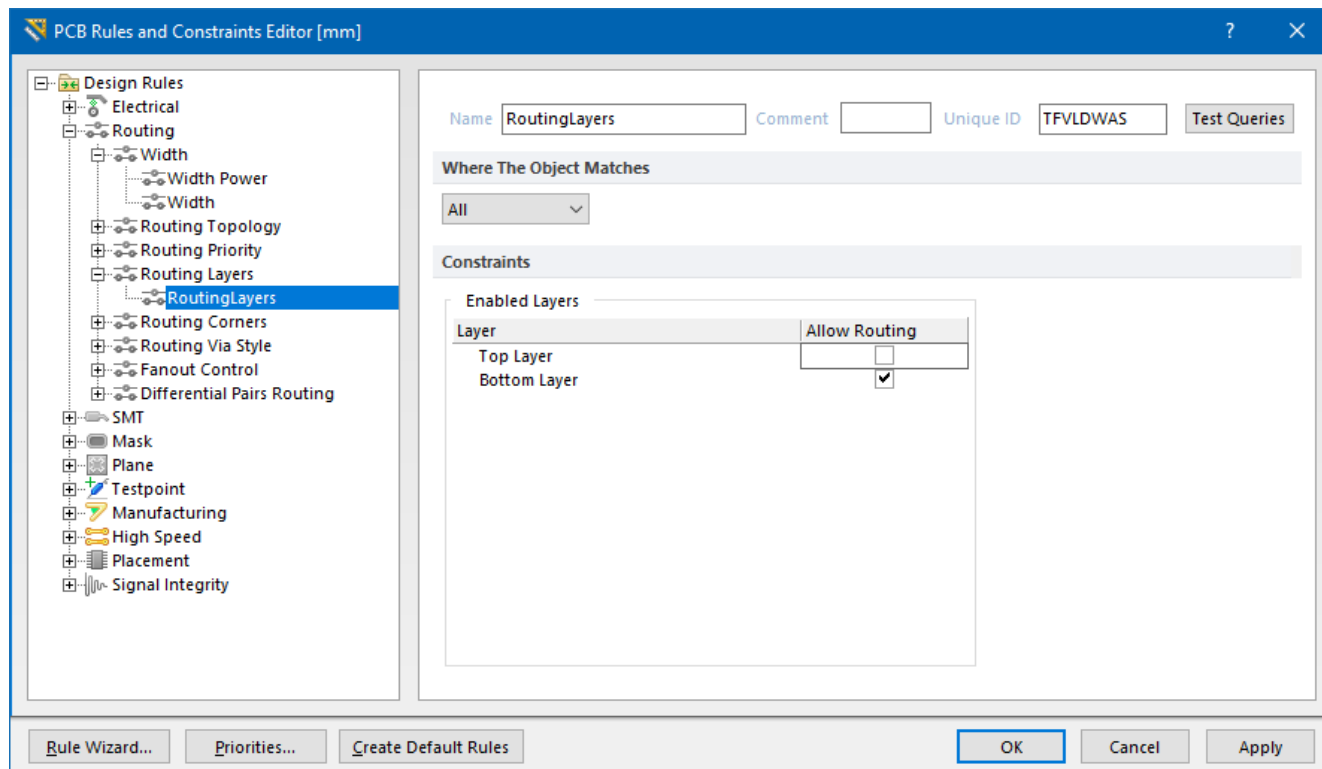


Рис. 11.25. Створення правила, за яким трасування відбуватиметься тільки на одній стороні плати

Після того, як було обрано перший або другий спосіб трасування тільки на одній стороні плати, необхідно натиснути кнопку **Route All** в правій нижній частині діалогового вікна **Situs Routing Strategies**. Буде запущено процес автоматичного трасування, після чого з'явиться панель **Messages** із відображенням інформаційних повідомлень. Якщо виникнуть попередження і помилки, то вони матимуть червоний колір. На них завжди слід звертати увагу, оскільки усі помилки мають бути виправлені, перш ніж технічна інформація про плату буде відправлена на виробництво для її виготовлення.

Нижче на рис. 11.26 показано один із можливих результатів трасування на одному шарі плати із використанням першого способу. Зверніть увагу, що ланцюги живлення та землі є ширшими за інші відповідно до окремого правила ширини, яке ми створили заздалегідь.

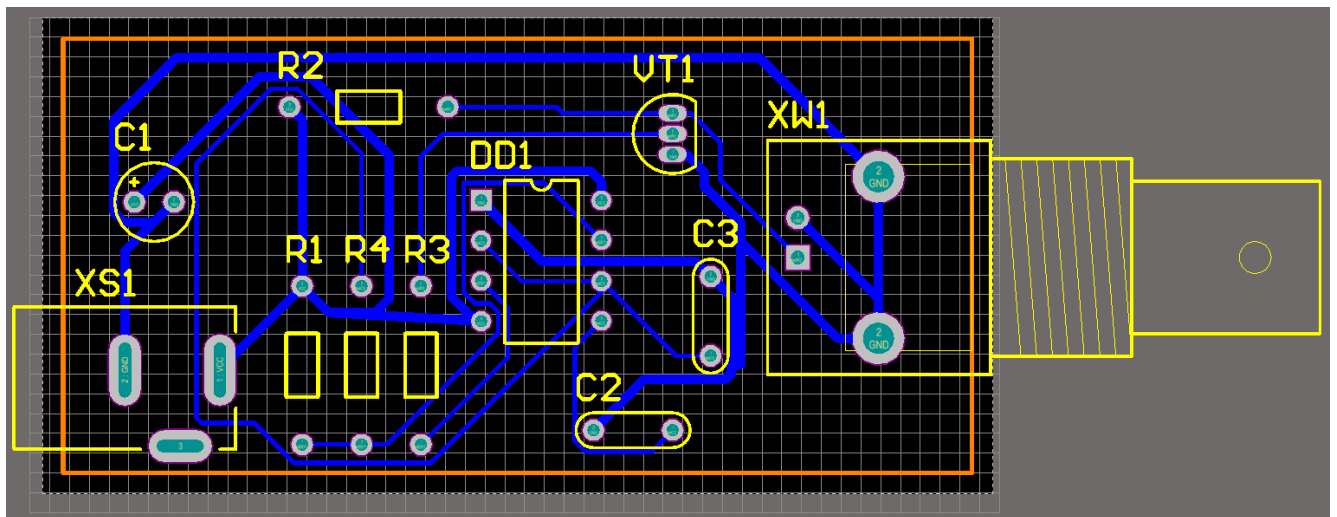


Рис. 11.26. Результати одностороннього автотрасування плати за першим способом

А на рис. 11.27 показано трасування плати на одному шарі із використанням другого способу.

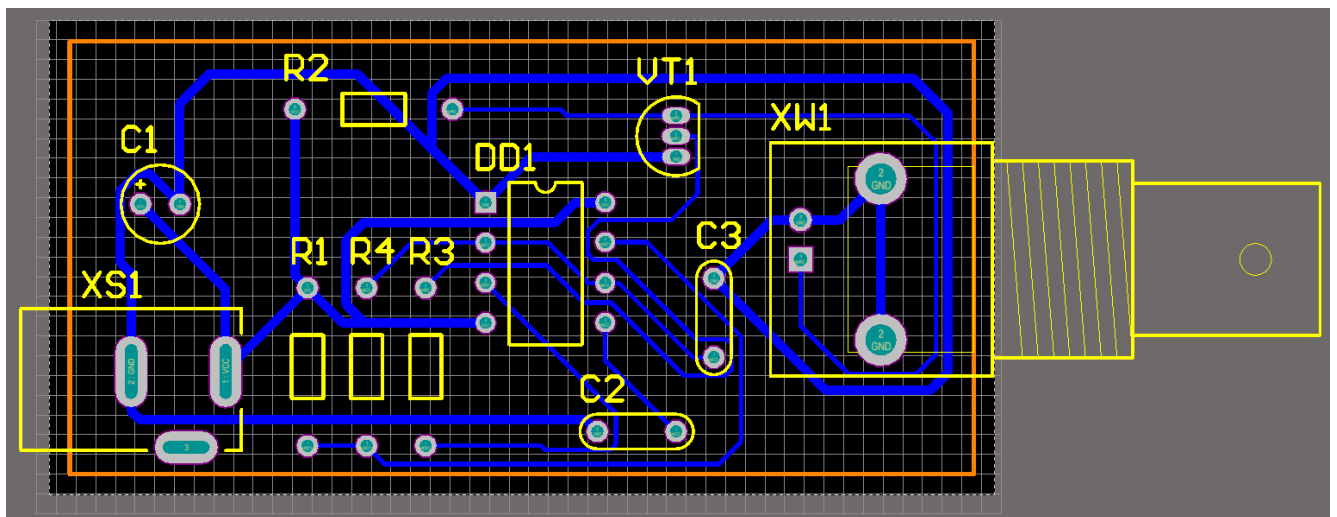


Рис. 11.27. Результати одностороннього автотрасування плати за другим способом

Не хвилюйтеся, якщо ваші результати відрізнятимуться від наведених на рис. 11.26 та рис. 11.27. Оскільки ви можете розмістити компоненти дещо інакше, то і результат трасування може бути іншим. Більше того, ви можете помітити, що результати автотрасування за першим та другим способом відрізняються між собою. Це нормально, оскільки вихідні умови для трасування кожним із способів є різними.

Після того, як автотрасувальник створив усі траси, ви можете залишитись незадоволеними деякими із прокладених трас, оскільки зрозуміло, що автотрасувальник не є ідеальним. Якщо вас щось не влаштовує, ви можете вручну

у режимі інтерактивного трасування виправити усі недоліки, наприклад, видаливши деякі ділянки траси і проклавши їх по-новому.

Якщо ж ви, наприклад, захочете видалити усі траси, щоб заново запустити автотрасування (можливо, із новими правилами або із новим розташуванням компонентів на платі), то спочатку необхідно виконати команду **Route** → **Un-Route** → **All**. Після цього усі траси будуть видалені.

Тепер, розуміючи, як працює автотрасувальник, спробуйте самостійно виконати трасування на двох сторонах плати. На рис. 11.28 представлений можливий варіант такого автотрасування.

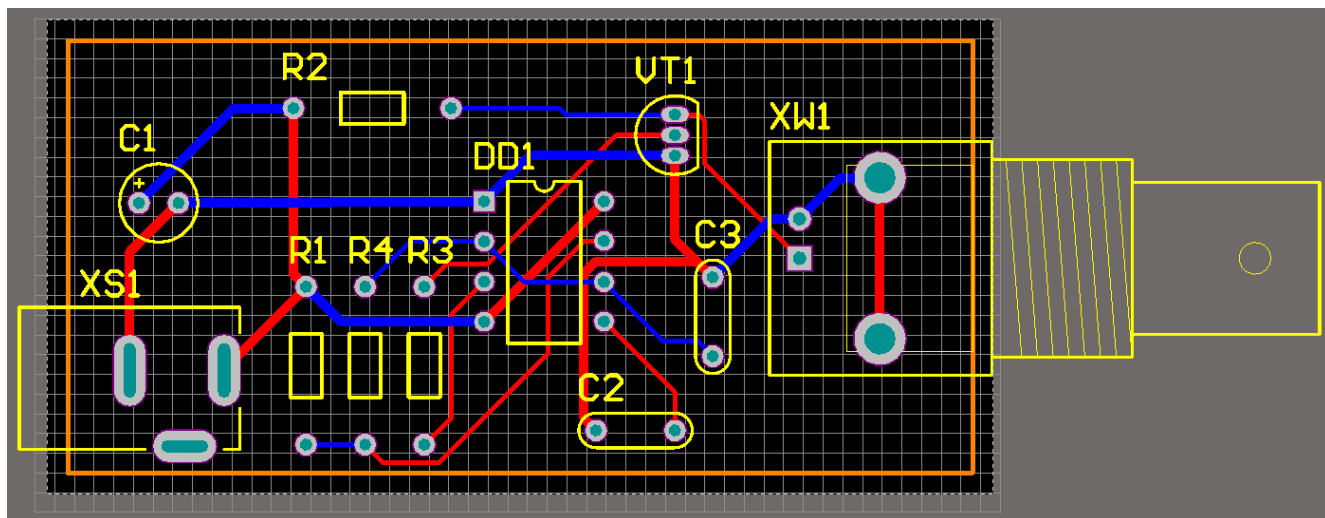


Рис. 11.28. Результат двостороннього автотрасування плати

Зверніть увагу, що доріжки в Редакторі плат відображаються тепер у двох кольорах: червоний колір означає, що траса розташовується на верхньому сигнальному шарі, синім кольором відмічені траси на нижньому шарі.

Для подальшої роботи залиште плату із двостороннім трасуванням. Не забудьте зберегти поточні результати.

3.13. Перевірка друкованої плати на відповідність заданим правилам

Зазвичай правила задаються на самому початку процесу проєктування, а перевірка їх виконання здійснюється як під час роботи над проєктом, так і на завершальному етапі.

Для того, щоб перевірити готову плату на відповідність заданим правилам, необхідно запустити механізм **Design Rule Check (DRC)**. Спочатку у меню **Design** виберіть пункт **Board Layers & Colors** (гаряча клавіша: <L>) та переконайтеся, що

у таблиці *System Colors* в рядку *DRC Error / Waived DRC Error Markers* увімкнена опція *Show*.

Далі у меню *Tools* виберіть команду *Design Rule Check* (послідовність клавіш: <T, D>). У діалоговому вікні *Design Rule Checker* налаштовуються правила для різних категорій (рис. 11.29). Якщо клацнути на обраній категорії (наприклад, *Electrical*), то у правій частині вікна будуть відображені відповідні правила. Для кожного правила доступні опції *Online* (постійна перевірка) і *Batch* (ручна перевірка). Правила ми залишимо незмінними, тобто заданими за замовчуванням.

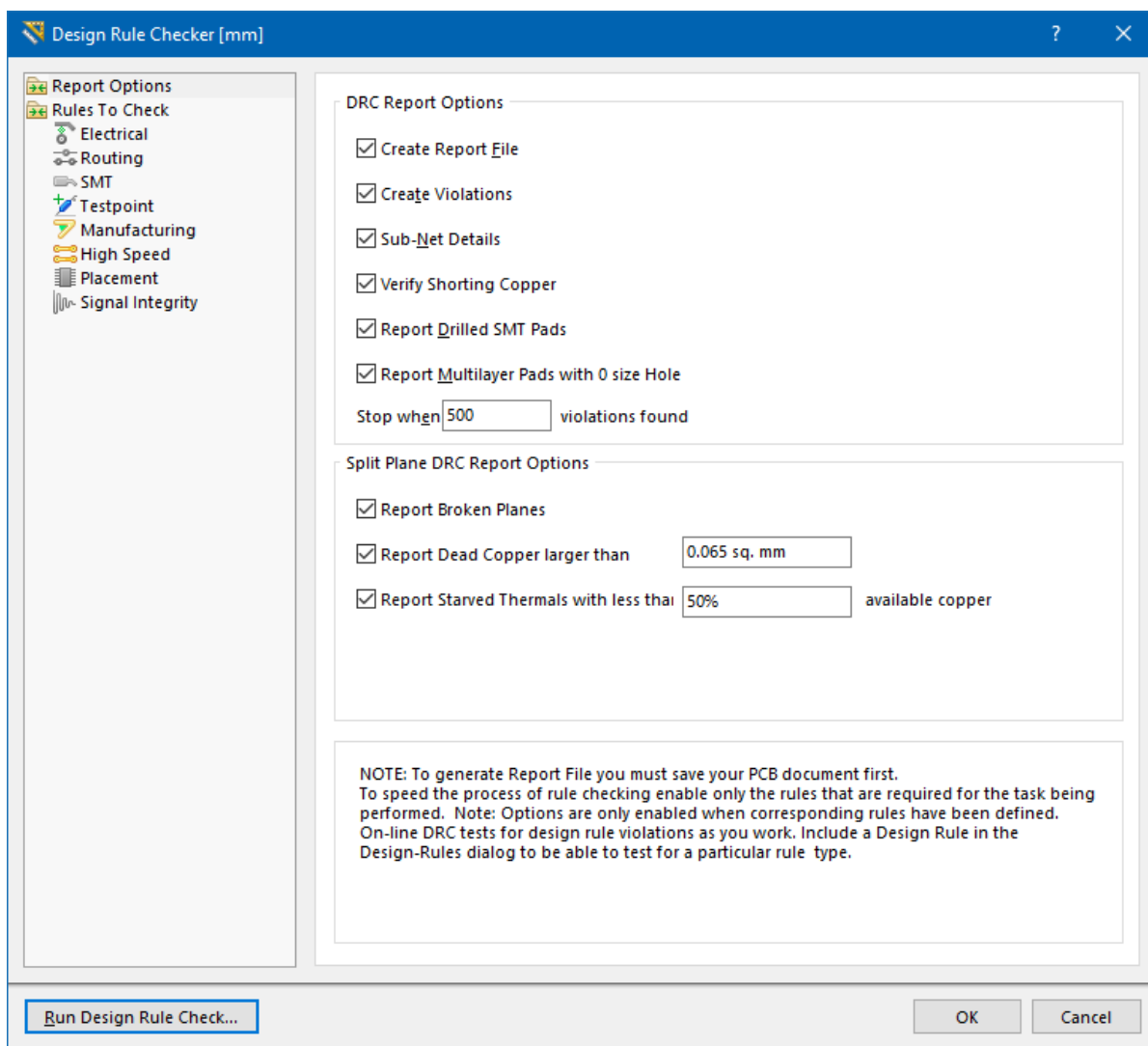


Рис. 11.29. Перевірка правил у діалоговому вікні *Design Rule Checker*

У лівій частині діалогового вікна оберіть пункт *Report Options*. Перегляньте, але не змінюйте встановлені опції, після чого натисніть кнопку **Run Design Rule Check**. Буде запущена перевірка, результати якої відображатимуться у новому вікні *Design Rule Verification Report*. Перегляньте розділ звіту, який називається

Rule Violations, і зверніть увагу на 3 типи виявлених порушень¹²⁹: **Hole Size Constraint (Min=0.025mm) (Max=2.54mm) (All)**, **Minimum Solder Mask Sliver (Gap=0.254mm) (All),(All)**, **Silk To Solder Mask (Clearance=0.254mm) (IsPad),(All)**.

Клацніть на посиланні **Hole Size Constraint (Min=0.025mm) (Max=2.54mm) (All)**, щоб перейти до відповідного розділу звіту, в якому описані всі випадки подібних порушень. Якщо тепер у цьому розділі клацнути на якому-небудь посиланні, ви перейдете до місця на ДП (рис. 11.30), в якому виявлено порушення¹³⁰.

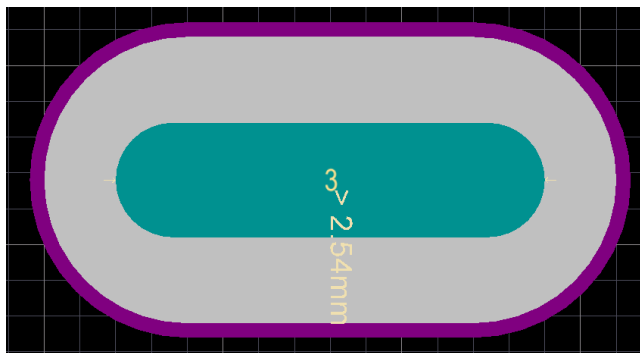


Рис. 11.30. КМ, що порушує правило **Hole Size Constraint**

Правило **Hole Size Constraint** визначає допустимі розміри для отворів КМ. За замовчуванням розміри отворів КМ не повинні перевищувати 2,54 мм. Однак у цьому проєкті використовується роз'єм DS-210, який має нестандартну форму отворів КМ, що не є помилкою. Тому необхідно відредагувати правило **Hole Size Constraint**.

[Задати нове або відредагувати існуюче правило можна у вікні редактора правил *PCB Rules and Constraints Editor*](#), яке можна відкрити, обравши у меню **Design** пункт **Rules**. Оскільки наразі ми хочемо змінити правило **Hole Size Constraint**, тому знаходимо у лівій частині вікна категорію **Manufacturing**, у якій відкриваємо тип правил **Hole Size** і обираємо наявне правило. Змініть максимальне значення діаметра отвору з 2,54 мм на 3,6 мм. Натисніть **OK**, щоб закрити вікно **PCB Rules and Constraints Editor**.

Знову запустіть перевірку проєкту на порушення правил (у меню **Tools** виберіть **Design Rule Check** і натисніть кнопку **Run Design Rule Check**).

¹²⁹ Три типи помилок, про які йде мова, – це помилки, які точно виникнуть під час розробки ДП у цьому комп'ютерному практикумі. Однак можуть виникнути і додаткові помилки внаслідок порушень інших правил, якщо ви щось робили не так, як написано в протоколі.

¹³⁰ Величина масштабування налаштовується на сторінці **System** → **Navigation** діалогового вікна **Preferences**. Поекспериментуйте, щоб знайти відповідне для себе значення.

Перегляньте результати у вікні **Design Rule Verification Report**, щоб впевнитись, що цього порушення більше немає у звіті.

Наступний тип порушення **Minimum Solder Mask Sliver** пов'язаний із тим, що мінімальна ширина ділянки в захисній масці є меншою за встановлену правилами (рис. 11.31). Паяльна захисна маска ДП – це покриття із міцного матеріалу (наприклад, на основі смол), що призначене для захисту провідників від забруднення, окислення, потрапляння припою і флюсу під час паяння, а також від перегріву. Маска закриває провідники і залишає відкритими КМ.

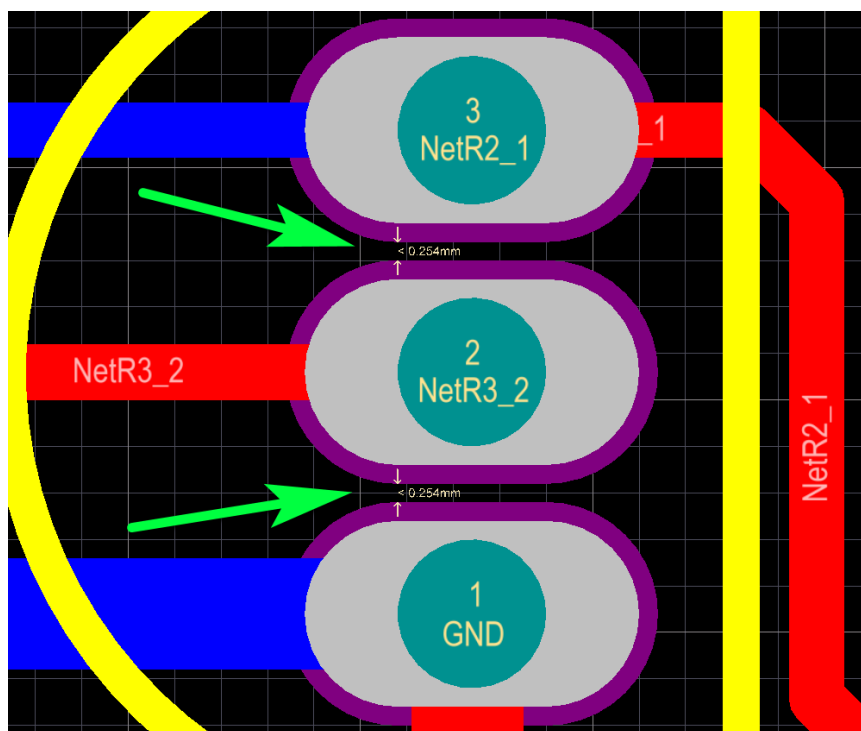


Рис. 11.31. Ширина маскового містка між КМ менше значення 0,254 мм, яке дозволене правилом **Minimum Solder Mask Sliver**

Якщо говорити про ділянку плати між КМ, то часто її називають масковим містком між КМ. Ширина маскового містка – це відстань між припусками паяльної маски двох сусідніх КМ, яка виконує захисну функцію.

Важливо розуміти, що чим менше масковий місток, тим важче його зробити і тим дорожчим буде виробництво такої плати. Тому, перш ніж встановлювати за допомогою правил ширину маскового містка, необхідно з'ясувати у виробника плати, які він має технологічні можливості. Наприклад, деякі виробники можуть забезпечити ширину містка 0,1 мм, а інші – не менше 0,15 мм тощо.

Щоб позбутись порушення **Minimum Solder Mask Sliver** можна або змінити правило, встановивши інше значення мінімальної ширини містка, або зменшити величину припуску паяльної маски (**Solder Mask Expansion**) КМ – своєрідний

«бортик» між КМ та паяльною маскою¹³¹. Ширина цього «бортика» задається у вікні властивостей КМ у розділі *Solder Mask Expansion*, де за замовчуванням активована опція *Expansion value from rules* (причому припуск паяльної маски становить 0,102 мм). «Бортик» між КМ та паяльною маскою охоплює по периметру весь КМ та позначається фіолетовим кольором у Редакторі посадкових місць і Редакторі плат.

У цьому проєкті ми змінимо як саме правило, вважаючи, що виробник може забезпечити мінімальну ширину містка 0,1 мм, так і величину припуску паяльної маски КМ. Спочатку відкрийте вікно *PCB Rules and Constraints Editor* та у категорії *Manufacturing* відредагуйте правило **Minimum Solder Mask Sliver**, встановивши значення параметра 0,1 мм (рис. 11.32).

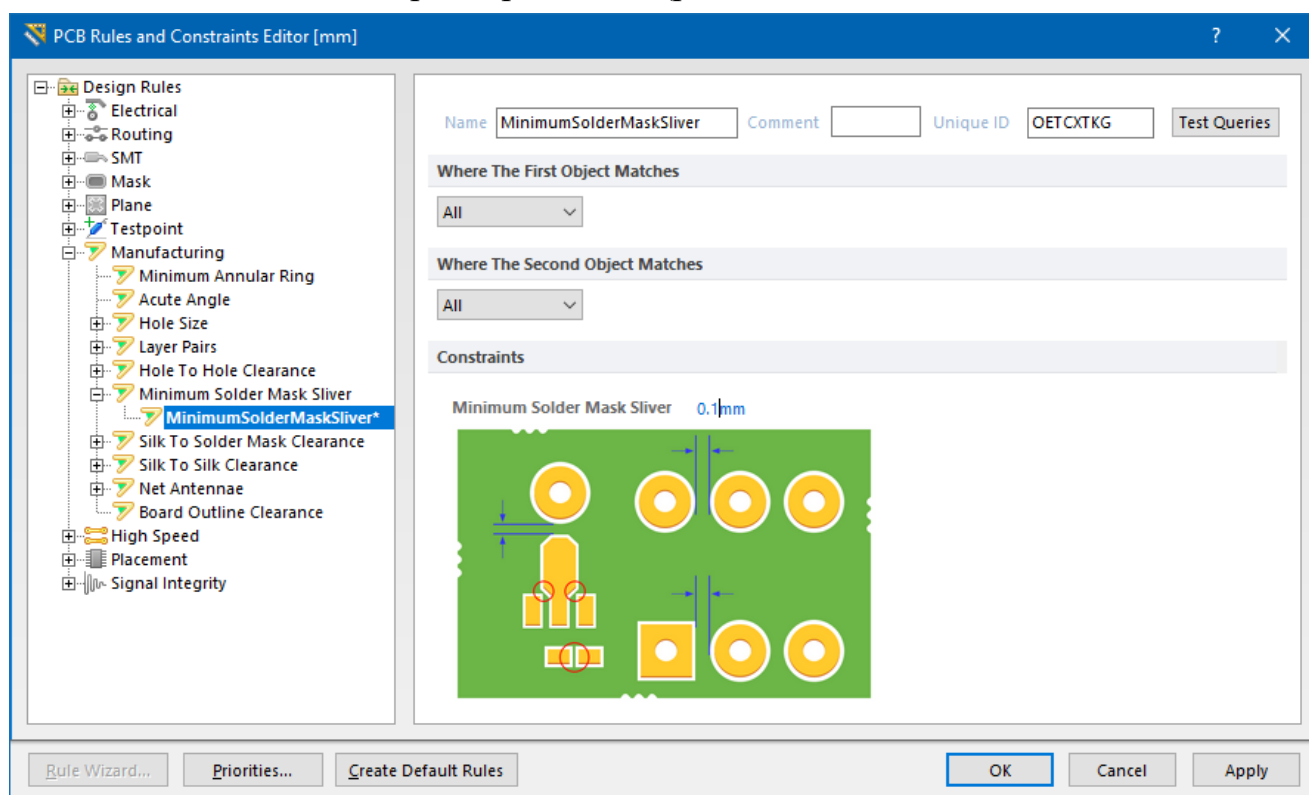


Рис. 11.32. Встановлення мінімальної ширини маскового містка між КМ

Далі для усіх трьох КМ транзистора змініть величину припуску паяльної маски. Знайдіть КМ транзистора на платі, відкрийте властивості кожного із них, після чого у розділі *Solder Mask Expansion* активуйте опцію *Specify expansion value*, а для параметра *Top* встановіть значення 0,1 мм (рис. 11.33).

¹³¹ В ідеалі, захисна маска має охоплювати всю плату і не перекривати КМ, проте внаслідок недосконалості технологічного процесу маска все ж може перекривати КМ, що не дозволить якісно припаяти до них виводи електронного компоненту. Щоб уникнути проблем під час паяння, **Altium Designer** надає можливість встановлювати зазор між маскою припою та КМ.

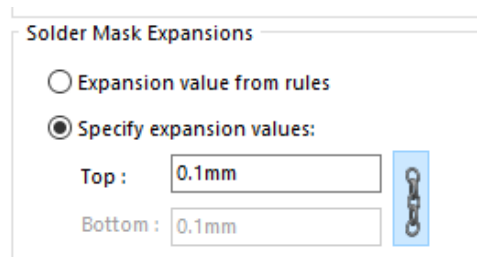
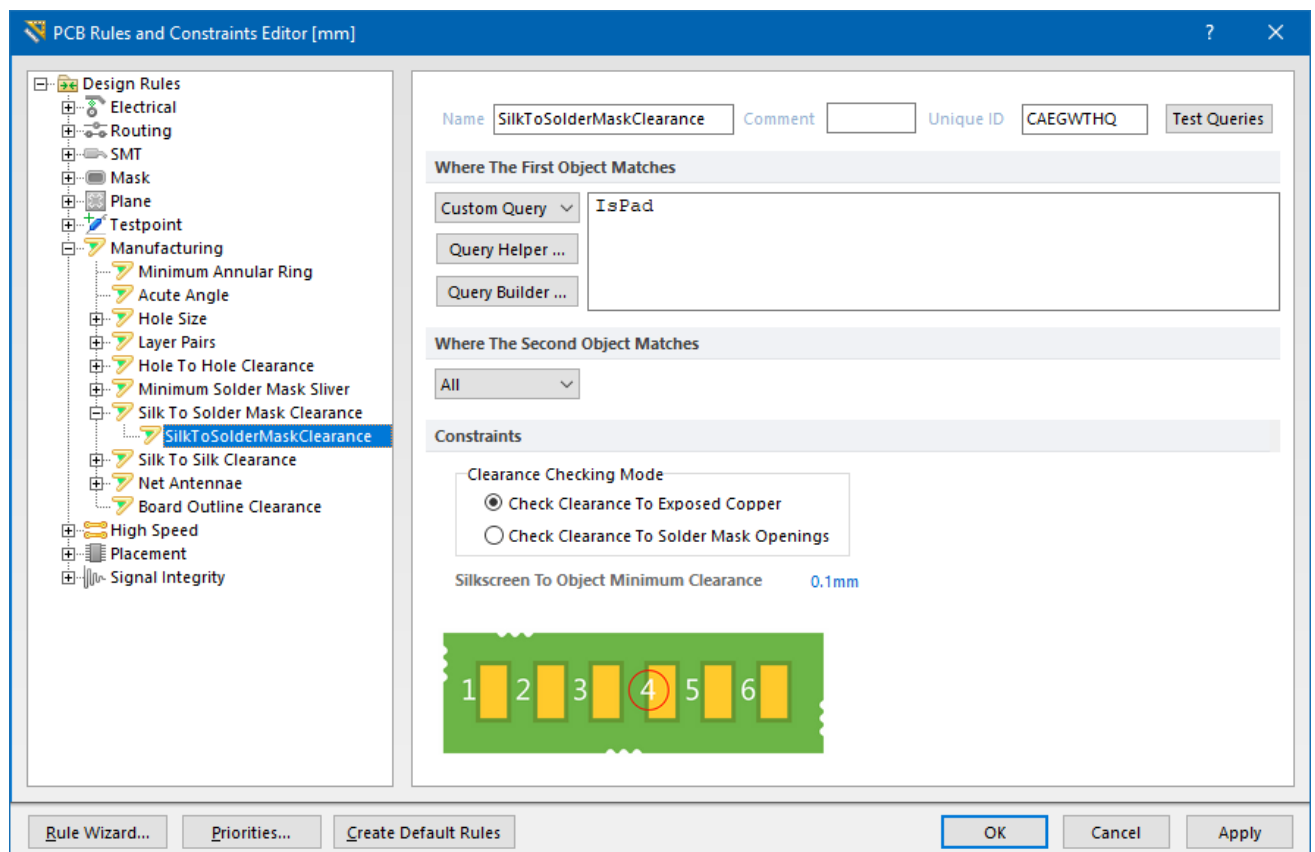


Рис. 11.33. Налаштування величини припуску паяльної маски

Знову запустіть перевірку проекту на помилки і перегляньте результати у вікні **Design Rule Verification Report**, щоб впевнитись, що порушення правила **Minimum Solder Mask Sliver** вже немає у звіті.

Тепер залишилось тільки одне порушення – **Silk To Solder Mask**. Це порушення пов'язане із тим, що відстань між деякими графічними елементами (контурами ПМ) на шарі шовкографії **Top Overlay** та елементами на шарі маски припою (**Top Solder** та **Bottom Solder**) є меншою, ніж встановлена відповідним правилом.

Відкрийте діалогове вікно **PCB Rules and Constraints Editor** та у категорії **Manufacturing** розгорніть тип правил **Silk to Solder Mask Clearance** і виберіть наявне правило (рис. 11.34). У лівій частині цього вікна відредагуйте правило **Silkscreen to Object Minimum Clearance**, встановивши нове значення 0,1 мм.

Рис. 11.34. Редагування правила **Silkscreen to Object Minimum Clearance**

Запустіть перевірку проєкту на помилки і перегляньте результати у вікні **Design Rule Verification Report**. На цей раз ви можете помітити, що хоча кількість порушень **Silk To Solder Mask** суттєво зменшилась, проте все ж залишилось кілька подібних порушень. Самостійно з'ясуйте, чому залишились деякі із порушень **Silk To Solder Mask**, та спробуйте вжити заходів, щоб їх позбутись.

Якщо під час перевірки проєкту у вас присутні інші порушення, які не описані у цьому комп'ютерному практикумі, – це означає, що або на етапі створення бібліотечного компонента ви допустили помилку, або під час розміщення компонентів у цьому комп'ютерному практикумі щось зробили не так. У такому випадку самостійно вивчіть повідомлення про порушення, з'ясуйте суть порушення та усуньте його.

Наразі окремо розглянемо порушення **Clearance Constraint**, яке додатково може виникнути у цьому комп'ютерному практикумі, та способи його усунення.

Правило **Clearance Constraint** визначає мінімальний зазор, що дозволений між будь-якими двома примітивами на шарі міді. Це правило задається в розділі **Clearance** категорії **Electrical** (рис. 11.35).

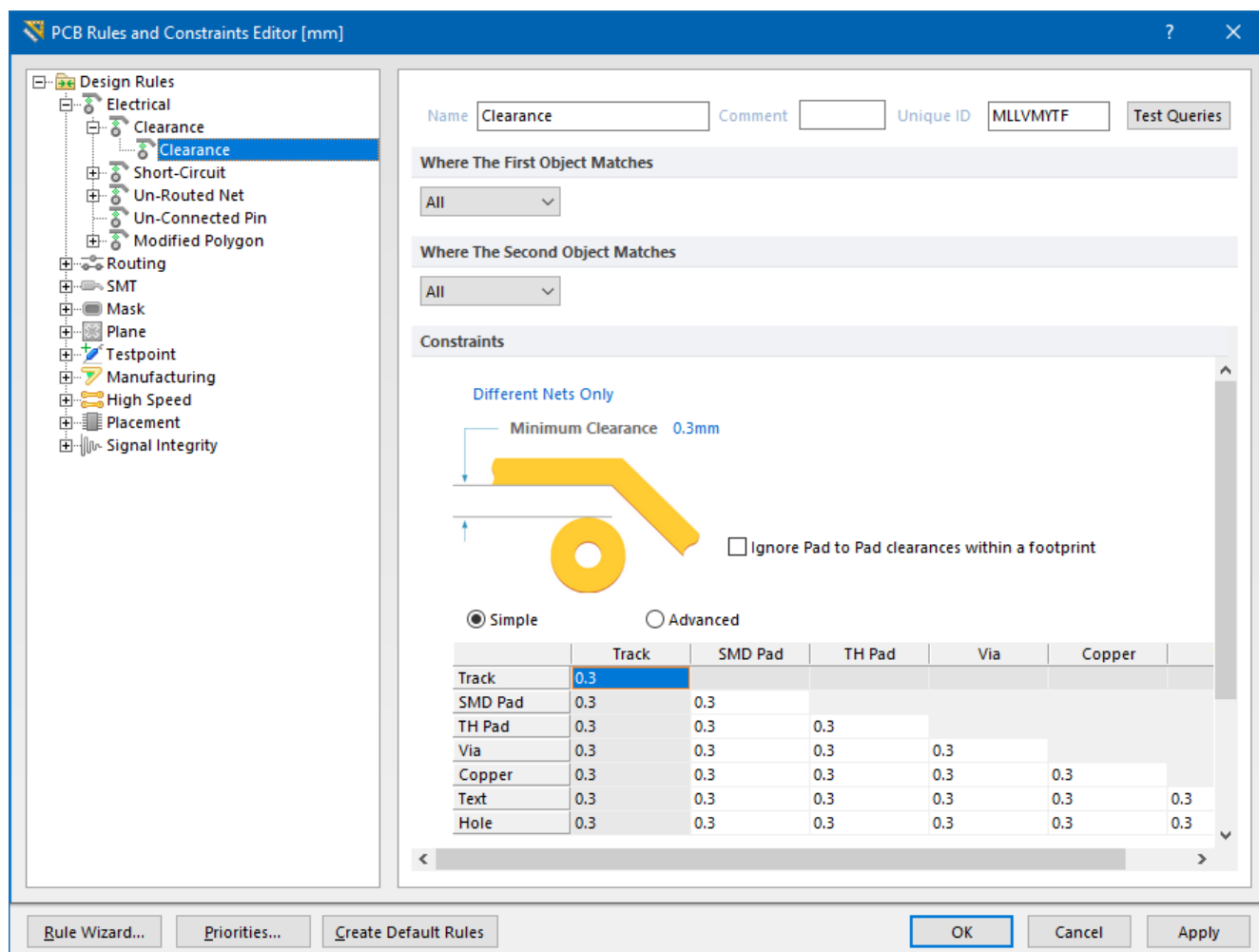


Рис. 11.35. Налаштування правила **Clearance Constraint**

Для правила **Clearance Constraint** можна вказати або одне значення зазору, яке буде використовуватись для усіх примітивів (параметр *Minimum Clearance*), або різні зазори для різних комбінацій примітивів (об'єктів), використовуючи спеціальну матрицю мінімальних зазорів, яка забезпечує гнучкість у створенні стислого та цілеспрямованого набору правил.

Наприклад, якщо помилка **Clearance Constraint** виникла тому, що відстань між КМ деякого ПМ виявилась менше мінімального встановленого значення, то можна встановити нове менше значення зазору між КМ. Для цього необхідно або змінити значення параметру *Minimum Clearance* (але тоді це вплине й на усі інші примітиви), або у матриці мінімальних зазорів на перетині рядка **TH Pad** та стовпчика **TH Pad** окремо задати потрібне значення. Іншим варіантом вирішення проблеми виникнення помилки внаслідок малого зазору між КМ є встановлення прапорця *Ignore Pad to Pad clearances within a footprint*, який дозволяє ігнорувати невідповідність зазорів між КМ в межах одного ПМ.

Після того, як були усунені усі порушення правил, у стовпчику **Count** таблиці **Rule Violations** повинні відображатись тільки нулі (рис. 11.36).

Design Rule Verification Report	
Date:	30.11.2020
Time:	17:33:13
Elapsed Time:	00:00:01
Filename:	C:\Users\Public\Documents\Altium\Projects\PulseGenerator\PulseGenerator.PcbDoc
Warnings:	0
Rule Violations:	0
Summary	
Warnings	Count
Total	0
Rule Violations	Count
Clearance Constraint (Gap=0.3mm) (All),(All)	0
Short-Circuit Constraint (Allowed=No) (All),(All)	0
Un-Routed Net Constraint ((All))	0
Modified Polygon (Allow modified: No), (Allow shelved: No)	0
Width Constraint (Min=0.3mm) (Max=1.2mm) (Preferred=0.6mm) ((InNet('VCC') OR InNet('GND')))	0
Width Constraint (Min=0.3mm) (Max=0.3mm) (Preferred=0.3mm) (All)	0
Power Plane Connect Rule(Relief Connect)(Expansion=0.508mm) (Conductor Width=0.254mm) (Air Gap=0.254mm) (Entries=4) (All)	0
Hole Size Constraint (Min=0.025mm) (Max=3.6mm) (All)	0
Hole To Hole Clearance (Gap=0.254mm) (All),(All)	0
Minimum Solder Mask Sliver (Gap=0.1mm) (All),(All)	0
Silk To Solder Mask (Clearance=0.1mm) (IsPad),(All)	0
Silk to Silk (Clearance=0.254mm) (All),(All)	0
Net Antennae (Tolerance=0mm) (All)	0
Height Constraint (Min=0mm) (Max=25.4mm) (Preferred=12.7mm) (All)	0
Total	0

Рис. 11.36. Звіт про перевірку правил проєктування, що не містить порушень

3.14. Формування файлу специфікації BoM

Перелік використовуваних електронних компонентів є важливим технічним документом у будь-якому проєкті ДП. Для його формування в **Altium Designer** використовується документ **Bill of Materials (BoM)**. Насправді **BoM** – це більше, ніж просто перелік електронних компонентів. Він містить найрізноманітнішу інформацію як про електронні компоненти, так і про плату, зокрема: кількість компонентів, тип, вартість виробів, посилання на сайти виробників та продавців тощо (частина цієї інформації береться із параметрів бібліотечних компонентів).

Для того, щоб створити документ BoM, необхідно виконати команду **Reports** → **Bill of Materials**¹³². У діалоговому вікні, що з'явиться (рис. 11.37), представлені різні параметри компонентів, які будуть включені до фінального звіту. Для кожного параметра виділено окремий стовпчик.

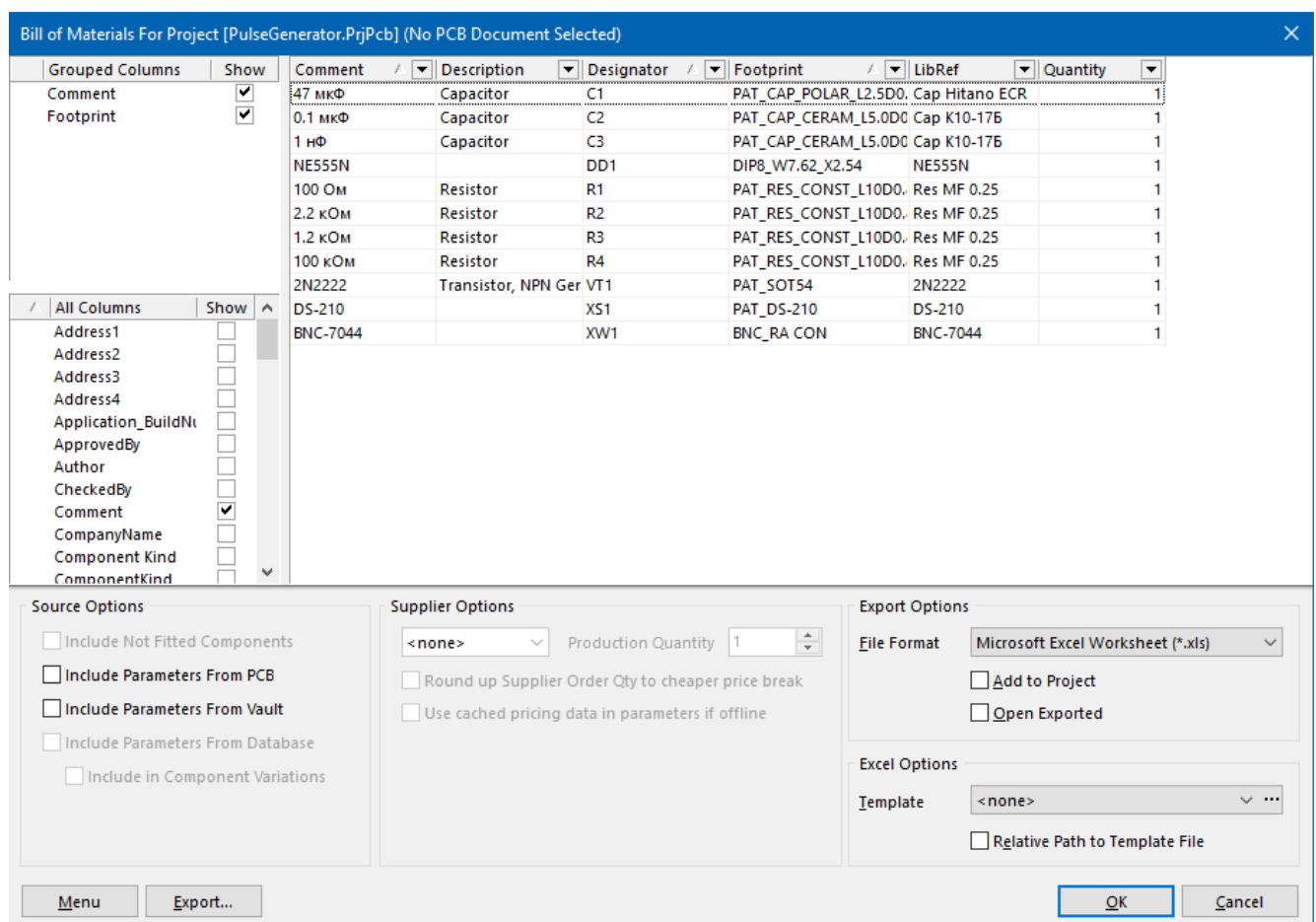


Рис. 11.37. Вікно *Bill of Materials For PCB Document*

¹³² Інший спосіб формування BoM полягає у тому, що необхідно натиснути ПКМ на назві проєкту у панелі **Projects** та у випадному меню обрати **Add New to Project** → **BoM**.

В області **All Columns** розробник може обирати на власний розсуд, які дані включати до згенерованого звіту специфікації (для даних, які необхідно відображати у звіті, у стовпчику **Show** має бути встановлений прапорець). Дані можуть бути згруповані, відсортовані та відфільтровані за необхідності, а також можуть містити додаткову інформацію із пов'язаної бази даних, якщо це потрібно.

За замовчуванням звіт зберігається у файлі таблиць Excel, однак можна обрати й інші формати.

Наразі ми не будемо вносити ніяких змін до автоматично формованого звіту. Для того, щоб експортувати звіт в обраний формат файлу, натисніть кнопку **Export**. У вікні збереження файлу задайте його назву та за потреби змініть шлях, за яким будете його зберігати. Після збереження перегляньте вміст файлу.

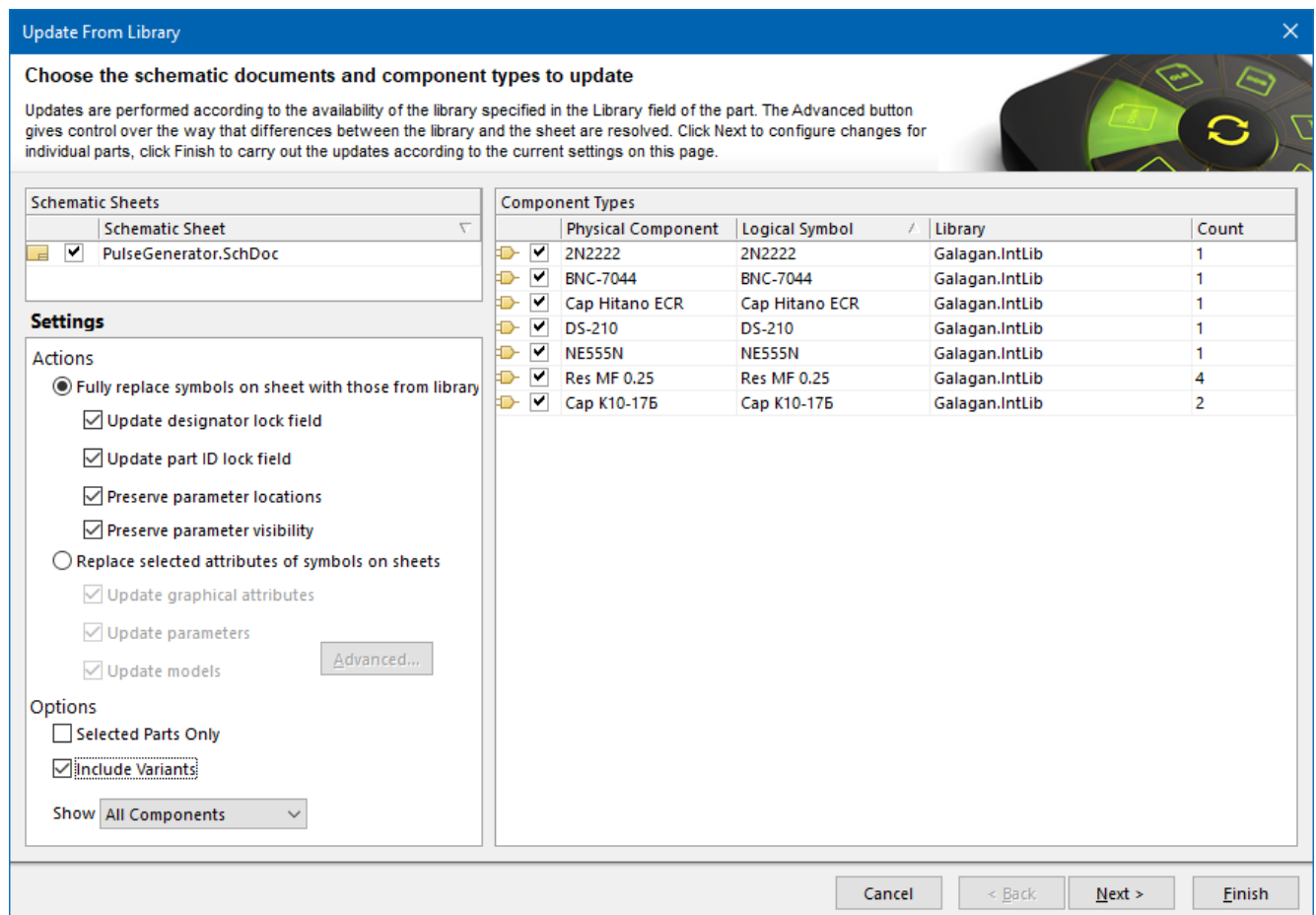
3.15. Заміна мікросхеми таймеру NE555N на NE555D на принциповій схемі та синхронізація проєктних даних

Мікросхема таймеру NE555N має корпус DIP. Припустімо, що внаслідок деяких міркувань нам потрібно у проєкті використовувати мікросхему NE555D у корпусі SOP.

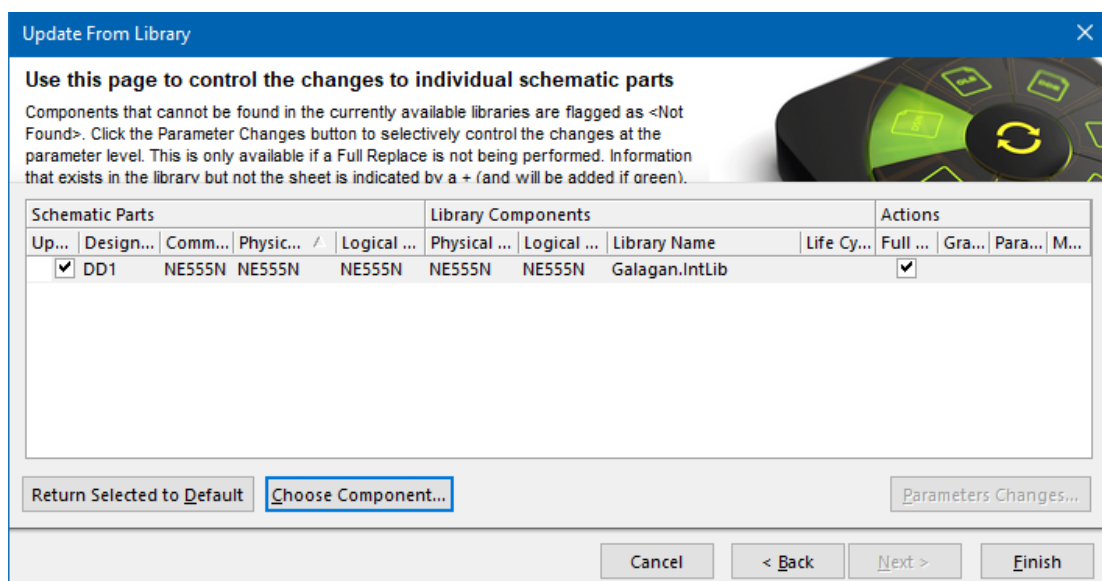
Для того, щоб виконати заміну одного електронного компонента на інший у Редакторі схем, можна застосувати різні підходи. Прямий підхід полягає у видаленні попереднього компонента і додаванні нового із бібліотеки. Водночас необхідно враховувати, що доведеться заново створювати з'єднання у Редакторі схем, виконувати компіляцію та синхронізацію даних. Якщо компаратор виявить усі зміни і коректно їх врахує, тоді все гаразд. Але за найгіршого сценарію доведеться також заново розміщувати компоненти на платі та виконувати трасування. Наразі поекспериментуйте і самостійно виконайте заміну розглянутим способом.

Ми ж далі розглянемо інший підхід, за допомогою якого також можна виконати заміну. Для цього із меню **Tools** виконайте команду **Update From Libraries**¹³³, після чого відкриється вікно [**Update From Library**](#) (рис. 11.38).

¹³³ Насправді команда **Update From Libraries** використовується у першу чергу тоді, коли ви створили проєкт, додали із бібліотеки усі компоненти, але з часом змінили деякі компоненти у самій бібліотеці. Нові зміни не підтягнуться автоматично у проєкт. Щоб їх відобразити, саме й використовується команда **Update From Libraries**.

Рис. 11.38. Вікно *Update From Library*

Оскільки нам потрібно замінити тільки мікросхему NE555N, тому в області **Component Types** прибираємо усі галочки, окрім тієї, що стоїть навпроти NE555N. Натисніть кнопку **Next**. У наступному вікні (рис. 11.39) натисніть кнопку **Choose Component**.

Рис. 11.39. Проміжне вікно, у якому необхідно натиснути кнопку **Choose Component**

Після натискання кнопки **Choose Component** відкриється нове вікно **Browse Library** (рис. 11.40), у якому оберіть власну бібліотеку, а зі списку компонентів – мікросхему NE555D.

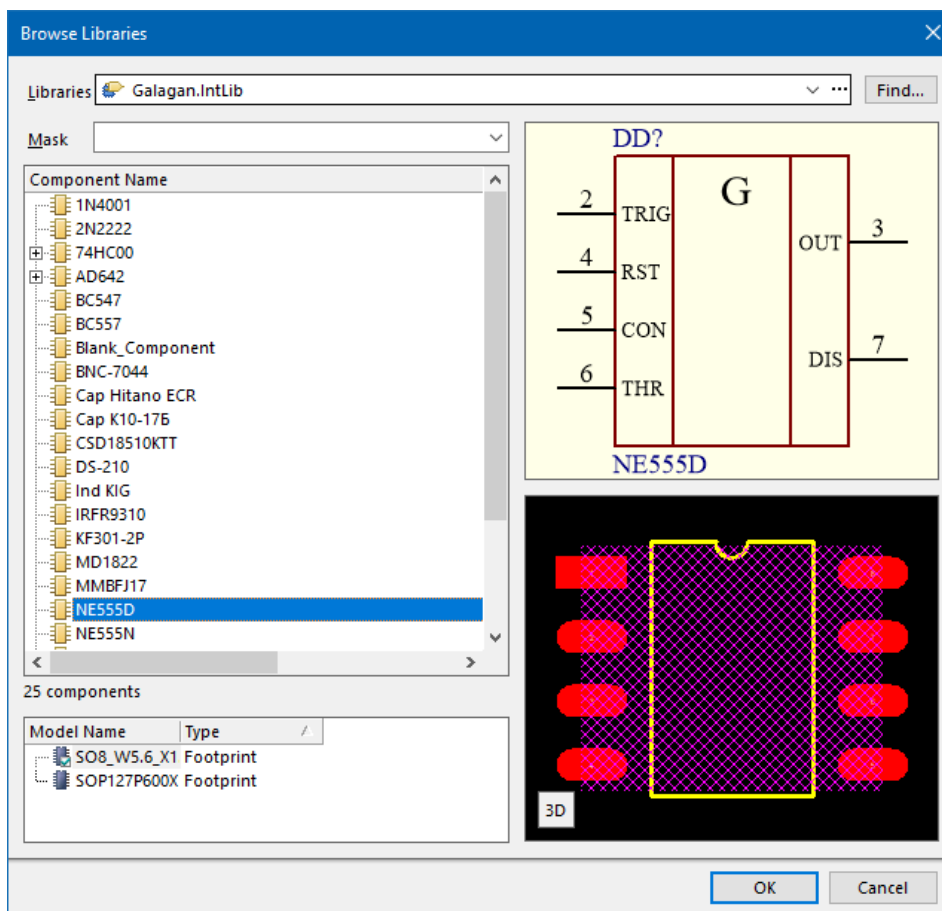


Рис. 11.40. Вікно **Browse Library**

Після вибору нового компоненту ви знову повернетесь у вікно **Update From Library** (див. рис. 11.39), проте тепер у стовпчику **Library Component** відображатиметься компонент NE555D. Натисніть кнопку **Finish**. З'явиться нове вікно **Engineering Change Order**, у якому спочатку натисніть кнопку **Validate Changes** і, якщо у стовпчику **Check** з'являться усі зелені галочки, натисніть кнопку **Execute Changes**. Після закриття цього вікна ви опинитесь у Редакторі схем, де буде розташована вже нова мікросхема таймера, проте не підключена до з'єднань, як попередня. Підберіть крок сітки таким чином, щоб переміщуючи УГП мікросхеми потрапити її виводами у ті ж з'єднання, які залишились від попередньої схеми (про правильність підключення виводів до з'єднань буде свідчити поява невеликої зеленої галочки у місці контакту).

Тепер із меню **Design** виконайте команду **Update PCB Document PulseGenerator.PcbDoc**. З'явиться вікно **Engineering Change Order** (рис. 11.41), у

якому прибери́ть галочку у рядку **Add Rooms**. Послідовно натисніть кнопки **Validate Changes** та **Execute Changes**, після чого закрийте вікно **Engineering Change Order**.

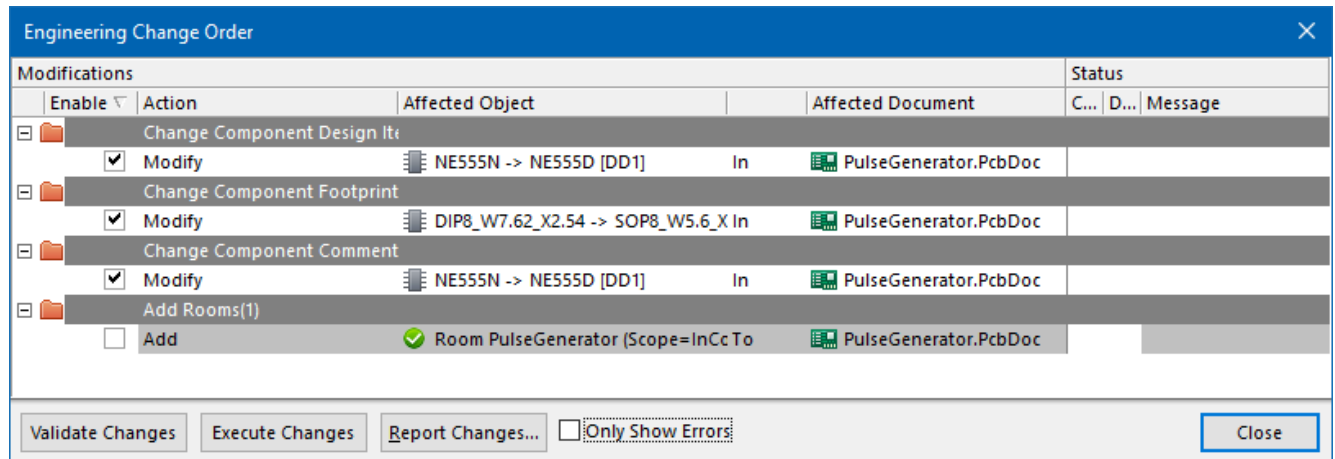


Рис. 11.41. Вікно **Engineering Change Order**

Після виконання усіх зазначених дій плата матиме такий вигляд, як на рис. 11.42. Як бачимо, ПМ мікросхеми таймеру змінилось. Проте зверніть увагу на зелені траси. Це означає помилку. Так має бути, адже внаслідок заміни корпусу DIP на SOP тепер попередні траси не з'єднані із новими КМ.

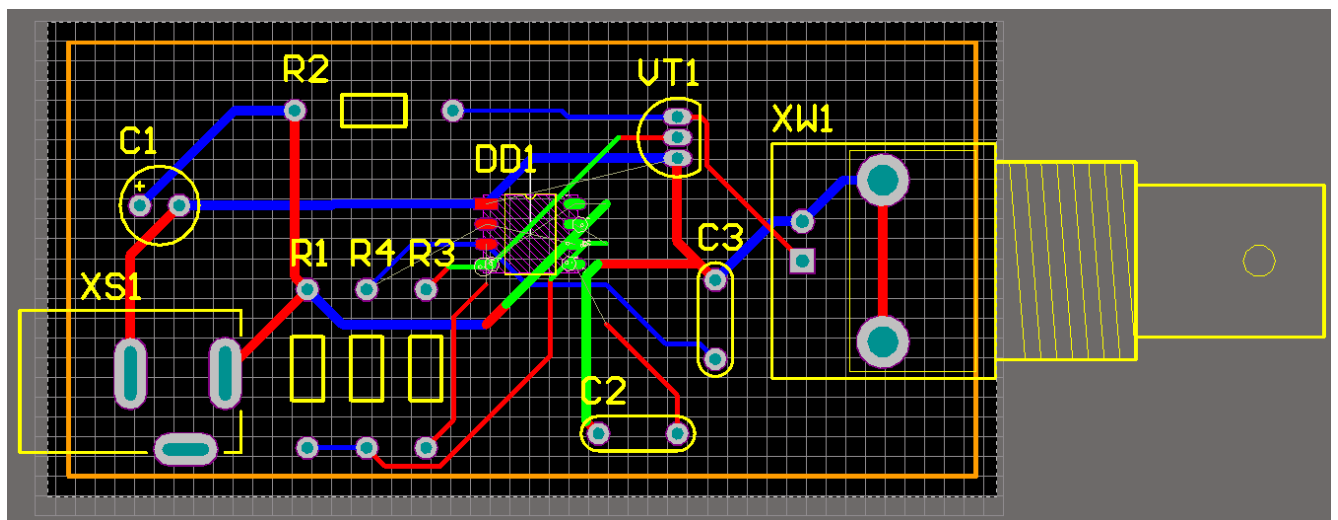


Рис. 11.42. Результат автоматичної заміни корпусу мікросхеми

Для того, щоб прибрати подібні помилки, можна виконати заново трасування тільки тих доріжок, які виявились непідключеними. А можна заново виконати трасування усієї плати. Саме так буде зроблено далі.

Спочатку виконайте команду **Route** → **Un-Route** → **All** для видалення усіх існуючих трас. Після цього, використовуючи команду **Route** → **Auto Route** → **All** та обравши на власний розсуд стратегію, виконайте нове трасування плати. Один із можливих результатів представлено на рис. 11.43.

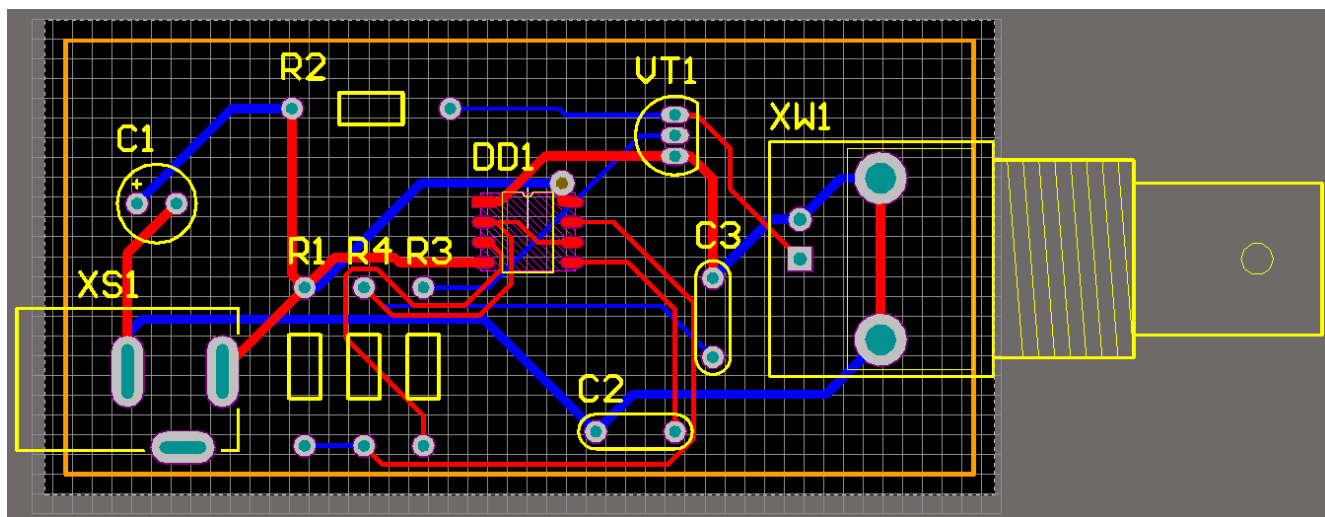


Рис. 11.43. Варіант автоматичного трасування плати у випадку використання мікросхеми таймера у корпусі SOP

3.16. Заміна посадкового місця мікросхеми таймеру NE555D

У комп'ютерному практикумі №5 для мікросхеми NE555D було розроблено два ПМ за допомогою різних інструментів, а саме: *Component Wizard* та *IPC Compliant Footprint Wizard*. Нехай спочатку на платі було використано ПМ **SOP8_W5.6_X1.27**. Замінімо його на ПМ **SOP127P600X180-8M**, що було створено за допомогою *IPC Compliant Footprint Wizard*.

Для виконання заміни подвійно клацніть ЛКМ на ПМ **SOP8_W5.6_X1.27**, щоб відкрити вікно властивостей. Знайдіть у цьому вікні область *Footprint*. У полі *Name* натисніть піктограму із трьома точками для відкриття вікна пошуку нового ПМ. Переконайтеся, що обрана ваша власна бібліотека. Знайдіть та виділіть ПМ **SOP127P600X180-8M**. Натисніть кнопку **ОК** для застосування змін. Якщо все зроблено правильно, то результат має бути таким, як представлено на рис. 11.44.

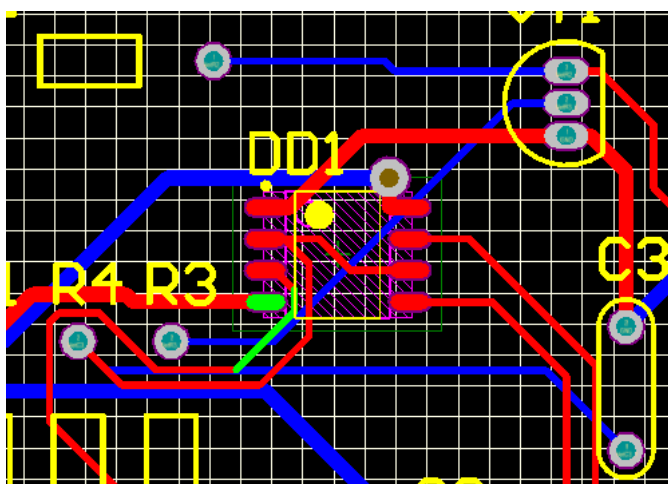


Рис. 11.44. Результат заміни ПМ **SOP8_W5.6_X1.27** на **SOP127P600X180-8M**

Оскільки ПМ для мікросхеми NE555D створювались за допомогою різних інструментів, то очевидно, що вони дещо відрізняються. Саме тому після заміни виникли проблеми із деякими трасами, які тепер підсвічуються зеленим кольором. Ця помилка виникла тому, що відстань між 4-м КМ та найближчою доріжкою порушує правило **Clearance Constraint**¹³⁴. Усуньте це порушення або зміною правила, або шляхом зміщення доріжки дещо далі від КМ.

Розглянутий підхід заміни є універсальним і дає змогу замінити деяке ПМ на будь-яке інше із бібліотеки. Але тут завжди потрібно бути обережним і уважним, щоб не виникало помилок.

Конкретно у випадку, коли до одного УГП додано кілька ПМ, краще скористатись іншим підходом. У Редакторі схем необхідно обрати потрібний компонент і викликати його властивості. У субвікні **Models** вікна властивостей компонента із випадного списку підключених ПМ (у стовпчику **Name**) обрати нове, після чого натиснути **OK**. Далі із головного меню **Design** виконати команду **Update PCB Document PulseGenerator.PcbDoc**. З'явиться вікно **Engineering Change Order**, у якому буде міститись інформація, що представлена на рис. 11.45. Після натискання усіх необхідних кнопок у цьому вікні відбудеться автоматична заміна одного ПМ на інше.

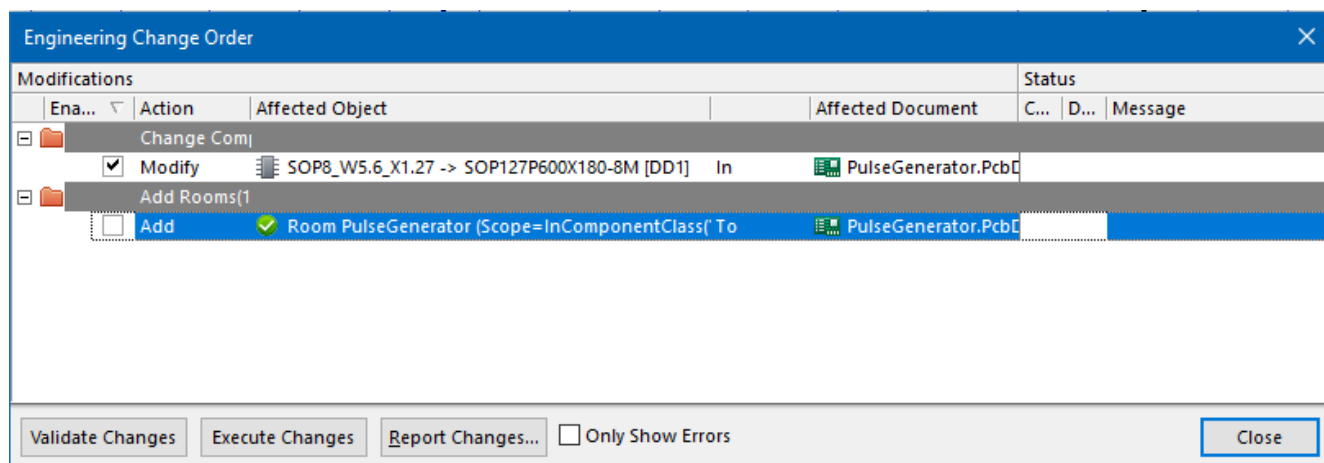


Рис. 11.45. Вікно **Engineering Change Order**

Насамкінець, існує третій спосіб зміни ПМ. Для цього у Редакторі схем із меню **Tools** необхідно виконати команду **Footprint Manager**. Після чого в

¹³⁴ Якщо у вас ця помилка не виникла, значить у вашому проєкті правило **Clearance Constraint** не порушується. Можливо, ви раніше його інакше налаштували, а, можливо, внаслідок автоматичного трасування у вас був сформований трохи інший провідний малюнок.

однойменному вікні потрібно для заданого компонента обрати нове ПМ. А далі стандартно виконується команда *Update PCB Document PulseGenerator.PcbDoc* для внесення змін.

3.17. Перегляд плати у 3D режимі

Щоб перейти у 3D режим, виберіть з меню *View* команду *3D Layout Mode* (або натисніть клавішу <3>). Тепер плата відображатиметься як тривимірний об'єкт.

Якщо ви, послідовно виконуючи комп'ютерні практикуми, додавали тривимірні моделі корпусів до усіх компонентів, які використовувались для створення схеми генератора тактових імпульсів, то результат перегляду у 3D режимі може бути приблизно таким, як показано на рис. 11.46.

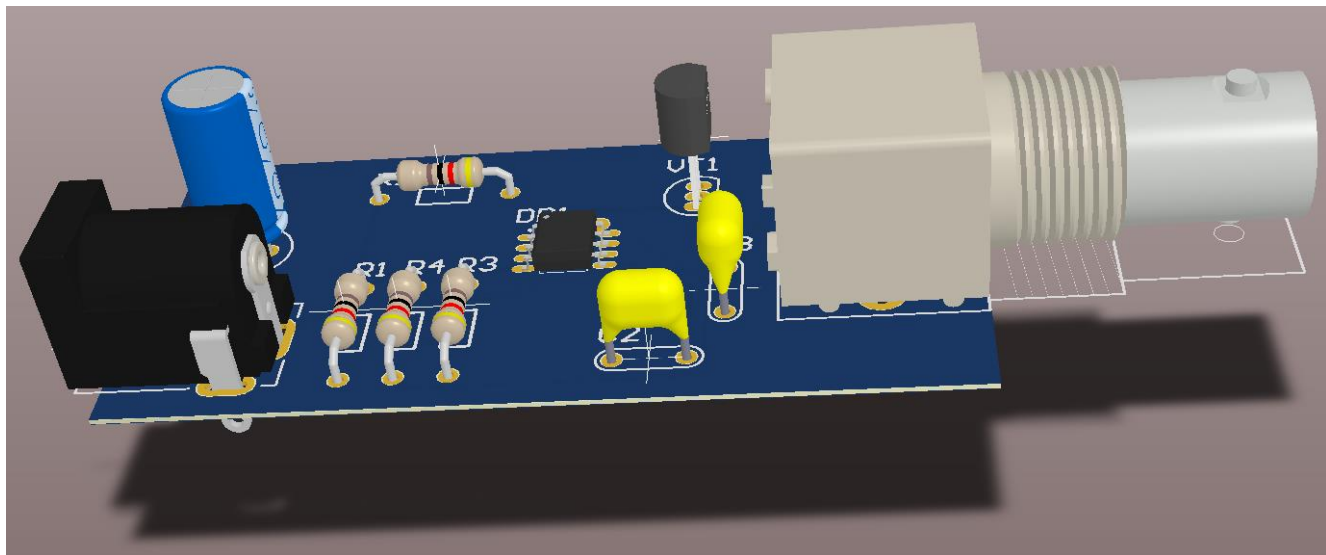


Рис. 11.46. Вигляд плати у 3D режимі

4. Питання для самоконтролю

1. Поясніть, для чого призначений генератор тактових імпульсів?
2. Поясніть, яким чином працює генератор тактових імпульсів на основі мікросхеми таймера 555?
3. Поясніть, для чого і яким чином використовуються схемні компоненти *GND Power Port* та *VCC Power Port*?
4. Яким чином виконується автоматична нумерація електронних компонентів на принциповій схемі в *Altium Designer*?
5. Поясніть призначення шару *Keep-Out Layer*.

6. Яким чином створити контур друкованої плати за допомогою майстра **PCB Board Wizard**?

7. Яким чином змінити колір примітивів на тому чи іншому шарі плати?

8. Поясніть, що таке правила проєктування та яким чином вони задаються?

9. Поясніть, яким чином налаштувати правило, за допомогою якого задають ширину доріжок на платі?

10. Поясніть, яким чином можна вирівняти групу об'єктів на платі?

11. Поясніть, яким чином виконати автоматичне трасування плати?

12. Що таке стратегія трасування?

13. Як виконати трасування лише на одній стороні плати?

14. Яким чином видалити результат попереднього трасування, якщо він вас не влаштовує, та почати трасування заново?

15. Яким чином перевірити плату на наявність помилок і порушень правил проєктування?

16. Поясніть, яким чином усунути порушення правила **Hole Size Constraint**?

17. Поясніть, яким чином усунути порушення правила **Minimum Solder Mask Sliver**?

18. Поясніть, яким чином усунути порушення правила **Silk To Solder Mask**?

19. Поясніть, яким чином усунути порушення правила **Clearance Constraint**?

20. Яким чином сформувати файл специфікації BoM?

21. Поясніть, яким чином можна замінити на принциповій схемі один компонент на інший?

22. Опишіть, як можна замінити одне посадкове місце мікросхеми на інше та виконати синхронізацію проєктних даних?

23. Поясніть, для чого використовується команда **Update From Libraries**?

ЗАВДАННЯ №1 ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ НА КОМП'ЮТЕРНИХ ПРАКТИКУМАХ

1. Теоретичні відомості

Кожен електронний компонент має певний набір параметрів, що вказується у **технічній документації**¹³⁵. Ця документація містить опис, параметри роботи, характеристики компонента, типові схеми увімкнення тощо. Документацію мають як прості компоненти (наприклад, резистор), так і складні (наприклад, мікроконтролери, ПЛІС тощо).

Для будь-якого розробника електронних схем та друкованих плат пристроїв вкрай важливою задачею є уміння виконувати аналіз технічної документації.

Із документації можна зрозуміти призначення електронного компоненту. У документації завжди вказується тип та геометричні характеристики корпусу; температурні режими роботи; електричні параметри (наприклад, значення мінімальних і максимальних напруг і струмів, які може витримати компонент, а також робочі діапазони цих величин); діаграма виводів (pinout) із поясненням їх призначення; функціональна діаграма електронного компонента та його внутрішня будова тощо. Важливою частиною документації є графіки, що пояснюють різноманітні режими роботи електронного компонента.

Важливо!

Неможливо розробити схему сучасного приладу, якщо не вміти користуватись технічною документацією на електронні компоненти.

У [Додатку К](#) наведено приклад аналізу технічної документації.

Технічну документацію у першу чергу можна знайти на сайті виробника того чи іншого електронного компонента. Також деякі продавці викладають технічну документацію на сторінці продажу електронного компоненту. Ще одним варіантом пошуку технічної документації є спеціалізовані сайти із базами даних, наприклад, <https://alldatasheet.com>.

¹³⁵ Англійською мовою така документація називається **datasheet**.

2. Завдання

1. Ознайомитись із технічною документацією¹³⁶ на електронний компонент, що відповідає вашому варіанту завдання (див. табл. CP1.1). Проаналізувати, яке його призначення та яку функцію він виконує. Зверніть увагу, що у деякій технічній документації описуються одразу декілька електронних компонентів. Вам необхідно використовувати тільки ту інформацію з технічної документації, яка стосується заданого у вашому варіанті електронного компонента.
2. На основі набутих під час виконання комп'ютерних практикумів знань та навичок спочатку в Редакторі символів намалювати УГП заданого електронного компоненту.
3. Далі у Редакторі посадкових місць розробити ПМ електронного компонента. Зверніть увагу, що один і той самий електронний компонент може випускатись у різних корпусах. Вам необхідно створити ПМ саме для того типу корпусу, який вказано у завданні.
4. Насамкінець об'єднати УГП та ПМ для створення завершеного бібліотечного компонентного модуля.

Таблиця CP1.1. Варіанти завдань для самостійної роботи №1

№ вар.	Назва мікросхеми	Тип корпусу	Виробник	Посилання на інтернет-сторінку із описом мікросхеми або безпосередньо на технічну документацію
1	74HC154	TSSOP24 (SOT355-1)	Nexperia	https://assets.nexperia.com/documents/data-sheet/74HC_HCT154.pdf
2	AD605	SOIC16	Analog Devices	https://www.analog.com/media/en/technical-documentation/data-sheets/ad605.pdf
3	ADS801	SOIC28 [DW (R-PDSO-G28)]	Texas Instruments	https://www.ti.com/product/ADS801
4	AD775	DIP24	Analog Devices	https://www.analog.com/media/en/technical-documentation/data-sheets/AD775.pdf

¹³⁶ Зверніть увагу, що у кожному варіанті є посилання на технічну документацію в інтернеті. Це зроблено лише для зручності пошуку необхідної інформації. Проте врахуйте, що із часом деякі посилання можуть припинити працювати внаслідок різних причин. Якщо так трапиться у вашому випадку, тоді самостійно виконайте пошук в інтернеті потрібної вам документації за назвою електронного компонента.

Продовження табл. СР1.1

№ вар.	Назва мікросхеми	Тип корпусу	Виробник	Посилання на інтернет-сторінку із описом мікросхеми або безпосередньо на технічну документацію
5	ADS800	SOIC28 [DW (R-PDSO-G28)]	Texas Instruments	https://www.ti.com/product/ADS800
6	CD4017B	TSSOP16 (PW0016A)	Texas Instruments	https://www.ti.com/product/CD4017B
7	CD4051B-Q1	TSSOP16 (PW0016A)	Texas Instruments	https://www.ti.com/product/CD4051B-Q1
8	CY7C4201V	TQFP32	Cypress Semiconductor	https://eu.mouser.com/datasheet/2/196/CYPR_S_A0001370355_1-3003265.pdf
9	ADG526A	SOIC28 (RW-28)	Analog Devices	https://www.analog.com/media/en/technical-documentation/data-sheets/ADG526A_527A.pdf
10	MAX1292	QSOP24	MAXIM	https://datasheets.maximintegrated.com/en/ds/MAX1290-MAX1292.pdf
11	ADC1005	DIP20 (J)	National Semiconductor	https://pdf1.alldatasheet.com/datasheet-pdf/view/8120/NSC/ADC1005.html
12	SN54LS138	CDIP16 [J (R-GDIP-T**)]	Texas Instruments	https://www.ti.com/product/SN54LS138
13	AD9280	SSOP28 (RS-28)	Analog Devices	https://www.analog.com/media/en/technical-documentation/data-sheets/AD9280.pdf
14	AD9050	SOIC28 (R-28)	Analog Devices	https://www.analog.com/media/en/technical-documentation/data-sheets/AD9050.pdf
15	THS1030	TSSOP28 [PW (R-PDSO-G28)]	Texas Instruments	https://www.ti.com/product/THS1030
16	TLC5540	SOP24 [NS (R-PDSO-G**)]	Texas Instruments	https://www.ti.com/product/TLC5540
17	AD876	SOIC28 (R-28)	Analog Devices	https://www.analog.com/media/en/technical-documentation/data-sheets/AD876.pdf
18	AD7226	PDIP20 (N-20)	Analog Devices	https://www.analog.com/media/en/technical-documentation/data-sheets/AD7226.pdf
19	SN54192	CDIP16 [J (R-CDIP-T**)]	Texas Instruments	https://pdf1.alldatasheet.com/datasheet-pdf/view/85760/TI/SN74LS192N.html
20	DM74LS161A	SOIC16 (M16A)	Fairchild Semiconductor	http://pdf.datasheetcatalog.com/datasheet/fairchild/DM74LS161A.pdf

Продовження табл. СР1.1

№ вар.	Назва мікросхеми	Тип корпусу	Виробник	Посилання на інтернет-сторінку із описом мікросхеми або безпосередньо на технічну документацію
21	MC14585B	SOIC16	ON Semiconductor	https://www.onsemi.com/pdf/datasheet/mc14585b-d.pdf
22	SN74LS247	DIP16	ON Semiconductor	https://pdf1.alldatasheet.com/datasheet-pdf/view/12638/ONSEMI/SN74LS247.html
23	CD4042BM	CDIP16 (J16A)	National Semiconductor	https://datasheet4u.com/datasheet-pdf/NationalSemiconductor/CD4042BM/pdf.php?id=109030
24	CD4027B	SOP16 (NS0016A)	Texas Instruments	https://www.ti.com/product/CD4027B
25	SN74HC164	SOP14 [NS (R-PDSO-G**)]	Texas Instruments	https://www.ti.com/product/SN74HC164
26	CD4043B	TSSOP16 [PW (R-PDSO-G**)]	Texas Instruments	https://www.ti.com/product/CD4043B
27	HCC4034B	Plastic DIP24 (0.25)	ST Microelectronics	https://pdf1.alldatasheet.com/datasheet-pdf/view/22328/STMICROELECTRONICS/HCC4034B.html
28	CD4029B	TSSOP (PW0016A)	Texas Instruments	https://www.ti.com/product/CD4029B
29	SN74LS153	DIP16 [N (R-PDIP-T**)]	Texas Instruments	https://www.ti.com/product/SN74LS153
30	SN74LS145	SOIC16 [D (R-PDSO-G16)]	Texas Instruments	https://www.ti.com/product/SN74LS145

Примітка. Зверніть увагу, що часто в технічній документації наведені геометричні характеристики того чи іншого корпусу мікросхеми, що містить іншу кількість виводів, ніж має бути. Так відбувається тому, що виробник наводить загальні геометричні характеристики корпусу на деякому прикладі. Задачею розробника є врахування загальних геометричних характеристик та створення на основі наведеного прикладу корпусу такого ПМ, яке відповідатиме конкретній мікросхемі із заданою кількістю виводів.

ЗАВДАННЯ №2 ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ НА КОМП'ЮТЕРНИХ ПРАКТИКУМАХ

1. Теоретичні відомості

Генератор тактових імпульсів (або генератор синхроімпульсів, англ. clock generator, pulse generator) призначений для формування серії прямокутних імпульсів, які використовуються для управління та синхронізації різних процесів у цифрових пристроях: годинниках, мікроконтролерах, сигнальних процесорах, телефонах, комп'ютерах тощо.

Сьогодні існує багато видів ГТІ. Проте в їх основі лежить один і той самий принцип реалізації – головними компонентами є резонансна схема та підсилювач сигналу.

Одними з найпростіших та поширених є генератори на логічних елементах. Їх принцип роботи ґрунтується на тому, що один або декілька підсилювальних каскадів (у ролі яких, власне, і виступають логічні елементи), охоплюються зворотним зв'язком, який містить електричні ланцюги, що забезпечують генерацію на необхідній частоті. Такими ланцюгами є RC-, LC-, RLC-ланцюги, а також п'єзокерамічні та кварцові резонатори. Частота генераторів із RC-, LC-, RLC-ланцюгами визначається співвідношенням номіналів резисторів, конденсаторів, індуктивностей, а от частота генераторів, що містять п'єзокерамічні або кварцові резонатори, визначається виключно частотою цих резонаторів. Діапазон частот генераторів тактових імпульсів на логічних елементах лежить від сотень герц до десятків і навіть сотень мегагерц.

Перевагою генераторів на логічних елементах є їх дешевизна, простота та швидкість створення. До недоліків можна віднести нестабільність частоти, але це стосується тільки тих генераторів, в яких використовуються RC-, LC- та RLC-ланцюги. Натомість генератори на логічних елементах із п'єзокерамічними або кварцовими резонаторами мають високу стабільність частоти, що обумовлена високою добротністю самого резонатора.

Із додатковою інформацією про різні схеми генераторів тактових імпульсів та їх реалізацією можете ознайомитись на сайті [cxem.net](http://cxem.net/beginner/beginner26.php), перейшовши за наступним посиланням <http://cxem.net/beginner/beginner26.php>.

2. Рекомендації та зауваження

Важливо пам'ятати, що на принципових схемах зображуються УГП окремих секцій (part) цифрових мікросхем, а не самі мікросхеми (хоча в деяких закордонних стандартах допустимо створювати УГП усієї цифрової мікросхеми). Зазвичай у використовуваних для виконання цієї самостійної роботи мікросхемах всередині міститься від 4 до 8 секцій. Позиційне позначення на схемі DD1.1 означає першу секцію першої мікросхеми, DD1.2 – другу секцію першої мікросхеми тощо. Якщо ж на схемі буде позиційне позначення DD2.1, то це означатиме, що або використані усі секції першої мікросхеми і тепер буде використовуватись друга, або у першій мікросхемі не використані усі секції, проте чомусь розробник за відомою тільки йому логікою вирішив усе одно використовувати дві мікросхеми.

Усе написане у попередньому абзаці необхідно враховувати у тому сенсі, що кількість корпусів електронних компонентів на платі може бути менше кількості зображених на принциповій схемі секцій компонентів, тому що в одному корпусі мікросхеми може міститись кілька секцій (наприклад, на схемі використано 3 секції з 4-х, однак на платі фізично буде розміщено тільки одну мікросхему).

Також виникає запитання, а що робити із вхідними та вихідними виводами тієї секції, яка не використовується? На схемі вона не вказується, але ж в мікросхемі, що монтується на плату, вона є і має відповідні виводи, які припадають до КМ.

Головне, що потрібно запам'ятати: входи як TTL, так і особливо CMOS мікросхем не повинні бути залишені непідключеними¹³⁷. Їх необхідно підключати

¹³⁷ Невикористовувані входи цифрової CMOS мікросхеми не можна залишати непідключеними через те, що вони мають дуже високий імпеданс, внаслідок чого є чутливими до зовнішніх завад (це можуть бути як паразитні завади самої плати, так і ефірні шуми). [Такі випадкові завади можуть призвести до того, що рівень напруги на вході буде «плавати» \(англ. float\) і зміститься в область між станом логічної одиниці та логічного нуля](#), яка відповідає порогу перемикавання. Якщо не вдаватись у подробиці, то схемотехніка окремої секції цифрової CMOS мікросхеми складається з кількох каскадів польових транзисторів, а [окремих вхід використовує MOSFET-транзистори в парах](#). Наявність на вході напруги, близької до порогу перемикавання, призводить до того, що пара транзисторів переходить у режим, при якому виникає прохідний (або наскрізний) струм. Внаслідок цього окрема логічна секція починає споживати неприпустимо великий струм. А якщо так відбувається досить довгий час, то ця секція перегрівається і, за найнесприятливішого сценарію, може обумовити теплове руйнування як окремої частини, так і усієї мікросхеми. Тому невикористовувані входи CMOS мікросхеми обов'язково повинні мати стабільний рівень нижче/вище за поріг перемикавання, що досягається їх підключенням або до живлення, або до землі.

або до ланцюгів землі, або ланцюгів живлення. Це можна робити або напряму, або через резистори. Якщо говорити про TTL мікросхеми, то спосіб підключення невикористовуваних входів може залежати від серії: для деяких серій є обов'язковим підключати через резистори до живлення або землі (до живлення – номіналом 1-10 кОм, до землі – 0,2-1 кОм¹³⁸), а для деяких серій – можна напряму. Якщо ж говорити про CMOS мікросхеми, то їх невикористовувані входи у переважній більшості випадків можна напряму без резисторів підключати до землі або живлення (хоча в мережі ви знайдете поради про необхідність підключення і через резистори¹³⁹). Напряму невикористовувані входи CMOS мікросхем можна підключати тому, що вони мають дуже високий імпеданс і додатковий резистор на вході ролі не відіграє.

Прочитати про способи підключення невикористовуваних входів мікросхем можна [тут](#) і [тут](#).

Окрім описаних вище загальних правил, які пояснюють, що робити із невикористовуваними виводами, необхідно уважно ознайомлюватись із технічною документацією на мікросхему, оскільки в ній можуть бути дані конкретні поради із цього приводу.

Якщо входні виводи правильно підтягнуті до землі, або до живлення, то відповідні їм невикористовувані вихідні виводи необхідно залишити ні до чого не підключеними (тобто для них допустимо «висіти у повітрі»).

Якщо ж говорити про TTL мікросхеми, то питання невикористовуваних входів стоїть не так гостро. Загалом, їх можна залишити непідключеними, оскільки внутрішня будова окремої секції TTL така, що сигнал від непідключеного входу сприймається як логічна одиниця. Але залишати вхід вільним все ж не рекомендується, оскільки внаслідок дії все тих же завад вхід буде плавати, призводячи до перемикання невикористовуваної секції. А оскільки вона все одно залишається підключеною до джерела живлення, то споживатиме, хоч і не значний, але зайвий струм, що погіршуватиме енергоефективність схеми загалом. Також постійні випадкові перемикання однієї секції можуть за несприятливих умов впливати на сусідні, сповільнюючи їх роботу або призводячи до помилкових спрацьовувань. Хоча такий розвиток подій малоімовірний.

¹³⁸ Насправді номінали у кожному конкретному випадку мають розраховуватись, зокрема за законом Ома.

¹³⁹ Використання підтягувальних резисторів пов'язане із тим, що деякі CMOS мікросхеми мають спеціальні режими, які можна використовувати для тестування. Ці режими можна перемикати за допомогою тестового обладнання, тільки якщо входи не підключені безпосередньо до живлення або землі. Іншою причиною використання підтягувальних резисторів у випадку CMOS мікросхеми може бути банальна звичка розробника, який раніше довго працював із мікросхемами TTL, де вони були потрібні.

3. Завдання

1. Ознайомитись із принциповою схемою генератора тактових імпульсів згідно заданого варіанту завдання.
2. Розробити УГП кожного електронного компонента, що використовується в схемі, та додати до власної бібліотеки (усі геометричні розміри УГП мають відповідати стандартам). Зверніть увагу, що більшість компонентів вже були створені на попередніх комп'ютерних практикумах. Створити потрібно тільки ті, яких немає у бібліотеці.
3. Якщо у схемі передбачене використання пасивних компонентів, то необхідно пам'ятати, що у ваших бібліотеках є резистори та конденсатори, що мають різні ПМ (як для монтажу в отвори, так і для поверхневого монтажу). У кожному варіанті вказано, в якому саме корпусі необхідно використовувати ці електронні компоненти (а це безпосередньо впливатиме на те, яке ПМ буде використане).
4. У кожному варіанті також вказано тип корпусу мікросхеми. ПМ потрібно розробляти конкретно під заданий тип корпусу (зокрема, корпуси DIP, SOIC або SOP¹⁴⁰). Це необхідно враховувати під час роботи із технічною документацією (datasheet) на мікросхему, оскільки в документації може бути вказано декілька різних видів ПМ. Розроблене ПМ необхідно зберегти у власну бібліотеку і об'єднати із УГП мікросхеми.
5. Створити новий проєкт, до якого додати аркуші принципової схеми та друкованої плати.
6. У редакторі Schematic намалювати задану принципову схему. Для створення схеми використовувати тільки компоненти із власної бібліотеки. Використовувані типи роз'ємів, мікросхеми та номінали електронних компонентів вказані у кожному варіанті.
7. У редакторі PCB Editor розробити друковану плату на основі створеної принципової схеми. Геометричні розміри плати підібрати таким чином, щоб усі електронні компоненти компактно розміщувались на ній. Виконати трасування

¹⁴⁰ SOIC в технічній документації часто називається Small Outline, SOP – Small Outline Package. Обидва ці корпуси призначені для поверхневого монтажу, мають однаковий крок виводів 1,27 мм і дуже схожі між собою візуально. Інколи їх вважають одним і тим самим корпусом. Проте це не зовсім так. Є деякі розбіжності між ними, а саме: корпус SOP ширший та дещо вищий за SOIC, також є невелика відмінність за довжиною виводів у місці їх контакту із платою.

плати (або у ручному режимі, або в автоматичному) та пошук помилок. Важливою вимогою є необхідність усунути усі наявні в проєкті помилки таким чином, щоб після запуску перевірки на відповідність заданим правилам не знаходилось жодної помилки.

8. У цій самостійній роботі пропонується виконати додаткове завдання, яке полягає у створенні та моделюванні роботи заданої електричної схеми ГТІ у програмах **Multisim** або **Proteus**. Після створення і моделювання схеми необхідно виконати автоматичне трасування друкованої плати за допомогою засобів, які надають ці програми (наприклад, в Multisim для трасування плати використовується модуль Ultiboard). Результати порівняти з тим, що було зроблено самостійно в **Altium Designer**. Хоча це завдання не є обов'язковим та не впливатиме на оцінку, проте його виконання значно підвищить рівень підготовки та розуміння принципів розробки та моделювання електричних схем.

Варіанти завдань до самостійної роботи №2

Варіант 1	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – CD4011BC (у корпусі PDIP).</p> <p>Конденсатор керамічний для монтажу в отвори C1 = 470 пФ.</p> <p>Резистори для монтажу в отвори R1 = 10 кОм, R2 = 15 кОм.</p> <p>Роз'єм живлення – DS-210, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 2	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – CD4030C (у корпусі SOP).</p> <p>Резистори для поверхневого монтажу $R1 = 100 \text{ кОм}$, $R2 = 20 \text{ кОм}$.</p> <p>Конденсатор керамічний для поверхневого монтажу $C1 = 660 \text{ пФ}$.</p> <p>Діод для монтажу в отвори VD1 – 1N4148.</p> <p>Роз'єм живлення – KF301-2P, роз'єм для виведення сигналу (на виході) – PLS-2.</p>

Варіант 3	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – CD4001BC (у корпусі PDIP).</p> <p>Конденсатор керамічний для монтажу в отвори $C1 = 10 \text{ мкФ}$.</p> <p>Резистори для монтажу в отвори $R1 = 100 \text{ кОм}$, $R2 = 5 \text{ кОм}$.</p> <p>Діод для монтажу в отвори VD1 – 1N4001.</p> <p>Роз'єм живлення – PLS-2, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 4	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – CD4011BC (у корпусі SOIC).</p> <p>Резистори для поверхневого монтажу $R1 = 200 \text{ кОм}$, $R2 = 1 \text{ МОм}$, $R3 = 10 \text{ кОм}$.</p> <p>Конденсатор керамічний для поверхневого монтажу $C1 = 1 \text{ мкФ}$.</p> <p>Діоди для монтажу в отвори VD1, VD2 – 1N4148.</p> <p>Роз'єм живлення – DS-210, роз'єм для виведення сигналу (на виході) – BNC.</p>

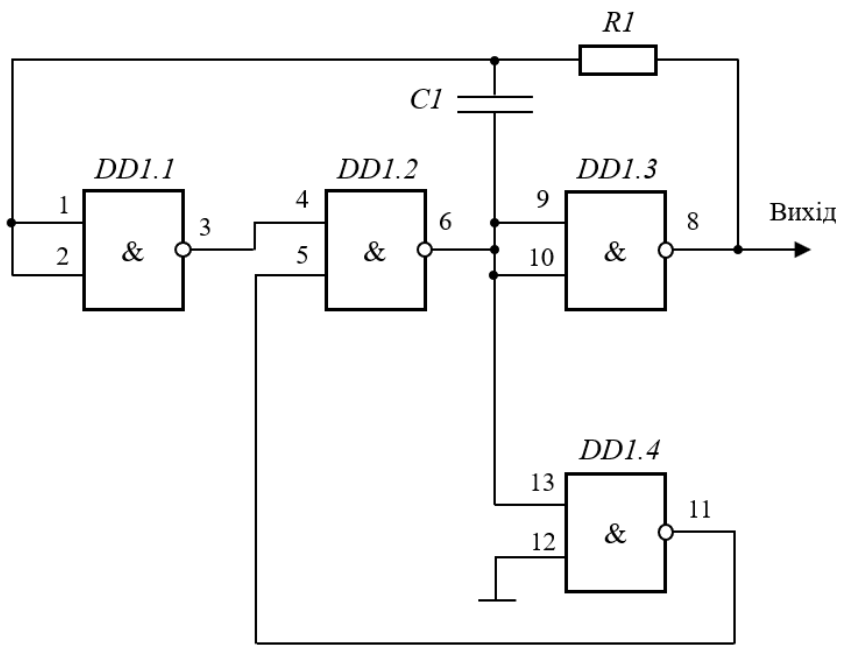
Варіант 5	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – CD4030B (у корпусі CDIP).</p> <p>Резистори для монтажу в отвори $R1 = R2 = 10 \text{ кОм}$.</p> <p>Конденсатори керамічні для монтажу в отвори $C1 = C2 = 330 \text{ пФ}$.</p> <p>Діоди для монтажу в отвори VD1, VD2 – 1N4001.</p> <p>Роз'єм живлення – KF301-2P, роз'єм для виведення сигналу (на виході) – BNC.</p>

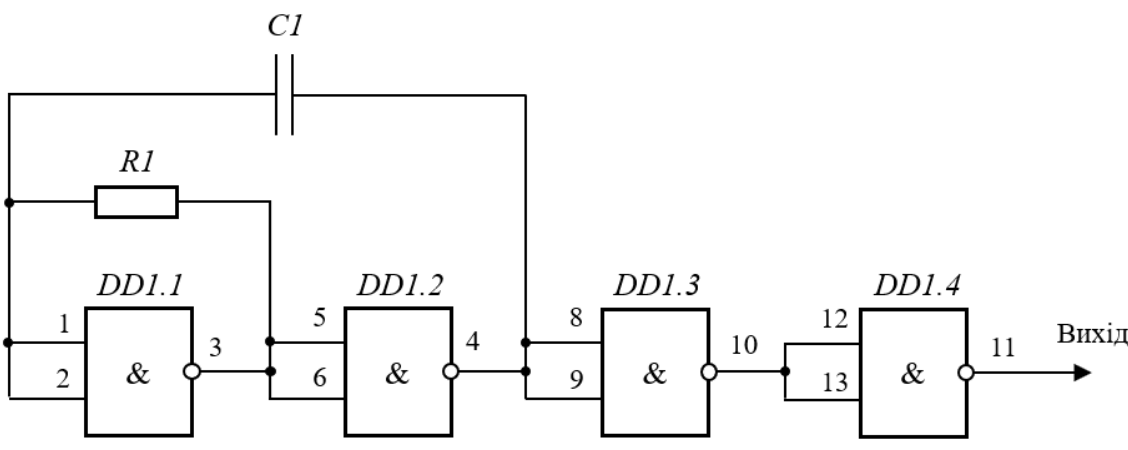
Варіант 6	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – CD4001BC (у корпусі SOP).</p> <p>Резистори для монтажу в отвори $R1 = R2 = 2,2 \text{ кОм}$.</p> <p>Конденсатори електrolітичні для монтажу в отвори $C1 = C2 = 40 \text{ мкФ}$.</p> <p>Роз'єм живлення – PLS-2, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 7	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – SN7400 (у корпусі SOP).</p> <p>Резистор для поверхневого монтажу $R1 = 1 \text{ кОм}$.</p> <p>Конденсатор керамічний для монтажу в отвори $C1 = 470 \text{ нФ}$.</p> <p>Роз'єм живлення – DS-210, роз'єм для виведення сигналу (на виході) – PLS-2.</p>

Варіант 8	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – SN7400 (у корпусі PDIP).</p> <p>Резистор для монтажу в отвори R1-R3 = 1 кОм.</p> <p>Конденсатор електrolітичний для монтажу в отвори C1 = 470 мкФ.</p> <p>Роз'єм живлення – KF301-2P, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 9	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – SN7400 (у корпусі DIP).</p> <p>Електrolітичні конденсатори для монтажу в отвори C1 = 500 мкФ, C2 = 1 мкФ.</p> <p>Резистори для монтажу в отвори R1 = R2 = 1 кОм.</p> <p>Роз'єм живлення – DS-210, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 10	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – SN7400 (у корпусі SOIC).</p> <p>Конденсатор керамічний для поверхневого монтажу $C1 = 900$ пФ.</p> <p>Резистор для поверхневого монтажу $R1 = 1,2$ кОм.</p> <p>Роз'єм живлення – PLS-2, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 11	
Схема	
Перелік електронних компонентів	<p>Мікросхема CD4011BC (у корпусі SOIC).</p> <p>Конденсатор керамічний для монтажу в отвори $C1 = 10$ мкФ.</p> <p>Резистор для монтажу в отвори $R1 = 10$ кОм.</p> <p>Роз'єм живлення – KF301-2P, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 12	
Схема	
Перелік електронних компонентів	<p>Мікросхема CD4011BC (у корпусі PDIP).</p> <p>Конденсатори керамічні для поверхневого монтажу $C1 = 0,22 \text{ мкФ}$, $C2 = 330 \text{ пФ}$.</p> <p>Резистори для монтажу в отвори $R1 = 360 \text{ кОм}$, $R2 = R3 = 2 \text{ МОм}$.</p> <p>Роз'єм живлення – PLS-2, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 13	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD – CD4001BC (у корпусі PDIP).</p> <p>Резистори для монтажу в отвори $R1 = R2 = 470 \text{ Ом}$.</p> <p>Конденсатор керамічний для поверхневого монтажу $C1 = 0,1 \text{ мкФ}$.</p> <p>Роз'єм живлення – DS-210, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 14	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – SN7400 (у корпусі PDIP).</p> <p>Резистори для монтажу в отвори $R1 = R2 = 4 \text{ кОм}$.</p> <p>Конденсатор керамічний для монтажу в отвори $C1 = 900 \text{ пФ}$.</p> <p>Роз'єм живлення – KF301-2P, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 15	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – SN7400 (у корпусі SOIC).</p> <p>Конденсатор керамічний для поверхневого монтажу $C1 = 1 \text{ нФ}$.</p> <p>Резистори для поверхневого монтажу $R1 = R2 = 510 \text{ Ом}$.</p> <p>Роз'єм живлення – DS-210, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 16	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – CD4049UB (у корпусі TSSOP).</p> <p>Конденсатор керамічний для поверхневого монтажу $C1 = 1 \text{ мкФ}$.</p> <p>Резистори для поверхневого монтажу $R1 = 50 \text{ кОм}$, $R2 = 10 \text{ кОм}$.</p> <p>Роз'єм живлення – KF301-2P, роз'єм для виведення сигналу (на виході) – BNC.</p>

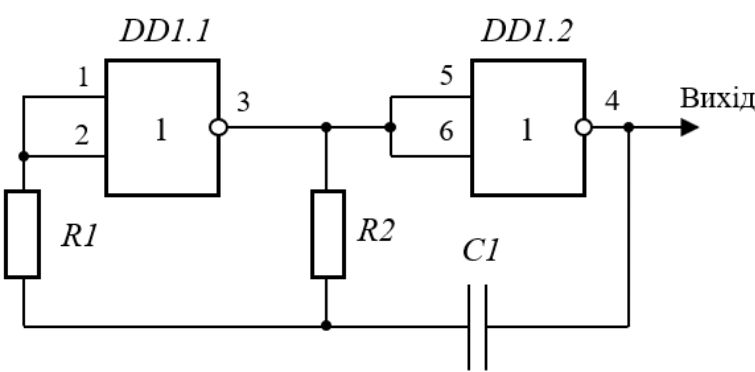
Варіант 17	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – CD4030B (у корпусі SOP).</p> <p>Конденсатор керамічний для монтажу в отвори $C1 = 100 \text{ нФ}$.</p> <p>Резистор для монтажу в отвори $R1 = 2 \text{ кОм}$.</p> <p>Роз'єм живлення – PLS-2, роз'єм для виведення сигналу (на виході) – BNC.</p>

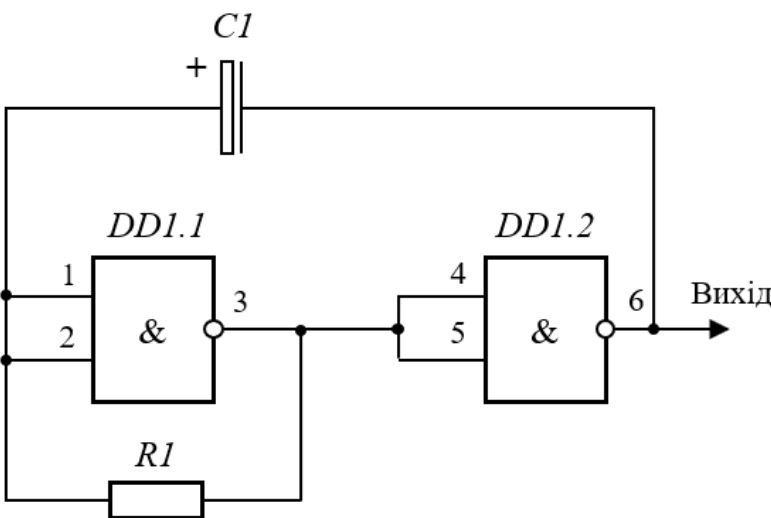
Варіант 18	
Схема	
Перелік електронних компонентів	<p>Мікросхема CD4049UB (у корпусі CDIP).</p> <p>Конденсатор керамічний для монтажу в отвори C1 = 2200 пФ.</p> <p>Резистор для монтажу в отвори R1 = 470 кОм.</p> <p>Роз'єм живлення – KF301-2P, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 19	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – CD4070B (у корпусі TSSOP).</p> <p>Резистори для поверхневого монтажу $R2 \geq 15 \text{ кОм}$, $R1 > 3 \cdot R2$.</p> <p>Конденсатор керамічний для поверхневого монтажу C1 = 500 пФ.</p> <p>Роз'єм живлення – DC-210, роз'єм для виведення сигналу (на виході) – PLS-2.</p>

Варіант 20	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – CD4070B (у корпусі CDIP).</p> <p>Резистори для монтажу в отвори R1 = 10 кОм, R2 = 80 кОм.</p> <p>Конденсатор керамічний для монтажу в отвори C1 = 100 пФ.</p> <p>Роз'єм живлення – KF301-2P, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 21	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – CD4093B (у корпусі SOP).</p> <p>Резистори для поверхневого монтажу R1 = 40 кОм, R2 = 20 кОм.</p> <p>Конденсатор керамічний для монтажу в отвори C1 = 8 нФ.</p> <p>Діоди для монтажу в отвори VD1, VD2 – 1N4148.</p> <p>Роз'єм живлення – DC-210, роз'єм для виведення сигналу (на виході) – PLS-2.</p>

Варіант 22	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – CD4001BC (у корпусі SOP).</p> <p>Конденсатор керамічний для поверхневого монтажу $C1 = 1$ мкФ.</p> <p>Резистори для поверхневого монтажу $R1 = 2$ кОм, $R2 = 50$ кОм.</p> <p>Роз'єм живлення – KF301-2P, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 23	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – SN7400 (у корпусі PDIP).</p> <p>Конденсатор електролітичний для монтажу в отвори $C1 = 50$ мкФ.</p> <p>Резистор для монтажу в отвори $R1 = 2$ кОм.</p> <p>Роз'єм живлення – PLS-2, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 24	
Схема	
Перелік електронних компонентів	<p>Мікросхема CD4049UB (у корпусі PDIP).</p> <p>Резистори для монтажу в отвори $R1 = 5 \text{ МОм}$, $R2 = 200 \text{ кОм}$.</p> <p>Конденсатори керамічні для монтажу в отвори $C1 = C2 = 22 \text{ пФ}$.</p> <p>Кварцовий резонатор $ZQ1 = 500 \text{ кГц}$ (у корпусі HC33U).</p> <p>Роз'єм живлення – KF301-2P, роз'єм для виведення сигналу (на виході) – PLS-2.</p>

Варіант 25	
Схема	
Перелік електронних компонентів	<p>Мікросхема CD4001BC (у корпусі PDIP).</p> <p>Резистор для монтажу в отвори $R1 = 62 \text{ кОм}$.</p> <p>Конденсатор керамічний для монтажу в отвори $C1 = 25 \text{ пФ}$.</p> <p>Кварцовий резонатор $ZQ1 = 200 \text{ кГц}$ (у корпусі HC33U).</p> <p>Роз'єм живлення – DC-210, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 26	
Схема	
Перелік електронних компонентів	<p>Мікросхема CD4001BC (у корпусі SOIC).</p> <p>Резистори для поверхневого монтажу $R1 = 22 \text{ МОм}$, $R2 = 560 \text{ кОм}$.</p> <p>Конденсатори керамічні для поверхневого монтажу $C1 = C2 = 15 \text{ пФ}$.</p> <p>Кварцовий резонатор $ZQ1=32768 \text{ Гц}$ (у корпусі TC26).</p> <p>Роз'єм живлення – KF301-2P, роз'єм для виведення сигналу (на виході) – PLS-2.</p>

Варіант 27	
Схема	
Перелік електронних компонентів	<p>Мікросхема CD4049UB (у корпусі SOP).</p> <p>Резистори для поверхневого монтажу $R1 = R2 = 560 \text{ Ом}$.</p> <p>Конденсатор керамічний для поверхневого монтажу $C1 = 1300 \text{ пФ}$.</p> <p>Кварцовий резонатор $ZQ1 = 3,2 \text{ МГц}$ (у корпусі HC49S).</p> <p>Роз'єм живлення – PLS-2, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 28	
Схема	
Перелік електронних компонентів	<p>Мікросхема DD1 – SN7400 (у корпусі DIP).</p> <p>Конденсатори керамічні для монтажу в отвори C1 = 5 нФ, C2 = 820 пФ.</p> <p>Резистор для монтажу в отвори R1 = 2,7 кОм.</p> <p>Котушка індуктивності для монтажу в отвори L1 = 3,5 мГн.</p> <p>Роз'єм живлення – DS-210, роз'єм для виведення сигналу (на виході) – BNC.</p>

Варіант 29	
Схема	
Перелік електронних компонентів	<p>Мікросхема CD4011BC (у корпусі PDIP).</p> <p>Резистори для монтажу в отвори R1 = R2 = 1,2 кОм.</p> <p>Конденсатори керамічні для поверхневого монтажу C1 = 10 нФ, C2 = 22 пФ.</p> <p>Кварцовий резонатор ZQ1 = 1,5 МГц (у корпусі HC33U).</p> <p>Роз'єм живлення – DC-210, роз'єм для виведення сигналу (на виході) – PLS-2.</p>

Варіант 30	
Схема	
Перелік електронних компонентів	<p>Мікросхема CD4011BC (у корпусі DIP).</p> <p>Конденсатор керамічний для монтажу в отвори C1 = 120 пФ.</p> <p>Котушка індуктивності для монтажу в отвори L1 = 220 мкГн.</p> <p>Роз'єм живлення – KF301-2P, роз'єм для виведення сигналу (на виході) – BNC.</p>

ЗАВДАННЯ №3 ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ НА КОМП'ЮТЕРНИХ ПРАКТИКУМАХ

1. Теоретичні відомості

САПР **Altium Designer** дозволяє розробляти проекти електронних приладів та систем будь-якої складності. У недосвідченого розробника виникає багато проблем на початках. Тому варто починати проектувати прості схеми, складність яких не перевищує того рівня, яким ви володієте, інакше ви даремно витратите час на проектування, виготовлення плати та матеріали, але результату не досягнете. Після набуття та удосконалення необхідних навичок вже можна переходити до більш складних проектів.

У цій самостійній роботі зібрані прості (але практичні) схеми, які не потребують значних знань для їх створення. Також ці схеми можна виготовити у домашніх умовах та використовувати для власних потреб. Наприклад, багато варіантів завдань містять схеми на основі операційних підсилювачів для керування потужністю звуку, що відтворюється за допомогою акустичних колонок. Також є декілька схем DC-DC перетворювачів, що можуть бути використані у проєктах розумного будинку, мобільних роботах, засобах автоматизації, автономних приладах неруйнівного контролю та технічної діагностики, системах автоматики тощо. Ще є схеми драйверів двигунів, які використовуються, зокрема, в робототехніці, системах автоматизації та управління.

Часто необхідні нам схеми вже кимось розроблені і доступні в інтернеті. Однак, щоб їх використати, потрібно навчитись в них розбиратись. У цьому допоможуть знання елементарних законів електротехніки (Ома, Кірхгофа), загальних принципів роботи напівпровідникових пристроїв, принципів побудови УГП, а також уміння «читати» схеми.

Також необхідно враховувати, що значна кількість простих схем, які доступні в інтернеті і які можна використовувати для власних потреб, не відповідають загальноприйнятим стандартам створення схем (особливо це може стосуватись використовуваних УГП мікросхем). Таке може відбуватися внаслідок різних причин, зокрема, тому що розробник використовував національні стандарти, які нам можуть бути невідомі, або він просто полінувався створювати схему за стандартами, а створив її нашвидкуруч, аби йому було зрозуміло.

Якщо ви вирішите розробити на основі такої схеми плату, то спочатку в **Altium Designer** у Редакторі схем вам доведеться намалювати обрану схему із усіма лініями з'єднань. Загалом, якщо схема проста і ви її створюєте лише для того, щоб розробити друковану плату на її основі, то можете вирішити використовувати ті самі графічні позначення електронних компонентів, що й автор схеми. Звичайно, вам доведеться спочатку створити відповідні бібліотечні компоненти в **Altium Designer**, орієнтуючись на зображення вихідної схеми. З іншого боку, ви можете розробити схему за стандартами, які ви знаєте. У такому випадку під час створення бібліотечних компонентів ви не будете орієнтуватись на вихідну схему, а будете орієнтуватись на вимоги стандартів щодо побудови УГП та принципових схем.

На рис. CP3.1 показана схема DC-DC перетворювача на основі мікросхеми LT1070. Цю схему можна знайти в [технічній документації на мікросхему LT1070](#). Можна помітити, що виводи на УГП розташовані по усіх сторонах прямокутника і не мають нумерації. Також не приховані виводи живлення і землі.

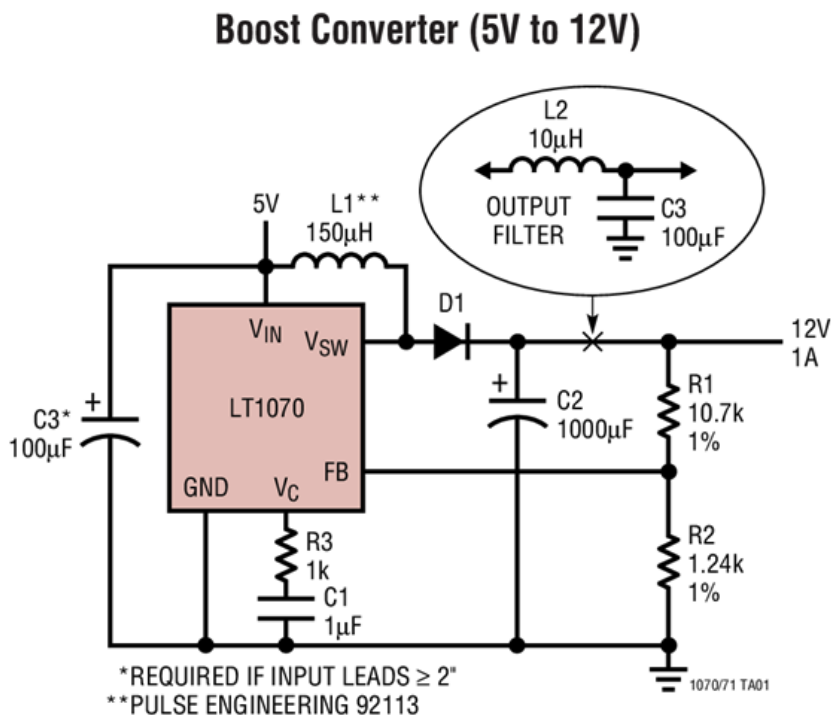
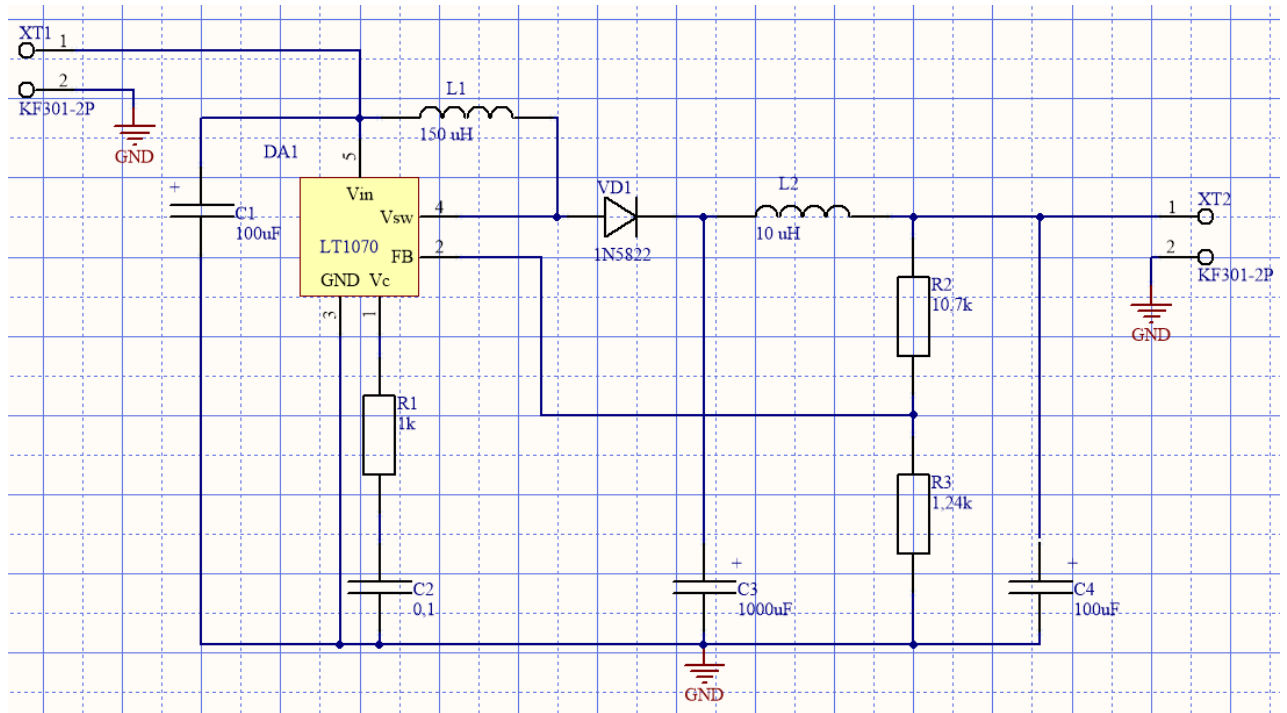


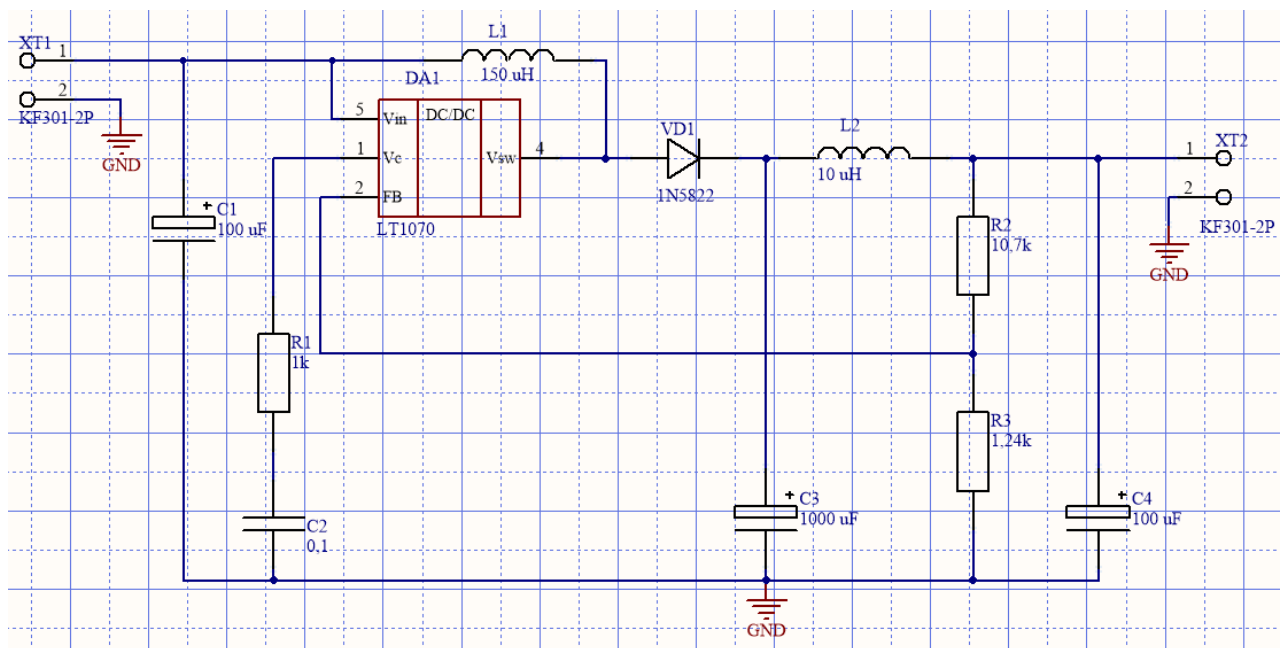
Рис. CP3.1. Схема DC-DC перетворювача

На рис. CP3.2 представлені варіанти реалізації в **Altium Designer** наведеної на рис. CP3.1 схеми DC-DC перетворювача.

Звичайно, можуть бути й інші варіанти побудови схеми в **Altium Designer**. Головне, щоб розроблена на їх основі плата електронного пристрою працювала і виконувала необхідні дії.



a



б

Рис. СР3.2. Реалізація схеми DC-DC перетворювача в **Altium Designer**: *a* – перший варіант, в якому УГП мікросхеми візуально повторює вихідне зображення, хоча інші символи такі, як були створені на комп'ютерних практикумах, *б* – другий варіант, в якому УГП усіх електронних компонентів виконано за стандартами, що використовувались під час виконання комп'ютерних практикумів

Зверніть увагу, що у цій самостійній роботі будуть використовуватись схеми, що є у вільному доступі в інтернеті. Причому вони будуть представлені у вигляді «як є». Коли будете створювати принципову схему в Редакторі схем, то можете на

власний розсуд вирішити, чи орієнтуватись на графічні позначення у вихідній схемі (тобто так, як вона представлена в інтернеті), чи все ж створити УГП на основі знань та навичок, набутих під час виконання комп'ютерних практикумів.

Насамкінець варто зазначити, що сьогодні вкрай важливим аспектом проєктування друкованих плат у САПР електронних пристроїв є уміння створювати цифровий двійник (digital twins) плати. Завдяки цьому вже на етапі проєктування можна розуміти, як виглядатиме плата, чітко визначити габарити, підібрати корпус, а у випадку, коли якийсь електронний компонент не вписується в конструкцію, замінити його аналогічним. Тому під час виконання цієї самостійної роботи буде необхідно розробити також і тривимірну модель плати.

Зауваження!

Під час роботи над цим завданням навчіться правильно підбирати електронні компоненти за їх характеристиками. Зокрема, резистор мало підібрати тільки за опором, ще необхідно враховувати розсіювану потужність (чим вона більша, тим зазвичай більший розмір самого резистора). Мікросхему мало просто підключити до джерела живлення, ще необхідно, щоб живлення було в заданих діапазонах. Електролітичний конденсатор мало просто встановити на платі, ще необхідно дотримуватись його полярності. Тому, наприклад, якщо згідно схеми потрібно встановити резистор з опором 4 Ом та потужністю 5 Вт, то використання у схемі резистора опором 4 Ом та потужністю 0,25 Вт вважатиметься помилкою виконання завдання. У реальному проєкті це призвело б до того, що резистор перегрівся, а прилад не працював (або вийшов з ладу).

2. Завдання

1. Ознайомитись із варіантом завдання.
2. Намалювати УГП кожного електронного компонента, що використовується в схемі, та додати до власної бібліотеки. Зверніть увагу, що багато УГП різних компонентів вже були створені на попередніх комп'ютерних практикумах. Розробити потрібно тільки ті, яких немає у бібліотеці.
3. Створити для нових електронних компонентів ПМ (тільки у випадку, якщо такі не були створені раніше на комп'ютерних практикумах) та зберегти їх у власну

бібліотеку. Зазвичай на попередніх комп'ютерних практикумах не були розроблені використовувані у цій роботі мікросхеми, тому необхідно самостійно в інтернеті знайти технічну документацію (datasheet) на них. На основі цієї технічної документації визначити тип корпусу мікросхеми та геометричні розміри ПМ.

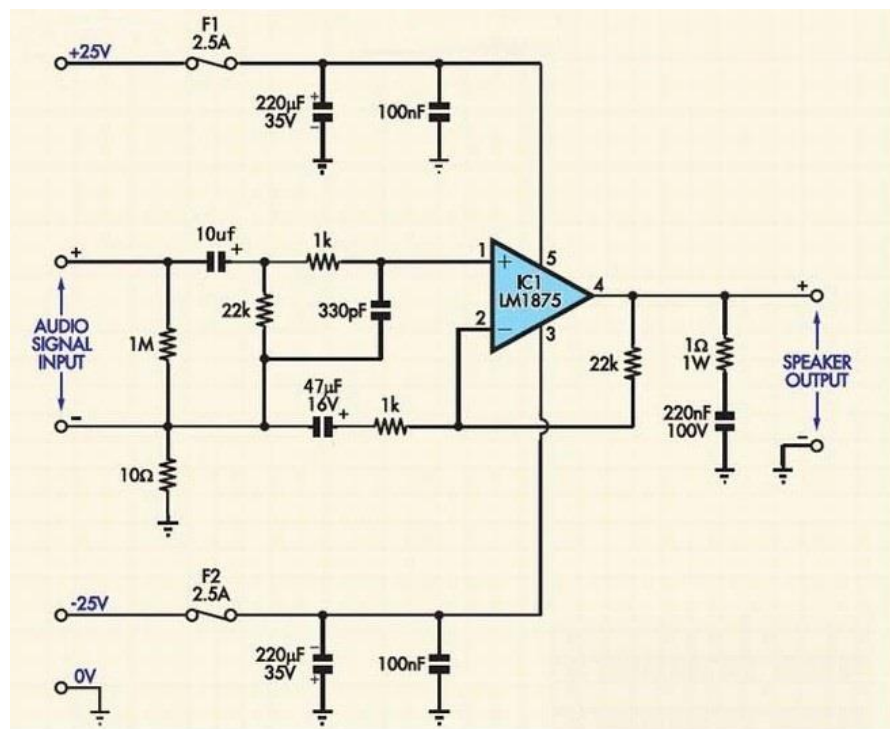
4. Об'єднати відповідні УГП та ПМ для створення завершених бібліотечних компонентних модулів. Для нових компонентів знайти або створити самостійно тривимірні моделі корпусів, після чого додати їх до відповідних ПМ.
5. Створити в **Altium Designer** принципову схему згідно отриманого варіанту завдання. Використовувати необхідно мікросхеми та номінали електронних компонентів, що вказані на принципових схемах. Зверніть увагу, що на самих принципових схемах роз'єми здебільшого не вказуються, проте ви можете зрозуміти, які роз'єми використовувати за фотографією самої плати. Якщо на фото плати немає роз'ємів, а є тільки отвори для припаювання провідників, то ви можете це так і залишити або обрати будь-який відповідний роз'єм для використання. Ще один нюанс, який необхідно враховувати: у деяких схемах на виході показано динамік. Зазвичай динамік повинен підключатись через роз'єми і він не є частиною плати, тож на схемі можете динамік не малювати, а завершити схему відповідним роз'ємом замість динаміка.
6. Розробити друковану плату на основі створеної принципової схеми. Виконати трасування плати (або у ручному режимі, або в автоматичному) та пошук помилок. Важливою вимогою є усунення усіх наявних помилок, щоб у результаті перевірки не знаходилось жодної помилки.
7. Фотографія плати використовується для прикладу¹⁴¹, щоб ви розуміли, яким може бути кінцевий результат. Це не означає, що ваша плата має бути один в один такою, як на фото, проте вона повинна бути якомога більше схожою на неї. Тобто, корпуси електронних компонентів бажано використовувати такі, які зображені на фото.
8. Виконати аналіз плати у 3D режимі та порівняти результат із фотографією плати, що була дана у завданні.

¹⁴¹ Більшість фотографій розміщена у відкритому доступі у мережі «Інтернет» та належать їх авторам. Вони використовуються у навчальному посібнику лише із ознайомчою метою.

Варіанти завдань до самостійної роботи №3

Варіант 1. Підсилювач (20 Ват)

Схема



Основна мікросхема – LM1875T

Зовн. вигляд плати



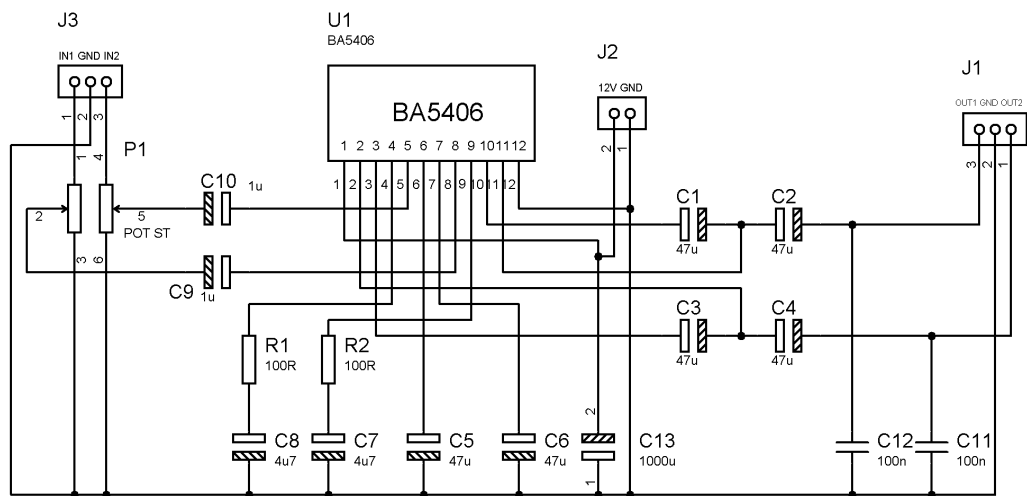
Посилання на
інф. ресурс

<https://aliradar.com/item/4000145289718-lm1875t-mono-kanal-stereo-audio-usilitel-platy-modul-diy-kit>

<https://www.youtube.com/watch?v=d9O7oZEYALI>

Варіант 2. Підсилювач

Схема



Основна мікросхема – BA5406

Зовн. вигляд плати

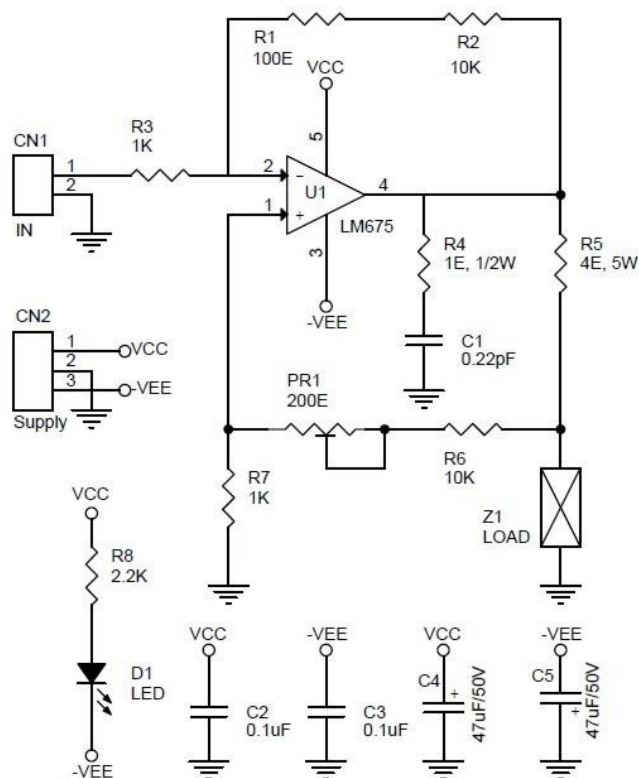


Посилання на
інф. ресурс

<https://www.youtube.com/watch?v=niN2ehCfXcY>

Варіант 3. Підсилювач потужності

Схема



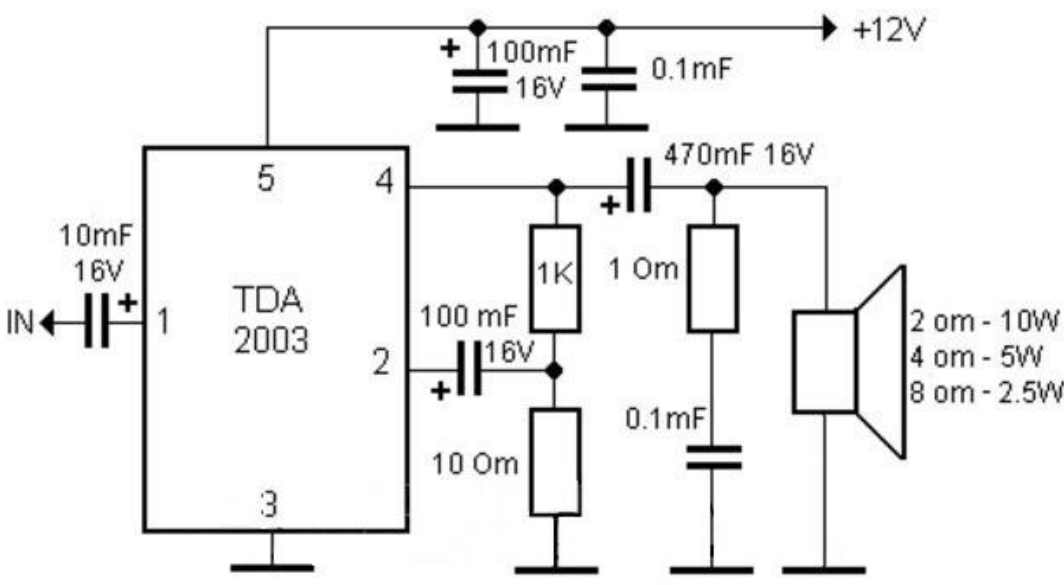
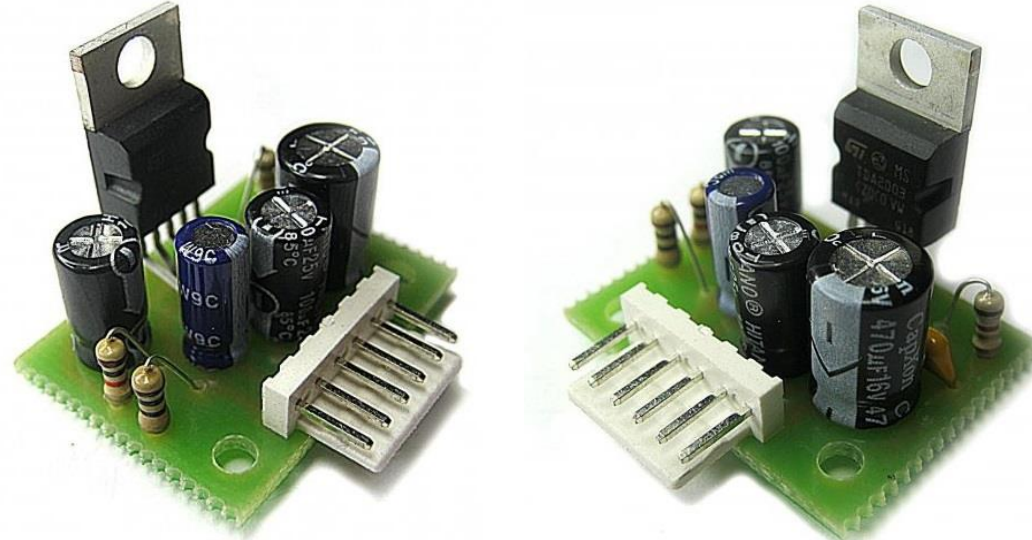
Основна мікросхема – LM675

Зовн. вигляд плати



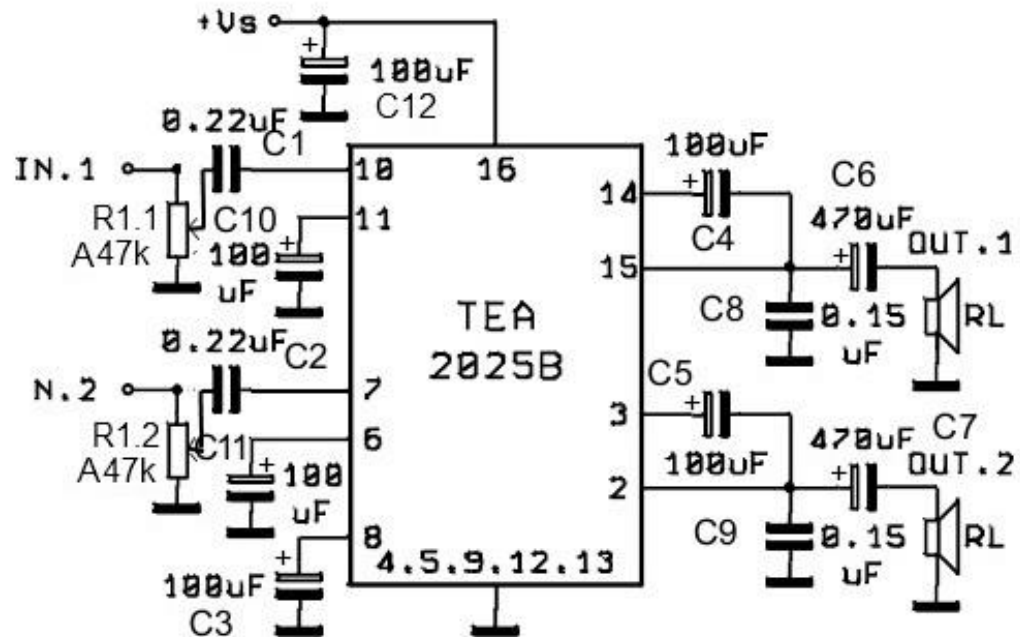
Посилання на
інф. ресурс

<https://www.twovolt.com/2019/07/17/3-amps-power-operational-amplifier-using-lm675/>

Варіант 4. Підсилювач	
Схема	 <p>Основна мікросхема – TDA2003</p>
Зовн. вигляд плати	
Посилання на інф. ресурс	<p> https://radio-kit.com/11-zvuk/10163-unch-1h-10vt-na-tda2003-k213-nabor https://radio-kit.com/image/catalog/pdf/K-213.pdf https://komponent-dnepr.com.ua/ua/p1647946527-radiokonstruktor-radiokit-k213.html </p>

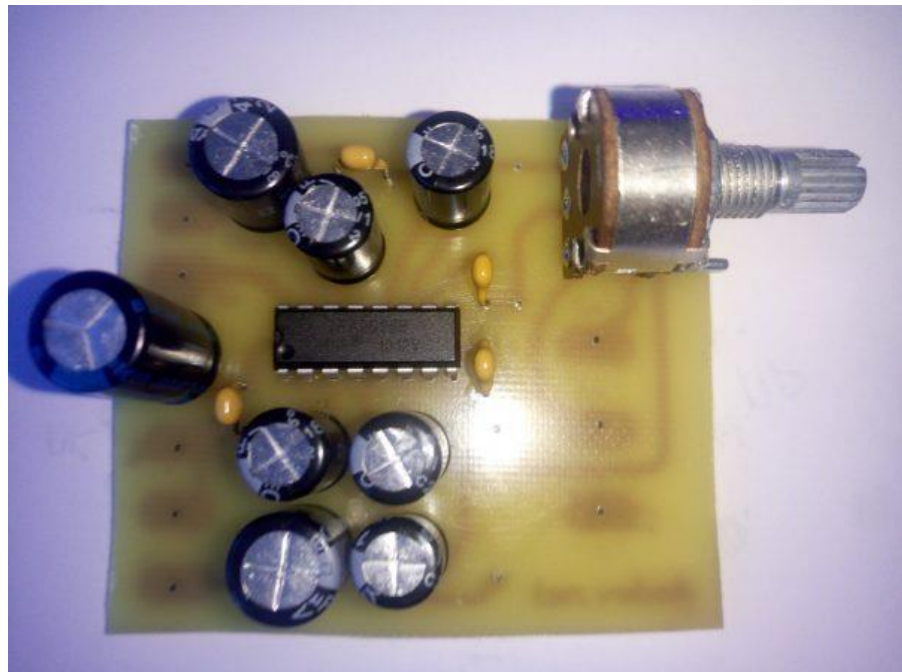
Варіант 5. Підсилювач

Схема



Основна мікросхема – TEA2025B

Зовн. вигляд плати

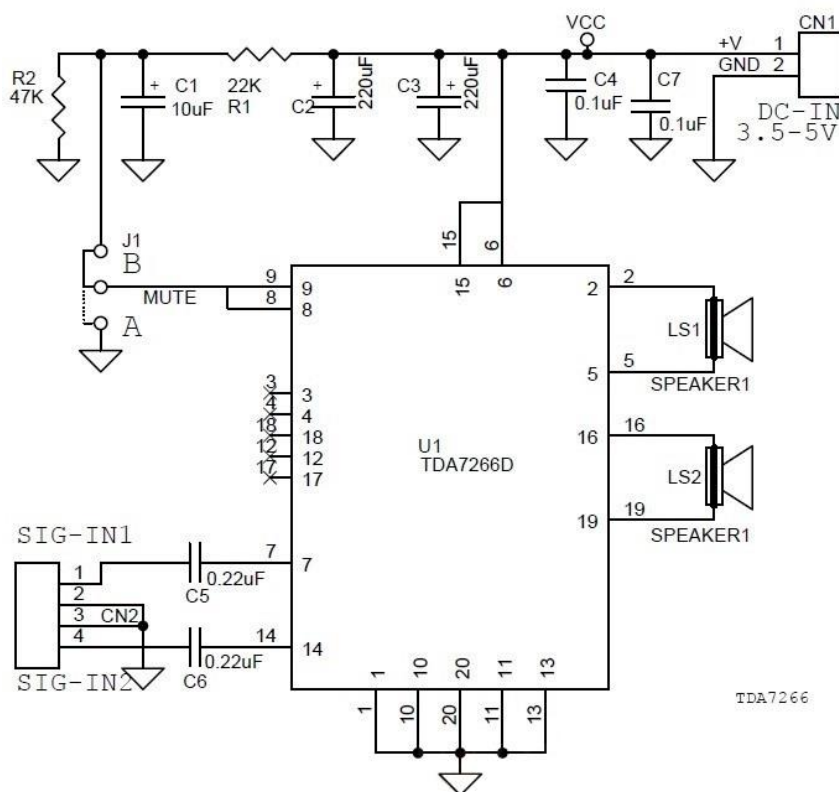


Посилання на
інф. ресурс

<https://diodov.net/lyuboj-usilitel-zvuka-svoimi-rukami/>

Варіант 6. Підсилювач

Схема



Основна мікросхема – TDA7266D

Зовн. вигляд плати

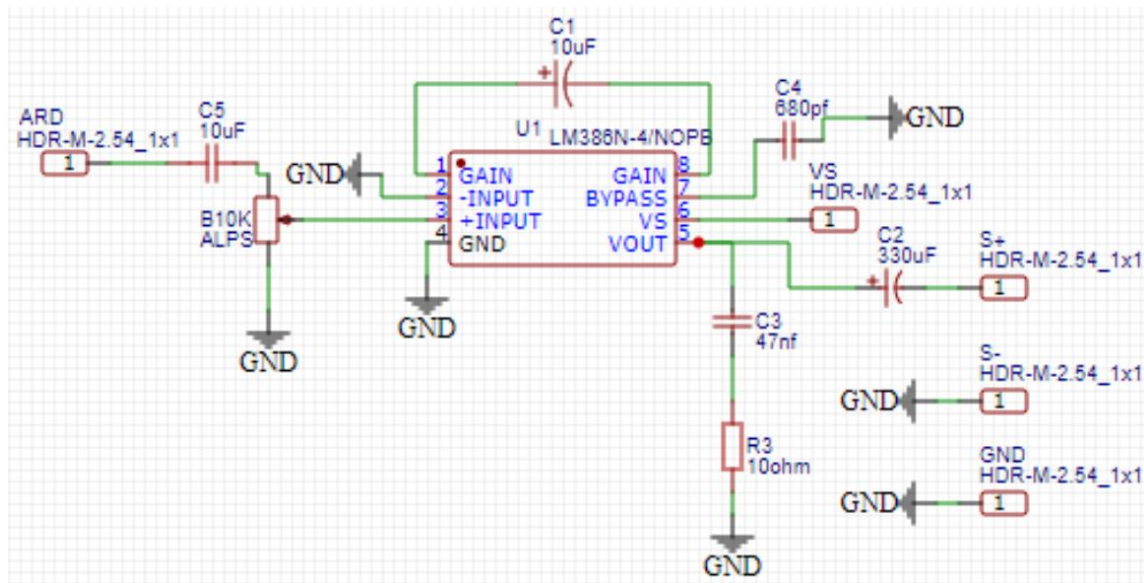


Посилання на
інф. ресурс

<https://www.electronics-lab.com/project/3w-stereo-audio-amplifier-using-tda7266d/>

Варіант 7. Підсилювач

Схема



Основна мікросхема – LM386

Зовн. вигляд плати

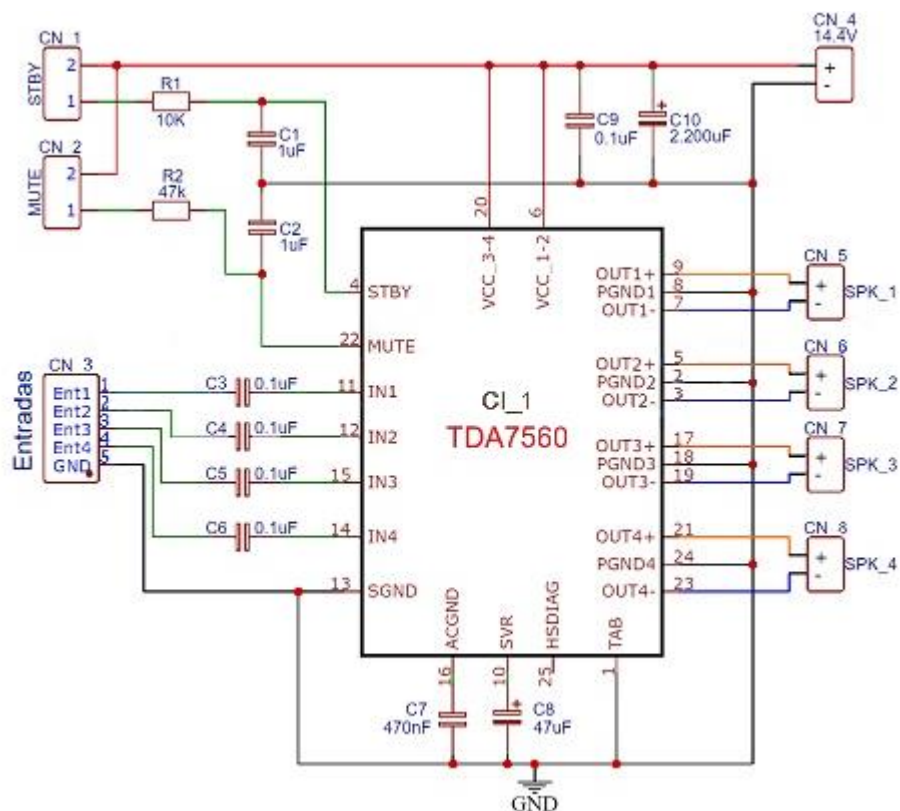


Посилання на
інф. ресурс

<https://www.instructables.com/Audio-Amplifier-With-the-LM386/>

Варіант 8. Підсилювач

Схема



Основна мікросхема – TDA7560

Зовн. вигляд плати

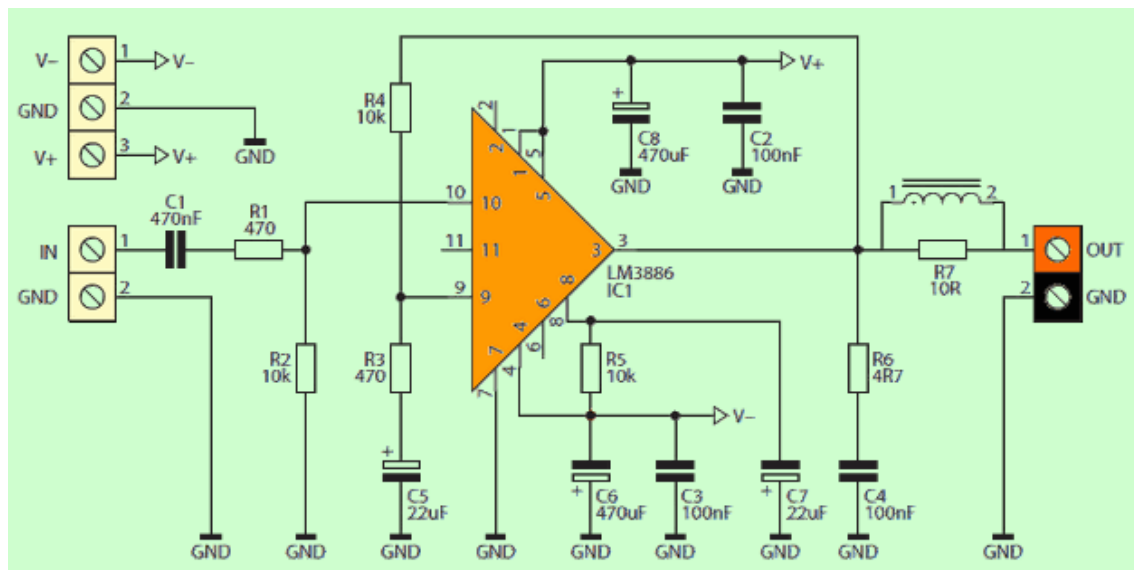


Посилання на
інф. ресурс

<https://www.elcircuits.com/2021/04/320w-power-audio-amplifier-powered-with.html>

Варіант 9. Підсилювач

Схема



Основна мікросхема – LM3886

Зовн. вигляд плати

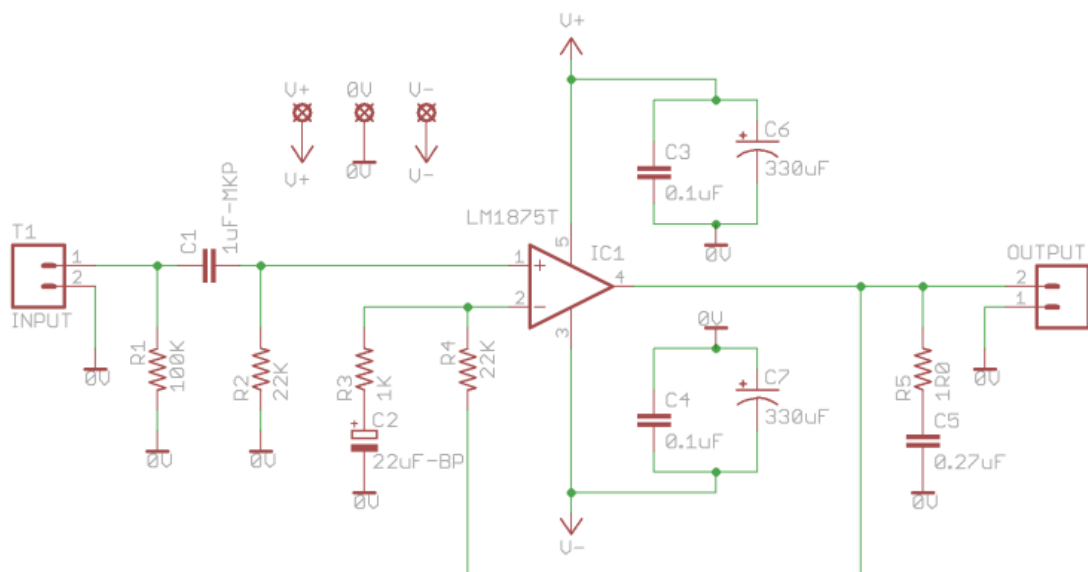


Посилання на
інф. ресурс

<https://320volt.com/en/lm3886-amplifier-module/>
<https://www.electronics-lab.com/project/50w-power-amplifier-with-lm3886/>

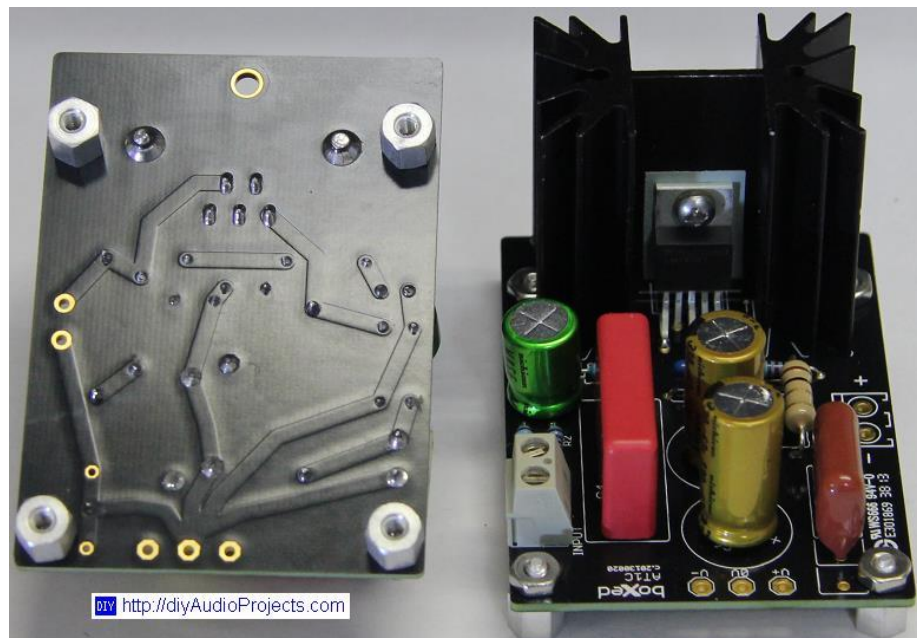
Варіант 10. Підсилювач

Схема



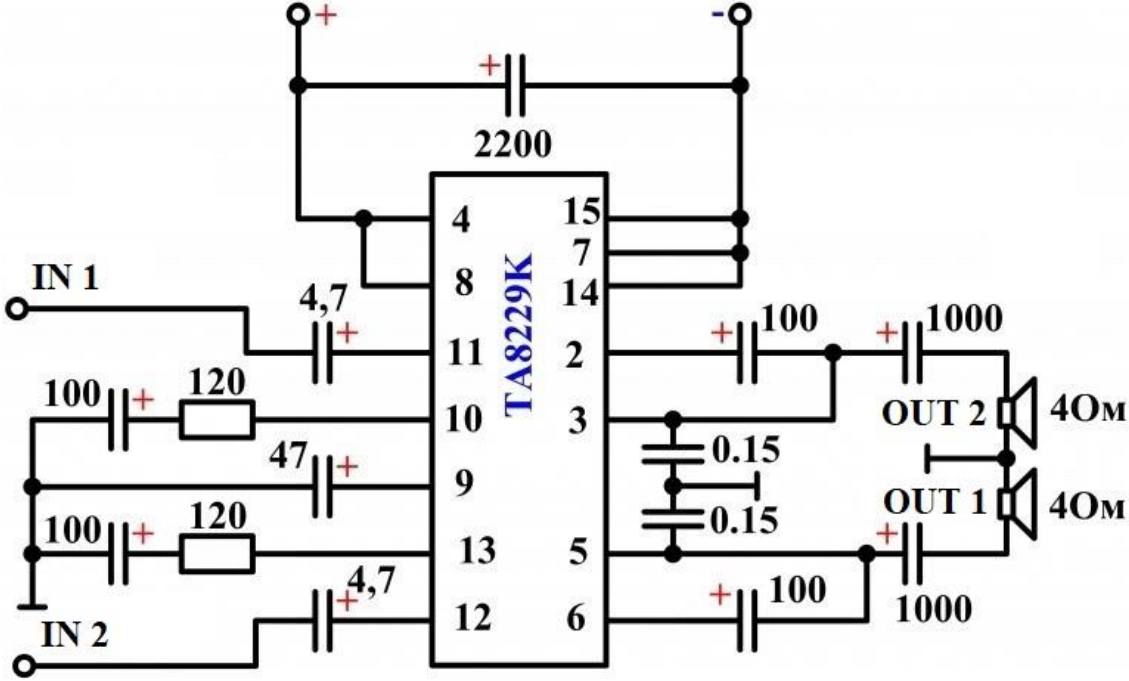

Основна мікросхема – LM1875

Зовн. вигляд плати



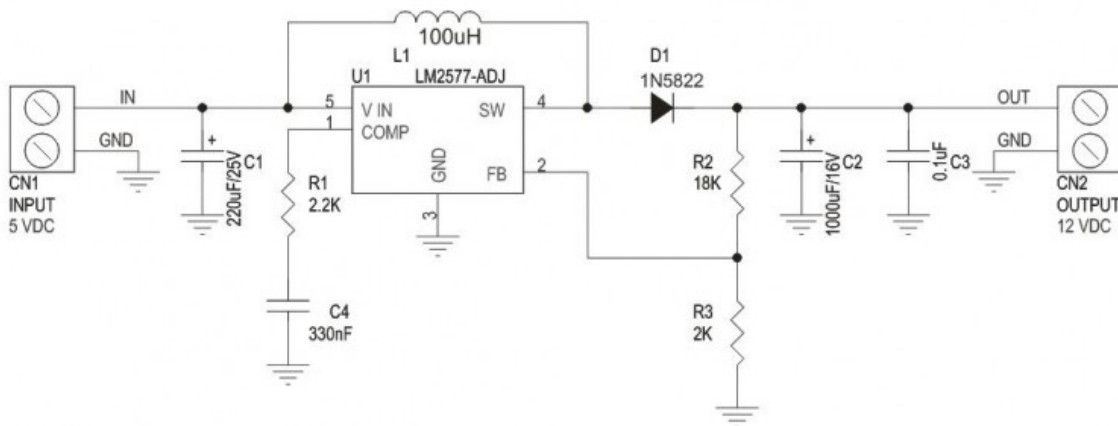
Посилання на
інф. ресурс

<https://diyaudioprojects.com/Chip/Gobo-LM1875-Amp-Kit/>

Варіант 11. Підсилювач	
Схема	 <p>Основна мікросхема – TA8229K</p>
Зовн. вигляд плати	
Посилання на інф. ресурс	<p>https://diodov.net/stereo-usilitel-zvuka-na-ta8229k/</p>

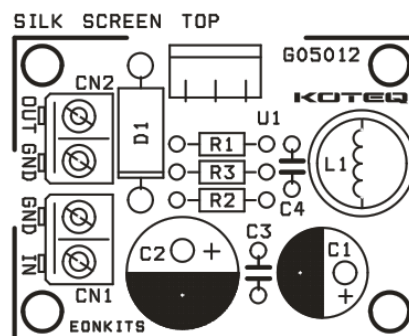
Варіант 12. Модуль DC-DC

Схема



Основна мікросхема – LM2577

Зовн. вигляд плати

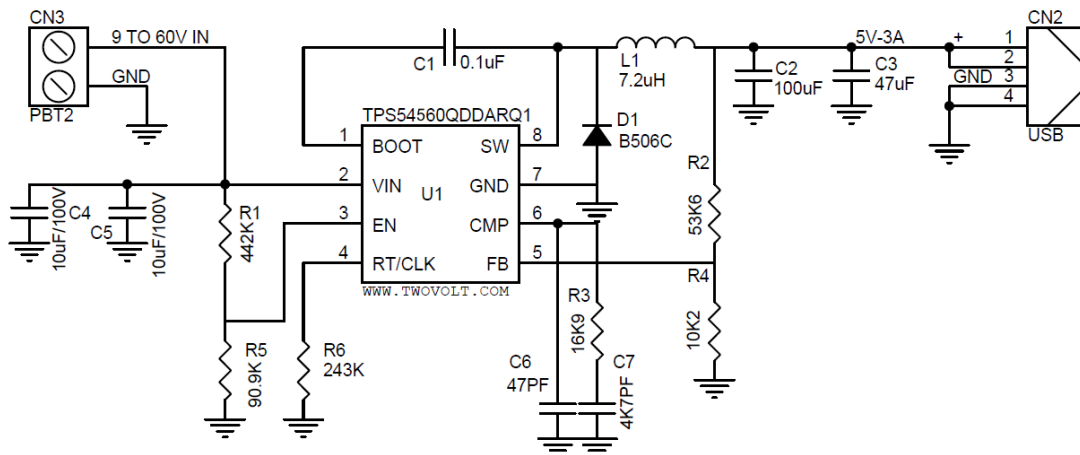


Посилання на
інф. ресурс

<https://www.twovolt.com/2016/08/23/5v-to-12v-step-up-dc-dc-converter-using-lm2577/>
<https://www.eleccircuit.com/lm2577-5v-to-12v-dc-converter-step-up-voltage-regulator/>

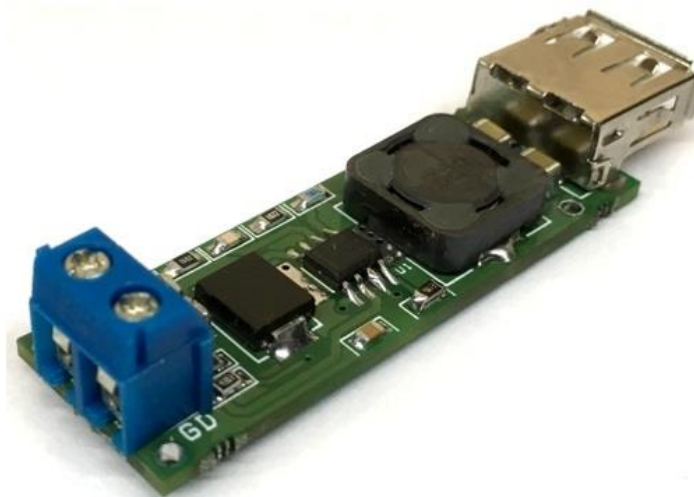
Варіант 13. Модуль DC-DC

Схема



Основна мікросхема – TPS54560

Зовн. вигляд плати

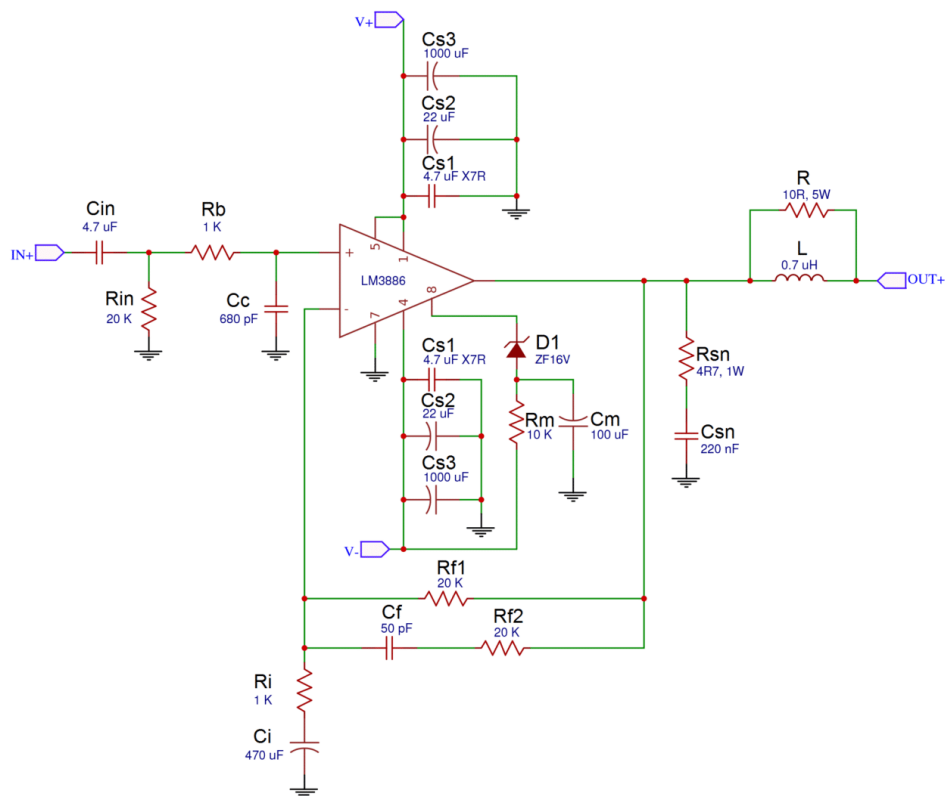


Посилання на
інф. ресурс

<https://www.twovolt.com/2018/12/08/60v-to-5v-2amps-usb-power-output-dc-dc-converter-for-e-vehicle/>

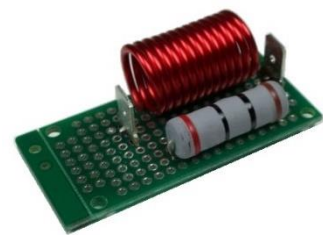
Варіант 14. Підсилювач

Схема



Основна мікросхема – LM3886

Зовн. вигляд плати



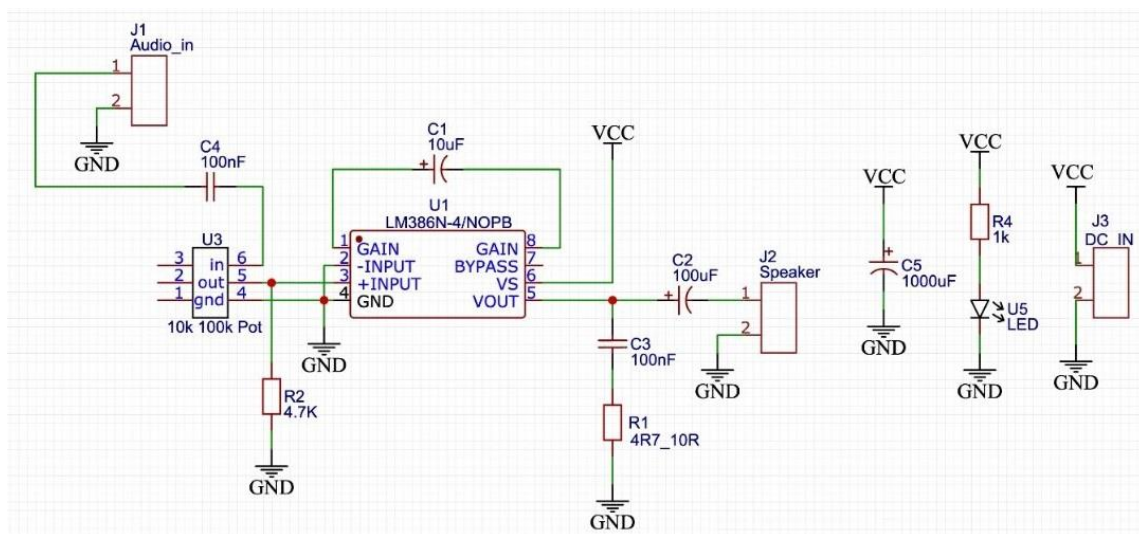
Зверніть увагу, що у цьому проєкті є одна схема, проте дві плати. Перша плата – основна, на ній розміщені практично усі елементи схеми. На другій платі розміщені тільки індуктивність та резистор, які у схемі знаходяться на виході та включені паралельно. Між собою плати з'єднуються проводами

Посилання на
інф. ресурс

<https://www.circuitbasics.com/design-hi-fi-audio-amplifier-lm3886/>

Варіант 15. Підсилювач

Схема



Основна мікросхема – LM386

Зовн. вигляд плати

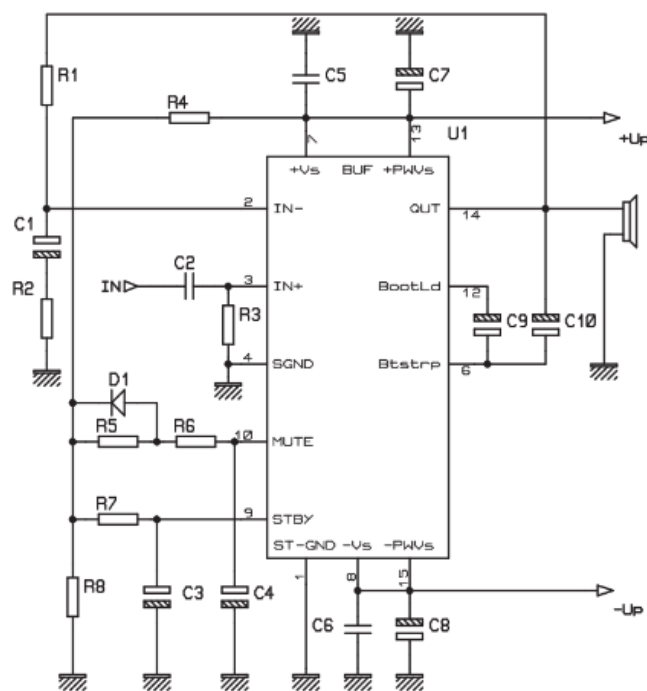


Посилання на
інф. ресурс

<https://circuitbest.com/audio-amplifier-circuit-with-lm386/>

Варіант 16. Підсилювач

Схема



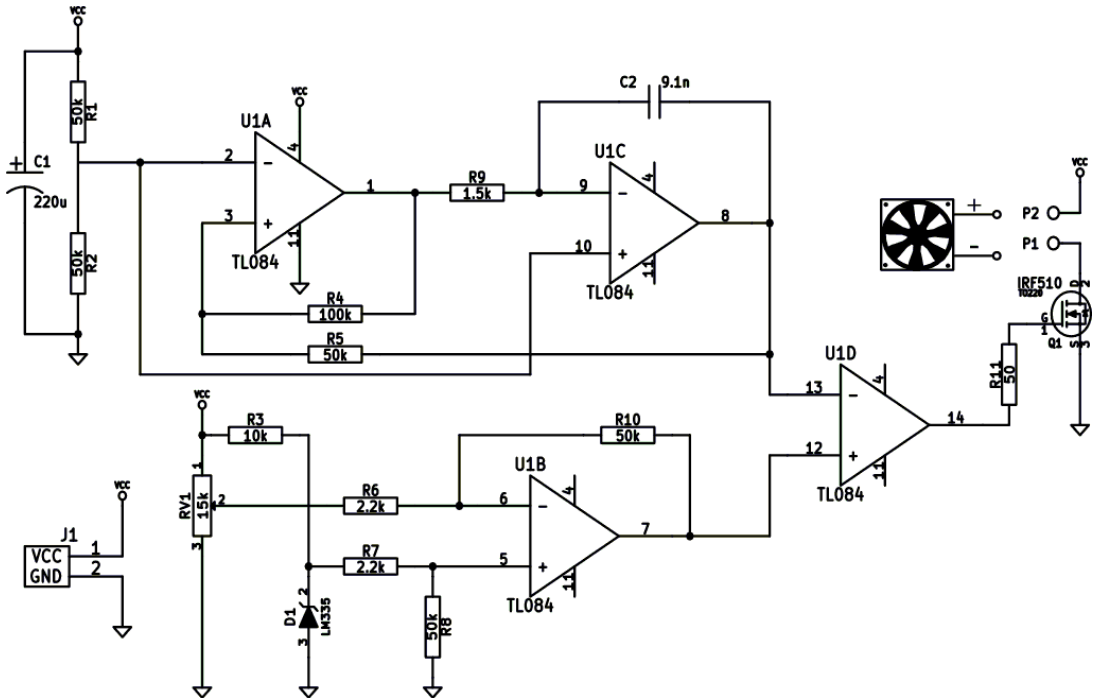

R1	18...24K
R2	680...910 Ом
R3	100...160K
R4	47...68K
R5	18...24K
R6	10...12K
R7	18...24K
R8	18...24K
U1	TDA7294, TDA 7293
D1	1N4148, 1N4448
C1	22...33 мкФ
C2	0,47 мкФ
C3	10...22 мкФ
C4	10...22 мкФ
C5	0,1...0,33 мкФ
C6	0,1...0,33 мкФ
C7	1000 мкФ 50В
C8	1000 мкФ 50В
C9	22...33 мкФ
C10	22...33 мкФ

Основна мікросхема – TDA7294. Наведена вище схема є загальною для TDA7294 та TDA7293. У випадку використання TDA7294 конденсатор C9 не використовується.

Зовн. вигляд плати

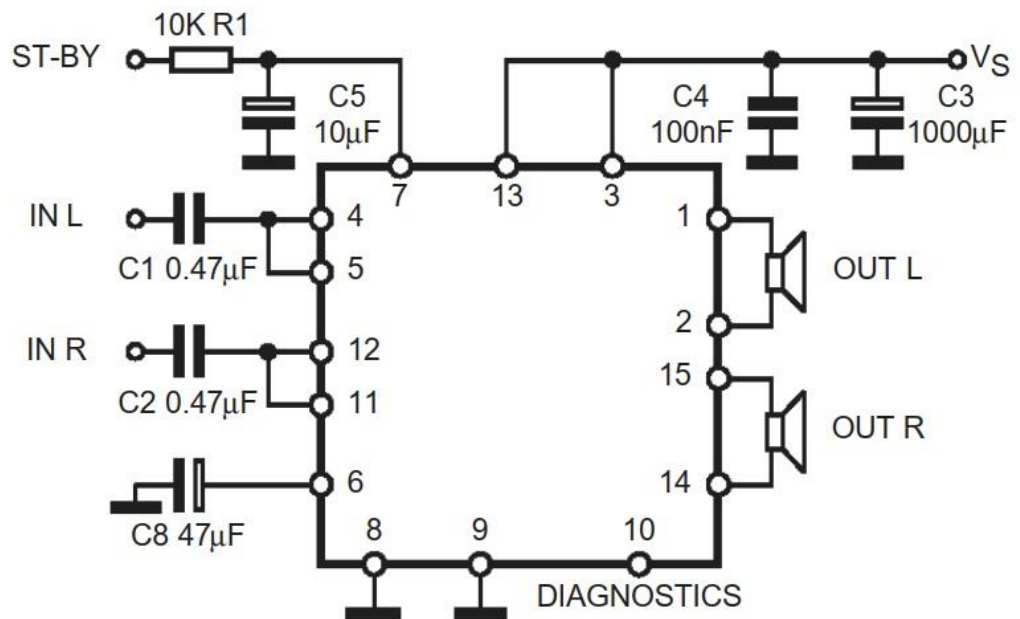
Посилання на
інф. ресурс

<https://radio-kit.com/image/catalog/pdf/K-207.pdf>
<https://www.youtube.com/watch?v=914tY0sjttk>

Варіант 17. Автоматичний регулятор швидкості вентиляторів ПК	
Схема	 <p>Основна мікросхема – TL084</p>
Зовн. вигляд плати	
Посилання на інф. ресурс	<p>https://electronoff.ua/ua/good/radiokonstruktor-k261-avtomaticheskij-regulyator-skorosti-ventilyatorov-pk.php</p> <p>https://radio-kit.com/image/catalog/pdf/K-261.pdf</p>

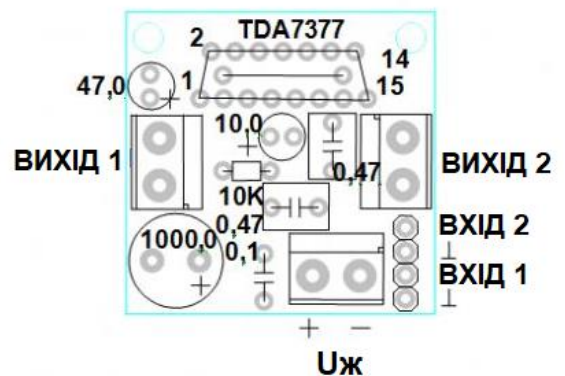
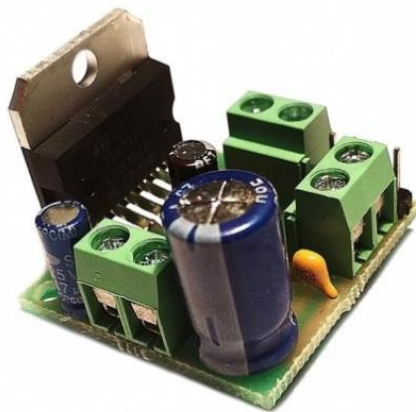
Варіант 18. Підсилювач

Схема



Основна мікросхема – TDA7377

Зовн. вигляд плати

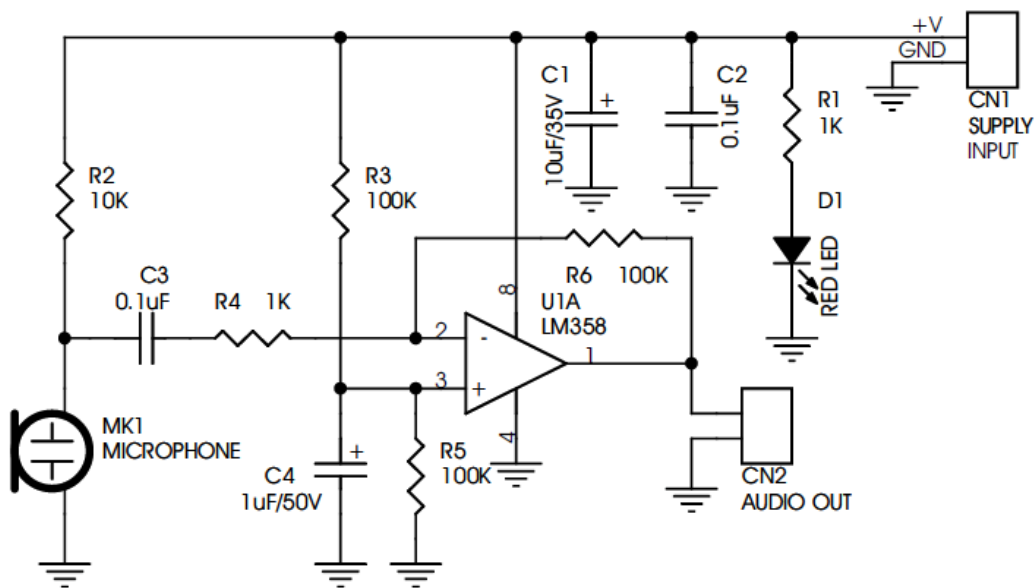


Посилання на
інф. ресурс

<https://datasheetspdf.com/pdf/378879/STMicroelectronics/TDA7377/1> (y
тех. документації дивіться Figure 2: Double Bridge)

Варіант 19. Підсилювач

Схема



Основна мікросхема – LM358

Зовн. вигляд плати

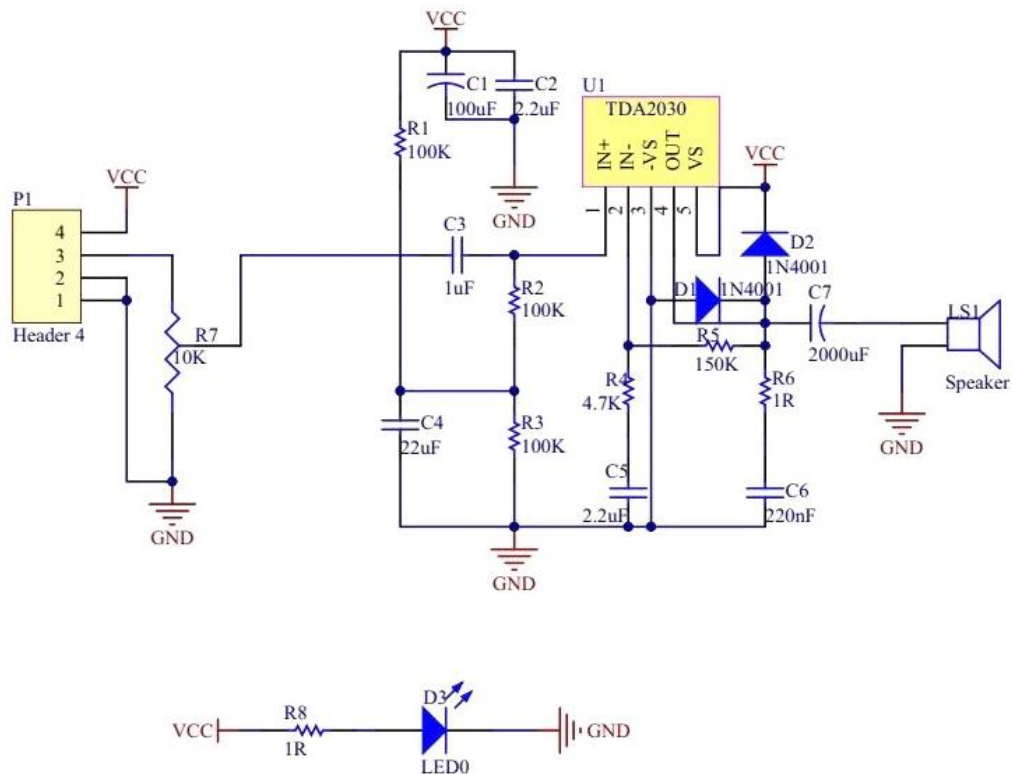


Посилання на
інф. ресурс

<https://www.twovolt.com/2018/04/23/mini-microphone-pre-amplifier-using-lm358/>

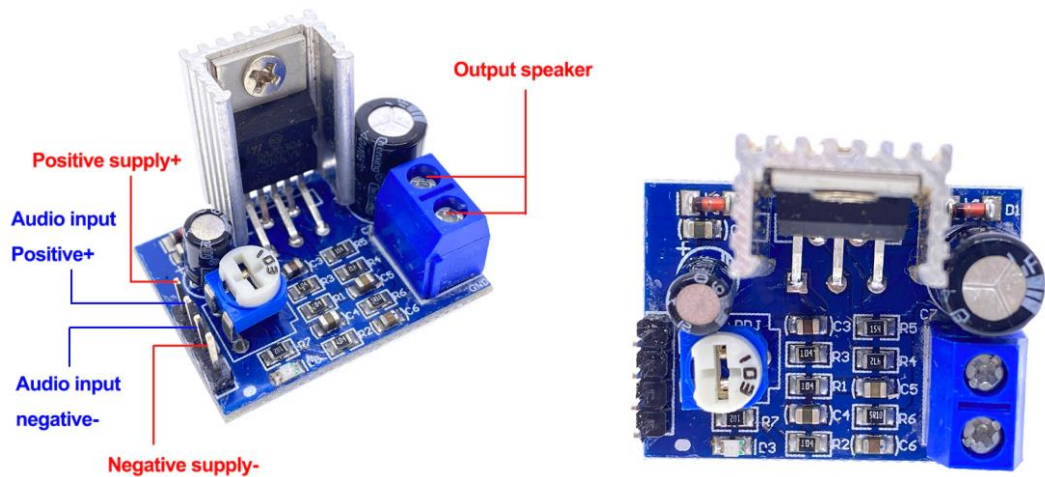
Варіант 20. Підсилювач

Схема



Основна мікросхема – TDA2030A

Зовн. вигляд плати



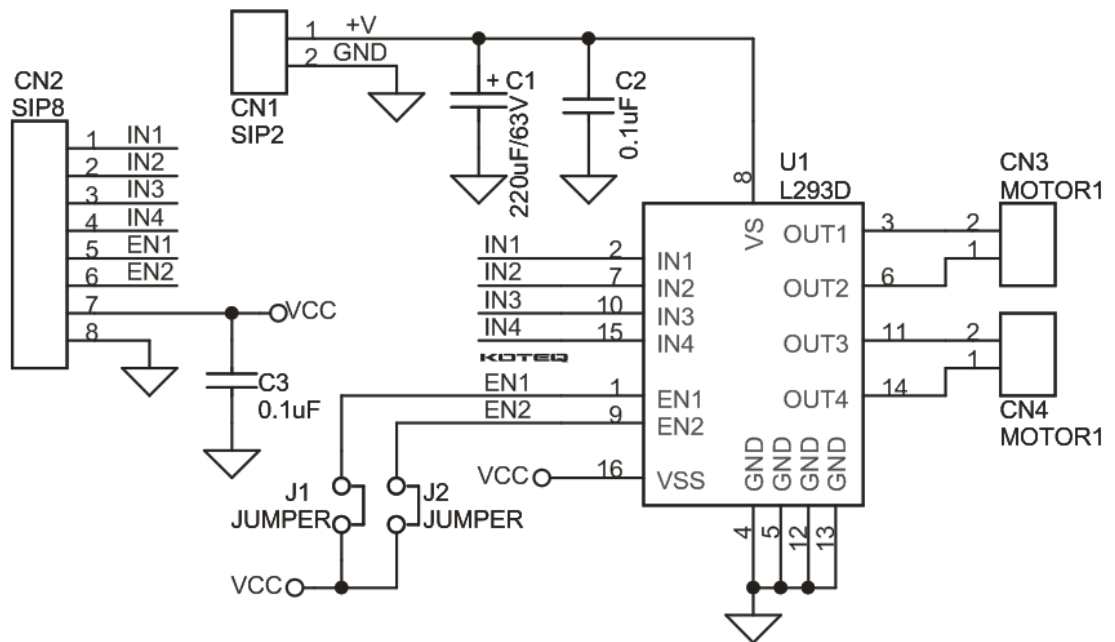
Зверніть увагу, що на схемі потенціометр позначено як R7, а обмежувальний резистор для світлодіода – як R8. Тоді як на платі потенціометр позначено як RES, а обмежувальний резистор – як R7.

Посилання на
інф. ресурс

<https://id.aliexpress.com/item/32513891467.html>
<https://id.aliexpress.com/item/32776443651.html>
<https://www.utmel.com/components/tda2030-audio-amplifier-datasheet-pinout-and-equivalents?id=562>

Варіант 21. Модуль драйвера двигунів L293D

Схема



Основна мікросхема – L293D

Зовн. вигляд плати

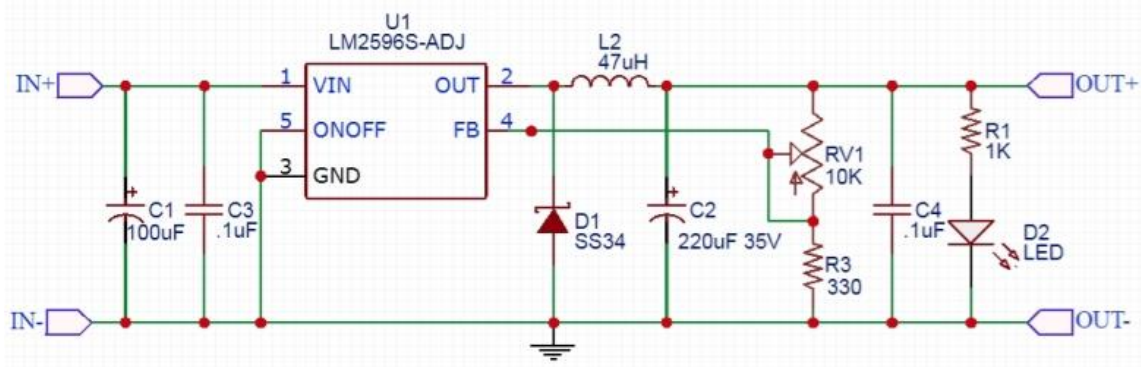


Посилання на
інф. ресурс

<https://www.electronics-lab.com/project/1293d-dc-motor-driver-module/>

Варіант 22. Модуль DC-DC Step-Down Converter

Схема



Основна мікросхема – LM2596S

Зовн. вигляд плати



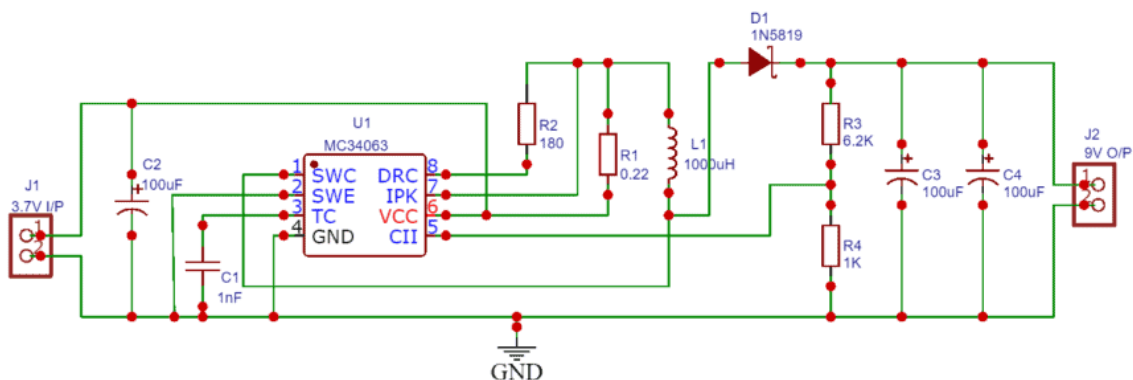
Зверніть увагу, що зазначений модуль виготовляється в Китаї та має кілька варіантів схем і плат, що зовні дуже схожі, але мають певні невеликі відмінності. Знайти його варіації ви можете на AliExpress.

Посилання на
інф. ресурс

<https://www.codrey.com/electronic-circuits/constant-current-cc-adapter-for-a-constant-voltage-cv-stepdown-voltage-regulator/>
<https://www.youtube.com/watch?v=9vMNa-H PRg>
<https://www.instructables.com/How-to-Use-DC-to-DC-Buck-Converter-LM2596/>

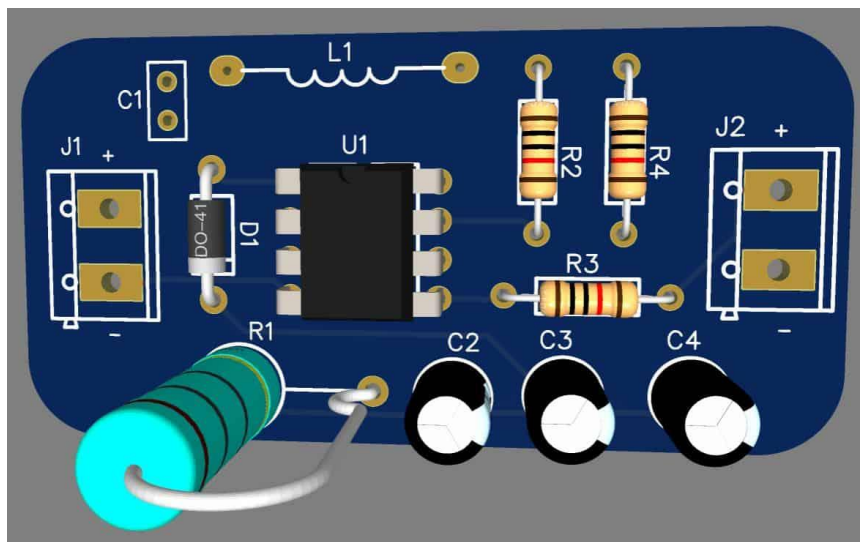
Варіант 23. Модуль DC-DC

Схема



Основна мікросхема – MC34063A

Зовн. вигляд плати

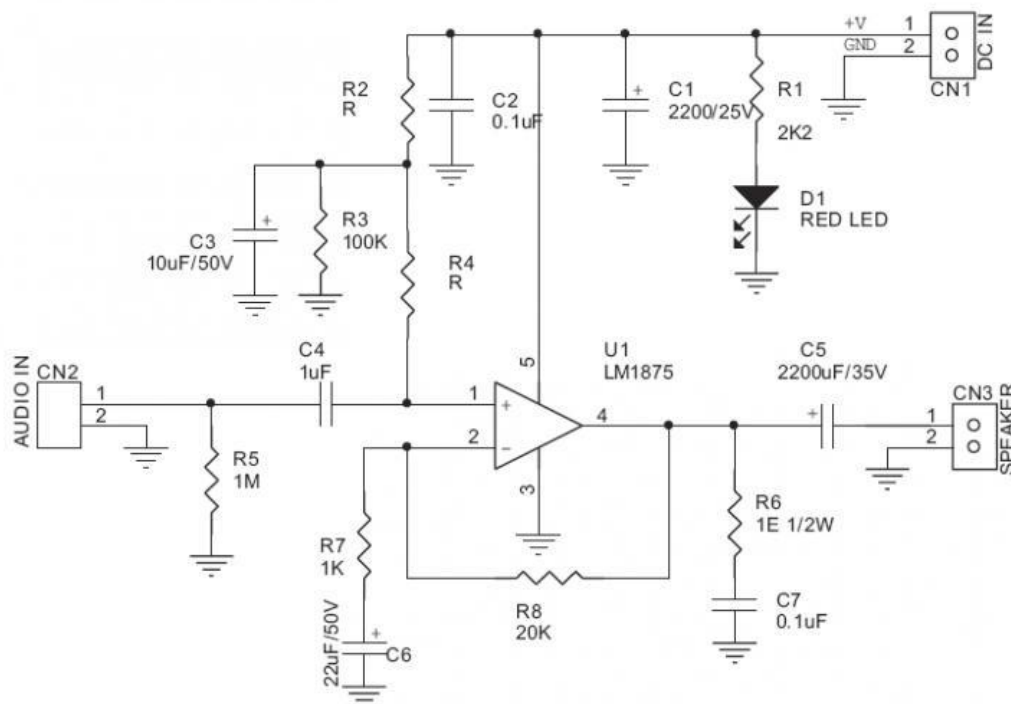


Посилання на
інф. ресурс

<https://how2electronics.com/diy-3-7v-to-9v-boost-converter-circuit-using-mc34063/>

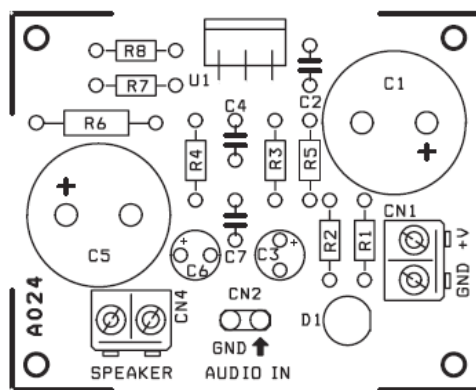
Варіант 24. Підсилювач

Схема



Основна мікросхема – LM1875

Зовн. вигляд плати

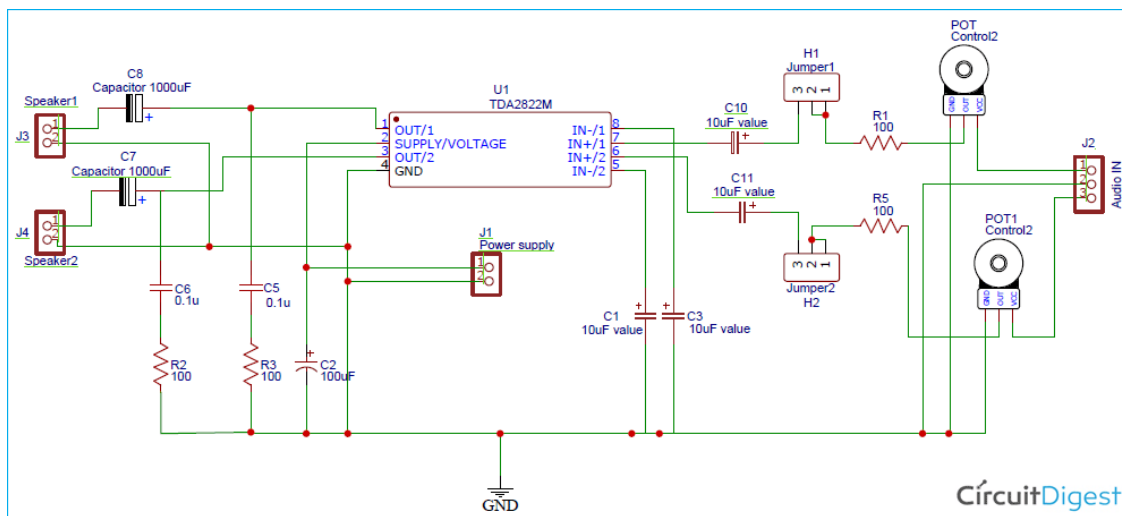


Посилання на
інф. ресурс

<https://www.electronics-lab.com/project/20w-audio-amplifier-using-lm1875/>
<https://www.twovolt.com/2016/08/18/20w-mono-audio-amplifier-using-lm1875/>

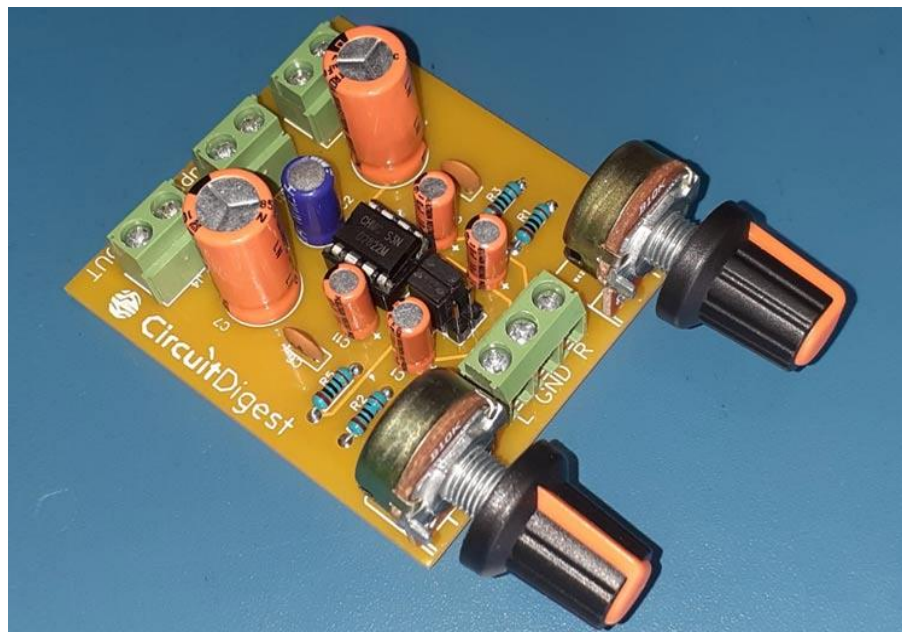
Варіант 25. Підсилювач

Схема



Основна мікросхема – TDA2822

Зовн. вигляд плати

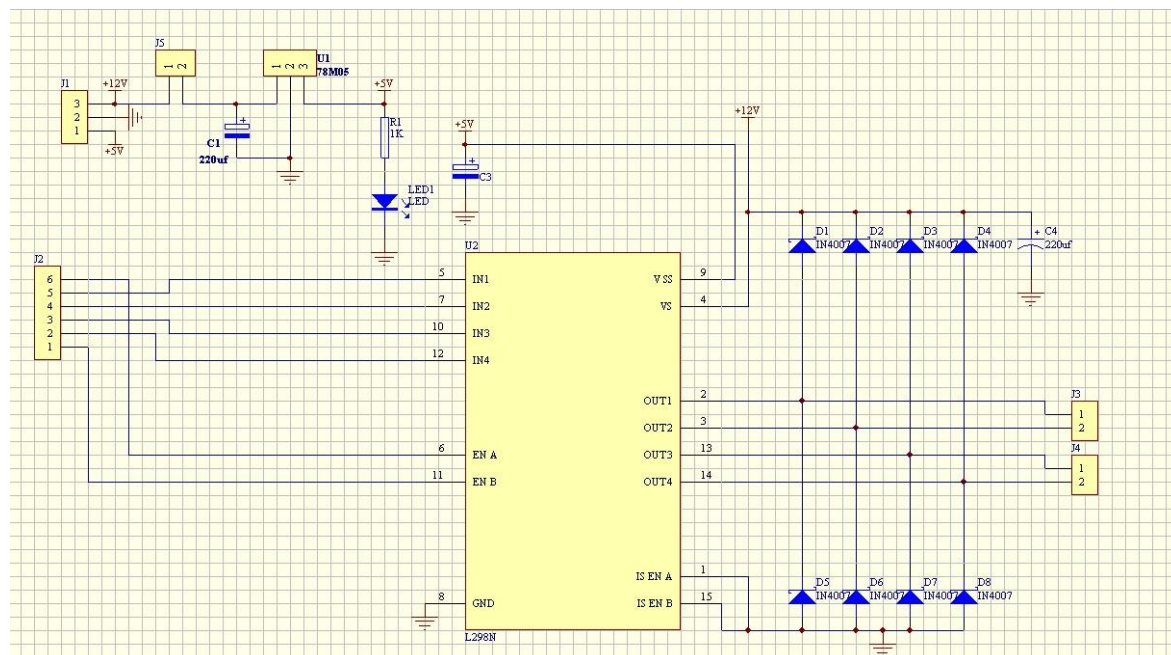


Посилання на
інф. ресурс

<https://circuitdigest.com/electronic-circuits/build-a-simple-stereo-audio-amplifier-board-using-tda2822>

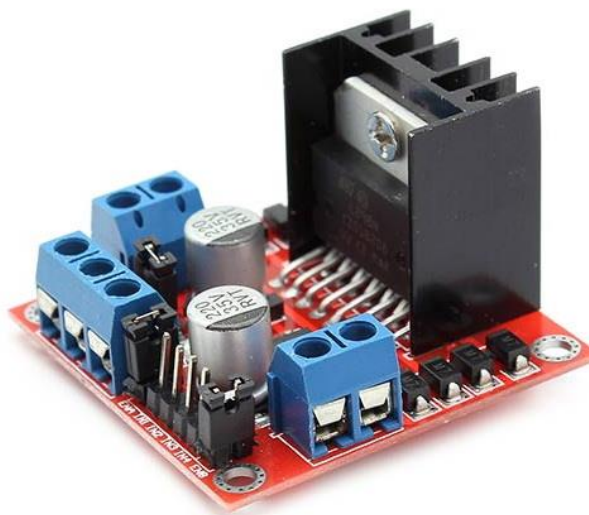
Варіант 26. Модуль драйвера двигунів L298N

Схема



Основна мікросхема – L298N

Зовн. вигляд плати

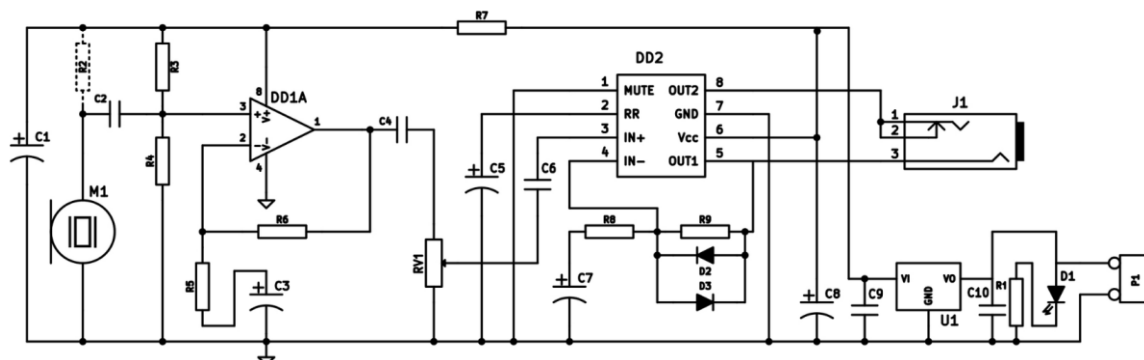


Посилання на
інф. ресурс

<https://www.hotmcu.com/motor-driver-2a-dual-l298-hbridge-p-45.html>
<https://arduino.ua/prod406-draiver-dvyh-dvigateli-na-l298n>

Варіант 27. Електронний стетоскоп

Схема



Основні мікросхеми – TL072, MC34119P

R1 = 1 кОм	RV1 = 50 кОм (потенціометр)	U1 - 78L05
R2 = 10 кОм	C1, C8 = 100 мкФ (16 В)	DD1 - TL072 або TL082
R3, R4 = 1 МОм	C2, C4 = 0,1 мкФ	DD2 - MC34119P
R5, R7 = 750 Ом - 1 кОм	C9, C10 = 0,1 мкФ	D1 - світлодіод Ø3 мм
R6 = 100 кОм	C6 = 0,1 мкФ	J1 - гніздо аудіо
R8 = 3 кОм	C3, C5 = 4,7 мкФ (16 В)	P1 - роз'єм для підключення живлення
R9 = 5,6 кОм	C7 = 1 мкФ (16 В)	M1 - роз'єм для підключення п'єзопластини

Зовн. вигляд плати

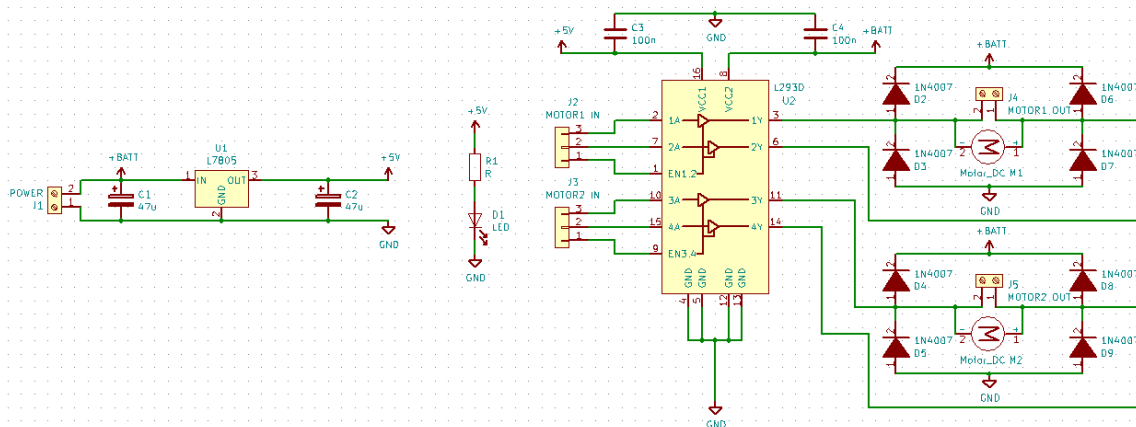


Посилання на
інф. ресурс

<https://radio-kit.com/image/catalog/pdf/K-245.pdf>

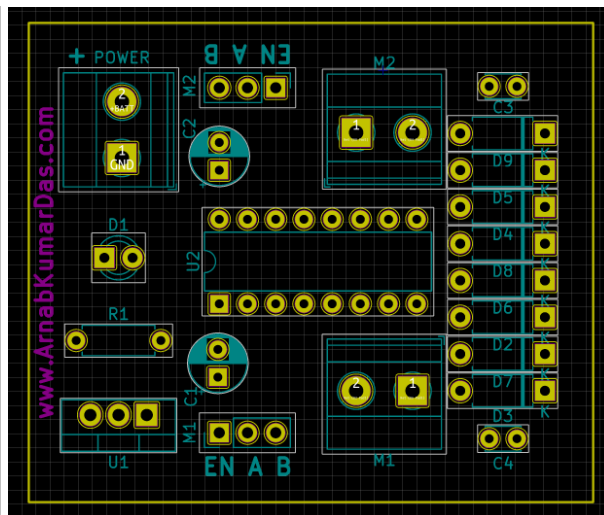
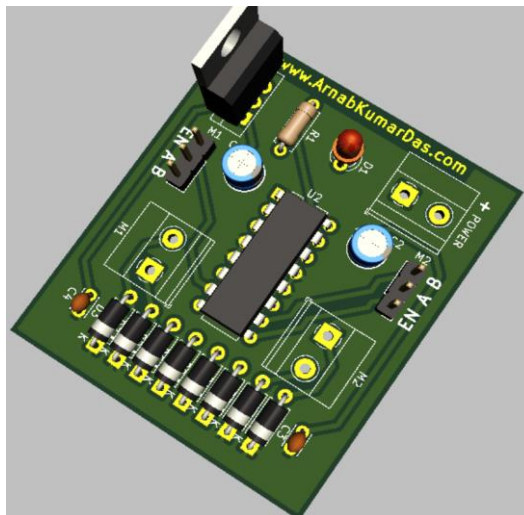
Варіант 28. Модуль драйвера двигунів L293D

Схема



Основна мікросхема – L293D

Зовн. вигляд плати

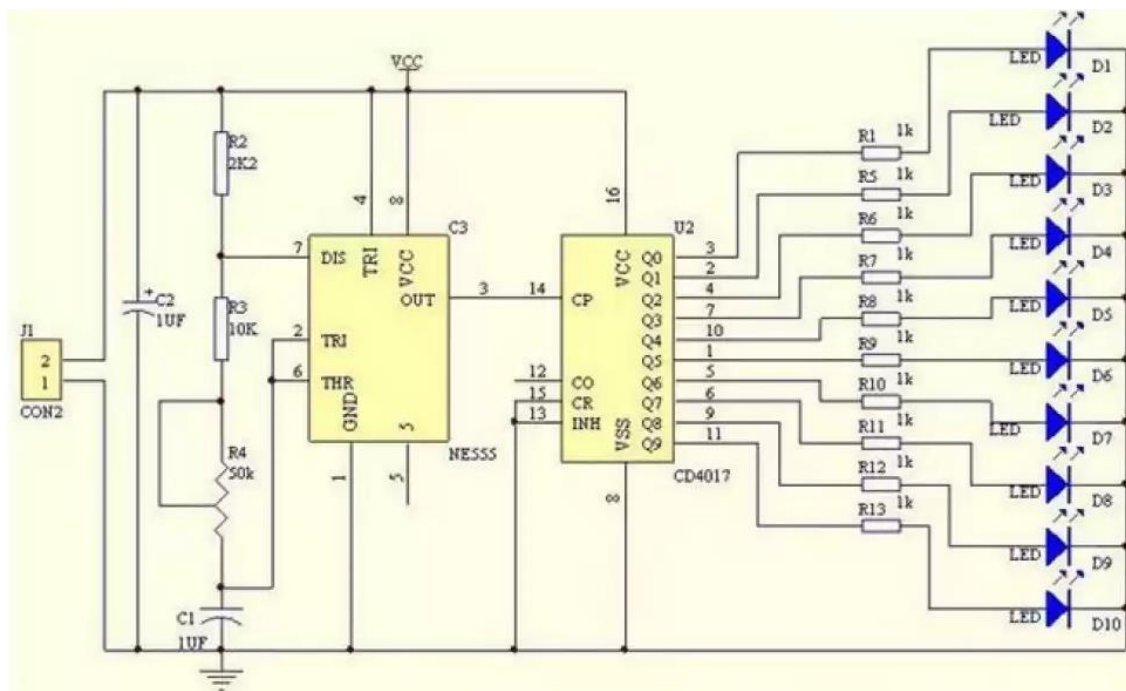


Посилання на
інф. ресурс

<https://www.arnabkumardas.com/topics/pcb/l293d-motor-driver-module-v1-0/>

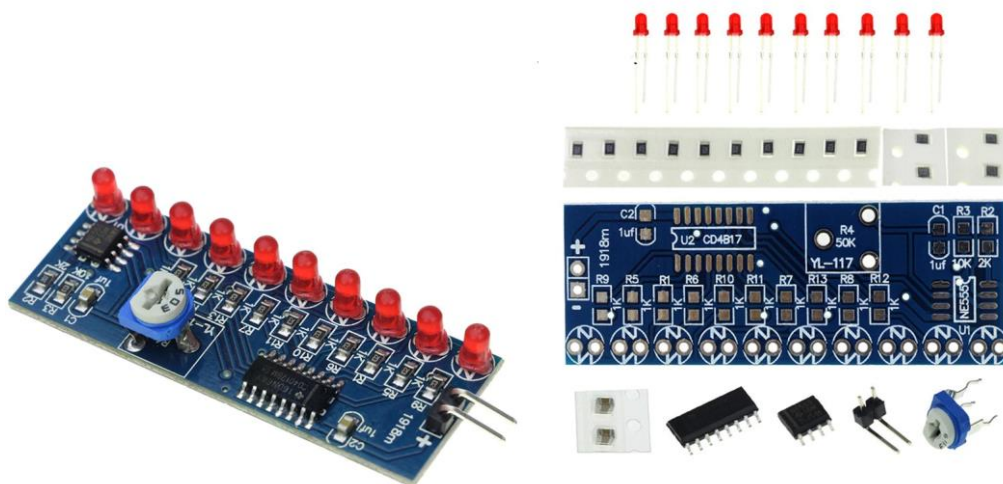
Варіант 30. Модуль, який реалізує «вогні, що біжать»

Схема



Основні мікросхеми – NE555 та CD4017

Зовн. вигляд плати



Посилання на
інф. ресурс

<https://www.aliexpress.com/item/4001146163467.html>

<https://www.onetransistor.eu/2017/12/cd4017-ne555-light-chaser-circuit.html>

ЛІТЕРАТУРА

1. Altium Designer Documentation [Електронний ресурс]. – Режим доступу: <https://www.altium.com/documentation/>
2. Altium's Guide Books [Електронний ресурс]. – Режим доступу: <https://resources.altium.com/guide-books>
3. Create a Footprint using IPC Compliant Footprint Wizard – Part 1 [Електронний ресурс]. – Режим доступу: <https://www.pcb-3d.com/tutorials/create-a-footprint-using-ipc-compliant-footprint-wizard-part-1/>
4. Комп'ютерне проєктування електронних схем: Методичні вказівки до виконання комп'ютерних практикумів для студентів напряму підготовки 6.051003 «Приладобудування» [Електронний ресурс] / Уклад.: Р.М. Галаган, К.М. Серий. – Електронні текстові данні (1 файл: 2,97 Мбайт). – К.: КПІ ім. Ігоря Сікорського, 2017. 163 с. – URL: <https://ela.kpi.ua/handle/123456789/21160>

ДОДАТОК А. РЕКОМЕНДОВАНІ ІНТЕРНЕТ-РЕСУРСИ

Інформаційні ресурси, що пов'язані безпосередньо із Altium Designer

www.altium.com – офіційний сайт фірми Altium, яка володіє торговими марками P-CAD, Protel та Altium Designer.

<https://www.altium.com/documentation/> – офіційна документація програми Altium Designer.

Відеоуроки, в яких пояснюються принципи роботи в Altium Designer

<https://www.youtube.com/playlist?list=PLXvLToQzgxdfKKQn2wmpuSXz6sROQmO6R>
– Altium Tutorial for Beginners - Starting with Altium Designer (YouTube-канал Robert Feranec).

<https://www.youtube.com/playlist?list=PLXvLToQzgdduBaD4horowdWgcG5uGUW4>
– Altium Tutorial for Beginners (YouTube-канал Robert Feranec).

https://www.youtube.com/playlist?list=PLDclr_SCaTAxEpaE0uf9RDQUNtW5YSoxW
– Altium Designer PCB Design Tutorial for Beginners (v20 and 21) (YouTube-канал Tech Ed Kirsch (TEK)).

https://www.youtube.com/playlist?list=PL3aaAq2OJU5H_Jj72DObh5kNh6Nr4xNS0
– Getting Started With Altium Designer (YouTube-канал Altium Academy).

https://www.youtube.com/playlist?list=PL3aaAq2OJU5H_V1wJfcxn9RtUf2hs2FU6
– Altium Designer 19 Essentials Online Training (YouTube-канал Altium Academy).

https://www.youtube.com/playlist?list=PL5qNU_CnKRIDc3bnHKuNPcNw4TZUzemfv
– ALTIUM TUTORIALS (YouTube-канал RGB Engineering)

Інформаційні ресурси, що містять бібліотеки 3D моделей корпусів електронних компонентів

<https://grabcad.com/> – сайт, на якому зібрані різноманітні 3D моделі, зокрема й моделі корпусів електронних компонентів. Доступ до моделей безкоштовний.

<https://www.3dcontentcentral.com/> – сайт, на якому зібрані 3D моделі корпусів електронних компонентів. Доступ до моделей безкоштовний.

<https://www.ultralibrarian.com> – сайт, що містить величезну кількість УГП, посадкових місць та 3D моделей корпусів електронних компонентів. Доступ до більшості моделей платний.

Інформаційні ресурси, що пов'язані зі стандартами в галузі електроніки

www.jedec.org – сайт комітету інженерної стандартизації в галузі електроніки.

<https://www.ipc.org/> – сайт міжнародної організації, що займається розробкою стандартів та вимог щодо виробництва електронної техніки, друкованих плат тощо.

<http://www.iec.ch/> – сайт міжнародної організації зі стандартизації у сфері електричних, електронних та суміжних технологій.

Інформаційні ресурси із інформацією про електроніку та електронні компоненти

<https://www.youtube.com/@ALLABOUTELECTRONICS> – YouTube-канал «ALL ABOUT ELECTRONICS».

https://www.youtube.com/playlist?list=PLWv9VM947MKjuqlJVp5m_Edf66SrFSHx2 – Electricity Basics (YouTube-канал The Engineering Mindset).

https://www.youtube.com/playlist?list=PLWv9VM947MKi_7yJ0_FCfzTBXpQU-Qd3K – Electrical Engineering Basics (YouTube-канал The Engineering Mindset).

<https://diodov.net/> – сайт, на якому розміщені уроки з електроніки для початківців та уроки з програмування мікроконтролерів.

<https://www.twovolt.com/> – сайт, на якому розміщена величезна кількість найрізноманітніших проєктів електронних схем.

ДОДАТОК Б.

СТАНДАРТИ З РОЗРОБКИ УМОВНИХ ГРАФІЧНИХ ПОЗНАЧЕНЬ, ЕЛЕКТРИЧНИХ СХЕМ, ПОСАДКОВИХ МІСЦЬ ЕЛЕКТРОННИХ КОМПОНЕНТІВ ТА ДРУКОВАНИХ ПЛАТ

Стандарти, що регламентують розробку умовних графічних позначень електронних компонентів

Важливо розуміти, що розробка УГП електронних компонентів та створення принципових схем регламентується стандартами, дотримання яких є обов'язковим. Це дозволяє будь-кому ознайомитися з принциповою схемою і відносно швидко зрозуміти, яким чином вона працює. УГП використовуються для відображення різних електронних компонентів і пристроїв на схемах: проводів, роз'ємів, пасивних та активних дискретних електронних компонентів, інтегральних мікросхем тощо. Використовуючи загальний набір УГП електронних компонентів, інженери з усього світу можуть передавати інформацію про схеми коротко та без двозначності.

Хоча сьогодні УГП електронних компонентів значною мірою [стандартизовані на міжнародному рівні](#), проте можуть відрізнятися від країни до країни або залежати від інженерної дисципліни, в якій використовуються власні традиційні угоди щодо зовнішнього вигляду УГП. Наразі існує значна кількість стандартів, які можуть бути як національними¹⁴², так і міжнародними. Це може призводити до плутанини та помилок.

Особливою ситуація нині¹⁴³ є в Україні, де фактично відсутні власні стандарти з розробки УГП. Це пов'язано із тим, що раніше використовувались так звані «ГОСТи», які були прийняті ще за часів Радянського Союзу. Деякий час вони були діючими на території України, проте наразі усі радянські ГОСТи втратили свою силу. Серед стандартів, що вже не актуальні, можна виділити наступні:

- **ГОСТ 2.759-82.** ЕСКД. Обозначения условные графические в схемах. Элементы аналоговой техники;
- **ГОСТ 2.743-91.** ЕСКД. Обозначения условные графические в схемах. Элементы цифровой техники;

¹⁴² Деякі національні стандарти є варіантами міжнародних.

¹⁴³ Можливо, з часом ситуація зміниться на краще і наведена далі інформація стане неактуальною.

- **ГОСТ 2.710-81.** ЕСКД. Обозначения буквенно-цифровые в электрических схемах;
- **ГОСТ 2.730-73.** Обозначения условные графические в схемах. Приборы полупроводниковые.

Наразі ці стандарти можна використовувати лише в ознайомчих цілях. Проте нескладно помітити, що перераховані вище стандарти були прийняті понад 30 років тому, а схемотехніка та електроніка зробили значний стрибок з того часу, що обумовило появу нових видів електронних компонентів. Очевидно, що застарілі стандарти мали б бути замінені новими ДСТУ, які враховували б сучасний стан речей. Однак на момент написання цього навчального посібника відповідних ДСТУ саме з розробки УГП, що мали б офіційний український переклад, не було.

В цій ситуації можна порекомендувати під час розробки УГП дотримуватись міжнародних стандартів. Проте завжди треба бути уважним і з'ясовувати, чи допустимі ці стандарти до застосування в Україні¹⁴⁴ без офіційного затвердження.

Основними міжнародними стандартами є наступні:

- **IEC 60617-DB-12M:2012. Graphical symbols for diagrams** – стандарт Міжнародної електротехнічної комісії, основою якого є британський стандарт BS 3939;
- **ANSI Y32.2. Graphic symbols for electrical and electronics diagrams (including reference designation class designation letters)** – американський стандарт для символів електронних компонентів, що також відомий як [IEEE Std315-1975 \(Reaffirmed 1993\)](#);
- **AS 1102. Graphical symbols for electrotechnology** – австралійський стандарт для символів електронних компонентів, який є дещо модифікованою версією стандарту IEC 60617.

Найбільш використовуваними є стандарти IEC та ANSI/IEEE. Хоча вони мають ряд відмінностей, проте загалом дуже подібні, оскільки організації зі стандартизації усвідомлюють потребу в загальних правилах побудови символів.

Важливо розуміти, що будь-яке УГП (незалежно від стандарту) є набором простих геометричних фігур, ліній та виводів.

¹⁴⁴ Загалом, якщо ви розробник електричної схеми, то використання тих чи інших стандартів ви у першу чергу узгоджуватимете із замовником або їх визначатиме фірма, у якій ви працюватимете. І навіть, якщо деякі стандарти не прийняті на державному рівні, то замовник (або фірма) може вам конкретно вказати, що потрібно використовували той чи інший стандарт.

Якщо говорити про відмінності між різними стандартами (не тільки переліченими вище), то здебільшого вони стосуються вигляду і розмірів символів, якими позначаються дискретні пасивні та активні електронні компоненти, а також можуть бути різні візуальні подання роз'ємів та перемикачів тощо. Якщо говорити про вигляд інтегральних мікросхем, то зазвичай основою їх УГП є прямокутник із іменованими виводами, що виходять із його сторін¹⁴⁵.

Будь-яке УГП, окрім простих геометричних фігур, ліній та виводів, також має містити позиційне позначення (англ. designator) – умовний напис, що характеризує клас електронного компоненту (наприклад, **R** – резистор, **C** – конденсатор, **VT** – транзистор тощо). Із загальноприйнятими стандартами позиційних позначень можна ознайомитись у [IEEE Std315-1975](#). Проте і тут можуть бути нюанси. Зокрема, так історично повелось, що у нас мікросхемам прийнято давати позиційні позначення у вигляді аббревіатури **DA** (аналогова мікросхема) та **DD** (цифрова мікросхема), тоді як західні стандарти регламентують використання символу **U** для усіх типів мікросхем.

Таким чином, можемо зробити висновок, що хоча різні стандарти і можуть бути близькими один до одного, проте завжди існують відмінності. Тому перед розробкою того чи іншого проекту електричної схеми потрібно визначитись із тим стандартом, який буде базовим для проекту.

Стандарти, що регламентують розробку електричних схем

Зі стандартами, що регламентують правила розробки електричних схем ситуація наступна. В середині 2000-х та на початку 2010-х відбулись спроби оновити деякі із застарілих стандартів. Причому для тих часів необхідно враховувати таку специфіку: спочатку створювались так звані «міждержавні стандарти» російською мовою, які приймалися групою країн, серед яких була і Україна. Зокрема, до таких стандартів належать **ГОСТ 2.701-2008. Единая система конструкторской документации. Схемы. Виды и типы. Общие требования к выполнению** та **ГОСТ 2.702-2011 ЕСКД. Правила выполнения электрических схем**. Обидва стандарти є діючими, причому другий стандарт має офіційний переклад та назву **ДСТУ ГОСТ 2.702:2013. Єдина система**

¹⁴⁵ Хоча тут бувають винятки. Наприклад, УГП підсилювача у деяких стандартах має вигляд трикутника, а УГП логічних елементів будується із використанням вигнутих ліній.

конструкторської документації. Правила виконання електричних схем (ГОСТ 2.702-2011, IDT), тоді як перший так і не оформлено як ДСТУ (відповідно, він не має українського перекладу).

Варто зауважити, що в цих стандартах в загальному також описані вимоги до створення УГП, які можна використовувати під час роботи над проєктами.

Стандарти, що регламентують розробку посадкових місць електронних компонентів

Зі стандартами, що регламентують розробку ПМ електронних компонентів, ситуація дещо простіша. Якщо зовнішній вигляд УГП дійсно може відрізнятись в різних країнах, то корпуси електронних компонентів, для яких розробляються ПМ, мають стандартні геометричні характеристики, тому тут не може бути різного трактування.

Здебільшого стандарти, що пов'язані із розробкою ПМ, визначають геометричні розміри КМ електронних компонентів, електричні, механічні та міцнісні характеристики з'єднань тощо.

Основними стандартами є наступні:

- **IEC 61188-7:2017.** Printed boards and printed board assemblies - Design and use - Part 7: Electronic component zero orientation for CAD library construction;
- **IEC 61188-6-1:2021.** Circuit boards and circuit board assemblies - Design and use - Part 6-1: Land pattern design - Generic requirements for land pattern on circuit boards;
- **IEC 61188-6-2:2021.** Circuit boards and circuit board assemblies - Design and use - Part 6-2: Land pattern design - Description of land pattern for the most common surface mounted components (SMD);
- **IEC 61188-6-4:2019.** Printed boards and printed board assemblies - Design and use - Part 6-4: Land pattern design - Generic requirements for dimensional drawings of surface mounted components (SMD) from the viewpoint of land pattern design;
- **IPC-7351.** Generic Requirements for Surface Mount Design and Land Pattern Standard.

Стандарти, що регламентують розробку друкованих плат та їх топологію

Стандарти, що пов'язані із розробкою ДП, визначають особливості компонування та розташування електронних компонентів, способи формування топології плати, механічні та міцнісні характеристики, способи виробництва тощо.

Основними стандартами є наступні:

- **IPC-2221A.** Generic Standard on Printed Board Design;
- **IPC-6011.** Generic Performance Specification for Printed Boards;
- **J-STD-001G.** Requirements for Soldered Electrical and Electronic Assemblies.

ДОДАТОК В.

СТВОРЕННЯ ПРАВИЛ ПРОЄКТУВАННЯ У РЕДАКТОРІ ПЛАТ ALTIUM DESIGNER

Редактор плат використовує концепцію правил проєктування, завдяки яким можна тонко налаштувати проєкт та мінімізувати кількість помилок. У версії **Altium Designer 17.1** всього існує 10 категорій правил проєктування, у кожній з яких можна виділити кілька підкатегорій. Правила проєктування охоплюють вимоги до електричних параметрів, трасування, виготовлення, розміщення та цілісності сигналу тощо.

Правильно налаштувавши правила перед початком роботи над платою, ви зможете повністю зосередитися на проєктуванні, тому що будете знати, що програма повідомить вам про кожну помилку.

Основну інформацію про те, яким чином переглядати, керувати та створювати правила, можна знайти у довідці на офіційному сайті [Altium](https://www.altium.com). Для цього необхідно знайти розділ [Defining, Scoping & Managing PCB Design Rules in Altium Designer](#). Зверніть увагу, що для різних версій програми цей розділ може містити різну інформацію. Ви можете самі обрати версію програми, для якої хочете переглянути довідку.

Додатково рекомендується ознайомитись із наступними відеоуроками:

- [PCB Rules | Altium Designer 19 Essentials | Module 19](#);
- [Essential PCB Rules - Altium Designer](#);
- PCB DESIGN RULES ALTIUM TUTORIAL | ALTIUM DESIGN RULE TUTORIAL ([part 1](#), [part 2](#), [part 3](#), [part 4](#)).

ДОДАТОК Г.

РЕКОМЕНДАЦІЇ З ВИКОРИСТАННЯ МИШКИ ТА ГАРЯЧИХ КЛАВІШ В ALTIUM DESIGNER 17.1

Для того, щоб ефективно працювати у програмі **Altium Designer**, важливо розуміти, як ефективно працювати із мишкою, а також знати і використовувати гарячі клавіші або комбінації клавіш.

Важливо!

Результат використання одних й тих самих гарячих клавіш може залежати від того, чи у Редакторі схем, чи у Редакторі плат вони застосовуються. А також може залежати від поточної дії (наприклад, натискання гарячої клавіші під час переміщення деякого об'єкта і без переміщення призводить до різних результатів) та системних налаштувань.

Перелік деяких загальних гарячих клавіш та їх комбінацій

- **<X>** – дзеркальне відображення компонента по осі X (працює під час розміщення або переміщення компонента);
- **<Y>** – дзеркальне відображення компонента по осі Y (працює під час розміщення або переміщення компонента);
- **<Shift + Space>** – зміна кута під час прокладання траси (у Редакторі плат: 45°/90°) або ланцюга (у Редакторі схем);
- Натискання клавіші **<Tab>** під час розміщення об'єкта (це може бути ланцюг, графічний примітив, посадкове місце, траса на платі тощо) відкриває вікно властивостей;
- Натискання клавіші **~ (тільда)** під час розміщення деякого об'єкта (лінія, доріжка, ланцюг тощо) відкриває контекстне меню, в якому перераховані усі доступні дії та гарячі клавіші;
- **<Ctrl + Page Down>** – автоматично підбирається такий масштаб, щоб на екрані відображались усі елементи документа;
- **<Ctrl + стрілки>** – переміщує виділений об'єкт у вузлах вздовж обраного напрямку (у Редакторі схем працює тільки, якщо об'єкт не зв'язаний із іншими об'єктами за допомогою з'єднань).

Перелік деяких гарячих клавiш Редактора схем

- **<Space>** – поворот виділеного компонента на 90° проти годинникової стрілки;
- **<Shift + Space>** – поворот виділеного компонента на 90° за годинниковою стрілкою;
- **<G>** – перемикання між сітками переміщення.

Перелік деяких гарячих клавiш Редактора плат

- **<Space>** – поворот компонента на 90° проти годинникової стрілки під час його переміщення (Важливо! Поворот за допомогою Пробілу у Редакторі плат працює тільки у режимі переміщення компоненту. Тобто, спочатку потрібно курсором миші виділити компонент, почати його переміщувати і тільки потім натискати Пробіл, щоб компонент повернувся на 90°);
- **<Shift + Space>** – поворот компонента на 90° за годинниковою стрілкою під час його переміщення;
- **<Ctrl + M>** – вимірювання відстаней (Зверніть увагу, що після завершення вимірювань відстані залишається графічна мітка із результатом. Для того, щоб очистити робоче поле від графічних міток із результатами вимірювань, необхідно натиснути комбінацію клавiш **<Shift + C>**);
- **<Shift + Ctrl + G>** – встановлення заданого кроку сітки.

Дії, що виконуються за допомогою мишки та клавiш клавіатури (загальні для Редактора схем і Редактора плат)

- Виділення деякого об'єкту відбувається за допомогою одноразового клацання на ньому **ЛКМ** (якщо кілька об'єктів візуально накладаються один на одного в точці виділення, то програма виведе додаткове контекстне вікно, в якому обирається потрібний об'єкт);
- Клацання **ЛКМ** на об'єктах при натиснутій клавiші **<Shift>** дає змогу виділити кілька об'єктів;
- Подвійне клацання **ЛКМ** відкриває вікно властивостей об'єкта;
- Клацання **ПКМ** на деякому об'єкті відкриває його контекстне меню, із якого, зокрема, можна викликати вікно властивостей;

- Рух мишкою при затиснутій **ПКМ** переміщує робочу область документа;
- Рух мишкою при одночасно затиснутій клавіші **<Ctrl>** та **ПКМ** масштабує робочу область документа;
- Обертання колеса мишки **Mouse Wheel** переміщує документ вгору/вниз (якщо водночас буде затиснута клавіша **<Shift>**, то переміщення відбуватиметься ліворуч/праворуч);
- Обертання колеса мишки **Mouse Wheel** при натиснутій клавіші **<Ctrl>** масштабує зображення.

Дії, що виконуються за допомогою мишки та клавіш клавіатури (тільки у Редакторі схем)

- Для переміщення об'єкта без відриву від ланцюга необхідно його виділити за допомогою **ЛКМ**, затиснути її та почати рухати мишкою¹⁴⁶ (якщо потрібно переміщувати із відривом від ланцюга, то додатково утримують натиснутою клавішу **<Ctrl>**).

Дії, що виконуються за допомогою мишки та клавіш клавіатури (тільки у Редакторі плат)

- Якщо об'єкт виділити за допомогою **ЛКМ** та, утримуючи її, почати рухати мишкою, то об'єкт переміщуватиметься із відривом¹⁴⁷ від траси (проте саме з'єднання з іншим компонентом у вигляді тонкої світлої лінії залишиться);

¹⁴⁶ Така поведінка в Редакторі схем можлива за умови, якщо в його загальних налаштуваннях (вікно **Preferences**) активована опція **Schematic** → **Graphical Editing** → **Always Drag**. Якщо ж ця опція не активована, то за замовчуванням переміщення компоненту на схемі відбуватиметься із відривом від ланцюга.

¹⁴⁷ Якщо ж потрібно переміщувати деякий об'єкт разом із трасою, то необхідно виконати такі дії. Спочатку в глобальних налаштуваннях (вікно **Preferences**) для опції **PCB Editor** → **General** → **Comp Drag** із випадного списку обрати пункт **Connected Tracks** та зберегти налаштування. Далі із меню **Edit** виконати команду **Move** → **Drag**, внаслідок чого подальше переміщення об'єкту відбуватиметься із подовженням сегменту траси від останньої точки з'єднання до нового розташування об'єкта (для виходу із цього режиму потрібно натиснути клавішу **<Esc>**). Зауважте, що якщо опція **Comp Drag** не буде активована, то команда **Move** → **Drag** працюватиме як звичайне переміщення із відривом від траси.

- Для того, щоб обертати плату в режимі 3D перегляду, необхідно натиснути та утримувати клавішу <Shift> при натиснутій ПКМ, після чого рухати мишкою;
- Для того, щоб «підсвітити» задану трасу на платі, необхідно затиснути клавішу <Shift> і одночасно навести курсор мишки на неї.

Повний перелік гарячих клавіш Altium Designer

Із повним переліком гарячих клавіш можна ознайомитись у довідці на офіційному сайті:

<https://www.altium.com/documentation/altium-designer/altium-designer-shortcut-keys?version=17.1>

Або за цим посиланням:

<http://microsin.net/adminstuff/others/altium-designer-editor-shortcuts.html>

ДОДАТОК Д.

РОЗМІРИ НАЙБІЛЬШ ПОПУЛЯРНИХ ТИПІВ SMD КОРПУСІВ ПАСИВНИХ ЕЛЕКТРОННИХ КОМПОНЕНТІВ

На рис. Д.1 схематично представлено типовий SMD корпус.

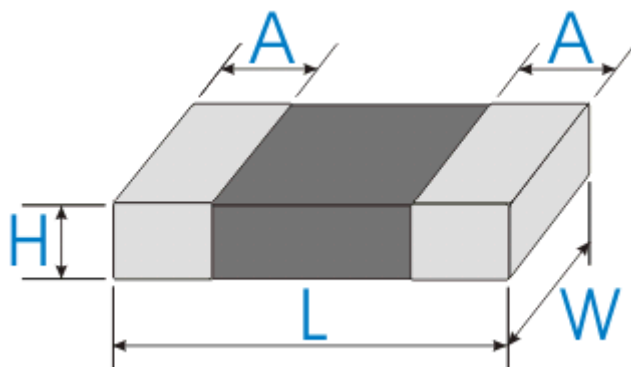


Рис. Д.1. Схематичне зображення SMD корпусів

Типорозмір SMD корпусів складається з чотирьох цифр, які відображають його довжину і ширину¹⁴⁸. Наприклад, маркування корпусу 0805 означає: довжина – 0,08 дюймів (≈ 2 мм), ширина – 0,05 дюймів ($\approx 1,25$ мм), а маркування 1210 означає: довжина – 0,12 дюймів ($\approx 3,2$ мм), ширина – 0,1 дюймів ($\approx 2,5$ мм). Маркування корпусу визначає тільки його довжину і ширину та нічого не говорить про висоту, яка буде залежати від типу компонента (наприклад, висота резистора у корпусі 0805 становить приблизно 0,5 мм, а висота конденсатора у цьому ж корпусі може досягати 1,3 мм).

У табл. Д.1 наведені геометричні характеристики SMD корпусів резисторів, а також вказана розсіювана потужність (однак потрібно враховувати, що це значення характеризує тільки резистори для універсального застосування, тоді як спеціалізовані резистори у відповідних корпусах можуть мати інші значення потужності). Зверніть увагу, що у стовпчику, де відображається висота корпусу H , вказується лише мінімальний допустимий розмір. Насправді він може бути і більшим, залежно від опору. Щоб чітко розуміти висоту корпусу, необхідно дивитись технічну документацію на виріб безпосередньо від виробника.

¹⁴⁸ Існує дві системи кодування розмірів SMD-компонентів. У одній із них для кодування типорозміру використовується довжина та ширина компонента у дюймах (стандарт EIA), а в іншій – у міліметрах.

Таблиця Д.1. Геометричні параметри SMD корпусів резисторів

Тип корпусу за стандартом EIA (у дужках - метричний)	L, мм	W, мм	H, мм	A, мм	Потужність, Вт
0201	0,6	0,3	0,23	0,13	0,05
0402 (1005)	1,0	0,5	0,35	0,2-0,25	0,062
0603 (1608)	1,6	0,8	0,45	0,3	0,1
0805 (2012)	2,0	1,2-1,25	0,45	0,4-0,5	0,125
1206 (3216)	3,2	1,6	0,55	0,5	0,25
1210 (3225)	3,2	2,5	0,55	0,5	0,25
1218 (3245)	3,2	4,5	0,55	0,5	0,25
1812 (4532)	4,5	3,2	0,55	0,5	0,5
2010 (5025)	5,0	2,5	0,6	0,5	0,5
2512 (6432)	6,4	3,2	0,6	0,5	1

У табл. Д.2 наведені геометричні характеристики SMD корпусів конденсаторів.

Таблиця Д.2. Геометричні параметри SMD корпусів конденсаторів

Тип корпусу за стандартом EIA	L, мм	W, мм	H, мм	A, мм
0603	1,6	0,8	0,7-0,9	0,4
0805	2,0	1,25	0,7-1,25	0,5
1206	3,2	1,6	0,7-1,25	0,75
1210	3,2	2,5	1,0-1,25	0,75

ДОДАТОК Е.

РЕКОМЕНДАЦІЇ ЩОДО РОЗРОБКИ КОНТАКТНИХ МАЙДАНЧИКІВ ДЛЯ SMD-РЕЗИСТОРІВ

На рис. Е.1 та у табл. Е.1 представлені геометричні розміри ПМ для SMD-резисторів.

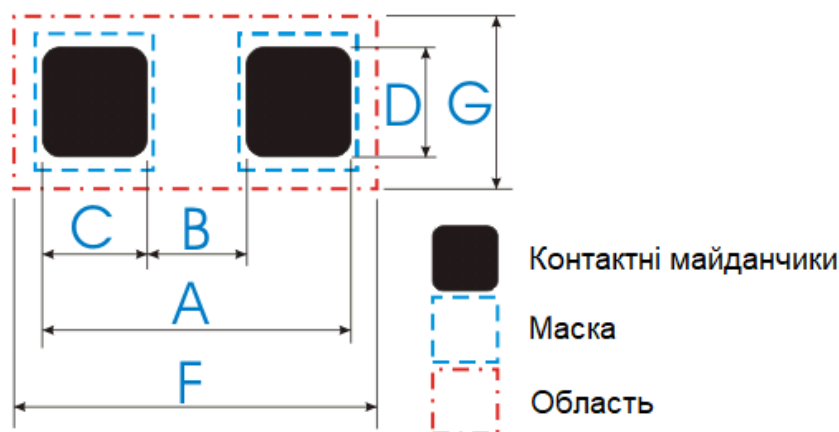


Рис. Е.1. Схематичне зображення посадкового місця SMD-резистору

Таблиця Е.1. Геометричні параметри посадкових місць SMD-резисторів

Пайка оплавленням припою та інфрачервоним випромінюванням							
Типорозмір	Розміри у міліметрах (мм)						Точність розміщення
	A	B	C	D	F	G	
0201	0,75	0,3	0,3	0,3	1,1	0,5	0,05
0402	1,5	0,5	0,5	0,6	1,9	1	0,15
0603	2,1	0,9	0,6	0,9	2,35	1,45	0,25
0805	2,6	1,2	0,7	1,3	2,85	1,9	0,25
1206	3,8	2	0,9	1,6	4,05	2,25	0,25
1218	3,8	2	0,9	4,8	4,2	5,5	0,25
2010	5,6	3,8	0,9	2,8	5,85	3,15	0,25
2512	7	3,8	1,6	3,5	7,25	3,85	0,25
Пайка хвилею							
Типорозмір	Розміри у міліметрах (мм)						Точність розміщення
	A	B	C	D	F	G	
0603	2,7	0,9	0,9	0,8	3,4	1,9	0,25
0805	3,4	1,3	1,05	1,3	4,3	2,7	0,25
1206	4,8	2,3	1,25	1,7	5,9	3,2	0,25
1218	4,8	2,3	1,25	4,8	5,9	5,6	0,25
2010	6,3	3,5	1,4	2,5	7	3,6	0,25
2512	8,5	4,5	2	3,2	9	4,3	0,25

ДОДАТОК Ж. РІЗНІ НЕСТАНДАРТНІ СИТУАЦІЇ, ЩО ПРОЯВЛЯЮТЬСЯ ПІД ЧАС РОБОТИ В ALTIUM DESIGNER, ТА СПОСОБИ ЇХ ПОДОЛАННЯ

Проблема із позиціонуванням примітиву *Designator*

Інколи виникають проблеми із позиціонуванням примітиву *Designator* над УГП електронного компонента. Зазвичай цей примітив має розміщуватись над УГП, проте він може зміститись так, що знаходитиметься в межах контуру або настільки далеко, що виходитиме за межі робочої області і його взагалі не буде видно¹⁴⁹ (таке може відбуватись під час зміни системи вимірювання або розміру робочої області).

Для того, щоб можна було вручну задати позицію *Designator*, необхідно у бібліотеці символів із меню *Tools* виконати команду *Document Options*. У вікні *Schematic Library Options*, що відкриється, у вкладці *Library Editor Options* активувати пункт *Always Show Comment/Designator*. Далі необхідно перейти до того компонента, із яким виникли проблеми, і вручну перемістити *Designator* у задану позицію.

Якщо ж *Designator* виходить за межі робочої області, то ви його зможете побачити, але не зможете виділити мишкою та переміщувати. Спочатку необхідно у властивостях документа встановити такі геометричні розміри, щоб примітив *Designator* опинився у межах цієї області. Після цього з'явиться можливість його переміщувати у задану позицію.

Проблема із відображенням посадкового місця у вікні попереднього перегляду

Після того, як модель ПМ була підключена до символу компонента, вона завжди повинна повністю відображатись у нижній частині робочого простору.

¹⁴⁹ Коли такий елемент буде розміщуватись на принциповій схемі, то система сприйматиме його загальний розмір із урахуванням зміщення примітиву *Designator*, внаслідок чого елемент буде неможливо вільно переміщувати у будь-яку позицію в робочій області принципової схеми.

Однак інколи ПМ не відображається, а видно лише пусту чорну (або сіру) область. Однією із причин цього може бути наступне: коли ви створювали ПМ, то могли деякий примітив (наприклад, лінію, дугу, КМ тощо) випадково розмістити далеко від самого ПМ (причому примітив знаходиться за межами робочої області і його не видно). Однак варто пам'ятати, що програма вважає, що усі графічні елементи у межах робочої області належать до моделі ПМ і, відповідно, масштабує область таким чином, щоб вмістити усі елементи. Саме за рахунок такого масштабування видно тільки темну область і не видно самого ПМ. Щоб виправити ситуацію, необхідно перейти у Редактор посадкових місць і просто видалити усі зайві елементи, які знаходяться далеко за межами моделі ПМ.

Проблема із тим, що не відображаються отвори КМ на платі або в бібліотеці посадкових місць

Інколи виникає ситуація із тим, що не видно отворів КМ на платі, причому в тих КМ, де вони мають бути (рис. Ж.1).

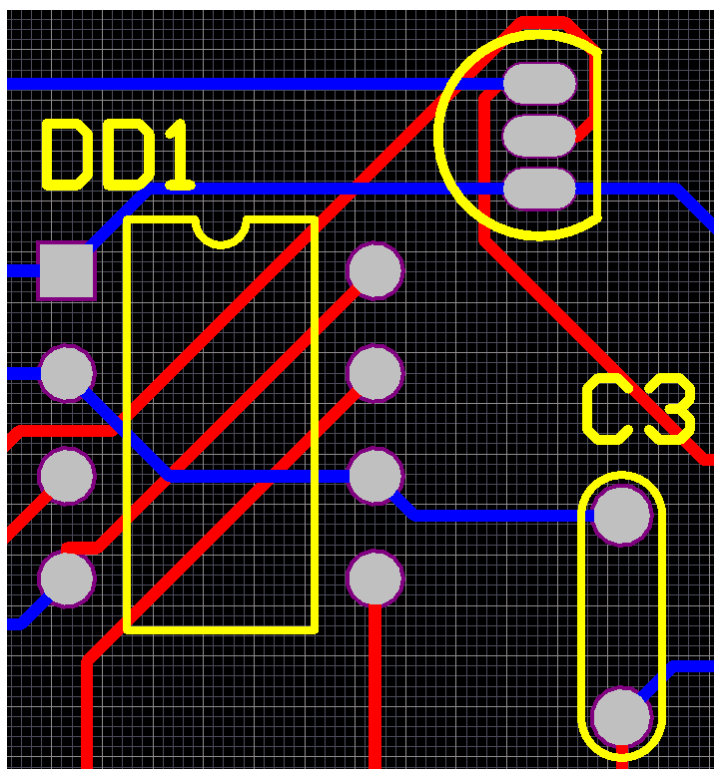


Рис. Ж.1. Приклад плати, на якій не відображаються отвори в КМ

Ця проблема пов'язана із неправильним налаштуванням відображення примітивів на платі. Щоб її виправити, необхідно зайти в меню **Design** та обрати

пункт **Board Layers & Colors**. Відкриється вікно **View Configurations** (рис. Ж.2). Необхідно в області **System Colors** встановити галочку у стовпчику **Show** для параметру **Pad Holes**.

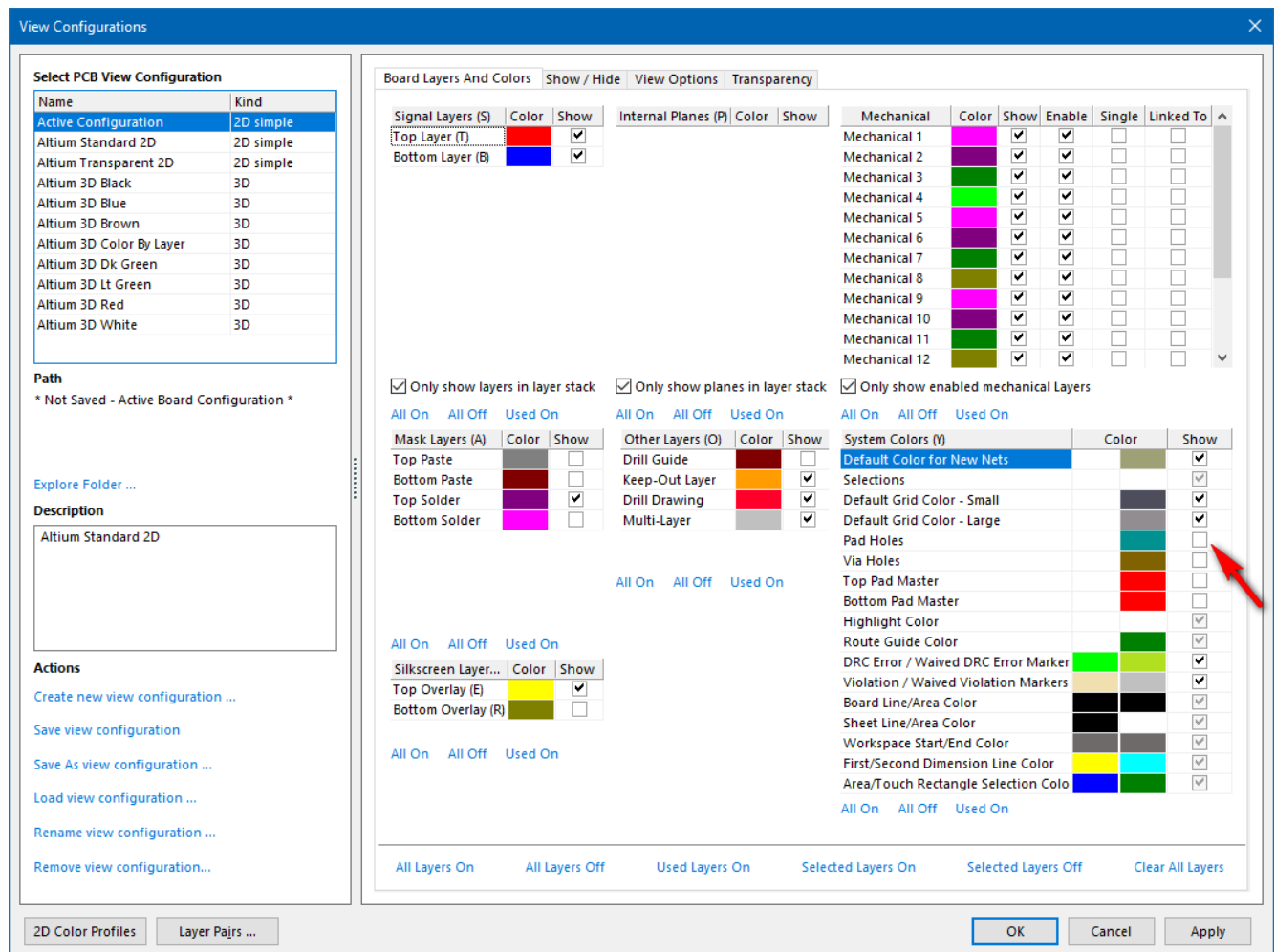


Рис. Ж.2. Налаштування відображення отворів в КМ

ДОДАТОК К.

АНАЛІЗ ТЕХНІЧНОЇ ДОКУМЕНТАЦІЇ (НА ПРИКЛАДІ МІКРОСХЕМИ AD5600)

Мікросхему AD5600 випускає фірма **Analog Devices**. На офіційному сайті знаходимо [технічну документацію](#).

Спочатку розбираємось із тим, яку функцію виконує мікросхема та які має особливості. На першій сторінці документації (рис. К.1) представлена основна інформація про мікросхему AD5600.

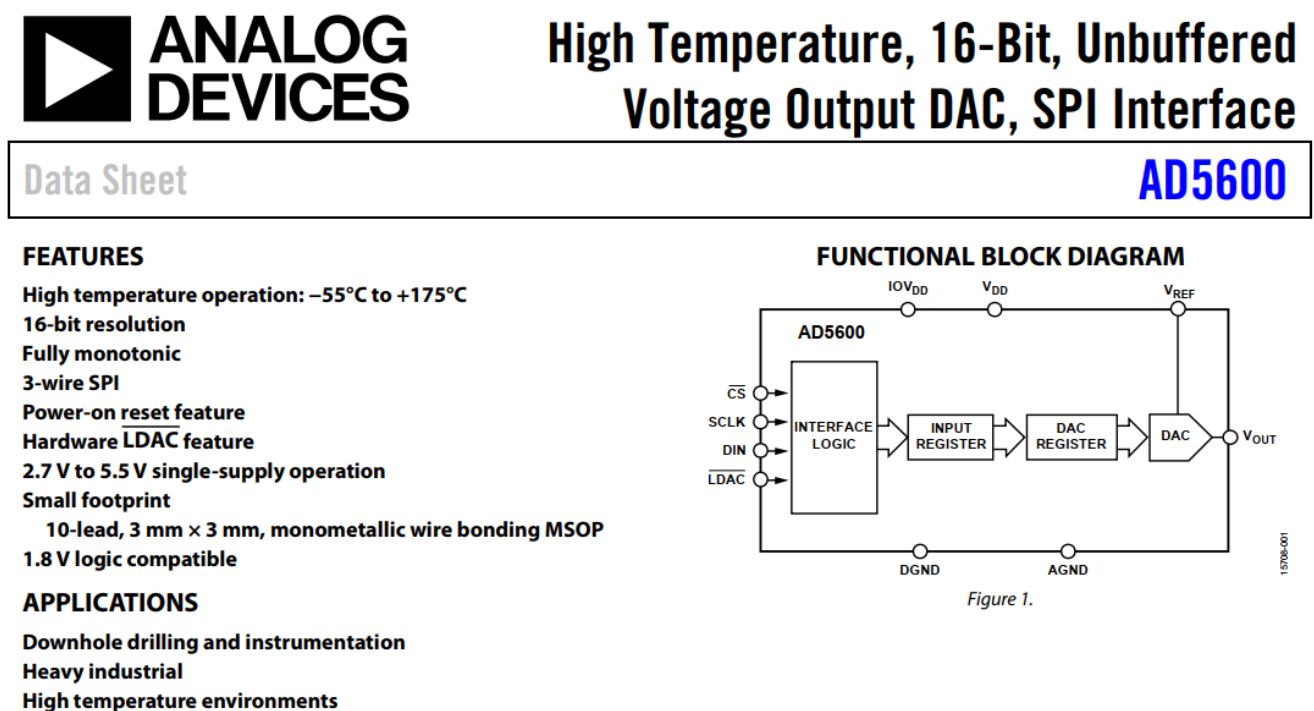


Рис. К.1. Опис мікросхеми AD5600 та пояснення її особливостей

Отже, AD5600 – це 16-бітний цифро-аналоговий перетворювач із послідовним інтерфейсом передачі даних (SPI). Особливістю є те, що мікросхема може працювати в широких температурних діапазонах, зокрема, до $+175^{\circ}\text{C}$.

На тій же першій сторінці документації (див. рис. К.1) також приведена функціональна блок-діаграма, яка може бути використана під час розробки УГП. Звертаємо увагу, що на цій блок-діаграмі представлені виводи та їх найменування, проте не пояснюється їх призначення. Для того, щоб зрозуміти призначення виводів, необхідно знайти інформацію про них. У документації на мікросхему AD5600 ця інформація міститься в таблиці 6 (рис. К.2).

Table 6. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	V _{DD}	Power Supply Input. The device can operate from 2.7 V to 5.5 V. Decouple V _{DD} to AGND.
2	V _{OUT}	Analog Output Voltage from the DAC.
3	AGND	Analog Ground. Ground reference point for all analog circuitry on the device.
4	V _{REF}	Voltage Reference Input. Connect this pin to an external voltage reference.
5	\overline{CS}	Chip Select Input Signal. \overline{CS} frames the serial data input.
6	SCLK	Serial Clock Input Signal. Data is clocked into the serial input register on the rising edge of SCLK.
7	DIN	Serial Data Input Signal. This device accepts 16-bit words. Data is clocked into the serial input register on the rising edge of SCLK.
8	\overline{LDAC}	\overline{LDAC} Input Signal. Pulsing this pin low allows the DAC register to be updated if the input register has new data. This pin can be tied permanently low. In this case, the DAC is automatically updated when new data is written to the input register on the rising edge of \overline{CS} .
9	DGND	Digital Ground. Ground reference point for all digital circuitry on the device.
10	IOV _{DD}	Digital Interface Supply Voltage. The voltage range is 1.65 V to 5.5 V. Decouple IOV _{DD} to DGND.

Рис. К.2. Опис призначення виводів мікросхеми AD5600

Таблиця із описом призначення кожного виводу дає змогу зрозуміти, які виводи є входами, а які – виходами, які виводи належать до виводів живлення, які виводи є двонаправленими, тактовими, інверсними тощо.

Виконаємо аналіз таблиці 6 (див. рис. К.2) із документації, щоб зрозуміти, які виводи належать до виводів живлення і можуть бути прихованими на УГП. Якщо вивід пов'язаний із напругою живлення, то зазвичай в документації використовується термін Power Supply, Supply Voltage тощо. Напруга живлення – це напруга, без якої схема взагалі не буде працювати. Виводи, що підключаються до землі, також належать до виводів живлення. Саме потенціал землі вважається нульовим і відносно нього розраховуються усі напруги в схемі.

В різній документації виводи живлення називаються по-різному, проте є деякі узагальнені позначення¹⁵⁰:

- виводи живлення зазвичай містять у своїй назві символ **V** і можуть позначатись як VCC, V_{CC}, V₊, V₋, +V_S, -V_S тощо;
- виводи землі зазвичай позначаються як GND, AGND, DGND, COM, а також як VSS, V_{SS}, -V_S тощо.

Якщо уважно прочитати опис виводів (див. рис. К.2), то стає очевидним, що виводами живлення є V_{DD} (номер 1), IOV_{DD} (номер 10), а виводами землі AGND (номер 3) – аналогова земля, DGND (номер 9) – цифрова земля. Ці виводи

¹⁵⁰ Представлений тут перелік символічних позначень виводів, які належать до виводів живлення і землі, не є вичерпним, оскільки виробники можуть використовувати власні позначення. Тому важливо уважно читати технічну документацію, щоб зрозуміти, чи належить деякий вивід до виводу живлення або землі.

обов'язково мають бути підключені до джерела живлення, щоб мікросхема могла функціонувати. На УГП вони можуть бути приховані¹⁵¹.

Також не варто думати, що будь-яке позначення, у якому міститься символ **V**, означає вивід живлення, який необхідно приховувати на УГП. Наприклад, зверніть увагу на виводи V_{OUT} (номер 2) та V_{REF} (номер 4). Вивід V_{OUT} хоча і містить у своїй назві літеру **V**, проте із опису очевидно, що він є виходом, з якого знімається сигнал, тому його точно не потрібно приховувати. Якщо говорити про вивід V_{REF} , то спочатку може здатись, що він належить до виводів живлення, без яких схема не функціонуватиме. Проте це не так. V_{REF} – це опорна напруга. З одного боку, щоб схема працювала – ця напруга не потрібна, а з іншого – без неї буде фізично неможливим виконати цифро-аналогове перетворення. Опорна напруга може бути взята від джерела живлення, але це не обов'язково, адже може використовуватись й окреме спеціальне джерело. Тому доцільно було б цей вивід не приховувати, а відображати на УГП.

Тепер, коли було з'ясовано призначення усіх виводів, можна переходити до розробки УГП. На рис. К.3 представлено один варіантів УГП мікросхеми AD5600.

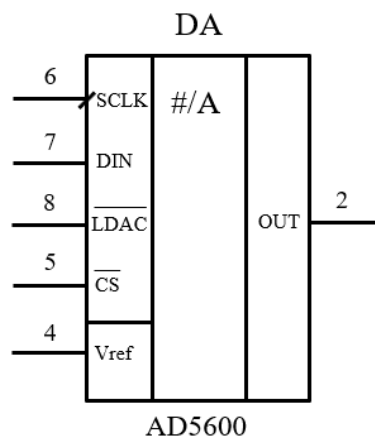


Рис. К.3. УГП мікросхеми AD5600

Наступним важливим кроком є визначення на основі технічної документації, у якому корпусі випускається мікросхема, що дозволить розробити відповідне ПМ. Часто буває так, що одна й та сама мікросхема випускається у різних корпусах. Це обов'язково вказується у технічній документації. У такому випадку розробник сам

¹⁵¹ Важливо розуміти, що виводи живлення не єдині, які можуть бути приховані на УГП. Наприклад, у деяких мікросхемах є виводи, які не виконують жодної функції і не повинні нікуди підключатись. Зазвичай вони позначаються як **NC** (not connected). Їх також не потрібно показувати на УГП.

вирішує, мікросхему в якому корпусі він буде використовувати, після чого створює ПМ під обраний тип корпусу.

Якщо говорити про мікросхему AD5600, то із технічної документації стає зрозумілим, що вона випускається тільки в одному корпусі – MSOP (рис. К.4).

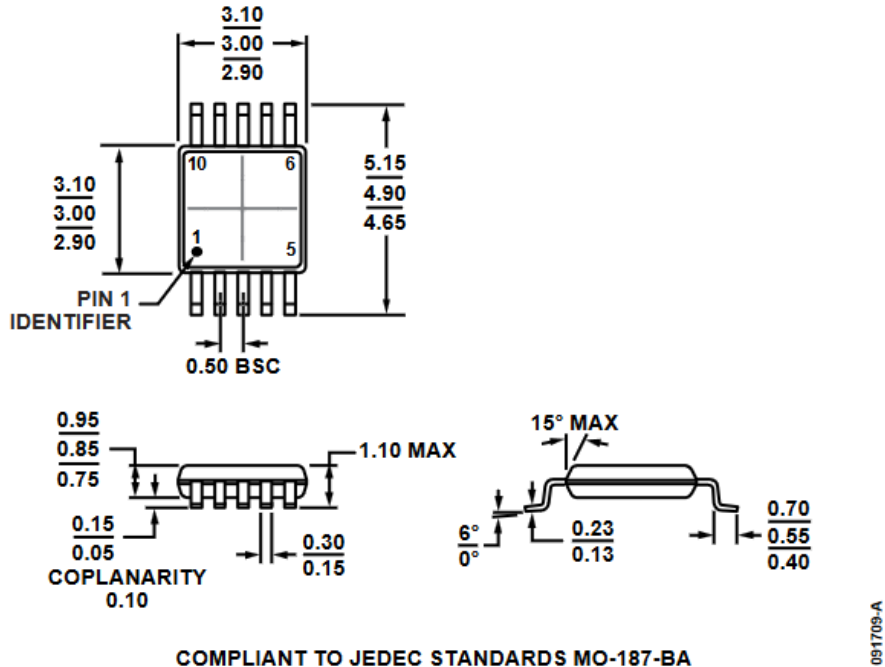


Рис. К.4. Геометричні характеристики корпусу мікросхеми AD5600
(усі розміри в міліметрах)

ПМ в **Altium Designer** для мікросхеми AD5600 створюватиметься саме на основі даних, що представлені на рис. К.4.

ДОДАТОК Л.

ДОДАТКОВІ ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

Додаткове завдання 1

Розробити тільки УГП багатоконтактних роз'ємів, а саме: дво- та чотириконтактного штирового роз'єму, а також дво- та чотириконтактного гніздового роз'єму. Типові геометричні характеристики УГП таких роз'ємів представлені на рис. Л.1.

Тип роз'єму	Символьний код	Умовне позначення роз'єму	Рекомендовані розміри умовного позначення																		
Штировий роз'єм	XP	<div></div> <table><thead><tr><th>Конт.</th><th>Ланцюг</th></tr></thead><tbody><tr><td>1</td><td></td></tr><tr><td>2</td><td></td></tr><tr><td>3</td><td></td></tr></tbody></table>	Конт.	Ланцюг	1		2		3		<table><thead><tr><th>Конт.</th><th>Ланцюг</th></tr></thead><tbody><tr><td>1</td><td></td></tr><tr><td>2</td><td></td></tr><tr><td>3</td><td></td></tr><tr><td>...</td><td></td></tr></tbody></table> <div><div>10 мм20 мм8 мм (min)</div></div>	Конт.	Ланцюг	1		2		3		...	
Конт.	Ланцюг																				
1																					
2																					
3																					
Конт.	Ланцюг																				
1																					
2																					
3																					
...																					
Гніздовий роз'єм	XS	<div></div> <table><thead><tr><th>Конт.</th><th>Ланцюг</th></tr></thead><tbody><tr><td>1</td><td></td></tr><tr><td>2</td><td></td></tr><tr><td>3</td><td></td></tr></tbody></table>	Конт.	Ланцюг	1		2		3												
Конт.	Ланцюг																				
1																					
2																					
3																					

Рис. Л.1. Умовні позначення багатоконтактних роз'ємів

Запустіть **Altium Designer**. Активуйте вкладку **SCH Library** і створіть новий компонент з іменем **Male Contact 2**.

Намалюйте символ штирового роз'єму на два контакти, використовуючи геометричні характеристики, що наведені на рис. Л.1 (причому висоту кожної комірки задайте як 8 мм). Після цього намалюйте чотириконтактний штировий роз'єм (щоб спростити роботу, створіть копію двоконтактного штирового роз'єму, після чого її відредагуйте), задавши його ім'я як **Male Contact 4**. Остаточний вигляд УГП штирових роз'ємів представлено на рис. Л.2¹⁵².

¹⁵² Зверніть увагу, що у такому вигляді створені УГП штирових роз'ємів є лише заготівками (саме тому наразі для них не додані ПМ). В майбутньому для кожного конкретного роз'єму на основі цих заготівок потрібно буде створити відповідний екземпляр, заповнивши додатково стовпчик «Ланцюг», в якому мають відображатись назви виводів роз'єму згідно його специфікації.

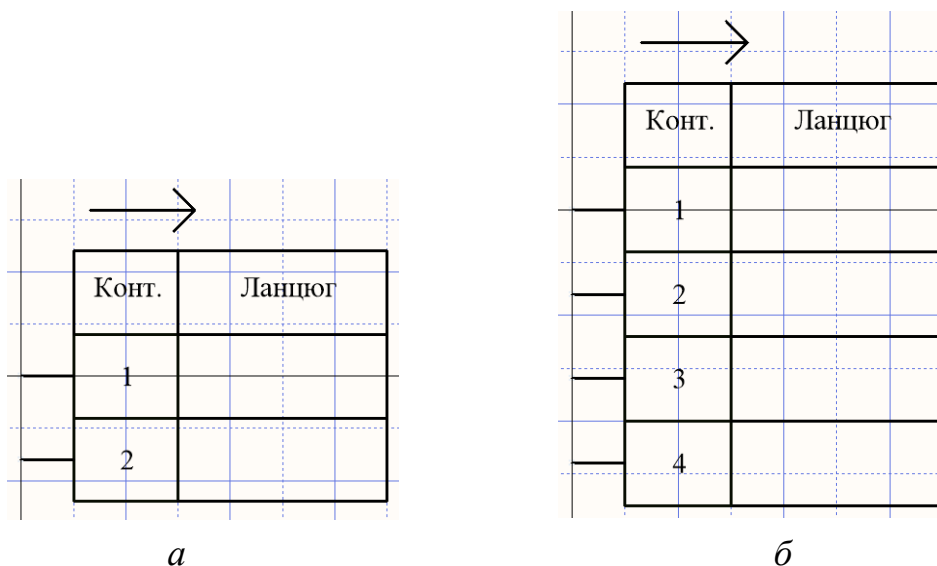


Рис. Л.2. УГП штирового роз'єму: *a* – на два контакти, *б* – на чотири контакти

Аналогічно¹⁵³ створіть УГП гніздових роз'ємів, задавши імена **Female Contact 2** та **Female Contact 4** (рис. Л.3).

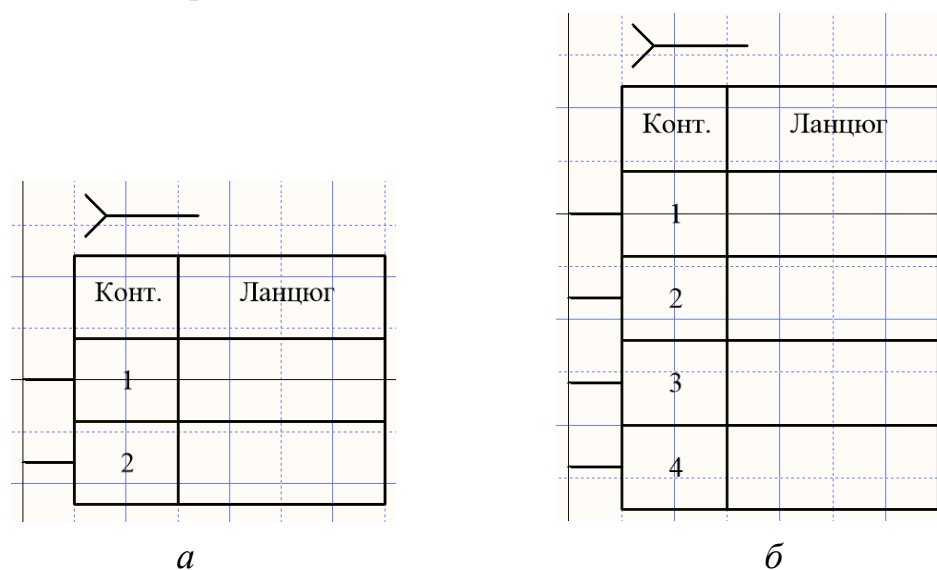


Рис. Л.3. УГП гніздового роз'єму: *a* – на два контакти, *б* – на чотири контакти

Додаткове завдання 2

Розробити компонентний модуль кварцевого резонатора у [корпусі HC49S](#) (рис. Л.4). Тривимірну модель корпусу створити, використовуючи виключно вбудовані графічні засоби **Altium Designer**.

¹⁵³ Зверніть увагу, що УГП гніздових роз'ємів фактично таке саме, як і штирових, за винятком символу над таблицею, який вказує тип роз'єму, та позиційного позначення.

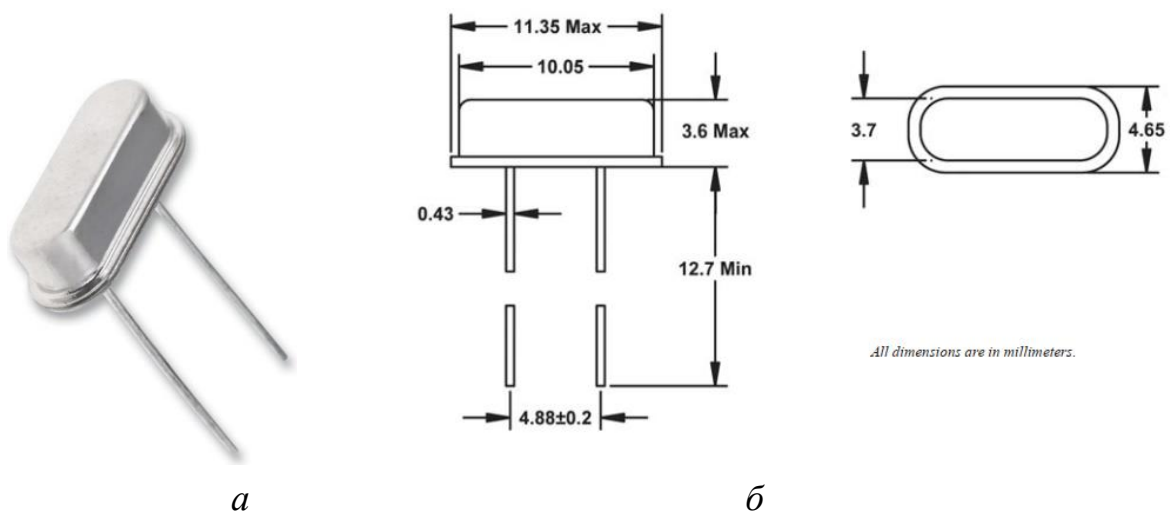


Рис. Л.4. Кварцовий резонатор: *a* – зовнішній вигляд, *б* – геометричні розміри корпусу

Розглянемо докладніше, як створити тривимірну модель корпусу кварцевого резонатору. Спочатку, орієнтуючись на розміри корпусу, що представлені на рис. Л.4б, «розкладемо» його на прості тривимірні об'єкти, які підтримує **Altium Designer**: екструдований прямокутник та циліндр (сферу у конкретно цьому випадку ми не будемо використовувати). Приклад розбиття на прості тривимірні об'єкти представлено на рис. Л.5. Шляхом об'єднання і суміщення цих об'ємних фігур можна отримати завершену складну модель корпусу.

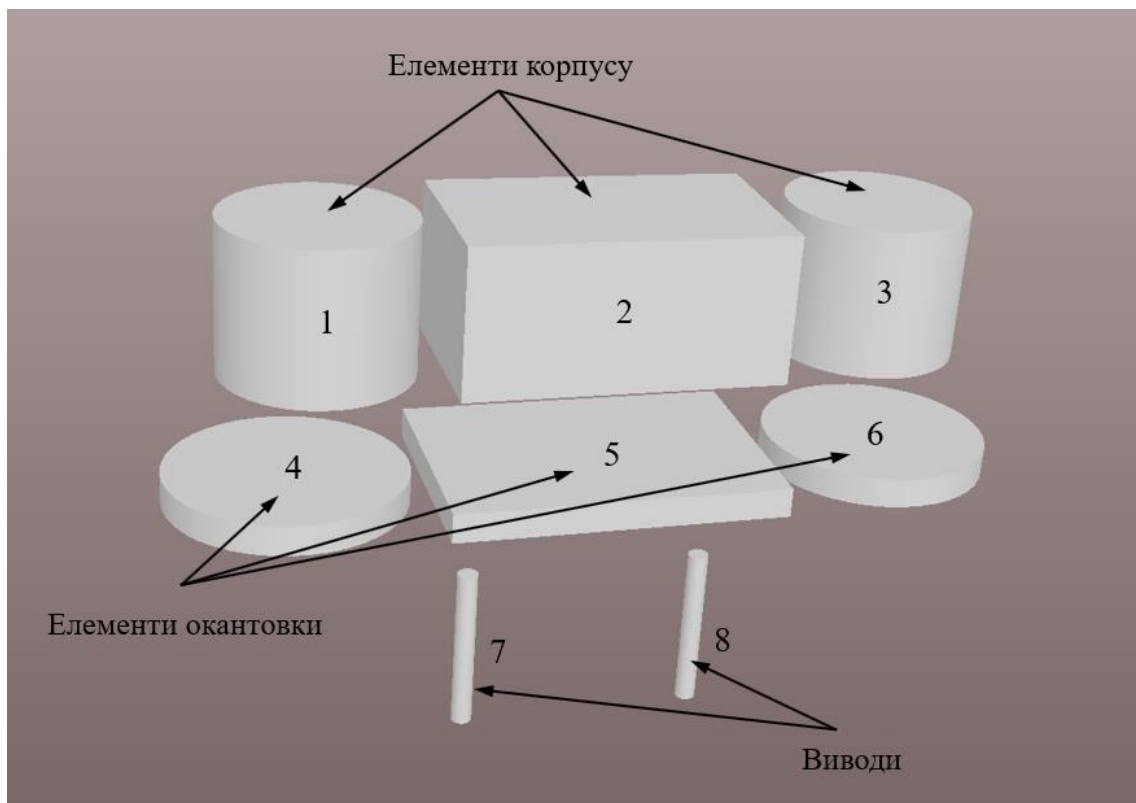


Рис. Л.5. Прості геометричні форми, з яких складається тривимірна модель корпусу

Розміри простих тривимірних об'єктів мають бути наступними (див. рис. Л.4б): 1, 3 – радіус 1,85 мм, висота 3 мм; 2 – довжина 6,35 мм, ширина 3,7 мм, висота 3 мм; 4, 6 – радіус 2,325 мм, висота 0,6 мм; 5 – довжина 6,7 мм, ширина 4,65 мм, висота 0,6 мм; 7, 8 – радіус 0,215 мм, висота 4 мм. Використовуючи вікно **3D Body** створіть усі прості об'ємні фігури¹⁵⁴, задавши світло-сірий колір. Зауважте, що висота деякого об'єкту у вікні **3D Body** вказується не безпосередньо, а розраховується як різниця між параметрами **Overall Height** та **Standoff Height**. Можете скористатись тими значеннями параметрів для відповідних об'єктів, які наведені на рис. Л.6.

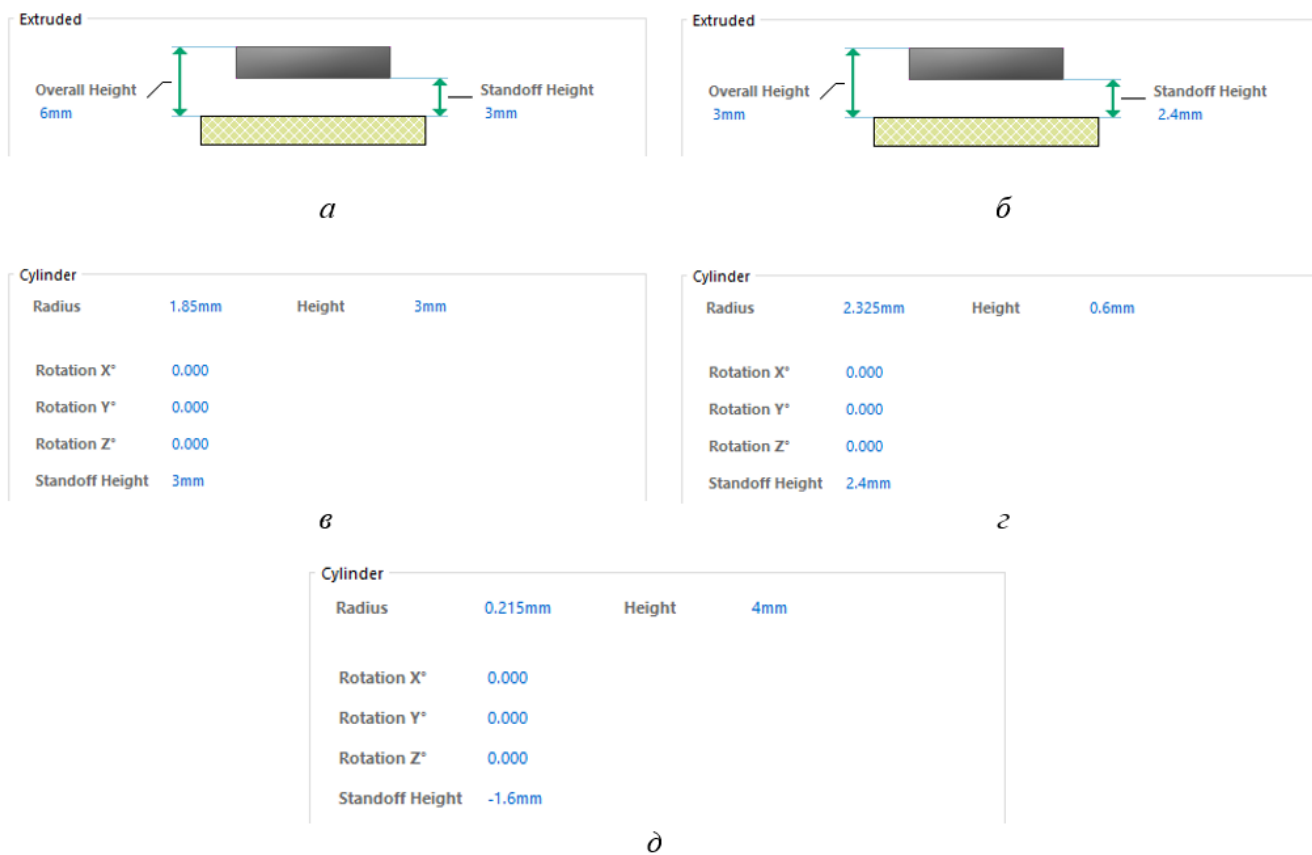


Рис. Л.6. Геометричні характеристики простих тривимірних об'єктів, які встановлюються у вікні **3D Body**: *а* – об'єкт 2, *б* – об'єкт 5, *в* – об'єкти 1 та 3, *з* – об'єкти 4 та 6, *д* – об'єкти 7 та 8

Після створення усіх об'єктів сумістіть їх таким чином, щоб отримати модель корпусу кварцевого резонатора, як показано на рис. Л.7. Спробуйте також самостійно підібрати кольорову гаму тривимірної моделі, яка б відповідала реальному кольору корпусу резонатора.

¹⁵⁴ Ви можете створити їх у будь-якій частині робочої області, після чого сумістити.

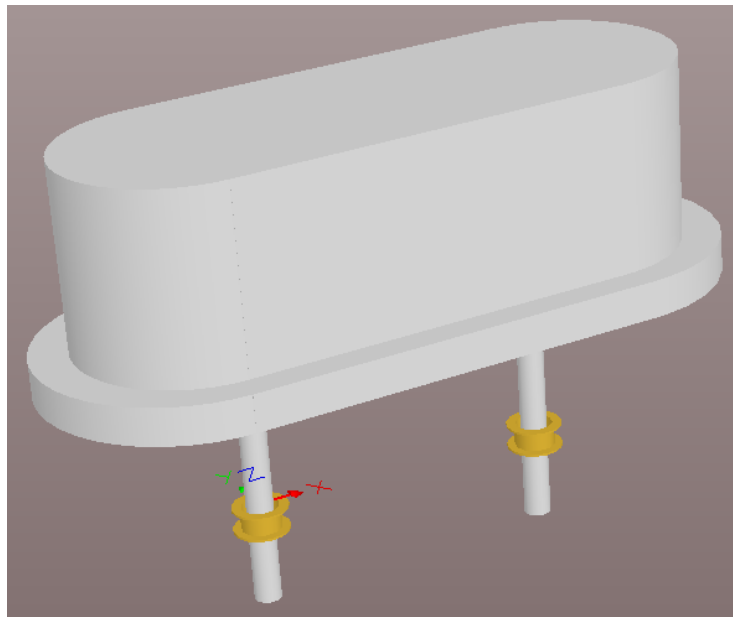


Рис. Л.7. Остаточний вигляд корпусу кварцевого резонатору, створений засобами **Altium Designer**

Додаткове завдання 3

Розробити компонентний модуль дроселя¹⁵⁵ у вигляді намотаного дроту діаметром 0,6 мм на циліндричному каркасі довжиною 12 мм, що має діаметр 3,4 мм. Причому відстань між отворами КМ посадкового місця дроселя становить 10 мм.

Варто зауважити, що дроселі зазвичай не є стандартними виробами (на відміну, наприклад, від резисторів та конденсаторів) і створюються під конкретні потреби, а їх геометричні розміри визначаються на основі розрахунків і залежать від таких параметрів, як діаметр дроту, кількість витків, значення індуктивності, форма осердя тощо (причому більшість із цих параметрів взаємозалежні). На рис. Л.8 наведені приклади безкорпусних дроселів для монтажу в отвори, у яких намотаний дріт не заховано всередині корпусу.

¹⁵⁵ Дросель – це гвинтова, спіральна або гвинтоспіральна котушка зі згорнутого ізольованого дроту, що володіє індуктивністю. Таким чином, дросель є ніщо інше, як різновид котушки індуктивності. Основною відмінністю дроселя від котушки індуктивності є те, що він володіє більшим опором змінному і малим опором постійному струму. Фактично, якщо котушку індуктивності включають у ланцюг для усунення (придушення) змінної складової струму, розділення або обмеження електричних сигналів різної частоти, то її називають дроселем. Однак найчастіше такої грані між дроселем і котушкою індуктивності не проводять, а ставлять знак рівності, тобто дросель = котушка індуктивності. Конструктивно дроселі зазвичай виконуються у вигляді намотаного дроту на каркасі у формі циліндра або тороїда (однак можуть і не містити каркас у випадку використання товстого дроту при малій кількості витків).

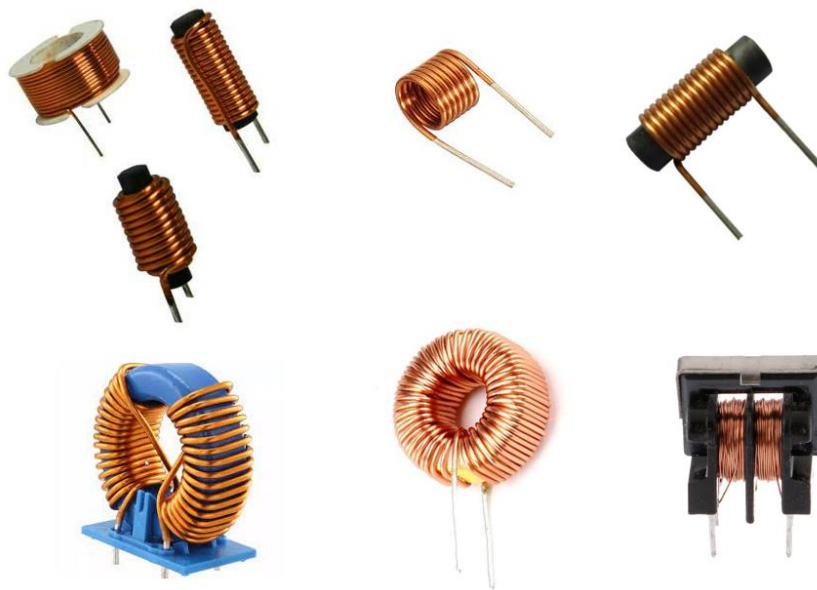


Рис. Л.8. Безкорпусні дроселі

Зверніть увагу, що УГП дроселя є аналогічним УГП котушки індуктивності. Оскільки під час виконання одного із комп'ютерних практикумів вже було створено УГП котушки індуктивності, тому можете зробити його копію і використати для нового компонентного модуля. Однак необхідно створити нове ПМ із двома КМ, що мають діаметр отворів 0,8 мм та розташовані на відстані 10 мм один від одного.

Далі ж докладніше розглянемо, як намалювати тривимірну модель дроселя в САПР **SolidWorks**.

Після запуску програми **SolidWorks** створіть новий проєкт деталі, для чого у меню **File** оберіть команду **New**. У вікні **New SOLIDWORKS Document**, що відкриється, оберіть пункт **Part**.

У новому пустому вікні у лівій бічній панелі на вкладці **FeatureManager Design Tree** клацніть ПКМ на рядку **Top Plane** і у спливаючому меню оберіть пункт **Sketch**. Після цього на панелі інструментів активуйте інструмент **Circle** та намалюйте коло радіусом 0,3 мм з центром в точці з координатами $X = -2$ мм, $Y = 0$ мм.

Далі побудуємо траєкторію закручування проводу, який намотується на осердя. Знову клацніть ПКМ на рядку **Top Plane** бокової панелі, у спливаючому меню оберіть пункт **Sketch**, а на панелі інструментів активуйте інструмент **Line**. Намалюйте лінію довжиною 9,2 мм, яка починається з центру координат та розташована вертикально (рис. Л.9).

Для того, щоб створити об'ємну фігуру на основі деякої траєкторії, необхідно відкрити панель **Features** та обрати команду **Swept Boss/Base**.

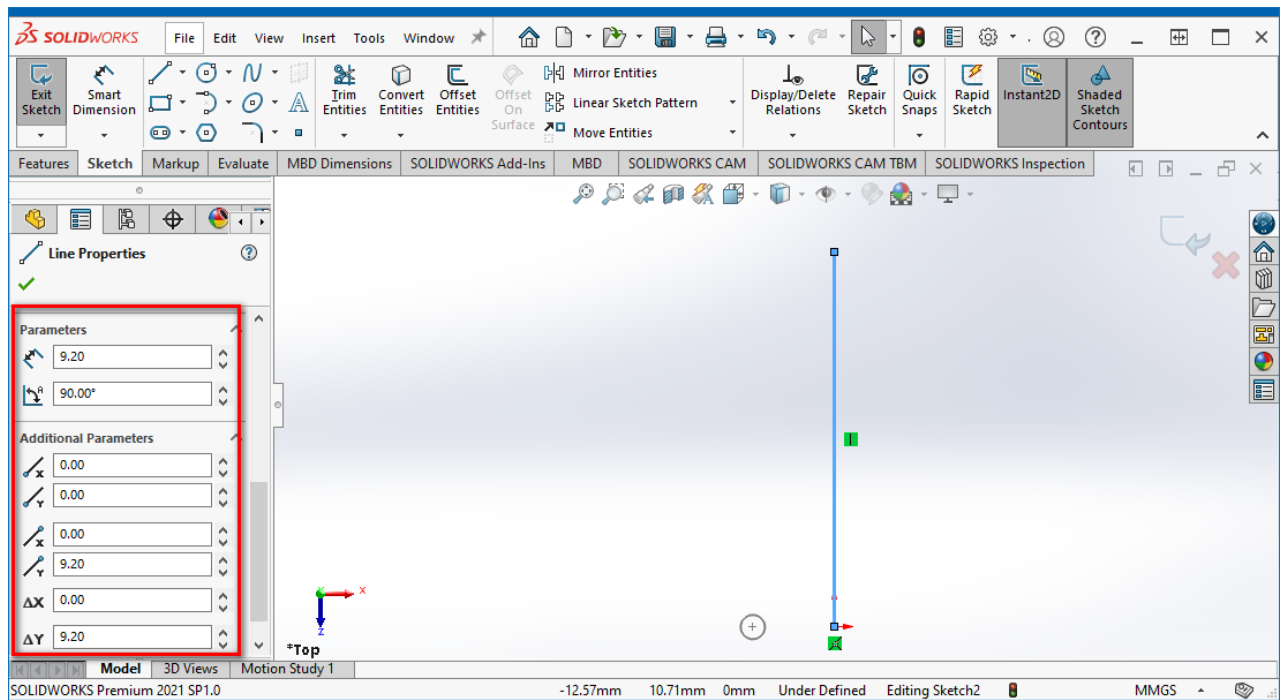


Рис. Л.9. Налаштування ескізу з прямою лінією

Після того, як була обрана команда *Sweep Boss/Base*, встановіть наступні параметри у *Менеджері властивостей* в області *Profile and Path*: як профіль оберіть перший ескіз із колом радіусом 0,3 мм, а як шлях – другий ескіз із прямою лінією. В області *Options* для параметра *Profile orientation* оберіть значення *Follow Path*, для *Profile Twist* – *Specify Twist Value*, для *Twist Control* – *Revolutions*, а для параметра *Direction 1* встановіть значення 9,5 мм (рис. Л.10).

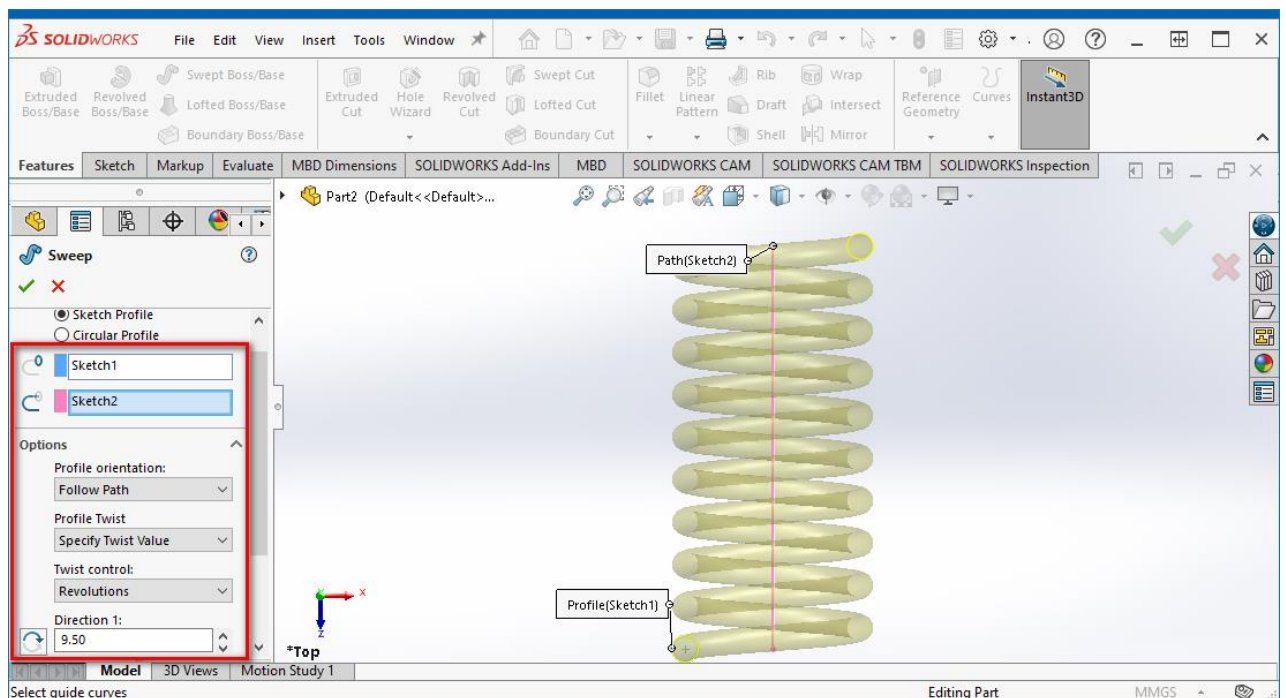


Рис. Л.10. Створення моделі обмотки дроселя

Тепер необхідно створити виводи дроселя, які будуть вставлятись у відповідні отвори на платі. Спочатку встановіть режим відображення деталі **Bottom**. Виділіть будь-яку основу (коло) створеної деталі, яка наразі нагадує пружину. Натисніть ПКМ та у спливаючому меню оберіть пункт **Sketch**¹⁵⁶. Далі затисніть клавішу клавіатури **<Shift>** та оберіть другу основу. Після цього на панелі інструментів у вкладці **Sketch** оберіть команду **Convert Entities**. Насамкінець активуйте вкладку **Features** на панелі інструментів, виконайте команду **Extruded Boss/Base** та у властивостях створюваної витягнутої фігури вкажіть довжину витягування 5 мм. Після виконання цих дій деталь має виглядати так, як це показано на рис. Л.11.

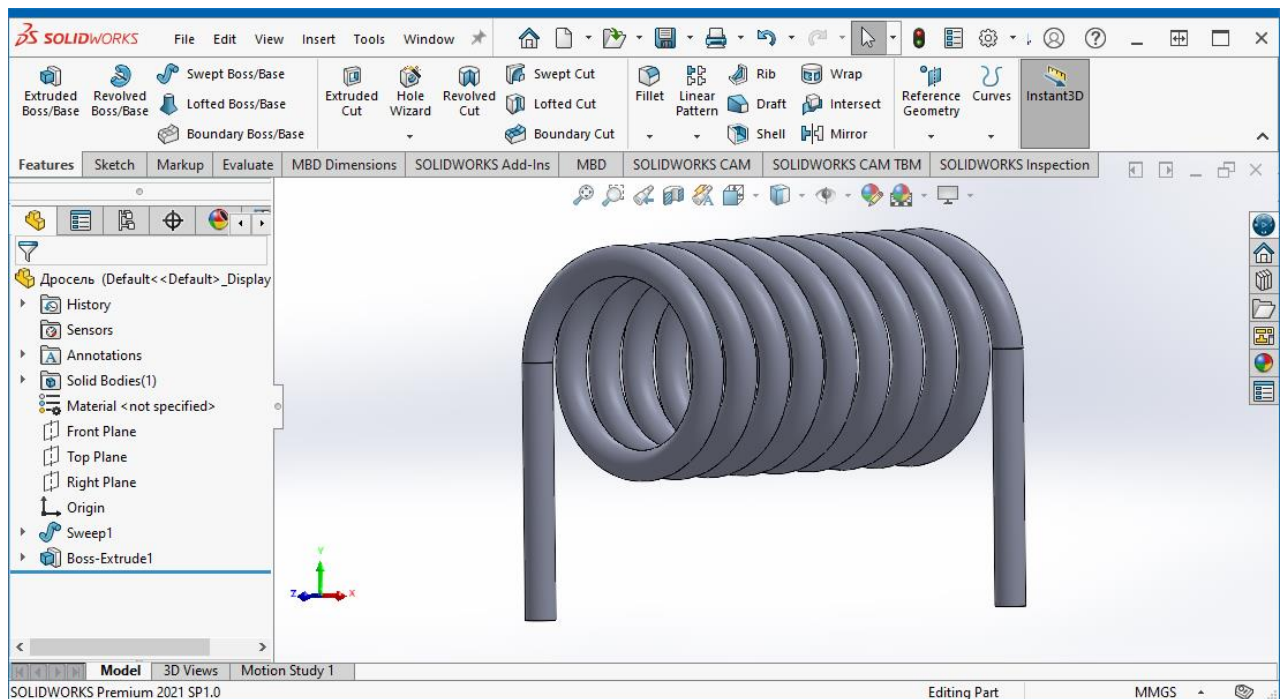


Рис. Л.11. Модель обмотки дроселя з виводами

Після того, як створена сама обмотка дроселя, побудуємо осердя у вигляді стрижня. На боковій панелі оберіть площину **Front Plane**, кладніть на ній ПКМ та у спливаючому меню активуйте пункт **Sketch**. Далі на панелі інструментів активуйте інструмент **Circle** та намалуйте коло радіусом 1,7 мм з центром в початку координат. Залишилось витягнути цей ескіз, для чого використовується команда **Extruded Boss/Base**. У властивостях витягування у напрямку 1 задайте довжину 1,4 мм, а у напрямку 2 – 10,6 мм.

¹⁵⁶ Зверніть увагу, що інколи внаслідок вибору не площини самої основи, а кола, на основі якого створювалась витягнута деталь, після натискання ПКМ з'являється пункт **Edit Sketch**. Однак це не те, що нам потрібно. Необхідно обрати саме площину основи, щоб у спливаючому меню з'явився пункт **Sketch**.

Після того, як осердя створене, зафарбуйте його у чорний колір, а саму обмотку – у коричневий (ближче до мідного відтінку). Кінцевий результат представлено на рис. Л.12.

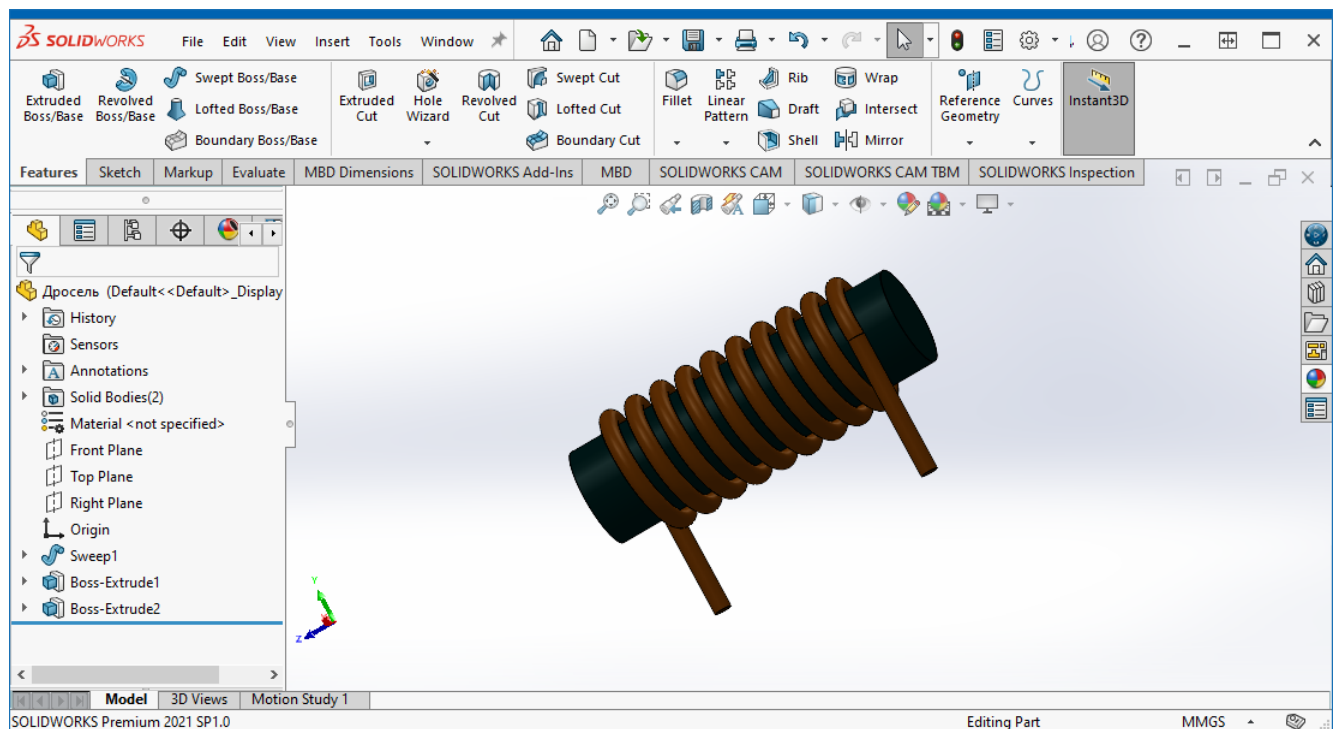


Рис. Л.12. Тривимірна модель дроселя

Збережіть тривимірну модель дроселя у форматі **STEP** та підключіть її до відповідного ПМ в **Altium Designer**.

Додаткове завдання 4

Розробити компонентний модуль міні-USB з'єднувача [Molex 56579-0576](#) (рис. Л.13).

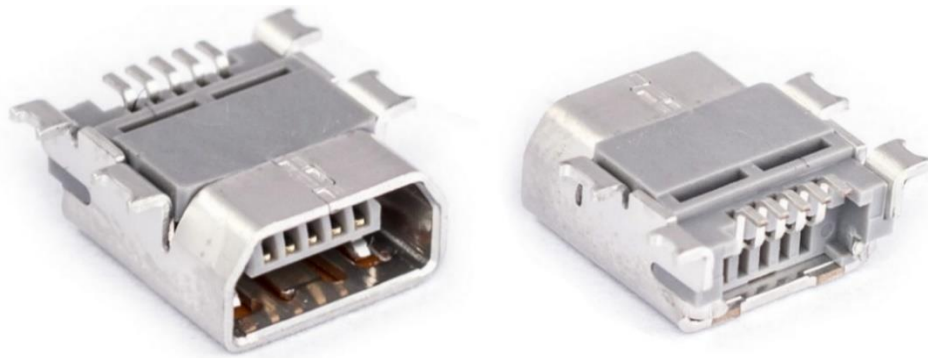


Рис. Л.13. Міні-USB з'єднувач Molex 56579-0576

За класифікацією цей роз'єм є розеткою, що містить 5 гніздових контактних елементів. Враховуючи це, розробіть УГП з'єднувача Molex 56579-0576.

Для розробки ПМ можете скористатись або [цією](#), або [цією](#) технічною документацією. Зверніть увагу, що це ПМ є складним (має багато КМ різного розміру та орієнтації), причому створювати його доведеться вручну.

Врахуйте, що в технічній документації на кресленику роз'єму вказана цифрова нумерація гніздових контактів, тоді як на ПМ контактні майданчики позначаються іменами: VBUS, D-, D+, ID, CND (рис. Л.14). Чи використовувати цифрову нумерацію КМ, чи задати імена (це робиться у властивостях КМ в полі **Designator**), вирішує розробник¹⁵⁷. Проте у цьому випадку краще задайте імена КМ, що забезпечить ліпший візуальний контроль правильності підключення відповідних трас (сигналів) на платі. Відповідність же нумерації виводів УГП та імен КМ ви можете встановити під час додавання ПМ до символу компонента у спеціальному вікні **Model Map**.

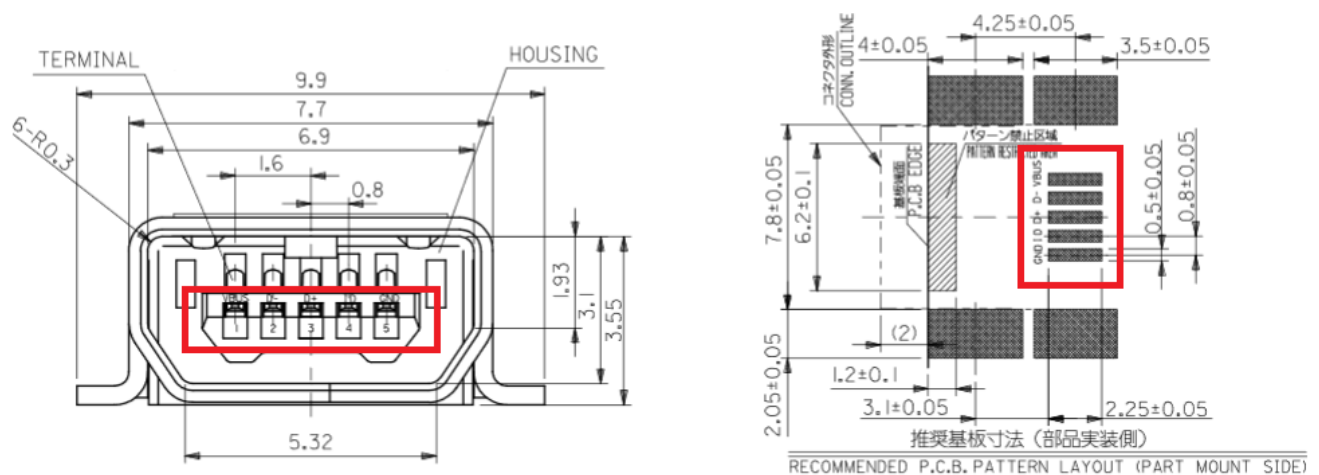


Рис. Л.14. Нумерація виводів та найменування КМ з'єднувача Molex 56579-0576

Корпус цього з'єднувача містить чотири бокові виводи (не сигнальні), за допомогою яких він припаюється до плати, що збільшує міцність утримування. Для припаювання цих виводів у будь-якому випадку потрібно створити планарні КМ. Оскільки роль цих КМ полягатиме тільки у припаюванні до них не сигнальних виводів, то вони можуть бути не підключеними ні до яких трас на платі. З іншого боку, враховуючи, що корпус з'єднувача є металевим, то його можна підключити

¹⁵⁷ Зокрема, у всіх комп'ютерних практикумах цього навчального посібника ми використовували виключно цифрову нумерацію.

до провідника землі. Якщо ви використовували імена КМ, як було вказано у попередньому абзаці, то у властивостях чотирьох КМ для кріплення потрібно в полі **Designator** вказати **GND**¹⁵⁸ (якщо ж ви використовували нумерацію, то потрібно вказати номер КМ, який відповідає землі).

Також зверніть увагу, що у рекомендованій в технічній документації моделі ПМ є зона, яка називається **Pattern Restricted Area** (див. рис. Л.14). Це означає, що в ній корпус з'єднувача може мати безпосередній контакт із платою (тоді як усі інші частини корпусу, звісно, окрім виводів, не дотикаються до поверхні). В такому випадку необхідно встановити заборону на розміщення в цій зоні будь-яких компонентів. І хоча теоретично топологія (траси) може бути в цій зоні, оскільки вона буде покрита захисною маскою, проте краще тут її заборонити (на всякий випадок, адже захисний шар може й пошкодитись, що призведе до проблем в майбутньому). Для заборони розміщення компонентів та топології необхідно на шарі **Top Layer** створити прямокутну область **Solid Region** із властивістю **KeepOut** (меню **Place** → **KeepOut** → **Solid Region**).

Після створення ПМ з'єднувача Molex 56579-0576, [завантажте](#) та підключіть тривимірну модель. Насамкінець ПМ із тривимірною моделлю додайте до розробленого УГП, щоб створити завершений компонентний модуль.

Додаткове завдання 5

Розробити компонентний модуль драйвера транзистора [TC4421](#), що виготовляється в корпусі ТО-220 (рис. Л.15). Тривимірну модель корпусу знайти на відповідних інтернет-ресурсах, після чого підключити до компонентного модуля.

¹⁵⁸ Або ж можна зробити дещо інакше, враховуючи той факт, що усі чотири бокових виводи об'єднані електрично уже в самому з'єднувачі. Тобто, якщо підключити тільки один до траси землі на платі, усі інші також електрично з'єднуються із землею. Для того, щоб забезпечити цю можливість, необхідно у властивостях кожного із цих чотирьох КМ встановити параметр **Jumper ID** у однакове значення, але відмінне від нуля (наприклад, можна задати 1). Таким чином ми вкажемо Редактору плат, що ці чотири КМ належать до одного фізичного внутрішнього з'єднання в роз'ємі. До речі, за замовчуванням для усіх нових КМ параметр **Jumper ID** = 0, що означає, що цей КМ не має електричних з'єднань з іншими КМ. При будь-яких інших значеннях, відмінних від нуля, КМ із однаковими **Jumper ID** вважаються фізично з'єднаним всередині електронного компонента.

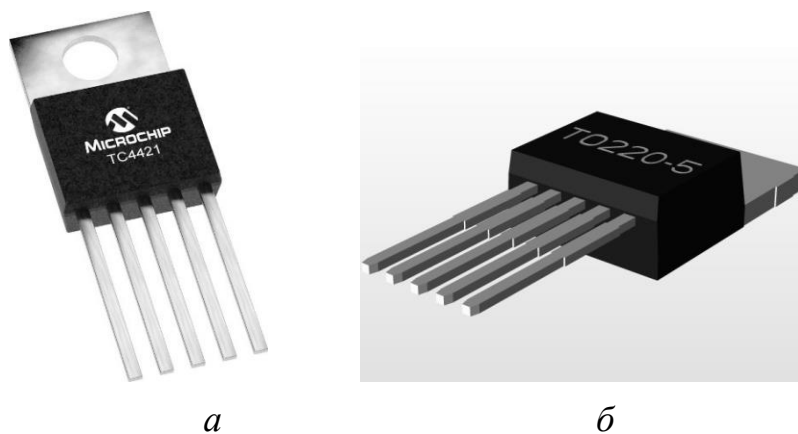


Рис. Л.15. Транзистор TC4421: *а* – реальне зображення, *б* – тривимірна модель корпусу

Додаткове завдання 6

Розробити принципову схему та друковану плату ГТІ, що представлена на рис. Л.16. Частота цього генератора визначається за формулою $f = 0,5 \cdot R1 \cdot C1$.

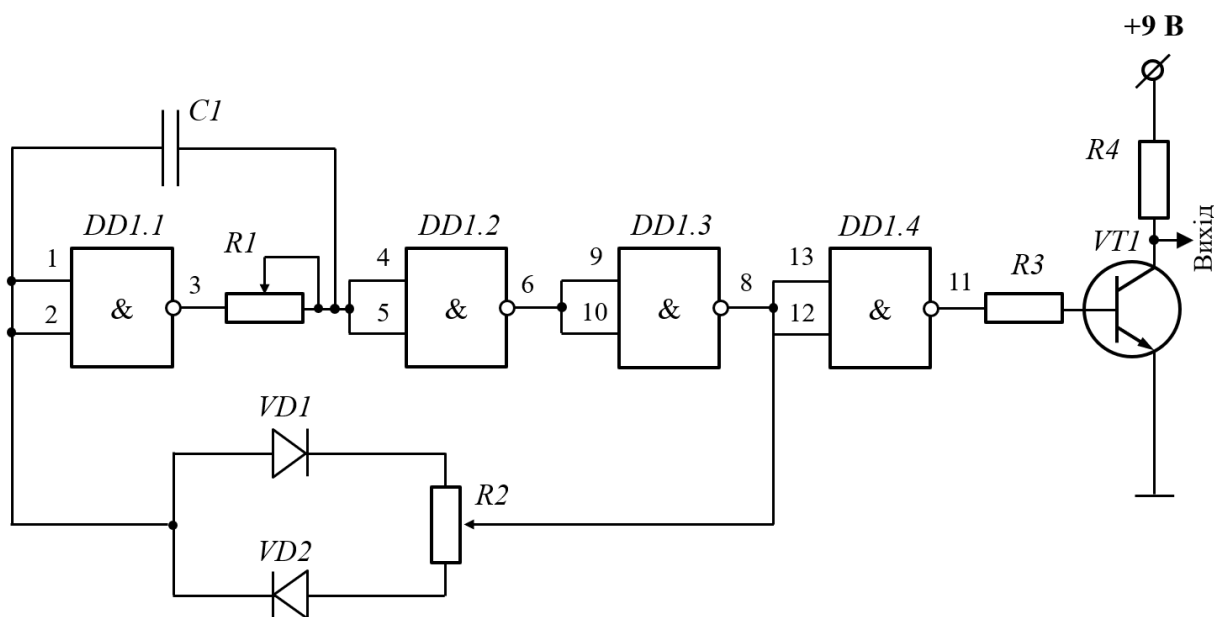


Рис. Л.16. Схема ГТІ

Схема ГТІ містить наступний перелік компонентів:

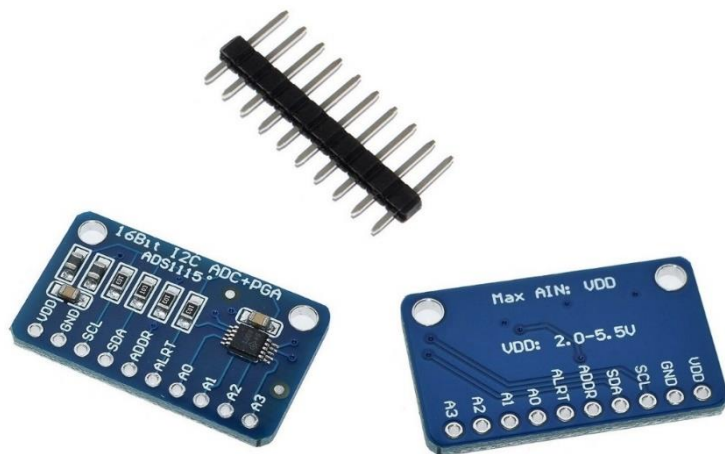
1. Мікросхема *DD1* – SN7400 (основа генератора).
2. Транзистор *VT1* на виході ГТІ – BFP719 (призначений для підсилення імпульсів за потужністю).
3. Діоди *VD1* та *VD2* – КД103Б.
4. Постійні резистори: *R3* – 2 кОм, *R4* – 1 кОм.

5. Змінні резистори: $R1$ – 1 кОм (використовується для налаштування частоти),
 $R2$ – 1 кОм (використовується для налаштування шпаруватості імпульсів).
6. Ємність $C1$ – 50 нФ.

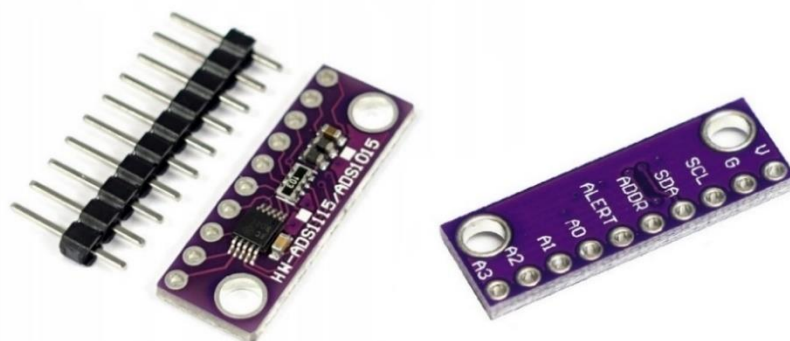
Після створення схеми спробуйте виконати аналіз сигналів за допомогою засобів **Altium Designer**. Для аналізу необхідно, щоб компонентні модулі мали SPICE- та/або IBIS-моделі. Тому для моделювання доведеться спочатку знайти відповідні аналоги електронних компонентів розроблюваної принципової схеми, які мали б такі моделі, після чого запозичені моделі підключаються до відповідних компонентних модулів.

Додаткове завдання 7

Розробити друковану плату модуля (рис. Л.17) аналого-цифрового перетворення на основі мікросхеми [ADS1115](#).



а



б

Рис. Л.17. Приклади плат модуля аналого-цифрового перетворення:

а – варіант 1, *б* – варіант 2

В інтернеті можна знайти принципові схеми цього модуля (рис. Л.18), зокрема на сайтах, на яких він продається, або сайтах, де пояснюються принципи його роботи.

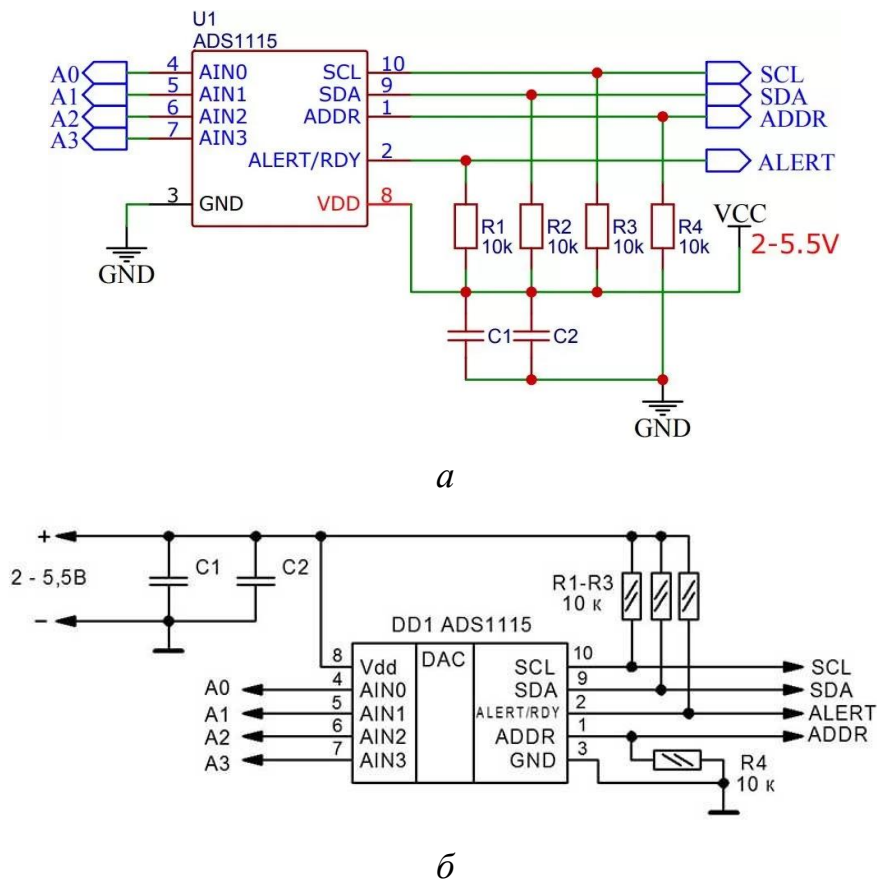


Рис. Л.18. Приклад принципових схем, які містять не всі електронні компоненти, що є на платі модуля аналого-цифрового перетворення: *a* – варіант 1, *б* – варіант 2

І хоча ці схеми загалом правильні, проте вони не повною мірою відображають оригінальну версію принципової схеми. Наприклад, якщо ви порівняєте рис. Л.17 та рис. Л.18, то помітите, що на платах розміщено більше електронних компонентів, ніж є на схемах (рис. Л.19).

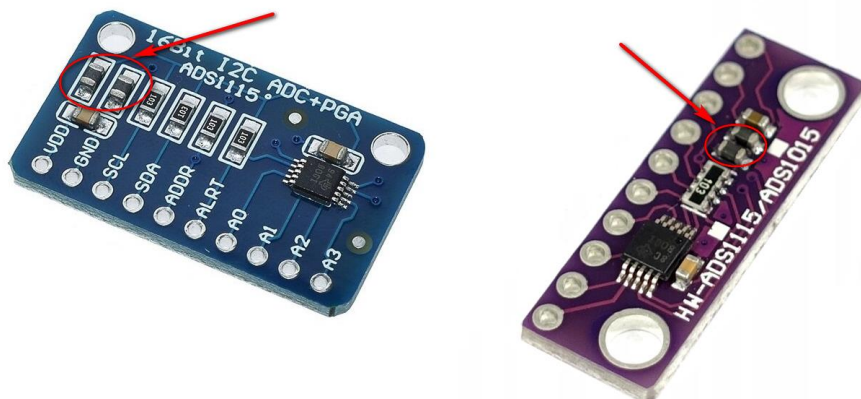


Рис. Л.19. Електронні компоненти, що є на платах, але відсутні на деяких схемах

Насправді електричні схеми, що представлені на рис. Л.18 є робочими, проте вони містять деякі спрощення. Щоб зрозуміти, які саме спрощення містять розглянуті схеми, необхідно знайти оригінал. [Оригінальну версію принципової схеми](#) (рис. Л.20) модуля аналого-цифрового перетворення можна знайти на сайті www.adafruit.com.

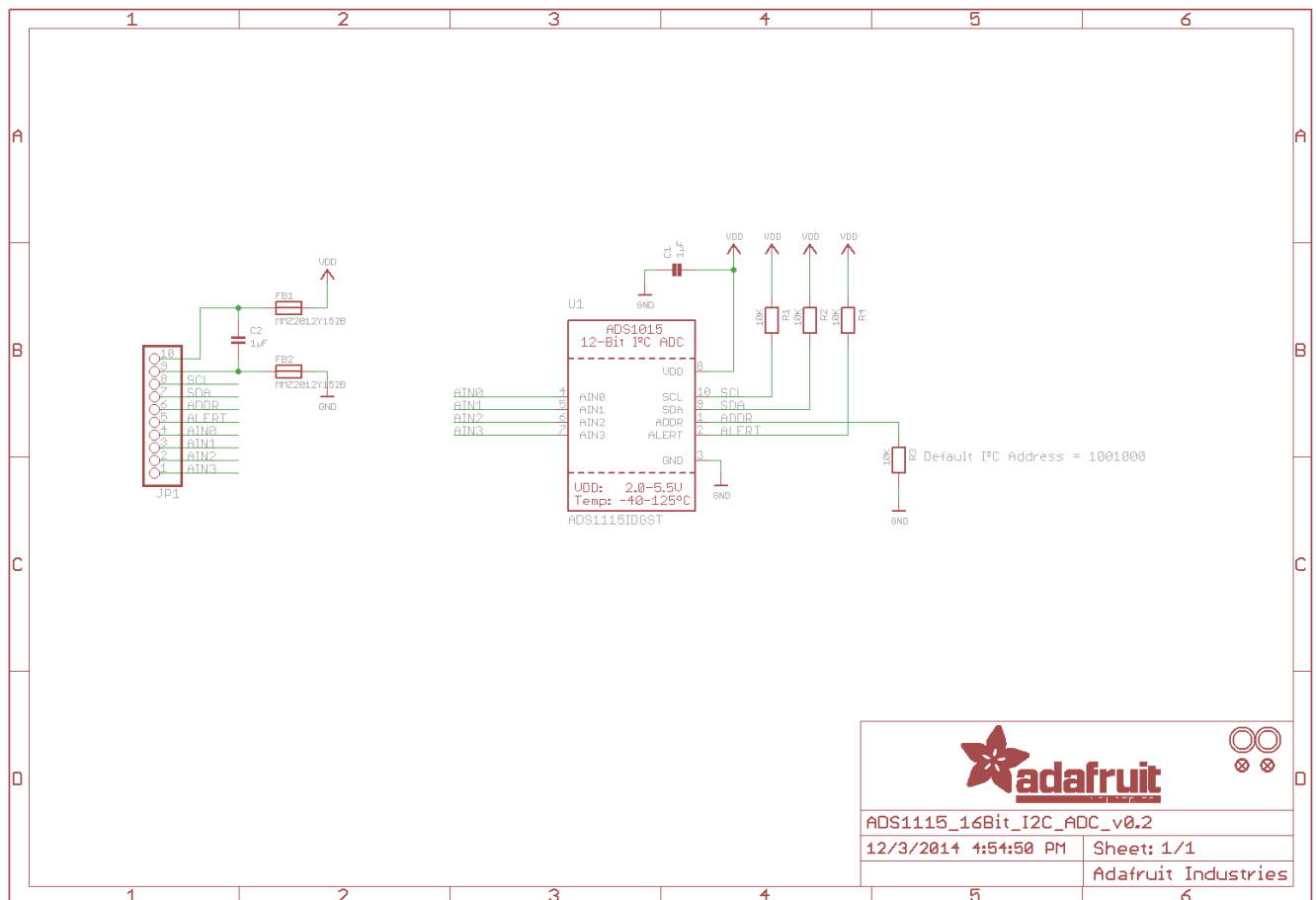


Рис. Л.20. Оригінальна принципова схема модуля на основі АЦП ADS1115

Як бачимо, оригінальна версія схеми містить два електронні компоненти [MMZ2012Y152B](#) (саме їх не вистачає на спрощених схемах, але вони є на платах) на лініях живлення та землі. Ці компоненти називаються феритовими намистинками або феритовими кульками (від англ. ferrite beads). Фактично, вони представляють собою провід, що охоплений феритом. [Оскільки феритові намистинки мають частотнозалежний опір, то вони використовуються для придушення завад](#). Тому часто їх ще називають феритовими фільтрами.

З одного боку, феритові фільтри не є обов'язковими (без них модуль також працюватиме). Але все ж вони є важливими електронними компонентами, оскільки суттєво зменшують шуми на лініях живлення і землі.

Оскільки феритові намистинки виконують роль, що аналогічна ролі котушок індуктивності, то деякі розробники створюють їх УГП ідентично до УГП котушок індуктивності. Однак існують й інші варіанти, які представлені на рис. Л.21.

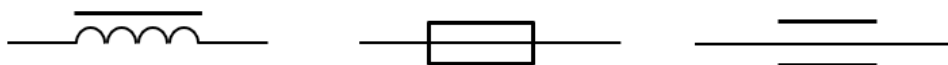


Рис. Л.21. Різні варіанти УГП феритових намистинок

Таким чином, під час роботи над цим завданням, ви повинні зрозуміти, що не завжди знайдені в інтернеті схеми відображають повну структуру розроблюваного електронного пристрою. Деякі можуть містити спрощення. Тому, якщо ви хочете взяти за основу той чи інший проєкт, то завжди намагайтесь шукати оригінальні файли проєкту.

Наразі ж, орієнтуючись на принципову схему, що представлена на рис. Л.20¹⁵⁹ (або ж її варіацію, яку можна переглянути [тут](#)) та зовнішній вигляд плати, що представлений на рис. Л.17, розробіть власний проєкт модуля аналого-цифрового перетворення на основі мікросхеми ADS1115.

Додаткове завдання 8

Спробуйте самостійно зробити проєкти, які представлені у відео на YouTube «[Незвичайний стабілітрон, Магнітний підсилювач, Ключ на діоді, Тиристор-генератор, Подвоювач напруги](#)».

Додаткове завдання 9

Створити принципову схему та плату ШІМ регулятора на мікросхемі таймера NE555 (рис. Л.22).

Про схему ШІМ регулятора та реалізацію плати на її основі можете подивитись у наступних відео на YouTube: «[Що таке ШІМ? Як ШІМ регулює яскравість, температуру, оберти двигуна та напругу? Розбираємось!](#)» та «[How To Make a PWM DC Motor Speed Controller using the 555 Timer IC](#)».

¹⁵⁹ До речі, під час розробки власної принципової схеми в **Altium Designer** ви можете скористатись оформленням схем, що представлені на рис. Л.18. Просто не забудьте створити та додати до бібліотеки електронні компоненти, яких не вистачає.

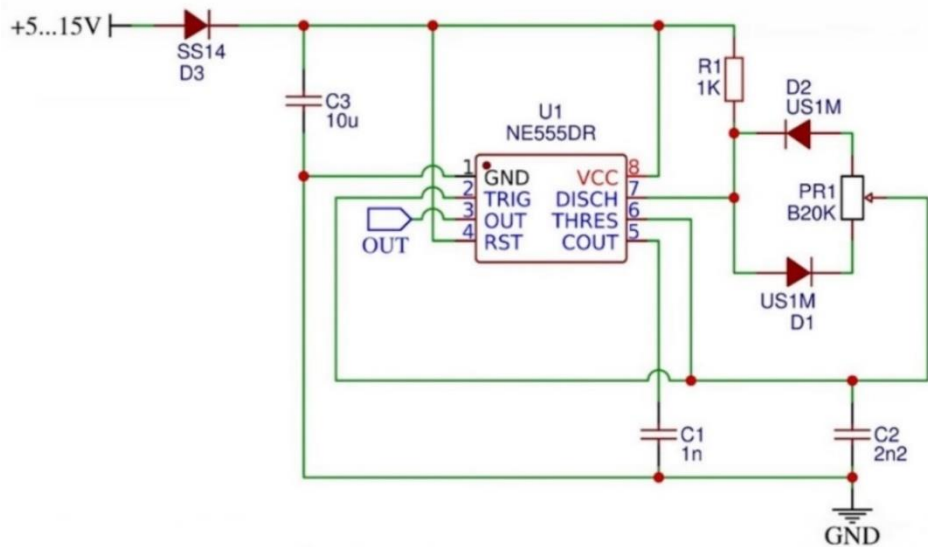


Рис. Л.22. Схема ШІМ регулятора

Додаткове завдання 10

Розробити плату зарядного пристрою для телефону на основі принципової схеми, що представлена на рис. Л.23.

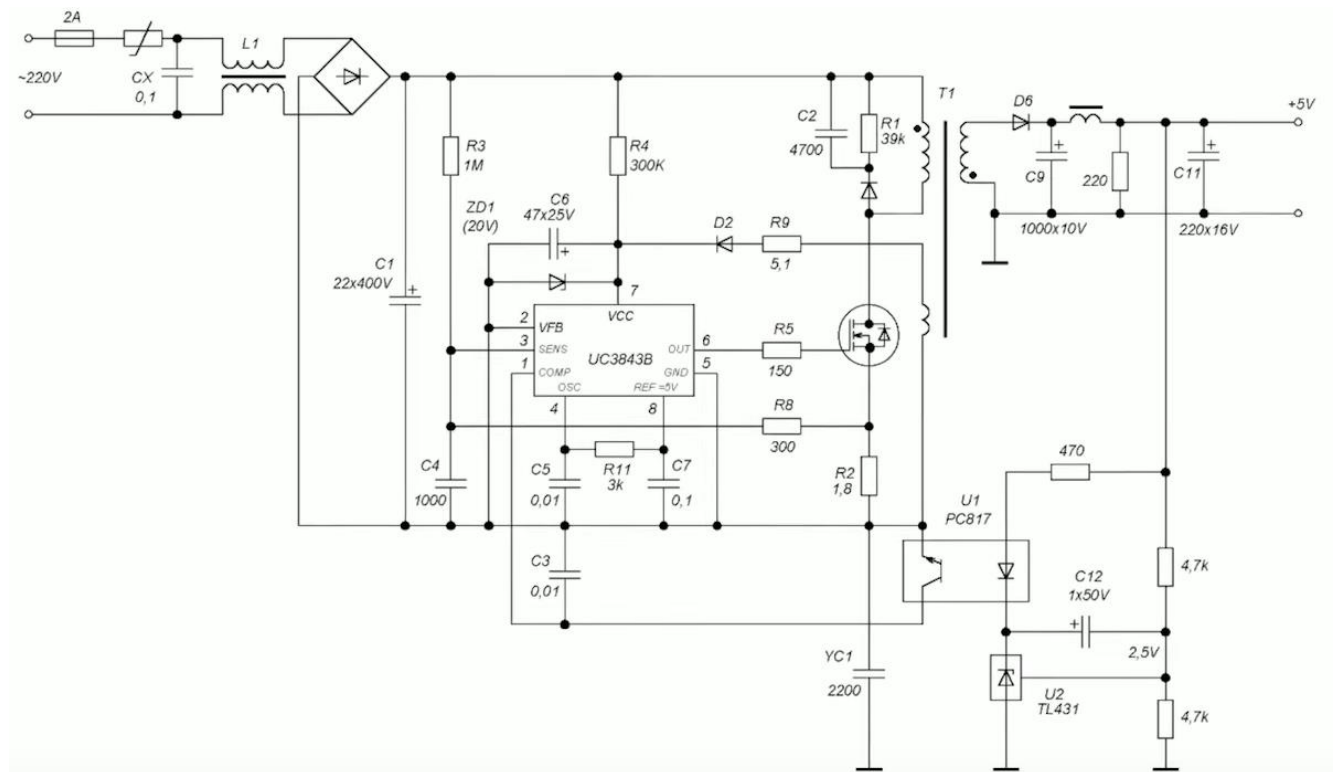


Рис. Л.23. Схема зарядного пристрою для телефону

Для того, щоб краще зрозуміти, яким чином працює плата зарядного пристрою для телефону (на прикладі наведеної вище схеми), подивіться наступні відео на

YouTube: «[Як працює зарядний пристрій телефону? Оберненоходовий блок живлення. Найбільш зрозуміле пояснення](#)» та «[How does a mobile charger work? SMPS with Opto-Coupler](#)».

Додаткове завдання 11

Створити принципову схему та плату FM приймача, орієнтуючись на проєкт [AK-270 HEX3653 FM Radio Receiver Module](#). Принципова схема представлена на рис. Л.24, зовнішній вигляд плати – на рис. Л.25, завершений виріб – на рис. Л.26.

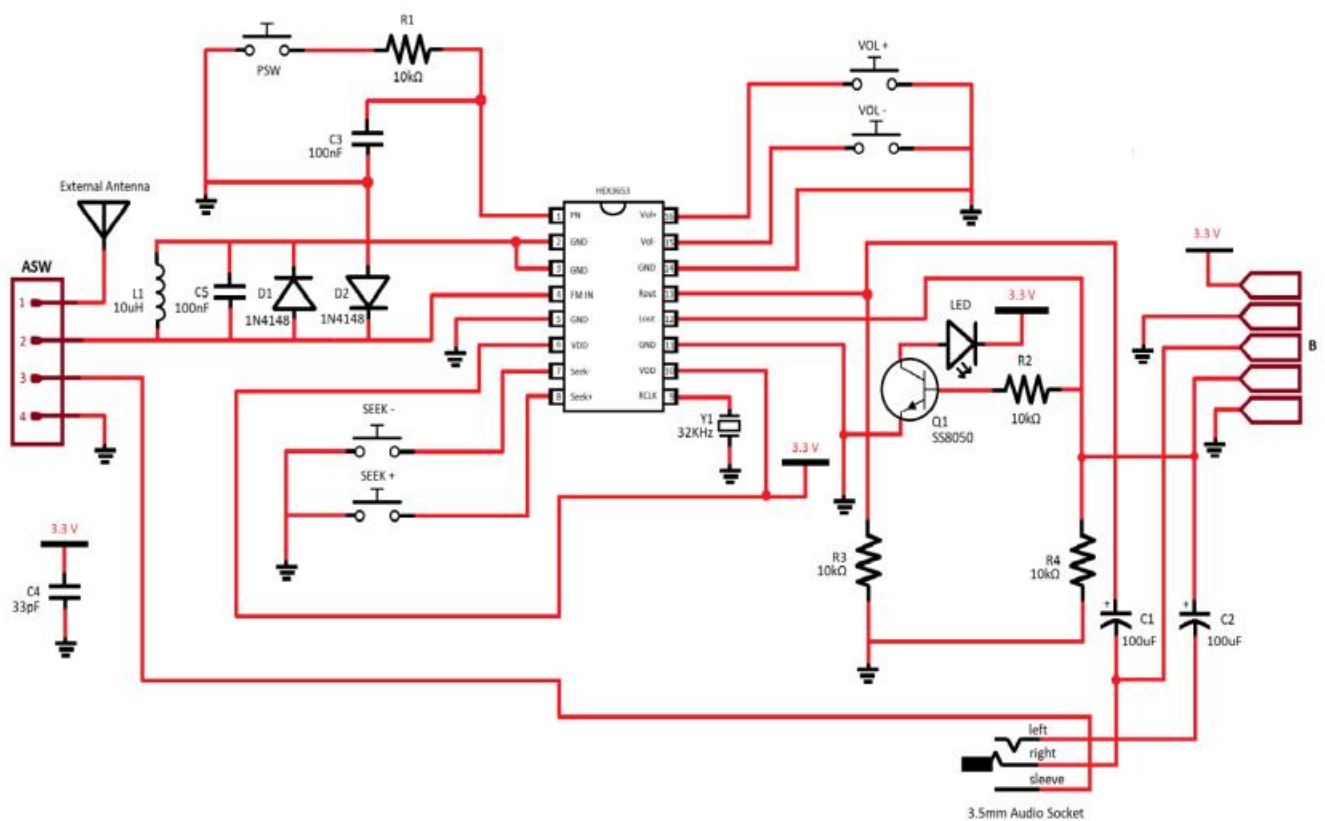
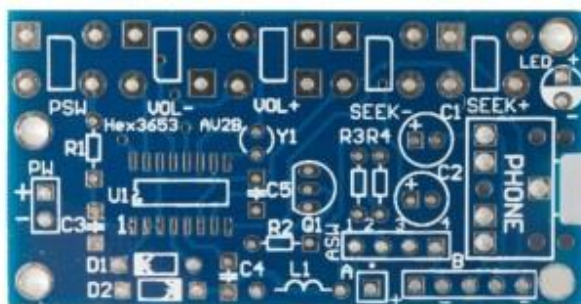


Рис. Л.24. Принципова схема FM приймача



Top Side



Bottom Side

Рис. Л.25. Верхня та нижня сторони плати FM приймача



Рис. Л.26. Зовнішній вигляд FM приймача: зверху – плата, знизу – тримач для двох батарейок (акумуляторів) типу АА

Додаткове завдання 12

Реалізувати власну плату на основі принципової схеми осцилографа [DSO150](#). Принципова схема складається з двох частин: [основної плати](#) та [аналогової плати](#). Розробити власну тривимірну модель корпусу осцилографа. Опис осцилографа DSO150 можна знайти [тут](#).