

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та безпеки

(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»
ПРИСТРІЙ ДЛЯ ТЕСТУВАННЯ КАБЕЛІВ НА ОСНОВІ
ПЛАТФОРМИ ARDUINO
ARDUINO-BASED CABLE TESTING DEVICE

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІ-42
Кіштя Діана Миколаївна

(підпис)

Керівник:
к.т.н., доцент
Гринюк Сергій Васильович

(підпис)

Кваліфікаційну роботу
допущено до захисту
« 06 » червня 2025 р.

Гарант освітньої програми:

к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Тарас ТЕРЛЕЦЬКИЙ

« 10 » 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Кішті Діані Миколаївні

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Пристрій для тестування кабелів на основі платформи Arduino

Керівник роботи к.т.н., доц. Гринюк Сергій Васильович

затверджені наказом закладу вищої освіти від «04» січня 2025 року № 11/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 10.06.2025р.

3. Вихідні дані до роботи джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування.

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

Вступ

Огляд літератури із досліджуваної теми

Проектування Usb-тестера

Практична реалізація пристрою

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Огляд літератури із досліджуваної теми</i>	<i>Гринюк С.В., доцент</i>		
<i>Проектування Usb-тестера</i>	<i>Гринюк С.В., доцент</i>		
<i>Практична реалізація пристрою</i>	<i>Гринюк С.В., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>		___%	
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст.викладач</i>		

7. Дата видачі завдання 10.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури із досліджуваної теми</i>	до 10.02.2025 р.	Виконано
2.	<i>Проектування Usb-тестера</i>	до 02.03.2025 р.	Виконано
3.	<i>Практична реалізація пристрою</i>	до 02.04.2025 р.	Виконано
4.	<i>Висновки та пропозиції</i>	до 10.04.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 15.04.2025 р.	Виконано
6.	<i>Формування додатків</i>	до 02.05.2025 р.	Виконано
7.	<i>Оформлення ілюстративного матеріалу</i>	до 10.05.2025 р.	Виконано
8.	<i>Представлення остаточного варіанту кваліфікаційної роботи керівникові</i>	до 15.05.2025 р.	Виконано
9.	<i>Нормоконтроль</i>	до 30.05.2025 р.	Виконано
10.	<i>Інструментальна перевірка на академічний плагіат</i>	до 03.06.2025 р.	Виконано
11.	<i>Здача кваліфікаційної роботи та всіх супровідних документів на кафедру</i>	до 10.06.2025 р.	Виконано

Здобувач вищої освіти

(підпис)

Кішта Д.М.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Гринюк С.В.

(прізвище, ініціали)

АНОТАЦІЯ

Кішта Д. М. Пристрій для тестування кабелів на основі платформи Arduino. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, переліку використаних джерел та додатків.

Перший розділ присвячений огляду сучасних пристроїв тестування кабелів, аналізу їхніх технічних характеристик, принципам трасування з'єднань у слаботочних мережах, а також огляду USB-інтерфейсів і типів сигналів. Проведено порівняння існуючих рішень, визначено їх переваги та недоліки, що сформувало технічні вимоги до власної розробки.

Другий розділ містить обґрунтування вибору апаратних компонентів, розробку структурної та принципової електричної схеми, проектування 3D-корпусу пристрою та друкованої плати для трасування кабелів. Представлено схематичні креслення, макети й обґрунтування конструктивних рішень, необхідних для реалізації поставлених функцій.

Третій розділ описує процес монтажу, складання та налаштування системи. Подано програмне забезпечення для Arduino, алгоритми роботи пристрою, проведено тестування різних типів кабелів. Результати експериментів дозволили оцінити ефективність розробки, її сильні сторони й потенційні напрямки вдосконалення. Ключові слова: онячний трекер, Arduino, мікроконтролер, інтелектуальна система, серводвигун, сенсор, автоматизація.

Ключові слова: USB-кабель, тестер, Arduino Mega, TFT-сенсорний дисплей, мікроконтролер, тестування.

ANNOTATION

Kishta D. Arduino-based cable testing device. Manuscript.

Qualification work of the bachelor of the specialty «Computer Engineering» of the specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

The qualification work consists of an introduction, three chapters, conclusions, a list of sources used and appendices.

The first chapter is devoted to a review of modern cable testing devices, an analysis of their technical characteristics, the principles of connection tracing in weak networks, as well as a review of USB interfaces and signal types. A comparison of existing solutions was carried out, their advantages and disadvantages were determined, which formed the technical requirements for our own development.

The second chapter contains a justification for the choice of hardware components, the development of a structural and schematic electrical diagram, the design of a 3D device case and a printed circuit board for cable tracing. Schematic drawings, layouts and justification of design solutions are presented, most of them for the implementation of the functions set.

The third section describes the process of installation, assembly and configuration of the system. The software for Arduino, the device's operating algorithms are presented, and various types of cables are tested. The results of the experiments allowed us to assess the effectiveness of the development, its strengths and show areas for improvement. Keywords: onyachny tracker, Arduino, microcontroller, intelligent system, servo motor, sensor, automation.

Keywords: USB cable, tester, Arduino Mega, TFT touch display, microcontroller, testing.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Аналіз пристроїв тестування кабелів	6
1.2 Принципи трасування з'єднань у слаботочних мережах	12
1.3 Огляд інтерфейсів USB та типів сигналів.....	15
РОЗДІЛ2 ПРОЄКТУВАННЯ USB-ТЕСТЕРА	18
2.1 Вибір апаратних компонентів пристрою	18
2.2 Розробка структурної та принципової електричної схеми	24
2.3 Проектування 3D-елементів пристрою	25
2.4 Проектування плати для кабельного трасування	27
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРИСТРОЮ	30
3.1 Монтаж пристрою	30
3.2 Програмування Arduino	33
3.3 Тестування пристрою	36
ВИСНОВКИ	40
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	42
ДОДАТКИ	44

ВСТУП

У сучасному світі інформаційних технологій кабельні з'єднання відіграють ключову роль у забезпеченні ефективного обміну даними, передавання електроенергії та функціонування численних пристроїв. Особливо це стосується USB-кабелів, які широко використовуються для підключення периферійних пристроїв, заряджання гаджетів та передачі інформації. Зростання кількості стандартів, таких як USB 2.0, USB 3.0, USB Type-C, а також постійне оновлення протоколів, створюють нові виклики щодо контролю якості та надійності кабелів.

Недостатньо якісні або пошкоджені кабелі можуть призводити до значних втрат продуктивності, проблем у передачі даних, збоїв у роботі пристроїв, а іноді – навіть до їх фізичного пошкодження. Тому питання створення доступного, точного й універсального засобу для тестування кабелів є надзвичайно актуальним. Сучасні промислові тестери часто є дорогими, складними у використанні або мають обмежений функціонал.

Мета кваліфікаційної роботи полягає у розробці USB-тестера кабелів та трасера на основі платформи Arduino з використанням сенсорного екрана. Такий пристрій дозволить не тільки перевіряти цілісність провідників, виявляти обриви чи короткі замикання, а й визначати правильність підключення контактів, що є критично важливим для нових стандартів, таких як USB Type-C.

Об'єктом дослідження є процес розробки та функціонування електронних пристроїв для тестування USB-кабелів і трасування їхніх з'єднань.

Предметом дослідження є апаратні та програмні засоби побудови USB-тестера кабелів із сенсорним інтерфейсом на базі мікроконтролера Arduino, а також методи перевірки цілісності й правильності з'єднання кабельних ліній.

Для досягнення поставленої мети необхідно вирішити такі завдання:

Для досягнення поставленої мети необхідно вирішити такі завдання:

– провести аналіз існуючих рішень та пристроїв для тестування USB-кабелів та трасування з'єднань;

- вивчити апаратні та програмні можливості платформи Arduino для реалізації функцій кабельного тестера;
- обґрунтувати вибір апаратних компонентів;;
- розробити програмне забезпечення для мікроконтролера Arduino, що забезпечує перевірку кабелів, трасування з'єднань і взаємодію з сенсорним інтерфейсом;
- проаналізувати результати тестування, оцінити переваги та недоліки створеного пристрою.

Апробація: практична значимість основних результатів дослідження підтверджена на Міжнародній науково-практичній конференції молодих вчених та студентів «Програмне та апаратне забезпечення в інформаційних технологіях» (6 травня 2025 р., м. Луцьк) [1].

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз пристроїв тестування кабелів

На сьогодні ринок пропонує широкий спектр пристроїв для тестування кабелів, які можна умовно поділити на кілька основних груп залежно від їх призначення. Основні функції таких пристроїв полягають у визначенні цілісності провідників, перевірці правильності розведення контактів, виявленні коротких замикань, а також у деяких моделях – вимірюванні характеристик живлення, таких як напруга, сила струму та споживана потужність.

Перша група – це комерційні USB-тестери, призначені насамперед для вимірювання параметрів живлення. Наприклад, пристрої типу Keweisi KWS-0VA дозволяють перевіряти напругу, струм і потужність, що передаються через кабель, а тестери Fnirsi FNB58 підтримують роботу з протоколами швидкої зарядки (PD, QC) та мають кольорові дисплеї для відображення результатів. Однак їхні функції здебільшого обмежені контролем електричних характеристик без перевірки внутрішньої структури з'єднань.

FNIRSI FNB58 (рис. 1.1) є високоточним USB-тестером, що використовується для вимірювання напруги, струму, потужності, а також аналізу протоколів швидкої зарядки. Він призначений для інженерів, техніків і дослідників, які займаються аналізом USB-зарядних пристроїв, адаптерів живлення та кабелів. Пристрій підтримує широкий діапазон напруги від 4 до 28 В, струму до 7 А та потужності до 120 Вт, а також оснащений 16-бітним зовнішнім аналого-цифровим перетворювачем, що забезпечує високу роздільну здатність вимірювань.

Пристрій обладнано 2,0-дюймовим кольоровим TFT-дисплеєм з гравітаційним сенсором, що дозволяє автоматично повертати зображення на екрані відповідно до положення тестера. Інтерфейси підключення включають USB-A, USB-C та micro-USB, а також можливість підключення до ПК для оновлення прошивки або зчитування даних. Крім того, FNIRSI FNB58 підтримує

Bluetooth-зв'язок, що дозволяє використовувати спеціальний додаток на Android для моніторингу даних у реальному часі [2].



Рисунок 1.1 – FNIRSI FNB58 [2]

Серед найважливіших функціональних можливостей варто відзначити автоматичне визначення та ініціювання основних протоколів швидкої зарядки, що робить цей пристрій універсальним інструментом для перевірки сумісності обладнання та кабелів. Тестер також забезпечує можливість аналізу пульсацій і шумів завдяки високошвидкісному вибірковому аналізу, побудови графіків напруги та струму з низькою або високою швидкістю та здійснення довготривалих записів параметрів до 9 годин.

Практичне застосування FNIRSI FNB58 охоплює вимірювання опору кабелів із зчитуванням E-Marker чіпів, аналіз продуктивності зарядних пристроїв, тестування сумісності зі швидкою зарядкою та діагностику несправностей у ланцюгах живлення [3].

Друга група – це тестери для перевірки цілісності провідників, серед яких, наприклад, можна виділити модель Mechanic DT3. Такі пристрої дозволяють визначити, чи всі провідники в кабелі з'єднані правильно, чи є обриви, короткі замикання або неправильне розведення контактів. Вони особливо корисні для тих

випадків, коли потрібно перевірити кабелі на підтримку не лише заряджання, а й передачі даних.

Mechanic DT3 (рис. 1.2) є компактним USB-кабельним тестером, призначеним для швидкої перевірки цілісності й функціональності кабелів типу USB-C, Micro-USB, Lightning та USB-A. Його принцип роботи ґрунтується на тому, що після підключення кабелю тестер використовує вбудовані світлодіоди для індикації стану основних ліній: живлення (VBUS), землі (GND), ліній даних (D+, D–), а також каналу конфігурації (CC), який є актуальним для USB-C. Якщо всі індикатори світяться, це вказує на справність кабелю, тоді як відсутність світіння певних ліній дозволяє швидко локалізувати несправність, наприклад, обрив або коротке замикання. За живлення відповідає порт USB-C з напругою 5 В, що забезпечує простоту використання без додаткового програмного забезпечення. Розміри пристрою складають приблизно 54×50×9 мм, а вага – лише 12 грамів, що робить його зручним для щоденного використання у майстернях і сервісних центрах [4].

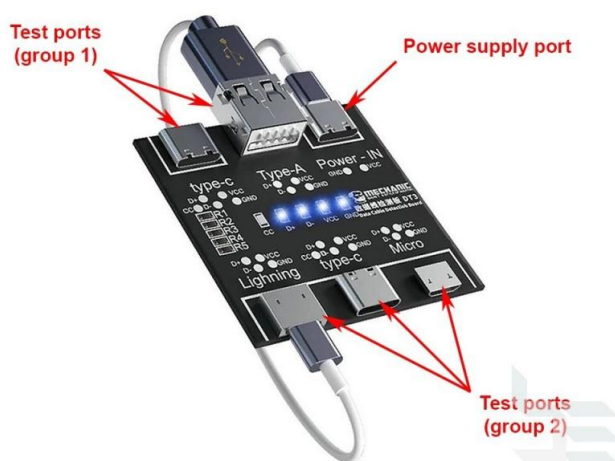


Рисунок 1.2 – Mechanic DT3 [4]

Практичне застосування Mechanic DT3 дозволяє швидко діагностувати типові несправності кабелів, що використовуються для зарядки смартфонів, планшетів або інших електронних пристроїв. Завдяки простоті інтерфейсу та

низькій ціні він є популярним серед майстрів з ремонту мобільних пристроїв. Однак варто зазначити, що цей тестер не призначений для вимірювання напруги, струму, опору або перевірки підтримки протоколів швидкої зарядки, а також не підходить для тестування активних кабелів із вбудованими мікросхемами [5].

Загалом Mechanic DT3 можна вважати базовим інструментом для первинної діагностики USB-кабелів, що значно економить час під час ремонту чи тестування обладнання. Пристрій доступний для замовлення на популярних платформах, таких як Parts4Laptops, Fixshop та GSM-Forsage, а його середня ринкова ціна варіюється в межах 3-8 євро залежно від комплектації.

Третю групу складають універсальні тестери, що поєднують функції обох попередніх типів. Наприклад, Treedix USB Cable Tester оснащений індикаторами для кожного провідника, підтримує перевірку різних типів USB-роз'ємів та дозволяє швидко визначити, чи кабель є повнофункціональним. Проте навіть такі комбіновані пристрої мають певні обмеження, пов'язані з їхньою жорсткою апаратною конструкцією та фіксованим набором функцій.

Treedix USB Cable Tester (рис. 1.3) є компактним пристроєм для швидкої перевірки справності USB-кабелів різних типів, включаючи USB Type-C, Type-A, Type-B, Micro-USB, Mini-USB та Lightning.

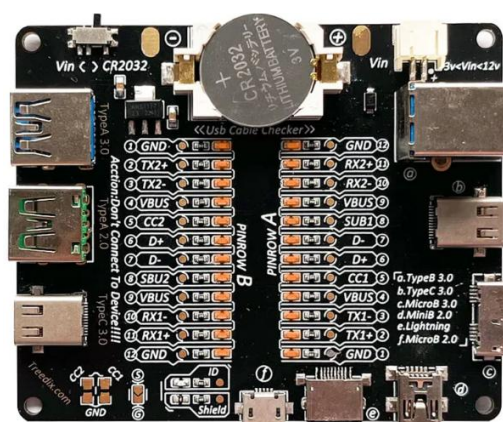


Рисунок 1.3 – Treedix USB Cable Tester [6]

Його основне завдання – виявлення обривів, коротких замикань та порушень контактів у лініях живлення і даних. Пристрій має 24 світлодіодні

індикатори, що дозволяють візуально оцінити стан кожного контакту в кабелі, де світіння індикатора означає справність, а його відсутність сигналізує про несправність або порушення ланцюга. За живлення тестера відповідає батарейка типу CR2032 або зовнішнє джерело живлення з напругою від 3 до 12 В, а конструкція тестера передбачає захист від неправильного підключення живлення, що підвищує його надійність [6].

Treedix USB Cable Tester є простим у використанні, не потребує додаткового програмного забезпечення й підходить як для професійного, так і для аматорського застосування. Пристрій допомагає технікам оперативно діагностувати стан кабелів, що є особливо корисним під час ремонту гаджетів, тестування аксесуарів або перевірки кабелів перед покупкою. Водночас варто зазначити, що цей тестер не призначений для вимірювання електричних параметрів, таких як напруга, струм чи опір, а також не може перевіряти підтримку протоколів швидкої зарядки або якість передачі даних на високих швидкостях.

Загалом Treedix USB Cable Tester є ефективним і доступним інструментом для базової перевірки кабелів, який значно полегшує роботу в сервісних центрах, ремонтних майстернях і домашніх умовах. Ціна пристрою на ринку коливається від 15 до 25 доларів США залежно від комплектації та каналу постачання, що робить його привабливим вибором для широкого кола користувачів.

Четверта група – кабельні тестери-трасери (наприклад, NOYafa NF-8209), які зазвичай використовуються в телекомунікаціях для тестування Ethernet-кабелів (RJ-45), але деякі моделі адаптовані й для USB. Ці пристрої здатні визначати послідовність підключення жил, виявляти обриви та короткі замикання, а також допомагають трасувати кабель у складній мережі. Такі пристрої є багатофункціональними, проте мають значну вартість і складні у використанні для аматорів.

Noyafa NF-8209 (рис. 1.4) – це професійний багатофункціональний тестер для діагностики мережевих кабелів, який поєднує в собі функції трасування, вимірювання довжини, перевірки PoE та виявлення несправностей у лініях

зв'язку. Цей пристрій призначений для роботи з кабелями типу Cat5e, Cat6, Cat6a (STP і UTP), а також підтримує телефонні та коаксіальні кабелі.



Рисунок 1.4 – Noyafa NF-8209 [7]

Основний блок тестера оснащений чітким ЖК-дисплеєм розміром 128×64 мм з яскравим підсвічуванням, що забезпечує зручне відображення результатів тестування. Живлення здійснюється від трьох батарейок типу ААА, а також передбачено автоматичне вимкнення через 15, 30 або 60 хвилин для економії енергії.

Пристрій дозволяє проводити тестування цілісності кабелю, виявляючи обриви, короткі замикання та неправильні з'єднання. Функція вимірювання довжини кабелю охоплює діапазон від 2,5 до 200 метрів з точністю до $\pm 1,6$ м. Для трасування кабелів передбачено три режими: аналоговий, цифровий та PoE, що дозволяє ефективно знаходити потрібний кабель навіть у складних умовах.

Тестер також підтримує перевірку PoE, визначаючи наявність живлення по Ethernet у діапазоні від 5 до 60 В, з автоматичним розпізнаванням стандартів IEEE 802.3af та 802.3at, а також нестандартних конфігурацій. Це особливо корисно при встановленні та обслуговуванні пристроїв, що живляться через Ethernet, таких як IP-камери або точки доступу Wi-Fi [7].

Додатково, NF-8209 оснащений функцією Port Flash, яка дозволяє ідентифікувати активний порт на комутаторі або маршрутизаторі, що полегшує пошук підключень у великих мережах. Вбудований ліхтарик та можливість підключення навушників підвищують зручність використання пристрою в умовах недостатнього освітлення або шумного середовища.

Комплект постачання включає основний тестер, приймач, віддалений модуль, м'який чохол для зберігання та транспортування, а також необхідні кабелі для тестування. Завдяки широкому функціоналу та зручному інтерфейсу, Noyafa NF-8209 є надійним інструментом для фахівців, які займаються встановленням, обслуговуванням та діагностикою мережевих систем.

Таким чином, проведений аналіз показує, що, попри наявність великої кількості пристроїв для тестування USB-кабелів, жоден з них не забезпечує одночасно доступності, багатофункціональності, трасування з'єднань та зручного сенсорного інтерфейсу. Це відкриває перспективи для розробки нового універсального рішення на базі Arduino, яке могло б поєднувати переваги готових тестерів та гнучкість саморобних проєктів, забезпечуючи інженерів, студентів та ентузіастів ефективним і доступним інструментом для роботи з USB-кабелями.

Отже, аналіз показує, що попри наявність великої кількості готових рішень, жоден з доступних тестерів не поєднує у собі одночасно доступну ціну, підтримку різних стандартів USB, трасування з'єднань та зручний сенсорний інтерфейс, що створює передумови для розробки нового пристрою, адаптованого під сучасні потреби.

1.2 Принципи трасування з'єднань у слаботочних мережах

Трасування з'єднань у слаботочних мережах – це процес визначення фізичного розташування кабелів, з'єднань і розгалужень для подальшого контролю, діагностики та обслуговування. Слаботочні мережі включають системи передачі даних через Ethernet, USB (рис. 1.5), сигналізації, домофонні

мережі, системи відеоспостереження, пожежної безпеки, охоронної сигналізації та інші мережі, де передається слабкий електричний сигнал (зазвичай до 50 В).



Рисунок 1.5 – USB-трасер [8]

Основний принцип трасування полягає у виявленні шляху, яким проходить сигнал від одного кінця кабелю до іншого, та фіксації всіх проміжних точок підключення. Для цього використовуються спеціалізовані пристрої – трасери, які складаються з передавача (генератора сигналу) та приймача. Передавач генерує слабкий тестовий сигнал, який подається на кабель або лінію, а приймач дозволяє визначити місце проходження цього сигналу навіть у прихованих кабельних каналах чи стінах (рис. 1.6).

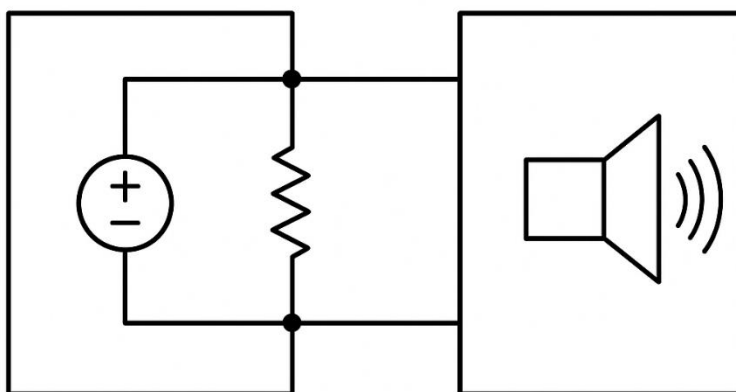


Рисунок 1.6 – Схема роботи трасера

Важливими технічними характеристиками трасерів є чутливість приймача, частота тестового сигналу, можливість фільтрації завад, підтримка різних типів роз'ємів (наприклад, RJ-45, USB, RJ-11) та наявність індикаторів для візуального або звукового контролю. Наприклад, сучасні USB-трасери мають функції визначення відповідності контактів, виявлення перехресних або обірваних ліній, а також перевірки симетрії передачі даних.

Процес трасування важливий не лише для правильного монтажу нових мереж, а й для діагностики існуючих. Наприклад, при наявності несправностей (обриви, короткі замикання, погані контакти) трасування дозволяє локалізувати проблемну ділянку та швидко провести ремонт. У слаботочних мережах особливо важливо враховувати низький рівень сигналів, оскільки занадто сильне тестове навантаження може пошкодити обладнання або вплинути на роботу інших систем.

Сучасні пристрої трасування можуть інтегруватися з мікроконтролерами (рис. 1.7), що дає можливість автоматизувати тестування та виводити інформацію на дисплей. Використання сенсорних екранів спрощує інтерфейс користувача, дозволяючи миттєво бачити схему підключень, результати перевірки та діагностичні повідомлення.

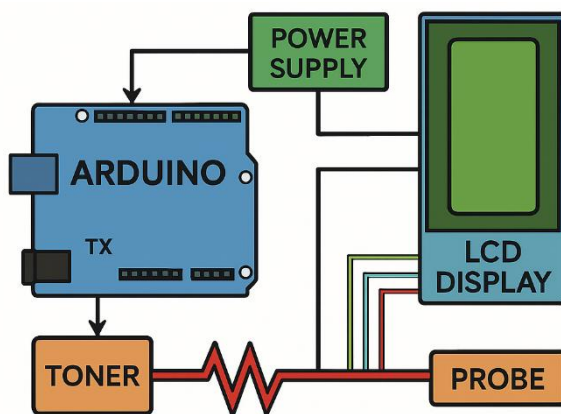


Рисунок 1.7 – Схема принципу роботи трасера на основі Arduino

Таким чином, принципи трасування з'єднань у слаботочних мережах базуються на точному визначенні шляху проходження сигналу, виявленні

порушень у підключенні та використанні спеціалізованих апаратних і програмних засобів для підвищення ефективності контролю мережі.

1.3 Огляд інтерфейсів USB та типів сигналів

Інтерфейс USB (Universal Serial Bus) є одним з найпоширеніших стандартів для підключення периферійних пристроїв до комп'ютерів і контролерів. Він забезпечує передачу живлення та даних між пристроями, має широку сумісність і підтримує «гаряче» підключення (hot swap). Існує кілька поколінь цього стандарту, серед яких найбільш поширеними є USB 2.0 та USB 3.0 [9].

USB 2.0 був запроваджений у 2000 році й забезпечує швидкість передачі даних до 480 Мбіт/с (High-Speed). У цьому стандарті використовується чотирипровідне з'єднання (рис. 1.8): два дроти для живлення (VCC та GND) та два диференціальні дроти D+ і D- для передачі даних. Дифсигнали (differential signals) дозволяють підвищити стійкість до завад і забезпечують ефективну передачу інформації на високих швидкостях. Основні типи сигналів — це low-speed, full-speed і high-speed режими, що визначаються електричними характеристиками та режимами роботи пристрою.

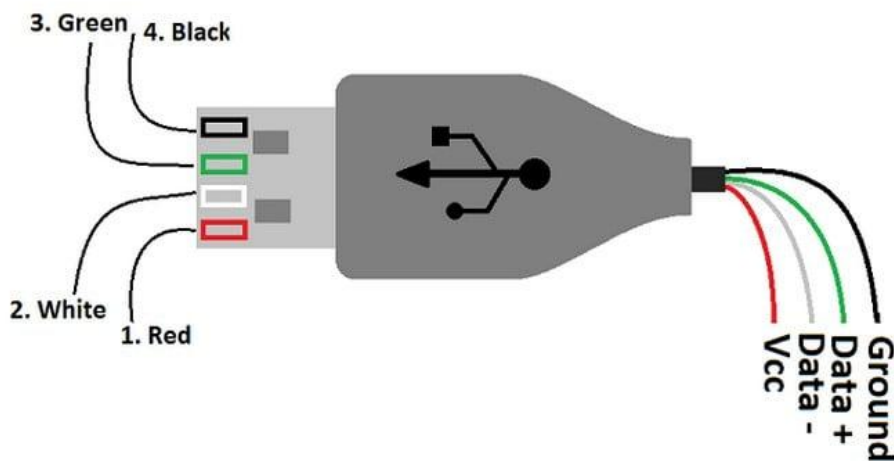


Рисунок 1.8 – Структура контактів USB 2.0 (Type-A) [9]

USB 3.0 (рис. 1.9) , запроваджений у 2008 році, забезпечує значно вищі швидкості – до 5 Гбіт/с (SuperSpeed). Він сумісний з USB 2.0, але має додаткові пари контактів: крім чотирьох основних, додаються п'ять нових (дві пари для SuperSpeed-ліній TX/RX і один GND). Така структура дозволяє одночасно підтримувати як старі, так і нові пристрої. У стандарті USB 3.0 дані передаються через окремі диференціальні пари, що розвантажує основні лінії D+ і D–, а також підвищує пропускну здатність і знижує латентність [10].

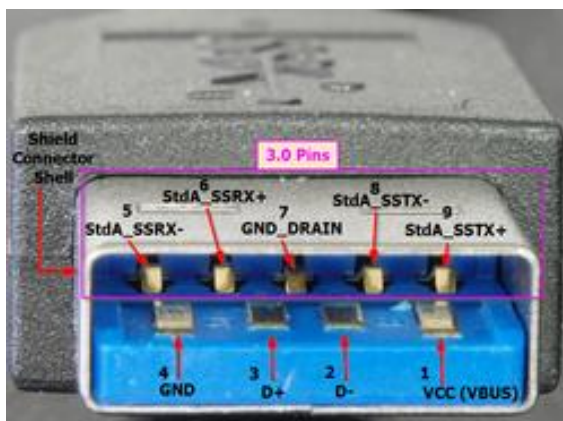


Рисунок 1.9 – Структура контактів USB 3.0 (Type-A) [10]

В таблиці 1.1 подано порівняльну характеристику інтерфесів.

Таблиця 1.1 – Порівняльна характеристика USB 2.0 та USB 3.0

Характеристика	USB 2.0	USB 3.0
Максимальна швидкість	480 Мбіт/с	5 Гбіт/с
Кількість контактів	4	9
Диференційні пари	1 (D+/D–)	3 (D+/D–, TX+/TX–, RX+/RX–)
Типи роз'ємів	Type-A, Type-B, Mini, Micro	Type-A, Type-B, Type-C
Сумісність	3 USB 1.1	3 USB 2.0 та 1.1

Типи сигналів у USB-передачі можна умовно поділити на кілька груп: сигнали даних (дані передачі/прийому), сигнали керування (керування потоком, синхронізація), сигнали живлення (VCC, GND) та диференціальні сигнали (D+/D–, TX+/TX–, RX+/RX–). Диференціальна передача (рис. 1.10) є ключовою

для зниження впливу електромагнітних завад, оскільки приймач відстежує різницю потенціалів між парами, а не абсолютні значення.

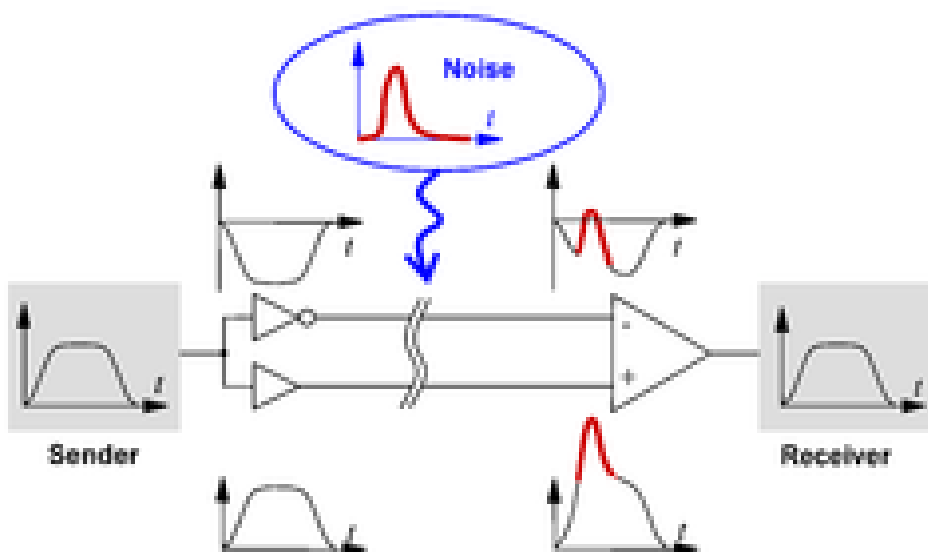


Рисунок 1.10 – Принцип диференціальної передачі сигналу [11]

У сучасних рішеннях також широко застосовується стандарт USB Type-C, який підтримує як USB 2.0, так і USB 3.0/3.1/3.2, забезпечуючи при цьому симетричність роз'єму, високу щільність контактів і підтримку додаткових режимів (наприклад, Power Delivery до 100 Вт або Alternate Mode для відеосигналу). Type-C має 24 контакти, які включають живлення, заземлення, лінії управління та кілька пар для високошвидкісної передачі даних.

Таким чином, огляд інтерфейсів USB 2.0 та 3.0 демонструє еволюцію технічних рішень, спрямованих на підвищення швидкості, сумісності та енергоефективності, а також важливість врахування типів сигналів при розробці пристроїв для тестування кабелів.

За результатами проведеного аналізу було зроблено висновок, що створення сучасного USB-тестера на базі Arduino є актуальним та технічно обґрунтованим завданням, яке поєднує програмні й апаратні засоби, а також потребує врахування специфіки різних поколінь USB-стандартів.

РОЗДІЛ 2

ПРОЄКТУВАННЯ USB-ТЕСТЕРА

2.1 Вибір апаратних компонентів пристрою

Для побудови універсального USB-тестера, що здатний працювати з різними стандартами (USB B/B3, USB A/A3, USB C, USB Mini, USB Micro/Micro3) та мати зручний сенсорний інтерфейс, було ретельно підібрано основні апаратні компоненти. Їхній вибір зумовлений технічними характеристиками, простотою інтеграції, доступністю на ринку та підтримкою широкого спектра бібліотек для програмного забезпечення.

Одним з центральних елементів системи є Arduino Mega.

Arduino Mega 2560 (рис. 2.1) – це пристрій на основі мікроконтролера ATmega2560. У його склад входить все необхідне для зручної роботи з мікроконтролером: 54 цифрових входи/виходи (з яких 15 можуть використовуватися в якості ШІМ-виходів), 16 аналогових входів, 4 UART (апаратних приймача для реалізації послідовних інтерфейсів), кварцовий резонатор на 16 МГц, роз'єм USB, роз'єм живлення, роз'єм ICSP для внутрішнього схемного програмування і кнопка скидання.

Для початку роботи з пристроєм досить просто подати живлення від AC/DC-адаптера або батарейки, або підключити його до комп'ютера за допомогою USB-кабелю. Arduino Mega сумісний з більшістю плат розширення, розроблених для Arduino Duemilanove і Diecimila.

Arduino Mega 2560 відрізняється від усіх попередніх плат тим, що в ньому для перетворення інтерфейсів USB-UART замість мікросхеми FTDI використовується мікроконтролер ATmega16U2 (ATmega8U2 в версіях плати R1 і R2).

На платі Mega 2560 версії R2 доданий резистор, що підтягує до землі лінію HWB мікроконтролера 8U2. Така дія дозволяє спростити процес оновлення прошивки і перехід пристрою в режим DFU [12].



Рисунок 2.1 – Arduino Mega 2560 R3 [13]

Arduino Mega 2560 відрізняється від усіх попередніх плат тим, що в ньому для перетворення інтерфейсів USB-UART замість мікросхеми FTDI використовується мікроконтролер ATmega16U2 (ATmega8U2 в версіях плати R1 і R2). Зміни на платі версії R3 перераховані нижче.

Терморегулятори: додані виведення SDA і SCL (біля виведення AREF), а також два нових виведення, розташованих біля виведення RESET. IOREF – дозволяє платам розширення підлаштовуватися під робочу напругу Arduino. Даний вивід передбачений для сумісності плат розширення як з 5 В Ардуіно на базі мікроконтролерів AVR, так і з 3,3В-платами Arduino Due. Другий вихід ні до чого не приєднаний і зарезервований для майбутніх цілей. Покращена стійкість ланцюга скидання

В таблиці 2.1 подано порівняльну характеристику мікроконтролерів Arduino Uno та Arduino Mega 2560.

Таблиця 2.1 – Порівняльна характеристика мікроконтролерів Arduino Uno та Arduino Mega 2560

Характеристика	Arduino Uno	Arduino Mega 2560
1	2	3
Мікроконтролер	ATmega328P	ATmega2560
Тактова частота	16 МГц	16 МГц
Flash-пам'ять	32 КБ	256 КБ
SRAM	2 КБ	8 КБ
EEPROM	1 КБ	4 КБ
Кількість цифрових входів/виходів	14 (з них 6 PWM)	54 (з них 15 PWM)
Кількість аналогових входів	6	16

Продовження таблиці 2.1

1	2	3
Апартні UART	1	4
Габарити	68,6 × 53,4 мм	101,6 × 53,3 мм
Напруга живлення	5 В	5 В
Напруга живлення від USB	7-12 В	7-12 В
Сумісність з додатковими модулями	Висока	Висока, але потребує більше місця

З аналізу таблиці 2.1 видно, що Arduino Uno підходить для простих проєктів із обмеженим числом каналів, тоді як Arduino Mega 2560 має значно більше портів введення/виведення, більшу пам'ять і кілька UART, що робить його оптимальним вибором для систем із сенсорними екранами, мультиінтерфейсними модулями та складною логікою.

Arduino Mega може житись від USB або від зовнішнього джерела живлення – тип джерела вибирається автоматично.

Виводи живлення які розташовані на платі, перераховані нижче (рис. 2.2):

- VIN. Напруга, що надходить в Arduino безпосередньо від зовнішнього джерела живлення (не пов'язане з 5 В від USB або іншим стабілізованою напругою). Через цей висновок можна як подавати зовнішнє живлення, так і споживати струм, коли пристрій живиться від зовнішнього адаптера;

- на цей вивід надходить напруга 5 В від стабілізатора напруги на платі, поза незалежності від того, як живиться пристрій: від адаптера (7-12 В), від USB (5 В) або через висновок VIN (7-12 В). Живити пристрій через висновки 5V або 3V3 не рекомендується, оскільки в цьому випадку не використовується стабілізатор напруги, що може привести до виходу плати з ладу;

- 3,3 В, що надходять від стабілізатора напруги на платі. Максимальний струм, споживаний від цього висновку, становить 50 мА;

- GND: заземлення;

- IOREF надає платам розширення інформацію про робочій напрузі мікроконтролера Ардуіно. Залежно від напруги, ліченого з виведення IOREF, плата розширення може переключитися на відповідний джерело живлення або задіяти перетворювачі рівнів, що дозволить їй працювати як з 5 В, так і з 3,3 В.

– на платах версії R1: замкнути перемичку на звороті плати (біля зображення Італії), після чого скинути 8U2;

– на платах версій R2 і вище – для спрощення переходу в режим DFU присутній резистор, що підтягує до землі лінію HWB мікроконтролера 8U2/16U2. Після переходу в DFU-режим для завантаження нової прошивки можна використовувати програмне забезпечення Atmel's FLIP (для Windows) або DFU programmer (для MacOS і Linux). Альтернативний варіант - прошити мікроконтролер через роз'єм для внутрішньосхемного програмування ISP за допомогою зовнішнього програматора, проте в цьому випадку DFU-завантажувач затреться [14].

Для побудови зручного графічного інтерфейсу в системі тестування USB-кабелів обрано резистивний сенсорний TFT LCD екран діагоналлю 2,8 дюйма (рис. 2.3). Такі екрани поєднують у собі компактність, низьке енергоспоживання, хорошу сумісність з Arduino та зручність взаємодії завдяки сенсорному керуванню.

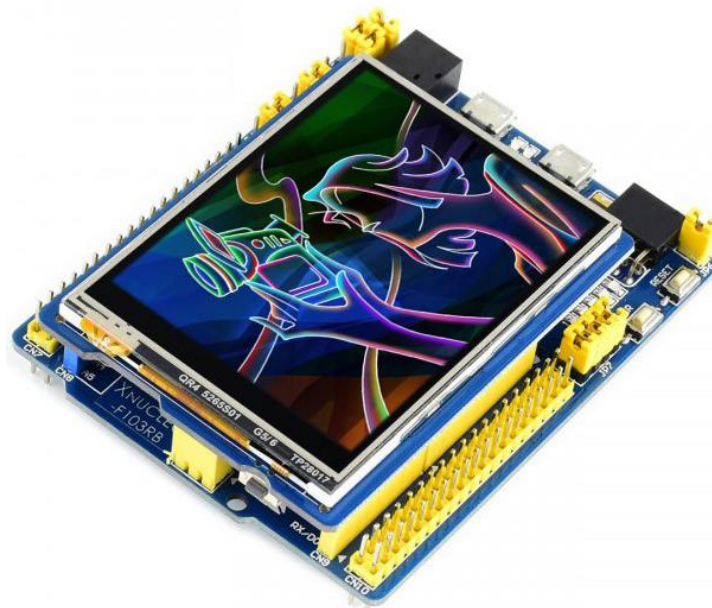


Рисунок 2.3 – TFT LCD-дисплей [15]

Основою екрана є тонкоплівковий транзисторний дисплей (Thin-Film Transistor, TFT), який забезпечує відображення зображень у кольоровому режимі

(часто до 262 тис. кольорів). Діагональ 2,8 дюйма та роздільна здатність близько 240×320 пікселів дозволяють виводити як текстову інформацію (дані перевірки, результати трасування), так і прості графічні елементи (наприклад, індикатори, схеми підключень).

Резистивний сенсорний шар складається з двох прозорих провідних пластин, розділених тонким діелектриком. При натисканні верхня пластина деформується, замикаючи коло в точці дотику, що дозволяє визначити координати X та Y. Такий принцип роботи має низку переваг: екран реагує на будь-який предмет (палець, стилус, рукавичку), є недорогим у виробництві та простим в інтеграції. Проте варто враховувати, що резистивні екрани менш чутливі, ніж ємнісні, та мають нижчу прозорість через багат шарову структуру [15].

Для підключення кабелів тестування використовується модуль USB- роз'ємів (Type-A, Type-C). Його наявність дозволяє працювати як зі звичними кабелями, так і з новітніми моделями з роз'ємами Type-C, що набувають все більшої популярності. Такі модулі часто оснащені пайковими контактами або спеціальними адаптерами для підключення до макетних плат. Їхнє використання спрощує підключення безпосередньо до ліній мікроконтролера та дозволяє реалізувати перевірку різних ліній передачі даних, живлення, заземлення та сигналів керування.

Особливу увагу приділено модулю живлення та захисту, адже USB-лінії можуть передавати напругу 5 В або вище (у випадку з Power Delivery – до 20 В). Для захисту мікроконтролера використовуються спеціальні перехідні модулі з понижувальними перетворювачами (наприклад, на базі AMS1117) та ланцюгами захисту від перенапруги (TVS-діоди, резистори, варистори). Також застосовуються модулі контролю споживаного струму, що дозволяє перевіряти навантажувальні характеристики кабелів. Використання такого екрана дозволяє побудувати зручне графічне меню, виводити результати перевірки у кольоровій формі (наприклад, зелений – успішний тест, червоний – виявлена помилка) та забезпечувати взаємодію з користувачем без потреби у фізичних кнопках.

2.2 Розробка структурної та принципової електричної схеми

Розробка структурної та принципової електричної схеми є ключовим етапом проектування пристрою для тестування USB-кабелів та трасування з'єднань. На цьому етапі визначається, як саме взаємодіють між собою основні апаратні компоненти системи, яким чином передається інформація між модулями та як забезпечується живлення та захист мікроконтролера та периферійних елементів.

Структурна схема (рис. 2.4) відображає загальну архітектуру пристрою. Центральним елементом є мікроконтролер Arduino Mega, який керує збором сигналів від тестованого USB-кабелю через модуль роз'ємів (USB Type-A, Type-C), опрацьовує отримані дані та виводить результати на сенсорний TFT LCD дисплей. Крім того, у систему інтегровано модуль живлення з напругою 5 В, що забезпечує стабільну роботу мікроконтролера, дисплея та допоміжних елементів.

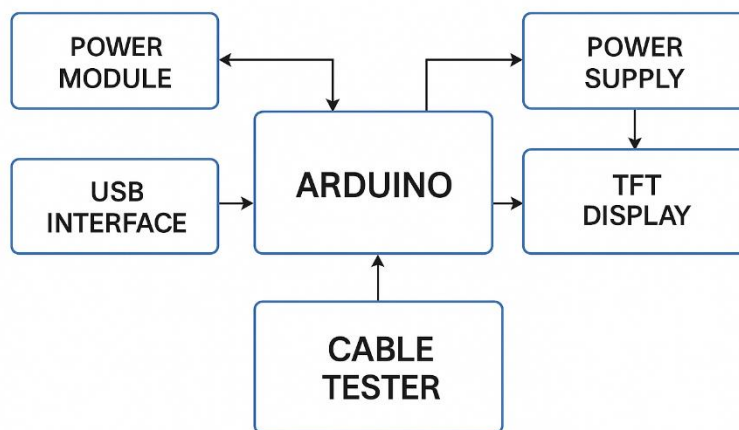


Рисунок 2.4 – Структурна схема пристрою тестування USB-кабелів на базі Arduino

На принциповій електричній схемі (рис. 2.5) деталізуються з'єднання між окремими контактами мікроконтролера та модулями. Для підключення TFT дисплея використовуються порти SPI (паралельний режим), що дозволяє передавати дані з високою швидкістю. Сенсорний шар підключається через аналогові входи A0–A1, забезпечуючи зчитування координат натискання. Для

USB-модулів виділяються цифрові порти D2-D9, які перевіряють стан ліній D+, D-, TX+, TX- тощо.

Живлення забезпечується через вхід V_{in} (7-12 В), що надходить на стабілізатор AMS1117 (або аналогічний), який видає стабільні 5 В на всі компоненти. Для захисту ліній USB від перенапруги застосовуються TVS-діоди, а для обмеження струму – резистори номіналом 100-220 Ом на кожному сигнальному проводі. Важливим компонентом є кнопка скидання (Reset), підключена до RST, що дозволяє перезавантажувати систему при зависанні або некоректній роботі.

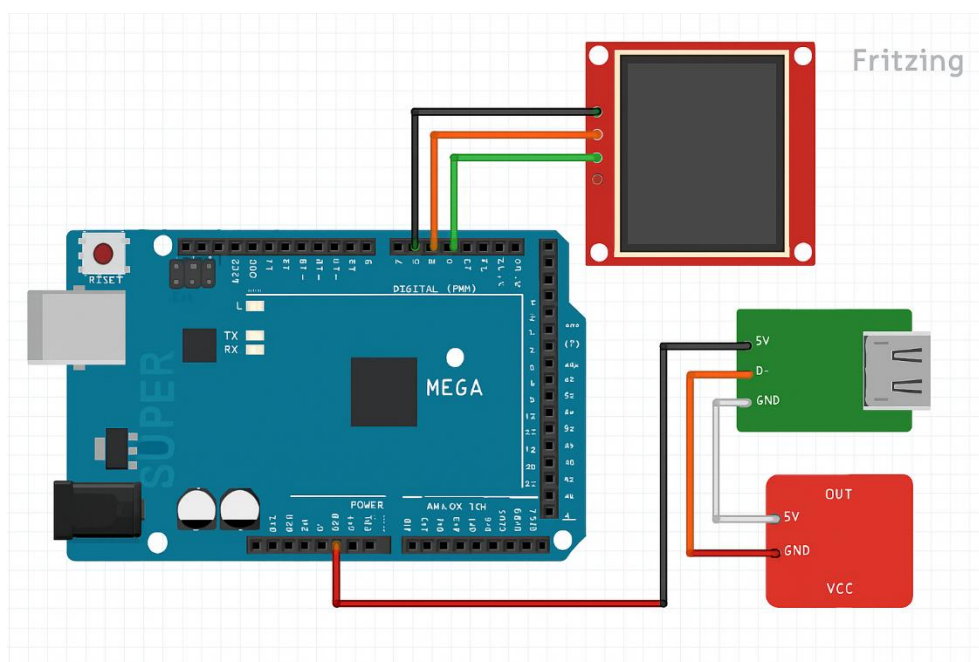


Рисунок 2.5 – Принципова електрична схема підключення компонентів пристрою до Arduino Mega

2.3 Проєктування 3D-елементів пристрою

Корпус є важливою складовою апаратної частини USB-тестера, оскільки забезпечує не лише захист електронних компонентів від зовнішнього впливу, але й формує ергономічність та естетику пристрою. Для цього проєкту корпус

проектується спеціально під габарити плати Arduino Mega, TFT-сенсорного дисплея, USB-модулів та допоміжних елементів.

Основні вимоги до корпусу включають: компактність, достатній внутрішній простір для монтажу компонентів, наявність отворів для підключення кабелів (Type-A, Type-C), отвір під дисплей, вентиляційні прорізи для уникнення перегріву, а також простоту складання. Для створення корпусу використовується CAD-середовище, наприклад Autodesk Fusion 360 або FreeCAD, де проектується фасад (рис. 2.6), кришка (рис. 2.7) та база корпусу (рис. 2.8).

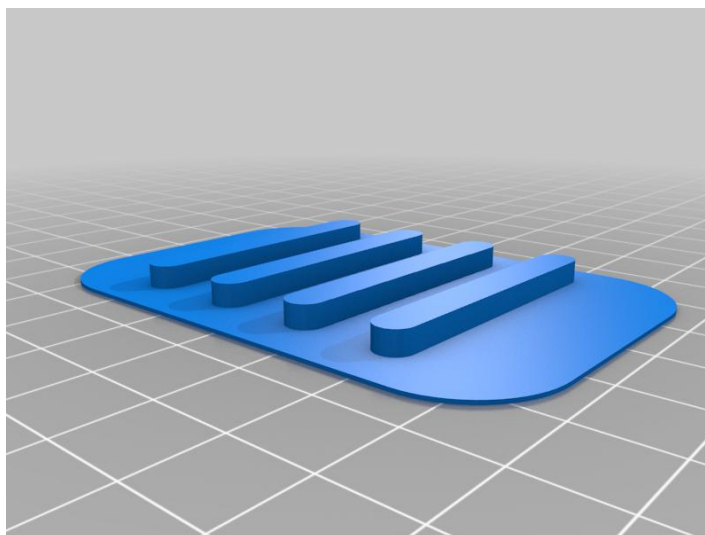


Рисунок 2.6 – Фасад корпусу

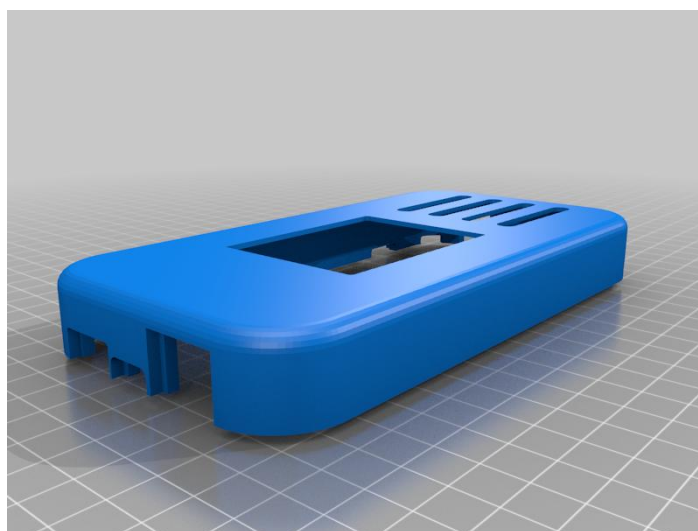


Рисунок 2.7 – Кришка корпусу

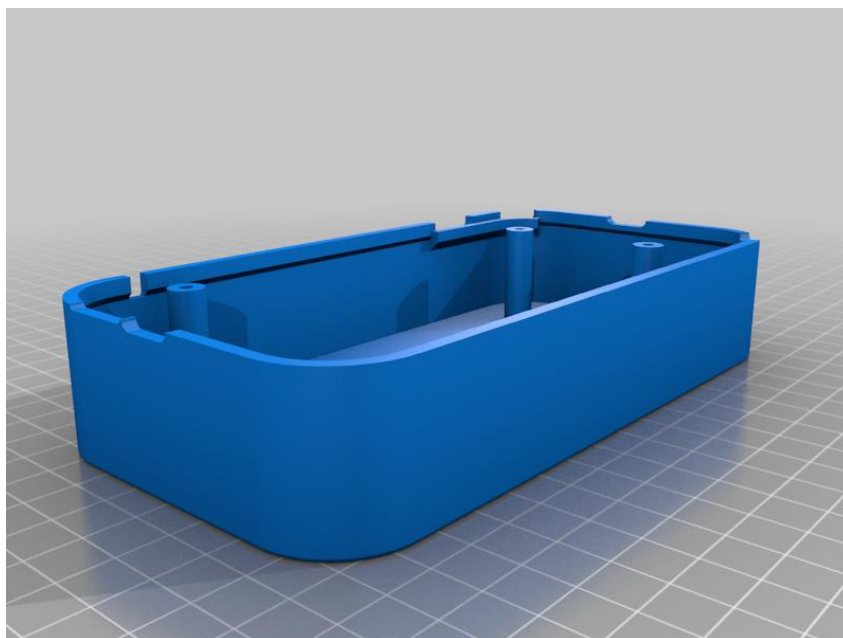


Рисунок 2.8 – База корпусу

Обраним матеріалом для виготовлення корпусу є пластик PLA, що застосовується при 3D-друці. PLA має хорошу жорсткість, біорозкладність і легкість друку, тоді як ABS забезпечує вищу термостійкість та ударостійкість. Друк корпусу здійснюється шарами товщиною 0,2-0,3 мм, що дозволяє досягти потрібної міцності при збереженні точності геометрії.

На передній панелі передбачається виріз під TFT-екран, захищений прозорою акриловою вставкою. Бокові панелі мають отвори під USB-роз'єми та, за потреби, роз'єм живлення. В корпусі передбачено місце для монтажу гвинтами або на заціпках, щоб забезпечити надійне кріплення плати всередині. Зарядки/

2.4 Проектування плати для кабельного трасування

Проектування друкованої плати (PCB) для кабельного трасера на основі Arduino Mega є важливим етапом, що визначає компактність, надійність і ефективність роботи пристрою.

Спершу у середовищі CAD (наприклад, KiCad, Eagle або EasyEDA) формується логічна схема – вона включає Arduino Mega як центральний контролер, модуль USB-роз'ємів (Type-A, Type-C), резистивний сенсорний TFT

дисплей, захисні компоненти (TVS-діоди, резистори), стабілізатори живлення та додаткові порти. Особливу увагу приділяють трасуванню високошвидкісних ліній, зокрема пар TX+/TX- та RX+/RX- для USB 3.0, які мають йти паралельно та однакової довжини, щоб уникнути збою сигналів.

На етапі розробки враховуються такі принципи (рис. 2.9, 2.10):

- окремі шини живлення та «землі» з достатньою товщиною доріжок;
- мінімізація перехресних перешкод між цифровими та аналоговими лініями;
- правильне розміщення байпас-конденсаторів біля джерел живлення;
- захисне коло на входах USB для запобігання стрибкам напруги;
- оптимізація розташування роз'ємів для зручності підключення кабелів.

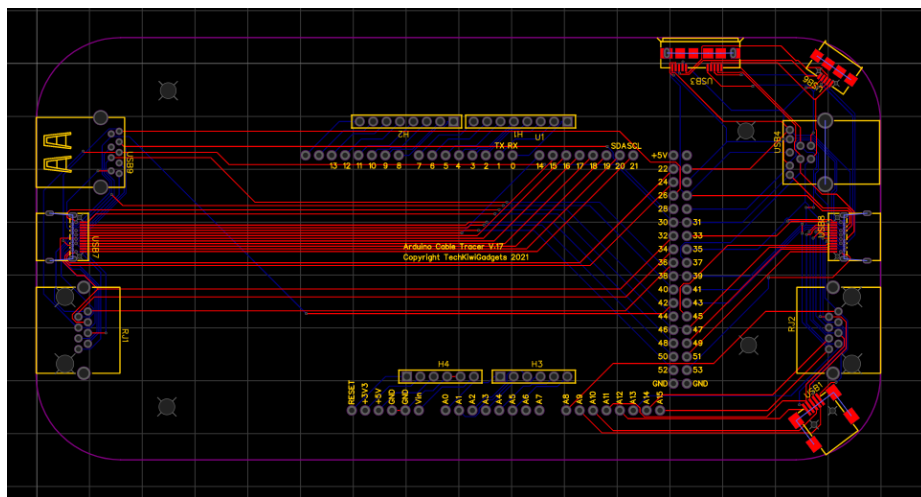


Рисунок 2.9 – Спроектована плата

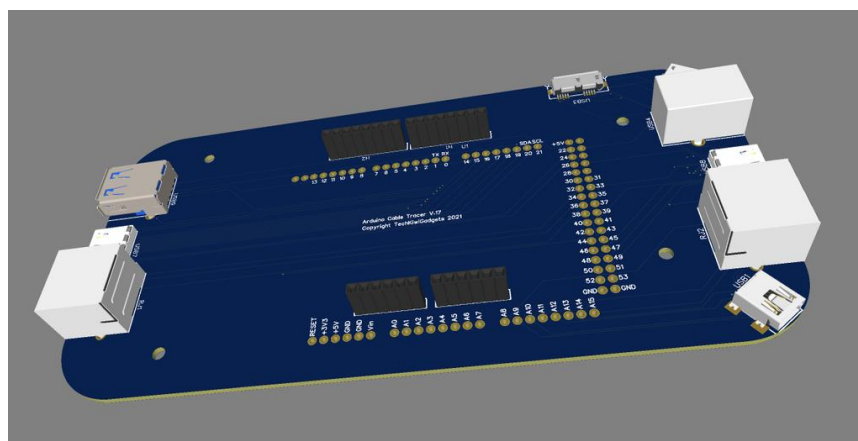


Рисунок 2.10 – 3D-модель плати

Готовий макет експортується у формат Gerber для подальшого виготовлення. Після виробництва плата перевіряється візуально (на наявність коротких замикань, дефектів доріжок), а потім монтуються компоненти – спочатку пасивні, потім активні (модулі, роз’єми, дисплей). На фінальному етапі проводиться електричне тестування: перевіряється цілісність ліній, правильність напруг живлення та функціональність усіх каналів.

На рисунку 2. 11 зображено виготовлену плату для кабельного трасера.



Рисунок 2.11 – Готова плата

Отже, виконано детальний аналіз і обґрунтування вибору апаратних компонентів для створення USB-тестера та трасера на основі Arduino Mega. Розглянуто особливості вибору мікроконтролера, TFT-сенсорного дисплея, модулів USB-роз’ємів та елементів живлення, а також їхню інтеграцію в єдину систему.

Розроблено структурну та принципову електричну схеми, що визначають архітектуру пристрою, схему взаємодії компонентів, живлення та захисту. Окремо описано проектування друкованої плати, з урахуванням вимог до розташування високошвидкісних ліній, електричної ізоляції та мінімізації перешкод. Також було приділено увагу проектуванню корпусу пристрою, який забезпечує механічний захист, зручність використання та доступність для обслуговування.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРИСТРОЮ

3.1 Монтаж пристрою

Після завершення етапів проектування схеми, розробки друкованої плати та виготовлення корпусу розпочинається процес збірки пристрою. Цей етап включає встановлення всіх компонентів на плату, їхнє надійне кріплення, підключення відповідних ліній, а також остаточне складання корпусу.

На рисунку 3.1 зображено підготовку компонентів до збирання пристрою для тестування USB-кабелів на основі Arduino Mega.



Рисунок 3.1 – Етап підготовки компонентів пристрою

Монтаж кабельного тестера:

– встановити дисплей на плату (рис. 3.2), переконавшись, що контакти вирівняні з роз'ємами. Після встановлення ще раз перевірте, чи всі контакти знаходяться в правильному положенні;

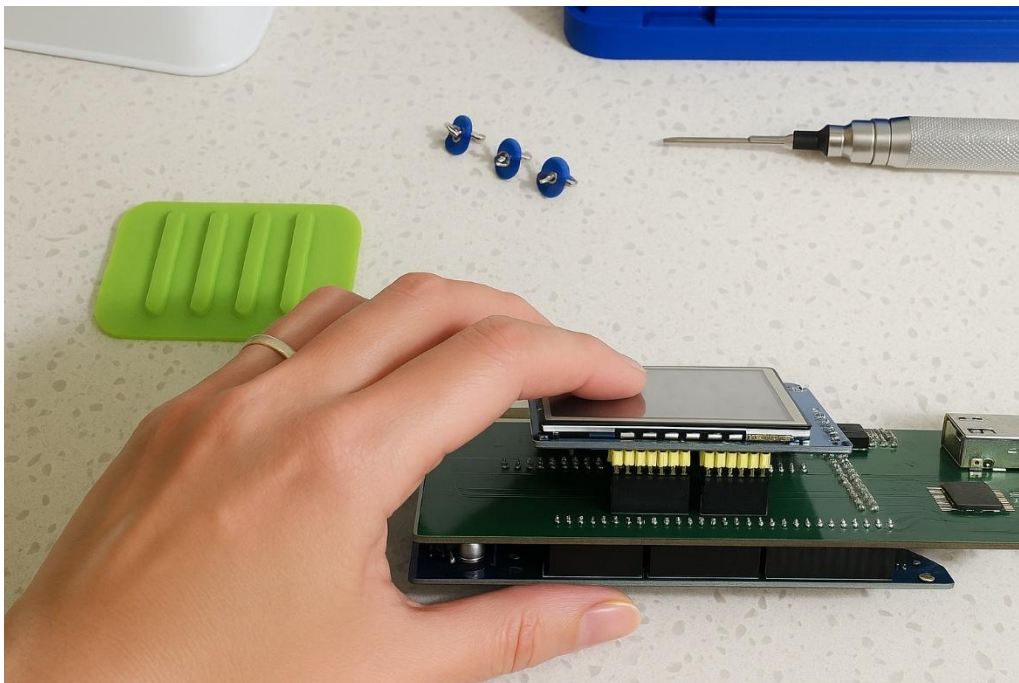


Рисунок 3.2 – Встановлення дисплею на плату

– підключити Arduino (рис. 3.3) у контакти роз'єму, розташовані на нижній стороні екрану;

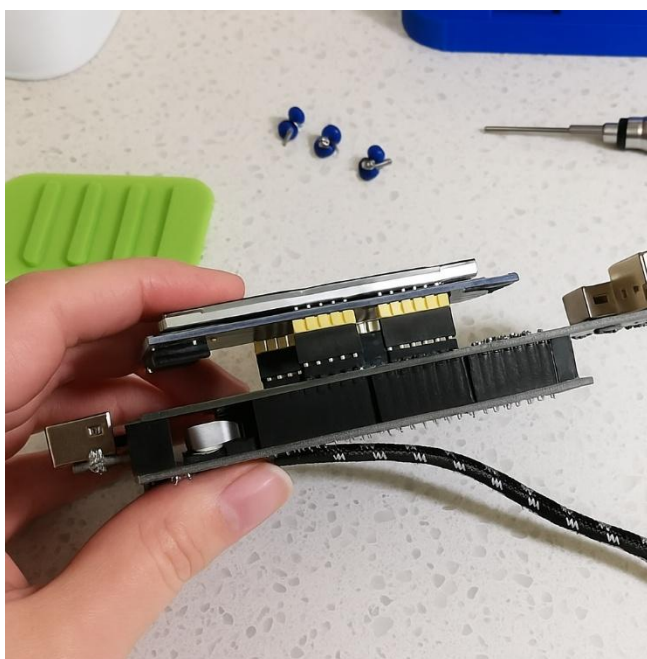


Рисунок 3.3 – Підключення Arduino

– далі варто підключити пристрій до джерела живлення USB, щоб провести швидкий функціональний тест і переконатися у відсутності проблем.

За відсутності підключених зовнішніх кабелів у лівому нижньому куті екрана не повинно бути підрахованих з'єднань, а сенсорний екран має працювати;

– розташувати нижній корпус та екран так, щоб вони були близько один до одного. Обережно протягніть кабель USB A через «Chord Grip», переконавшись, що пристрій, коли він знаходиться в основі, не заважає кабелю (рис. 3.4);



Рисунок 3.4 – Встановлення екрана в нижній корпус

– прикрутити екран до нижнього корпусу (рис. 3.5);



Рисунок 3.5 – Фіксація екрану

– скласти кришку та базу корпусу разом (рис. 3.6).



Рисунок 3.6 – Фіксація кришки та бази корпусу

На завершальному етапі проводиться тестовий запуск пристрою: перевіряється подача живлення, ініціалізація компонентів, коректність відображення даних на екрані та робота сенсорного інтерфейсу. Лише після цього пристрій вважається готовим до проведення практичного тестування.

3.2 Програмування Arduino

Програмування Arduino є важливим етапом реалізації пристрою для тестування USB-кабелів, оскільки саме програмне забезпечення забезпечує взаємодію між мікроконтролером, сенсорним екраном, модулями роз'ємів та іншими компонентами. Для розробки програмного коду використовується

середовище Arduino IDE (рис. 3.7), яке дозволяє створювати, компілювати та завантажувати скетчі безпосередньо на плату Arduino Mega.

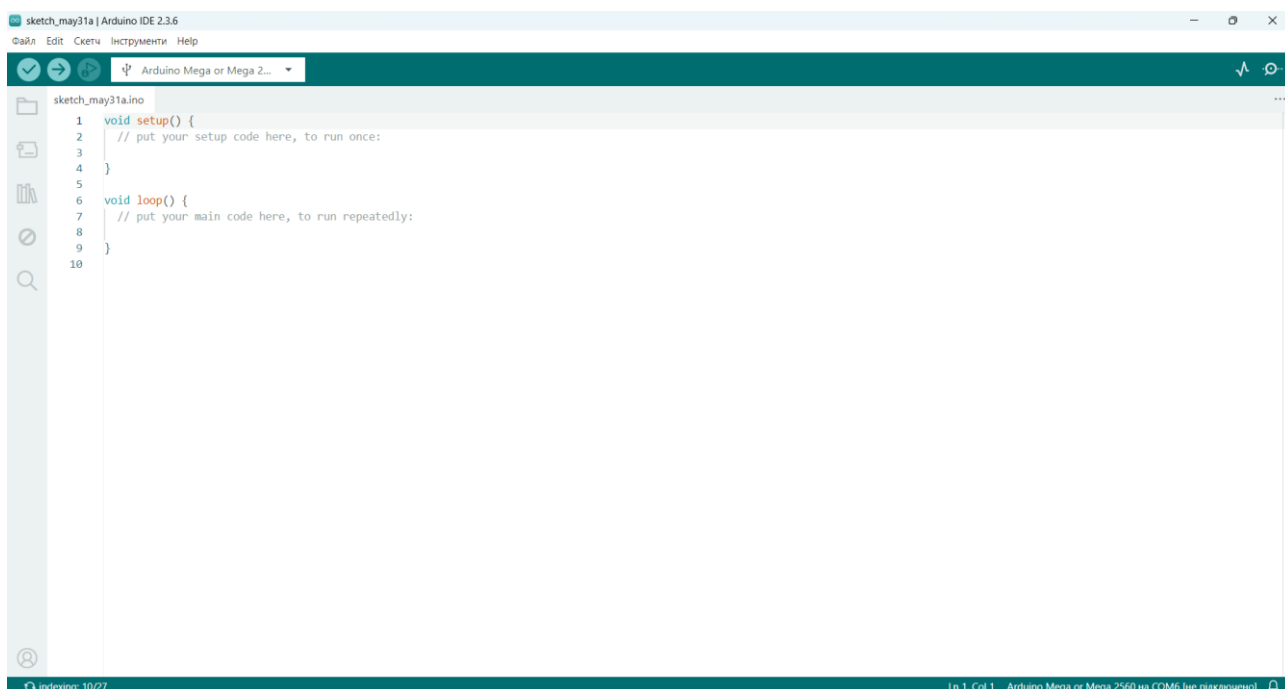


Рисунок 3.7 – Середовище Arduino IDE

Підключіть Arduino Mega до USB-порту за допомогою кабелю USB A-B, наданого виробником плати. Переконайтеся, що плата знаходиться на непровідній поверхні, щоб уникнути короткого замикання. Після підключення система має автоматично визначити пристрій та призначити йому COM-порт.

Щоб екран працював з Arduino Mega, в Arduino IDE потрібно завантажити дуже специфічні бібліотеки. Спочатку видалити наступні бібліотеки з папки /Arduino/Libraries:

- SPI.h;
- Adafruit_GFX.h;
- MCUFRIEND_kbv.h;
- TouchScreen.h.

Перейти до Скетч – Бібліотеки – Керувати бібліотеками (Manage Libraries). Знайти і встановити потрібні бібліотеки.

Основні програмні модулі будуть включати:

- ініціалізацію портів вводу/виводу, до яких підключено USB-лінії;
- ініціалізацію бібліотек для роботи з TFT LCD (наприклад, Adafruit GFX, TFT_eSPI), що забезпечують відображення текстової та графічної інформації;
- опрацювання даних, що надходять із контактів тестованого кабелю, порівняння їх зі зразками правильних з'єднань, визначення наявності обривів чи помилок;
- обробку подій від сенсорного шару (натискання кнопок на екрані, перемикання режимів);
- реалізацію алгоритмів тестування, наприклад послідовної перевірки ліній D+/D-, TX+/TX-, GND, VCC для USB 2.0/3.0 та Type-C;
- формування візуальних повідомлень – індикаторів успіху (зелений), помилки (червоний), детальної інформації про перевірені лінії.

Завантажити програмний код (Додаток А), Arduino IDE та перевірити код на наявність синтаксичних помилок, натиснути кнопку «Перевірити» (значок із галочкою на панелі інструментів), яка запускає процес компіляції. У разі відсутності помилок на наступному етапі натиснути кнопку «Завантажити» (значок із правою стрілкою), що передає скомпільований код на мікроконтролер через USB-кабель. Після завершення завантаження пристрій перезавантажується, і програма починає працювати автоматично.

Далі від'єднавши USB-кабель або кабель живлення від Arduino Mega, обережно вставити екран у Mega, переконавшись, що контакти вирівняні. Підключити живлення до USB-порту та перевірте, чи правильно відображається заставка, а меню сенсорного екрана працює належним чином (рис. 3.8).



Рисунок 3.8 – Перевірка екрану

3.3 Тестування пристрою

Після завершення збірки та завантаження програмного забезпечення пристрій переходить до етапу тестування USB-кабелів різних стандартів. Перевірка проводиться з урахуванням особливостей кожного типу кабелю – USB 2.0, USB 3.0 та USB Type-C, адже кожен із них має специфічну розпіновку, кількість сигнальних ліній і електричні характеристики.

Робота пристрою для тестування USB-кабелів на основі Arduino Mega організована у вигляді послідовного алгоритму, що складається з кількох ключових етапів. Після подачі живлення мікроконтролер проходить фазу ініціалізації – відбувається налаштування портів вводу/виводу, ініціалізація сенсорного TFT-дисплея, запуск необхідних бібліотек для обробки сенсорних натискань та підготовка інтерфейсу користувача.

Далі пристрій переходить у режим очікування, під час якого він постійно перевіряє підключення кабелю до тестових роз'ємів. Як тільки кабель підключено, система запускає опитування контактів. Для кожного типу USB (2.0, 3.0, Type-C) визначається набір ліній, які потрібно перевірити, включаючи лінії живлення (VCC, GND), сигнали даних (D+/D-, TX+/TX-, RX+/RX-) та допоміжні контакти.

Під час перевірки мікроконтролер послідовно надсилає тестові сигнали або вимірює рівні напруги на окремих контактах, порівнюючи їх з еталонними значеннями. Якщо сигнал відхиляється від норми (наприклад, обрив або коротке замикання), система фіксує це як помилку. Всі результати передаються на дисплей: користувач бачить графічне відображення перевірених ліній, статус кожного елемента (зокрема, підсвічуванням кольорів).

Після завершення тестування система повертається у режим очікування, готуючись до наступного циклу перевірки. Якщо під час роботи виникають помилки – наприклад, відсутність підключення або некоректна робота сенсорного шару – передбачено механізм скидання (через кнопку Reset або програмний перезапуск).

Результати тестування відображаються на сенсорному TFT-дисплеї у вигляді кольорових індикаторів (наприклад, зелений – успіх, червоний – помилка) та текстових повідомлень із деталями про кожен перевірений контакт. У разі виявлення несправностей система підказує, на якій саме лінії виникла проблема, що значно спрощує подальшу діагностику та ремонт кабелю.

Тестування USB Type-C – USB Type-C (рис. 3.9) відбувається шляхом послідовного опитування всіх пар контактів, що підведені до модуля роз'ємів Type-C. Arduino Mega послідовно перевіряє наявність напруги на лініях живлення, правильність розведення сигналів, а також симетричність підключення, оскільки роз'єм Type-C може вставлятися будь-якою стороною. Окремо аналізуються лінії CC (Configuration Channel), які визначають роль пристрою (джерело чи споживач живлення), та SBU (Sideband Use), що використовуються в деяких режимах, наприклад, для передачі аналогових аудіосигналів.



Рисунок 3.9 – Тестування USB Type-C – USB Type-C

Під час тестування кабелю USB-A – USB Type-C (рис. 3.10) послідовно перевіряються основні лінії живлення (VCC, GND), диференціальні пари D+/D–, а також у разі наявності – високошвидкісні лінії TX+/TX–, RX+/RX–. Особливу увагу приділяють правильності підключення Configuration Channel (CC) на стороні Type-C, адже вони визначають, який кінець кабелю бере на себе роль джерела живлення або даних.



Рисунок 3.10 – Тестування USB-A– USB Type-C

Під час тестування кабелів RJ45 – RJ45 (рис. 3.11) пристрій послідовно перевіряються всі 8 пар контактів, визначаючи цілісність кожної лінії, наявність обривів, коротких замикань, а також правильність послідовності підключення. Arduino Mega, підключений до модулів RJ45, через цифрові порти опитує сигнали на кожному контакті, фіксуючи співпадіння контактів між двома кінцями кабелю.



Рисунок 3.11 – Тестування RJ45 – RJ45

Описано процес реалізації пристрою для тестування USB-кабелів і трасування з'єднань, включаючи етапи збірки, монтажу, програмування мікроконтролера Arduino Mega та проведення експериментального тестування. Було розглянуто алгоритм роботи системи, особливості перевірки різних типів кабелів – USB Type-A, USB Type-C, комбінованих A-C, а також мережевих кабелів RJ45.

Розроблений пристрій успішно пройшов перевірку на реальних зразках, продемонструвавши здатність визначати обриви, короткі замикання, неправильне розведення сигналів та інші поширені проблеми з кабелями. Використання сенсорного TFT-дисплея значно підвищило зручність взаємодії з системою, а гнучкість Arduino-платформи дозволила реалізувати адаптивні алгоритми під різні стандарти кабелів.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було повністю реалізовано всі поставлені завдання, кожне з яких зробило свій внесок у створення повнофункціонального пристрою для тестування USB-кабелів та трасування з'єднань.

Проведено ґрунтовний аналіз існуючих рішень та пристроїв, доступних на ринку: розглянуто як комерційні тестери, так і саморобні рішення на основі Arduino та інших платформ. Це дало змогу визначити, що більшість готових пристроїв мають обмежену функціональність або високу вартість, а тому існує актуальна потреба в розробці недорогого універсального кабельного тестера з можливістю адаптації під різні стандарти.

Детально вивчено апаратні та програмні можливості платформи Arduino, зокрема Arduino Mega, яка має достатню кількість портів для підключення численних сигнальних ліній, підтримує роботу з TFT-сенсорними дисплеями, модулем живлення, а також забезпечує можливість розширення функціональності через бібліотеки. Ці переваги дозволили закласти основу для створення багатофункціональної системи.

При аналізі функціональних потреб та технічних можливостей було обґрунтовано вибір основних апаратних компонентів, включаючи Arduino Mega, TFT-сенсорний дисплей, модулі USB Type-A та Type-C, корпус та допоміжні елементи для надійного складання та безпечної роботи.

Розроблене програмне забезпечення забезпечує повний цикл перевірки кабелів, включаючи опитування сигнальних ліній, аналіз контактів, виявлення обривів, коротких замикань та помилок розведення, а також взаємодію з користувачем через графічний сенсорний інтерфейс. Програма також передбачає захист від програмних помилок і можливість перезавантаження пристрою для коректної роботи.

Проведено тестування створеного пристрою, що дозволило оцінити його переваги: гнучкість, масштабованість, зручність використання, простота обслуговування, низька собівартість у порівнянні з комерційними аналогами. Серед виявлених недоліків – потреба в ретельному налаштуванні сенсорного інтерфейсу, обмеження швидкодії при тестуванні великої кількості високошвидкісних ліній), а також залежність від стабільності програмного коду.

Загалом виконана кваліфікаційна робота підтвердила доцільність розробки власного пристрою на базі Arduino для ефективного тестування кабелів і відкрила перспективи для подальшого вдосконалення, наприклад, додавання підтримки Power Delivery, аналізу якісних параметрів сигналу та підключення до комп'ютера для розширеного протоколювання результатів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кішта Д. М., Гринюк С. В. Пристрій для тестування кабелів на основі платформи Arduino. *Програмне та апаратне забезпечення в інформаційних технологіях: матеріали Міжнар. наук.-практ. конф. молодих вчених та студентів* (6 травня 2025 р). / відп. ред. Т.В. Терлецький. Вип. 1. Луцьк: ЛНТУ, 2025. с.80-81.
2. Fnirsi FNB58 USB Tester (Review). *Elektormag*. URL: <https://www.elektormagazine.com/review/fnirsi-fnb58-review> (дата звернення: 08.02.2025).
3. Review, Tested: Fnirsi FNB58 USB Fast Charge Tester. *Goughlui*. URL: <https://goughlui.com/2023/04/29/review-tested-fnirsi-fnb58-usb-fast-charge-tester/> (дата звернення: 08.02.2025).
4. USB cables diagnostic tester Mechanic DT3. *Parts4laptops*. URL: <https://parts4laptops.eu/en/usb-testers/3624-usb-cables-diagnostic-tester-mechanic-dt3.html> дата звернення: 08.02.2025).
5. Mechanic DT3 - Data Cable Tester. Fixshop. *Parts4laptops*. URL: <https://www.fixshop.eu/tools-test-measuring-instruments/mechanic-dt3-data-cable-tester-lightning-usb-c-microusb/> (дата звернення: 08.02.2025).
6. Treedix USB Cable Tester Board. Fixshop. *Treedix*. URL: <https://treedix.com/> (дата звернення: 08.02.2025).
7. Noyafa NF-8209. *Un-it*. URL: <https://un-it.kz/p81450999-mnogofunktionalnyj-tester-8209.html> (дата звернення: 08.02.2025).
8. Тестер LAN дроти з USB KYS0411. *Prom*. URL: <https://prom.ua/ua/p1527819893-tester-lan-provoda.html> (дата звернення: 08.02.2025).
9. Розпіновка USB: все про кольори дротів і схеми підключення для роз'ємів різного типу. *Telemart*. URL: [ua/reviews/rozpinovka-usb-vse-pro-kolory-drotiv-i-skhemu-pidklyuchennya-dlya-rozymiv-riznoho-typu.html](https://telemart.ua/reviews/rozpinovka-usb-vse-pro-kolory-drotiv-i-skhemu-pidklyuchennya-dlya-rozymiv-riznoho-typu.html) (дата звернення: 08.02.2025).

10. USB 3.0. Telemart. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/USB_3.0 (дата звернення: 08.02.2025).
11. Диференційний сигнал. *Вікіпедія*. URL: <https://uk.wikipedia.org/wiki/> (дата звернення: 08.02.2025).
12. Arduino Mega 2560. *Wiki*. URL: https://wiki.tntu.edu.ua/Arduino_Mega_2560/ (дата звернення: 15.03.2025).
13. Arduino Mega 2560. *Ampermarket*. URL: <https://ampermarket.kz/arduino/analog-arduino-mega2560/> (дата звернення: 15.03.2025).
14. Arduino ОПТА. Arduino. URL: <https://www.arduino.cc/> (дата звернення: 15.03.2025).
15. TFT LCD 320x240 резистивний сенсорний екран. *Arduino*. URL: <https://surl.li/zoqiig> (дата звернення: 15.04.2025).

ДОДАТКИ

Додаток А

Код мікроконтролера

```

define LCD_CS A3 // Chip Select goes to Analog 3
#define LCD_CD A2 // Command/Data goes to Analog 2
#define LCD_WR A1 // LCD Write goes to Analog 1
#define LCD_RD A0 // LCD Read goes to Analog 0
#define LCD_RESET A4 // Can alternately just connect to Arduino's reset pin

#include <SPI.h> // f.k. for Arduino-1.5.2
#include "Adafruit_GFX.h" // Hardware-specific library
#include <MCUFRIEND_kbv.h>
MCUFRIEND_kbv tft;
//#include <Adafruit_TFTLCD.h>
//Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);

#include <TouchScreen.h>

#define YP A3 // must be an analog pin, use "An" notation!
#define XM A2 // must be an analog pin, use "An" notation!
#define YM 9 // can be a digital pin
#define XP 8 // can be a digital pin

// Revised Touch Calibration numbers
#define TS_MINX 150
#define TS_MINY 130
#define TS_MAXX 850
//#define TS_MAXY 927
#define TS_MAXY 880

// For better pressure precision, we need to know the resistance
// between X+ and X- Use any multimeter to read it
// For the one we're using, its 300 ohms across the X plate
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);

// Assign human-readable names to some common 16-bit color values:
#define BLACK 0x0000
#define BLUE 0x001F
#define RED 0xF800
#define GREEN 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF

```

```

#ifndef min
#define min(a, b) (((a) < (b)) ? (a) : (b))
#endif

int scangraphic = 0; // Pointer to coordinates of graphic being printed within
portscan array

boolean screendrawn = false; // Used to minimise screen refreshes

boolean manualmode = false; // Flag used to indicate whether in manual mode or
scan mode

boolean manualconnectorsdrawn = false; // Stops endless of redrawing of screen

// These variables store a count of connections detected for each configuration of
USB port on left and right
byte legacy_scan_count = 1;

byte manualmenuselection = 0; // Determines which menu item is selected. 0 is no
selection

// These variables point to the first row of text for this port when drawing screen
byte usbCleftpointer = 0; //
byte usbAleftpointer = 70; //
byte usbA3leftpointer = 70; //
byte usbMinirighttpointer = 160;
byte usbMicrorighttpointer = 160;
byte usbCrightpointer = 0;

// These flags indicate if one of 9 valid cable configurations has been detected
and are used as a future baseline for refresh of the screen

boolean usbAusbMinicable_det = false;
boolean usbAusbMicrocable_det = false;
boolean usbAusbC_det = false;

boolean usbA3usbMinicable_det = false;
boolean usbA3usbMicrocable_det = false;
boolean usbA3usbC_det = false;

boolean usbCusbMinicable_det = false;
boolean usbCusbMicrocable_det = false;
boolean usbCusbC_det = false;

boolean manualscreeendrawn = false;
boolean autoscreendrawn = false;

byte scanicon = 0; // Icon postion scan indicator

byte current_scan_count = 0; // variable to count number of active connections

```

```

boolean ignoreccrpins = true; // User defined Variable used to flag mssing out CCR
pins on USBC cables due to internal cable marking behaviour
boolean blockccrpins = false; // Flag used to indicate CCR pins have been detected
and should be ignored based on condition set by user

// Create a Data Structure for each Left hand side
struct leftside {
    byte pin;          // 24 input pins
    byte cleft;       // The number of left pin connections detected for this pin
    byte cright;      // The number of right pin connections detected for this pin
    boolean pinprinted; // Flag to indicate the pin has been displayed already on the
screen and should not be duplicated
    byte pinleft[24]; // Store an integer for every pin connected on the left to this
pin
    byte pinright[24]; // Store an integer for every pin connected on the left to
this pin

    // char* pinleft; // Sring containing 2 characters for every pin connected on the
left to this pin
    // char* pinright; // Sring containing 2 characters for every pin connected on
the right to this pin

}
leftside[24] = {
{ 46,0,0,false},
{ 48,0,0,false},
{ 50,0,0,false},
{ 52,0,0,false},
{ 14,0,0,false},
{ 15,0,0,false},
{ 16,0,0,false},
{ 17,0,0,false},
{ 18,0,0,false},
{ 19,0,0,false},
{ 20,0,0,false},
{ 21,0,0,false},
{ 22,0,0,false},
{ 24,0,0,false},
{ 26,0,0,false},
{ 28,0,0,false},
{ 30,0,0,false},
{ 32,0,0,false},
{ 34,0,0,false},
{ 36,0,0,false},
{ 38,0,0,false},
{ 40,0,0,false},
{ 42,0,0,false},
{ 44,0,0,false}
};

```

```

// Create a Data Structure for each Right hand side
struct rightside {
    byte pin;          // 24 input pins
    byte cleft;       // The number of left pin connections detected for this pin
    byte cright;      // The number of right pin connections detected for this pin
    boolean pinprinted; // Flag to indicate the pin has been displayed already on the
                        // screen and should not be duplicated
    byte pinleft[24]; // Store an integer for every pin connected on the left to this
pin
    byte pinright[24]; // Store an integer for every pin connected on the left to
this pin
    /*
    char* pinleft; // Sring containing 2 characters for every pin connected on the
left to this pin
    char* pinright; // Sring containing 2 characters for every pin connected on the
right to this pin
    */
}

rightsided[24] = {
{ 51,0,0,false},
{ 49,0,0,false},
{ 47,0,0,false},
{ 45,0,0,false},
{ 62,0,0,false},
{ 63,0,0,false},
{ 64,0,0,false},
{ 65,0,0,false},
{ 66,0,0,false},
{ 67,0,0,false},
{ 68,0,0,false},
{ 69,0,0,false},
{ 23,0,0,false},
{ 25,0,0,false},
{ 27,0,0,false},
{ 29,0,0,false},
{ 31,0,0,false},
{ 33,0,0,false},
{ 35,0,0,false},
{ 37,0,0,false},
{ 39,0,0,false},
{ 41,0,0,false},
{ 43,0,0,false},
{ 53,0,0,false}
};

```

```

// Create a Data Structure for each USB type
// Sensor Number , Pin Number, Status, Text description of pin and function

struct usbCleft {
    byte pinnumber;           // 62 to 69, 14 to 29 sensor pin addresses
    char* function;          // Text description of function and logical pin number
    byte refnum;             // Position of pin in rightside or leftside array
}

usbCleft[24] = {

{ 46,"A1  GND",0},
{ 48,"A2  TX1+",1},
{ 50,"A3  TX1-",2},
{ 52,"A4  vBUS",3},
{ 14,"A5  CC1",4},
{ 15,"A6  D+",5},
{ 16,"A7  D-",6},
{ 17,"A8  SBU1",7},
{ 18,"A9  vBUS",8},
{ 19,"A10 RX2-",9},
{ 20,"A11 RX2+",10},
{ 21,"A12 GND",11},
{ 22,"B1  GND",12},
{ 24,"B2  TX2+",13},
{ 26,"B3  TX2-",14},
{ 28,"B4  vBUS",15},
{ 30,"B5  CC2",16},
{ 32,"B6  D+",17},
{ 34,"B7  D-",18},
{ 36,"B8  SBU2",19},
{ 38,"B9  vBUS",20},
{ 40,"B10 RX1-",21},
{ 42,"B11 RX1+",22},
{ 44,"B12 GND",23}
};

struct usbCright {
    byte pinnumber;           // 30 to 53 sensor pin addresses
    char* function;          // Text description of function and logical pin number
    byte refnum;             // Position of pin in rightside or leftside array
}

usbCright[24] = {
{51,"GND  A1",0},
{49,"TX1+ A2",1},
{47,"TX1- A3",2},
{45,"vBUS A4",3},

```

```

{62,"CC1  A5",4},
{63,"D+   A6",5},
{64,"D-   A7",6},
{65,"SBU1 A8",7},
{66,"vBUS A9",8},
{67,"RX2- A10",9},
{68,"RX2+ A11",10},
{69,"GND  A12",11},
{23,"GND  B1",12},
{25,"TX2+ B2",13},
{27,"TX2- B3",14},
{29,"vBUS B4",15},
{31,"CC2  B5",16},
{33,"D+   B6",17},
{35,"D-   B7",18},
{37,"SBU2 B8",19},
{39,"vBUS B9",20},
{41,"RX1- B10",21},
{43,"RX1+ B11",22},
{53,"GND  B12",23}
};

struct usbAleft {
    byte pinnumber;    // 30 to 33 sensor pin addresses
    char* function;    // Text description of function and logical pin number
    byte refnum;       // Position of pin in rightside or leftside array
}

usbAleft[4] = {
{ 36 ,"4   GND",19},
{ 38 ,"3   D+",20},
{ 40 ,"2   D-",21},
{ 42 ,"1   Vcc",22}
};

struct usbA3left {
    byte pinnumber;    // 26 to 42 sensor pin addresses
    char* function;    // Text description of function and logical pin number
    byte refnum;       // Position of pin in rightside or leftside array
}

usbA3left[9] = {
{ 26,"9 StdA_SSTX+",14},
{ 28,"8 StdA_SSTX-",15},
{ 30,"7 GND_DRAIN ",16},
{ 32,"6 StdA_SSRX+",17},
{ 34,"5 StdA_SSRX-",18},
{ 36,"4 USB2  GND",19},
{ 38,"3 USB2  D+",20},

```

```

{ 40,"2 USB2   D-",21},
{ 42,"1 USB2  vBUS",22}
}
struct usbMiniright {
    byte pinnumber;      // 34 to 42 sensor pin addresses
    char* function;      // Text description of function and logical pin number
    byte refnum;         // Position of pin in rightside or leftside array
}

usbMiniright[5] = {
{62,"GND  5",4},
{63,"ID   4",5},
{64,"D+   3",6},
{65,"D-   2",7},
{66,"Vcc  1",8}
};

struct usbMicroright {
    byte pinnumber;      // sensor pin addresses
    char* function;      // Text description of function and logical pin number
    byte refnum;         // Position of pin in rightside or leftside array
}

usbMicroright[5] = {
{27,"GND   5",14},
{29,"ID    4",15},
{31,"D+    3",16},
{33,"D-    2",17},
{35,"Vcc   1",18}
};

struct usbBright {
    byte pinnumber;      // sensor pin addresses
    char* function;      // Text description of function and logical pin number
    byte refnum;         // Position of pin in rightside or leftside array
}

usbBright[4] = {
{33,"GND  4",17},
{35,"D+   3",18},
{37,"D-   2",19},
{39,"vBUS 1",20}
};

struct usbB3right {
    byte pinnumber;      // sensor pin addresses
    char* function;      // Text description of function and logical pin number

```

```

byte refnum;          // Position of pin in rightside or leftside array

}
usbB3right[9] = {
{ 23,"StdA_SSTX+ 9",12},
{ 25,"StdA_SSTX- 8",13},
{ 27,"GND_DRAIN 7",14},
{ 29,"StdA_SSRX+ 6",15},
{ 31,"StdA_SSRX- 5",16},
{ 33,"GND USB2 4",17},
{ 35,"D+ USB2 3",18},
{ 37,"D- USB2 2",19},
{ 39,"vBUS USB2 1",20}
};

struct RJ45left {
    byte pinnumber;      // 26 to 42 sensor pin addresses
    char* function;     // Text description of function and logical pin number
    byte refnum;        // Position of pin in rightside or leftside array
}

RJ45left[8] = {
    {22,"1          W/Or",12},
    {24,"2          Or  ",13},
    {26,"3          W/Gr",14},
    {28,"4          Bl  ",15},
    {30,"5          W/Bl",16},
    {32,"6          Gr  ",17},
    {34,"7          W/Br",18},
    {36,"8          Br  ",19}
};

struct RJ45right {
    byte pinnumber;      // 26 to 42 sensor pin addresses
    char* function;     // Text description of function and logical pin number
    byte refnum;        // Position of pin in rightside or leftside array
}

RJ45right[8] = {
{62," 1",4},
{63," 3",5},
{64," 2",6},
{65," 5",7},
{66," 4",8},
{67," 7",9},
{68," 6",10},
{69," 8",11}
};

```

```

struct usbMicro3right {
  byte pinnumber;      // 26 to 42 sensor pin addresses
  char* function;     // Text description of function and logical pin number
  byte refnum;        // Position of pin in rightside or leftside array
}

usbMicro3right[10] = {
{ 23,"SSRX+   10",12},
{ 25,"SSRX-   9",13},
{ 27,"GND_DRAIN 8",14},
{ 29,"SSTX+   7",15},
{ 31,"SSTX-   6",16},
{ 33,"GND     5",17},
{ 35,"ID      4",18},
{ 37,"D-     2",19},
{ 39,"vBUS    1",20},
{ 41,"D+     3",21}
};

//Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);

#define BOX 78 // Block to clear screen
#define BOXSIZE 80
#define BUTTON 20

byte lsc;

byte tempreading = 0;

void setup() {
  Serial.begin(9600);
  uint32_t when = millis();
  // while (!Serial) ; //hangs a Leonardo until you connect a Serial
  if (!Serial) delay(5000); //allow some time for Leonardo
  Serial.println("Serial took " + String((millis() - when)) + "ms to start");
  // tft.reset(); //hardware reset
  uint16_t ID = tft.readID(); //
  Serial.print("ID = 0x");
  Serial.println(ID, HEX);
  if (ID == 0xD3D3) ID = 0x9481; // write-only shield
  // ID = 0x9329;

```

```
// force ID
  tft.begin(ID);
  tft.setRotation(1); // Rotation of 1 for 0x9325 and 3 for IL9341 screen
  tft.fillScreen(BLACK);

pinMode(13, OUTPUT);

#define MINPRESSURE 5
#define MAXPRESSURE 400

splashscreen();

resetpins(); // Resets all sensor pins to pull up condition
}
```