

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 кібербезпеки

(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

СИСТЕМА ДІАГНОСТИКИ АВТОМОБІЛЯ НА БАЗІ
МІКРОКОНТРОЛЕРНОЇ ТЕХНІКИ

CAR DIAGNOSTICS SYSTEM BASED ON MICROCONTROLLER
TECHNOLOGY

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІсз-21
Ворон Назар Ігорович

(підпис)

Керівник:
к.т.н., доцент
Костючко Сергій Миколайович

(підпис)

Кваліфікаційну роботу
допущено до захисту
« 11 » червня 2024 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

проф. Н.Черняшук

« 10 » 01 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Ворону Назару Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Система діагностики автомобіля на базі і мікроконтролерної техніки

Керівник роботи к.т.н., доцент Костючко Сергій Миколайович

затверджені наказом закладу вищої освіти від «30» грудня 2023 року № 459/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 11.06.2024р.

3. Вихідні дані до роботи Джерелом розробки є науково-технічна література та вітчизняні 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

Обґрунтування актуальності теми кваліфікаційної роботи

Вибір та налаштування технічних засобів

Тестування та відлагодження розробленого пристрою

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Обґрунтування актуальності теми кваліфікаційної роботи</i>	<i>Костючко С.М., доцент</i>		
<i>Вибір та налаштування технічних засобів</i>	<i>Костючко С.М., доцент</i>		
<i>Тестування та відлагодження розробленого пристрою</i>	<i>Костючко С.М., 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., асистент</i>		

7. Дата видачі завдання 10.01.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Провести огляд літератури із досліджуваної проблеми</i>	до 15.02.2024 р.	Виконано
2.	<i>Провести аналіз практичного застосування технічних засобів діагностики автомобіля</i>	до 15.03.2024 р.	Виконано
3.	<i>Розробка та впровадження пристрою для діагностики автомобіля</i>	до 04.05.2024 р.	Виконано
4.	<i>Висновки та пропозиції</i>	до 07.05.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 10.05.2024 р.	Виконано
6.	<i>Формування додатків</i>	до 15.05.2024 р.	Виконано
7.	<i>Оформлення ілюстративного матеріалу</i>	до 20.05.2024 р.	Виконано
8.	<i>Нормоконтроль</i>	до 01.06.2024 р.	Виконано
9.	<i>Інструментальна перевірка на академічний плагіат</i>	до 04.06.2024 р.	Виконано
10.	<i>Представлення кваліфікаційної роботи бакалавра до захисту</i>	до 11.06.2024 р.	Виконано

Здобувач вищої освіти

(підпис)

Ворон Н.І.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Костючко С.М.

(прізвище, ініціали)

АНОТАЦІЯ

Ворон Н.І. Система діагностики автомобіля на базі мікроконтролерної техніки. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2024. 65 с.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел.

Перший розділ присвячено огляду предметної області, а саме проаналізовано сучасний стан автомобільної промисловості та технологій діагностики транспортних засобів. Підкреслюється важливість використання мікроконтролерної техніки для підвищення ефективності та точності діагностичних процедур.

Другий розділ розкриває процес вибору відповідних мікроконтролерів, необхідних для створення системи діагностики. Обговорюються технічні характеристики, можливості та обмеження різних компонентів, їхнє взаємодія та інтеграція в систему.

Третій розділ описує процес тестування створеної системи діагностики. Також в цьому розділі аналізуються отримані результати, оцінюється ефективність розробленої системи та робляться висновки щодо її практичного застосування.

Мета роботи – розробка системи діагностики автомобіля, що базується на використанні мікроконтролерної техніки, здатної забезпечити точне виявлення і аналіз можливих несправностей в критичних вузлах і агрегатах автомобіля.

Об'єкт дослідження – процес діагностики технічного стану автомобіля.

Предмет дослідження – система діагностики автомобіля на базі мікроконтролерної техніки.

Ключові слова: Raspberry Pi, мікроконтролер, віддалений робочий стіл, діагностика, програмування.

ANNOTATION

Kyndiuk M.R. Car diagnostics system based on microcontroller technology. Manuscript.

Qualifying work of a bachelor of EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2024. 55 p.

Qualification work consists of an introduction, three chapters, conclusions and a references.

The first chapter is dedicated to the overview of the subject area, namely, the current state of the automotive industry and vehicle diagnostic technologies is analyzed. The importance of using microcontroller technology to increase the efficiency and accuracy of diagnostic procedures is emphasized.

The second section describes the process of selecting the appropriate microcontrollers needed to create a diagnostic system. Technical characteristics, capabilities and limitations of various components, their interaction and integration into the system are discussed.

The third section describes the process of testing the created diagnostic system. This section also analyzes the obtained results, evaluates the effectiveness of the developed system, and draws conclusions regarding its practical application.

The purpose of the work is to develop a car diagnostic system based on the use of microcontroller technology, capable of providing accurate detection and analysis of possible malfunctions in critical components and units of the car.

The object of the research is the process of diagnosing the technical condition of the car.

The subject of research is a car diagnostic system based on microcontroller technology.

Keywords: Raspberry Pi, microcontroller, remote desktop, diagnostics, programming.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ОБГРУНТУВАННЯ АКТУАЛЬНОСТІ ТЕМИ КВАЛІФІКАЦІЙНОЇ РОБОТИ.....	9
1.1 Огляд сучасних технологій діагностики автомобільного транспорту	9
1.2 Аналіз існуючих пристроїв для віддаленої діагностики	11
1.3 Обґрунтування вибору технічних рішень	15
1.4 Постанова задачі.....	18
РОЗДІЛ 2 ВИБІР ТА НАЛАШТУВАННЯ ТЕХНІЧНИХ ЗАСОБІВ.....	20
2.1 Проектування моделі на основі обраних компонентів.....	20
2.2 Встановлення програмного забезпечення.....	27
РОЗДІЛ 3 ТЕСТУВАННЯ ТА ВІДЛАГОДЖЕННЯ РОЗРОБЛЕНОГО ПРИСТРОЮ	45
3.1 Тестування роботи модулів пристрою	45
3.2 Алгоритм пошуку несправностей.....	55
3.3 Можливості подальшого удосконалення та модифікацій модулів пристрою.....	61
ВИСНОВКИ	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	64

ВСТУП

У сучасному світі автомобільна промисловість стрімко розвивається, пропонуючи все нові технологічні рішення для підвищення комфорту, безпеки та ефективності транспортних засобів. Однією з ключових задач сучасного автомобілебудування є підтримка належного технічного стану автомобіля, що вимагає високоефективних систем діагностики. Використання мікроконтролерної техніки в системах діагностики автомобілів дозволяє значно підвищити точність і швидкість обробки даних, а також забезпечити більш глибокий контроль за станом транспортного засобу.

Актуальність дослідження полягає в потребі розробки ефективних, доступних і точних систем діагностики автомобілів, які б використовували сучасні досягнення в області мікроконтролерної техніки для оперативного виявлення та діагностики несправностей, мінімізації простоїв автомобіля та зниження витрат на обслуговування.

Мета роботи – розробка системи діагностики автомобіля, що базується на використанні мікроконтролерної техніки, здатної забезпечити точне виявлення і аналіз можливих несправностей в критичних вузлах і агрегатах автомобіля.

Об'єкт дослідження – процес діагностики технічного стану автомобіля.

Предмет дослідження – система діагностики автомобіля на базі мікроконтролерної техніки.

Для досягнення мети роботи були поставлені наступні завдання:

1. Аналіз існуючих систем діагностики автомобілів та визначення їхніх основних недоліків.
2. Розробка концепції системи діагностики на базі мікроконтролерів.
3. Вибір компонентів і розробка схеми пристрою.
4. Програмування мікроконтролера для збору та аналізу даних з датчиків автомобіля.
5. Тестування системи на автомобілі.

Ця робота спрямована на створення надійної та функціональної системи, яка б дозволила автомобілістам своєчасно отримувати важливу інформацію про стан своїх автомобілів, підвищуючи тим самим безпеку дорожнього руху та ефективність використання транспортних засобів.

РОЗДІЛ 1

ОБГРУНТУВАННЯ АКТУАЛЬНОСТІ ТЕМИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

1.1 Огляд сучасних технологій діагностики автомобільного транспорту

1.1.1 Системи віддаленого доступу

Віддалений доступ поділяється на фізичну і програмну складові. Фізична складова включає провідні з'єднання, такі як Ethernet або оптоволокно, та інтерфейси, такі як USB, VGA, DVI, HDMI, які вимагають знерухомилення при використанні. Бездротові засоби, наприклад Wi-Fi і Bluetooth, надають більшу свободу переміщення завдяки здатності підтримувати зв'язок на відстані.

Прогрес у розробці програмного забезпечення зробив використання цих технологій простішим, зменшивши час, необхідний для навчання, та полегшивши впровадження інновацій у різноманітні галузі. Програми варіюються від універсальних до тих, що спеціалізуються на конкретних завданнях, та пропонуються як у безкоштовних, так і у платних версіях. Їх встановлення на особистий комп'ютер зазвичай швидке та нескладне, а доступ контролюється через введення унікального ідентифікатора та пароля.

Популярні програмні засоби, такі як TeamViewer та AnyDesk, доступні для загального використання та супроводжуються докладними настановами, що допомагають новим користувачам легко освоїти ці інструменти. Ці програми забезпечують інструкції через свої веб-сайти, де користувачі можуть знайти корисні поради для ефективного використання.

1.1.2 Сучасні технології в діагностиці автотранспорту

Діагностика автомобілів включає в себе візуальний огляд та аналіз технічного стану, що вимагає певних знань і досвіду. Цей процес може бути складним і займати багато часу. З метою спрощення цих процедур були розроблені спеціалізовані програми, які використовуються як професійними

механіками, так і автоентузіастами. Спектр таких програмних рішень широкий і постійно розширюється, оскільки інженери прагнуть створити унікальні технології, адаптовані до конкретних автомобілів.

Наприклад, багато програм розробляються для корпорацій, які володіють декількома автобрендами, такими як VAG (Volkswagen Aktiengesellschaft), що включає Volkswagen, Audi, SEAT, Skoda та інші. Інструмент, як VAG EEPROM (рис. 1.1), використовується для діагностики виробів цих брендів.

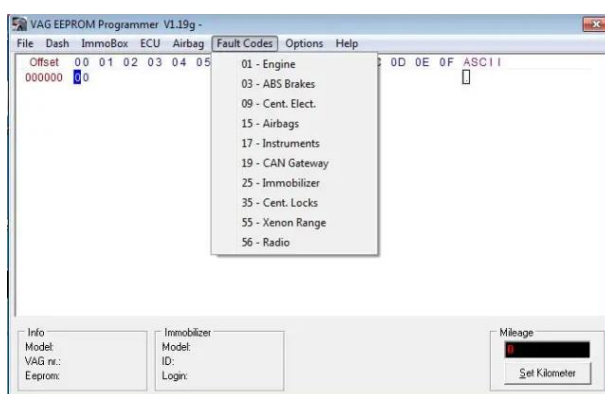


Рисунок 1.1 – Вікно програми VAG EEPROM [1]

Програма TOAD PRO (рис. 1.2) здатна діагностувати близько 15,000 типових проблем і відобразити дані через графіки та таблиці, що робить її незамінною у роботі автомайстрів.



Рисунок 1.2 – Інтерфейс програми TOAD PRO [1]

Враховуючи що смартфони і планшети завжди під рукою, вони також можуть використовуватись для моніторингу стану двигуна за допомогою програм як Torque Pro (рис. 1.3), яка дозволяє контролювати блок управління двигуном і збирати дані з підключених датчиків, а також користуватись GPS і акселерометром пристрою.



Рисунок 1.3 – Інтерфейс програми Torque Pro [1]

1.2 Аналіз існуючих пристроїв для віддаленої діагностики

Комп'ютерна діагностика автомобілів, відома як OBD (on-board diagnostics), є універсальним методом перевірки різних систем транспортного засобу. Це дозволяє відображати результати діагностики на панелі приладів, які також використовуються механіками для уточнення ремонтних робіт. Впровадження систем OBD почалося у 1980-х роках, і з того часу технологія значно еволюціонувала, використовуючи стандартизовані цифрові інтерфейси для точного моніторингу стану автомобіля та ідентифікації проблем через діагностичні коди несправностей (DTC).

Існують такі основні інтерфейси і версії OBD:

– AL DL (Assembly Line Diagnostic Link): Рання система від General Motors, яка не стала загальним стандартом, але заклала основи для подальших розробок в OBD. Використовувала різні типи роз'ємів (5, 10, і 12 контактів) та

швидкості передачі даних від 160 до 8192 біт/с, забезпечуючи двосторонній зв'язок з модулем управління двигуном (PCM).

– OBD-I: Версія, що зобов'язала виробників авто встановлювати системи контролю за викидами.

– OBD 1.5: Проміжна версія між OBD-I та OBD-II, яку використовував General Motors на деяких моделях у середині 90-х.

– OBD-II: Сучасний стандарт, запроваджений у середині 90-х, який дозволяє комплексну діагностику систем автомобіля, включаючи двигун і кузов, а також моніторинг мережі управління.

– EOBD (European On-Board Diagnostics): Європейська версія OBD-II, розроблена згідно з екологічними вимогами ЄС, введена для автомобілів стандарту EURO 3.

– EOBD2: Маркетинговий термін для позначення розширених функцій, що іноді використовуються виробниками для підкреслення особливостей обладнання, не включених до стандартного пакету EOBD.

– JOBD (Japan On-Board Diagnostics): Варіант OBD-II, призначений для японського ринку.

Специфікація OBD-II передбачає використання уніфікованого діагностичного роз'єму (DLC - Diagnostic Link Connector), що має 16 контактів і розташовується так, щоб бути доступним для водія, зазвичай поблизу рульової колонки. Розташування виходів на SAE J1962 (рис. 1.4):

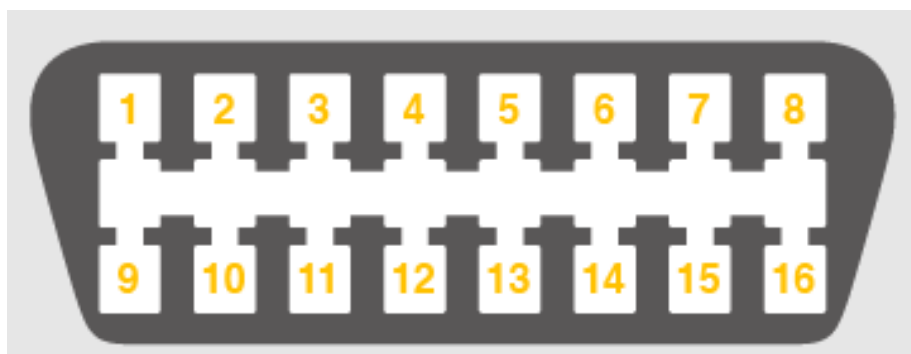


Рисунок 1.4 – Розташування виходів [2]

Призначення контактів діагностичного роз'єму описується наступним чином:

1. OEM (протокол виробника) - для власних потреб виробника.
2. Шина + (Bus positive Line) - для протоколів SAE-J1850 PWM, SAE-1850 VPW.
3. Заземлення кузова – забезпечує заземлення для кузова автомобіля.
4. Сигнальне заземлення – для заземлення сигналів.
5. Лінія CAN-High високошвидкісної шини CAN Highspeed (ISO 15765-4, SAEJ2284) – для високошвидкісних передач.
6. K-Line (ISO 9141-2 і ISO 14230) – для спілкування між різними системами автомобіля.
7. Лінія CAN-Low низькошвидкісної шини CAN Lowspeed – для низькошвидкісних передач.
8. Шина (Bus negative Line) – використовується в протоколах SAE-J1850 PWM, SAE-1850 VPW.
9. Лінія CAN-Low високошвидкісної шини CAN Highspeed (ISO 15765-4, SAEJ2284) – для високошвидкісних передач.
10. L-Line (ISO 9141-2 і ISO 14230) – альтернативний зв'язок.
11. Живлення +12В від АКБ – забезпечує електроживлення.

Конструкція роз'єму включає додаткові невикористані контакти, призначення яких може бути визначене виробником автомобіля. Основні діагностичні дані OBD-II дозволяють отримати інформацію з різних систем автомобіля, включно з блоком керування двигуном (ECU), і є ключовим ресурсом при діагностиці та ремонті.

Стандарт SAE J1979 встановлює методи запиту діагностичних даних через ідентифікатори параметрів (PID), доступні у ECU. Виробники можуть не використовувати всі стандартні PID, дозволяючи додавати власні специфічні PID, що розширює можливості діагностики за допомогою створення наборів нестандартних OBD-II PID, доступ до яких обмежений і регулюється Equipment and Tool Institute (ETI).

Серед сучасних діагностичних засобів також виділяються мультифункціональні планшети, наприклад, професійний мультимарочний сканер LAUNCH X-413 PADV II (рис. 1.5), що поєднує різні функції для забезпечення ефективної та швидкої діагностики.



Рисунок 1.5 – LAUNCH X-413 PADV [3]

Сучасний діагностичний сканер PAD V II включає в себе віддалену діагностику SmartLink, можливості онлайн програмування та підтримку додаткових модулів для розширення функціоналу, що робить його невід'ємним інструментом для фахівців у сфері автосервісу. Це повнофункціональний діагностичний центр, який неперервно оновлюється і покращується. Пристрій оснащений 13-дюймовим планшетом на базі операційної системи Android 9.0, восьмиядерним процесором Qualcomm, 8 Гб оперативної пам'яті та 256 Гб FLASH пам'яті, а також батареєю ємністю 19 Аг.

Сканер PAD V II здатний діагностувати електронні системи управління автомобілями з напругою 12V та 24V, включаючи транспортні засоби

європейського, американського та азіатського виробництва, як правосторонні, так і лівосторонні. Серед основних функцій сканера:

- Інтелектуальна діагностика.
- Підтримка системи ADAS.
- Автоматичне розпізнавання моделі автомобіля через сканування VIN.
- Ведення детального журналу діагностичних звітів.
- Швидкий тест для автоматичного виявлення діагностичних кодів помилок (DTC) у всіх блоках управління та створення звітів DTC.
- Читання та видалення кодів помилок систем автомобіля.
- Ідентифікація контрольних блоків.
- Читання актуальних параметрів систем.
- Перевірка роботи виконавчих механізмів.
- Адаптація блоків управління.
- Кодування блоків управління.
- 32 окремі програми для сервісних функцій у розділі RESET.
- Онлайн програмування та віддалена діагностика.

Цей пристрій не призначений для автомобілістів, які не займаються систематичним ремонтом чи розробкою автомобілів, адже його функціонал та можливості орієнтовані на професійне використання в автосервісах.

1.3 Обґрунтування вибору технічних рішень

Для розробки системи віддаленої діагностики автомобіля необхідно насамперед обзавестися адаптером OBD-II, який забезпечує підключення до бортової системи авто. На ринку існує чимало варіантів таких адаптерів, але серед них виділяється автосканер EML 327 (рис. 1.6), який користується великою популярністю через свою надійність, доступність та легкість у використанні.



Рисунок 1.6 – Сканер-адаптер ELM 327 [4]

Міні ELM327 OBD2 Bluetooth – це компактний сканер для автодіагностики, який підключається до діагностичного роз'єму OBD2 і синхронізується з відповідним програмним забезпеченням через Bluetooth. Ця особливість дозволяє використовувати сканер не лише з персональними комп'ютерами, але й з мобільними пристроями на Android та iOS, що забезпечує гнучкість у діагностиці автомобілів.

ELM327 Bluetooth міні-сканер є ефективним інструментом для діагностики автомобіля, підтримує всі стандартні протоколи OBD-II. Цей адаптер підходить для більшості легкових автомобілів з інжекторним двигуном, випущених після 1996 року, забезпечує можливість читання і очищення діагностичних кодів помилок, а також вимкнення індикатора МІЛ.

Сканер може відображати широкий спектр параметрів, включаючи обороти двигуна, навантаження на двигун, температуру охолоджуючої рідини, параметри паливної системи, швидкість автомобіля, показники споживання палива, абсолютний тиск в повітряному колекторі, кут випередження запалювання, температуру повітря, що всмоктується, масовий розхід повітря, положення дросельної заслінки та інші.

Для підключення та управління модулем OBD-II можна використати мікрокомп'ютер, наприклад Raspberry Pi 4 model B (рис. 1.7), що забезпечує зручність установки в автомобілі та доступ до портів.

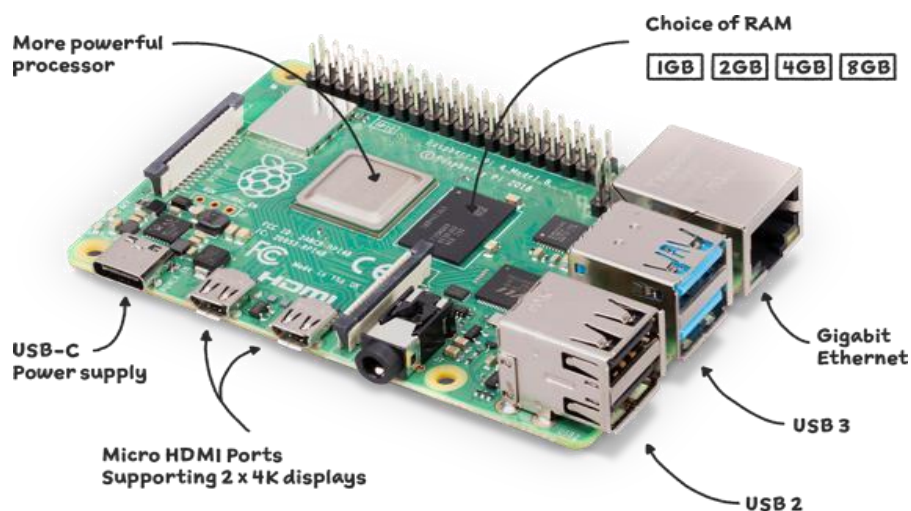


Рисунок 1.7 – Raspberry Pi 4 model B [5]

Пристрій має компактні розміри аналогічні до розмірів банківської картки (рис. 1.8).

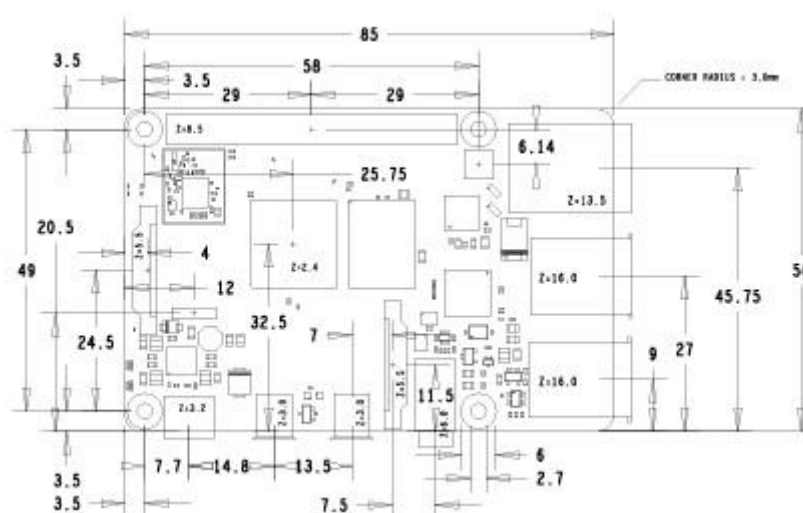


Рисунок 1.8 – Розміри Raspberry Pi 4 model B [5]

Raspberry Pi 4 Model B – це потужний міні-комп'ютер, оснащений чотириядерним процесором ARM-Cortex A72 з тактовою частотою 1.5GHz. Існує кілька варіантів конфігурації з різним обсягом оперативної пам'яті RAM: 1Гб, 2Гб, 4Гб та 8Гб, що забезпечує гнучкість у використанні залежно від потреб користувача.

Raspberry Pi 4 включає в себе широкий спектр портів і підключень:

- Бездротове з'єднання Wi-Fi 802.11 b/g/n/ac для стабільного доступу до інтернету.
- Bluetooth 5.0 з BLE, що підтримує бездротові з'єднання з низьким енергоспоживанням.
- Слот для SD-карти для зберігання даних і запуску операційних систем.
- Два порти мікро-HDMI, які підтримують виведення відео з роздільною здатністю до 4K при 60 кадрах за секунду.
- По два порти USB2 та USB3 для підключення периферійних пристроїв.
- Гігабітний Ethernet порт, який може бути удосконалений для підтримки PoE через додатковий PoE NAT.
- Порт для камери Raspberry Pi та порт для дисплея з підтримкою стандартів MIPI CSI та MIPI DSI відповідно.
- 28 контактів GPIO з підтримкою множини інтерфейсів, включаючи до 6 UART, 6 I2C, 5 SPI, а також інші інтерфейси для розширених проєктів.

Програмне забезпечення для Raspberry Pi 4 базується на наборі інструкцій ARMv8 та міцному стеку Linux. Система постійно підтримується та оновлюється, включаючи актуальні драйвери, стабільний користувацький інтерфейс і широкі можливості для використання GPU через стандартні API.

Для ефективної роботи Raspberry Pi 4 необхідний якісний блок живлення з роз'ємом USB-C, що забезпечує напругу 5 В при струмі 3 А. У разі підключення периферійних пристроїв, які споживають менше 500 мА, можливе використання блоку живлення з параметрами 5 В, 2.5 А.

1.4 Постанова задачі

Мета кваліфікаційної роботи полягає у створенні системи для спрощення процесу діагностики автомобілів за допомогою OBD-II приймача, який базується на мікрокомп'ютері Raspberry Pi 4 Model B. Основною ідеєю є надання можливості власнику автомобіля, який перебуває на дорозі в будь-якому куточку світу, швидко діагностувати несправності та консультуватися з

автомайстром. Це дозволить не лише швидко ідентифікувати проблему, але й визначити оптимальний спосіб її вирішення, чи то ремонт на місці, чи евакуація до найближчого сервісного центру.

Після розробки та впровадження цієї системи можна буде розглянути можливість створення спеціальної програми для мобільних бригад, які займатимуться допомогою у ремонті або евакуації автомобілів. Також система може бути адаптована для автоматичного виклику екстреної допомоги у випадку аварії, включаючи точне місцезнаходження автомобіля та інформацію про його власника.

Цей проєкт відкриває шлях до значного підвищення безпеки дорожнього руху та зручності водіїв завдяки сучасним технологіям та інтеграції мобільних та діагностичних інструментів.

РОЗДІЛ 2

ВИБІР ТА НАЛАШТУВАННЯ ТЕХНІЧНИХ ЗАСОБІВ

2.1 Проектування моделі на основі обраних компонентів

Проектування імітаційної моделі включає в себе декілька ключових етапів: моделювання, збірка фізичної моделі та налаштування необхідного програмного забезпечення.

Перший етап процесу проектування полягає у створенні детальної схеми підключень для пристрою. Це включає в себе визначення усіх необхідних компонентів, їх розміщення та взаємозв'язки між ними. Схема підключень служить як основа для подальшої збірки фізичної моделі та інтеграції програмного забезпечення, забезпечуючи чітке візуальне представлення всієї системи.

Завдяки детальному плануванню на початковому етапі, процес збірки та програмування стає більш зрозумілим і менш схильним до помилок, забезпечуючи ефективну та точну роботу імітаційної моделі (рис. 2.1).

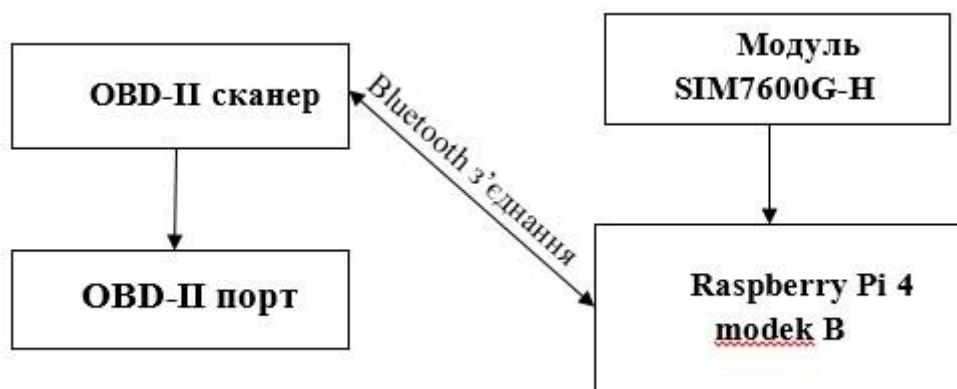


Рисунок 2.1 – Схема роботи моделі

На цьому етапі проектування, потрібно забезпечити правильне під'єднання сканера OBD-II до діагностичного роз'єму OBD-II, який зазвичай знаходиться під рульовим колесом автомобіля. Слід уважно виконати наступні кроки:

– Локалізація роз'єму OBD-II, розташованя у легкодоступному місці під рульовим колесом, недалеко від педалей.

– Перевірка сумісності сканера, сканер OBD-II повинен бути сумісний із автомобілем. Деякі сканери можуть потребувати специфічного програмного забезпечення або додаткових налаштувань для коректної роботи з певними марками або моделями автомобілів.

– Підключення сканера, під'єднати сканер OBD-II до роз'єму. Зазвичай це робиться простим вставлянням штекера сканера в роз'єм. Забезпечення надійного з'єднання, щоб уникнути поганого контакту або випадкового відключення під час діагностики.

– Запустити сканер і переконавшись, що йому вдається зчитати дані з автомобіля. Перевірити наявність помилок у підключенні, якщо сканер не може знайти або ідентифікувати автомобіль.

Ці кроки допоможуть забезпечити, що сканер OBD-II правильно підключений (рис. 2.2) і готовий до використання для діагностики автомобіля, що є критично важливим для подальшого успішного проектування та реалізації імітаційної моделі.



Рисунок 2.2 – Підключення OBD-II

Для безпечного та ефективного підключення OBD-II сканера до автомобіля важливо дотримуватися наступних інструкцій:

1. Підключення сканера:

– Перед підключенням сканера переконайтеся, що двигун автомобіля вимкнено. Це допоможе уникнути можливих пошкоджень електроніки як сканера, так і автомобіля.

– Підключіть сканер до роз'єму OBD-II, який зазвичай знаходиться під рульовим колесом. Увімкніть запалювання автомобіля, щоб забезпечити живлення сканера.

2. Налаштування Bluetooth:

– Після вмикання запалювання сканер автоматично активується і готовий до спарювання через Bluetooth. Необхідно переконаватися, що пристрій (смартфон, планшет або мікрокомп'ютер) увімкнено і готовий до пошуку Bluetooth пристроїв.

– Якщо на сканері встановлено PIN-код для безпеки, потрібно ввести його під час процесу спарювання.

3. Підключення до програми:

– Якщо спарювання завершено, необхідно встановити і запустити на пристрої спеціальний додаток, сумісний з вашим OBD-II сканером. Цей додаток використовується для відображення діагностичної інформації та управління даними.

4. Підготовка Raspberry Pi:

– Raspberry Pi 4 Model B не має вбудованої постійної пам'яті для зберігання операційної системи, тому для його запуску необхідна мікро SD-карта. Рекомендується використовувати карту ємністю щонайменше 4 Гб, але для кращої продуктивності в цій роботі використовується 32-гігабайтна SD-карта (рис. 2.3).

– Вставити SD-карту з встановленою операційною системою в слот на Raspberry Pi і підключити необхідні периферійні пристрої для початку роботи.

Завдяки цим крокам ви зможете використовувати Raspberry Pi 4 для прийому та обробки даних з автомобіля через OBD-II, забезпечуючи ефективну діагностику та вирішення проблем на місці.



Рисунок 2.3 – SD-карта та адаптер [1]

Для ефективної роботи на мікрокомп'ютері Raspberry Pi 4 з процесором ARM Cortex-A72, вам потрібно вибрати операційну систему, яка оптимізована під цю архітектуру і не є надто великою за розміром, щоб зменшити навантаження на процесор та забезпечити стабільну роботу.

Raspberry Pi OS (раніше відома як Raspbian) – це офіційна операційна система для Raspberry Pi, базується на Debian. Вона легка і спеціально розроблена для Raspberry Pi, що робить її ідеальним варіантом для проектів, які не вимагають великої потужності. Raspberry Pi OS доступна у декількох версіях, включно з «Lite» версією, яка є меншою за розміром та не має графічного інтерфейсу користувача. Початкове вікно програми Raspberry Pi imager зображено на рис. 2.4.



Рисунок 2.4 – Початкове вікно програми Raspberry Pi imager

У цій програмі перша кнопка дозволяє вибрати з переліку операційних систем (рис. 2.5), які рекомендовані офіційним виробником для використання на їхніх комп'ютерах.

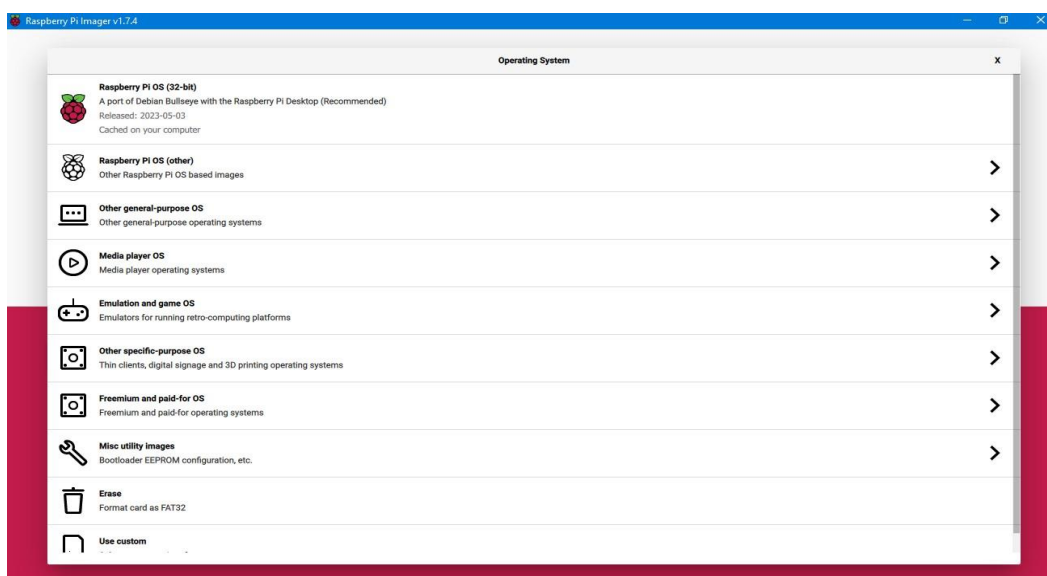


Рисунок 2.5 – Вибір операційної системи

На вибір користувачу пропонуються наступні операційні системи:

- Raspbian OS (32-bit).
- Raspbian OS – різні варіанти ОС: зміна розрядності (32-bit, 64-bit), наявність робочого столу та встановлені утилітами та програмами.

- Інші ОС (варіанти дистрибутивів Linux).
- Медіа операційні системи.
- Операційна система для ігрових приставок.
- Операційна система для серверів.
- Операційна система для взаємодії з іншими пристроями.
- Форматування SD-карти.
- Завантаження власної операційної системи.

У цьому проекті використано Raspbian OS, яка є першою в списку, має графічний інтерфейс і забезпечує стабільну роботу, задовольняючи основні потреби користувача.

Далі в процесі йде вибір місця для запису образу системи (рис. 2.6).

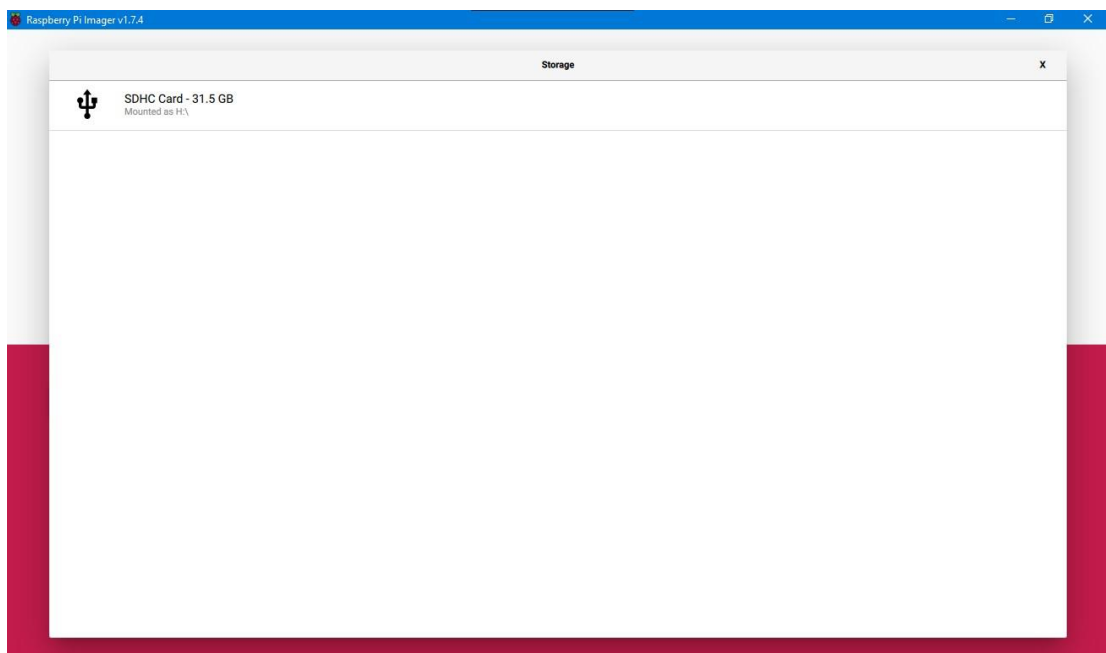


Рисунок 2.6 – Вибір місця запису образу операційної системи

Вибір місця для запису образу здійснюється після активації другої кнопки в головному меню програми. Одразу після цього відкривається доступ до функції запису, яка активується за допомогою третьої кнопки. При натисканні цієї клавіші з'являється вікно з підтвердженням (рис. 2.7), де користувачеві пропонується перевірити вибране місце для запису та

підтвердити можливість форматування цієї пам'яті для відповідності необхідним параметрам.

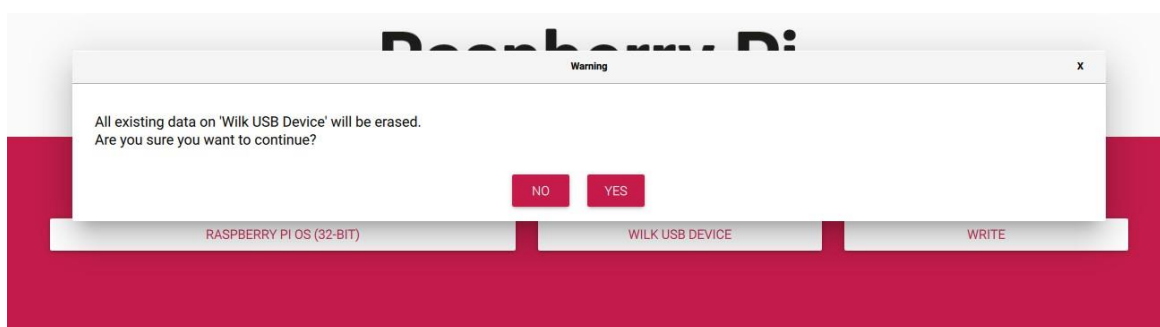


Рисунок 2.7 – Вікно підтвердження дій користувача

Якщо всі параметри перевірені потрібно натиснути кнопку «Так» та дочекайтеся завершення процесу запису образу на SD-карту (рис. 2.8).

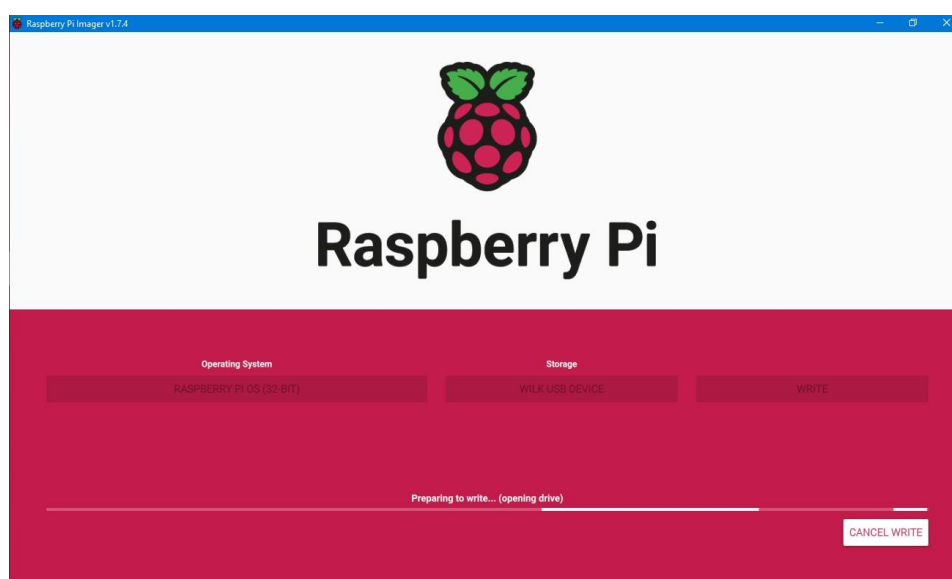


Рисунок 2.8 – Завантаження та запис операційної системи

Тепер вставте SD-карту з записаним образом операційної системи в відповідний порт на мікрокомп'ютері Raspberry Pi (рис. 2.9). Для відображення зображення підключіть HDMI кабель між мікрокомп'ютером та монітором або телевизором. Забезпечте електроживлення, під'єднавши USB type-C кабель до

джерела живлення. Після цього Raspberry Pi готовий до запуску та використання.

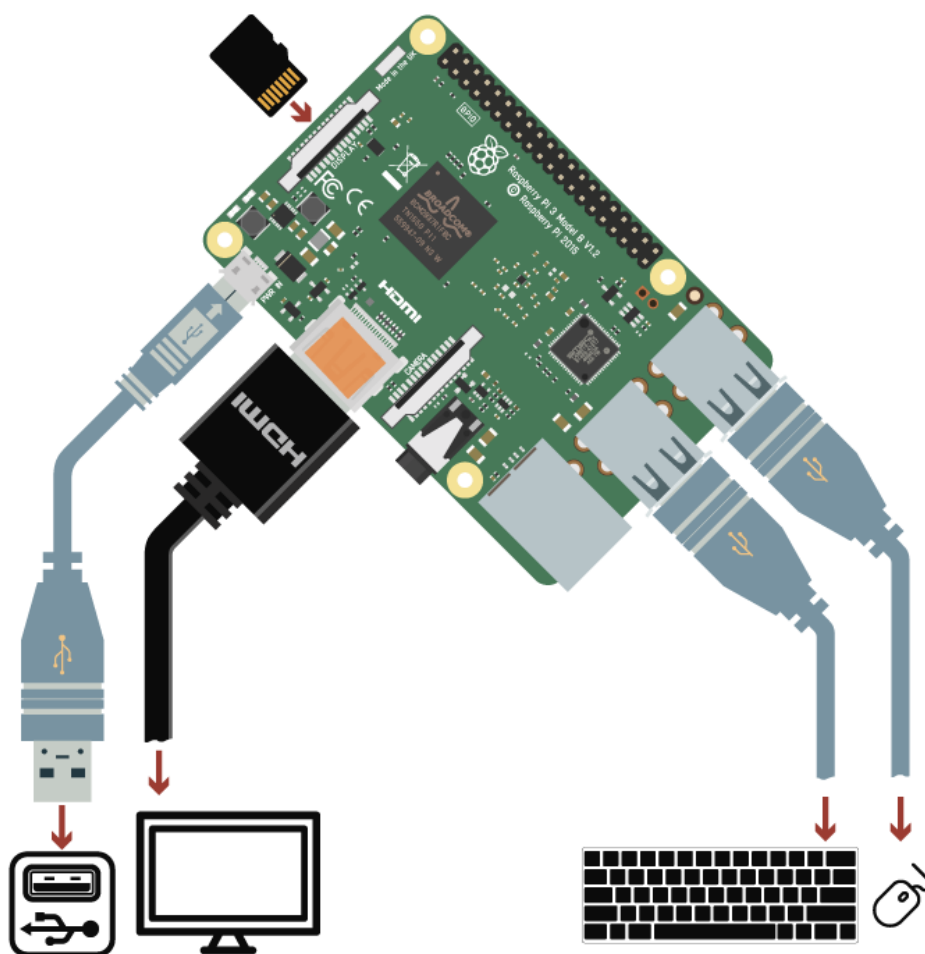


Рисунок 2.9 – Підключення до мікрокомп'ютера [8]

2.2 Встановлення програмного забезпечення

Перше вмикання мікрокомп'ютера (рис. 2.10) Raspberry Pi зазвичай триває довше, оскільки відбувається автоматичне розгортання образу операційної системи з SD-карти. Коли пристрій вперше підключається до джерела живлення, система ініціює процес розпакування та встановлення необхідних файлів. Після завершення цього процесу на екрані з'являється вікно привітання, що вказує на те, що Raspberry Pi готовий до подальшої настройки та використання.



Рисунок 2.10 – Перше вікно запуску

Після початкового привітання система відображає сповіщення про дії, які вона виконує. Однією з ключових операцій є зміна розміру файлової системи на SD-карті. Цей процес адаптує файлову систему до повного доступного простору на карті, оптимізуючи використання пам'яті (рис. 2.11).

Тривалість цієї операції залежить від обсягу сховища на SD-карті та швидкості, яку SD-карта може підтримувати. Високошвидкісні карти дозволяють швидше завершити цей процес, забезпечуючи ефективніше використання ресурсів Raspberry Pi.

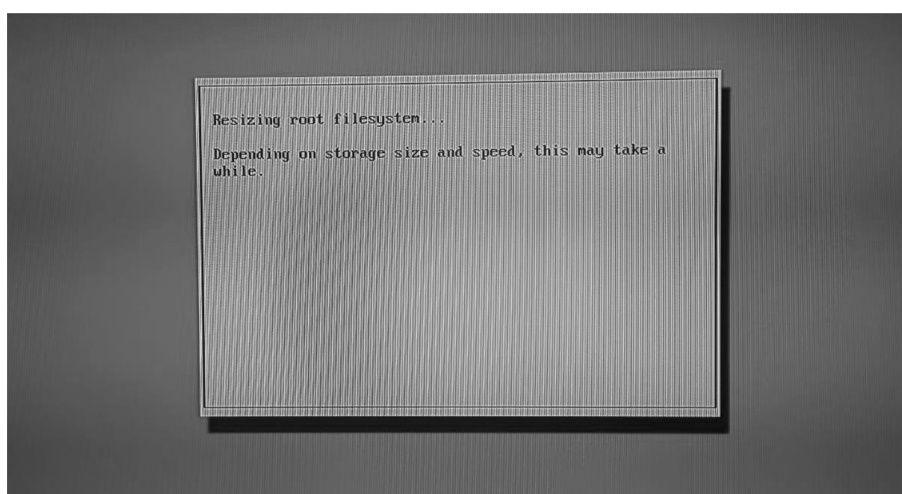


Рисунок 2.11 – Розподіл пам'яті

Оскільки обрана операційна система базується на ядрі Linux, основною кореневою директорією є символ «/». Від цієї кореневої директорії відгалужуються основні піддиректорії, такі як «Home», який є домашнім каталогом для користувачів, і «etc», де зберігаються конфігураційні та тимчасові файли системи.

Наступний важливий крок у налаштуванні системи – створення SSH ключів (рис. 2.12). SSH, або Secure Shell, це мережевий протокол, який забезпечує захищене підключення та передачу даних між двома пристроями через незахищені мережі. Протокол використовує криптографічне шифрування для автентифікації користувачів та забезпечення конфіденційності та цілісності даних під час передачі. Автентифікація через SSH відбувається за допомогою пар ключів: приватного та відкритого. Відкритий ключ розміщується на сервері або в пристрої, до якого необхідно отримати доступ, а приватний ключ залишається у володінні користувача, який ініціює з'єднання.

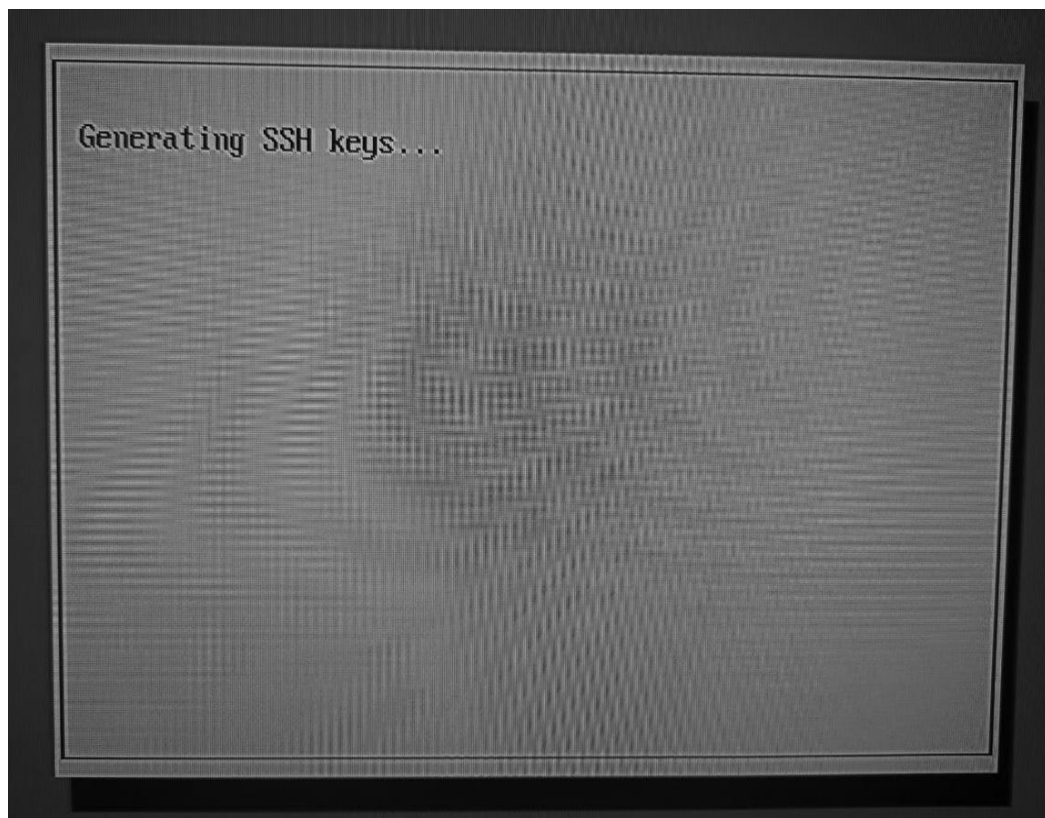


Рисунок 2.12 – Створення SSH ключів

Після створення ключів шифрування наступним кроком є налаштування PARTUUID (Partition UUID), який виступає як унікальний ідентифікатор розділу на диску в системах, де застосовується формат таблиці розділів GUID (GPT). PARTUUID допомагає в ідентифікації окремих розділів на диску, що значно спрощує процес визначення та доступу до цих розділів системою.

Кожен розділ у таблиці GPT отримує свій унікальний PARTUUID (рис. 2.13), який представляє собою 128-бітне число. Цей глобально унікальний ідентифікатор призначається автоматично під час формування таблиці GPT і зберігається в заголовку GPT. Завдяки цьому, операційна система або інші системні утиліти можуть використовувати PARTUUID для точного посилання на певний розділ, уникаючи потреби у використанні традиційних ідентифікаторів пристроїв. Це особливо корисно в багатодискових середовищах, де ідентифікація розділів за іменем пристрою може бути ускладнена.

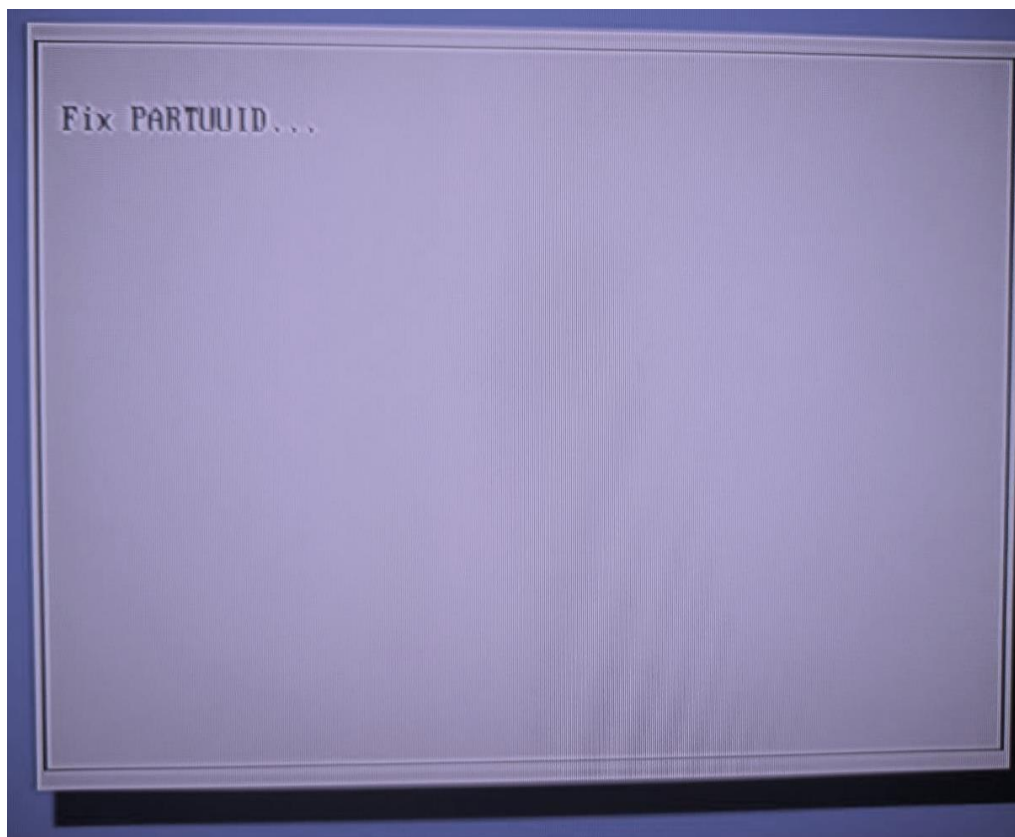


Рисунок 2.13 – Налаштування PARTUUID

Після завершення налаштування ключових параметрів системи необхідно провести перезапуск (рис. 2.14), щоб зміни вступили в силу та встановлені файли були правильно інтегровані в систему. Це забезпечить, що всі нові налаштування та конфігурації будуть активовані та система буде працювати з оновленим функціоналом.

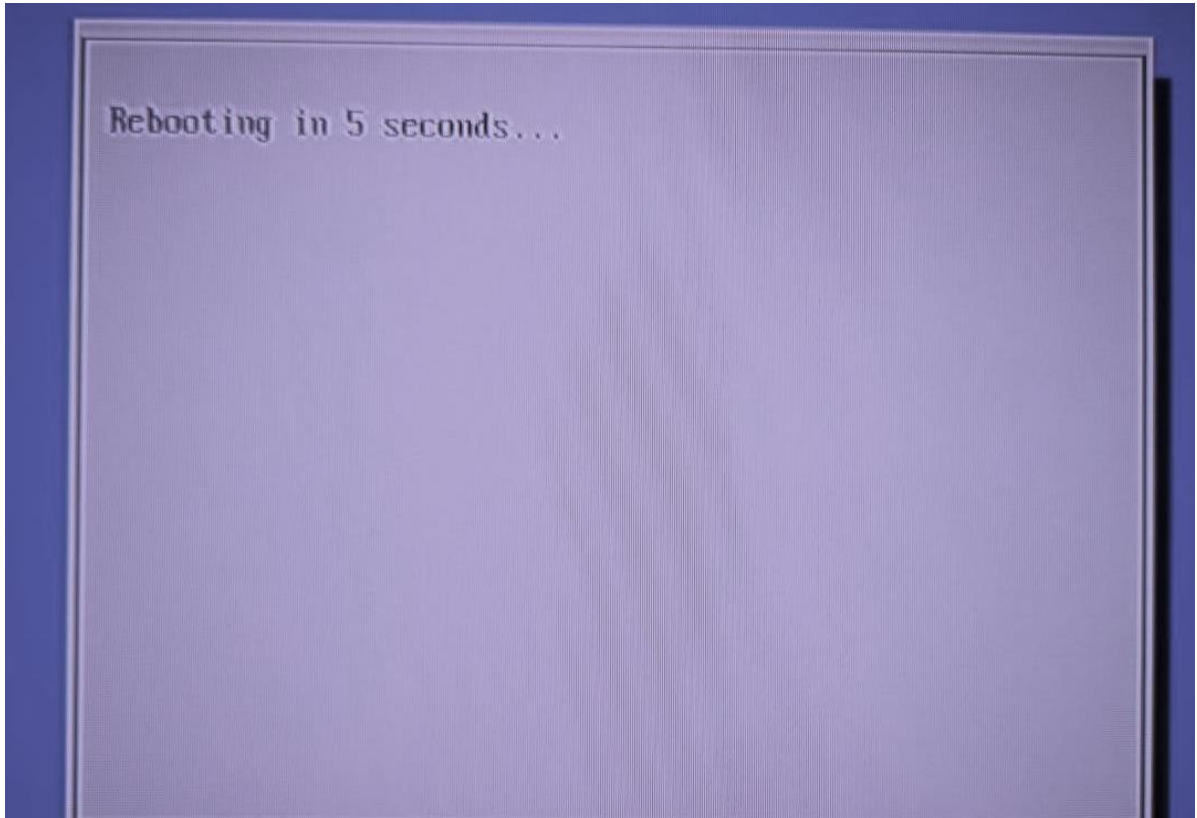


Рисунок 2.14 – Перезавантаження системи

Після перезапуску системи може знадобитися деякий час для завантаження всіх компонентів (рис. 2.15). Після завершення цього процесу на екрані з'явиться вікно привітання, яке також пропонує пройти через додаткові налаштування. Це може включати конфігурацію мови, часового поясу, Wi-Fi, а також інші користувацькі налаштування, які допоможуть оптимізувати роботу системи відповідно до ваших потреб.

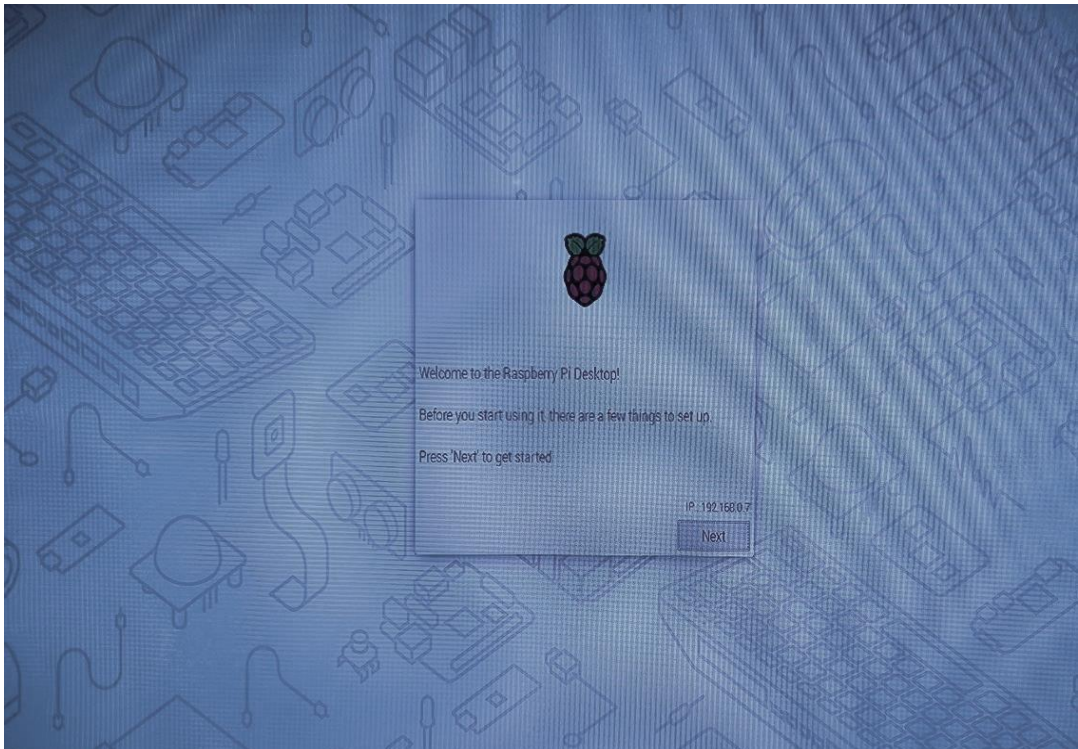


Рисунок 2.15 – Початок роботи

Після натискання на кнопку продовження з'явиться вікно, де можна вибрати регіон, розкладку клавіатури, часовий пояс та мову для вашої операційної системи (рис. 2.16). Ці налаштування дозволяють адаптувати систему до місцевих умов користування, забезпечуючи зручність і точність у відображенні часу, тексту та інтерфейсу.



Рисунок 2.16 – Вибір регіону і мови

Перевірка вибраних налаштувань регіону, розкладки клавіатури, часового поясу та мови зазвичай займає кілька хвилин. Після цього, система пропонує створити обліковий запис головного користувача. Для цього необхідно ввести логін та пароль, а також здійснити їх перевірку для підтвердження. Хоча можна використовувати прості значення для логіна та пароля, рекомендується вибрати більш складні комбінації для забезпечення безпеки системи.

Також система може запропонувати опцію автоматичного входу без запиту логіна і пароля при наступних запусках системи (рис. 2.17). Це може бути зручно, але важливо зважати на можливі ризики для безпеки, особливо якщо пристрій буде використовуватися в місцях з публічним доступом або де є інші користувачі.



Рисунок 2.17 – Створення користувача

Після створення облікового запису користувача, система запрошує налаштувати параметри дисплея, щоб забезпечити стабільне і чітке відображення зображення на моніторі. Цей процес включає автоматичний

вибір оптимального розширення екрану відповідно до можливостей підключеного монітора і зазвичай займає лише декілька секунд.

Далі система пропонує налаштувати підключення до мережі (рис. 2.18).

Це можна зробити двома основними способами:

– Провідне підключення: підключення до мережі за допомогою Ethernet кабелю, що забезпечує стабільне і швидке з'єднання. Цей метод є надійним варіантом для стаціонарних установок, де доступ до мережі не змінюється часто.

– Бездротове підключення: налаштування Wi-Fi з'єднання, що дозволяє підключатися до мережі без використання фізичних кабелів. Цей метод є ідеальним для мобільних установок або коли прямий доступ до мережевого кабелю неможливий.

Кожен з цих методів має свої переваги і може бути обраний в залежності від конкретних умов та потреб користувача. Вибір способу підключення до мережі є ключовим кроком для забезпечення доступу до мережевих ресурсів та Інтернету.

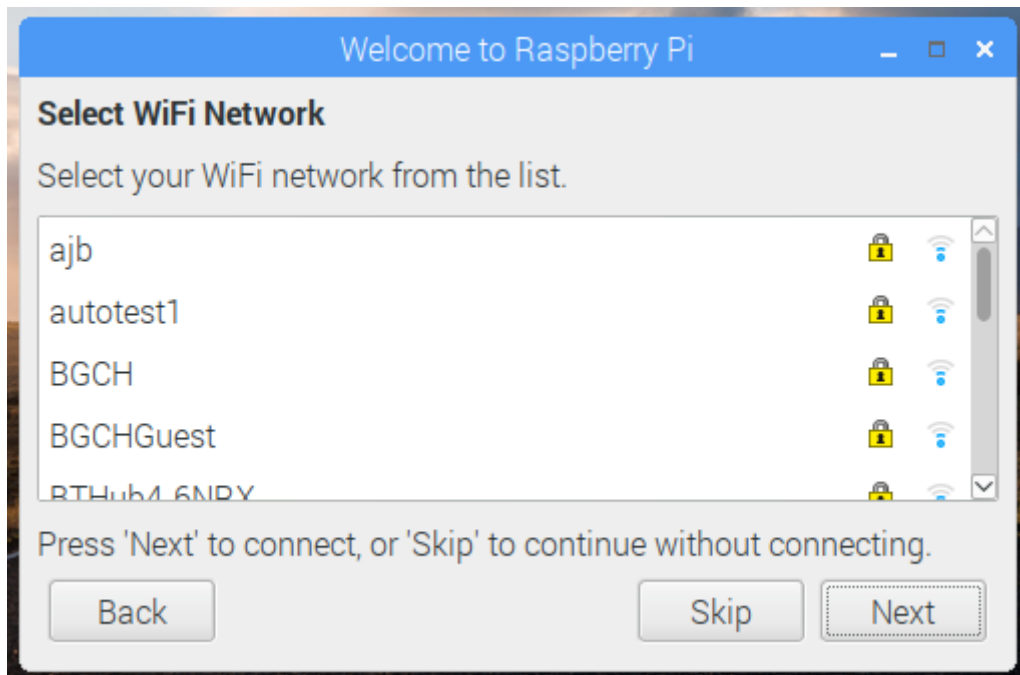


Рисунок 2.18 – Налаштування мережевого підключення

Після вибору методу підключення до мережі, наступний крок полягає у стандартному підключенні до обраної мережі, що включає введення пароля для доступу до Wi-Fi, якщо це необхідно. З успішним підключенням до інтернету, система надає можливість оновити програмне забезпечення.

Оновлення програмного забезпечення є критично важливим кроком (рис. 2.19), оскільки воно включає установку найновіших патчів безпеки, оновлення системних бібліотек, поліпшення стабільності та виправлення помилок, які могли бути присутніми в базовій версії ОС, що була встановлена на SD-карті. Це також може включати оновлення драйверів для кращої підтримки апаратного забезпечення та нових пристроїв, з якими може працювати Raspberry Pi.

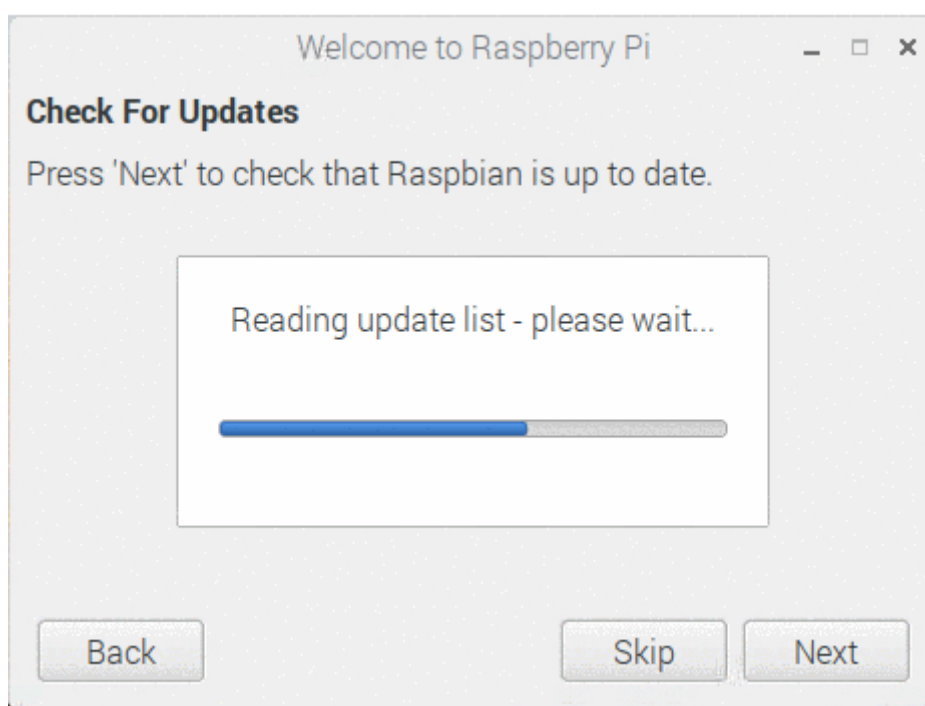


Рисунок 2.19 – Вікно оновлення операційної системи

Цей етап налаштування також надає можливість пропустити встановлення оновлень, що може бути корисним, якщо користувач обмежений у часі або має недостатньо стабільне інтернет-з'єднання. Проте, якщо оновлення були пропущені на цьому етапі, система нагадає про необхідність

їх встановлення при наступному перезапуску мікрокомп'ютера. Це забезпечує, що користувач не упустить важливі оновлення безпеки та функціональності.

Після завершення усіх налаштувань та оновлень користувачу відкривається головний екран операційної системи (рис. 2.20). На цьому етапі, користувач має доступ до основного робочого середовища, де можна запускати додатки, налаштовувати системні параметри або продовжувати налаштування додаткових компонентів і служб за потребою. Цей екран є відправною точкою для всіх подальших дій із використанням Raspberry Pi, від освітніх проектів до серйозних розробок на базі цього мікрокомп'ютера.

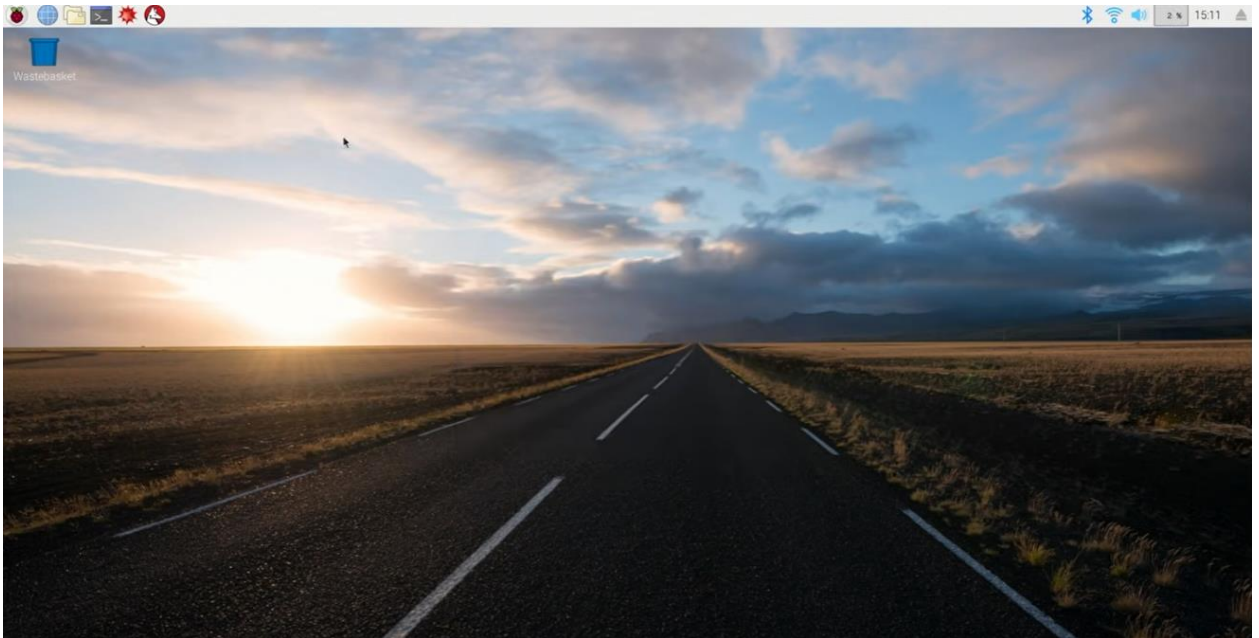


Рисунок 2.20 – Робочий стіл Raspbian

На цьому етапі робочий стіл Raspberry Pi налаштований, але система ще не може взаємодіяти з модулем SIM7600G-N 4G NAT, оскільки відсутні необхідні драйвери і налаштування послідовного порту. Щоб встановити необхідне програмне забезпечення та активувати COM порт, потрібно виконати наступні кроки:

- Натиснути на чорну кнопку в лівому верхньому куті екрану, щоб відкрити термінальне вікно.

– В термінальному вікні ввести команду ``sudo raspi-config`` та натиснути Enter.

– Можна використовувати стрілки на клавіатурі для навігації по меню. Перейти до розділу «Interface Options» (Параметри інтерфейсу) і натиснути Enter.

– Вибрати опцію «Serial Port» (Послідовний порт) і натиснути Enter.

– На запитання «Would you like the login shell to be accessible over serial?» (Чи бажаєте ви, щоб оболонка входу була доступна через послідовний порт?) вибрати «No» (Ні) і натиснути Enter.

– На наступне запитання «Would you like the serial hardware to be enabled?» (Бажаєте, щоб апаратне забезпечення послідовного порту було ввімкнено?) вибрати «Yes» (Так) і натиснути Enter.

– Буде показано підтвердження, що послідовна оболонка вимкнена, а послідовний інтерфейс увімкнений. Вибрати «Ok» (Гаразд) і натиснути Enter для завершення налаштувань.

– Вийти з меню ``raspi-config`` і перезапустити Raspberry Pi для застосування змін.

Після виконання цих кроків, Raspberry Pi зможе взаємодіяти з модулем SIM7600G-H 4G NAT, і ви зможете використовувати його для мережевих з'єднань і передачі даних.

Коли система запропонує перезавантаження після виконаних налаштувань, обов'язково виконати це, щоб застосувати всі зміни. Далі для роботи з послідовним портом на Raspberry Pi можна встановити програму Minicom, яка є простим текстовим комунікатором для послідовних з'єднань.

Кроки для встановлення Minicom:

– Відкрити вікно терміналу, скориставшись чорною кнопкою у верхньому лівому куті екрана.

– У відкритому термінальному вікні ввести команду:

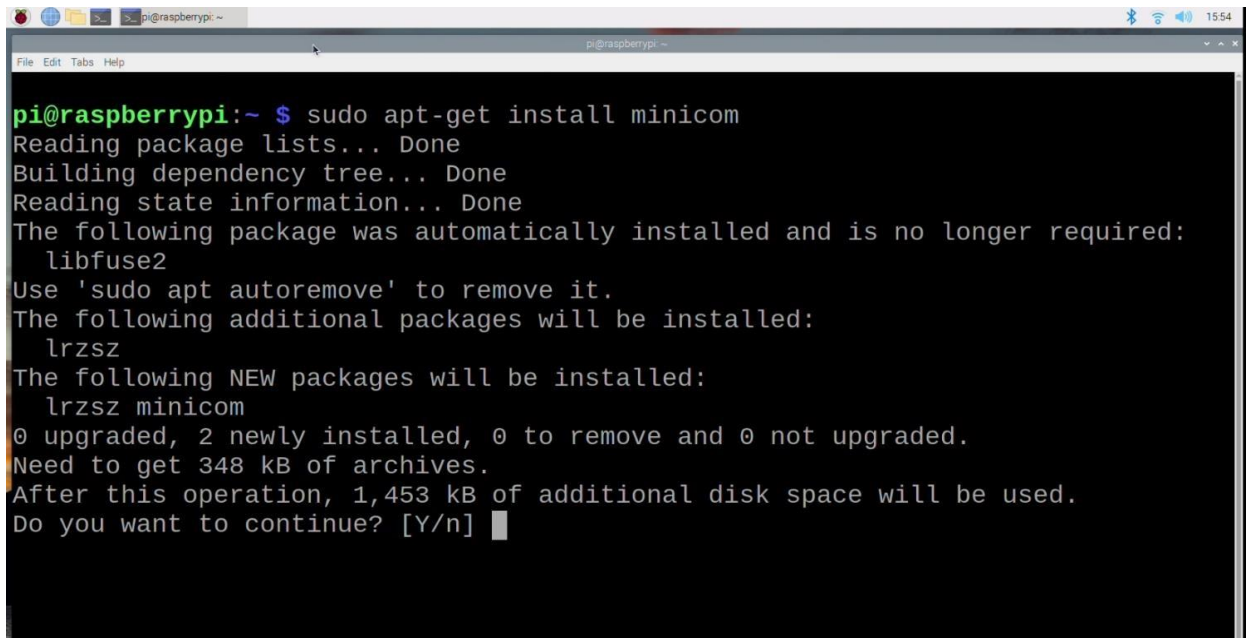
...

```
sudo apt-get install minicom
```

...

- Натиснути Enter для виконання команди.
- Якщо з'явиться запитання «Ви хочете продовжити? (y/n)», натиснути `Y` і потім клавішу Enter, щоб підтвердити і продовжити процес встановлення.

Ця процедура встановить Minicom (рис. 2.21), що дозволить здійснювати послідовні з'єднання через термінал, що є корисним для налагодження з'єднань із зовнішніми модулями, як SIM7600G-N 4G HAT. Завдяки цьому інструменту можна відправляти команди і отримувати відповіді від обладнання, підключеного через послідовний порт.



```

pi@raspberrypi:~ $ sudo apt-get install minicom
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  lrzsz
The following NEW packages will be installed:
  lrzsz minicom
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 348 kB of archives.
After this operation, 1,453 kB of additional disk space will be used.
Do you want to continue? [Y/n] █

```

Рисунок 2.21 – Командна стрічка виконання `sudo apt-get install minicom`

Для завантаження демо-кодів та драйверів, які необхідні для роботи з Waveshare 4G HAT, потрібно виконати наступні кроки:

- Скористатись чорною кнопкою у верхньому лівому куті екрана, щоб відкрити нове вікно терміналу.
- Ввести наступну команду в термінальне вікно для завантаження зір-файлу з прикладами кодів і драйверами (рис. 2.22):

```

...

wget https://www.waveshare.com/w/upload/2/29/SIM7600X-4G-HAT-
Demo.7z

```

– Натиснути Enter для виконання команди.

Ця команда ініціює завантаження архіву з офіційного сайту Waveshare, який містить усе необхідне для розробки та тестування програмного забезпечення, що взаємодіє з Waveshare 4G HAT. Завантажений файл міститиме приклади коду, які можна використовувати як базу для ваших проєктів, а також драйвери, необхідні для коректної роботи обладнання.

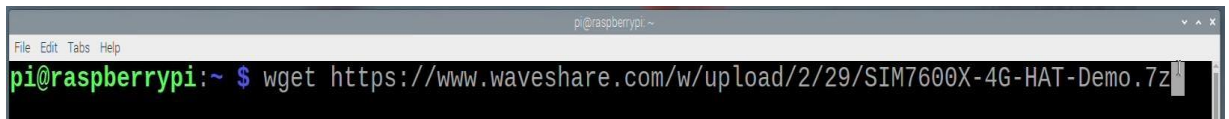


Рисунок 2.22 – Команда завантаження драйверів

Після завершення завантаження, необхідно виконати наступні кроки в тому ж термінальному вікні для розпакування та налаштування необхідних компонентів для роботи з Waveshare 4G HAT:

– Встановити програмне забезпечення 7-Zip, яке дозволить розпакувати завантажений файл. Ввести та виконати наступну команду:

```

...

sudo apt-get install p7zip-full
...

```

– Натисніть `Enter` для запуску процесу встановлення.

– Після встановлення 7-Zip розпакувати завантажений архів за допомогою наступної команди:

```

...

7z x SIM7600X-4G-HAT-Demo.7z -r -o/home/pi
...

```

– Це розпакує файли у директорію `/home/pi`.

– Надайти відповідні права для директорії з розпакованими файлами, щоб уникнути проблем з доступом:

...

```
sudo chmod 777 -R /home/pi/SIM7600X-4G-HAT-Demo
```

...

– Відкрити файл `/etc/rc.local` для редагування:

...

```
sudo nano /etc/rc.local
```

...

– У цьому файлі необхідно додати команду, яка ініціалізує модуль SIM7600G-H 4G HAT при кожному запуску Raspberry Pi:

...

```
sh/home/pi/SIM7600X-4G-HAT-Demo/Raspberry/sim7600_4G_hat_init
```

...

– Вставити цю команду перед рядком `exit 0`.

– Зберегти зміни, натиснувши `Ctrl+X`, потім `Y` для підтвердження, і `Enter` для закриття редактора.

Після цих кроків драйвери та програми для Waveshare 4G HAT будуть автоматично запускатися при кожному завантаженні Raspberry Pi, забезпечуючи стабільне з'єднання і функціональність модуля.

Для завершення налаштування та ініціалізації драйвера Waveshare 4G HAT, слід виконати наступні кроки в терміналі Raspberry Pi:

– Після внесення необхідних змін в файл `/etc/rc.local`, зберегти їх, натиснувши `Ctrl+X`, потім `Y` для підтвердження збереження, і нарешті `Enter` для виходу з редактора nano.

– Відкрити нове вікно терміналу для продовження роботи з драйверами.

– Ввести наступну команду для переходу до директорії, де знаходиться драйвер `bcm2835`, який використовується для роботи з Waveshare 4G HAT:

...

```
cd /home/pi/SIM7600X-4G-HAT-Demo/Raspberry/c/bcm2835
```

```

– Натисніть `Enter`.

– Ввести наступний рядок команд, щоб надати виконавчі права файлу налаштування, виконати його, зібрати та встановити драйвер:

```

```
chmod +x configure && ./configure && sudo make && sudo make install
```

```

– Ці команди встановлять необхідні компоненти для повноцінної роботи модуля Waveshare 4G HAT на вашому Raspberry Pi.

– Після введення команди натиснути `Enter`, щоб розпочати процес компіляції та встановлення драйвера.

Ці кроки дозволять коректно ініціалізувати та налаштувати драйвер для модуля Waveshare 4G HAT (рис. 2.23), забезпечуючи його стабільну роботу з Raspberry Pi. Після завершення цих кроків ваша система буде готова до використання зовнішнього модуля зв'язку.

```

pi@raspberrypi:~/SIM7600X-4G-HAT-Demo/Raspberry/c/bcm2835 $ chmod +x configure && ./configure && sudo make && sudo make install
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
/home/pi/SIM7600X-4G-HAT-Demo/Raspberry/c/bcm2835/missing: unknown '--is-lightweight' option
Try '/home/pi/SIM7600X-4G-HAT-Demo/Raspberry/c/bcm2835/missing --help' for more information
configure: WARNING: 'missing' script is too old or missing
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for style of include used by make... GNU
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files...

```

Рисуюнок 2.23 – Процес виконання команди

Для забезпечення віддаленого доступу до Raspberry Pi та налаштування можливостей демонстрації даних з OBD-II, потрібно встановити та налаштувати програму AnyDesk, а також віртуальну оболонку для Android, так звану Anbox:

Встановлення AnyDesk

– Відкрити термінальне вікно і ввести наступні команди для оновлення вашої системи:

```
```bash
```

```
sudo apt update
sudo apt full-upgrade
...
```

– Це забезпечить, що система містить найновіші оновлення та патчі безпеки.

– Завантажити пакет AnyDesk, використовуючи wget:

```
```bash
wget https://download.anydesk.com/rpi/anydesk_6.1.1-1_armhf.deb
...`
```

– Встановити AnyDesk, використовуючи dpkg:

```
```bash
sudo dpkg -i anydesk_6.1.1-1_armhf.deb
...`
```

– Виконати автоматичне вирішення залежностей:

```
```bash
sudo apt-get install -f
...`
```

– Іконка AnyDesk має з'явитись у меню. Якщо цього не сталося, можливо потрібно буде перезапустити Raspberry Pi.

Встановлення Anbox для Android-емуляції

– Використати чорну кнопку у верхньому лівому куті для відкриття нового термінального вікна.

– Оновити систему, ввести:

```
```bash
sudo apt update
...`
```

– Для встановлення Anbox потрібно виконати додаткові кроки для встановлення необхідних залежностей і самої програми, але перш за все перевірити, чи є доступні пакети для вашої версії Raspberry Pi OS, оскільки Anbox може вимагати специфічних налаштувань для архітектури ARM.

Після встановлення всіх необхідних програм, потрібно переконайтися, що все налаштовано правильно, і віддалено підключитися через AnyDesk та використовувати Anbox для запуску Android-додатків, які можуть взаємодіяти з OBD-II пристроєм (рис. 2.24).

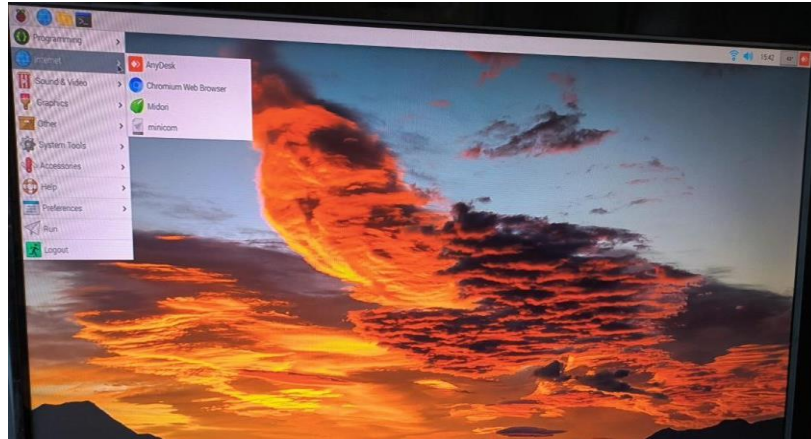


Рисунок 2.24 – Розміщення AnyDesk

Щоб встановити Snap та використовувати його для управління пакетами на Raspberry Pi, потрібно виконати наступні кроки:

Встановлення Snapd

– Якщо вікно ще не відкрите, скористатись чорною кнопкою у верхньому лівому куті екрана.

– Ввести команду:

```
```bash
sudo apt install snapd
```
```

– Натиснути `Enter` для виконання команди і дочекайтесь завершення інсталяції.

– Після завершення інсталяції, перезапустити Raspberry Pi для активації змін:

```
```bash
sudo reboot
```
```

Встановлення основного компонента Snap

– Після перезапуску, відкрити термінальне вікно знову і ввести команду:

```
```bash
sudo snap install core
```
```

– Ця команда оновить snap до останньої версії та встановить базові компоненти.

Встановлення Anbox через Snap

– Ввести наступну команду для встановлення Anbox з каналу beta:

```
```bash
sudo snap install anbox --beta
```
```

– Це може бути рекомендовано для тестування останніх функцій Anbox, версія beta може містити помилки або бути нестабільною.

– Якщо система запропонує іншу версію Anbox, яка краще підходить для вашої версії ОС, можна вибрати її для встановлення. Буде запропоновано підтвердити встановлення цієї версії.

Після завершення встановлення, Anbox повинен з'явитися у меню «пуск» під розділом «інше», звідки можна запускати його для використання Android-додатків на Raspberry Pi (рис. 2.25).

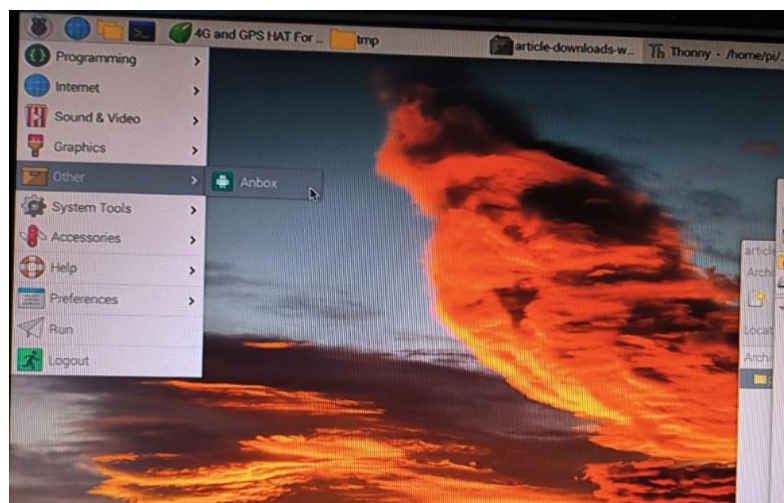


Рисунок 2.25 – Розміщення Anbox

РОЗДІЛ 3

ТЕСТУВАННЯ ТА ВІДЛАГОДЖЕННЯ РОЗРОБЛЕНОГО ПРИСТРОЮ

3.1 Тестування роботи модулів пристрою

Для проведення тестових випробувань створеного пристрою на Raspberry Pi з використанням модуля SIM7600G-H та перевірки віддаленого підключення.

Тестування модуля SIM7600G-H:

– Натиснути на значок малини в лівому верхньому куті для доступу до меню «пуск».

– Перейти до секції «Програмування» і вибрати інтегроване середовище розробки (IDE), наприклад Thonny Python IDE або інше, що вже встановлене у системі (рис. 3.1).

– У вибраному IDE створити новий скрипт для тестування базових функцій модуля SIM7600G-H, таких як встановлення з'єднання, відправка SMS, або підключення до інтернету.

– Наприклад, можна написати простий Python скрипт для відправлення SMS або здійснення HTTP запиту через 4G мережу.

Перевірка віддаленого підключення:

– Переконайтеся, що AnyDesk запущений на Raspberry Pi.

– Занотувати ID AnyDesk вашого пристрою.

– Віддалено підключитися до Raspberry Pi з іншого комп'ютера, де також встановлено AnyDesk.

– Після встановлення з'єднання через AnyDesk, виконати різні операції на Raspberry Pi віддалено, наприклад, перезапустити пристрій, запустити додатки, або редагувати файли.

Після виконання цих тестів, можна оцінити стабільність з'єднань та функціональність пристрою. Це дасть змогу виявити можливі проблеми або потреби у подальшій оптимізації. У разі виникнення проблем з модулем

SIM7600G-H або з віддаленим доступом, може знадобитися додаткова налаштування або консультації з технічною підтримкою.

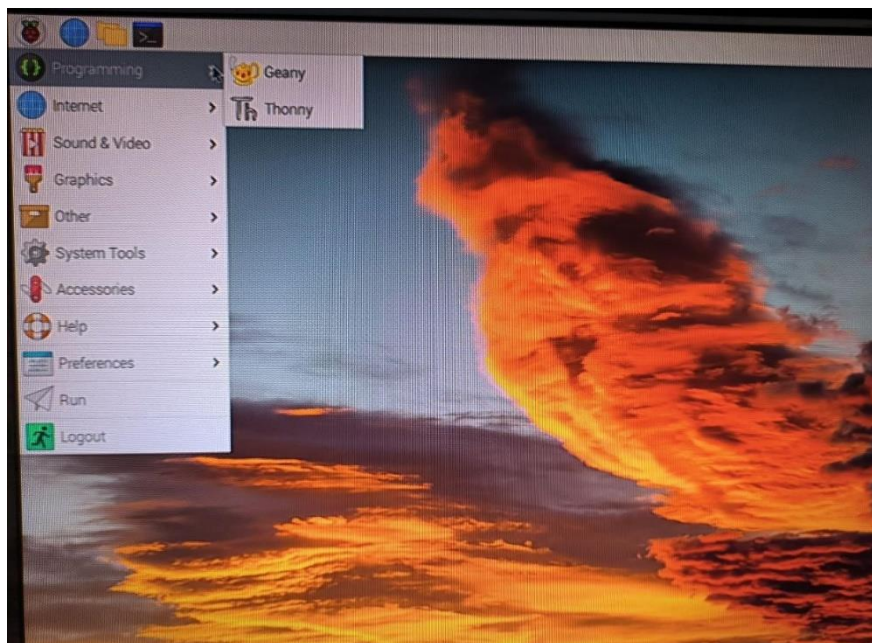


Рисунок 3.1 – Розташування Thonny Python IDE

Для випробування телефонного зв'язку через Raspberry Pi із використанням модуля SIM7600G-H, можна виконати наступні кроки для запуску сценарію, що здійснює телефонний дзвінок.

Відкриття сценарію PhoneCall.py:

- На робочому столі або у будь-якому вікні файлового менеджера Raspberry Pi, відкрити каталог, де зберігається скрипт.

- Шлях до файлу: ``/home/pi/SIM7600X-4G-NAT-Demo/Raspberry/Python/PhoneCall``.

- Клацнути правою кнопкою миші на файлі ``PhoneCall.py``.

- Вибрати опцію «Відкрити за допомогою».

- Вибрати «Thonny Python IDE» або іншу програму для програмування.

Перевірка та виконання скрипту:

- Перед виконанням скрипту, перевірити, що він правильно налаштований для використання з SIM7600G-H модулем. Зокрема, перевірити, чи правильно вказано номер телефону, на який буде здійснено виклик.

– Можливо, знадобиться внести зміни у параметри дзвінка або конфігурації залежно від оператора мобільного зв'язку.

– Після перевірки та збереження будь-яких змін, запусити скрипт, натиснувши кнопку «Run» у Thonny IDE або іншому середовищі розробки (рис. 3.2).

– Слідкуйте за виводом у консолі IDE, щоб побачити результати виконання скрипту і переконатися, що дзвінок було здійснено успішно.

Вирішення можливих проблем:

– Якщо виникають помилки під час виконання скрипту, перевірте логи або повідомлення в консолі для діагностики проблем. Можливо, потрібно буде внести додаткові налаштування у конфігурацію модуля або скрипту.

Цей тест дозволить перевірити базову функціональність вашого налаштування Raspberry Pi і модуля SIM7600G-H у реальних умовах використання. Це також допоможе виявити потенційні проблеми в налаштуванні, які можуть вплинути на подальшу роботу системи.

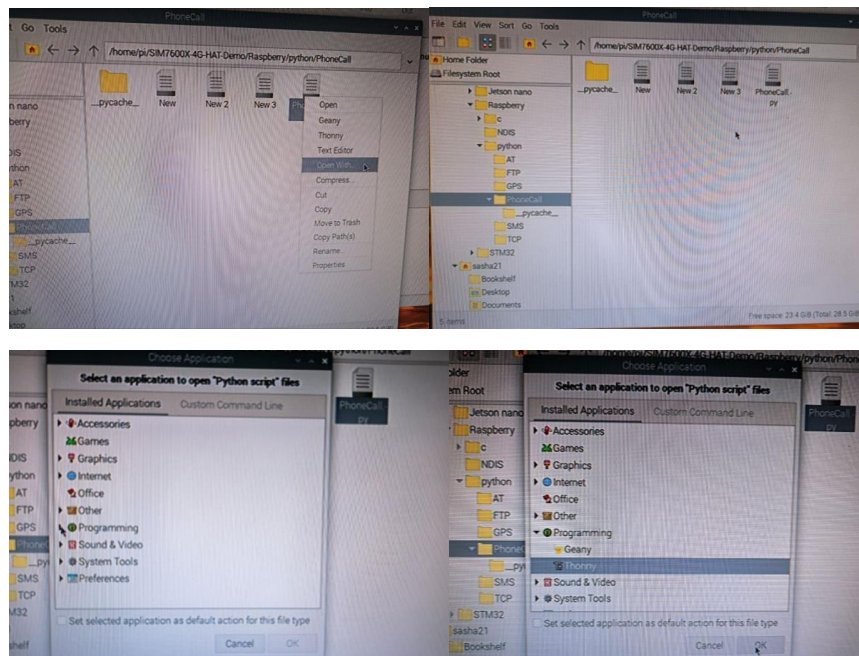


Рисунок 3.2 – Відкриття файлу через Thonny Python IDE

Перед тим як розпочати тестування телефонного зв'язку через Raspberry Pi з модулем SIM7600G-H, важливо переконатися, що скрипт налаштовано правильно. Ось як можна це зробити.

Налаштування скрипту PhoneCall.py:

- Відкрити файл `PhoneCall.py` у Thonny Python IDE або в іншому текстовому редакторі.

- Знайти рядок коду, де вказується номер телефону, на який буде здійснено виклик. Це може бути щось на кшталт `phone_number = «1234567890»`. Замінити `«1234567890»` на актуальний номер телефону, на який бажано зробити дзвінок.

- Переконатись, що використовується правильний порт для моделі Raspberry Pi. За замовчуванням для Raspberry Pi 3B, 3B+, і 4 Model B використовується `ttyS0`, тому рядок ініціалізації серійного з'єднання має виглядати так:

```
```python
ser = serial.Serial('/dev/ttyS0', 115200)
```
```

- Якщо ви використовуєте Raspberry Pi 2B або Zero, вам потрібно змінити цей рядок на:

```
```python
ser = serial.Serial('/dev/ttyAMA0', 115200)
```
```

- Це забезпечує коректне підключення до модуля через UART.

Збереження та запуск скрипту (рис. 3.3):

- Після внесення всіх необхідних змін, зберегти файл у IDE або текстовому редакторі.

- Запустити скрипт в IDE або з терміналу, щоб перевірити функціональність телефонного дзвінка.

Виконуючи ці кроки, можна забезпечити правильне налаштування і випробування системи для здійснення телефонних дзвінків через Raspberry Pi

і модуль SIM7600G-H. Це важливо для перевірки з'єднань та функціональності обладнання перед його використанням у реальних умовах.

```

1 #!/usr/bin/python
2
3 import RPi.GPIO as GPIO
4 import serial
5 import time
6
7 ser = serial.Serial('/dev/ttyS0', 115200)
8 ser.flushInput()
9
10 phone_number = '0954709049'
11 power_key = 6
12 rec_buff = ''
13
14 def send_at(command, back, timeout):
15     rec_buff = ''
16     ser.write((command + '\r\n').encode())
17     time.sleep(timeout)
18     if ser.inWaiting():
19         time.sleep(0.01)
20         rec_buff = ser.read(ser.inWaiting())
21     if back not in rec_buff.decode():
22         print(command + ' ERROR!')
23         print(command + ' back:\t' + rec_buff.decode())
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
```

– Якщо дзвінок успішно встановлено, підхопити його і переконайтеся, що зв'язок триває менше 20 секунд, якщо це вказано у сценарії.

Відлагодження:

– Якщо дзвінок не відбувся або виникли інші проблеми, переглянути логи в Thonny IDE або консолі для діагностики помилок. Перевірте налаштування UART, стан SIM-карти, і чи активна мобільна мережа для модуля.

– Якщо потрібно внести зміни у параметри дзвінка або обробку подій після дзвінка, потрібно відкрити сценарій і внести необхідні корективи. Зберегти зміни перед повторним запуском.

Ці кроки допоможуть переконатися, що ваша система з Raspberry Pi і модулем Waveshare 4G HAT належним чином налаштована та здатна здійснювати мобільні дзвінки, що є важливим підтвердженням функціональності вашої мережі та обладнання.

Для перевірки працездатності GSM антени через використання модуля SIM7600G-H на вашому Raspberry Pi, можна виконати тест з використанням GPS сценарію. Ось як можна це зробити.

Відкриття та запуск сценарію GPS.py:

- Відкрити файловий менеджер на Raspberry Pi.
- Перейти до каталогу `~/home/pi/SIM7600X-4G-HAT-Demo/Raspberry/Python/GPS``.
- Знайти файл ``GPS.py``.
- Натиснути правою кнопкою миші на файлі ``GPS.py``.
- Вибрати опцію «Відкрити за допомогою».
- Вибрати Thonny Python IDE або іншу програму для програмування.
- Переглянути код у файлі ``GPS.py`` перед запуском. Переконайтесь, що він налаштований для відстеження GPS даних, які модуль SIM7600G-H може отримувати. Код може виглядати як запит на отримання координат чи статусу GPS (рис. 3.4).

– Запустити скрипт, натиснувши на кнопку «Run» в Thonny IDE. Це ініціює процес отримання даних від GPS.

– Стежити за виводом в консолі IDE. Результатом мають бути GPS дані, такі як координати, швидкість, напрямок тощо.

Вирішення можливих проблем:

– Знаходження GPS сигналу може зайняти деякий час, особливо якщо модуль знаходиться всередині будівлі або в місці з поганим сигналом. При потребі перемістити пристрій на відкрите місце.

– Якщо GPS дані не отримуються або їх якість погана, переконайтесь, що GSM антена належним чином підключена та оптимально розташована.

– Якщо в консолі з'являються помилки, проаналізувати їх для вирішення проблем, які можуть включати налаштування антени, конфігурації модуля або проблеми з самим кодом.

Виконання цих кроків допоможе переконатися в тому, що GSM антена налаштована правильно і здатна приймати GPS сигнали для подальшої обробки. Таким чином, можна використовувати Raspberry Pi для навігації.

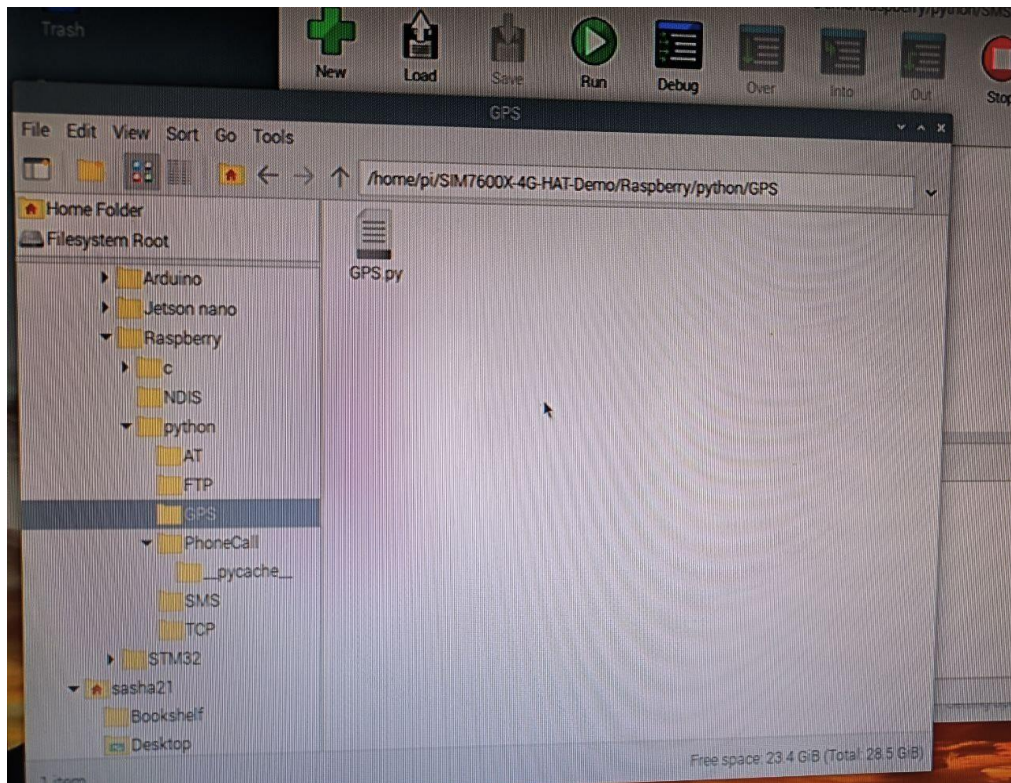


Рисунок 3.4 – Розміщення сценарію GPS

Завдяки цьому можна відстежувати географічне розташування пристрою в реальному часі. Ось кроки, які необхідно здійснити, та додаткові поради, які можуть бути корисними під час тестування.

Запуск та моніторинг GPS сценарію:

- Натиснути правою кнопкою миші на файлі ``GPS.py`` у каталозі ``/home/pi/SIM7600X-4G-HAT-Demo/Raspberry/Python/GPS``.

- Вибрати «Відкрити за допомогою» і оберати Thonny Python IDE або інше середовище розробки.

- Натиснути на зелений трикутник «Run» для запуску сценарію.

- Дати системі кілька хвилин, щоб сценарій зміг ініціалізувати GPS-модуль та почати отримувати дані.

- Простежити за виводом у Python Shell. Сценарій має регулярно виводити дані GPS, такі як широта, довгота, висота, швидкість та інші параметри.

- Якщо ваш Raspberry Pi рухатиметься, координати повинні змінюватися відповідно.

Перевірка і вирішення можливих проблем:

- Якщо дані не отримуються або оновлюються дуже рідко, переконайтеся, що GPS-антена має хороший прийом сигналу.

- Якщо у консолі з'являються помилки або попередження, вони можуть допомогти зрозуміти причину проблем із отриманням даних. Перевірити конфігурацію та код сценарію на предмет помилок у логіці або викликах функцій.

Завдяки цьому тестуванню можна не лише перевірити роботу GSM антени, а й переконатися в можливостях Raspberry Pi з модулем SIM7600G-H для отримання і обробки даних GPS в реальному часі. Це важливий крок у перевірці системи перед її застосуванням в практичних цілях (рис. 3.5).

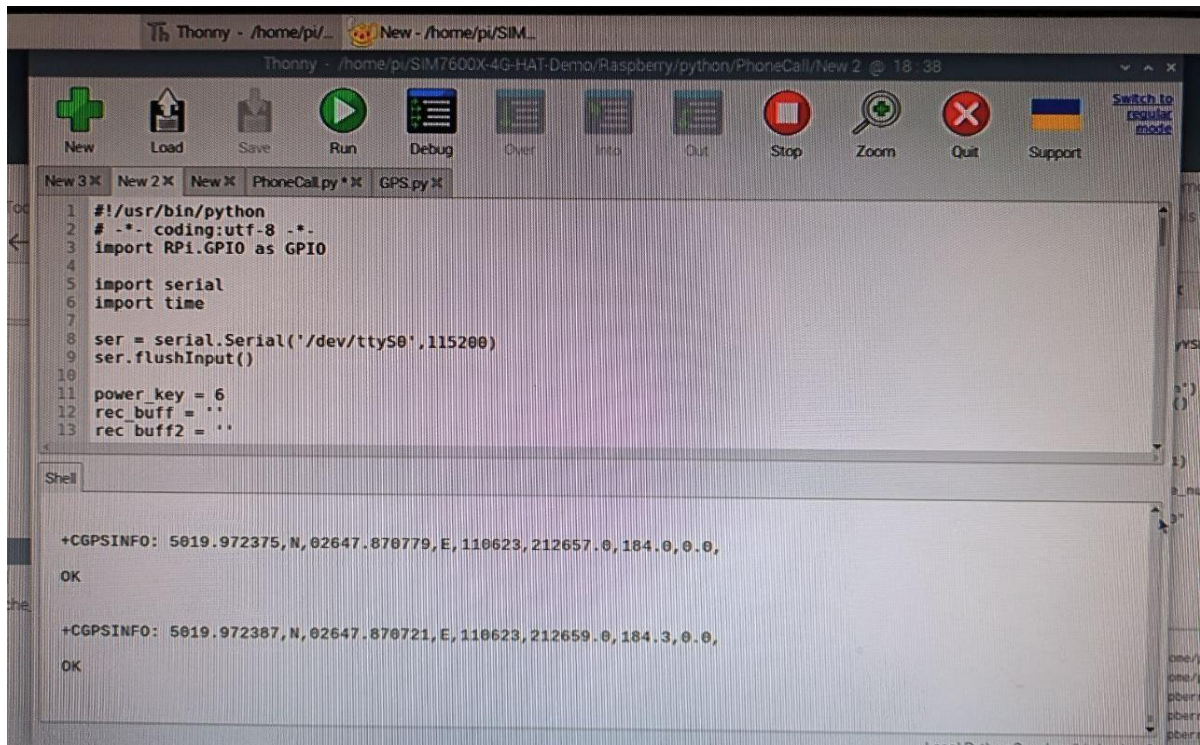


Рисунок 3.5 – Вікно Thonny Python IDE

Якщо виявляється, що координати, отримані через GPS-модуль Raspberry Pi, мають невелику помилку порівняно з даними Google Maps, це може бути пов'язано з кількома чинниками, що впливають на точність GPS (рис. 3.6).

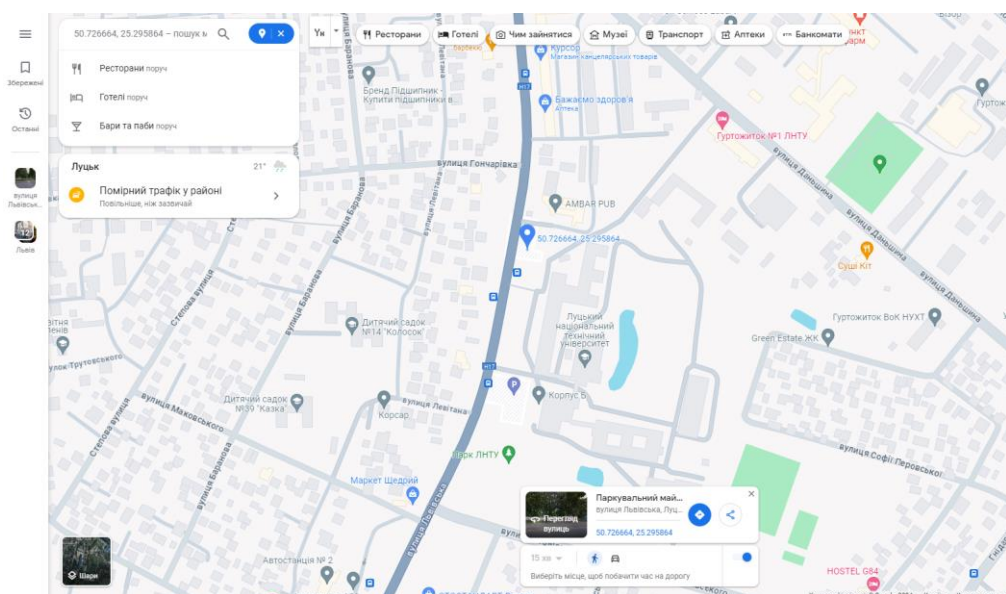


Рисунок 3.6 – Координати розташування RPI на Google Maps

Варто зазначити, що погодні умови можуть впливати на з'єднання з супутниками. Під час тестування у ясний день знадобилося близько 30 секунд для встановлення з'єднання, тоді як у дощовий день – близько 60 секунд, що свідчить про незначні затримки при встановленні супутникового зв'язку.

Теперішній Python сценарій за замовчуванням працює належно, проте він виводить дані розташування у форматі NMEA, який є стандартним для GPS пристроїв, але не зовсім зручним для сприйняття. Відтак, було розроблено інший сценарій, «GPS-Human-Readable.py», який виводить широту та довготу у форматі десяткових градусів. Ці дані можна безпосередньо вставити в пошук Google Maps для визначення точної локації пристрою.

Таким чином, одноплатний комп'ютер Raspberry Pi інтегровано з мобільною мережею та GPS, дозволяючи використовувати обидва ресурси одночасно. Було створено сценарій, що дозволяє не лише ідентифікувати GPS-локацію пристрою, але й відсилати SMS із зазначенням точного місцезнаходження. Повідомлення формулюється легко зрозумілим текстом.

Для цього відкриємо сценарій «FindGPSthenSMSFinal.py», який є поєднанням попередніх демонстраційних скриптів. Після налаштування номера телефону для надсилання SMS, запускаємо сценарій, що починає сеанс GPS і при успішному визначенні локації переходить до режиму відсилання SMS.

Це демонструє, як Raspberry Pi може взаємодіяти з телефоном, і відкриває можливості для двосторонньої взаємодії. Наприклад, використання GPIO пінів Raspberry Pi дозволяє контролювати зовнішні пристрої, як світлодіоди, які можуть активуватися від прийому SMS з певним ключовим словом. Сценарій «GPIO-LED-Control.py» може бути налаштований так, що він активується тільки від SMS від певного номера, що забезпечує додатковий рівень безпеки та персоналізації.

3.2 Алгоритм пошуку несправностей

Після завершення випробувань, можливе інтегрування всіх компонентів з автомобілем, починаючи з підключення OBD-II сканера. Далі з'єднуємо мікрокомп'ютер з джерелом живлення через адаптер у роз'ємі прикурювача. Коли всі компоненти готові, можна завести двигун і синхронізувати сканер з мікрокомп'ютером, що автоматично запустить програму для діагностики та сценарій SMS інформування автомеханіка.

Наступним кроком автомеханік зможе застосувати програму AnyDesk на своєму комп'ютері для з'єднання з одноплатним Raspberry Pi, який розміщений у автомобілі. Це забезпечить з'єднання необхідне для управління і перевірки стану авто.

Перше підключення має бути здійснене вручну, для цього потрібно активувати Anbox, віртуальну Android-платформу, і запустити додаток Car Scanner (рис. 3.7). Це дозволить отримувати детальну інформацію про параметри автомобіля без необхідності синхронізації з OBD-II сканером, доступ до базових функцій буде надано одразу після запуску програми.

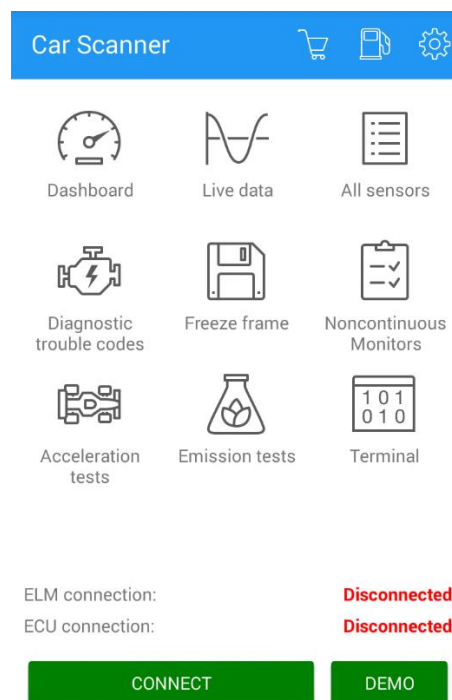


Рисунок 3.7 – Початкове вікно програми Car Scanner

Після натискання на зелену кнопку «Підключити», система автоматично почне пошук активованого сканера, який знаходиться поблизу (рис. 3.8).

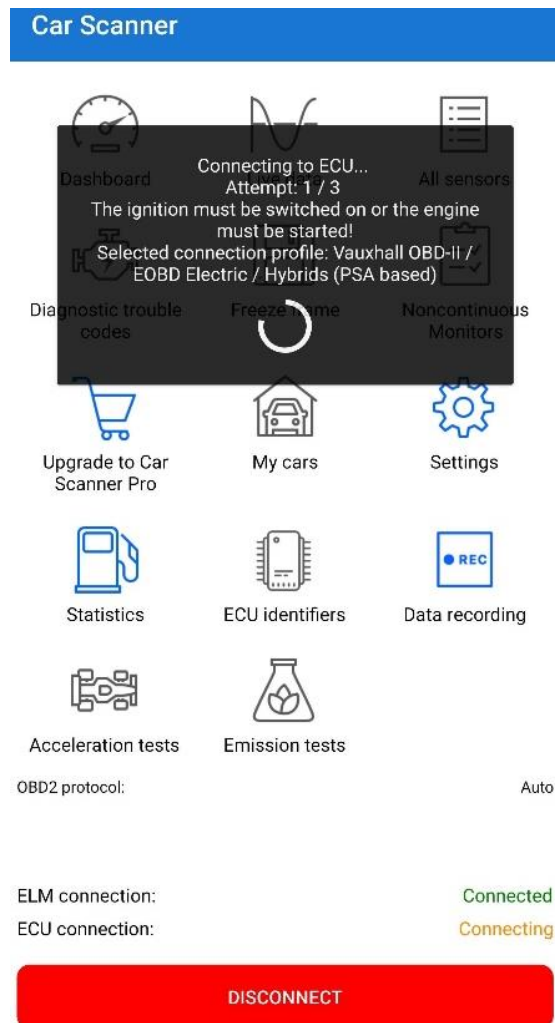


Рисунок 3.8 – Підключення програми Car Scanner через OBD-II

Після успішного встановлення з'єднання між сканером та програмою, можливе детальне тестування автомобіля на предмет виявлення будь-яких несправностей, а також моніторинг поточних параметрів авто, оцінка витрат пального та аналіз інших важливих аспектів роботи транспортного засобу.

Як простий приклад, натискання на першу іконку відкриє панель приладів, де будуть відображені такі дані, як оберти двигуна, швидкість руху автомобіля та температура охолоджувальної рідини (рис. 3.9).

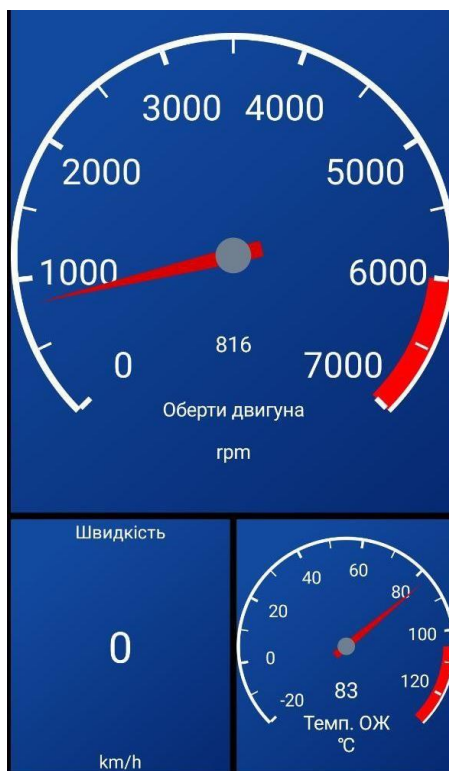


Рисунок 3.9 – Показники стоячого транспортного засобу

У нашому випадку особливу увагу варто приділити проблемі із загорянням індикатора «Check Engine» на панелі приладів автомобіля (рис. 3.10). Цей індикатор сигналізує про наявність проблеми у системі автомобіля, але не уточнює її характер, що може потребувати додаткового діагностування для ідентифікації та виправлення несправностей.



Рисунок 3.10 – Індикатор Check на панелі автомобіля

Щоб провести діагностику з приводу загоряння індикатора «Check Engine», необхідно натиснути на іконку «Моніторинг ЕБУ» (Електронний Блок Управління) (рис. 3.11). Після цього почнеться процес зчитування інформації з датчиків автомобіля, що допоможе виявити джерело проблеми.

A blue rectangular button with the text "Прочитати" (Read) in white.

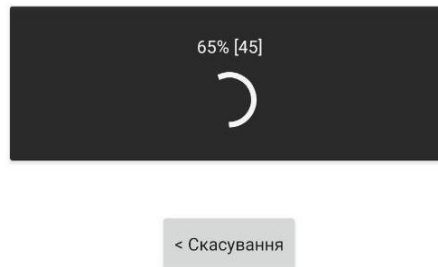


Рисунок 3.11 – Сканування датчиків ЕБУ

Електронний блок управління (ЕБУ) є ключовим контролером в автомобілі, який збирає та обробляє дані з усіх підключених датчиків (рис. 3.12). У сучасних автомобілях може бути встановлено до 80 таких блоків, кожен з яких спеціалізується на керуванні певною системою або функцією транспортного засобу. Ці блоки інтегровані у єдину систему керування, до якої ми отримуємо доступ через сканер OBD-II для діагностики та моніторингу стану автомобіля.



Рисунок 3.12 – Електронний блок управління (ЕБУ)

Після завершення сканування, отримані результати будуть відображені на моніторі у наступний спосіб (рис. 3.13).

| < Назад | | Усі датчики | | < Назад | | Усі датчики | | < Назад | | Усі датчики | |
|--|---|--|---------|---|-----------|--|---|----------|--|-------------|---|
| Напруга (ELM327) | 13.9 V | Швидкість автомобіля | 0 km/h | Вартість палива | 0.08 e | Статус тестів з моменту скидання параметрів | Кут випередження запалення | 7 ° | Вартість палива(загальний) | 0.43 e | Лампа Check Engine:ВИКЛЮЧЕНА |
| Коди несправностей:0 | Недоступний/Завершено | Температура повітря, що всмоктується | 63 °C | Використане паливо | 0 L | Пропуски запалювання: Недоступний/Завершено | Положення дросельної заслінки | 0 % | Використане паливо(загальний) | 0.02 L | Коди несправностей:0 |
| Паливна система: Доступний/Завершено | Компоненти: Доступний/Завершено | Датчик кисню 1 Блок 1 Напруга | 0.72 V | Миттєва витрата палива (л/100 км) | ∞ L/100km | Каталізатор: Доступний/Завершено | Датчик кисню 1 Блок 1 Напруга | 0.72 V | Миттєва витрата палива (л/100 км) | ∞ L/100km | Каталізатор, що обігрівается: Недоступний/Завершено |
| Еvaporative System: Недоступний/Завершено | Secondary Air System: Недоступний/Завершено | Датчик кисню 2 Блок 1 Напруга | 0 V | Моментальна потужність двигуна (від витрати пального) | 3.54 hp | А/C refrigerant: Недоступний/Завершено | Датчик кисню 2 Блок 1 Короткострокова корекція | -3.13 % | Оберти двигуна x1000 | 0.8 rpm | Датчик кисню: Недоступний/Завершено |
| Нагрівач датчика кисню: Доступний/Не завершено | Система EGR: Недоступний/Завершено | Коригування по ДК | 0 V | Оберти двигуна x1000 | 0.8 rpm | Нагрівач датчика кисню: Доступний/Не завершено | Датчик кисню 2 Блок 1 Короткострокова корекція | 0 V | Прискорення | 0 g | Система EGR: Недоступний/Завершено |
| Статус паливної системи | Коригування по ДК | Розрахункове значення навантаження на двигун | 1.96 % | Пройдений шлях | 0 km | Статус паливної системи | Підтримуваний стандарт OBD | Unknown | Пройдений шлях(загальний) | 0 km | Коригування по ДК |
| Температура охолодної рідини | 91 °C | Температура охолодної рідини | 91 °C | Розрахунковий наддув | -0.72 bar | Температура охолодної рідини | поточний час | 19:42:28 | Середня витрата палива (л/100 км) | ∞ L/100km | 91 °C |
| Короткочасна корекція - Блок 1 | -3.91 % | Короткочасна корекція - Блок 1 | -3.91 % | Середня витрата палива (л/100 км) 10 sec | ∞ L/100km | Короткочасна корекція - Блок 1 | Fuel economizer (based on fuel system status and throttle position) | 1 | Середня витрата палива (л/100 км) 10 sec | ∞ L/100km | -3.91 % |
| Довготривала корекція - Блок 1 | 0 % | Довготривала корекція - Блок 1 | 0 % | Вартість палива | e | Довготривала корекція - Блок 1 | Тиск у впускному колекторі (абсолютний) | 29 kPa | Вартість палива(загальний) | e | 0 % |
| Тиск у впускному колекторі (абсолютний) | 29 kPa | Тиск у впускному колекторі (абсолютний) | 29 kPa | Вартість палива | e | Тиск у впускному колекторі (абсолютний) | Оберти двигуна | 789 rpm | Вартість палива(загальний) | e | 29 kPa |
| Оберти двигуна | 789 rpm | Оберти двигуна | 789 rpm | Вартість палива(загальний) | e | Оберти двигуна | Швидкість автомобіля | - | Вартість палива(загальний) | e | 789 rpm |
| Швидкість автомобіля | - | Швидкість автомобіля | - | Вартість палива(загальний) | e | Швидкість автомобіля | Швидкість автомобіля | - | Вартість палива(загальний) | e | - |

Рисунок 3.13 – Результати сканування датчиків ЕБУ

Після аналізу результатів сканування було з'ясовано, що не виявлено жодних незвичайних або ненормальних відхилень, що могли б свідчити про проблеми з роботою автомобіля. Ці дані можуть здатися не дуже корисними для власника авто, але програма може бути застосована для вирішення більш простих та побутових завдань. Наприклад, якщо власник бажає визначити кількість палива, яка буде витрачена на подолання 100 кілометрів, чи розрахувати загальну витрату палива за весь час поїздки, він може скористатися більш інтуїтивно зрозумілим інтерфейсом, який відкривається через іконку «Статистика» (рис. 3.14).



Рисунок 3.14 – Вкладка «Статистика»

Ці та інші функції можуть бути здійснені з будь-якої точки світу, потрібен лише віддалений доступ до мікрокомп'ютера.

3.3 Можливості подальшого удосконалення та модифікацій модулів пристрою

Удосконалення розробленої моделі залежить від додавання нових модулів або функцій в скриптах.

1. Відслідковування автомобіля у випадку крадіжки:

Оскільки наша модель не є частиною офіційної системи автомобіля, її можна розмістити у важкодоступному місці, наприклад, під панеллю приладів або під сидінням. Можливо написати скрипт, який дозволить модулю передавати свої координати кожну хвилину після отримання DTMF-сигналу з певним кодом. Це дозволить поліції швидко реагувати та відстежити автомобіль.

2. Автоматичний виклик швидкої допомоги в разі аварії:

Можна розробити сценарій, який використовуватиме дані з програми сканування автомобіля для визначення аварій. Наприклад, значні збої у роботі датчиків можуть свідчити про серйозні пошкодження. Скрипт може автоматично надсилати координати автомобіля та контактні дані власника службі порятунку.

Ці рішення демонструють, як модифікації та додаткові скрипти можуть значно розширити можливості вашої системи, зробивши її більш адаптивною та функціональною для різних ситуацій.

ВИСНОВКИ

В рамках кваліфікаційної роботи було успішно створено імітаційну модель GSM пристрою, що дозволяє здійснювати віддалену діагностику автомобільних несправностей. Основою для моделі слугував модуль зв'язку SIM7600G-H, який забезпечує передачу даних мобільним шляхом.

Аналіз ринку існуючих систем діагностики показав, що багато з них страждають від високої вартості, складності у використанні та обмеженого функціоналу. Також було виявлено, що існуючі системи часто не здатні забезпечити реальний час обробки даних, що критично важливо для своєчасного виявлення несправностей. Ці недоліки створюють потребу у розробці більш ефективної, доступної та зручної у використанні системи.

Концепція нової системи базується на використанні мікроконтролерів для забезпечення швидкої обробки даних та легкої інтеграції з існуючими автомобільними системами. Мікроконтролери дозволяють розробити модульну систему, яка може бути легко налаштована під різні марки та моделі автомобілів, що робить систему універсальною та масштабованою.

Підбір компонентів був здійснений з урахуванням їхньої продуктивності, вартості та доступності на ринку. Розроблена схема пристрою включає мікроконтролер, різноманітні датчики для вимірювання технічного стану автомобіля та інтерфейс для зв'язку з внутрішньою мережею автомобіля.

Програмне забезпечення для мікроконтролера було розроблено для забезпечення стабільного збору даних з датчиків та їх аналізу. Алгоритми обробки даних оптимізовані для виявлення типових несправностей автомобіля, що дозволяє своєчасно інформувати водія про потенційні проблеми.

Проведене тестування на різних марках автомобілів показало високу ефективність і надійність системи. Вона здатна точно ідентифікувати різні технічні проблеми, надаючи водіям детальну інформацію про стан автомобіля.

Тестування також підтвердило, що система є енергоефективною та стабільною в роботі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Топ програм для діагностики авто – Сканер ELM327. URL: <https://auto-blog.com.ua/uk/skaner-elm327-ua/> (дата звернення: 10.02.2024).
2. Andreev, S. Horelyk, A. Nechausov, D. Saul-Hoze Застосування геоінформаційних технологій для побудови картографічних моделей небезпечних метеорологічних явищ. Системи управління, навігації та зв'язку. Збірник наукових праць. Полтава: ПНТУ, 2022. Т. 1 (67). С. 4-12.
3. Закон Бернуллі – Вікіпедія. URL: https://uk.wikipedia.org/wiki/Закон_Бернуллі (дата звернення: 10.02.2024).
4. Історія науки про зміни клімату – Вікіпедія. URL: https://uk.wikipedia.org/wiki/Історія_науки_про_зміни_клімату (дата звернення: 10.02.2024).
5. Жан Батист Жозеф Фур'є – Вікіпедія. URL: https://uk.wikipedia.org/wiki/Жан_Батист_Жозеф_Фур'є (дата звернення: 10.02.2024).
6. Математичне моделювання або як метеорологи досліджують атмосферу. URL: <https://expedicia.org/matematichne-modelyuvannya-abo-yak-meteo/> (дата звернення: 10.02.2024).
7. Advances in Real-Time Rendering in 3D Graphics and Games. URL: <https://www.advances.realtimerendering.com> (дата звернення: 10.02.2024).
8. Simulation-based education involving online and on-campus models in different European universities - International Journal of Educational Technology in Higher Education. URL: <https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-020-0181-у> (дата звернення: 10.02.2024).
9. Climate Models. URL: <https://climate.mit.edu/explainers/climate-models> (дата звернення: 10.02.2024).

10. Як виробляють штучний сніг: процес і технології. URL: <https://ukr.zapisi.cx.ua/yak-viroblyayut-shtuchniy-snig-proces-i-tekhnologii/> (дата звернення: 10.02.2024).
11. Weather Research & Forecasting Model (WRF) | Mesoscale & Microscale Meteorology. URL: <https://www.mmm.ucar.edu/models/wrf> (дата звернення: 10.02.2024).
12. Global Forecast System (GFS). URL: <https://www.ncei.noaa.gov/products/weather-climate-models/global-forecast> (дата звернення: 10.02.2024).
13. Environmental Modeling Center (EMC). URL: <https://www.emc.ncep.noaa.gov/emc.php> (дата звернення: 10.02.2024).
14. Погодні мапи. Вид із супутника і метеорологічний радар наживо – meteoblue. URL: <https://www.meteoblue.com/uk/weather/maps> (дата звернення: 10.02.2024).
15. Arduino Uno Rev3 від Keyestudio купити в Києві та Україні. URL: <https://arduino.ua/prod2493-arduino-uno-rev3-ot-keyestudio> (дата звернення: 10.02.2024).
16. Microchip Studio for AVR and SAM Devices. URL: <https://www.microchip.com/en-us/tools-resources/develop/microchip-studio> (дата звернення 10.02.2024)
17. VisualMicro. Arduino IDE For Visual Studio. URL: <https://www.visualmicro.com/> (дата звернення: 12.02.2024).
18. Реле – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Реле> (дата звернення: 12.02.2024).
19. Симуляція погодних умов: ентузіаст створив чудернацький механізм у Minecraft – відео. URL: https://games.24tv.ua/simulyatsiya-pogodnih-umov-minecraft-chudernatskiy-novini-sogodni_n1572429 (дата звернення: 12.02.2024).

20. Enviro - Sky and Weather | Particles/Effects | Unity Asset Store. URL: <https://assetstore.unity.com/packages/tools/particles-effects/enviro-sky-and-weather-33963> (date of access: 12.02.2024).

21. Двоканальний димер 220В для Arduino від RobotDyn купити в Києві та Україні. URL: <https://arduino.ua/prod3118-dimmer-220v-2-kanala-dlya-arduino-ot-robotdyn> (дата звернення: 12.02.2024).

22. Інтернет речей (IoT) в сільському господарстві: 9 прикладів використання технологій для точного землеробства (і виклики, які слід врахувати). URL: <https://www.agrilab.ua/internet-rechej-iot-v-silskomu-gospodarstvi-9-prykladiv-vykorystannya-tehnologij-dlya-tochnogo-zemlerobstva-i-vyklyku-yaki-slid-vrahuvaty/> (дата звернення: 12.02.2024).

23. Модуль потужного реле 5В 30А з опторозв'язкою купити в Києві та Україні. URL: <https://arduino.ua/prod1767-1-channel-5v-30a-high-power-relay-module-for-arduino> (дата звернення: 12.02.2024).

24. Arduino UNO – популярна плата розробки – IT Master - електроніка та програмування. URL: <https://itmaster.biz.ua/directory/kits-nabory/arduino-uno.html> (дата звернення: 13.02.2024).

25. Arduino UNO: A Short Review – EduGonist. URL: <https://www.edugonist.com/arduino-uno/> (дата звернення: 13.02.2024).