

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: *Комп'ютерна система зчитування та відображення
показників двигуна авто через CAN шину*

Виконав: студент IV курсу, групи СІс-42
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Гарнік В.Р.

(прізвище та ініціали)

Керівник

(підпис)

Луцків А.М.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Луцик Н.С.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

Литвиненко Я.В.

(прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« ____ » _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студенту Гарніку Владиславу Ростиславовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система зчитування та відображення показників двигуна авто через CAN шину

Керівник роботи Луцків Андрій Мирославович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «26» квітня 2024 року № 4.7-468

2. Термін подання студентом завершеної роботи 24.06.2024 р.

3. Вихідні дані до роботи Принцип організації та функціонування CAN-шини, стандарт OBD II показники агрегатів електронного блоку управління автомобіля, тип екрану

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз вимог до комп'ютерної системи зчитування даних з CAN-шини. 2. Проектування комп'ютерної системи зчитування та відображення показників двигуна авто 3. Програмне забезпечення комп'ютерної системи зчитування та відображення показників двигуна авто 4. Безпека життєдіяльності, основи охорони праці. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Концептуальна архітектура системи зчитування та візуалізації даних з CAN-шини.
2. Структурна схема організації системи зчитування даних з CAN-шини.
3. Алгоритм розрахунку споживання води.
4. Структурна схема модулів програмного забезпечення

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Пилипець М.І., д.т.н., проф. каф. МТ</i>		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Розробка і затвердження технічного завдання</i>	<i>26.04-28.04.2024</i>	
2.	<i>Аналіз технічного завдання</i>	<i>28.04-03.05.2024</i>	
3.	<i>Аналіз вимог до комп'ютерної системи зчитування даних з CAN-шини</i>	<i>05.05-08.05.2024</i>	
4.	<i>Проектування комп'ютерної системи зчитування та відображення показників двигуна авто</i>	<i>08.05-18.05.2024</i>	
5.	<i>Програмне забезпечення комп'ютерної системи зчитування та відображення показників двигуна авто</i>	<i>18.05-25.05.2024</i>	
6.	<i>Розробка інструкцій щодо використання комп'ютерної системи</i>	<i>26.05-05.06.2024</i>	
7.	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>05.06-10.06.2024</i>	
8.	<i>Оформлення кваліфікаційної роботи</i>	<i>10.06-14.06.2024</i>	
9.	<i>Попередній захист кваліфікаційної роботи</i>	<i>14.06-19.06.2024</i>	
10.	<i>Захист кваліфікаційної роботи</i>	<i>24.06-28.06.2024</i>	

Студент

(підпис)*Гарнік Владислав Ростиславович*_____
(прізвище та ініціали)

Керівник роботи

(підпис)*Луцків Андрій Мирославович*_____
(прізвище та ініціали)

АНОТАЦІЯ

Комп'ютерна система зчитування та відображення показників двигуна авто через CAN шину // Кваліфікаційна робота на здобуття освітнього ступеня бакалавр // Гарнік Владислав Ростиславович // ТНТУ, спеціальність 123 «Комп'ютерна інженерія»// Тернопіль, 2024 // с.– 77 , рис. – 51 , табл. – 6, аркушів А1 – 5, бібліогр. – 24.

Ключові слова: комп'ютерна система, зчитування, відображення, показник, двигун, авто.

У роботі запропоновано проект системи, яка забезпечує зчитування показників двигуна та відображає їх значення в реальному часі, а також щосекундно виконує записи цих показників у CSV-файл.

При проектуванні комп'ютерної системи, яка отримує дані з CAN-шини використано сучасні апаратні та програмні засоби. Апаратні засоби, які забезпечують отримання даних і їх відображення включають в себе: Raspberry PI 3, сенсорний екран 3,5”RPI LCD.

Для забезпечення живлення мінікомп'ютера Raspberry PI розроблено відповідну схему, яка дає змогу перетворювати постійну напругу 12 В від автомобільного акумулятора у напругу 5В та використовує лінію живлення АСС. При цьому у схемі використовується мікроконтролер Attiny85, конвертор постійного струму LM2596 та регулятор постійної напруги 78L33, які дозволяють безпечно завершити роботу Raspberry PI протягом 45 секунд після вимкнення двигуна авто.

Програмне забезпечення комп'ютерної системи зчитування та відображення показників двигуна через CAN-шини забезпечує процеси комунікації між апаратними пристроями і складається з 5 модулів, які написані мовою програмування Python.

ABSTRACT

A computer system for obtaining and displaying car engine indicators via the CAN bus// Bachelor thesis // Harnik Vladyslav// TNTU, speciality 123 «Computer engineering»// Ternopil, 2024 // p.– 77, fig. – 51 , tab. – 6, posters A1 – 5, ref. – 24.

Keywords: computer system, reading, displaying, indicator, engine, car.

The paper proposes a project of a system that provides reading of engine indicators and displays their values in real time, as well as records these indicators in a CSV file every second. When designing a computer system that receives data from the CAN bus, modern hardware and software tools were used.

Hardware that provides data acquisition and display includes: Raspberry PI 3, 3.5" RPI LCD touch screen. To provide power for the Raspberry PI minicomputer, a suitable circuit has been developed that allows you to convert a constant voltage of 12 V from a car battery into a voltage of 5 V and uses the ACC power line. At the same time, the circuit uses an Attiny85 microcontroller, an LM2596 DC-DC converter, and a 78L33 DC voltage regulator, which allow the Raspberry PI to be safely terminated within 45 seconds after the car engine is turned off.

The software of the computer system for reading and displaying engine indicators via CAN buses provides communication processes between hardware devices and consists of 5 modules written with Python.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ВИМОГ ДО КОМП'ЮТЕРНОЇ СИСТЕМИ ЗЧИТУВАННЯ ДАНИХ З CAN-ШИНИ.....	9
1.1 Аналіз CAN-протоколу	9
1.2 Стандарти передачі даних з електронних блоків сучасних автомобілів	15
РОЗДІЛ 2 ПРОЕКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ЗЧИТУВАННЯ ТА ВІДОБРАЖАННЯ ПОКАЗНИКІВ ДВИГУНА АВТО	18
2.1 Концептуальна архітектура комп'ютерної системи зчитування та візуалізації показників двигуна авто	18
2.2 Пристрій управління системою зчитування та візуалізації даних з CAN-шини	21
2.3 Дисплей для відображення даних показників CAN-шини.....	26
2.4 Проектування схеми живлення для Raspberry PI.....	29
2.4.1 Характеристики Attiny85	32
2.4.2 Перетворювач струму «постійний-постійний» LM2596	34
2.4.3 Регулятор лінійної напруги 78L33	35
РОЗДІЛ 3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ЗЧИТУВАННЯ ТА ВІДОБРАЖЕННЯ ПОКАЗНИКІВ ДВИГУНА АВТО	37
3.1 Встановлення операційної системи Raspbian Lite	37
3.2 Налаштування SSH та доступу до WiFi	40

					КС КРБ 123.311.00.00 ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Гарнік В.Р.			Комп'ютерна система зчитування та відображення показників двигуна авто через CAN шину	Літ.		Арк.	Аркушів
Перевір.		Луцків А.М.						6	
Реценз.						ТНТУ, каф. КС, гр. СІс-42			
Н. Контр.		Луцик Н.С.							
Затверд.		Осухівська Г.М.							

3.3	Налаштування програмного підключення сенсорного екрану до Raspberry PI	44
3.4	Встановлення і налаштування віртуальної клавіатури для сенсорного екрану	46
3.5	Розробка програмного забезпечення зчитування та відображення показників двигуна	49
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ		55
4.1	Організація служби охорони праці на підприємстві	55
4.2	Заходи, які забезпечують створення оптимальних метеорологічних умов у приміщеннях з використанням ПК	58
ВИСНОВКИ		62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		63
Додаток А Технічне завдання		
Додаток Б Перелік елементів		

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						7
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ВСТУП

Досягнення та прогрес сучасних технологій побудови комп'ютерних систем стимулює їх впровадження у різних сферах діяльності, починаючи від звичайних повсякденних побутових пристроїв, завершуючи складними і надзвичайно точними комплексами дослідження мозку. Не винятком впровадження розумних технічних рішень є і галузь автомобілебудування. У сучасному автотранспорті, технології не лише вдосконалюють процес водіння, але й роблять його безпечнішим та ефективнішим завдяки використанню комп'ютерних систем.

Однією з ключових технологій, що забезпечує надійне функціонування та діагностику двигуна автомобіля, є CAN-шини. Ці шини дозволяють зчитувати та відображати різноманітні показники роботи агрегатів транспортних засобів в реальному часі. Це у свою чергу дає змогу водію контролювати технічну справність автомобіля та вчасно реагувати на проблеми і їх швидке усунення.

Окрім цього, розвиток у галузі комп'ютерної інженерії та IoT дають змогу провести модернізацію автомобілів, які виготовлені на початку 2000-х років, коли ще не було такого розвитку інформаційних технологій. Транспортні засоби, які випускались на початку 21 століття ще не мали таких ефективних засобів для моніторингу показників стану двигуна або вони були не надто точними. Для водіїв, які не хочуть, або не можуть замінити своє авто на нове, актуальним є впровадження комп'ютерної системи, яка б давала змогу отримувати дані про температуру охолоджуючої рідини, оберти двигуна, миттєве споживання палива при різних умовах пересування та ряду інших.

У кваліфікаційній роботі пропонується розробка проекту комп'ютерної системи, що базується на використанні CAN-шини для зчитування та відображення показників роботи двигуна, що дозволить підвищити якість контролю та управління технічним станом автотранспортних засобів.

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ВИМОГ ДО КОМП'ЮТЕРНОЇ СИСТЕМИ ЗЧИТУВАННЯ ДАНИХ З CAN-ШИНИ

1.1 Аналіз CAN-протоколу

CAN-протокол, який був розроблений фірмою Bosh, сьогодні є стандартом, який використовується при проектуванні промислових комп'ютерних систем, зокрема, при управлінні пристроями в автомобілях. Стандартом OBDII передбачається застосування каналів «CAN-High» і «CAN-Low». На сьогоднішній день цей протокол забезпечує високу швидкість і демонструє високий рівень надійності при обміні повідомленнями сучасного авто.

Серед особливостей, що одночасно є і перевагою стандарту CAN, потрібно відмітити відсутність обмежень та не регламентованість швидкості передачі даних по шинах, а тип контролерів, які використовуються для управління може бути довільним.

Підтримувана швидкість обміну повідомленнями коливається у діапазоні від 20 КБ/с до 1 МБ/с і залежить від типу конкретного контролера.

Як і будь-який протокол, протокол CAN визначає деяку множину правил. Ці правила створені та імплементовані у галузі автомобілебудування та використовуються в якості стандартів при розробці проекту окремих вузлів управління.

Протокол CAN може бути ефективно застосований у випадку проектування мереж з вимогами передачі пакетів даних розміром 8Б при швидкості до 1МБ/с. Це обумовлено розвиненістю засобів виявлення та контролю помилок.

					КС КРБ 123.311.00.00 ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Гарнік В.Р.			Аналіз вимог до комп'ютерної системи зчитування даних з CAN-шини	Літ.		Арк.	Аркушів
Перевір.		Луцків А.М.						9	
Реценз.						ТНТУ, каф. КС, гр. СІс-42			
Н. Контр.		Луцик Н.С.							
Затверд.		Осухівська Г.М.							

Протокол CAN строго не визначає вимог і властивостей, якими повинно володіти фізичне середовище для обміну повідомленнями. Тому його доцільно використовувати при передачі даних між вузлами системи із застосуванням, наприклад, витой пари чи оптичних каналів комунікації. Цей тип протоколу дає змогу сформувати передачу даних на канальному рівні, тобто дозволяє створювати пакетні повідомлення, накладати обмеження на поширення помилок, арбітраж та підтвердження одержання повідомлень. Структуру вузла мережі на основі CAN-протоколу представлено на рис. 1.1.



Рисунок 1.1 – Вузол мережі CAN

При організації комунікації між пристроями різних виробників можуть також застосовуватися протоколи, подібні до CANopen. Вони також функціонують на прикладному рівні мережі. Проте, у випадку, коли не виникає такої необхідності, то, зазвичай, найбільш оптимально використовувати оригінальний набір правил.

Вузол мережі на основі CAN-протоколу передбачає використання:

- зовнішнього мікроконтролера;
- CAN-контролера;
- приймача;

Сьогодні пропонуються рішення мікроконтролерів, які містять в собі вбудований CAN контролер. Це дозволяє спростити організацію мережі та підвищити ефективність виконання схемотехнічного рішення при реалізації відповідних вузлів. Для окремих спеціалізованих завдань можна також використати автономні CAN-контролери, що працюють з інтерфейсом SPI. При цьому, пристрій, який відіграє роль ресивера чи передавача, підключається за допомогою термінаторів з опором 120 Ом. Термінатори фіксуються на двох кінцях виті пари.

Для того, щоб згенерувати логічну одиницю потрібно на два задіяних провідники виті пари подати напругу величиною, що рівна половині напруги між нулем та «VCC».

Для того, щоб згенерувати логічний нуль потрібно на виті пару подати диференціальну напругу, як показано на рис. 1.2.

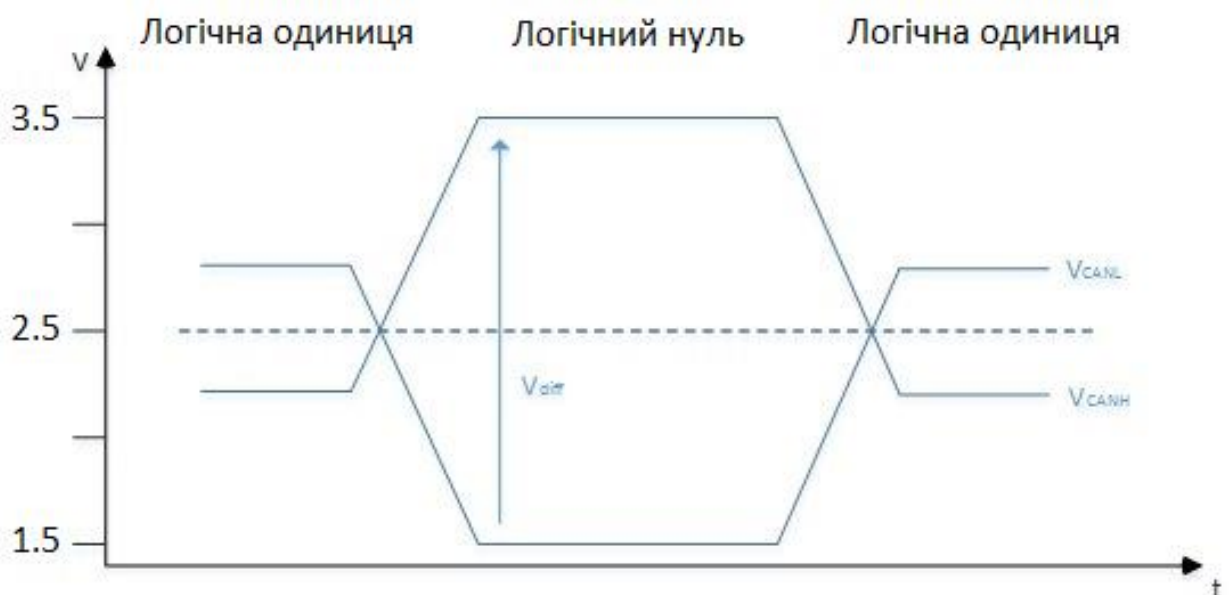


Рисунок 1.2 – Логічні рівні сигналу на CAN-шині

Забезпечення швидкості передачі даних по витій парі зі швидкістю на рівні 1 Мбіт/с можливе в тому випадку, коли довжина кабелю не перевищує 40 м. У деяких літературних джерелах [1-3], зазначають, що швидкість передачі даних на рівні до 10 Кбіт/с можлива, якщо довжина мережі становитиме приблизно 1,5 км.

На рис. 1.3 продемонстровано структуру пакету повідомлення, яке передається по CAN-шині.



Рисунок 1.3 – Структура пакету повідомлення CAN

Безпосереднє створення повідомлення покладено на CAN контролер, а програмне забезпечення прикладного рівня відповідає лише за встановлення унікального ідентифікатора, довжини і надання байтів даних. Зміна внутрішньої структури пакету даних є обмеженою, проте існує можливість змінити деякі його параметри. Параметри, які можна змінити у пакеті при застосування CAN-шини показано у табл. 1.1.

Таблиця 1.1 – Параметри CAN шини

Поле	Діапазон
Ідентифікатор (11 – біт)	0 – 0x7FF
Ідентифікатор (29 – біт)	0 – 0x1FFFFFFF
Довжина повідомлення	0 – 8
Дані пакету	від 0 до 8 байт даних у пакеті

Для забезпечення ідентифікації повідомлення, яке відправляється у пакеті, застосовується унікальний ідентифікатор. При цьому повідомлення надходить до всіх вузлів мережі, а значення ідентифікатора дозволяє встановити котрому з пристроїв його слід опрацювати.

Висока пропускна здатність шини, яка функціонує за правилами протоколу CAN може бути забезпечена шляхом зменшення максимального розміру повідомлення, тобто менше 8 Б.

Пріоритетність повідомлень визначається значенням величини ідентифікатора – чим менше значення ідентифікатора, тим вищий пріоритет повідомлення.

Налаштування швидкості передачі даних через CAN шину відбувається шляхом формування квантів часу, на відміну від інших протоколів, які підтримують послідовну передачу, застосовуючи при цьому дільник швидкості.

Зазвичай швидкість передачі даних через CAN-шину кратна 10 або 25 і може набувати таких значень як 10, 20, 50, 100, 125, 500, 800 Кбіт/с, або 1Мбіт/с. налаштування рівня швидкості передачі даних можливе через спеціалізовані програми, наприклад, PcanView (рис. 1.4).

The image shows a configuration window for PcanView. It includes a 'Baudrate' dropdown menu set to '500kBaud'. There are checkboxes for 'listen only' and 'high res. timer'. Below these are sections for 'obsolete devices (G'W-001/G'W-002)' with input fields for 'BTR0' (00) and 'BTR1' (1C), and 'devices of 3rd or 4th generation' with input fields for 'BTR Ext' (00050741) and another field (401C0002). At the bottom, there is a section for 'logical devices with two channels' with radio buttons for 'CAN Channel 0' (selected) and 'CAN Channel 1'.

Рисунок 1.4 – Налаштування швидкості в програмі PcanView

Аналізуючи рис. 1.4, можна помітити, що при виборі значення швидкості, наприклад, на рівні 500Кбіт/с, автоматично встановлюються всі інші залежні параметри за замовчуванням. Проте інколи виникають ситуації, коли потрібно задати нешаблонні значення швидкості.

Для прикладу, деякий пристрій, що підтримує CAN-протокол, потребує швидкості передачі даних на рівні 83 Кбіт/с. Тоді потрібно обчислити значення інших параметрів шляхом ручного розрахунку або з використанням спеціалізованого калькулятора. При ручному розрахунку швидкості потрібно зважати на те, що при передачі кожного біту повідомлення буде використовуватися декілька квантів, а інтервал передачі включатиме три сегменти, як показано на рис. 1.5.

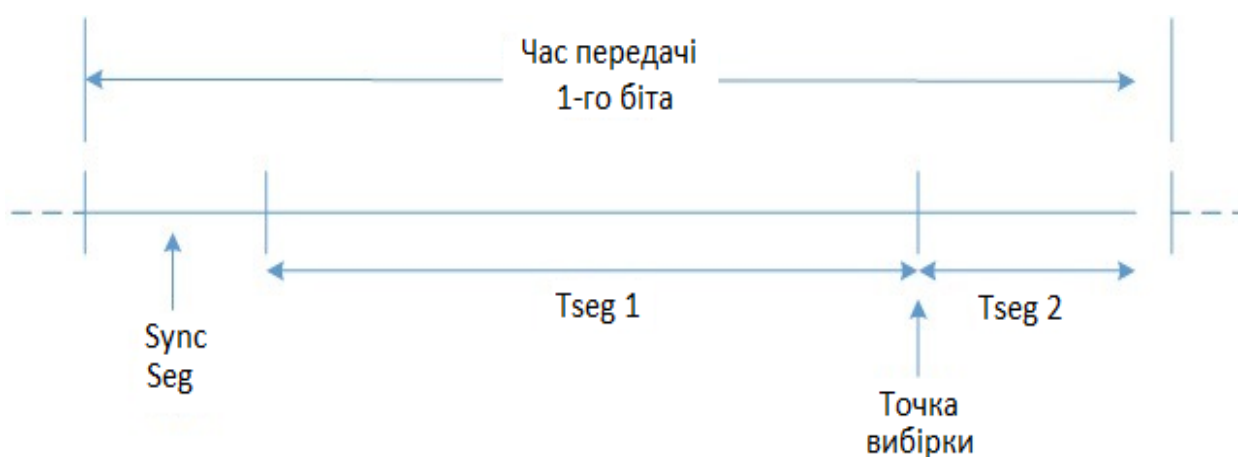


Рисунок 1.5 – Сегменти при передачі даних

Як видно з рис. 1.5, перший сегмент при передачі даних є фіксованим і рівний одному кванту. Наступні сегменти можна налаштовувати, тобто кількість квантів задається користувачем в діапазоні [1].

На межі другого і третього сегментів знаходиться точка семплювання (вибірки), тобто користувач має можливість задати ширину імпульса синхронізації для того, щоб налаштувати швидкість ресивера (пристрою, що отримує повідомлення) в проміжку, що відповідає значенням від 1 до 4 квантів часу.

1.2 Стандарти передачі даних з електронних блоків сучасних автомобілів

Діагностика стану систем сучасного автомобіля, окремих його вузлів чи компонентів, сьогодні здійснюється із застосуванням сканерів, які підтримують стандарт OBD II. Даний стандарт підтримує п'ять базових протоколів, які забезпечують обмін повідомленнями між сканером та блоком керування авто.

За допомогою DLC-роз'єму, який відповідає нормам стандарту SAE J1962, можна виконати підключення зовнішнього пристрою зчитування даних до електронного блоку. На рис. 1.6 показано розташування контактів цього роз'єму та їх позначення.

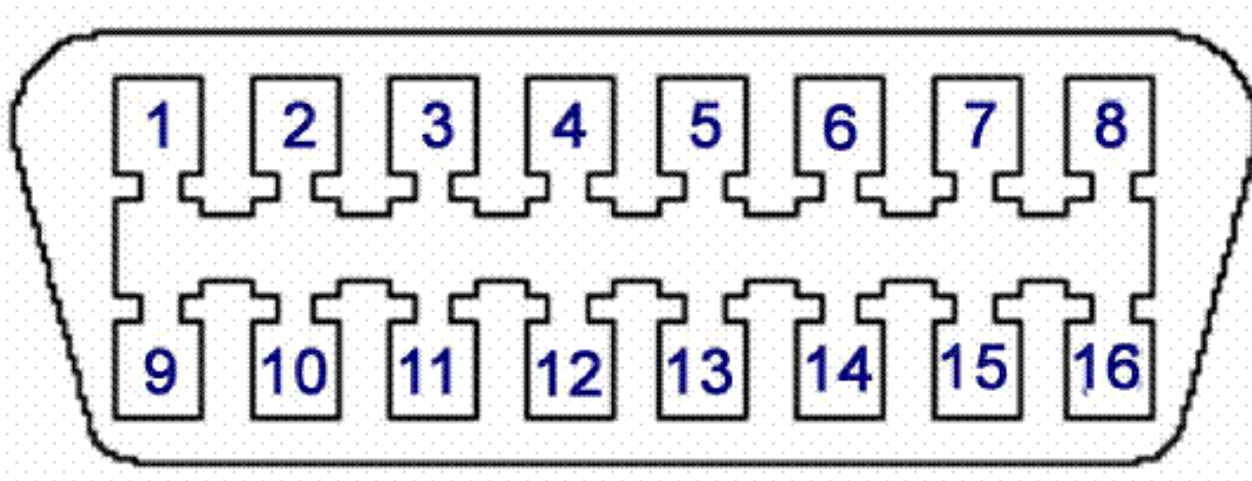


Рисунок 1.6 – Роз'єм DLC

Загалом, роз'єм складається з 16 виводів, які розташовано рівномірно у 2 ряди. Функціональне призначення кожного контакту роз'єму DLC описано у табл. 1.2.

Таблиця 1.2 – Призначення контактів роз'єму DLC

№ контакту	Призначення	Примітка
1	Протокол виробника автомобіля (ОЕМ)	При включенні замка запалювання — комутація +12 В

№ контакту	Призначення	Примітка
2	Шина + (Bus positive Line)	2SAE-J1850 PWM, SAE-1850 VPW.
3	-	-
4	Заземлення	Заземлення від кузова автомобіля
5	Сигнальне заземлення	
6	Лінія CAN-High високошвидкісної шини CAN Highspeed	ISO 15765-4, SAE- J2284
7	K-Line.	ISO 9141-2 та ISO 14230
8		
9	Лінія CAN-Low, низькошвидкісної шини CAN Lowspeed	
10	Шина – (Bus negative Line)	SAE-J1850 PWM, SAE –1850 VPW
11		
12		
13		
14	Лінія CAN-Low високошвидкісної шини CAN	ISO 15765-4, SAE- J2284
15	L-Line	ISO 9141-2 та ISO 14230
16	Живлення +12 В від аккумуляторної батареї	

При описі контактів у табл. 1.2 для деяких з них не вказано функціональне призначення. Це пов'язано з тим, що виводи можуть використовуватися на власний розсуд виробника і залежать від того, які саме функції з точки зору виробника будуть реалізовані. Всі інші контакти є стандартизованими і не можуть зазнавати змін.

Для забезпечення ергономічних вимог передбачається, що розташування роз'єму DLC не повинно перевищувати 61 см від рульового колеса. Проте, деякі виробники авто встановлюють роз'єм, який відповідає формфактору роз'єму стандарту OBD-II, але у ньому не передбачається підтримки протоколів цього стандарту.

Для того, щоб забезпечити можливість читання інформації з електронного блоку керування авто застосовуються спеціалізовані сканери, які підтримують заводські протоколи виробника. До таких авто можна віднести, наприклад, окремі моделі Opel. Іноді трапляються такі випадки, що протоколи стандарту OBD-II імплементовані в авто, а сам роз'єм не підтримує DLC.

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						17
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 2 ПРОЕКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ЗЧИТУВАННЯ ТА ВІДОБРАЖАННЯ ПОКАЗНИКІВ ДВИГУНА АВТО

2.1 Концептуальна архітектура комп'ютерної системи зчитування та візуалізації показників двигуна авто

У кваліфікаційній роботі виконується проектування та розробка системи зчитування даних з CAN-шини на основі стандарту OBD-II для автомобіля BMW M3 випуску 2000 років.

Сучасні транспортні засоби надають багато можливостей контролю показників їхньої роботи, зокрема, температури охолоджуючої рідини двигуна, тиску у шинах, DTC і перетворюються на засоби збору інформації для подальшої оптимізації виробництва та підвищення надійності експлуатації.

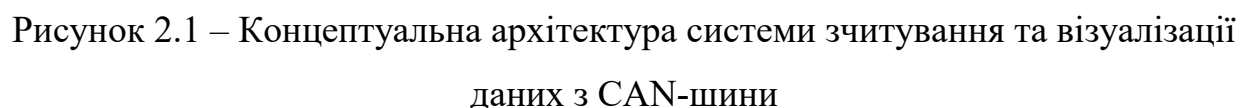
Усі зібрані дані також повинні бути зареєстровані так, щоб їх можна було проаналізувати в майбутньому (наприклад, аналіз ефективності використання потужності двигуна). Результатом виконання кваліфікаційної роботи повинен бути проект реалізації комп'ютерної системи, яка інтегрується у 3D-панель та кріпиться у зоні центральної консолі автомобіля.

Основними складовими комп'ютерної системи зчитування показників двигуна через CAN-шину є:

- Raspberry PI 3 – центральний пристрій керування системи зчитування та відображення показників двигуна;
- сенсорний екран – безпосередньо відповідає за відображення даних про температуру охолоджуючої рідини, передача, яка увімкнена у конкретний момент часу, навантаження та оберти двигуна та ін.;
- блок живлення – забезпечує живленням мікроконтролер Raspberry PI та сенсорний дисплей;

					КС КРБ 123.135.00.00 ПЗ							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Гарнік В.Р.			Проектування комп'ютерної системи зчитування та відображення показників двигуна авто				Літ.	Арк.	Аркушів	
Перевір.		Луцків А.М.									18	
Реценз.									ТНТУ, каф. КС, гр. СІс-42			
Н. Контр.		Луцик Н.С.										
Затверд.		Осухівська Г.М.										

На рис. 2.1 проілюстровано загальну концепцію архітектурного рішення щодо побудови системи зчитування та візуалізації даних з CAN-шини.



Програмне забезпечення реалізується через сукупність модулів, написаних мовою програмування Python. Після опрацювання даних ЕБУ на рівні Raspberry PI виконується модуль, що відповідає за відображення інформації для водія на 3,5 -дюймовому резистивному сенсорному екрані. Raspberry PI 3 та сенсорний екран комунікують за допомогою шини SPI. На рис. 2.2 показано структурну схему організації системи зчитування та відображення даних про показники транспортного засобу.

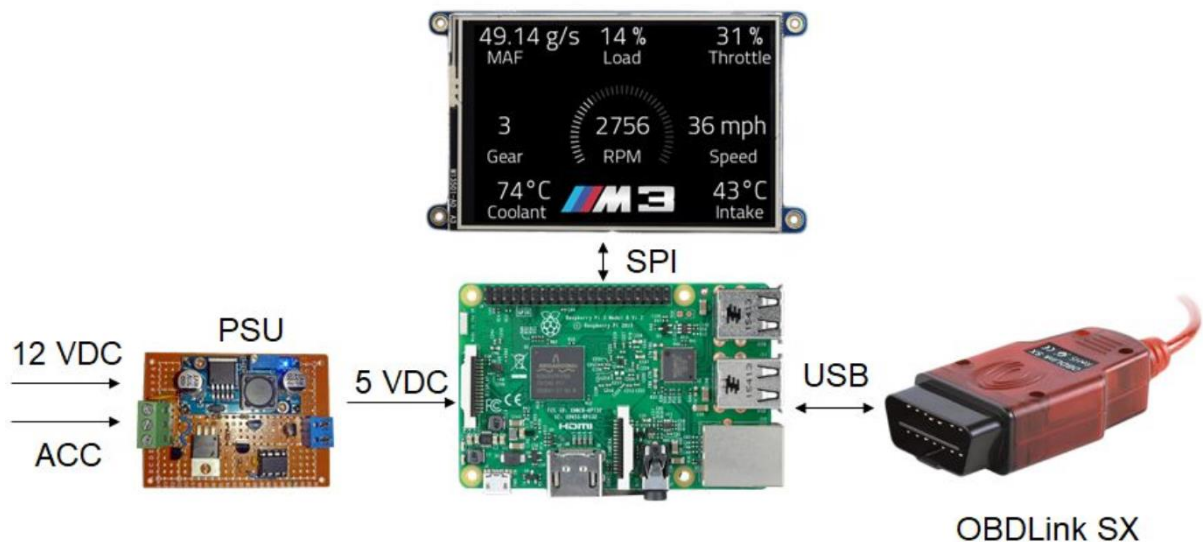


Рисунок 2.2 – Структурна схема організації системи зчитування даних з CAN-шини

У системі зчитування та візуалізації даних з CAN-шини передбачається можливість запису історичних даних у файл, тобто Raspberry PI щосекунди реєструє інформацію показників температури, обертів двигуна та інших важливих характеристик у файл CSV.

У подальшому такі дані будуть доступні водієві для перегляду та аналізу працездатності, а також надійності авто. Raspberry PI працює завдяки живленню від модуля управління, що підключений до акумулятора через лінії замка запалювання.

Модуль перетворення та управління живленням використовує мікроконтролер Atmel Attiny85, коли запалювання увімкнене. У випадку виключення запалювання, контролер забезпечує подачу живлення ще протягом

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

45 секунд для того, щоб безпечно вимкнути Raspberry PI. Наступний етап роботи полягає у виборі та аналізі технічних характеристик апаратних складових, розробці схем підключення та написанні програмного забезпечення системи зчитування та візуалізації показників роботи двигуна.

2.2 Пристрій управління системою зчитування та візуалізації даних з CAN-шини

На сьогоднішній день мінікомп'ютер Raspberry PI вважається одним з найбільш ефективних та продуктивних засобів при створенні комп'ютерних систем, які реалізуються за допомогою IoT пристроїв і дають змогу забезпечити розробку як прототипу системи, так і повноцінної промислово-орієнтованої структури.

При розробці системи зчитування та візуалізації даних з CAN-шини транспортного засобу пропонується використати третю версію Raspberry PI, а саме Model B.

Обґрунтованість цього вибору обумовлена вартістю мікроконтролера та габаритними розмірами, які не перевищують розмірів банківської карти. Окрім цього, Raspberry PI 3 можна підключати до екранів практично будь-яких його виробників, що важливо при візуалізації показників стану авто. Зважаючи на наявність достатньої кількості USB-портів, при проектуванні системи зчитування та візуалізації даних з CAN-шини можна підключати пристрої вводу, зокрема миші і клавіатури, які підтримують USB-інтерфейс.

Raspberry PI можна віднести до доволі простих та зручних програмно-апаратних пристроїв, які можна ефективно застосовувати при вивченні програмування мікроконтролерів, реалізації прототипів електронних систем, використанні як персонального ПК для вирішення офісних задач користувачів. Порти та зовнішній вигляд RPI 3 показано на рис. 2.3 [10].

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

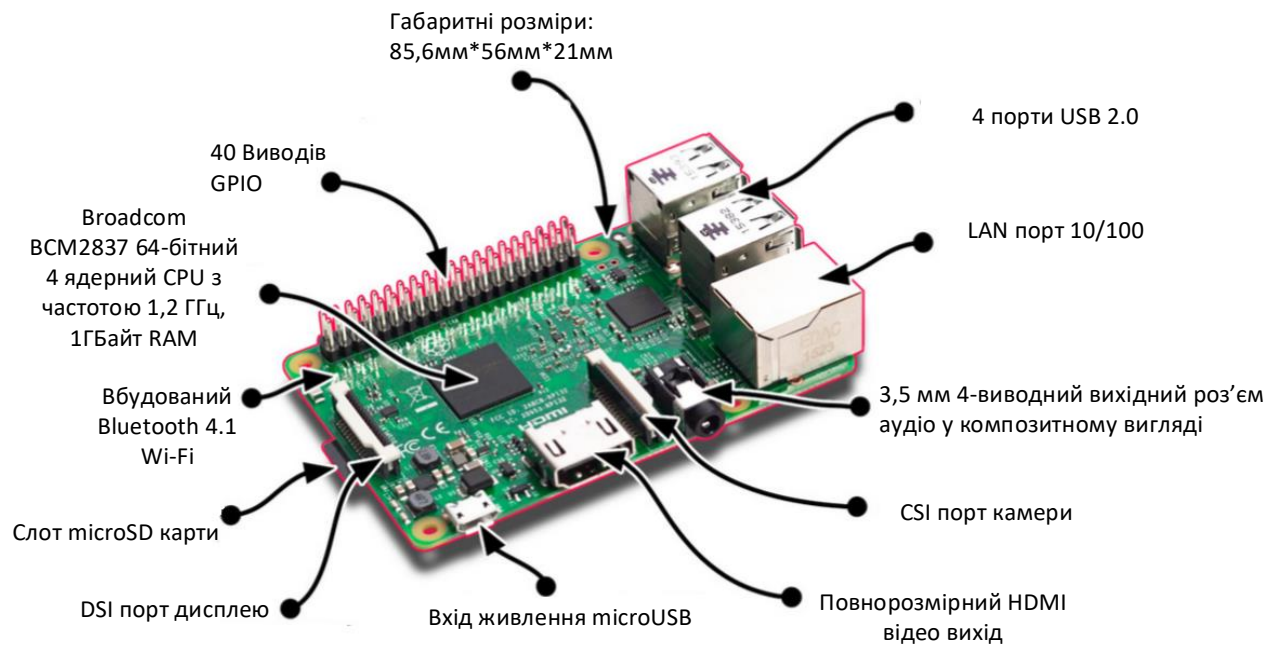


Рисунок 2.3 – Типовий вигляд Raspberry PI 3

Лінійка мінікомп'ютерів Raspberry PI на сьогодні є дуже потужною і включає багато різних модифікацій, які дають змогу оптимально розв'язувати задачі поставлені перед розробниками комп'ютерних систем. Порівняльна характеристика різних типів мікроконтролерів Raspberry PI представлено у табл. 2.1.

Таблиця 2.1 – Різновиди Raspberry PI

Модель	SoC	Тактова частота, МГц	RAM	Технології Ethernet	Wireless	Blue-tooth
Raspberry Pi Model A+	BCM2835	700	512МБ	-	-	-
Raspberry Pi Model B+	BCM2835	700	512МБ	100Base-T	-	-
Raspberry Pi 2 Model B	BCM2836/7	900	1 ГБ	100Base-T	-	-

Модель	SoC	Тактова частота, МГц	RAM	Технології Ethernet	Wireless	Blue-tooth
Raspberry Pi 3 Model B	BCM2837 A0/B0	1200	1 ГБ	100Base-T	802.11n	4.1
Raspberry Pi 3 Model A+	BCM2837 B0	1400	512МБ	-	802.11 ac/n	4.2
Raspberry Pi 3 Model B+	BCM2837 B0	1400	1 ГБ	1000Base-T	802.11 ac/n	4.2
Raspberry Pi 4 Model B	BCM2711	1500	2 ГБ	1000Base-T	802.11 ac/n	5.0
Raspberry Pi 4 Model B	BCM2711	1500	4 ГБ	1000Base-T	802.11 ac/n	5.0
Raspberry Pi 4 Model B	BCM2711	1500	8 ГБ	1000Base-T	802.11 ac/n	5.0
Raspberry Pi Zero	BCM2835	1000	512МБ	-	-	No
Raspberry Pi Zero W	BCM2835	1000	512МБ	-	802.11n	4.1
Raspberry Pi 400	BCM2711	1800	4ГБ	1000Base-T	802.11 ac/n	5.0

Для проектування системи зчитування та візуалізації параметрів транспортного засобу використано модель Raspberry PI 3 Model B.

Для забезпечення комунікації між компоненти RPI та здатності підключення зовнішніх периферійних пристроїв використовується інтерфейс GPIO. Для інтерфейсу GPIO визначено три основні функціональні призначення:

- забезпечення живленням зовнішніх пристроїв з двома можливими типами номіналів напруги (3,3 В або 5 В), з врахуванням особливостей периферійного пристрою;
- заземлення – дозволяє організувати заземлення зовнішнього компонента;
- передача сигналів – дає змогу організувати обмін даними та відповідними сигналами керування між пристроями, які підключаються до мікроконтролера та самим Raspberry PI.

Перевага використання GPIO полягає у гнучкості програмного налаштування виводів, яка проявляється у можливості використання одного і того ж контакту в якості вхідного або вихідного в залежності від потреб [11]. Схему і характеристики інтерфейсу GPIO наведено на рис. 2.4.

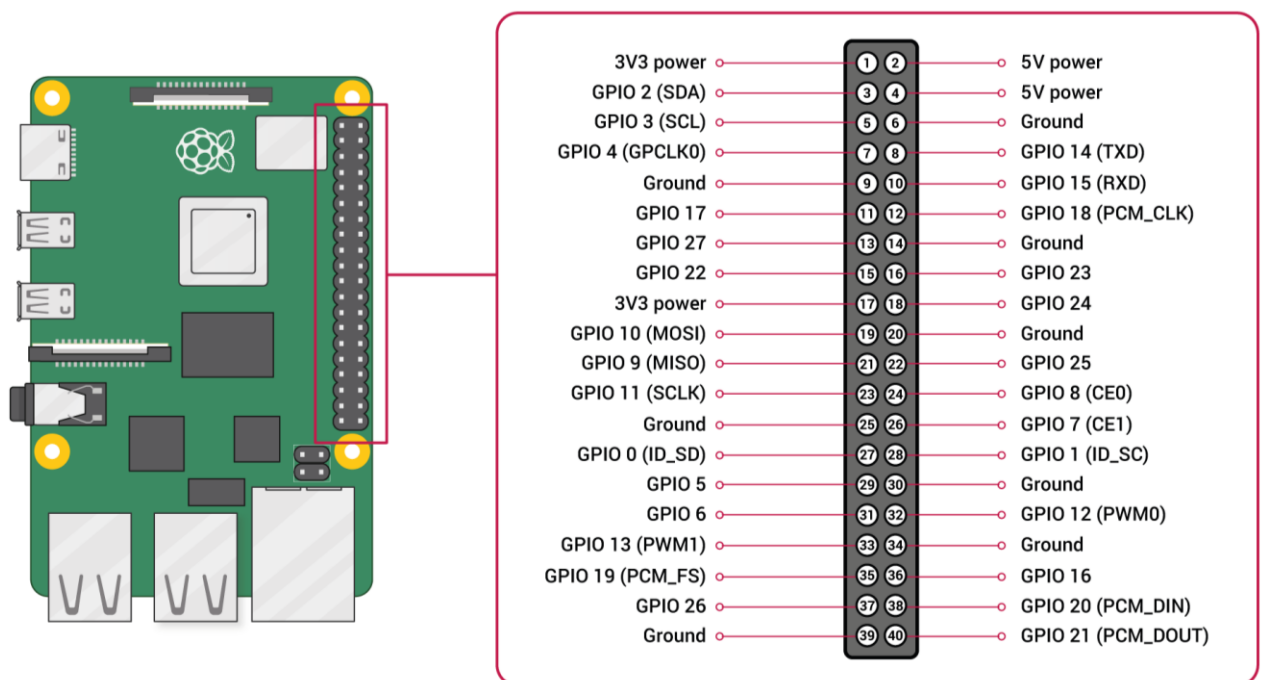


Рисунок 2.4 – Інтерфейс GPIO

Як було зазначено раніше, контакти інтерфейсу GPIO можуть бути гнучко налаштовані і в залежності від ситуації можуть витупати як вхідні контакти, так і вихідні. Проте в інтерфейсі наявні два виводи спеціального призначення – VSM0 (27) і VSM1(28). Дані виводи використовуються з метою під'єднання плат

розширення, тому не доцільно їх застосовувати в якості контактів для підключення зовнішніх периферійних пристроїв. При використанні контактів для подачі живлення потрібно враховувати той факт, що номінал максимальної сили струму на контактів дорівнює 50 мА.

Більш детальна схема інтерфейсу GPIO наведена на рис. 2.5.

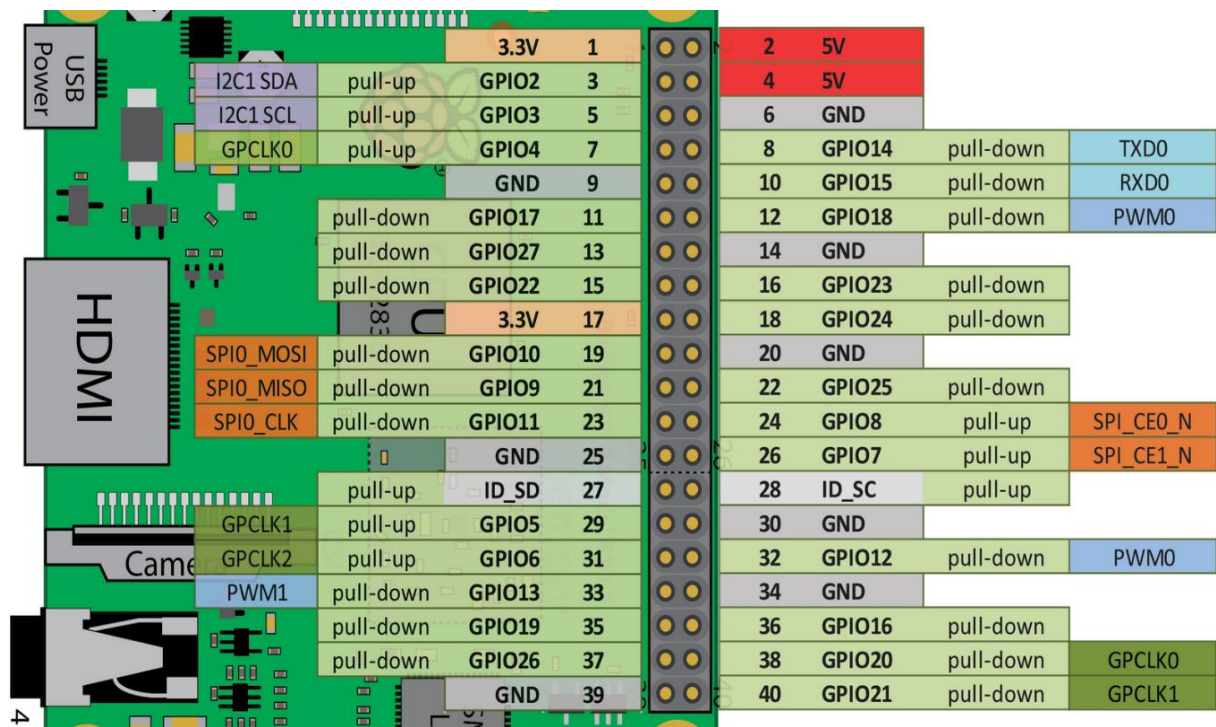


Рисунок 2.5 – Детальна схема GPIO Raspberry PI

Для забезпечення функцій керування виводами використовується програмне забезпечення Raspberry PI, яке може бути реалізоване за допомогою таких мов програмування як C++, Python, C та ін.

При проектуванні систем, які використовують Raspberry PI і його цифровий інтерфейс GPIO потрібно враховувати організацію схеми електричної принципової відповідних виводів (рис. 2.6).

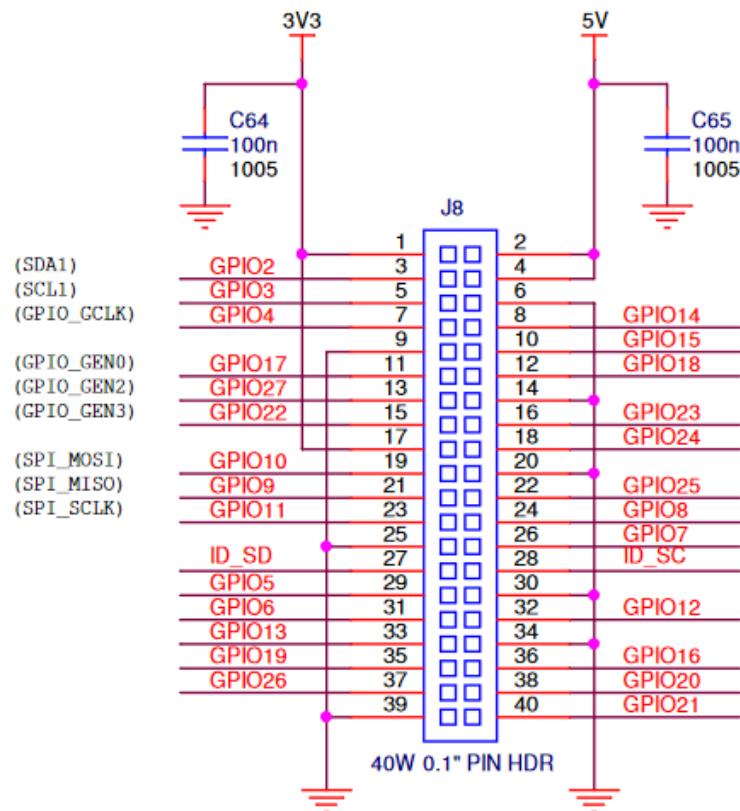


Рисунок 2.6 – Схема електрична принципова GPIO

Таким чином, Raspberry PI 3 цілком підходить в якості пристрою управління системи зчитування та візуалізації показників двигуна авто.

2.3 Дисплей для відображення даних показників CAN-шини

З метою відображення опрацьованих Raspberry PI даних, одержаних через CAN-шину запропоновано використати сенсорний дисплей, сумісний з даним мікроконтролером – 3,5”RPi LCD.

До основних технічних характеристик такого екрану відноситься:

- роздільна здатність на апаратному рівні 480*320.
- резистивне сенсорне управління;
- повна сумісність з усіма версіями Raspberry Pi.
- наявність драйверів для роботи з різними операційними системами

Raspbian, Ubuntu, Kali та ін;

- підтримка програмного драйвера FBSP, який може програмно задавати роздільну здатність та підтримувати подвійний екран;
- відповідність розмірам Raspberry Pi 4;
- зручність та практичність використання.

На рис. 2.7 показано зовнішній вигляд сенсорного дисплею, що використовується при реалізації проекту системи зчитування та візуалізації даних щодо показників двигуна авто.



Рисунок 2.7 – Екран 3,5”RPi LCD

Цифровий інтерфейс GPIO Raspberry Pi містить 40 виводів, а на РК-дисплеї наявні 26 контактів. Призначення і застосування виводів екрану 3,5”RPi LCD наведено у табл. 2.2.

Таблиця 2.2 – Призначення контактів сенсорного дисплею

Номер контакту	Ідентифікація	Опис
1	3.3 V	Живлення (вхід 3,3 В)
2	5 V	Живлення (вхід 5 В)
3	NC	NC
4	5 V	Живлення (вхід 5 В)
5	NC	NC

Номер контакту	Ідентифікація	Опис
6	GND	Заземлення
7	NC	NC
8	NC	NC
9	GND	Заземлення
10	NC	NC
11	TP_IRQ	Переривання у випадку натиску на сенсорну панель
12	NC	NC
13	NC	NC
14	GND	Заземлення
15	NC	NC
16	NC	NC
17	3.3V	Живлення (3,3В вхід)
18	LCD_RS	Команда/Вибір регістру даних
19	LCD_SI / TP_SI	LCD екран / вхідні дані SPI дані сенсорної панелі
20	GND	Заземлення
21	TP_SO	Вихідні дані SPI сенсорної панелі
22	RST	Перезавантаження
23	LCD_SCK / TP_SCK	Сигнал годинника SPI для екрану LCD/сенсорної панелі
24	LCD_CS	Вибір сигналу LCD, низький рівень обирає LCD
25	GND	Заземлення
26	TP_CS	Сигнал вибору сенсорної панелі, низький рівень обирає сенсорну панель

3,5” RPi LCD дисплеєм можна керувати двома способами:

- встановлення драйвера для ОС Raspbian.
- використання готового образу для якого попередньо встановлено драйвер LCD.

Інсталяція операційної системи на Raspberry Pi та встановлення драйвера для управління сенсорним екраном більш детально описано у наступному розділі.

Сенсорний екран 3,5”RPi LCD комунікує з Raspberry Pi використовуючи шину SPI шляхом підключення відповідних виводів GPIO. Він також не потребує зовнішнього джерела живлення, оскільки отримує його від мікроконтролера.

2.4 Проектування схеми живлення для Raspberry Pi

У випадку використання Raspberry Pi 3 необхідно враховувати такі аспекти як забезпечення низької напруги живлення певного номіналу. У транспортних засобах 2000 років відсутні будь-які типи портів USB, які можна було використати для живлення мікроконтролера. Однак у будь-якому авто наявний акумулятор, який видає 12 В постійної напруги. Це означає, що для забезпечення живленням RPi необхідно передбачити перетворювач постійного струму, який здатний був би перетворити 12 В від акумулятора на 5 В для мікроконтролера.

Проте, виникає ще одна проблема, яка полягає в тому, що 12В від акумулятора завжди подається, незалежно від того чи вмикається, чи вимикається двигун автомобіля.

У свою чергу, якщо автомобіль не заведений, Raspberry Pi споживає дуже невелику кількість енергії з акумулятора автомобіля (~ 200 мА в режимі очікування). Однак, якщо систему зчитування та візуалізації показників двигуна залишити увімкнутою більше, ніж на тиждень, то це призведе до повного розрядження акумулятора.

Для вирішення цієї проблеми було запропоновано використовувати лінію АСС. Це дає змогу визначити стан, коли транспортний засіб насправді працює,

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

що пов'язано зі специфікою замикання контактів в замку запалювання. Зазвичай, для будь-якого авто визначено чотири стани замка запалювання:

- ключ вставлений у замок – живлення не подається на вузли авто;
- ключ повернутий на одне положення – замикання лінії АСС, що супроводжується подачею живлення на окремі вузли транспортного засобу;
- ключ повернутий до положення подачі живлення на усі пристрої;
- стан запуску автомобіля.

Використання лінії АСС означає також, що підключений мікроконтролер миттєво втратить подачу живлення, як тільки автомобіль заглушити і витягнути ключ із замка запалювання. А це потенційно може призвести до пошкодження SD -карти на мікроконтролері.

Для того, щоб використовувати лінію АСС по якій передається 12 В при увімкненому двигуні, а також при вимиканні живленні цієї лінії і забезпечення коректного завершення роботи Raspberry PI потрібно додатково встановити невеликий мікроконтролер, який би виконував таку функцію протягом 45 с.

В якості такого мікроконтролера запропоновано використати 8-бітний мікроконтролер Attiny85. Схема управління живленням Raspberry PI на основі даного мікроконтролера показана на рис. 2.8.

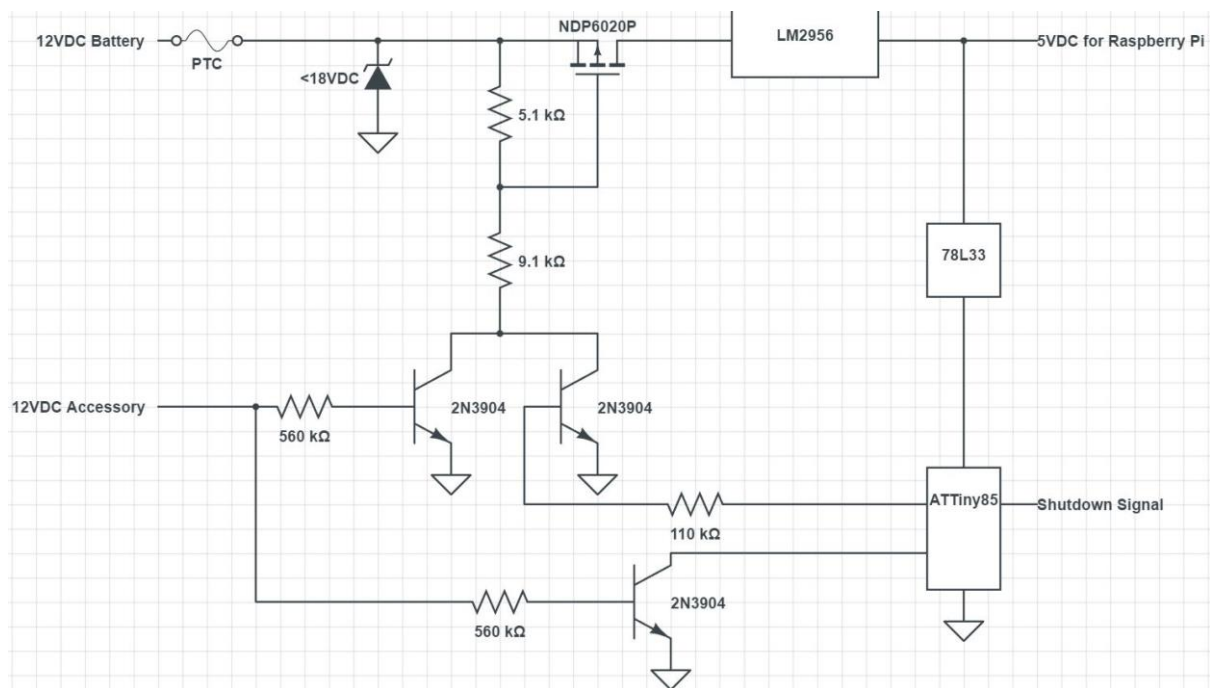


Рисунок 2.8 – Схема електрична принципова для живлення Raspberry PI

Після цього, схему було імплементовано на друкованій платі, яку можна вмонтувати поряд з Raspberry PI (рис. 2.9).

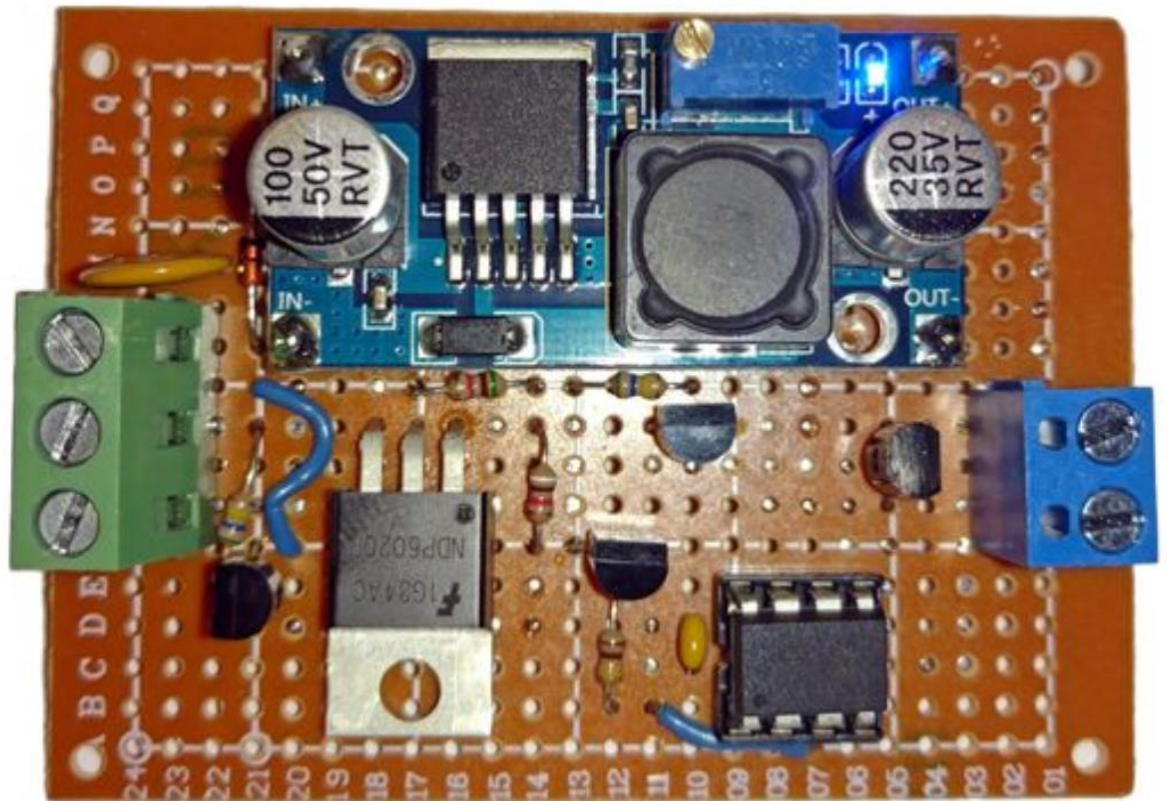


Рисунок 2.9 – Схема для живлення Raspberry PI 3 та мікроконтролера

Розроблена плата має три входи:

- заземлення;
- АСС;
- 12В.

Використовуючи ці входи, плата забезпечує живленням Raspberry PI при активації лінії АСС. Після вимкнення двигуна подача живлення триватиме ще протягом 45 секунд. На платі використовується також LM2596 в якості перетворювача постійного струму з номіналом напруги 12В у постійний струм з номіналом напруги 5 В.

Окрім перетворювача струму у схемі використано регулятор лінійної напруги 78L33, що використовується для пониження напруги від 5 В до 3,3 В для мікроконтролера Attiny85. Коли напруга живлення лінії АСС знижується,

плата надсилає високий рівень сигналу через вихідну лінію, підключену до контакту GPIO на Raspberry Pi.

Як тільки Raspberry Pi бачить, що на відповідному контакті високий рівень сигналу, то одразу переходить в режим повного відключення `sudo -h`. Цей процес займає лише 10 секунд, або близько того часу, а це означає, що мікроконтролер керування системи зчитування та візуалізації даних показників з CAN-шини буде повністю вимкненим, перш ніж до нього перестане подаватися живлення.

2.4.1 Характеристики Attiny85

Застосування восьми бітного AVR мікроконтролера ATTINY85 пов'язано з необхідністю забезпечувати аналіз наявності живлення в АСС лінії для безпечного вимкнення Raspberry Pi при вимкненні двигуна автомобіля. Для проекту використано такий мікроконтролер в корпусі DIP-8. На рис. 2.10 показано зовнішній вигляд ATTINY85.

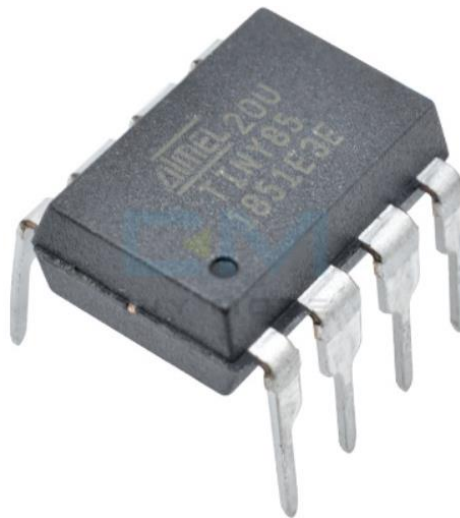


Рисунок 2.10 – Зовнішній вигляд ATTINY85

Робоча напруга мікроконтролера, представленого на рис. 2.10 коливається в межах від 2.7В до 5.5В, об'єм флеш пам'яті становить 8КБ, оперативної пам'яті – 512Б та енергонезалежної – 512 Б.

На рис. 2.11 представлено розпіновку мікроконтролера ATTINY85, а на рис. 2.12 – його структурну схему.

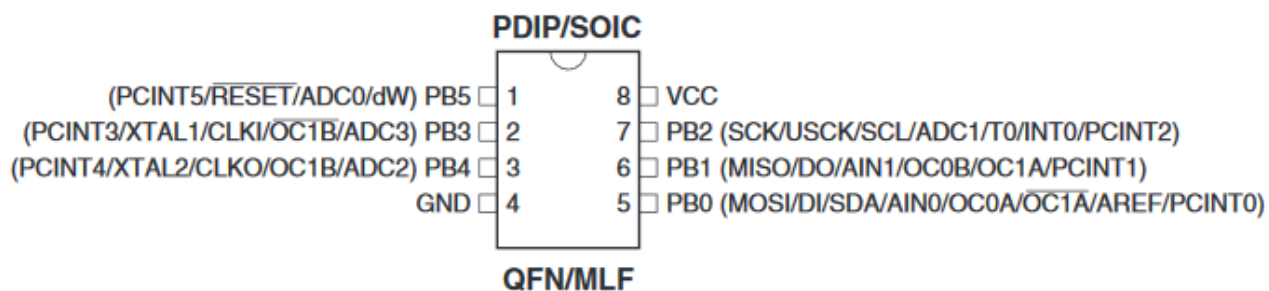


Рисунок 2.11 – Розпіновка мікроконтролера ATTINY85

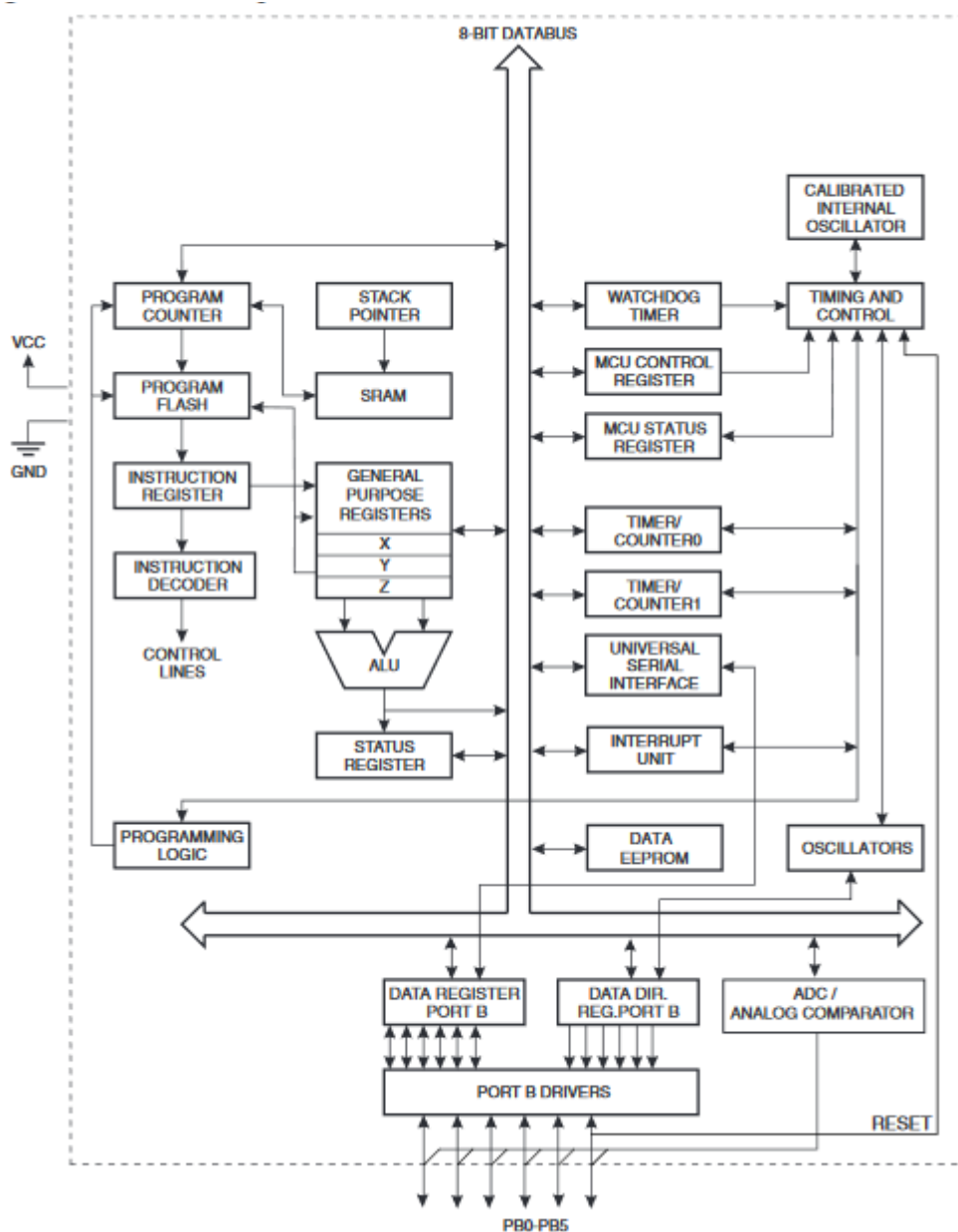


Рисунок 2.12 – Структурна схема ATTINY85

Функціональність ядра AVR включає в себе множину інструкцій, яка становить тридцять два робочі регістри загального призначення. Кожен з

регістрів має безпосередній зв'язок з АЛП, що дає змогу одержувати доступ до 2-ох незалежних між собою регістрів однієї інструкції. Така інструкція виконує опрацювання даних за один такт.

2.4.2 Перетворювач струму «постійний-постійний» LM2596

Конвертер LM2596 – це монолітна інтегральна схема, яка ідеально підходить для простого та зручного проектування понижувального імпульсного регулятора (понижувального перетворювача). Він здатний керувати навантаженням 3,0 А з регулюванням лінії та навантаження.

Цей пристрій доступний у версії з регульованим вихідним сигналом і має внутрішню компенсацію, щоб мінімізувати кількість зовнішніх компонентів для спрощення конструкції джерела живлення. Оскільки перетворювач LM2596 є імпульсним джерелом живлення, його ефективність значно вища порівняно з популярними лінійними стабілізаторами з трьома полюсами, особливо з вищими вхідними напругами.

LM2596 працює на частоті перемикання 150 кГц, що дозволяє використовувати компоненти фільтра меншого розміру, ніж те, що було б необхідно для регуляторів перемикання з нижчою частотою. Доступний у стандартному 5-вивідному корпусі TO-220 з декількома різними варіантами вигину виводу та пакеті D2 PAK для поверхневого монтажу. На рис. 2.13 показано вигляд LM2596.

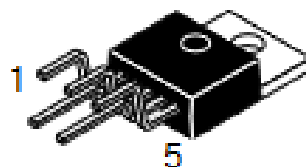


Рисунок 2.13 – Конвертер «постійний-постійний» LM2596

Інші функції LM2596 включають гарантований допуск на 4% вихідної напруги в межах заданих вхідних напруг і умов вихідного навантаження, а також 15% на частоту генератора.

					КС КРБ 123.311.00.00 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

Функції самозахисту включають циклічне обмеження струму для вихідного вимикача, а також теплове відключення для повного захисту в умовах несправності

2.4.3 Регулятор лінійної напруги 78L33

Серія регуляторів лінійної напруги L78L00 представляє собою екземпляри триполюсних позитивних регуляторів, які використовують внутрішнє обмеження струму та термічне відключення, що робить їх практично незнищувальними.

Якщо передбачено відповідний радіатор, вони можуть видавати вихідний струм до 100 мА. Регулятор 8L33 може використовуватися як стабілізатор фіксованої напруги напруги в широкому діапазоні застосувань, включаючи локальне або вбудоване регулювання для усунення шуму та проблем розподілу, пов'язаних з одноточковим регулюванням.

Крім того, його можна використовувати з елементами пропуску потужності для виготовлення сильноточових стабілізаторів напруги. Серія L78L00, яка використовується як заміна комбінації стабілітрон/резистор, забезпечує ефективне покращення вихідного опору, зазвичай, на два порядки разом із меншим струмом спокою та меншим шумом. На рис. 2.14 показано схему електричну принципову регуляторі 78L33.

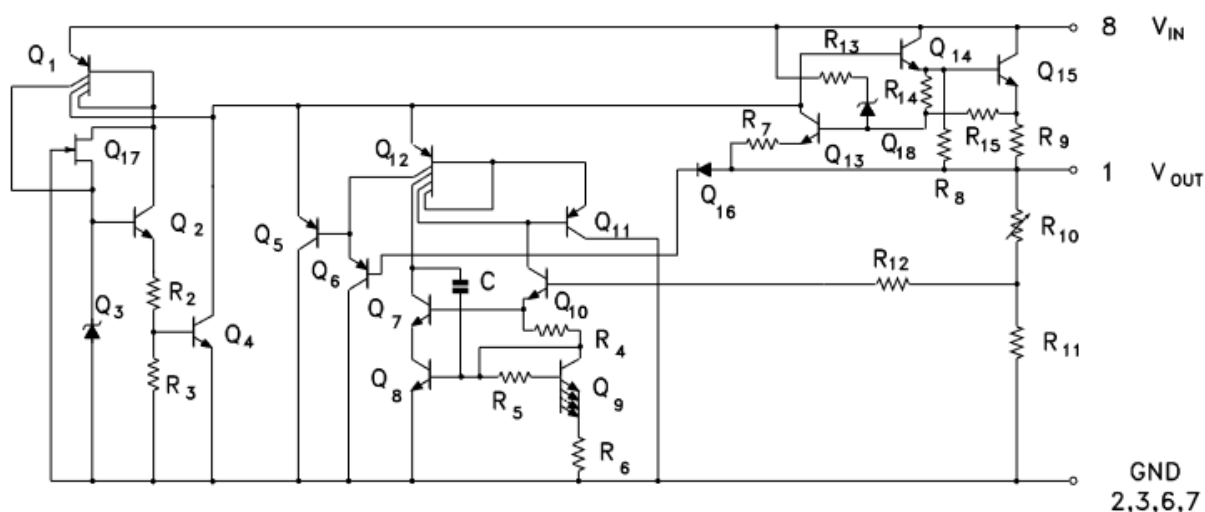


Рисунок 2.14 – Схема електрична принципова 78L33

В залежності від типу виконання та корпусу 78L33, виводи мікросхеми можуть використовуватися як показано на рис. 2.15.

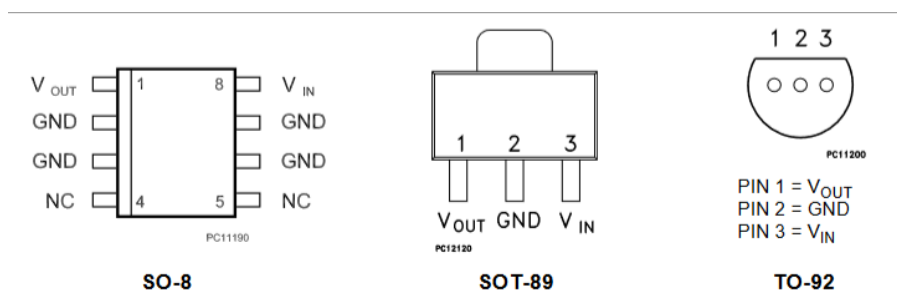


Рисунок 2.15 – Виводи 78L33 у різних корпусах

Основними характеристиками, які задовольняють вирішення задачі для безпечного вимкнення Raspberry PI з точки зору застосування 78L33 є:

- вихідний струм до 100 мА;
- вихідна напруга 3,3В та 5 В, а також в діапазоні від 6В до 24В;
- захист від термічних перевантажень;
- захист від короткого замикання;
- відсутність зовнішніх компонентів;
- доступні напруги $\pm 5\%$ (АС) АБО $\pm 10\%$ (С).

Після проведення проектування системи зчитування та візуалізації даних з CAN шини потрібно встановити апаратні компоненти у транспортному засобі.

Порт OBD розташований внизу панелі з боку водія, тому доцільно використати прямий роз'єм OBD, щоб підключити порт через кабель USB, який прокидається через нижню панель до Raspberry PI на центральній консолі.

Після цього потрібно прокласти лінію живлення від акумулятора 12 В і АСС, що входять у стерео та підключити 40 -контактний кабель для під'єднання сенсорного екрану до виводів GPIO на Raspberry Pi. Екран насправді не використовує всі 40 контактів GPIO (лише 5 виводів SPI, а також GPIO 24 та 25), тому він забезпечує 26 комунікацію з GPIO на звороті екрана. Далі приєднуються два провідники перемички з контакту сигналу Power Board та сигнального заземлення на два контакти GPIO на екрані.

РОЗДІЛ 3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ЗЧИТУВАННЯ ТА ВІДОБРАЖЕННЯ ПОКАЗНИКІВ ДВИГУНА АВТО

3.1 Встановлення операційної системи Raspbian Lite

Перед тим, як реалізувати логіку зчитування та візуалізації даних про показники двигуна необхідно інсталиувати операційну систему на Raspberry Pi та виконати її налаштування.

Для забезпечення продуктивності роботи центрального мікроконтролера та функцій проектованої системи запропоновано встановити операційну Raspbian Lite. Перший крок при встановленні операційної системи полягає у завантаженні образу з офіційного сайту у вигляді zip-архіву та добуванні файлу `img.xz`.

Після цього необхідно записати файл-образ на SD-картку. У випадку використання операційної системи Linux для виконання цієї процедури можна скористатися утилітою `gnome-disk-utility`. На рис. 3.1 показано вікно після запуску утиліту, що пропонує відформатувати SD-карту.

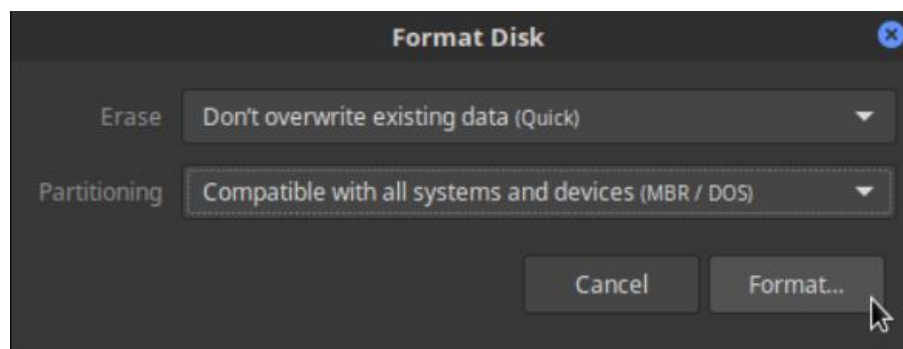


Рисунок 3.1 – Форматування SD-карти за допомогою `gnome-disk-utility`

Після натиснення кнопки «Format», з'явиться вікно (рис. 3.2) для того, щоб підтвердити вибір щодо форматування.

					<i>КС КРБ 123.311.00.00 ПЗ</i>		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Гарнік В.Р.			Програмне забезпечення комп'ютерної системи зчитування та відображення показників двигуна авто	Літ.	Арк.
Перевірив.		Луцків А.М.					37
Консульт.						ТНТУ, каф. КС, гр. СІс-42	
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					

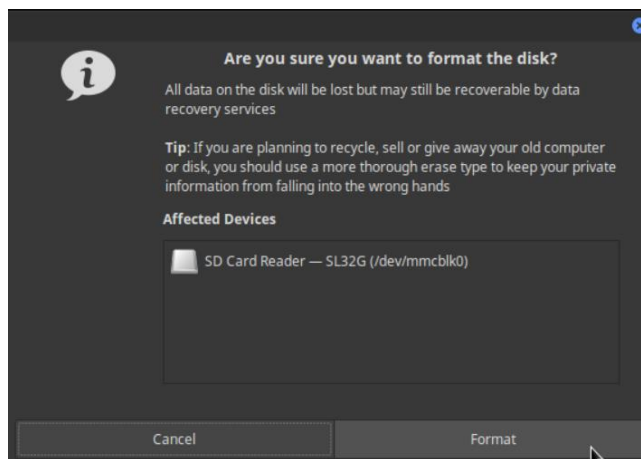


Рисунок 3.2 – Інформаційне вікно для підтвердження форматування карти

Наступний крок полягає у виборі власне простору, який підлягає форматуванню, як показано на рис. 3.3.

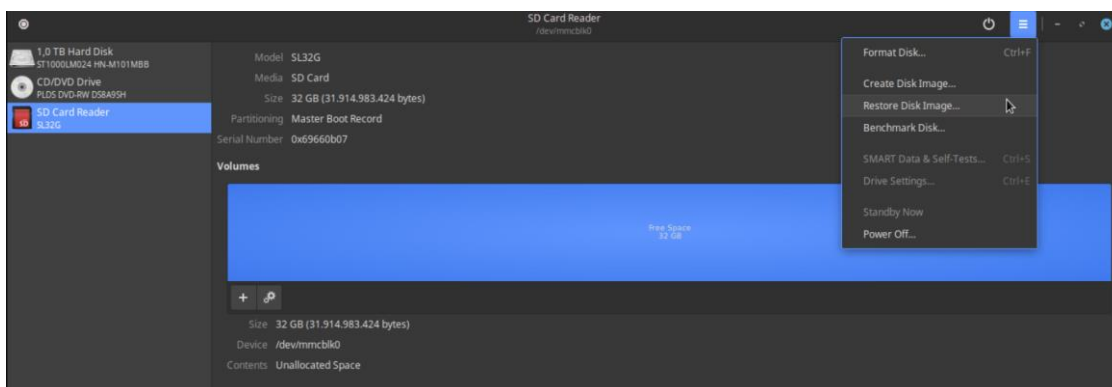


Рисунок 3.3 – Вибір диску для форматування

Обравши SD-картку, необхідно задати розмір диску, який необхідно відформатувати та обрати образ для встановлення операційної системи. Така процедура відбувається у два кроки, як показано на рис.3.4 та рис. 3.5.

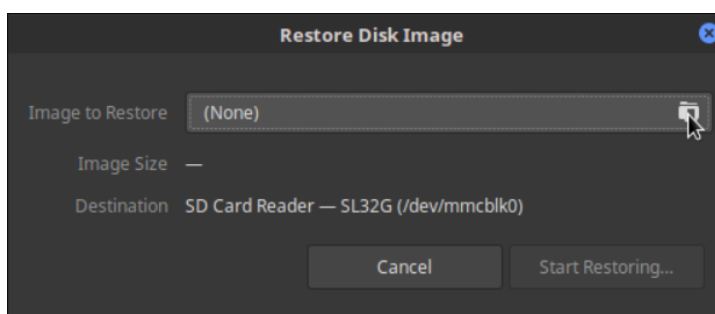


Рисунок 3.4 – Вікно вибору образу операційної системи

Безпосередній вибір образу Raspbian Lite для встановлення на Raspberry PI показано на рис. 3.5.

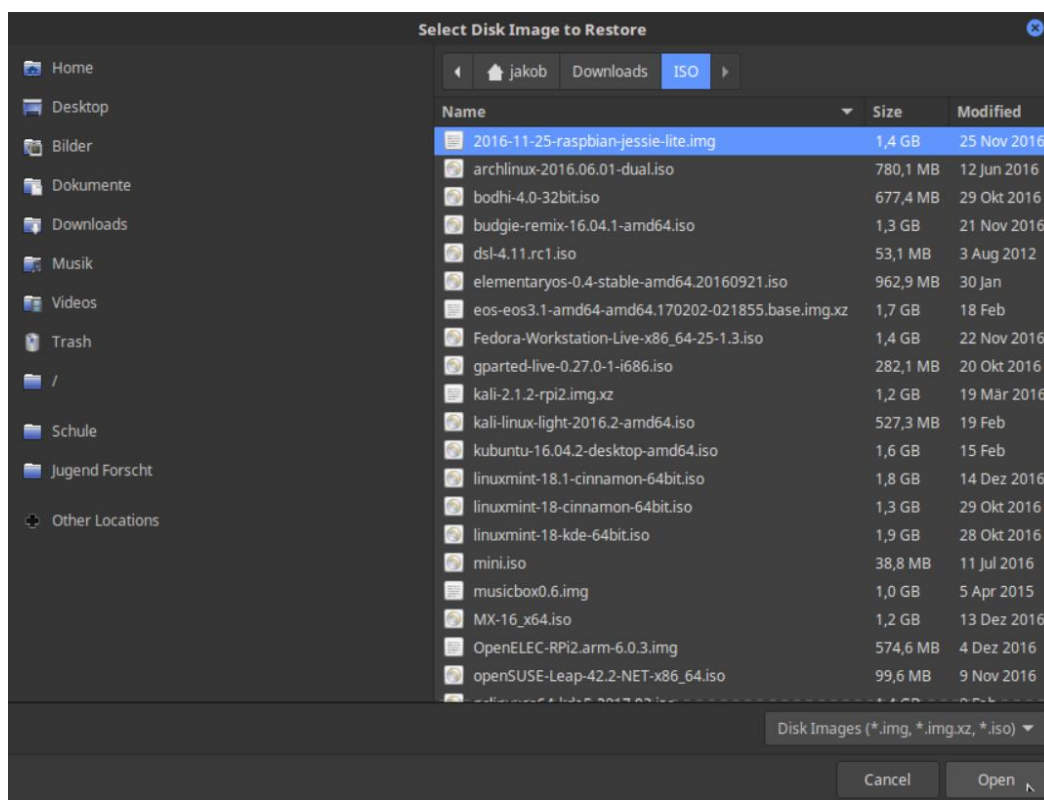


Рисунок 3.5 – Вибір образу Raspbian Lite

Після того, як зроблено вибір образу, з'являється вікно (рис. 3.6) в якому необхідно натиснути кнопку «Start Restoring».

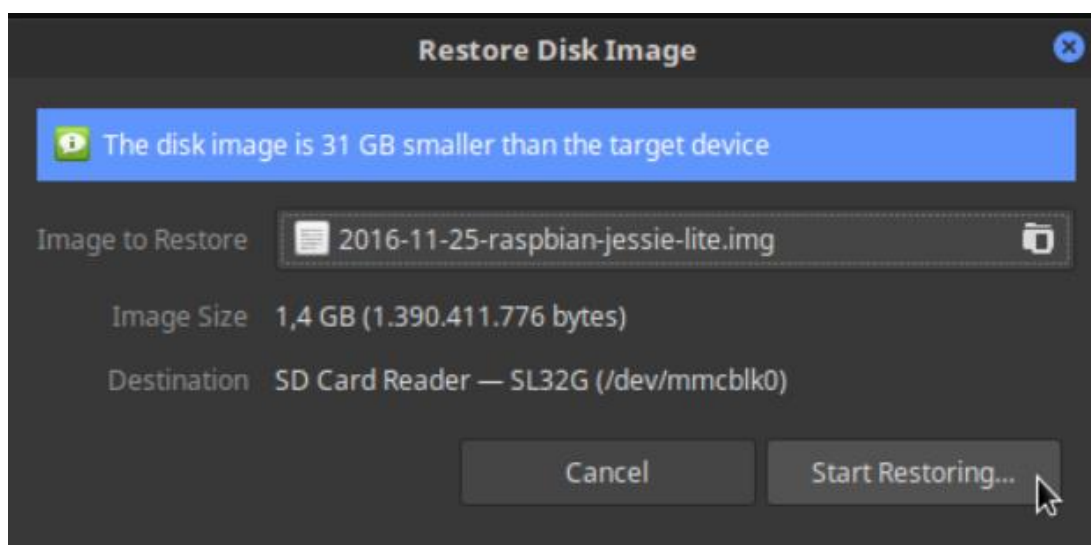


Рисунок 3.6 – Початок процесу інсталяції ОС Raspbain Lite

3.2 Налаштування SSH та доступу до WiFi

Після успішного встановлення операційної системи потрібно забезпечити можливість доступу до Raspberry PI через SSH та можливість доступу до мережі інтернет з використання безпроводної передачі даних.

У випадку використання Linux, розділи будуть відкриті після успішного запису. Далі потрібно відкрити новий термінал і перейти на обліковий запис root за допомогою `sudo su` (рис. 3.7).

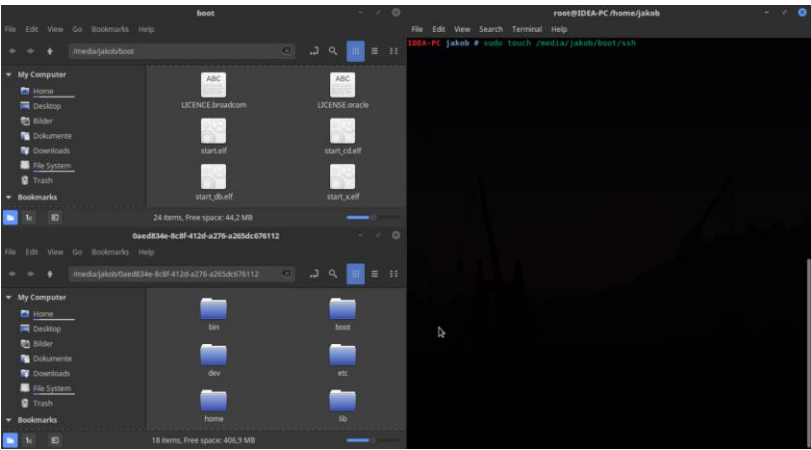


Рисунок 3.7 – Перехід в обліковий запис root при налаштуванні SSH

Наступний крок полягає у створенні файлу з назвою `ssh` у завантажувальному розділі, тобто потрібно обрати `/path/to/boot/partition/ssh`, як показано на рис. 3.8.

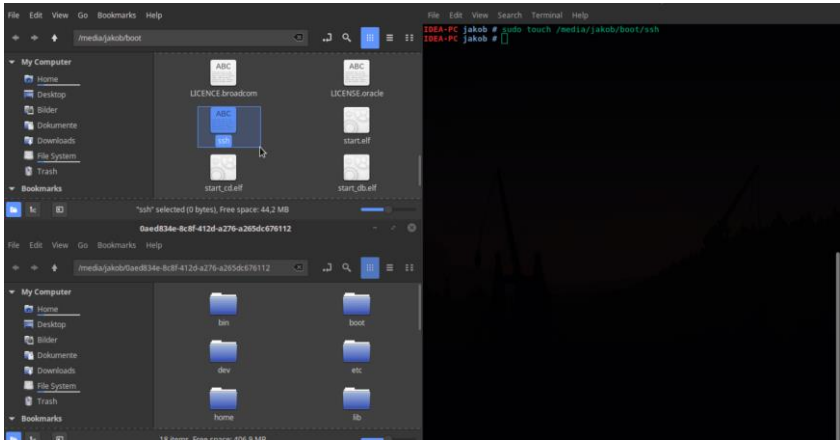


Рисунок 3.8 – Вибір SSH

Таким чином налаштовано можливість доступу до Raspberry PI за допомогою SSH. Налаштування параметрів WiFi відбувається наступним чином.

Спочатку треба відкрити файл `wpa_supplicant.conf` і ввести налаштування мережі – `«nano /path/to/root/partition/etc/wpa_supplicant/wpa_supplicant.conf
network={ ssid="ВАШ WIFI SSID ТУТ" psk="ВАШ ПАРОЛЬ WIFI ТУТ" }»`
(рис. 3.9).

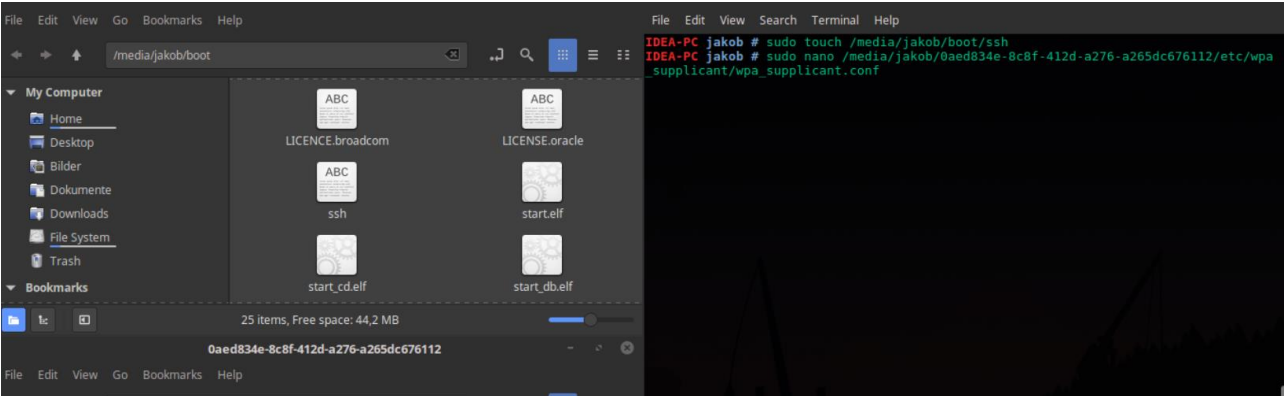


Рисунок 3.9 – Налаштування конфігурації WiFi

Вікно, де показано налаштування паролю до WiFi показано на рис. 3.10.

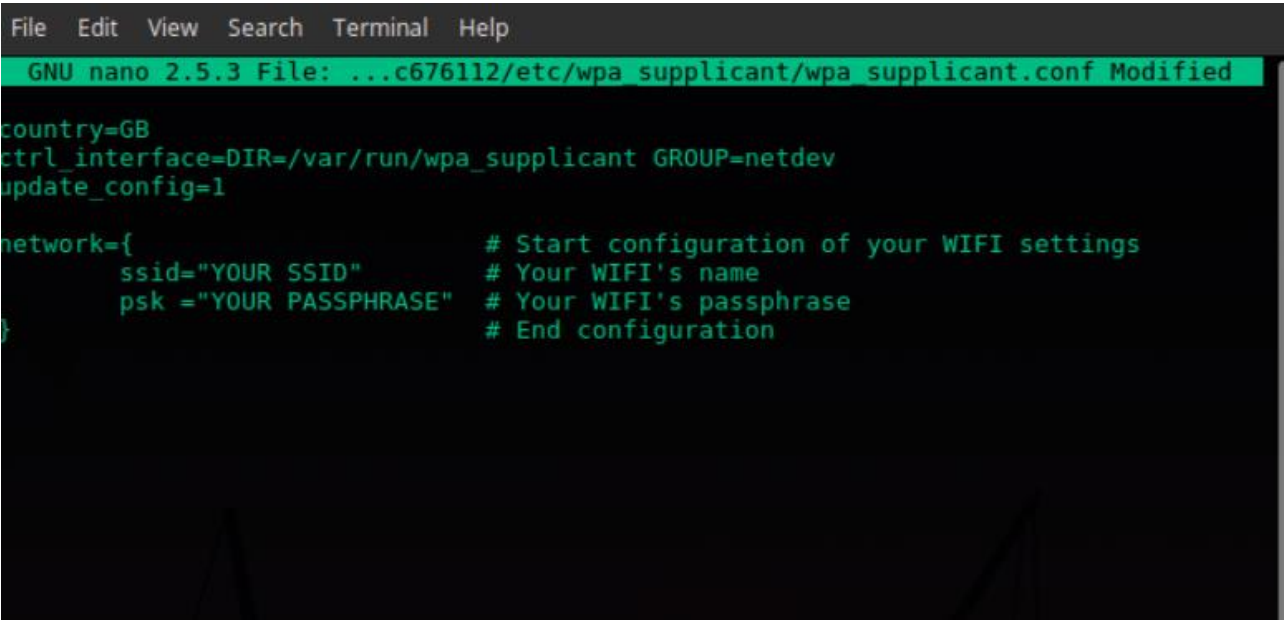


Рисунок 3.10 – Налаштування параметрів доступу до WiFi

Після проведення усіх налаштувань виконуються команди для їх збереження («CTRL+O, RETURN, CTRL+X»)

Далі потрібно відкрити файли інтерфейси і ввести метод DNS nano /path/to/root/partition/etc/network/interfaces (рис. 3.11) та змінити усі ручні методи на DHCP (рис. 3.12).

```

root@IDEA-PC /home/jakob
File Edit View Search Terminal Help
IDEA-PC jakob # sudo touch /media/jakob/boot/ssh
IDEA-PC jakob # sudo nano /media/jakob/0aed834e-8c8f-412d-a276-a265dc676112/etc/wpa
supPLICANT/wpa_supPLICANT.conf
IDEA-PC jakob # sudo nano /media/jakob/0aed834e-8c8f-412d-a276-a265dc676112/etc/net
work/interfaces
interfaces          interfaces.d/          interfaces.dpkg-old
IDEA-PC jakob # sudo nano /media/jakob/0aed834e-8c8f-412d-a276-a265dc676112/etc/net
work/interfaces
  
```

Рисунок 3.11 – Відкриття файлів інтерфейсів

```

File Edit View Search Terminal Help
GNU nano 2.5.3 File: ...-412d-a276-a265dc676112/etc/network/interfaces Modified

# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpd
# For static IP, consult /etc/dhcpd.conf and 'man dhcpd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet dhcp
    wpa-conf /etc/wpa_supPLICANT/wpa_supPLICANT.conf

allow-hotplug wlan1
iface wlan1 inet dhcp
    wpa-conf /etc/wpa_supPLICANT/wpa_supPLICANT.conf
  
```

Рисунок 3.12 – Налаштування DHCP

Далі потрібно спочатку на програмному рівні безпечно від'єднати SD-карту шляхом виконання команди: «umount -fl /path/to/root/partition /path/to/boot/partition», а після того витягнути SD-картку з відповідного слота.

Коли Raspberry PI завантажиться в терміналі потрібно ввести команду: «sudo raspi-config», щоб запустити інструмент налаштування програмного забезпечення (рис. 3.13).

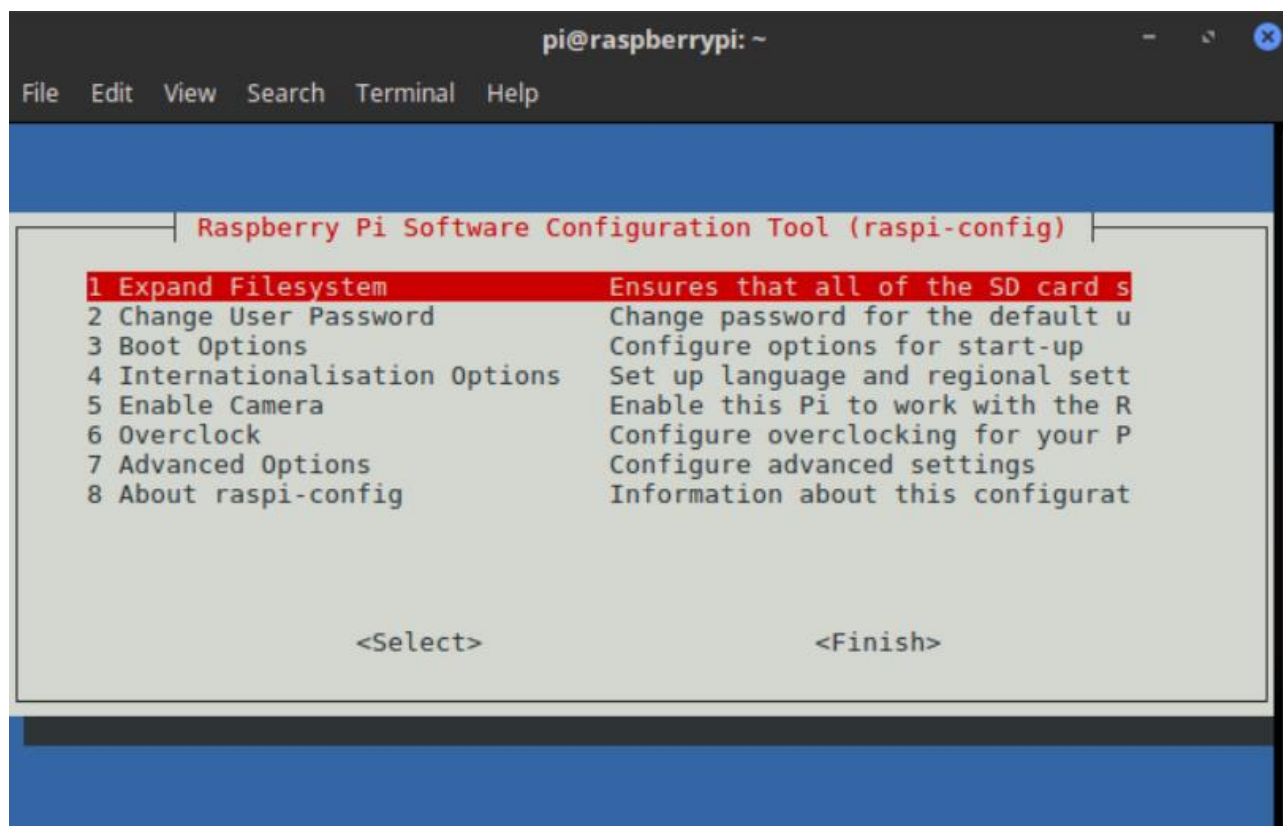


Рисунок 3.13 – Налаштування програмного забезпечення Raspberry PI

У вікні, як показано на рис. 3.13, потрібно обрати «Розгорнути файлову систему», щоб розширити кореневий розділ. За замовчуванням встановлено більше 25 ГБ вільного місця, яке надається для Raspberry Pi під час розширення кореневого розділу. Після завершення цієї процедури необхідно повернутися в меню «Пуск».

Для налаштування параметрів локалізації потрібно обрати відповідний пункт меню, який дозволяє встановити локалі, клавіатуру, часовий пояс і країну

Після того, як встановлено усі важливі налаштування, потрібно перейти за допомогою клавіш зі стрілками до кнопки Finish і натиснути Return. Фінальний кроком є перезавантаження Raspberry PI.

3.3 Налаштування програмного підключення сенсорного екрану до Raspberry PI

Для того, щоб повноцінно функціонував сенсорний дисплей 3,5”RPi LCD необхідно встановити драйвер в операційну систему Raspbian Lite. Це можна зробити двома способами:

- встановлення драйвера на вже існуючу операційну систему;
- завантаження і встановлення образу операційної з уже вбудованим драйвером для роботи з LCD-екраном;

Оскільки у попередньому пункті вже виконано інсталяцію операційної системи, то потрібно використовувати перший спосіб. Алгоритм інсталяції драйвера дисплею передбачає виконання наступної команди (рис. 3.14).

```
git clone https://github.com/waveshare/LCD-show.git  
cd LCD-show/
```

Рисунок 3.14 – Команда встановлення драйвера для сенсорного екрану

Важливо, щоб Raspberry PI мав доступ до мережі з можливістю підключення до Інтернет, оскільки в іншому випадку драйвер не буде інстальований. Після цього виконується ще одна команда (рис. 3.15), яка дозволить працювати сенсорному екрану.

```
chmod +x LCD35-show  
./LCD35-show
```

Рисунок 3.15 – Налаштування сенсора дисплею

Функції сенсорів екрану працюватимуть після перезапуску мікроконтролера. Для зручності використання можна встановити орієнтацію екрана.

Потрібно відміти те, що виконання оновлення «apt-get» призведе до того, що РК-дисплей не працюватиме належним чином. У цьому випадку потрібно відредагувати файл config.txt на SD-карті та видалити рядок: «dtoverlay=ads7846».

У випадку використання ОС Raspbian Lite, команда встановлення сенсорного драйвера має відповідати команді, яка наведена на рис. 3.16.

```
./LCD35-show lite
```

Рисунок 3.16 – Інсталяція драйвера для ОС Raspbian Lite

Сенсорний дисплей можна відкалібрувати за допомогою спеціалізованої програми xinput-calibrator за умови підключення Raspberry Pi до мережі. Для встановлення програми використовується код, показаний на рис. 3.17.

```
sudo apt-get install xinput-calibrator
```

Рисунок 3.17 – Інсталяція програми калібрування сенсорного дисплею

Після успішної інсталяції проводиться калібрування екрану шляхом вибору пункту «Меню» на панелі задач, «Налаштування» -> «Калібрувати сенсорний екран». Завершення калібрування відбувається згідно до підказок майстра налаштувань.

Зазвичай, для активації відкаліброваного сенсорного дисплею потрібно перезавантажити Raspberry Pi. Окрім цього, можна створити файл 99-calibration.conf для збереження параметрів дотику (не обов'язково, якщо файл існує). Для цього використовується команда показана на рис. 3.18.

```
sudo nano /etc/X11/xorg.conf.d/99-calibration.conf
```

Рисунок 3.18 – Відкриття файлу для збереження параметрів калібрування

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

Результат запису даних щодо параметрів калібрування наведено на рис. 3.19.

```
Section "InputClass"
    Identifier      "calibration"
    MatchProduct    "ADS7846 Touchscreen"
    Option "Calibration" "208 3905 288 3910"
    Option "SwapAxes" "0"
EndSection
```

Рисунок 3.19 – Збережені параметри калібрування сенсорного екрану

3.4 Встановлення і налаштування віртуальної клавіатури для сенсорного екрану

Для того, щоб інсталиувати віртуальну клавіатуру на сенсорний екран на першому кроці потрібно виконати команди, які показані на рис. 3.20.

```
sudo apt-get install update
sudo apt-get install matchbox-keyboard
sudo nano /usr/bin/toggle-matchbox-keyboard.sh
```

Рисунок 3.20 – Перший крок встановлення віртуальної клавіатури

Далі потрібно скопіювати наведені нижче рядки коду до toggle-matchbox-keyboard.sh і зберегти зміни (рис. 3.21).

```
#!/bin/bash
#This script toggle the virtual keyboard
PID=`pidof matchbox-keyboard`
if [ ! -e $PID ]; then
killall matchbox-keyboard
else
matchbox-keyboard -s 50 extended&
fi
```

Рисунок 3.21 – Модифікація файлу з віртуальною клавіатурою

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

Третій етап налаштування віртуальної клавіатури передбачає виконання команд, які проілюстровані на рис. 3.22.

```
sudo chmod +x /usr/bin/toggle-matchbox-keyboard.sh
sudo mkdir /usr/local/share/applications
sudo nano /usr/local/share/applications/toggle-matchbox-keyboard.desktop
```

Рисунок 3.22 – Третій етап налаштування віртуальної клавіатури

Наступним кроком передбачається повторна модифікація файлу конфігурації програмної клавіатури шляхом запису даних, як показано на рис. 3.23.

```
[Desktop Entry]
Name=Toggle Matchbox Keyboard
Comment=Toggle Matchbox Keyboard`
Exec=toggle-matchbox-keyboard.sh
Type=Application
Icon=matchbox-keyboard.png
Categories=Panel;Utility;MB
X-MB-INPUT-MECHANSIM=True
```

Рисунок 3.23 – Повторна зміна файлу конфігурації віртуальної клавіатури

Після модифікації файлу конфігурації сенсорного екрану потрібно виконати команди, які наведено на рис. 3.24. При цьому потрібно використовувати дозвіл користувача «Pi» замість root.

```
sudo nano /etc/xdg/lxpanel/LXDE-pi/panels/panel
```

Рисунок 3.24 – Команди для відкриття файлу з налаштування панелі сенсорного екрану

Далі потрібно знайти фрагмент опису конфігурації, що представлений на рис. 3.25 і додати програмний опис, наведений на рис. 3.26.

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

```

Plugin {
  type = launchbar
  Config {
    Button {
      id=lxde-screenlock.desktop
    }
    Button {
      id=lxde-logout.desktop
    }
  }
}

```

Рисунок 3.25 – Фрагмент опису конфігурації панелі сенсорного екрану для додавання кнопки керування

```

Button {
  id=/usr/local/share/applications/toggle-matchbox-keyboard.desktop
}

```

Рисунок 3.26 – Фрагмент для додавання екранної клавіатури

У результаті внесення змін, структура файлу для конфігурації екранної клавіатури матиме вигляд, як показано на рис. 3.27.

```

Plugin {
  type=space
  Config {
    Size=8
  }
}
Plugin {
  type=launchbar
  Config {
    Button {
      id=/usr/local/share/applications/toggle-matchbox-keyboard.desktop
    }
    Button {
      id=/usr/share/applications/lxde-x-www-browser.desktop
    }
    Button {
      id=/usr/share/raspi-ui-overrides/applications/pcmanfm.desktop
    }
    Button {
      id=/usr/share/raspi-ui-overrides/applications/lxterminal.desktop
    }
    Button {
      id=/usr/share/applications/wolfram-mathematica.desktop
    }
    Button {
      id=/usr/share/applications/wolfram-language.desktop
    }
  }
}
Plugin {
  type=space
  Config {
    Size=8
  }
}

```

Рисунок 3.27 – Оновлений файл конфігурації для відображення екранної клавіатури

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

Після перезавантаження Raspberry Pi та у випадку правильного налаштування віртуальної клавіатури ліворуч від панелі буде відображатися значок клавіатури.

3.5 Розробка програмного забезпечення зчитування та відображення показників двигуна

Забезпечення визначеної функціональності системи зчитування та відображення показників двигуна реалізується за допомогою скриптів написаних мовою програмування Python. Залежність між розробленими програмними модулями показана на рис. 3.28.

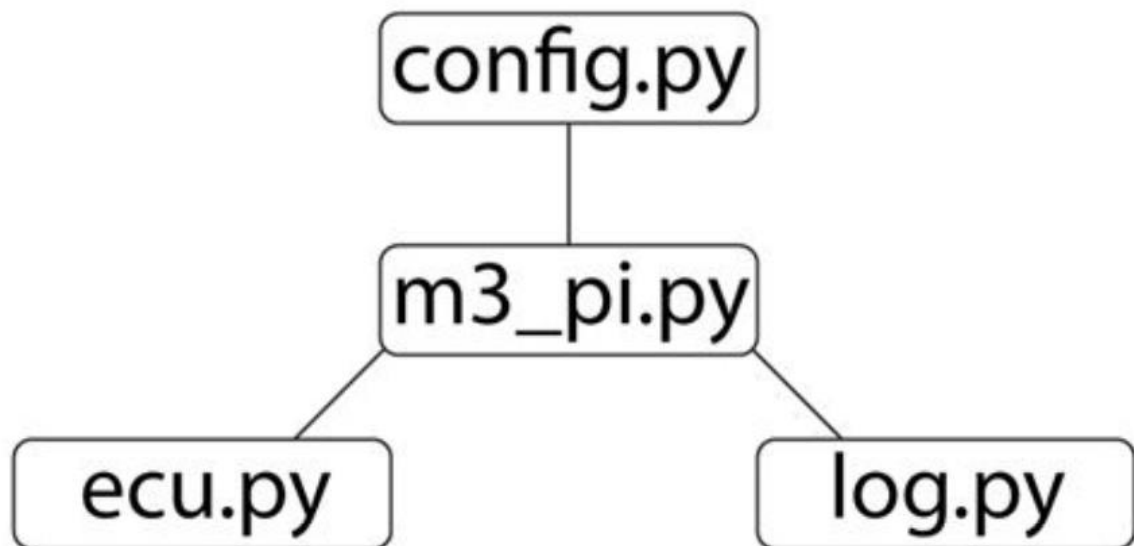


Рисунок 3.28 – Програмні модулі системи зчитування та відображення показників двигуна

Скрипт config.py містить усі глобальні змінні та налаштування користувачів, які будуть поділені між модулями.

Модуль M3_pi.py виконує функції основного контролера і забезпечує управління візуалізацією користувацького інтерфейсу, а також координує модулі esi.py та log.py.

Модуль ECU.py відповідає за управління з'єднанням з електронним блоком управління і забезпечує зберігання усіх даних, що містяться у змінних, модуля config.py для отримання доступу усіма іншими модулями.

Модуль log.py виконує опрацювання та реєстрацію всіх даних у файл CSV. На рис. 3.27 не зображено скрипт shutdown.py, оскільки він є надто примітивним і працює за дуже простим сценарієм. Він запускається окремо від m3_pi.py і призначений для безпечного вимкнення Raspberry PI.

Програмний код файлу конфігурації з визначеними глобальними змінними та масивом значень діапазонів обертів двигуна показано на рис. 3.29.

```
import ecu
import datetime
import numpy as np
# Globals.
logLength = 0
dtc_iter = 0
time_elapsed_since_last_action = 0
gui_test_time = 0
logIter = 1
ecuReady = False
debugFlag = False
settingsFlag = False
piTFT = True
startTime = datetime.datetime.today().strftime('%Y%m%d%H%M%S')
RESOLUTION = (480, 320)
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
# LUT representing the speeds at each of the five gears. Each entry is +200 RPM, and
# is directly linked to rpmlist.
speedArr = np.array([[4, 7, 10, 14, 17], [5, 9, 14, 18, 23], [7, 11, 17, 23, 28], [8,
14, 21, 28, 34], [9, 16, 24, 32, 40], [11, 18, 27, 37, 46], [12, 21, 31, 41, 51],
[14, 23, 34, 46, 57], [15, 25, 38, 50, 63], [16, 27, 41, 55, 68], [18, 30, 45, 60,
74], [19, 32, 48, 64, 80], [20, 34, 51, 69, 85], [22, 37, 55, 73, 91], [23, 39, 58,
78, 97], [24, 41, 62, 83, 102], [26, 43, 65, 87, 108], [27, 46, 69, 92, 114], [28,
48, 72, 96, 120], [30, 50, 75, 101, 125], [31, 53, 79, 106, 131], [33, 55, 82, 110,
137], [34, 57, 86, 115, 142], [35, 59, 89, 119, 148], [37, 62, 93, 124, 154], [38,
64, 96, 129, 159]])
# List of RPM values for above LUT.
19rpmlist = np.array([750, 1000, 1250, 1500, 1750, 2000, 2250, 2500, 2750, 3000, 3250,
3500, 3750, 4000, 4250, 4500, 4750, 5000, 5250, 5500, 5750, 6000, 6250, 6500, 6750,
7000])
```

Рисунок 3.29 – Програмний код файлу конфігурації

Програмний код модуля безпечного вимкнення Raspberry PI показано на рис. 3.30. У даному випадку для вимкнення мікроконтролера використовується стан сигнал на 17-му цифровому виводі GPIO.

```
import RPi.GPIO as GPIO
import os, time
# Set GPIO pin 17 as input for shutdown signal.
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.IN)
# Print message to console.
print("Running shutdown script...")
while True:
    if (GPIO.input(17)):
        os.system("sudo shutdown -h now")
        break
```

Рисунок 3.30 – Програмний код безпечного вимкнення Raspberry PI

Фрагмент програмного коду модуля log.py з описом програмних функцій для створення CSV-файлу логування показників двигуна показано на рис. 3.31.

```
import csv, os
from config import *
# Function to create a csv with the specified header.
def createLog(header):
    # Write the header of the csv file.
    with open('/home/pi/logs/' + startTime + '.csv', 'wb') as f:
        w = csv.writer(f)
        w.writerow(header)
    # Function to append to the current log file.
    def updateLog(data):
        with open('/home/pi/logs/' + startTime + '.csv', 'a') as f:
            w = csv.writer(f)
            w.writerow(data)
    # Function to close the log and rename it to include end time.
    def closeLog():
        endTime = datetime.datetime.today().strftime('%Y%m%d%H%M%S')
        os.rename('home/pi/logs/' + startTime + '.csv', 'logs/' + startTime + "_" +
            endTime + '.csv')
```

Рисунок 3.31 – Фрагмент модуля log.py

					КС КРБ 123.311.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

Фрагмент програмного опису класу і його конструктора для роботи з електронним блоком керування автомобіля, що входить у модуль ECU.py, показано на рис. 3.32.

```
class ecuThread(Thread):
    def __init__(self):
        Thread.__init__(self)
        self.daemon = True
        self.start()
    def run(self):
        global connection
        ports = obd.scan_serial()
        print ports
        # DEBUG: Set debug logging so we can see everything that is happening.
        obd.logger.setLevel(obd.logging.DEBUG)
```

Рисунок 3.32 – Фрагмент класу для роботи з ЕБУ автомобіля

Для забезпечення зручного користувацького інтерфейсу та відображення показників двигуна реалізовано модуль m3_pi.py, фрагмент якого наведено на рис. 3.32.

```
import config, ecu, log, time, datetime, sys
import pygame, time, os, csv
from pygame.locals import *

# Helper function to draw the given string at coordinate x,y, relative to center.
def drawText(string, x, y, font):
    if font == "readout":
        text = readoutFont.render(string, True, config.WHITE)
    elif font == "label":
        text = labelFont.render(string, True, config.WHITE)
    elif font == "fps":
        text = fpsFont.render(string, True, config.WHITE)
    textRect = text.get_rect()
    textRect.centerx = windowSurface.get_rect().centerx + x
    textRect.centery = windowSurface.get_rect().centery + y
    windowSurface.blit(text, textRect)

# Connect to the ECU and GPS.
if not config.debugFlag:
    ecu.ecuThread()

# Give time for the ECU to connect before we start the GUI.
while not config.ecuReady:
    time.sleep(.01)

# Load all of our tach images into an array so we can easily access them.
background_dir = 'tach/'
background_files = ['%i.png' % i for i in range(0, 42)]
ground = [pygame.image.load(os.path.join("/home/pi/tach/", file)) for file in background_files]
```

Рисунок 3.33 – Фрагмент програмного коду модуля користувацького інтерфейсу

					КС КРБ 123.311.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

На жаль, Python насправді не має жодної вбудованої функціональності для проектування графічного користувацького інтерфейсу, але є кілька бібліотек Python GUI, які можна використовувати для створення власного інтерактивного додатка. У роботі використано бібліотеку Pygame16, що представляє собою набір Python-модулів, які в основному призначені для створення відеоігор.

Базовий графічний інтерфейс користувача показано на рис. 3.34.

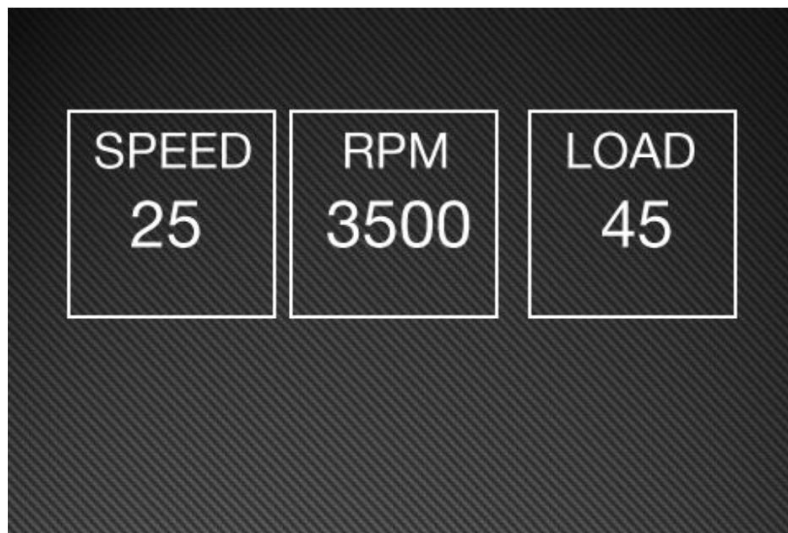


Рисунок 3.34 – Базова концепція графічного інтерфейсу

Базовий інтерфейс, що показаний на рис. 3.33 використовується для налагодження програмного забезпечення при зчитуванні даних з CAN-шини.

Далі за допомогою фізичного адаптера USB OBD-II та бібліотеки з відкритим кодом Python-obd19 створено тестовий модуль для отримання даних про швидкість, об/хв та навантаження двигуна з автомобіля в реальному часі.

Для зручності налагодження тестового модуля використано емулятор Freematics OBD-II MK220. Після налаштування емулятора, він почав підтримувати протокол ISO 9141-2, як цільовий автомобіль.

За допомогою емулятора одержано тестові дані з блоку управління авто для відображення показників на основному екрані. Проте, графічний інтерфейс надзвичайно повільно оновлюється. Після налагодження встановлено, що запити до електронного блоку керування авто були синхронними, і, таким чином,

блокували інтерфейс користувача. Це означало, що кожного разу, коли було здійснено запит, програма блокує очікування даних, а не оновлює графічний інтерфейс, поки запит не закінчиться. Для виправлення цього недоліку використано асинхронну функцію підключення Python-OBD, що дозволило оновлювати графічний інтерфейс в режимі реального часу.

При розробці графічного інтерфейсу, який виводився на сенсорний екран, обрано найбільш корисні для водія показники:

- швидкість руху авто;
- оберти двигуна (тахометр);
- навантаження на двигун;
- положення дросельної заслінки;
- температура теплоносія.

На рис. 3.35 показано графічний інтерфейс системи зчитування та відображення показників авто, одержаних через CAN-шину.



Рисунок 3.35 – Графічний інтерфейс системи зчитування та відображення показників двигуна авто

Таким чином, реалізовано комп'ютерну систему, яка забезпечує зчитування даних з CAN-шини транспортного засобу та виводить їх значення на сенсорний екран.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Організація служби охорони праці на підприємстві

Роботодавець зобов'язаний згідно Закону України «Про охорону праці» стаття 13 «Управління охороною праці та обов'язки роботодавця» створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці.

Із цією метою роботодавець забезпечує функціонування системи управління охороною праці, а саме:

- створює відповідні служби і призначає посадових осіб, які забезпечують вирішення конкретних питань охорони праці, затверджує інструкції про їхні обов'язки, права та відповідальність за виконання покладених на них функцій, а також контролює їх додержання;
- розробляє за участю сторін колективного договору і реалізує комплексні заходи для досягнення встановлених нормативів та підвищення існуючого рівня охорони праці;
- забезпечує виконання необхідних профілактичних заходів відповідно до обставин, що змінюються;
- впроваджує прогресивні технології, досягнення науки і техніки, засоби механізації та автоматизації виробництва, вимоги ергономіки, позитивний досвід з охорони праці тощо;
- забезпечує належне утримання будівель та споруд, виробничого обладнання та устаткування, моніторинг за їх технічним станом;
- забезпечує усунення причин, що призводять до нещасних випадків, професійних захворювань, та здійснення профілактичних заходів, визначених

					КС КРБ 123.311.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата	Безпека життєдіяльності, основи охорони праці			Літ.	Арк.	Аркушів	
Розроб.		Гарнік В.Р.									
Перевірів.		Луцків А.М.								55	
Консульт.		Пилипець М.І.						ТНТУ, каф. КС, гр. СІс-42			
Н. Контр.		Луцик Н.С.									
Затверд.		Осухівська Г.М.									

комісіями за підсумками розслідування цих причин;

- організовує проведення аудиту охорони праці, лабораторних досліджень умов праці, оцінку технічного стану виробничого обладнання та устаткування, атестацій робочих місць на відповідність нормативно-правовим актам з охорони праці в порядку і строки, що визначаються законодавством, та за їх підсумками вживає заходів з усунення небезпечних і шкідливих для здоров'я виробничих факторів;

- розробляє і затверджує положення, інструкції, інші акти з охорони праці, що діють у межах підприємства та встановлюють правила виконання робіт і поведінки працівників на території підприємства, у виробничих приміщеннях, на будівельних майданчиках, робочих місцях відповідно до нормативно-правових актів з охорони праці, забезпечує безоплатно працівників нормативно-правовими актами підприємства з охорони праці;

- здійснює контроль за дотриманням працівником технологічних процесів, правил поведінки з машинами, механізмами, устаткуванням та іншими засобами виробництва, використанням засобів колективного та індивідуального захисту, виконанням робіт відповідно до вимог з охорони праці;

- організовує пропаганду безпечних методів праці та співробітництво з працівниками у галузі охорони праці.

Роботодавець несе безпосередню відповідальність за порушення нормативно-правових актів з охорони праці. Служба охорони праці створюється роботодавцем на підприємстві з кількістю працівників 50 і більше. На підприємстві з кількістю працівників менше 50 осіб функції цієї служби можуть виконувати у порядку сумісництва особи, що пройшли перевірку знань з охорони праці відповідними державними службами. Якщо кількість працівників менше 20 осіб, для виконання функцій служби охорони праці можуть залучатися сторонні спеціалісти на договірних засадах. Служба охорони праці підпорядковується безпосередньо роботодавцю і прирівнюється до керівників і спеціалістів основних виробничо-технічних служб.

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

Спеціалісти служби охорони праці у разі виявлення порушень охорони праці мають право:

- видавати керівникам структурних підрозділів підприємства обов'язкові для виконання приписи щодо усунення наявних недоліків, одержувати від них необхідні відомості, документацію і пояснення з питань охорони праці;
- вимагати відсторонення від роботи осіб, які не пройшли передбачених законодавством медичного огляду, навчання, інструктажу, перевірки знань і не мають допуску до відповідних робіт або не виконують вимог нормативно-правових актів з охорони праці;
- зупиняти роботу виробництва, дільниці, машин, механізмів, устаткування та інших засобів виробництва у разі порушень, які створюють загрозу життю або здоров'ю працівників;
- надсилати роботодавцю подання про притягнення до відповідальності працівників, які порушують вимоги щодо охорони праці.

Ліквідація служби охорони праці допускається тільки у разі ліквідації підприємства чи припинення використання найманої праці фізичною особою.

Законодавство про охорону праці передбачає і обов'язки працівників. Зокрема вони зобов'язані:

- дбати про особисту безпеку і здоров'я, а також про безпеку і здоров'я оточуючих людей у процесі виконання будь-яких робіт під час перебування на території підприємства;
- знати і виконувати вимоги нормативно-правових актів з охорони праці, правила поведінки з машинами, механізмами, устаткуванням та іншими засобами виробництва, користуватися засобами колективного та індивідуального захисту;
- проходити у встановленому законодавством порядку попередні та періодичні медичні огляди.

Працівник несе безпосередню відповідальність за порушення зазначених вимог. Дотримання правил безпеки і виробничої санітарії залежить не тільки

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

від виконання роботодавцем своїх обов'язків, а й від того, наскільки кожен працівник знає і виконує правила під час роботи. Тому всі працівники при прийомі на роботу і в процесі роботи проходять на підприємстві інструктаж з охорони праці, надання першої медичної допомоги потерпілим від нещасних випадків, правил поведінки при виникненні аварій.

Навчання й інструктаж працівників з охорони праці є складовою частиною системи управління охороною праці і проводиться з усіма працівниками в процесі їхньої трудової діяльності. Інструктаж працівників залежно від характеру та часу його проведення буває вступний (при прийомі на роботу); первинний (на робочому місці з усіма працівниками: на роботах із підвищеною небезпекою - один раз на квартал, на інших роботах — один раз на півроку; проводиться або індивідуально, або з групою працівників, що виконують однотипні роботи, за програмою первинного інструктажу); позаплановий (при зміні правил з охорони праці, заміні устаткування чи за інших змін факторів, що впливають на безпеку праці); цільовий (при виконанні разових робіт, не пов'язаних із прямими обов'язками за фахом).

Навчання та інструктаж працівників з охорони праці проводиться у відповідності до Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці від 26.01.2005 р. № 15 – НПАОП 0.00.4.36 – 05.

4.2 Заходи, які забезпечують створення оптимальних метеорологічних умов у приміщеннях з використанням ПК

Метеорологічні умови визначаються такими параметрами:

- температурою повітря, t (°C);
- відносною вологістю, ϕ (%);
- швидкістю повітря, v (м/с).

Крім цих параметрів, що є основними, не слід забувати і про атмосферний тиск (P , Па), який впливає не тільки на парціальний тиск основних компонентів повітря (кисень та азот), а й на процес дихання.

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Життєдіяльність людини проходить в умовах достатньо широкого діапазону тиску 734—1276 гПа. Однак тут треба пам'ятати, що для здоров'я людини є небезпечною швидка зміна тиску, а не сама величина цього тиску. Наприклад, швидке зниження тиску лише на декілька гектопаскалів щодо нормальної величини 1013 гПа спричиняє хворобливі відчуття.

Необхідність урахування основних параметрів метеорологічних умов диктується наслідками в змінах стану людини. Особливо переконливо це можна пояснити під час розглядання теплового балансу між організмом людини і навколишнім середовищем.

Величина тепловиділення (Q) організмом людини залежить від ступеня фізичного напруження у певних метеорологічних умовах і складає від 85 (у стані спокою) до 500 Дж/с (тяжка робота).

Людина постійно перебуває в процесі теплової взаємодії з навколишнім середовищем. Для того, щоб фізіологічні процеси проходили нормально, теплота, що виділяє організм, повинна віддаватись в навколишнє середовище. Співвідношення між кількістю цієї теплоти й охолоджувальною здатністю середовища характеризує умови як комфортні. В умовах комфорту у людини не виникає турбот щодо її температурних відчуттів охолодження чи перегрівання.

Віддача теплоти організмом людини в навколишнє середовище відбувається через теплопровідність крізь одяг (Q_T), конвекцією тіла (Q_K), випромінюванням на навколишні поверхні (Q_B), випаровуванням вологи з поверхні шкіри ($Q_{\text{вип}}$). Частина теплоти витрачається на нагрівання повітря, яким дихає людина (Q_r).

Кількість теплоти, яка віддається організмом людини будь-якими шляхами, залежить від того чи іншого параметра мікроклімату. Так, тепловіддача конвекцією залежить від температури навколишнього повітря і швидкості його переміщення. Випромінювання теплоти відбувається у напрямі поверхонь, що оточують людину, мають нижчу температуру поверхні одягу (27—31 °С) і відкритих частин тіла людини (близько 33,4 °С). Під час впливу високих температур навколишньої поверхні (30—35 °С) тепловіддача випромінюванням повністю відсутня, а під час впливу більш високих

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

температур теплообіг йде у зворотному напрямі — від поверхні до людини. Віддача теплоти за рахунок випаровування залежить від відносної вологості і швидкості переміщення повітря. У стані спокою, коли температура навколишнього середовища 18°C , частка Q_K складає близько 30 % всієї теплоти, яка віддається людиною, $Q_{\text{Вім}} = 20 \%$ і $Q_n \sim 5 \%$.

Під час зміни температури повітря, швидкості його руху і вологості, наявності близько людини нагрітої поверхні, в умовах її фізичної праці тощо — це співвідношення змінюється.

Нормальне теплове самопочуття (комфортні умови), відповідно до конкретних видів роботи, забезпечується при дотриманні теплового балансу: $Q = Q_T + Q_K + Q_{\text{Вім}} + Q_n >$ тому температура внутрішніх органів людини залишається постійною (близько $36,6^{\circ}\text{C}$). Ця здатність людського організму до утримання постійної температури під час зміни параметрів мікроклімату та під час виконання роботи будь-якої важкості називається *терморегуляцією*.

Висока температура впливає на людину і сприяє розширенню судин кровообігу. Відповідно має місце підвищений приплив крові до поверхні тіла, і тепловіддача в навколишнє середовище значно підвищується. Однак, коли температура навколишнього середовища і поверхні досягає $30\text{—}35^{\circ}\text{C}$, віддача теплоти конвекцією і випромінюванням в основному припиняється. Більш висока температура повітря сприяє тому, що більша частина теплоти віддається через випаровування її з поверхні шкіри. В таких умовах організм губить відповідну кількість вологи, а разом з нею і солі, які відіграють важливу роль в життєдіяльності організму.

В умовах зниження температури повітря реакція людського організму на ці зміни інша — судини кровообігу шкіри звужуються, приплив крові до поверхні тіла зменшується, і віддача теплоти конвекцією і випромінюванням зменшується. Таким чином, для теплового самопочуття людини важливим є певне сполучення температури, відносної вологості і швидкості руху повітря.

Вологість повітря значною мірою впливає на терморегулювання організму. Підвищена вологість ($\phi > 85 \%$) ускладнює терморегулювання через зниження випару поту, а досить низька

					КС КРБ 123.311.00.00 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

вологість ($\phi < 20 \%$) спричиняє сухоту слизових оболонок шляхів дихання. Оптимальні величини відносної вологості складають 40 — 60 %.

Рух повітря в приміщеннях є важливим чинником, який впливає на теплове самопочуття людини. В умовах спеку рух повітря сприяє підвищенню віддачі теплоти організмом і поліпшує його стан, але в холодну пору року цей вплив не є сприятливим.

Мінімальна швидкість руху повітря, яку відчуває людина, складає 0,2 м/с. Взимку швидкість руху повітря не повинна перевищувати 0,2—0,5 м/с, а влітку 0,2—1,0 м/с.

Швидкість повітря також впливає на розподіл шкідливих речовин у приміщенні. Повітряні потоки можуть розповсюджувати їх по всьому об'єму приміщення, переводити пил з осілого у зважений стан.

Під впливом високої температури повітря, інтенсивного теплового випромінювання виникає загроза перегрівання організму людини, яке характеризується підвищенням температури тіла, рясним потовиділенням, прискореним пульсом і диханням, різкою слабкістю, запамороченням, а в тяжких випадках — появою судом і виникненням теплового удару.

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

При виконанні кваліфікаційної роботи бакалавра проаналізовано та досліджено особливості організації комп'ютерних систем, які дають змогу в реальному часі зчитувати показники роботи агрегатів транспортних засобів та візуалізувати їх. У роботі запропоновано проект системи, яка забезпечує зчитування показників двигуна та відображає їх значення в реальному часі, а також щосекундно виконує записи цих показників у CSV-файл.

При проектуванні комп'ютерної системи, яка отримує дані з CAN-шини використано сучасні апаратні та програмні засоби. Апаратні засоби, які забезпечують отримання даних і їх відображення включають в себе:

- Raspberry PI 3;
- сенсорний екран 3,5”RPI LCD.

Для забезпечення живлення мінікомп'ютера Raspberry PI розроблено відповідну схему, яка дає змогу перетворювати постійну напругу 12 В від автомобільного акумулятора у напругу 5В та використовує лінію живлення АСС. При цьому у схемі використовується мікроконтролер Attiny85, конвертор постійного струму LM2596 та регулятор постійної напруги 78L33, які дозволяють безпечно завершити роботу Raspberry PI протягом 45 секунд після вимкнення двигуна авто.

Програмне забезпечення комп'ютерної системи зчитування та відображення показників двигуна через CAN-шини забезпечує процеси комунікації між апаратними пристроями і складається з 5 модулів, які написані мовою програмування Python. Конфігураційний модуль містить множину глобальних змінних і є зв'язуючою ланкою для інших модулів. Модуль для роботи з електронним блоком управління (через CAN-шину) забезпечує зчитування даних, а модуль логування – виконує функції запису показників у CSV-файл. Модуль виводу на екран значень показників двигун забезпечує асинхронність операцій з відображення та запису даних, що дає змогу водієві бачити в реальному часі зміну показників двигуна. Модуль вимкнення Raspberry PI є найбільш функціонально простим і забезпечує ручне вимкнення системи.

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. G. Guido, A. Vitale, V. Astarita, F. Saccomanno, V. P. Giofr'e, and V. Gallelli, "Estimation of safety performance measures from smartphone sensors," *Procedia—Social and Behavioral Sciences*, vol. 54, pp. 2012. 1095–1103.
2. OBD2 protocols – OBDTester. URL: http://www.obdtester.com/obd2_protocols (дата звернення: 25.04.2024).
3. OBDII pinout - 0xicf. URL: <https://0xicf.wordpress.com/tag/obdii-pinout/> (дата звернення: 25.04.2024).
4. SAE J1850- Interfacebus.com. URL: http://www.interfacebus.com/Automotive_SAE_J1850_Bus.html (дата звернення: 29.04.2024).
5. ISO 14230-4:2000 - ISO.org. URL: <https://www.iso.org/standard/28826.html> (дата звернення: 29.04.2024).
6. ISO 15765-4:2016 - ISO.org. URL: <https://www.iso.org/standard/67245.html> (дата звернення: 01.05.2024).
7. ISO 9141-2:1994 - ISO.org. URL: <https://www.iso.org/standard/16738.html> (дата звернення: 01.05.2024).
8. Захарченко М. В. Асиметричні методи шифрування в телекомунікаціях О.: ОНАЗ, 2011. 184 с.
9. Микитишин А.Г., Митник М.М., Стухляк П.Д., Пасічник В.В. Комп'ютерні мережі. Книга 2. Львів, «Магнолія 2006», 2014. 312 с.
10. Микитишин А.Г., Митник М.М., Стухляк П.Д. Телекомунікаційні системи та мережі. Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2016. 384 с.
11. Agranat I. Detecting Bats with Ultrasonic Microphones. Multidisciplinary Digital Publishing Institute. p. 14.
12. Agrawal S. Raspberry Pi Camera and its Variety. URL: <https://iot4beginners.com/raspberrypi-camera-and-its-variety/> (дата звернення: 05.05.2024 р.
13. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

14. Raspberry Pi Computer Boards. URL: <https://www.okdo.com/c/pi-shop/the-raspberry-pi/> (дата звернення 05.05.2024 р.).

15. J1979_201408 - SAE International. URL: http://standards.sae.org/j1979_201408/ (дата звернення 05.05.2024 р.).

16. OBD - Elm Electronics. URL: <https://www.elmelectronics.com/products/ics/obd/> (дата звернення 08.05.2024 р.)

17. OBDLink SX USB | OBDLink. URL: <http://www.obdlink.com/sxusb/> (дата звернення 08.05.2024 р.)

18. STN1110 - Multiprotocol OBD Interpreter IC. URL: <http://www.obdsol.com/solutions/chips/stn1110/> (дата звернення 10.05.2024 р.)

19. FT230X USB Bridge | UART - FTDIChip. URL: <http://www.ftdichip.com/Products/ICs/FT230X.html> (дата звернення 10.05.2024 р.)

20. Pygame.org. URL: <https://www.pygame.org/> (дата звернення 12.05.2024 р.).

21. Осухівська Г.М., Луцик Н.С., Луцик Н.С., Паламар А.М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль, ТНТУ. 2022. 28 с.

22. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Київ. 2018.

23. Катренко Л.А., Катренко А.В. Охорона праці в галузі комп'ютингу. Львів: Магнолія-2006. 2012. 544 с.

24. Бедрій Я. Основи охорони праці користувачів персональних комп'ютерів: навчальний посібник для студентів ВНЗ та інженерів-практиків. Навчальна книга-Богдан. 2014. 144 с.

					<i>КС КРБ 123.311.00.00 ПЗ</i>	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС

_____ Осухівська Г.М.

“ ____ ” _____ 2024 р

КОМП'ЮТЕРНА СИСТЕМА ЗЧИТУВАННЯ ТА ВІДОБРАЖЕННЯ

ПОКАЗНИКІВ ДВИГУНА АВТО ЧЕРЕЗ CAN ШИНУ

ТЕХНІЧНЕ ЗАВДАННЯ

на 10 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

_____ к.т.н., доц. Луцків А.М.

« ____ » _____ 2024 р.

«ВИКОНАВЕЦЬ»

Студент групи СІс-42

_____ Гарнік В.Р.

« ____ » _____ 2024 р.

Тернопіль 2024

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютерна система зчитування та відображення показників двигуна авто через CAN шину».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.311.00.00

1.2 Виконавець

Студент групи СІс-42, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Гарнік Владислав Ростиславович.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/7-408 від 24.04.2024 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 24.04.2024 р.

Плановий термін завершення виконання кваліфікаційної роботи – 24.06.2024 р.

1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи.

Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

Комп'ютерна система зчитування та відображення показників двигуна авто через CAN шину призначена для збору інформації про стан та параметри використання ресурсів агрегатів транспортного засобу, а також їх візуалізації в режимі реального часу. Впровадження такої системи є актуальним для власників авто, які не мають сучасних пристроїв діагностики, зокрема, фіксації обертів двигуна, температури охолоджуючої рідини, миттєвого розходу палива та інших.

Окрім цього, комп'ютерна система може використовуватися для збору даних, які в подальшому будуть опрацьовані з метою прогнозування поведінки керування транспортним засобом водіїв, для виявлення факторів, які найбільш негативно впливають на роботу агрегатів та при прогнозуванні зносу структурних елементів двигуна.

До складу комп'ютерної системи зчитування та відображення показників двигуна повинні входити мікроконтролер для керування процесом збору даних з

CAN-шини, дисплей для зручного і водночас простого сприйняття значень цих показників і як допоміжний пристрій – схема живлення мікроконтролера і за потреби сенсорного екрану.

Розширенням функціональності системи може бути інтеграція можливостей збереження та аналізу даних для подальшого використання. Наприклад, збір і збереження історії показників може допомогти виявити зміни в роботі двигуна та інших систем транспортного засобу з часом. Також можливими є розширені можливості візуалізації, такі як графіки та діаграми, що дозволять користувачам легше аналізувати та розуміти зібрані дані.

Ще однією важливою можливістю може бути інтеграція системи з іншими пристроями транспортного засобу або зовнішніми системами. Наприклад, зв'язок з мобільним додатком або хмарним сервісом може дозволити власникам отримувати сповіщення про стан авто або аналізувати дані віддалено.

Також, важливим аспектом може бути забезпечення системи захисту даних та конфіденційності. Застосування шифрування та інших заходів захисту може запобігти несанкціонованому доступу до зібраних даних про транспортний засіб.

Загалом, розширення функціональності комп'ютерної системи зчитування та відображення показників двигуна може підвищити користувацький досвід, забезпечити більшу безпеку та ефективність у використанні транспортного засобу.

2.2 Мета створення системи

Мета створення комп'ютерної системи зчитування та відображення показників двигуна через CAN-шину полягає у розширенні функціональності базових сервісів, які надають електронні системи автомобіля шляхом їх отримання і візуалізації на центральній панелі транспортного засобу.

Для досягнення поставленої мети потрібно розв'язати наступні задачі:

- дослідження принципів роботи електронного блоку автомобіля;
- аналіз стандартів і вимог, які висуваються до CAN-вузлів ;

- дослідження особливостей стандарту OBD II;
- проектування концептуальної архітектури комп'ютерної системи зчитування та відображення показників двигуна через CAN-шину;
- розробка структурної схеми комп'ютерної системи;
- обґрунтування вибору апаратного забезпечення комп'ютерної системи зчитування та відображення показників двигуна;
- розробка схеми для живлення і безпечного вимкнення мікроконтролера для керування процесами зчитування та відображення даних;
- налаштування системне програмного забезпечення для ефективного і безпечного функціонування комп'ютерної системи;
- розробка програмного забезпечення для зчитування та відображення даних показників авто транспортних засобів;
- розробка графічного користувацького інтерфейсу системи.

2.3 Характеристика об'єкту

Сучасні транспортні засоби надають багато можливостей контролю показників їхньої роботи, зокрема, температури охолоджуючої рідини двигуна, тиску у шинах, DTC і перетворюються на засоби збору інформації для подальшої оптимізації виробництва та підвищення надійності експлуатації.

Усі зібрані дані також повинні бути зареєстровані так, щоб їх можна було проаналізувати в майбутньому (наприклад, аналіз ефективності використання потужності двигуна). Результатом роботи повинен бути проект реалізації комп'ютерної системи, яка інтегрується у 3D-панель та кріпиться у зоні центральної консолі автомобіля.

Діагностика стану систем сучасного автомобіля, окремих його вузлів чи компонентів, сьогодні здійснюється із застосуванням сканерів, які підтримують стандарт OBD II. Даний стандарт підтримує п'ять базових протоколів, які забезпечують обмін повідомленнями між сканером та блоком керування авто.

За допомогою DLC-роз'єму, який відповідає нормам стандарту SAE J1962, можна виконати підключення зовнішнього пристрою зчитування даних до електронного блоку.

Найбільш важливим компонентом апаратного забезпечення комп'ютерний системи зчитування даних з CAN-шини є мікроконтролер. Він комунікує з центральним електронним блоком управління транспортного засобу через адаптер USB OBD. Після цього одержані дані опрацьовуються розробленим програмним забезпеченням.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

Комп'ютерна система зчитування та відображення показників двигуна авто через CAN шину повинна дозволяти виконувати наступні функції:

- можливість підключення до електронного блоку керування автомобілем через CAN-шину;
- можливість зчитувати показники рівня температури охолоджуючої рідини в режимі реального часу;
- можливість отримувати дані про показник двигуна щодо кількості обертів на хвилину в поточний момент часу;
- можливість розраховувати поточну швидкість автомобіля на основі існуючих даних з врахуванням радіуса коліс;
- можливість визначати навантаження на двигун;
- здатність вимірювати величину потоку повітря, яке використовується при роботі двигуна;
- здатність розраховувати на основі обертів двигуна та швидкості руху авто, яка передача в даний час увімкнута;

- можливість візуалізувати перераховані вище показники на сенсорному екрані автомобіля;
- здатність фіксувати показники двигуна шляхом запису інформації у файл.
- забезпечення функції вимкнення мікроконтролера користувачем за допомогою графічного інтерфейсу.

3.1.1 Вимоги до способів та засобів зв'язку між компонентами системи

Основними вимогами до способів та засобів зв'язку між компонентами комп'ютерної системи зчитування та відображення показників двигуна через CAN-шину є наявність фізичного з'єднання між такими компонентами як:

- блок електронного управління авто – зв'язок через CAN-шину;
- Raspberry PI – фізичне підключення сенсорного екрану, блоку живлення та CAN-шини;
- сенсорний екран 3,5” RPI LCD – підключається до Raspberry PI за допомогою цифрових контактів;
- SD-карта – поміщається у порт Raspberry PI.

Потрібно відмітити, що комунікація між OBD портом та Raspberry PI організовується за допомогою перехідника OBD II-USB. Передача та обмін даними між Raspberry PI та сенсорним дисплеєм здійснюється по шині SPI..

Для програмного доступу до Raspberry PI використовується безпроводна технологія WiFi та SSH.

3.1.2 Вимоги по діагностуванню системи

Діагностика комп'ютерної системи зчитування та відображення показників двигуна через CAN-шину повинна проводитися згідно розкладу регламентних робіт та у випадку збою функціональності або відсутності комунікації між пристроями. Діагностичні процедури передбачають верифікацію параметрів працездатності апаратної і програмної складової системи.

На апаратному рівні проводяться перевірки щодо встановлення відповідності технічних характеристик пристроїв до визначених у специфікації параметрів. При цьому можуть застосовуватися зовнішні вимірювальні пристрої.

На програмному рівні доцільно проводити тестування кожного окремого компонента, а також перевірку системи в цілому.

3.1.3 Перспективи розвитку, модернізація системи

Розширенням функціональності системи може бути інтеграція можливостей збереження та аналізу даних для подальшого використання. Наприклад, збір і збереження історії показників може допомогти виявити зміни в роботі двигуна та інших систем транспортного засобу з часом. Також можливими є розширені можливості візуалізації, такі як графіки та діаграми, що дозволять користувачам легше аналізувати та розуміти зібрані дані.

Ще однією важливою можливістю може бути інтеграція системи з іншими пристроями транспортного засобу або зовнішніми системами. Наприклад, зв'язок з мобільним додатком або хмарним сервісом може дозволити власникам отримувати сповіщення про стан авто або аналізувати дані віддалено.

Загалом, розширення функціональності комп'ютерної системи зчитування та відображення показників двигуна може підвищити користувацький досвід, забезпечити більшу безпеку та ефективність у використанні транспортного засобу.

3.1.4 Вимоги до надійності системи

Надійність комп'ютерної системи зчитування та відображення показників двигуна авто через CAN-шину визначається сумарним значенням показників апаратного і програмного забезпечення, а також самого транспортного засобу.

Окрім цього, важливим є показник надійності каналів передачі даних як на фізичному рівні, так і на логічному. Це стосується безпроводної передачі даних та організованого каналу зв'язку з мережею Інтернет.

Важливим атрибутом надійності є також точність і вчасність отримання даних з CAN-шини та асинхронність запису та відображення даних на екрані.

3.1.5 Вимоги до функцій та задач, які виконує система

До основних вимог і функцій комп'ютерної системи зчитування та відображення показників двигуна належить можливість візуалізації отриманих значень показників в режимі реального часу на сенсорному екрані, вмонтованому у центральну консоль автомобіля.

Окрім цього, важливою функціональністю є здатність кожної секунди здійснювати запис даних про показники роботи агрегатів автомобіля у файл формату csv.

3.1.6 Вимоги до апаратного забезпечення

До апаратного забезпечення комп'ютерної системи зчитування та відображення показників двигуна авто належать:

- центральний блок електронного управління авто;
- Raspberry PI 3 Model B;
- сенсорний екран 3,5” RPI LCD;
- перехідник OBD II-USB – OBDLink SX;
- SD-карта;
- плата керування подачею живленням.

3.1.7 Вимоги до програмного забезпечення

Вимогами до програмного забезпечення на системному рівні є:

- операційна система Raspbian Lite з налаштованими сервісами SSH та доступом до WiFi мережі;
- драйвер сенсорного екрану.

Програмне забезпечення для управління системою зчитування та відображення показників двигунів повинно включати наступні модулі:

- модуль комунікації з центральним електронним блоком керування авто;
- модуль візуалізації показників двигуна;
- модуль логування показників двигуна;
- модуль управління системою;
- модуль вимкнення Raspberry PI.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:

- 1 Концептуальна архітектура системи зчитування та візуалізації даних з CAN-шини.
- 2 Структурна схема організації системи зчитування даних з CAN-шини.
- 3 Схема підключення компонентів системи до Arduino Uno.
- 4 Структурна схема модулів програмного забезпечення.

*Примітка: У комплект документації можуть вноситися міни та доповнення в процесі розробки.

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ Етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка і затвердження технічного завдання	26.04-28.04.2024
2	Аналіз технічного завдання	28.04-03.05.2024
3	Аналіз вимог до комп'ютерної системи зчитування даних з CAN-шини	05.05-08.05.2024

Продовження таблиці 1

№ Етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
4	Проектування комп'ютерної системи зчитування та відображення показників двигуна авто	08.05-18.05.2024
5	Програмне забезпечення комп'ютерної системи зчитування та відображення показників двигуна авто	18.05-25.05.2024
6	Розробка інструкцій щодо використання комп'ютерної системи	26.05-05.06.2024
7	Безпека життєдіяльності, основи охорони праці	05.06-10.06.2024
8	Оформлення кваліфікаційної роботи	10.06-14.06.2024
9	Попередній захист кваліфікаційної роботи	14.06-19.06.2024
10	Захист кваліфікаційної роботи	24.06-28.06.2024

6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

Додаток Б

Перелік елементів