

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
Кафедра електроніки і комп'ютерної техніки

## ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

бакалавр на тему:

Бортовий модуль контролю і управління автомобілем на основі  
Arduino

Завідуючий кафедрою

А. С. Опанасюк

Керівник кваліфікаційної роботи  
бакалавра

М,С. Шевченко

Виконав студент

С. В . Киричко

Суми - 2024

## РЕФЕРАТ

У дипломному проекті розроблено бортовий модуль контролю і управління автомобілем на основі Arduino, який можна встановити на будь-який автомобіль з двигуном внутрішнього згоряння.

У цьому проекті було проведено аналіз відповідної літератури, потім було поставлено задачу до модуля враховуючи всі необхідності. На основі цього аналізу було спроектовано бортовий модуль контролю і управління автомобілем, а саме було виконано: розроблення схеми електричної принципової, а також алгоритму, вибір елементної бази, розроблено програмне забезпечення, а також був розроблений окремий розділ присв'ячений налаштуванню бортового модуля.

Пристрій може встановлюватися на будь-який автомобіль після попереднього узгодження параметрів авто, на яку передбачається встановлення, з даними, зазначеними в технічному завданні.

Дипломний проект складається з пояснювальної записки обсягом 54 сторіноки, яка включає 15 рисунків, 2 таблиці, 12 джерел та 2 додатки.

Ключові слова: Бортовий модуль, мікроконтролер, Arduino Nano.

## ЗМІСТ

РЕФЕРАТ.....	2
ЗМІСТ.....	3
ВСТУП.....	4
1. ОГЛЯД ЛІТЕРАТУРИ ТА ПОСТАНОВКА ЗАДАЧІ ПРОЕКТУВАННЯ.....	5
1.1 Бортовий модуль.....	5
1.1.1 Хронологія розвитку дистанційного управління об'єктом .....	6
1.1.2 Типи бортових комп'ютерів .....	7
1.1.3 Стандартний інтерфейс OBD-II.....	7
1.1.4 Протоколи стандарту OBD-II.....	9
1.1.5 Специфікація роз'єму .....	11
1.1.6 Режим роботи OBD-II.....	12
1.2 Постановка завдання проектування.....	17
2. СИНТЕЗ І ОБҐРУНТУВАННЯ АЛГОРИТМУ РОБОТИ І СТРУКТУРНОЇ СХЕМИ ПРИСТРОЮ.....	19
2.1 Розробка алгоритму бортового модуля контролю і управлінням автомобілем на основі Arduino .....	19
2.2 Розробка структурної схеми .....	21
3. РОЗРОБКА ФУНКЦІОНАЛЬНОЇ СХЕМИ.....	23
3.1 Розробка функціональної схеми пристрою.....	23
4. РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ .....	25
4.1 Вибір елементної бази.....	25
5. НАЛАШТУВАННЯ ПРИСТРОЮ .....	37
6. ВИСНОВКИ.....	40
НАУКОВІ ПРАЦІ СТУДЕНТА .....	41
7. СПИСОК ЛІТЕРАТУРИ.....	42
ДОДАТОК А .....	44
ДОДАТОК Б.....	45

					<b>ЕЛІТ 6.171.00.10.088 ПЗ</b>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>				
<i>Розроб.</i>		Киричко С.В			<b>Бортовий модуль контролю і управління автомобілем на основі Arduino</b>	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Перевір.</i>		Шевченко М.С					3	48
<i>Реценз.</i>						<b>СумДУ ЕС-01</b>		
<i>Н. Контр.</i>		Гапич Н.В.						
<i>Утверд.</i>		Опанасюк А.С.						

## ВСТУП

У сучасному автомобільному світі виникає необхідність у забезпеченні безпеки та підвищенні ефективності бюджетних і старих моделей автомобілів. Такі автомобілі часто залишаються без уваги через впровадження новітніх технологій. Особливо актуальним є те, що відсутність сучасних систем керування та моніторингу в таких автомобілях може призвести до зростання ризиків при їх експлуатації, що негативно впливає на безпеку водіїв та пасажирів.

Ця бакалаврська робота спрямована на розробку бортового модуля контролю і управління автомобілем, базованого на платформі Arduino. Основна мета цієї роботи полягає у створенні доступного та ефективного рішення для покращення функціональності та безпеки бюджетних та старих автомобілів.

Даний модуль взаємодіє з автомобільною системою, отримуючи інформацію зі сканера ELM 327, розшифровує її, а потім виводить на екран через інтерфейс Arduino. Докладний опис принципів роботи модуля та використаних технологій дозволить досягти поставленої мети, що є вагомим внеском у сферу автомобільної безпеки та технологій.

Таким чином, ця робота не лише розглядає можливості покращення старих автомобілів, але й сприяє підвищенню їхньої надійності та безпеки. Реалізація даного проекту може значно знизити ризики, пов'язані з використанням застарілих транспортних засобів, та забезпечити водіям і пасажирам додатковий рівень захисту.

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		4

# 1. ОГЛЯД ЛІТЕРАТУРИ ТА ПОСТАНОВКА ЗАДАЧІ ПРОЕКТУВАННЯ

## 1.1 Бортовий модуль

**Бортовий модуль** - це пристрій автомобільної електроніки, який вбудований у автомобілі для контролю, вимірювання та управління різними системами автомобіля. Цей модуль зазвичай включає в себе різні сенсори, програмне забезпечення та інтерфейси, які дозволяють зчитувати та обробляти дані з різних систем автомобіля, таких як двигун, трансмісія, системи безпеки, кондиціонування повітря та інші. Бортові модулі також виконують функцію діагностики для виявлення несправностей та відображати інформацію на панелі приладів для водія.



Рисунок 1.1-Приклад бортового модуля управління

Різноманітні бортові модулі управління, які використовуються в сучасних автомобілях, на сьогоднішній день представляють собою складні технічні системи, що забезпечують вимірювання різноманітних параметрів автомобіля, та надають водіям значну кількість інформації про функціональний стан автомобіля.

Ці модулі виявляють широке застосування в контролі та діагностиці автомобільних систем, починаючи від вимірювання рівня палива та витрати палива, і закінчуючи аналізом миттєвих параметрів руху, таких як температура двигуна, швидкість руху, кількість обертів двигуна, навантаження на двигун та трансмісію, тощо.

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		5

Також хочу зауважити , що бортові модулі управління є не лише засобом контролю за станом автомобіля, але й елементом, що забезпечує безпеку водіїв та пасажирів. Функція аварійного режиму реагує на критичні поломки авто та при необхідності обмежує потужність двигуна (тобто входить в аварійний режим), або навіть блокувати його роботу, забезпечуючи тим самим додатковий рівень безпеки на дорозі.

Отже, можна зробити висновок, що бортові модулі управління є важливими складовими частинами сучасного автомобіля, вони сприяють ефективному та безпечному функціонуванню, а також роблять керування автомобілем безпечнішим

### 1.1.1 Хронологія розвитку дистанційного управління об'єктом

Останнім часом, майже всі авто комплектуються модулями управління авто, тобто бортовими комп'ютерами. Такі пристрої, з'явилися доволі давно, але мали значно менший функціонал, вони лише слідкували за роботою тих чи інших систем авто. Зараз такі пристрої можуть повністю керувати авто, від систем циркуляції повітря до обертів двигуна, положенням заслонок, випередженням запалення.

Розвиток поратативних комп'ютерів почав стрімко зростати з винайденням транзисторів, приблизно в 40-х роках минулого століття. Використання транзисторів дозволило робити комп'ютери більш потужними при менших габаритах. На початку 1970 років комп'ютери почали з'являтися в автомобілях.

Першою в цьому стала компанія Volkswagen. Вона використала комп'ютер як керуючий орган авто. Компанія змогла реалізувати електронне управління системою впрыску палива, це значно збільшило потужність, та знизило розхід.

Через деякий час двигуни почали оснащувати безконтактним запаленням.

Через 10 років БК вже рахував розхід палива, виходячи з часу відкриття форсунок і відстані на яку хватає топлива в бензобакові авто. А ще через рік було розроблено антиблокувальну систему (ABS). БК під час гальмування перевіряв чи не заблоковані колеса і якщо вони були заблоковані, то зменшував гальмівне зусилля, це значно підвищило ефективність гальмування.

З часом виробники почали виносити функцію одної системи в різні і зв'язувати їх магістральною шиною, якою кожен окремий блок міг передавати

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		6

інформацію іншому , а ЕБК (Електронний блок керування ) в свою чергу аналізує ці дані , та керує окремими блоками .

### 1.1.2 Типи бортових комп'ютерів

Існують різні типи бортових комп'ютерів (БК), які можна класифікувати за їх призначенням та функціональністю. Одним з основних типів є маршрутний комп'ютер, який надає водієві інформацію про основні параметри поїздки. Він відображає такі дані, як швидкість, частота обертів двигуна, температура двигуна, залишок пального, температура повітря в салоні та зовні, напруга електричних систем та багато іншого. Хоча частково ці дані можна отримати зі штатних датчиків автомобіля, маршрутний комп'ютер залишається актуальним через точніші показники. Крім того, він може коригувати вихідні дані для обчислень після заміни різних комплектуючих і систем автомобіля.

Маршрутний комп'ютер також може допомогти з плануванням маршруту, підбирати оптимальну швидкість на певній ділянці шляху, підраховувати кілометри згідно з залишком палива в баку та проводити перевірку всіх вузлів і систем автомобіля.

Ще одним важливим типом бортових комп'ютерів є сервісний комп'ютер, який потрібен для виявлення неправильної роботи автомобіля. Він проводить діагностику всіх систем автомобіля та моментально сповіщає водія про виявлені проблеми або поломки. Сервісний комп'ютер є незамінним інструментом для підтримання автомобіля в належному технічному стані та попередження серйозних поломок.

Щодо дисплеїв, які використовуються в бортових комп'ютерах, то у дорогих моделях зазвичай встановлюються рідкокристалічні дисплеї, які забезпечують високу чіткість і зручність зчитування інформації. У більш простих комп'ютерах застосовуються цифрові дисплеї з трьома або чотирма розрядами, які також надають необхідну інформацію, але менш детально.

### 1.1.3 Стандартний інтерфейс OBD-II

Для спрощеного доступу до БК було розроблено стандартний інтерфейс, який майже у всіх автомобілів, починаючи з близько 2005 року, називається OBDII

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		7

(On-board diagnostic). Будь-який бортовий комп'ютер, який використовує протоколи стандарту OBD-II, може встановити зв'язок з майже будь-яким сучасним автомобілем.

Стандарт OBD II був розроблений в середині 1990-х років. Він дозволяє проводити спостереження в реальному часі за показниками двигуна, коробки передач, датчиків авто, додаткових пристроїв, а також діагностувати електричну мережу автомобіля на предмет несправностей. У цьому стандарті автовиробники використовують кілька різних протоколів для з'єднання з блоками управління автомобіля.

Майже всі нові автомобілі оснащені ЕБК, який збирає та аналізує в реальному часі дані про роботу двигуна та інших компонентів автомобіля. OBD-II автомобіля.

Завдання на специфікацію було видано в 1988 році, а перші автомобілі, що відповідали вимогам цієї специфікації, були випущені в 1994 році. Цей стандарт розроблявся організацією SAE (Society of Automotive Engineers - "Співтовариство автомобільних інженерів") і був прийнятий як обов'язковий для всіх легкових та легких комерційних автомобілів у США, що випускаються з 1996 року.

У 90-х роках існувало три протоколи:

- SAE J1850 VPW - General Motors;
- SAE J1850 PWM - Ford;
- ISO 9141-2 - протокол, який використовували більшість європейських та японських виробників автомобілів.

У зв'язку з цим співтовариство автомобільних інженерів включило всі три протоколи в стандарт OBD-II. Пізніше з'явився протокол ISO 14230-4 (інша назва KWP2000), який був покращеною версією ISO 9141-2.

Також, ще в 1980-х роках компанією Bosch був запропонований стандарт CAN (Controlled Area Network - "контрольована мережа"). Цей стандарт з'явився в автомобілях вже у 2003 році.

У Європі було розроблено інший варіант автодіагностики під назвою EOBD, який також базується на OBD-II. Цей стандарт став обов'язковим для всіх виробників Європи з січня 2001 року. ЄС навіть оштрафував Peugeot за невідповідність стандарту.

Також існує японський стандарт - JOBD. До OBD-II японські виробники застосовували версію OBD-I, яка починала свій шлях від 1989 року. Також версія

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		8



OBD-I використовувалась на деяких автомобілях ВАЗ , та ЗАЗ ,але широкого розповсюдження не отримала .

Версія OBD-III зараз розробляється інженерами. З 2008 року всі виробники автомобілів повинні використовувати лише CAN-протокол для зв'язку ЕБК з двигуном, трансмісією , додатковими пристроями та контролерами різних виконавчих механізмів та систем безпеки. Іншими словами, всі виробники автомобілів рухаються до єдиного протоколу, це дозволить спростити діагностування різних марок авто .

#### 1.1.4 Протоколи стандарту OBD-II

В даний час в стандарті OBD-II використовуються наступні протоколи:

- SAE J1850 PWM (Pulse Width Modulation - "модуляція ширини імпульсів"):
  - швидкість: 41.6 Кбайт/с;
  - контакт 2: Шина + (Bus positive Line);
  - контакт 10: Шина - (Bus negative Line);
  - довжина повідомлення: 12 байт.
- SAE J1850 VPW (Variable Pulse Width - "змінна ширина імпульсів"):
  - швидкість: 10.4 / 41.6 Кбайт/с;
  - контакт 2: Шина + (Bus positive Line);
  - довжина повідомлення: 12 байт.
- ISO 9141-2 та ISO 14230 (KWP2000) - ідентичні на фізичному рівні, але ISO 14230 є повільнішим. Розроблені Міжнародною організацією по стандартизації (ISO).

ISO 9141-2:

- швидкість: 10 Кбайт/с;
- контакт 7: K-Line;
- контакт 15: L-Line (опціонально);
- довжина повідомлення - 12 байт.

ISO 14230:

- швидкість: 10 Кбайт/с;
- контакт 7: K-Line;

						ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата			9





### 1.1.6 Режими роботи OBD-II

Автовиробники не зобов'язані підтримувати всі режими роботи, а також вони можуть вводити свої власні режими з порядковим номером, який перевищує 09. Проте правильне розуміння призначення всіх особливостей режимів допоможе зрозуміти принцип роботи системи OBD-II, тому розглянемо список режимів стандарту SAE J1979 детально.

0x01. Show current data (англ. – «показати поточні дані») – зчитування параметрів системи управління в режимі реального часу. В цьому режимі виводяться поточні параметри електронного блоку управління. Ці параметри поділяються на три групи:

- статуси моніторів. Монітор, в даному випадку, - спеціальна підпрограма, яка відповідає за запуск і виконання різних діагностичних тестів, а статус – просто показник стану, який може бути одним із чотирьох варіантів: «підтримується», «не підтримується», «завершено», «не завершено». Такі показники і відображаються на дисплеї сканера. Є два типи моніторів: постійні і непостійні. Постійні монітори відтворюються блоком постійно, відразу після запуску двигуна. Непостійні запускаються тільки при певних умовах і режимах роботи двигуна. Саме робота підпрограм в блоку управління в значній мірі визначає потужні діагностичні можливості сканерів нового покоління. Проте наявність тих чи інших моніторів сильно залежить від конкретної моделі автомобіля, тобто деякі монітори можуть відсутні.

- PIDs (англ. Parameter Identification Data – «ідентифікація параметрів даних»). Це основні параметри роботи датчиків і величин, якими характеризуються управляючі сигнали. Стандарт SAE J2012 визначає список цих кодів, але виробники мають право додавати свої специфічні коди. Звичайно, якщо подивитися, скільки параметрів насправді знає блок управління, то дане списку параметрів стандарту OBD-II не вражає. Така мала кількість параметрів реального часу – один із недоліків цього стандарту. Проте частіше за все цього мінімуму вистачає. Важливо знати, що всі параметри вже інтерпретовані електронним блоком управління (крім сигналів датчиків кисню), тобто у цьому списку немає параметрів, які

										Лист
										12
Вим.	Лист	№ докум.	Підпис	Дата						

характеризували б фізичні величини сигналів. Наприклад, замість того, щоб показати напругу на датчику температури охолоджуючої рідини, сканер відобразить саме значення температури. Це зручно, оскільки часто показники у заводських протоколах представлені в дуже малоінформативній формі. У той час як використання протоколу OBD-II дозволяє отримати перевагу в вигляді інтерпретації даних з датчиків.

- Поточна команда ЕБК на лампу Check Engine (англ. – «перевірте двигун»). Це всього один параметр, і він показує лише стан, тобто він не цифровий. Він відображає інформацію про те, яку команду блок відправив на включення/виключення цієї лампи. Як і в Україні, американські спеціалісти вивчили, як обманювати людей в такий спосіб: лампу Check Engine підключають паралельно до лампи тиску масла. Це роблять для того, щоб приховати світитися лампу «перевірте двигун», але не залишити слідів, адже коли повертають ключ запалювання - всі лампи повинні запалитися, включаючи і Check Engine. А коли двигун запуститься - всі лампи повинні згасити. Таким чином, Check Engine гасне одночасно з лампою аварійного тиску масла і створюється вигляд того, що помилка в ЕБК немає, а отже, з автомобілем все в порядку. Лампа Check Engine загоряється при виявленні відхилень або несправностей, які призводять до збільшення шкідливих викидів більш ніж в 1,5 рази порівняно з допустимими. При цьому в пам'ять записуються відповідні коди несправності.

0x02. Show freeze frame data (англ. – «показати дані замороженого кадру») – виведення збереженого кадру поточних параметрів блоку управління на момент виникнення кодів несправностей. Обертатися до цього режиму варто тільки тоді, коли в пам'яті ЕБК зберігаються які-небудь коди несправностей. У такому випадку сканер відобразить збережений кадр параметрів того моменту, коли був зафіксований код несправності. Іншими словами, миттєві дані з режиму 0x01 під час виникнення несправності.

Це корисна функція, яка допоможе знайти несправність, адже за її допомогою можна зрозуміти умови виникнення цієї самої несправності. Але найголовніше полягає в тому, що дані «замороженого» кадру потрібні для як можна більш точного відтворення цих умов під час тестової поїздки, коли всю

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		13

діагностичну роботу виконує сам блок управління, активуючи вже згадані вище монітори.

Кодів несправності в пам'яті може бути багато, а от «стоп-кадр» - зазвичай лише один (так поступає більшість автовиробників). Номер коду несправності, що відповідає збереженому кадру, можна знайти у цьому кадрі, зазвичай він у самому початку списку.

0x03. Show stored Diagnostic Trouble Codes (англ. – «показати збережені діагностичні коди несправностей») – вивід кодів несправностей із пам'яті ЕБК.

Діагностичне обладнання запитує зчитування кодів несправностей, а блок управління або відправляє ці коди, або повідомляє, що несправностей немає. У стандарті SAE J2012 заложена дуже інформативна система кодів несправностей ЕБК. Цю систему прийняло абсолютне більшість виробників автомобілів, як для OBD-II-протоколів, так і для власних OEM-протоколів.

Перший символ – альфа-вказівник DTC. Він вказує, в якому компоненті автомобіля виникла несправність.

Другий символ – показує, який код визначений. Код «0» («P0») – відкритий код несправності, визначений SAE. Код «1» («P1») – OEM-код, тобто визначений виробником. Код P0xxx означає одну й ту саму несправність для будь-якої марки автомобіля, що підтримує вимоги OBD-II. Проте розширені коди (P1xxx, P2xxx) розшифровуються по-різному для кожного автовиробника. Це, звичайно ж, створює проблеми для діагностики, адже розшифровка OEM-кодів під силу лише найбільш продвинутих OBD-II сканерам, і навіть найкращі прилади можуть не впоратися з цією задачею (дилерські прилади звичайно ж не враховуються).

Третій символ – позначає систему, в якій виявлена несправність.

Четвертий і п'ятий символи розглядаються разом. Це сам код помилки, тобто порядковий номер. Частіше за все ці коди відповідають помилкам зі старого стандарту OBD-I.

0x04. Очистка діагностичних кодів несправностей і збережених значень – цей режим дозволяє видалити всі діагностичні коди, збережені кадри поточних параметрів, результати тестування датчиків кисню і скинути статуси тестових моніторів.

В цьому режимі доступна функція очищення пам'яті блоку управління. Сканер видаляє навіть ті коди, які не може розшифрувати. Однак є проблема. З одного боку, деякі несправності блокують виконання тестових моніторів, що

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		14

означає, що якщо не згасити лампу "Check Engine", стираючи коди несправностей, деякі монітори ніколи не запуснуться і не завершаться. З іншого боку, при видаленні пам'яті ЕБК, крім кодів помилок, стираються "заморожені" кадри і вся інформація, накопичена під час роботи моніторів. Іншими словами, монітори обнуляються і знову ініціалізуються. Щоб усі монітори знову стали "завершеними", потрібно провести складні тестові поїздки.

0x05. Результати тестування, моніторинг датчиків кисню – зчитування і виведення результатів тестування датчиків кисню. Цей режим доступний тільки для не-CAN шини.

Цей режим дозволяє отримати майже всю інформацію про роботу датчиків кисню: амплітуду значень напруги на датчиках, час перемикавання з одного рівня на інший, рівні напруг переходу і т.д. Інформація стає доступною тільки після того, як відпрацює цей монітор повний цикл і отримає статус "завершений". Багато виробників використовують функції цього режиму в своїх OEM-протоколах, хоча відображається інформація не завжди в повному обсязі, що приходить з блоку управління.

0x06. Результати тестування, моніторинг інших компонентів/систем – зчитування і виведення результатів моніторингу непостійно тестованих систем і компонентів, таких як каталізатор, система рециркуляції вихлопних газів, система вентиляції паливного бака. Цей режим доступний тільки для CAN шини.

Цей режим надає доступ до результатів моніторингу (не до статусів моніторів, як в режимі 0x01) для непостійно тестованих систем, таких як каталізатор, системи поглинання випарів палива, система рециркуляції вихлопних газів, датчики кисню, підігрів датчиків кисню, система кондиціонування повітря, а також термостат системи охолодження і клапан системи вентиляції картера.

З назви режиму зрозуміло, що ці монітори працюють не завжди, а тільки за певних умов. Щоб вони отримали статус "завершених", потрібно здійснити складні тестові цикли. Параметри таких циклів можуть відрізнятися навіть у різних моделей тієї ж марки автомобіля.

0x07. Показати відкладені діагностичні коди несправностей, виявлені під час поточного або останнього їздового циклу – виведення результатів моніторингу постійно тестованих систем, поки виконуються умови тесту.

У цьому режимі виводяться результати моніторів постійно тестованих систем. Ці монітори активуються постійно, з моменту запуску двигуна до його

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		15

зупинки. Таких моніторів всього три: монітор компонентів, монітор системи паливної корекції/адаптації та монітор виявлення пропусків запалювання суміші.

Результат таких моніторів виводиться у формі діагностичних кодів несправностей, але лише якщо ці коди зареєстровані протягом одного їздового циклу (або циклу прогріву). Такі коди називаються "відкладеними". Якщо протягом 40-60 їздових циклів код не підтверджується, він видаляється з пам'яті ЕБУ. Але якщо код знову зареєстровано, він перестає бути "відкладеним" і зберігається в пам'яті блоку, що дозволяє отримати інформацію про несправність, використовуючи режим 0x03.

0x08. Управління операціями бортових компонентів/систем – цей режим дозволяє управляти різними виконавчими компонентами. Такі функції майже завжди підтримуються будь-якими заводськими протоколами. Однак різниця полягає в тому, що в протоколі OBD-II цей режим орієнтований на системи зменшення токсичності, наприклад, клапан системи рециркуляції вихлопних газів, клапан продувки адсорбера і т.д. Це дозволяє швидко перевірити роботу діагностованої системи без необхідності в тестових поїздках та моніторингу. Однак частіше для таких перевірок потрібне додаткове обладнання і/або спеціалізована інформація. Тому режим 0x08 ще не отримав широкого поширення.

0x09. Запит інформації про транспортний засіб – цей режим дозволяє отримати ідентифікаційні параметри діагностованого автомобіля: VIN-код, калібрувальні дані.

У цьому режимі виводяться такі параметри, як VIN-код автомобіля, код калібрування, завантаженого в постійний запам'ятовуючий пристрій, контрольна сума цього калібрування. Ця інформація може знадобитися з двох причин:

- оперативно відслідковувати застарілі або проблемні версії ПЗ та замінити їх на більш сучасні/нові;
- контролювати блок щодо втручання в калібрування.

Розрахунок контрольної суми здійснюється ЕБУ кожен раз після включення запалювання.

Виробники автомобілів не поспішають додавати цей режим у свої блоки управління. Навіть на нових автомобілях американського ринку інформація про ідентифікаційні параметри може підтримуватися не в повному обсязі або взагалі не підтримуватися.

						ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата			16



Таким чином, навіть коли протокол виробника автомобілів працює на діагностичному обладнанні коректно, завжди є сенс звертатися до електронного блоку управління автомобіля через протоколи OBD-II. У багатьох випадках можна розраховувати на додаткову інформацію, якої немає в OEM-протоколі або вона просто недоступна. Загалом, інтерфейс OBD-II значно розширив доступ звичайного автомобіліста до легкої та зручної діагностики систем автомобіля. Процес діагностики, в цілому, є процесом аналізу інформації. Чим ширше і різноманітніше зібрана інформація, тим більше ймовірність прийняти правильне рішення щодо ремонту автомобіля, що і є головним результатом.

## 1.2 Постановка завдання проектування

Основна мета розробки бакалаврської кваліфікаційної роботи є створення електронного пристрою для контролю і управління автомобілем на основі Arduino і полягає у втіленні протоколів обміну інформацією між бортовими комп'ютерами та відображенні цієї інформації на екрані мікроконтролера.

Добрий бортовий комп'ютер повинен мати можливість:

- Читання помилок
- Відображення інформації датчиків авто
- Можливість скидання помилок
- Рахувати розхід авто
- Попереджувати про критичні несправності авто

Бортовий модуль повинен мати змінні параметри при налаштуванні під авто, оскільки форсунки для палива на різних автомобілях, або навіть на автомобілях з різними двигунами, мають різну продуктивність. Це означає, що об'єм палива, який проходить через них, буде різним. Також блоки управління форсунками працюють на різних тактових частотах. Показники швидкості можуть змінюватися залежно від діаметра встановлених коліс. Наприклад, якщо встановити більші колеса, то показники швидкості будуть меншими, оскільки колесо більшого розміру здійснює менше обертів при тих самих обертах двигуна. Отже, показники з двох автомобілів однієї моделі можуть відрізнитися. Тому, якщо в бортовому комп'ютері не буде можливості вносити зміни в показники, то існує велика ймовірність, що дані будуть недостовірними.

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		17

Дипломна робота бакалавра повинна містити опис алгоритмів функціонування пристроїв керування автомобілем на основі Arduino та роботи пристроїв діагностики автомобілів, програмне забезпечення, структурні схеми функціональні та принципові схеми підключення

При розробці цього електронного пристрою необхідно враховувати наступні вимоги:

- 1) довговічність;
- 2) універсальність (робота за багатьма марками автомобілів)
- 3) простота схеми (мінімально можлива кількість компонентів);
- 4) довговічність;
- 5) низька собівартість

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		18

## 2. СИНТЕЗ І ОБҐРУНТУВАННЯ АЛГОРИТМУ РОБОТИ І СТРУКТУРНОЇ СХЕМИ ПРИСТРОЮ

### 2.1 Розробка алгоритму бортового модуля контролю і управлінням автомобілем на основі Arduino

Розробка алгоритму роботи є важливим етапом при проектуванні бортового модуля контролю і управлінням автомобілем. Це завдання зазвичай викликає труднощі під час реалізації. Розробку алгоритму варто здійснювати одночасно з розробкою структурної схеми пристрою. Алгоритм визначає ключові функції пристрою та послідовність виконання дій. Бортовий модуль повинен бути надійним та довговічним. Блок схема алгоритму показана на рисунку 2.1.

Після початку роботи, починається ініціалізація бортового модуля відбувається підготовка дисплея та встановлення Bluetooth-з'єднання. Після успішного з'єднання модуль надсилає запит через Bluetooth для отримання необхідних даних з автосканера. Отримані дані розшифровуються для подальшого використання. Наступним кроком є перевірка температури двигуна з отриманих даних. Якщо температура перевищує максимально допустиму, на дисплеї виводиться повідомлення про високу температуру двигуна. Якщо температура в межах норми, дані відображаються на дисплеї екрану 1 для користувача. Після цього перевіряється стан кнопок, якщо натиснута кнопка 1, відбувається перемикання на наступний екран і відповідні дані виводяться на дисплей; якщо натиснута кнопка 2, відбувається повернення до попереднього екрану з виведенням даних на дисплей. Після обробки всіх даних і натискання кнопок перевіряється, чи завершена робота модуля. Якщо роботу не завершено, алгоритм повторюється знову з початку основного циклу. Якщо роботу завершено, мікроконтролер переходить у режим низького енергоспоживання до наступного циклу. Цей алгоритм забезпечує ефективну роботу бортового модуля, дозволяючи контролювати важливі параметри автомобіля та відображати їх на дисплеї в режимі реального часу.

									Лист
									19
Вим.	Лист	№ докум.	Підпис	Дата					

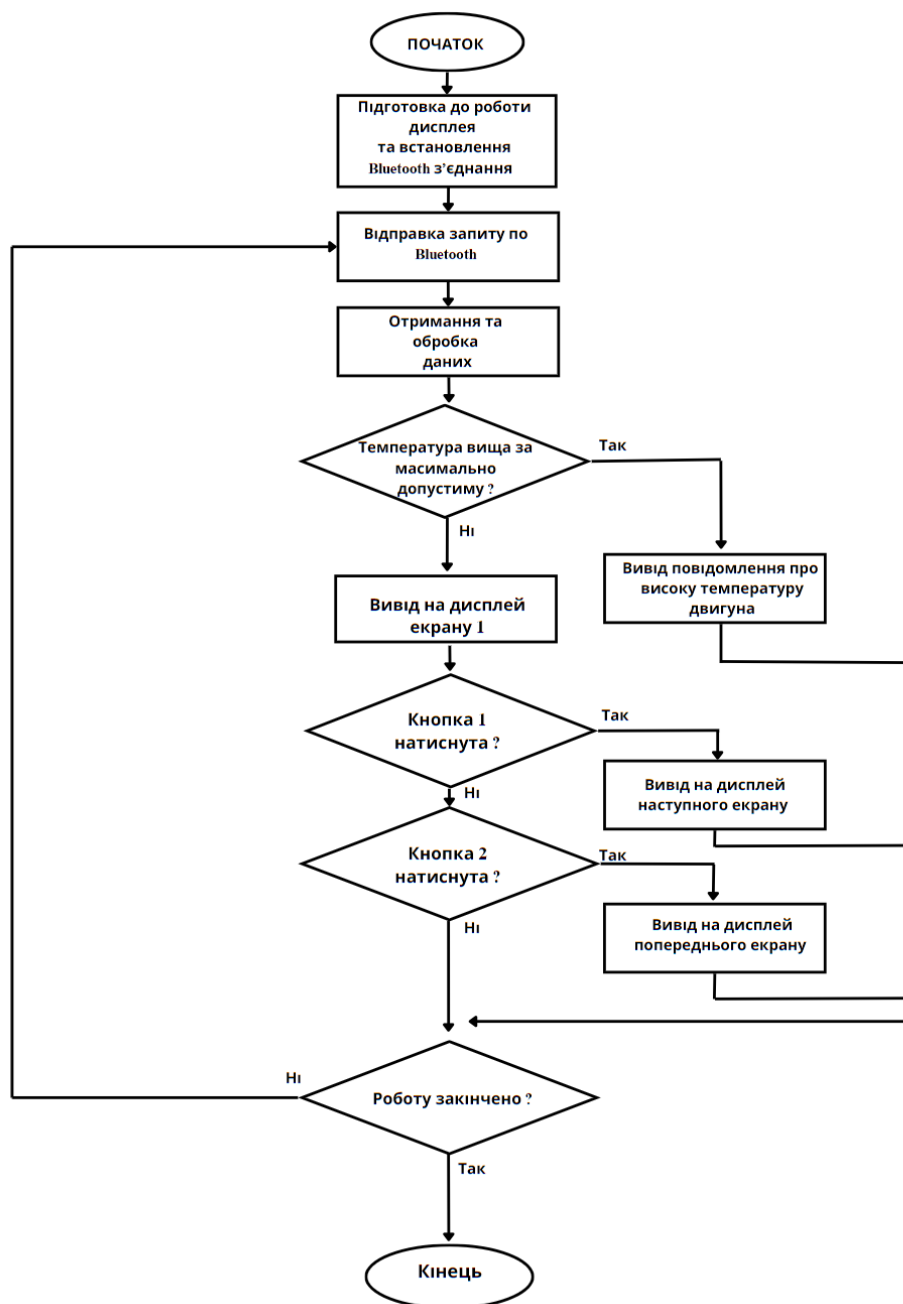


Рисунок 2.1 - Алгоритм роботи бортового модуля контролю і управлінням автомобілем на основі Arduino

## 2.2 Розробка структурної схеми

Згідно з розробленим алгоритмом роботи, ми створимо структурну схему бортового модуля. Ця схема складається з основних блоків, кожен з яких виконує визначені функції, і зв'язків між ними. Зв'язки слід позначати між блоками, які взаємодіють в процесі роботи пристрою.

Структурну схему зображено на рисунку 2.2

Схема складається з пристрою управління та розрахунку, блоку кнопок, блоку звукової індикації, дисплея, Bluetooth модуля та автосканера.

- Автосканер

Для того, щоб отримати інформацію від автомобіля, потрібен прилад, який зчитує дані з його електронних систем. Таким приладом є автосканер який підключається до діагностичного роз'єму автомобіля і дозволяє надсилати запит на отримання даних авто за допомогою спеціальних команд, ці дані надходять у шістнадцятковій системі числення, а також зашифрованими.

- Блок кнопок

Цей блок нам потрібен зокрема для, перемикання між «екранами» тобто для того щоб змінювати які саме параметри відображаються.

- Блок звукової індикації

Блок звукової індикації потрібен щоб звернути увагу водія, коли показання параметра температури виходить за межу критично допустимої

- Пристрій управління та розрахунку

В нашому випадку за управління та обчислення параметрів у схемі відповідає arduino. А саме він відповідає за надсилання команд через Bluetooth модуль на автосканер, отриману інформацію він розшифровує та виводить у зрозумілій формі на дисплей. Також приймає та обробляє команди з блоку кнопок для зміни інформації що відображається.

- Дисплей

В даному модулі буде використано LCD дисплей, Він буде відображати дані які було прийнято і оброблено від автосканера.

						ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата			21



Рисунок 2.2 - Структурна схема бортового модуля контролю і управління автомобілем на основі Arduino

Реалізація пристрою що проектується потрібно здійснювати на основі функціонально завершених блоків , які надійно виконують поставлені задачі

Блоки що входять до складу пристрою , мають узгоджено працювати в одній системі .

### 3. РОЗРОБКА ФУНКЦІОНАЛЬНОЇ СХЕМИ

#### 3.1 Розробка функціональної схеми пристрою

Наступним завданням після розробки структурної схеми є створення функціональної схеми пристрою. Функціональна схема є прообразом, на основі якого буде побудована принципова схема пристрою.

Приклад функціональної схеми на рисунку 3.1

Основою схеми є платформа Arduino, в якій використовується мікроконтролер, зазвичай це ATmega328P. Принцип роботи Arduino включає наступні етапи: програмування, завантаження, виконання, взаємодія та живлення.

Спочатку використовується середовище розробки Arduino IDE для написання коду (скетчу) на мові програмування, яка базується на C/C++. Після написання програми вона завантажується на мікроконтролер через USB-кабель. Програма, вбудований у плату Arduino, перетворює дані з комп'ютера у формат, зрозумілий для мікроконтролера.

Після завантаження програми мікроконтролер починає виконувати інструкції. Програма може взаємодіяти з різними периферійними пристроями через входи та виходи мікроконтролера. Arduino може зчитувати дані з датчиків (наприклад, температури, світла, вологості) через аналогові або цифрові входи. На основі отриманих даних мікроконтролер може виконувати відповідні дії, наприклад, вмикати світлодіоди, двигуни, реле або надсилати дані на комп'ютер.

Плата Arduino може живитися від USB-кабеля або зовнішнього джерела живлення (акумулятор, адаптер), що дозволяє використовувати її у автономних проектах.

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		23





## 4. РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ

### 4.1 Вибір елементної бази

При виборі контролера який буде обробляти інформацію . Було вирішено обрати Arduino Nano . Цей вибір був зроблений з кількох причин:

1. Велика популярність платформи , зручність використання , а також простота мови програмування ;
2. Можливість програмувати без програматора ;
3. Велика кількість бібліотек для роботи з модулями .

Існує велика кількість платформ Arduino , але я обрав Arduino Nano 3.0 , тому що , ця версія має малі габарити , має достатню кількість контактів , для мого пристрою .

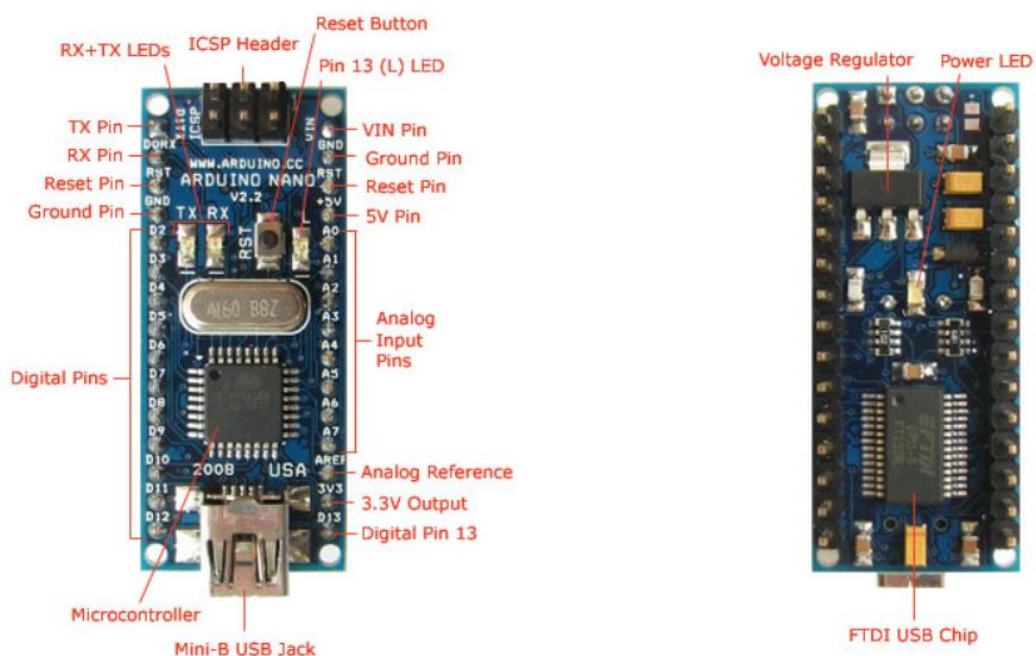


Рисунок 4.1 - Arduino Nano

Цю плату розроблено на базі мікроконтролера ATmega328P . Живлення плати можливе через USB , або через (пін «VIN» ) , в цьому випадку напруга має бути в межах 6-12 В , оскільки плата має вбудований стабілізатор напруги , а також можливе живлення плати від 5В (пін «5V»)

Технічні характеристики:

- Мікроконтролер: ATmega328P;
- Робоча напруга: 5В;

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		25

- Тактова частота: 16МГц;
- Вхідна напруга живлення: 6-12В;
- Цифрові входи/виходи: 14 (6 з них ШІМ);
- Аналогові входи: 6;
- Максимальний струм одного виводу: 40мА;
- Максимальний струм одного виводу (3.3В): 50мА;
- Інтерфейси: I2C/TWI, SPI, UART, PWM;
- USB-перетворювач: CH340G;
- Flash-пам'ять: 32Кб;
- Оперативна пам'ять (SRAM): 2Кб;
- EEPROM-пам'ять: 1Кб;
- Розмір: 44.6x17.8x7мм;
- Вага: 6г.

#### Силові виводи:

- VIN – Вхід для подачі живлення від зовнішнього джерела.
- 5V – Регульоване джерело напруги. Використовується для живлення мікроконтролера та компонентів на платі.
- 3V3 – Джерело напруги 3,3 В. Генерується вбудованим регулятором на платі.

#### Деякі з 14 цифрових виводів мають особливі функції:

- послідовна шина: виводи 0 (RX) і 1 (TX) використовуються для прийому та передачі TTL-даних;
- зовнішнє переривання: виводи 2 і 3 можна налаштувати для виклику переривання;
- ШІМ: виводи 3, 5, 6, 9, 10, 11 забезпечують ШІМ з роздільною здатністю 8 біт;
- SPI: через виводи 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) здійснюється SPI-зв'язок;
- LED: до виводу 13 підключений вбудований світлодіод. Якщо на виводі високий потенціал, світлодіод світиться.

Платформа Arduino Uno має кілька пристроїв для здійснення зв'язку з комп'ютером, іншими пристроями Arduino та мікроконтролерами. Мікроконтролер ATmega328 підтримує послідовний інтерфейс UART TTL.

						ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата			26

Мікроконтролер ATmega8U2 спрямовує цей інтерфейс через USB для зв'язку плати з ПК.

Платформа програмується за допомогою ПО Arduino. ATmega328 поставляється з завантажувачем, що спрощує запис нових програм, не потрібно використовувати зовнішні програматори.

Існує можливість не використовувати завантажувач, а запрограмувати мікроконтролер через внутрішньосхемне програмування (вивід ICSP).

Бібліотека SoftwareSerial дозволяє створити послідовний прийом або передачу даних через будь-який цифровий вивід платформи Arduino. Ця можливість є рішенням наступної проблеми: дані з автомобіля надходять через Bluetooth-інтерфейс, і їх потрібно якось приймати на Arduino.

Є дві обов'язкові функції в будь-якій програмі для Arduino:

- `setup()`: Викликається при старті скетчу. Використовується для ініціалізації змінних, визначення режимів роботи виводів, запуску бібліотек тощо. Виконується лише один раз після кожного увімкнення живлення або скидання плати Arduino.
- `loop()`: Виконується після `setup()` і повторюється в циклі, дозволяючи програмі виконувати обчислення та реагувати на них. Використовується для активного керування платою Arduino.

Для прийому даних з автомобіля на плату Arduino Nano необхідний Bluetooth-модуль, спеціально розроблений для Arduino, щоб спростити встановлення зв'язку між цими пристроями. Для Arduino існує безліч різних Bluetooth-модулів, але я обрав модуль HC-05, оскільки це сучасна версія, яка є широко доступною, має низьку вартість, а також легко підключається і налаштовується, а головне що нам потрібно, це те що даний модуль підключається в режимі «Master».

						ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата			27





що дозволяє легко читати символи при різних умовах освітлення, а також простоту використання, що дозволяє швидко інтегрувати його в різні проекти.

I2c конвертер дозволяє підключити LCD-1602 до мікроконтролера через інтерфейс i2c, що значно зменшує кількість необхідних для підключення проводів. Основні переваги використання i2c конвертера: мінімізація проводів (замість 16 проводів для підключення дисплея безпосередньо, з i2c конвертером потрібно лише 4 проводи) і зручність керування, адже i2c інтерфейс дозволяє легко керувати дисплеєм через програмні бібліотеки, що зменшує складність програмування.

Загалом, поєднання LCD-1602 та i2c конвертера робить наш бортовий комп'ютер ефективним і та надійнішим . Фото конвертера та дисплею представлено на рисунку 4.3 та 4.4 відповідно .

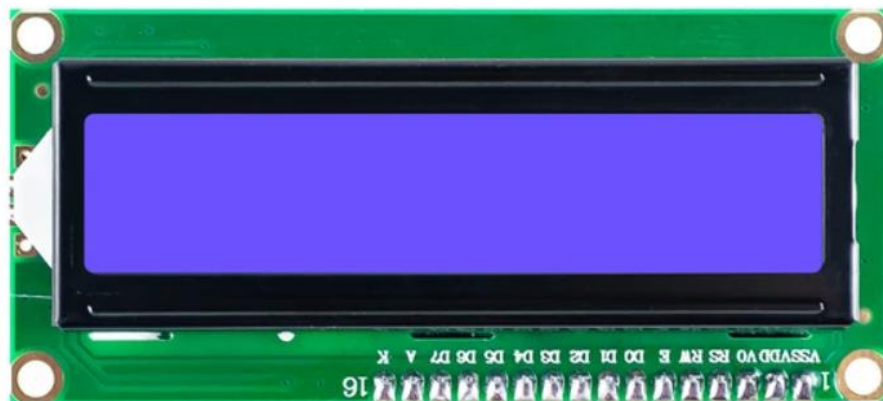


Рисунок 4.3 - LCD-1602

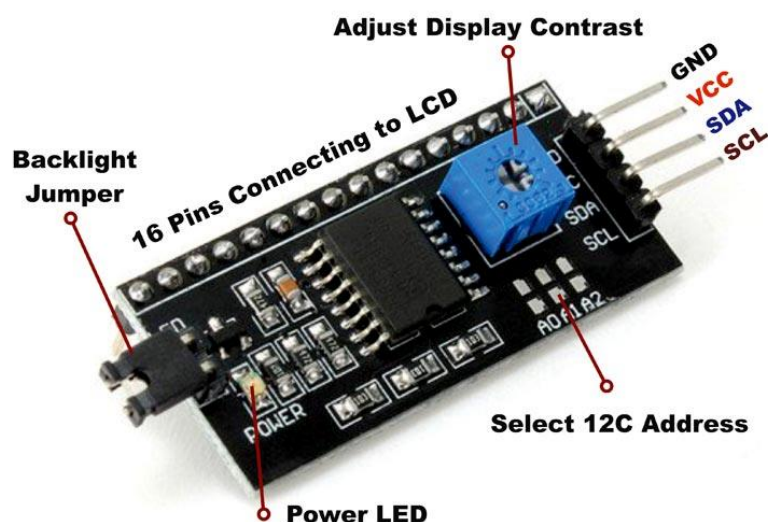


Рисунок 4.4 Модуль перехідник LCD1602 ІІС / І2С

Технічні характеристики дисплея :

- Розміри: 80 x 36 мм
- Робоча температура: 0 ~ 50°C
- Підсвітка: блакитна
- Колір символів: білий
- Розмір символу: 4.35 x 2.95 мм
- Формат: 16 x 2
- Розміри точки: 0.5 x 0.5 мм
- Інтерфейс: HD44780
- Видима область: 64.5 x 13.8 мм
- Живлення: 5В

Підключення до Arduino :

- GND – земля.
- VCC – живлення 5В.
- SCL – A4
- SDA – A5

Також нам потрібен автосканер, який буде зчитувати інформацію з блока управління авто та передавати її на бортовий модуль. Це дозволить отримувати дані про стан автомобіля в режимі реального часу, що є надзвичайно важливим для забезпечення безпеки та ефективності експлуатації транспортного засобу.

Для вибору сканера були визначені декілька основних критеріїв. Перш за все, сканер повинен підтримувати інтерфейс OBD-II, який є стандартом для всіх сучасних автомобілів і дозволяє отримувати широкий спектр діагностичних даних. По-друге, ціна пристрою повинна бути доступною, щоб його придбання не завдало значних фінансових витрат. І нарешті, надійність – важливий фактор, адже сканер повинен працювати стабільно та безвідмовно протягом тривалого часу.

Враховуючи всі ці вимоги, ми обрали діагностичний адаптер ELM327. Цей адаптер зарекомендував себе як надійний і доступний засіб для зчитування та передачі діагностичної інформації з автомобіля. Він підтримує інтерфейс OBD-II і здатен працювати з більшістю автомобілів, що робить його універсальним рішенням для різних потреб.

									Лист
									31
Вим.	Лист	№ докум.	Підпис	Дата					



Рисунок 4.5 - ELM 327

Назва адаптера ELM327 походить від назви мікросхеми PIC18F2480, яка є контролером інтерфейсу периферії від компанії "Microchip", з програмним забезпеченням від канадської компанії "ELM Electronics". Цей адаптер призначений для перетворення сигналів між бортовим діагностичним портом OBD-II і стандартним портом RS-232 персонального комп'ютера.

RS-232 є стандартом фізичного рівня для асинхронного інтерфейсу, розробленим Асоціацією Електронної Промисловості (EIA). Цей стандарт також відомий як послідовний порт ПК або COM-порт.

Мікросхема ELM327 може автоматично розпізнавати і конвертувати найбільш поширені протоколи, що використовуються сьогодні для зв'язку мережі автомобіля та діагностичного сканера. Для повного функціоналу ELM327 необхідні деякі зовнішні компоненти.

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		32





буде вимкнена за замовчуванням. Функцією пам'яті можна завжди керувати за допомогою AT-команд M0 і M1.

– Baud Rate (Контакт 6) Цей вхід контролює швидкість передачі даних через інтерфейс RS-232. Якщо подати логічну одиницю під час увімкнення живлення або скидання, то швидкість передачі даних буде встановлено на 38400 бод. Якщо логічний нуль, то швидкість буде 9600 бод.

– LFmode (Контакт 7) Цей вхід використовується для вибору режиму переходу на новий рядок, який буде використовуватися після увімкнення живлення або скидання системи. Логічна одиниця, подана на цей вхід, означає, що повідомлення, відправлене ELM327, буде завершуватися символом переводу каретки і рядком. Логічний нуль - повідомлення будуть завершуватися лише переводом каретки. Поведінку можна змінити за допомогою AT-команд L0 і L1.

– VSS (Контакти 8, 19) Загальна ланка повинна бути підключена до цих виводів.

– XT1 і XT2 (Контакт 9 і 10) Кварцовий генератор 4 МГц підключається між цими двома контактами. Конденсатори, згідно з вимогами кристала (зазвичай по 27 пФ кожен), також повинні бути підключені між кожним з цих виводів і землею VSS.

– VPW In (Контакт 11) Активний високий вхід для J1850 VPW даних. Повинен бути на низькому рівні в стані спокою (шина неактивна).

– ISO In (Контакт 12) Це активний низький вхід для ISO 9141 і ISO 14230 даних. Повинен бути на високому рівні в стані спокою (шина неактивна).

– PWM In (Контакт 13) Це активний низький вхід для J1850 PWM даних. Повинен бути на високому рівні в стані спокою (шина неактивна).

– J1850 Bus- (Контакт 14) Цей активний високий вихід використовується для підняття від'ємної шини J1850 до активного рівня. Якщо не використовується, то можна залишити вивід роз'єднаним.

– RTS (Контакт 15) Активний низький вхід "запит передачі", використовується для обробки переривань, щоб надіслати нову команду.

– Busy (Контакт 16) Активний високий вихід, показує поточний стан ELM327. Якщо він знаходиться на низькому рівні, то процесор готовий приймати команди або ASCII-символи. Якщо на високому - команди обробляються.

– RS232Tx (Контакт 17) Вихід передачі даних RS-232. Рівень сигналу сумісний з більшістю інтерфейсних мікросхем.

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		34

– RS232Rx (Контакт 18) Це вхід отримання даних RS-232. Рівень сигналу сумісний з більшістю мікросхем інтерфейсів, а також може використовуватися і з іншими інтерфейсами, оскільки на вході є спусковий Шмітта.

– VDD (Контакт 20) Вивід є позитивним контактом живлення.

– ISO K і ISO L (Контакт 21 і 22) Активні високі вихідні сигнали, які використовуються для управління ISO 9141 і ISO 14230 шинами. Багато нових автомобілів не вимагають L-Line, тому можна залишити контакт 22 роз'єднаним.

– CAN Tx і CAN Rx (Контакт 23 і 24) Вхід і вихід сигналів CAN-інтерфейсу. Якщо вони не використовуються, то вивід 24 повинен бути з'єднаним з VDD.

– RS232 Rx LED, RS232 Tx LED, OBD Rx LED, OBD Tx LED (Контакти 25-28) Ці чотири виходи зазвичай знаходяться на високому рівні. Вони переходять до низького рівня під час передачі або отримання даних ELM327. Виходи можуть бути використані для керування світлодіодами напряму через обмежувальні резистори або через іншу логіку для звітів про стан. Якщо вони не використовуються, ці виводи слід залишити роз'єднаними.

Переваги використання мікросхеми ELM327: – автоматичне визначення діагностичного протоколу; – низьке споживання енергії; – наявність режиму очікування; – швидкість передачі через RS-232 до 500 кбіт/с.

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		35



## 5. НАЛАШТУВАННЯ ПРИСТРОЮ

Після розробки всіх схем , та вибору елементної бази , бортовий модуль потрібно налаштувати .

Налаштування потрібно почати з Bluetooth модуля HC-05:

Спочатку заливаємо скетч для налаштування модуля **Додаток А**

```
1. #include <SoftwareSerial.h>
2.
3. //SoftwareSerial BTSerial(5, 6); // RX | TX
4. SoftwareSerial BTSerial(10, 11); // RX | TX
5.
6. void setup()
7. {
8.   Serial.begin(9600);
9.   Serial.println("Enter AT commands:");
10.  BTSerial.begin(38400); // HC-05 default speed in AT command mode
11. }
12.
13. void loop()
14. {
15.   if (BTSerial.available()) // read from HC-05 and send to Arduino Serial
16.     Serial.write(BTSerial.read());
17.
18.   if (Serial.available()) // Keep reading from Arduino Serial Monitor and
19.     BTSerial.write(Serial.read());
20. }
```

Рисунок 5.1 - Приклад коду для налаштування Bluetooth модуля

Підпаюємо Arduino до відповідних пінів Bluetooth: (Схему з підключенням зображено на рисунку 5.2.

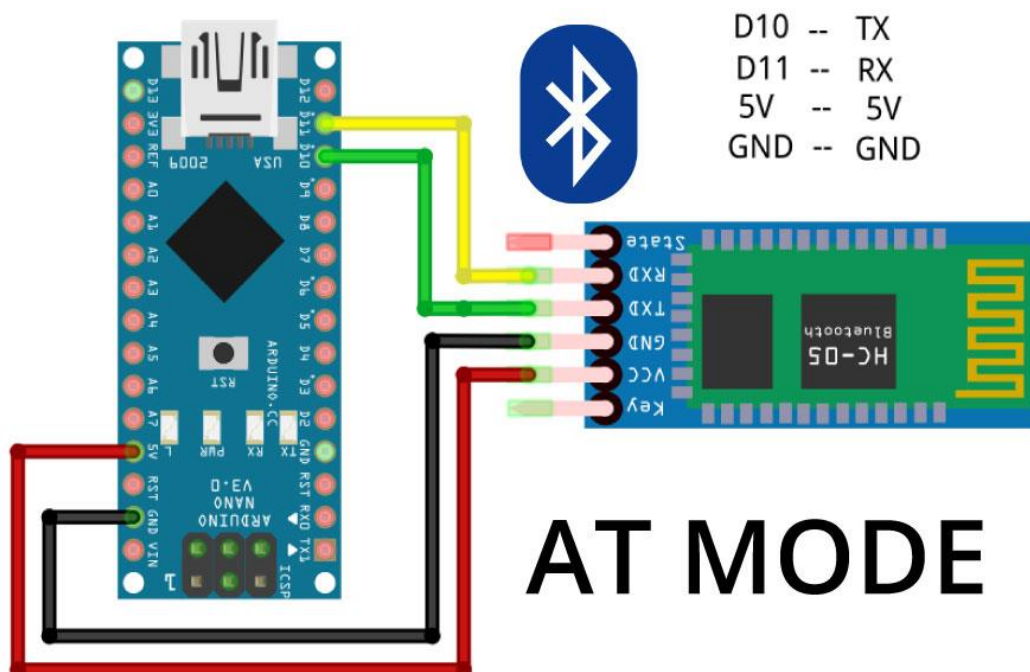


Рисунок 5.2 - Схема підключення модуля для налаштування .

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		37

Отже, робимо з'єднання, як показано на схемі вище.

- D10 Arduino до TX модуля Bluetooth
- D11 Arduino до RX модуля Bluetooth
- 5V Arduino до 5V модуля Bluetooth
- GND Arduino до GND модуля Bluetooth

При налаштуванні модуля потрібно натиснути кнопку KEY щоб ввести модуль в режим програмування .

Після успішного завантаження скетчу відкриваємо: Сервіс -> Монітор порта. Далі внизу встановлюємо швидкість 9600 бод і NL+CR разом.

Потім вводимо команди по одній і натискаємо [Надіслати]. Після кожного введення має бути відповідь.

AT // (можливо 1 раз буде Error, повторюємо знову)

AT+NAME=Car // Присвоюємо ім'я модулю Car

AT+ROLE=1 // Переводимо модуль в режим Майстер

AT+PSWD=1234 // Встановлюємо пароль 1234 як на OBD ELM327

AT+BIND=AABB,CC,112233 // Прописуємо Mac-адресу OBD ELM327.

AT+CMODE=1 // Підключення модуля з фіксованою адресою

Зверніть увагу, що MAC-адреса виду: «AA:BB:CC:11:22:33» вводиться як «AABB,CC,112233». MAC-адресу свого модуля ELM327 можете подивитися, підключившись спочатку до нього зі свого мобільного. (Стандартні паролі зазвичай: 1234, 6789, 0000).

Все, налаштування модуля Bluetooth завершено.

Тепер потрібно зібрати схему Arduino, Bluetooth, LCD-екран.

Почнемо з підключення модуля Bluetooth HC-05:

- TX модуля під'єднуємо до 7 піну (Rx) Arduino (саме TX до RX, не так, як раніше);
- RX модуля під'єднуємо до 5 піну (Tx) Arduino;
- 5V пін модуля під'єднуємо до піну 5V Arduino;
- GND пін модуля під'єднуємо до GND Arduino;

Підключаємо LCD-дисплей:

- GND дисплея до GND Arduino;
- VCC дисплея до 5V Arduino;
- SDA дисплея до A4 Arduino

										Лист
										38
Вим.	Лист	№ докум.	Підпис	Дата						

Потрібно налаштувати контрастність екрану за допомогою резистора на I2C конвертері

Підключаємо додаткову кнопку для перемикання екранів :

- Перша кнопка: один кінець нормально-відкритої кнопки під'єднуємо до GND Arduino, а другий кінець до пін 10.
- Друга кнопка: GND + пін 9.
- Біпер для звукових попереджень під'єднуємо за наступною схемою: "+" до пін 13, а "-" до GND Arduino.

Завантаження скетчу в Arduino за допомогою Arduino IDE

### Додаток Б

Потрібно змінити значення змінних відповідно до об'єму вашого автомобіля. Скетч написаний так, щоб було зрозуміло кожному, і будь-хто міг змінити те, що захоче.

Також необхідно обов'язково врахувати змінні:

1. ED=1.398. Наприклад, об'єм двигуна в літрах 1.398;
2. tcorrect=1.014 (калібрування часу).
3. speed\_korrek\_val (калібрування показів швидкості)

Можливо, Arduino буде неточно рахувати час за допомогою команди millis(). Тут потрібно вручну коригувати значення. На екрані «технологічний 2» буде вказаний час у секундах: наприклад, time\_start: 23. Порівняйте значення з реальним. Наприклад, засічіть 10 хвилин, коли значення time\_start буде, наприклад, рівне 23. І вийде, що через 600 секунд реально покаже 605. Тобто  $623 - 605 = 18$  секунд відставання. Поправочний коефіцієнт tcorrect буде рівний  $623/605 = 1,02975$ .

4. Управління
  - [Кнопка 1], [Кнопка 2] — перемикання екранів вперед-назад.

									Лист
									39
Вим.	Лист	№ докум.	Підпис	Дата					

## 6. ВИСНОВКИ

У результаті виконання дипломної роботи було розроблено бортовий модуль контролю і управління автомобілем на основі Arduino. Система складається з мікроконтролера Arduino, модулю Bluetooth HC-05, адаптера ELM327 для зчитування даних з OBD-II, а також інтерфейсного дисплея LCD 1602.

Ми розробили алгоритм роботи цього пристрою та створили структурну схему. Створено та розраховано функціональні та принципові схеми всіх компонентів пристрою. Розроблено та представлено програму роботи для забезпечення надійної роботи бортового модуля .

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		40



## НАУКОВІ ПРАЦІ СТУДЕНТА

Кулик І.А., доцент, Шевченко М.С., асистент;  
Киричко С.В., ЕС-01, студент  
Сумський державний університет, Суми, Україна

На сьогоднішній день актуальним є питання підвищення відмовостійкості, якості та багатофункціональності управління транспортними засобами, зокрема автомобілем. З цією метою пропонується до розгляду проектування та впровадження бортового модуля для контролю і управління автомобілем на базі поширеної платформи Arduino. Застосований модуль є спеціально адаптованим для використання з k-line шинами, які є стандартом у багатьох бюджетних та старих моделях автомобілів. Головними завданнями використання модуля для контролю і управління автомобілем на базі платформи Arduino є забезпечення підключення до шин автомобіля, здійснення моніторингу показань датчиків, можливість скидання помилок та ряд інших необхідних операції для якісного управління.

Принцип роботи бортового модуля для контролю і управління автомобілем на базі платформи Arduino полягає у взаємодії з k-line шинами автомобіля для отримання даних від різних систем транспортного засобу. Bluetooth модуль HC-05 призначений для зв'язку з ELM сканером. Бортовий модуль для контролю і управління автомобілем на базі платформи Arduino попередньо програмується для налагодження з'єднання з ELM, що дозволяє отримувати дані від електронного блока управління автомобіля.

Для встановлення з'єднання та можливості отримання інформації від електронного блока управління автомобіля, бортовий комп'ютер повинен надіслати наступні команди: ATZ, ATSP5, ATSH8110F1, ATFI, 2101. Після отримання відповіді від ELM у вигляді рядка даних у шістнадцятковій системі числення Bluetooth модуль HC-05 передає ці дані на блок управління. Платформа Arduino, в свою чергу, декодує рядок даних, перетворює його у десяткову систему числення та виводить на екран LCD 1602. Цей процес повторюється з максимальною швидкістю, щоб забезпечити мінімальну затримку та миттєвий доступ до інформації щодо стану керованого автомобіля.

						ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата			41

## 7. СПИСОК ЛІТЕРАТУРИ

1. Монастирьов К. С. Архітектура пристроїв збору даних на базі мікроконтролерів Arduino : робота на здобуття кваліфікаційного ступеня магістра : спец. 171 - електроніка / наук. кер. А. М. Логвинов. Суми : Сумський державний університет, 2023. 36 с.

2. Марченко, В.В. Голосовий інформатор автомобіля [Текст] / В.В. Марченко, А.І. Новгородцев // Фізика, електроніка, електротехніка: матеріали та програма науково-технічної конференції, м. Суми, 5-9 лютого 2019 р. / Відп. за вип. С.І. Проценко. – Суми: СумДУ, 2019. – С. 99.

3. Коваль, Р.М. Мікроконтролерна система керування двигуном [Текст]: робота на здобуття кваліфікаційного рівня магістра; спец.: 171 – електроніка / Р.М. Коваль; наук. керівник І.А. Кулик. – Суми: СумДУ, 2020. – 87 с.

4. Рибіцький О. М. Дослідження методів комп'ютерної діагностики автомобіля за технологією OBD-2 та використання її в інформаційній системі : пояснювальна записка до атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 121 — Інженерія програмного забезпечення / О. М. Рибіцький ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2021. – 76 с.

5. Падалка О. С. Бортова діагностична система транспортного засобу / О. С. Падалка, науковий керівник – ст. викл. Карнаушенко В. П. // Радіоелектроніка та молодь у XXI столітті : матеріали 25-го Міжнар. молодіжн. форуму, 20-22 квітня 2021 р. – Харків : ХНУРЕ, 2021. – Т. 1. – С. 55–56.

6. Задрикін А. О. Розробка нечіткої системи прогнозування витрат палива автомобіля на основі даних про стиль водіння та параметри дороги / А. О. Задрикін // Радіоелектроніка та молодь у XXI столітті : тези доповідей 27-го Міжнародного молодіжного форуму, 10–12 травня 2023 р. – Харків : ХНУРЕ, 2023. – Т. 7. – С. 129–130.

7. Познахор, Д. О. Бортовий комп'ютер на мікроконтролері для автомобіля з двигуном внутрішнього згорання : дипломний проект ... магістра : 6.050902 Радіоелектронні апарати / Познахор Дмитро Олександрович. - Київ, 2019. - 51 с.

						ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата			42

8. Острянюк, О. В. Пристрій для перевірки електронних систем автомобіля / Острянюк О. В. // Електронна та Акустична Інженерія : науково-технічний журнал. – 2020. – Т. 3, № 2. – С. 25-28.

9. Желяскова, А. З. Цифрова система керування швидкістю автомобіля : індивідуальний дослідницький проєкт ... бакалавра : 126 Інформаційні системи та технології / Желяскова Анастасія Захарівна. – Київ, 2022. – 73 с.

10. Д'яченко С. Ф. Бездротовий контроль температури за допомогою Bluetooth технології з використанням мікропроцесорної техніки / С. Ф. Д'яченко, науковий керівник – проф. Новоселов С. П. // Радіоелектроніка та молодь у XXI столітті : матеріали 25-го Міжнар. молодіжн. форуму, 20-22 квітня 2021 р. – Харків : ХНУРЕ, 2021. – Т. 2. – С. 29–30.

11. Бондаренко І. М. Мікропроцесорні системи контролю та керування : навч. посібник / І. М. Бондаренко, О. В. Бородин, В. П. Карнаушенко ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків : ХНУРЕ, 2020. – 244 с.

12. Горелов Д. О. Raspberry пі як ефективний й універсальний мозок будь-якої системи / Д. О. Горелов // Радіоелектроніка та молодь у XXI столітті : матеріали XXIV Міжнар. молодіжн. форуму, 7–9 квітня 2020 р. – Харків : ХНУРЕ, 2020. – С. 134–135.

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		43

## ДОДАТОК А

Скетч для налаштування Bluetooth модуля HC-05

```
#include <SoftwareSerial.h>
//SoftwareSerial BTSerial(5, 6); // RX | TX
SoftwareSerial BTSerial(10, 11); // RX | TX
void setup()
{
  Serial.begin(9600);
  Serial.println("Enter AT commands:");
  BTSerial.begin(38400); // HC-05 default speed in AT command mode
}
void loop()
{
  if (BTSerial.available()) // read from HC-05 and send to Arduino Serial Monitor
    Serial.write(BTSerial.read());
  if (Serial.available()) // Keep reading from Arduino Serial Monitor and send to HC-05
    BTSerial.write(Serial.read());
}
```

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		44

## ДОДАТОК Б

Скетч для Arduino( програма бортового модуля)

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <EEPROM.h>
#define RxD 5          //Arduino pin connected to Tx of HC-05
#define TxD 7          //Arduino pin connected to Rx of HC-05
LiquidCrystal_I2C lcd(0x3F, 16, 2);
SoftwareSerial blueToothSerial(RxD,TxD);
// Коректувати -----
double tcorrect=1.014;
double speed_korrek_val=1.1;
double ED = 1.399;
//-----
// Не змінюємо -----
byte count_display=9 ;
byte pin=10, pin2=9, pin3=8;
int engine_on_rpm=600;
byte delay_var=250;
//-----
boolean off2=true;
char rxData[129];
char rxIndex=0;
byte selmon, off, value, value2, value3 , t1, kol_check_engine_error_val, fss_val;
int speed_error, tmp_error, rpm_error, dvk_error, iat_error, rnd_error, dts_error,
uoz_error, maf_error, pdz_error, ut_error, tm_error;
long dvk_var, intake_air_temp_var, davlenie_topliva_var, tmp_masla_var, speed_var,
tmp_var, t2, check_engine_km, check_engine_flash;
unsigned long time_new, time_old, time_old_gurnal;
double rpm_var, uoz_var, raschet_nagruzka_dvigatelya_var, maf_var,
polozh_dross_zaslona, uroven_topliva_var, IMAP, MAF,
FuelFlowGramsPerSecond,FuelFlowLitersPerSecond,time,
```

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		45



```

blueToothSerial.println("ATZ");
delay(1000);
blueToothSerial.println("ATSP5");
delay(1500);
  blueToothSerial.println("ATSH8110F1"); //rezet
delay(1000);
blueToothSerial.println("ATFI");
delay(delay_var);
blueToothSerial.println("ATZ");
delay(1000);
blueToothSerial.println("ATSP5");
delay(1500);
  blueToothSerial.println("ATSH8110F1"); //rezet
delay(1000);
blueToothSerial.println("ATFI");
delay(200);
lcd.clear();
lcd.createChar(1, grad);
}
void loop()
{
  send2101();
  if (tmp_var>102) {
    lcd.clear();
    lcd.setCursor(4,0);
    lcd.print("WARNING!");
    lcd.setCursor(3,1);
    lcd.print("TEMP:");
    lcd.setCursor(8,1);
    lcd.print(tmp_var); //Температура
    lcd.print(char(223));
    lcd.print("C");
    tone(13, 3000, 500);
  } else {

```

										Лист
										47
Вим.	Лист	№ докум.	Підпис	Дата						

```

value=digitalRead(pin); // Перевірка кнопок
value2=digitalRead(pin2);
if (value==LOW) {
  if (selmon==count_display) {
    lcd.clear();
    selmon=0;
  } else {
    lcd.clear();
    selmon++;
  }
  delay(150);
}
if (value2==LOW) {
  if (selmon==0) {
    lcd.clear();
    selmon=count_display;
  } else {
    lcd.clear();
    selmon--;
  }
  delay(150);
}
switch (selmon) {
case 0:
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(speed_var); // Швидкість
  lcd.print(" km/h");
  lcd.setCursor(10,0);
  lcd.print(tmp_var);
  lcd.print(char(223));
  lcd.print("C ");
  lcd.setCursor(0,1);
  lcd.print(int(rpm_var)); //Оберти

```

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		48



```

lcd.print(" rpm");
lcd.setCursor(10,1);
lcd.print(uoz_var); //Запалювання
lcd.print(" V");

break;
case 1:
lcd.clear();
lcd.setCursor(0,0);
if (odometr>0.1) {
  lcd.print(LP100);
} else {
  lcd.print("-.--");
}
lcd.print(" L");
lcd.setCursor(10,0);
lcd.print(intake_air_temp_var); //Температура
lcd.print(char(223));
lcd.print("C ");
lcd.setCursor(0,3);
lcd.print(odometr);
lcd.print(" km");
lcd.setCursor(10,3);
lcd.print(LPH);
lcd.print(" L/H");
break;
case 2:
lcd.setCursor(0,0);
  lcd.print(selmon);
  lcd.print(".");
  lcd.print(" L/100");
  lcd.setCursor(9,0);
  lcd.print(LP100);
  lcd.setCursor(3,1);

```

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
Вим.	Лист	№ докум.	Підпис	Дата		49

```

    lcd.print("L/H");
    lcd.setCursor(9,1);
    lcd.print(LPH);
    break;
case 3:
    lcd.setCursor(0, 0);
    lcd.print(selmon);
    lcd.print(".");
    lcd.print("Temp");
    lcd.setCursor(6, 0);
    lcd.print(" ohlazhdai");
    lcd.setCursor(0, 1);
    lcd.print("zhidkosti");
    lcd.setCursor(11, 1);
    lcd.print(tmp_var);
    lcd.print(char(223));
    lcd.print("C ");
    break;
case 4:
    lcd.setCursor(0, 0);
    lcd.print(selmon);
    lcd.print(".");
    lcd.print("Polozhenie dro");
    lcd.setCursor(0, 1);
    lcd.print("zaslonki");
    lcd.setCursor(11,1);
    lcd.print(polozh_dross_zaslon);
    lcd.print(" % ");
    break;

case 5:
    lcd.setCursor(0, 0);
    lcd.print(selmon);
    lcd.print(".");

```

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
						50
Вим.	Лист	№ докум.	Підпис	Дата		

```

    lcd.print("Davlenie vpusk");
    lcd.setCursor(0, 1);
lcd.print("kollektor");
    lcd.setCursor(10,1);
    lcd.print(dvk_var);
    lcd.print("kPa");
    break;
case 6:
    lcd.setCursor(0, 0);
    lcd.print(selmon);
    lcd.print(".");
    lcd.print("Oborototy dvig");
    lcd.setCursor(8, 1);
    lcd.print(rpm_var);
    lcd.print(" rpm");
    break;
case 7:
    lcd.setCursor(0, 0);
    lcd.print(selmon);
    lcd.print(".");
    lcd.print("Skorost avto");
    lcd.setCursor(8, 3);
    lcd.print(speed_var);
    lcd.print(" km/h ");
    break;
case 8:
    lcd.setCursor(0,0);
    lcd.print(selmon);
    lcd.print(".");
    lcd.print("Napryazhenie");
    lcd.setCursor(9,1);
    lcd.print(uoz_var); //Напряга
    lcd.print(" V ");
break;

```

					ЕЛІТ 6.171.00.10.088 ПЗ	Лист
						51
Вим.	Лист	№ докум.	Підпис	Дата		



