

Прикарпатський національний університет імені Василя Стефаника
Кафедра комп'ютерної інженерії та електроніки

Дзінько Владислав Миколайович
Dzinko Vladyslav

УДК 004.42[629.33/.34:53.083.7]

Спеціальність 123 «Комп'ютерна інженерія»

Дипломна робота
на здобуття освітньо-кваліфікаційного рівня магістр

Розробка телеметрії автомобіля на базі одноплатного комп'ютера
Development of car telemetry based on a single-board computer

Науковий керівник:

к.ф.-м.н., доц. Павлюк М.Ф.

Рецензент:

к.ф.-м.н., проф. Никируй Л.І.

Івано-Франківськ

2021

АНОТАЦІЯ

Автомобілі на сьогодні вже стали невід'ємною частиною нашого життя, але для збільшення комфорту користування сучасному водію, який вже звик користуватись незамінними у наш час смартфонами хотілось би бачити у своєму транспортному засобі великий та зручний у використанні сенсорний дисплей, який збільшив би інформативність типового щитку приладів та розширив його функціональність.

Автомобілі не завжди оснащені достатнім набором відображення доступної інформації, котра необхідна водію. Оскільки ця інформація є у внутрішній шині автомобіля, стає можливим за допомогою додаткового обладнання розширити цей набір даних вивівши необхідну інформацію на додатковий екран.

Ціль роботи полягає у створенні пристрою та додатку, що вміють зчитувати з внутрішньої шини автомобіля системні повідомлення, розробити алгоритм їхньої обробки та відображати оброблену інформацію в режимі реального часу.

					<i>123.УДК: 004.42[629.33/.34:53.083.7]</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Анотація	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушіє</i>
Розробив		Дзінько В.						
Перевірив		Павлюк М.					3	1
Н. Контр.								
Затвердив								

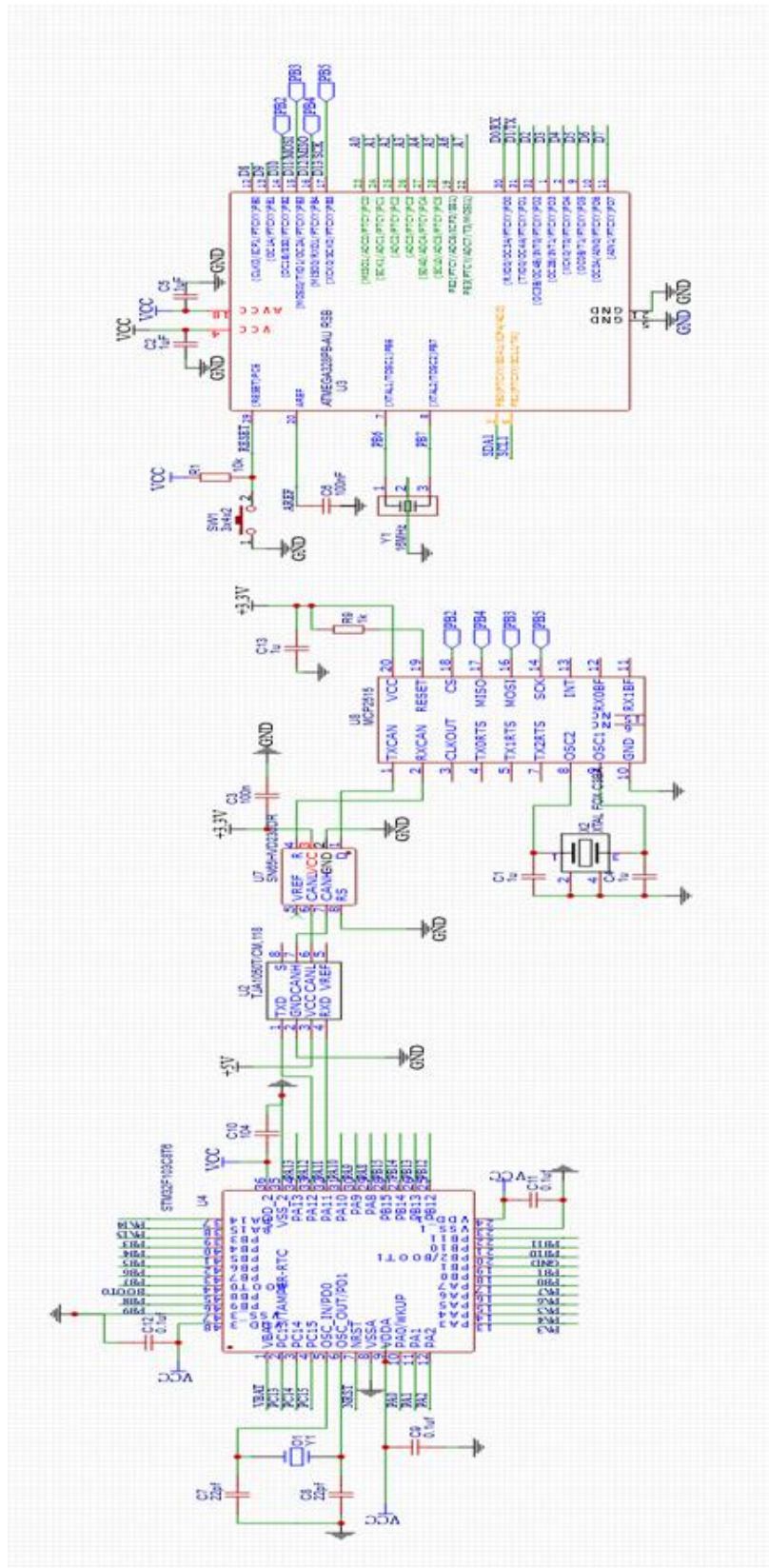
ABSTRACT

Cars have become an integral part of our lives today, but to increase the comfort of use, a modern driver, who is already accustomed to using indispensable smartphones today, would like to see a large and easy-to-use touch screen in his vehicle, which would increase the informativeness of a standard instrument panel and expanded its functionality.

Cars are not always equipped with a sufficient set of display of available information that the driver needs. Since this information is in the car's internal bus, it becomes possible with the help of additional equipment to expand this data set by displaying the necessary information on an additional screen.

The aim of the work is to create a device and application that can read system messages from the internal bus of the car, develop an algorithm for their processing and display the processed information in real time.

					<i>123.УДК: 004.42[629.33/.34:53.083.7]</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Abstract	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушіє</i>
Розробив		ДЗІНЬКО В.						
Перевірів		ПАВЛЮК М.					4	1
Н. Контр.								
Затвердив								

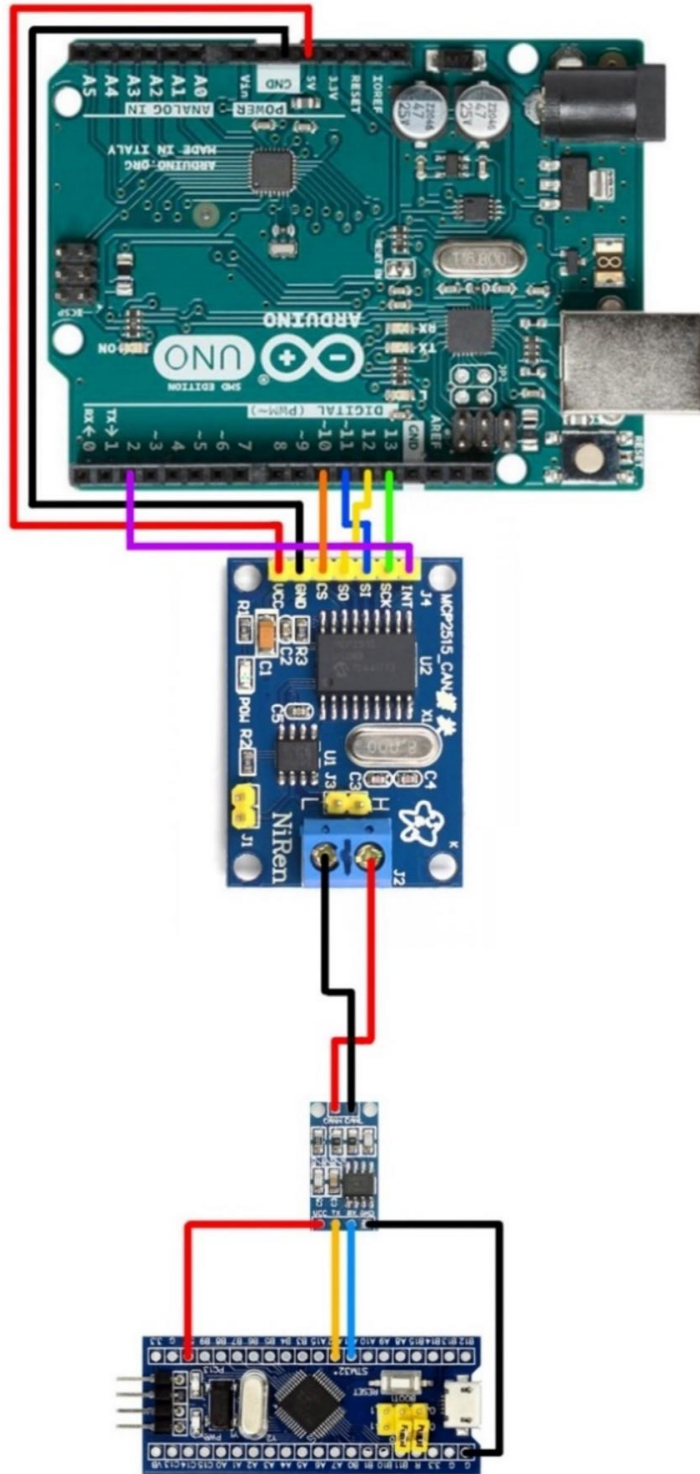


123.УДК: 004.42[629.33/.34:53.083.7]

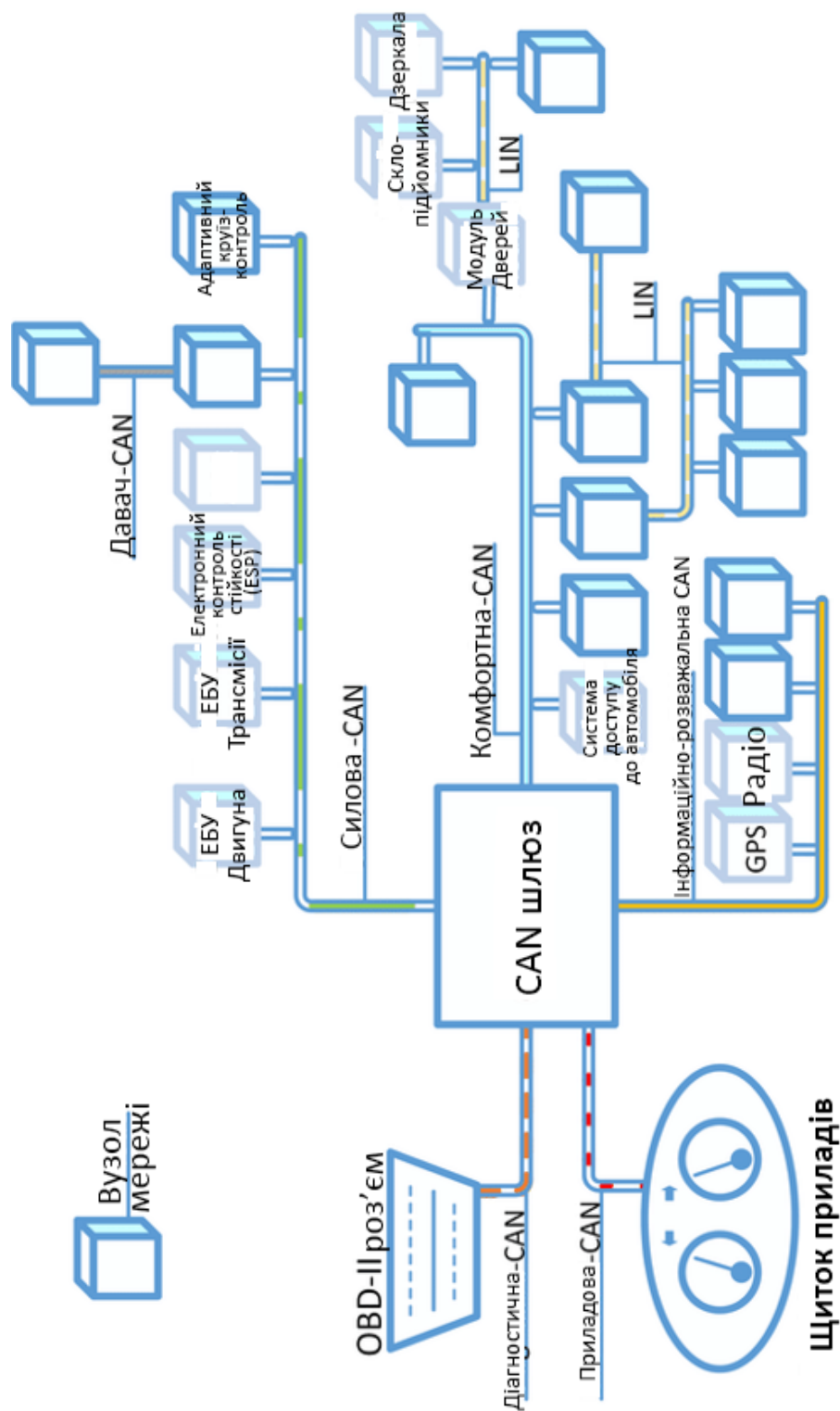
Змн.	Арк.	№ докум.	Підпис	Дата
Розробив	Дзінько В.			
Перевірив	Павлюк М.			
Н. Контр.				
Затвердив				

Схема підключення
приймача і емулятора
шини

Літ.	Арк.	Аркушів
	5	1



					<i>123.УДК: 004.42[629.33/.34:53.083.7]</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Під'єднання приймача і емулятора шини	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
Розробив		Дзінько В.					6	1
Перевірів		Павлюк М.						
Н. Контр.								
Затвердив								



123.УДК: 004.42[629.33/.34:53.083.7]

Змн.	Арк.	№ докум.	Підпис	Дата
Розробив		Дзінько В.		
Перевірив		Павлюк М.		
Н. Контр.				
Затвердив				

Структурна схема CAN мережі

Літ.	Арк.	Аркушів
	7	1

Пояснювальна записка
до магістерської роботи

на тему:

«Розробка телеметрії автомобіля на базі одноплатного комп'ютера»

					<i>123.УДК: 004.42[629.33/.34:53.083.7]</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Пояснювальна записка	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
Розробив		Дзінько В.						
Перевірив		Павлюк М.					8	117
Н. Контр.								
Затвердив								

ЗМІСТ

ВСТУП.....	11
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	13
1.1. Розвиток автомобільної промисловості.....	13
1.1.1. Велика спадщина автомобільної промисловості.....	14
1.1.2. Зміна відбитку автомобільної промисловості.....	10
1.2. «Розумний» автомобіль	23
1.2.1. Визначення ІТ автомобіля.....	23
1.2.2. Сфери застосування ІТ автомобіля.....	25
1.2.3. Інтернет речей у контексті ІТ автомобіля.....	32
1.2.4. Ринок приєднаних автомобілів.....	33
1.2.5. Розвиток ІТ автомобіля.....	35
1.2.6. Нові моделі комунікації.....	39
1.2.6.1. Комунікація Автомобіль-Автомобіль.....	39
1.2.6.2. Комунікація Автомобіль-Інфраструктура.....	40
1.2.6.3. Комунікація Автомобіль-Дім.....	41
1.2.6.4. Комунікація Автомобіль-Підприємство.....	41
1.3. Розвиток інтерфейсів користувача в автомобілі.....	41
РОЗДІЛ 2. СИСТЕМА ТА АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ.....	47
2.1. CAN шина.....	47
2.1.1. Історія мережі контролерів.....	48
2.1.1.1. Стандарти мережі контролерів.....	51
2.1.2. Технічні характеристики мережі контролерів.....	52
2.1.2.1. Бітове кодування.....	53
2.1.2.2. Кадри мережі контролерів.....	53
2.1.2.3. Арбітраж.....	59
2.1.2.4. Керування помилками.....	61
2.2. Вибір одноплатного комп'ютера.....	70

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

2.2.1. Інтегровані середовища розробки для одноплатних комп'ютерів.....	76
2.2.2. Одноплатний комп'ютер Raspberry Pi.....	77
2.2.3. Плата Асус тінкер.....	82
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРИСТРОЮ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	89
3.1. Реалізація апаратної частини.....	89
3.2. Реалізація програмної частини.....	96
3.2.1. Опис алгоритму роботи	102
РОЗДІЛ 4. ЕКОНОМІЧНА ЧАСТИНА.....	105
ВИСНОВКИ.....	107
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	109
ДОДАТКИ.....	112

					<i>123.УДК: 004.42[629.33/.34:53.083.7]</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		10

ВСТУП

Інформація про стан важливих агрегатів автомобіля необхідна водієві та в деяких випадках може бути критичною, але набору стандартних індикаторів на приладовій панелі не завжди достатньо. Також стає можливим вивчати поведінку певних частин і модулів автомобіля, шукаючи зв'язок між даними, що передаються.

Основні технологічні досягнення останніх десятиліть включають цифровізацію та створення мереж. Хоча використання комп'ютерів є невід'ємною частиною розробки продукту, мережа автомобіля досі розвивалась дуже мало. Але потенціали оптимізації також можуть бути виявлені при серійному використанні, зокрема за допомогою використання хмарних обчислень (від автомобіля до хмари), необхідні дані в автомобілі потрібно зібрати та передати на обробку.

Системи шин в автомобілях дозволяють здійснювати комунікацію, що означає обмін даними між електронними блоками управління (ЕБУ), розумними датчиками та виконавчими механізмами. Залежно від відповідних вимог використовуються різні системи шин.

Актуальність теми роботи полягає в тому, що не всі сучасні та автомобілі старшого року випуску не завжди обладнані достатнім набором відображення інформації, яка потрібна водію, оскільки ця інформація є у внутрішній шині автомобіля то за допомогою додаткового обладнання можна розширити цей набір інформації вивівши необхідну інформацію на додатковий екран. До прикладу – на панелі приладів автомобіля немає індикатора(показника) температури охолоджувальної рідини двигуна, але ця інформація є присутня в шині цього автомобіля, це означає, що є можливість вивести цю інформацію використовуючи додаткове обладнання на додатковий екран чи індикатор, навіть, на телефон. Це додаткове обладнання може бути використано не тільки як другорядний пристрій, але і як основний, наприклад при виході з ладу заводської панелі приладів автомобіля.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

Можуть підтримуватись автомобілі з CAN шиною, а це приблизно автомобілі з 1990 року випуску.

Розробка телеметрії автомобіля це досить комплексна тема, яка зачіпає багато різних сфер починаючи від архітектури автомобіля з точки зору електроніки закінчуючи розробкою інтуїтивного та зрозумілого інтерфейсу користувача.

Також є можливість демонструвати різноманітну інформацію, яка присутня в шині автомобіля, це може бути: стан дверей (відкриті або закриті, і які саме), поточна витрата палива(миттєва витрата в літрах), поточні оберти двигуна(в об\хв), поточний рівень палива в баку в літрах, стан стоянкового гальма, температура в салоні(в °C), зовнішня температура(в °C), температура оливи(в °C), одометр(пробіг) в кілометрах, стан покажчиків повороту, температура охолоджувальної рідини(в °C), стан освітлювальних приладів(дальнє чи ближнє світло фар) та багато інших параметрів.

Метою роботи є розробка телеметрії автомобіля на базі одноплатного комп'ютера.

Предмет дослідження – збір телеметрії автомобіля, її обробка та її обмін із комп'ютером.

Для досягнення поставленої мети в роботі визначені і розв'язані такі **завдання**. Створення пристрою та додатку, які виконують наступні операції:

- під'єднання до шини автомобіля через роз'єм діагностування,
- зчитування даних з внутрішньої шини автомобіля,
- отримання повідомлень формату фреймів мережі контролерів,
- емуляція надсилання пакетів мережі контролерів для розробки в умовах без під'єданого автомобіля,
- створення алгоритму обробки фреймів мережі контролерів,
- використання набору обробників для аналізу обраних пакетів,
- отримання параметру з CAN повідомлення,
- використання інтуїтивного інтерфейсу користувача,
- візуалізація обробленої інформації в режимі реального часу.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Розвиток автомобільної промисловості

Автомобільна промисловість (АП) сформувала промислові обриси світової економіки у ХХ столітті і продовжує залишатися ключовим гравцем нинішнього масштабного соціально-технічного переходу, спричиненого цифровою революцією та пошуком нових систем мобільності. Поки ці процеси розвиваються, зростає очікування, що електричні та автономні транспортні засоби разом із моделями бізнесу та праці, що базуються на онлайн-платформах та взаємопов'язаних системах, перетворять усю автомобільну промисловість у тому вигляді, в якій ми її знаємо.

За словами Шеллера та Уррі (2006) та Уррі (2004), автомобільна промисловість створила потужну «систему автомобільності»: тобто потужну систему, залежну від автомобіля, яка створила архетиповий виготовлений об'єкт, пов'язаний із знаковими фірмами минулого століття; основний елемент індивідуального споживання, пов'язаний із зображеннями соціального статусу та того, що становить «гарне життя»; промисловий комплекс, пов'язаний з кількома іншими галузями промисловості; та переважна форма приватної мобільності, яка підпорядковувала інші види мобільності.

Ці особливості є очевидними і сьогодні, хоча вони можуть приймати різні образи залежно від економічних, географічних, соціальних та технічних кордонів. Факт залишається фактом, що автомобілі для багатьох залишаються «машинами свободи»: унікальними артефактами, які пропонують рішення для перевезення вантажів та людей, заохочуючи вираження особистих поглядів. Дійсно, мало товарів є такими важливими для людей та суспільства, як сьогодні автомобілі. Дослідження 2015 року, здійснене за дорученням Міжнародної організації виробників автотранспортних засобів (ОІСА), показало, що 57% світових споживачів не могли собі уявити життя без автомобіля, і що автомобілі асоціюються з неперевершеним комфортом, ефективністю для щоденних подорожей та сильним бажанням бути власником.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

«Система автомобільного транспорту» спиралася на потужний фундамент промислової парадигми минулого століття, яка формувала виробничі, розподільні та інституційні рамки, які регулювали індустріальне суспільство і які розвивалися з конвеєра Генрі Форда. Виходячи з цього, США стали монополістичним центром світової економіки ХХ століття, а масове виробництво стало технологічним та організаційним шаблоном для інших країн. Згідно з параграфом масового виробництва, праця та керівництво брали участь у вирішальних битвах та уклали колективні договори, які покращили розподіл доходів та сприяли розв'язанню доброзичливого кола між виробництвом та споживанням, що стояв за економічним бумом після Другої світової війни. Це був золотий вік капіталізму, період, який тривав до першої половини 70-х років ХХ століття.

1.1.1. Велика спадщина автомобільної промисловості

Багато фактів і даних свідчать про значну спадщину та багатство галузі. Глобальні доходи автомобільної промисловості досягли 3,8 трлн доларів у 2018 році після того, як вони зростали щорічно на 3,3% за останні п'ять років. Це означає, що якби автовиробництво було країною, воно мало б п'яту за величиною економіку у світі, після США (з ВВП 20,4 трлн доларів), Китаю (14), Японії (5,1) та Німеччини (4,2)).[1]

11 найбільших автовиробників (відтепер OEM-виробники) – Toyota (6-е місце), VW (7-е), Daimler (16-те), GM (21-е), Ford (22-е), Honda (30-те), SAIC (36-е), BMW (51-е), Nissan (54-е), Dongfeng Motor (65-е) і Hyundai (78-е). Навіть найбільша корпорація автозапчастин, Бош, входить до сотні найкращих (займає 75 місце). Тому цей сектор займає 12% частки топ-100 світових корпорацій та 20% топ-10 (Fortune 2018). [2] Ці автомобільні корпорації управляють більшими ресурсами, ніж більшість економік світу. Доходи Toyota та VW лише поряд з багатством 16 кращих економік світу. Інакше кажучи, їх економічний охоплення більший, ніж у 169 країн.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

Промисловість стимулює імідж того, що вона стимулює економічне зростання, конкурентоспроможність та ефект наздоганяння в тій мірі, в якій щороку країни, що розвиваються, намагаються проникнути або закріпити позиції у цьому секторі. Якщо в ХХ столітті було виготовлено мільярд автомобілів, переважно в США, Західній Європі та Японії, лише у 2017 році 40 країн виготовили 98,1 мільйона автомобілів-левів: 74,1 мільйона автомобілів та 24,2 мільйона комерційних автомобілів. Репутація автомобільної промисловості зміцнюється відомими економічними фактами: для типової потужної країни, що виробляє автомобілі, ця галузь є найбільшою стійкою виробничою діяльністю, чистим експортером, основним генератором міжнародних грошових переказів та важливим джерелом досліджень та розвитку, а також надання офіційної, кваліфікованої роботи.

Витрати на автомобілі становлять значну частину загальних витрат споживачів. У розвинених країнах цей показник становить близько 10%; проте варіації є значними. У Сполучених Штатах у 2016 році витрати на транспортні засоби становили у середньому 9135 доларів США, що становить 15% від загальних споживчих витрат. У менш розвинених країнах це може бути набагато вище. Наприклад, у Мексиці витрати на транспортні засоби можуть становити до чверті загальних особистих витрат.

У 2017 році автомобілі були першим світовим експортним продуктом з чистою вартістю 739,2 млрд доларів, що на 9,2% більше, ніж у 2013 році. На десять країн припадає понад три чверті всього експорту (Німеччина, Японія, США, Канада, Велика Британія, Мексика, Південна Корея, Іспанія, Бельгія та Франція), при цьому лише європейські країни представляють 55% частки з них. Автомобільна промисловість є лідером за витратами на НДДКР та кількістю працюючих інженерів. За даними Центру автомобільних досліджень, на них припадає 17% від загального обсягу фінансування НДДКР для всіх галузей промисловості з річним бюджетом 99 млрд. доларів США: п'ять найбільших виробників техніки входять до топ-20 корпоративних споживачів у сфері НДДКР у всьому світі. Галузь наймає більше інженерів на 1000 робочих місць, ніж будь-

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

яка інша промисловість (Центр автомобільних досліджень). Крім того, галузь підтримує зворотні зв'язки з багатьма іншими галузями промисловості, такими як сталь, залізо, алюміній, скло, пластмаса, килимове покриття, текстиль, комп'ютерні чіпи та гума, тоді як її прями зв'язки підключаються до потужної нафтової промисловості та таких різноманітних постачальників послуг, як фінансування, страхування, реклама, маркетинг, паркування, ремонтні майстерні та вторинний ринок.

Так само, як вона відіграє ключову роль у стимулюванні країн наздоганяти та залишатися конкурентоспроможною у світовій економіці, автомобільна промисловість також є одним з найбільших постачальників промислової зайнятості. Близько 5% робочих місць у виробництві – це прями робочі місця для цієї промисловості. Крім того, якщо врахувати, що на кожну пряму роботу галузь впливає на п'ять непрямих робочих місць. До автопрому відноситься більш ніж 50 мільйонів робочих місць. Вони відповідають як різним рівням постачальників автозапчастин та комплектуючих, так і роздрібним торговцям та службам.

Враховуючи його вплив на ринки праці, автомобільна промисловість мала вирішальне значення у визначенні кордонів відносин з управлінням працею разом із прагненнями тисяч працівників, включаючи кваліфіковану та технічну робочу силу, у здобутті вищих доходів та просуванні по соціальних сходах. Той факт, що збірні підприємства були задумані як вертикальні промислові комплекси, що об'єднують усі матеріали, процеси та технології, необхідні для створення автомобіля, означав, що це промислові установки, якими керують тисячі робітників. Таким чином, місця в цій галузі стали ідеальними робочими місцями для профспілкової діяльності, яка в кінцевому підсумку мала один з найвищих показників профспілки працівників у цілій економіці. До другої половини минулого століття автомобільна промисловість стала не лише домом для деяких наймасштабніших битв між працею та керівництвом, але й поступово стала визначальною для колективних переговорів. Протягом другої половини ХХ століття Об'єднані автовиробники та автовиробники у США уклали трудові угоди, які стали прикладом для наслідування як на національному, так і на

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

міжнародному рівнях. Основними ознаками цієї моделі праці були механізми покращення компенсацій, пільг, які захищали зайнятість, стандартизація заробітної плати та правил праці всередині та між фірмами, а також профспілковий контроль за роботою, включаючи детальну класифікацію вакансій, права стажу та зміст роботи.

Промисловий союз металістів став найбільшим об'єднанням Німеччини та найбільшим промисловим об'єднанням у Європі. Знову ж таки, колективні договори автомобільної галузі, особливо ті, що були укладені у центрі виробництва автомобілів у Баден-Вюртемберзі, де проживають Daimler та Bosch, стали першими в країні. У Бразилії профспілка металургів створила Central Unica dos Trabalhadores в регіоні Сан-Паулу, ще один вузол для виробництва автомобілів, з якого з'явилися обидва шаблони трудових контрактів для решти країни, а також Партія робітників. Луїс Ігнасіо Лула, колишній металообробник, який згодом керуватиме країною на посаді президента. Автопрофспілки навіть першими створили міжнародні мережі для боротьби з глобалізацією корпорацій. Ще в 1966 році UAW створила профспілкові мережі в «Детройтській трійці». Підводячи підсумок, поєднання потужних профспілок та провідних фірм створило спадщину підвищення заробітної плати, кращих виплат та більшої безпеки праці в автомобільній промисловості.

Ці дані розповідають історію столітньої промисловості, яка має величезне економічне, технологічне та трудове значення і продовжує бути активним гравцем у світовій економіці. Тим не менш, протягом останніх чотирьох десятиліть автомобільна промисловість переживає глибокі трансформації, які в даний час сходяться зі зміною середовища та незліченною кількістю інституційних вимог, викликаних цифровою революцією. В результаті галузь переходить і перебудовується в умовах найбільш руйнівного середовища, яке вона коли-небудь відчувала.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

1.1.2. Зміна відбитку автомобільної промисловості

Очевидно, що теперішній географічний слід галузі розвивається. Незважаючи на скорочення у розвинених країнах, центри та ринки виробництва автомобілів стрімко зростають у країнах, що розвиваються. Насправді, починаючи з 2009 року, країни, що розвиваються виробляють більше і мають більшу частку ринку, ніж розвинені країни. Лише Китай виробляє майже третину світового виробництва автомобілів (більше 32 мільйонів одиниць щорічно) і претендує на більшу частку ринку. Зіткнувшись з цією новою географічною конфігурацією, класична концепція територіального розподілу стала застарілою, тобто уявлення про те, що промисловість міститься в регіональних виробничих системах, де домінують Сполучені Штати у Північній Америці, округах Західної Європи, та Японії у Східній Азії. Фахівці з Міжнародної мережі автомобільної промисловості та її співробітників звернули увагу на напруженість у галузі, створеній у «світі регіонів», що складається з регіональних економічних просторів у розпал епохи глобалізації. Тим не менш, менш ніж через два десятиліття жоден із цих блоків не був таким, яким він передбачався, і ні його члени, ні керівництво не залишаються тими ж. Епіфеномна епоха Китаю в Азії, Брексіт у Європі та новий договір між США та Мексикою та Канадою у Північній Америці представляють процеси, коли уряди та суб'єкти інтенсивно заперечують панування галузі на місцевих та світових ринках.

Жульєн та Парді (у 2013) виявили, що ці тенденції створили подвійний процес реструктуризації старої промисловості, та структурування нової майбутньої галузі в країнах, що розвиваються.[3] В результаті до бізнес - стратегій були додані нові складності, які перевіряють відповідність продуктової стратегії більшому розмаїттю ринків, а також організацію виробничих ланцюгів та зобов'язань щодо управління між суб'єктами. Хоча попереднє регіональне стримування галузі розбилося, результати цієї напруги наразі непередбачувані.

Технологічні кордони галузі зазнають однаково глибоких змін. Наприкінці ХХ століття попередження про зміну клімату та посилення урядових норм щодо контролю за викидами та забрудненням спонукали пошуки альтернативних

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

систем двигуна та низьковуглецевих технологій. Це посилюється, оскільки ЄС посилює проблеми забруднення навколишнього середовища з приватних автомобілів на основі двигунів внутрішнього згорання (ДВЗ). Завдяки зростаючій увазі громадськості та правилам щодо контролю забруднюючих викидів, 5 автовиробників прискорили експерименти з цілим рядом альтернативних силових агрегатів, головним чином з електричними транспортними засобами, включаючи електричні акумулятори, паливні елементи, гібридно-електричні та з'єднувальні гібриди транспортних засоби, вдосконалені ДВЗ, підвищення продуктивності горючих та біопалив тощо.

Виробники тільки почали пристосовуватися до попередніх правил, коли прогрес у напрямку взаємопов'язаних виробничих систем, штучного інтелекту та Інтернету речей – ниток цифрових технологій – знову потряс індустрію. Дві можливі руйнівні події були центральними у цьому: експерименти з автономними транспортними засобами без водія та нові моделі бізнесу та праці на основі онлайн-платформ та служб спільної мобільності. Ці можливості уособлюються в нових гравцях, таких як американські Тесла та Убер, азіатські Ола та Діді Чуксін, а також у високотехнологічних лайнерах на кшталт Гугл Waymo, які проникають у цей сектор. Виробники, заявляючи про свою готовність реагувати на них і стати архітекторами епохи нових мобільностей, несамоовито шукають союзів з новими гравцями та між собою. Знову ж таки, за цим криється запекла боротьба за якщо не за лідерство галузі, то принаймні за виживання.

Зі змінами зверху та технологічною траєкторією шляху, кінцевий результат обробляється та оспорується на місцевому рівні. Це середовище, в якому державна політика стає як ніколи важливою. Контрастні пріоритети очевидні в державній політиці, спрямованій на порятунок виробників оригінального обладнання за рахунок введення історичних рівнів ресурсів, проти зосередження уваги на пануючій парадигмі електромобілів, проти залучення інвестицій та зайнятості з низькими зарплатами, порівняно з тими, які роблять ставку на нову індустрію сталої мобільності та комплексних транспортних послуг, проти політики, спрямованої на активізацію інновацій у технологічній парадигмі ХХ

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

століття, з метою підвищення ефективності ДВС та продовження терміну служби нафтової ери тощо.

Машина, яка змінила світ Вомака та ін. (1990) передбачали, що ощадливе виробництво врешті-решт перемаже і стане стандартом у галузі, створивши свої «зв'язки» практики людських ресурсів, які використовують організаційні зобов'язання, виконуючи при цьому середовище співпраці у трудових відносинах. [4] Прогноз вірно провалився. Фінансова та грошова криза в Японії 1990-х рр. виявила слабкі сторони ощадливої системи, і на рубежі нового століття навіть Toyota відкинула основні риси цієї практики, такі як довічна зайнятість, просування по службі зсередини та щорічне підвищення заробітної плати.

Автомобільні ринки праці стрімко зростають у багатьох країнах, що розвиваються, що кидає виклик спроможності існуючих інституційних механізмів відповідно навчати, наймати, сертифікувати та оплачувати працівників та регулювати відносини з управлінням працею. На противагу цьому, ринки праці розвинених країн, зокрема країн Великої Сімки – США, Канади, Японії, Італії, Франції, Німеччини та Великобританії – або скоротилися, або потрапили в застій. Найбільш помітний вплив цього був на рівень профспілки, права та доходи працівників. Сполучені Штати відображають одну крайність того, що було визначено як гонка до дна. У 1978 році промисловість у Детройті мала близько 1,1 мільйона робочих місць. Через три десятиліття ця сума була скорочена до 945 тисяч. Подібним чином, у 1987 році членство в Об'єднаних автомобільних працівниках становило близько мільйона, а через два десятиліття воно досягло найнижчої позначки – 355 000 осіб, тоді як зарплата знизилася на третину. Зміщення промисловості в бік ЄС стимулювало цю гонку до дна, особливо у випадках, коли промисловість переїхала до Мексики та до Центральної Європи на єдиний європейський ринок. У багатьох ранніх дослідженнях стверджувалося, що реальна або загрозлива можливість переміщення більшого обсягу виробництва в ці сфери дешевої робочої сили

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

надасть керівництву більші важелі впливу, щоб закликати до заморожування заробітної плати та утисків робочої сили.

Крім того, автомобільна індустрія не втекла від світової тенденції промисловості за останні 30 років, де спостерігався зростаючий розрив між продуктивністю праці та заробітною платою. Це означало, що багато робочих місць, створених як у країнах, що розвиваються, так і в розвинених країнах, є нестабільними. Після чотирьох десятиліть послаблення ринкових відносин та відносин із зайнятістю та дерегуляції після неоліберального кредо мало залишилося тієї епохи, коли участь у автомобільній промисловості була гарантованою путівкою до суспільного просування. Це, як правило, замінюється низькими зарплатами та меншим соціальним захистом, особливо для нових працівників. Проте ці елементи трудових відносин не є ні єдиними, ні універсальними. Варіації підкоряються інституційній спадщині кожної країни, а також балансу сил системи виробничих відносин, зокрема, можливостям та стратегіям реагування організованої праці. Наприклад, у Німеччині система співвизначення та трудових рад, а також права автомобільних працівників залишилися в основному незмінними. Тим часом у таких країнах ЄС, як країни БРІКС (Бразилія, Росія, Індія та Китай), бум промисловості створив прискорений процес сегментації робочої сили, де скорочене ядро з кращими доходами та безпекою робочих місць контрастує з рештою працівників, які підпорядковуються субпідрядникам, які передають на аутсорсинг, мають менший дохід та відсутність або обмежені права.

Промислові ринки праці та трудові відносини на очах у фальшивих компаній чи відносини без зайнятості, що видно з бізнес-моделей мобільних онлайн-платформ, таких як Убер. Це два різні ринки праці: один представляє офіційну зайнятість і захищений традиційними промисловими соціальними контрактами, а інший – це економіка гнучкої роботи на неповний робочий день, яку виконують незалежні підрядники. Незрозуміло, коли 2,5 мільйона водіїв у всьому світі, наприклад, мають Убер, які пропонують прокат власних автомобілів

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

і без будь-яких договірних відносин з компанією, може перетинатися зі сформованою промисловістю та безпосередньо впливати на її робочі місця.

Подібним чином, з прийняттям електромобілів, незрозуміло, скільки пройде часу, перш ніж технологія, яка готова витіснити домінуючу парадигму ДВС разом із впровадженням автономних транспортних засобів, насправді її замінить. Безперечно те, що наразі це перебуває на експериментальній стадії, коли його учасники ще мають працювати з прибутковим бізнесом, що дозволяє їм успішно орієнтуватися у таких змінних, як баланс витрат, діапазон та безпека руху, а також коригування правил уряду. Таким чином, такі компанії, як Тесла та Убер, продовжують втрачати гроші, а інвестори продовжують робити ставку на своє майбутнє, підвищуючи свою ринкову вартість вище, ніж більшість виробників оригінального обладнання.

Ініціатива щодо електричних транспортних засобів, очолювана Міжнародним енергетичним агентством, пропонує у своєму сценарії мету досягти 30% частки ринку електромобілів до 2030 року. У 2018 року 5 мільйонів електромобілів вже на дорозі у всьому світі, а продажі досягли 2 мільйонів, більше половини з них у Китаї (Енергетичне агентство). Тим не менш, електромобілі все ще становлять не більше 2% частки ринку. Щодо самокерованих автомобілів, проспективне дослідження підрахувало, що протягом десяти років після затвердження правил їх розповсюдження складе 60% автопарків, принаймні в Сполучених Штатах. [5] Незалежно від точності цього прогнозу, впровадження значно вплине на ринки праці та трудові відносини в галузі. Електромобілі займають шосту частину традиційного ДВС; його складання займає на 30% менше часу, а акумуляторному заводу потрібно п'ята частина робочої сили двигунобудівного заводу. Одне дослідження, проведене компанією IG Metall, показало, що з нинішніх 872 000 робочих місць у німецькій автомобільній промисловості 78 000 робочих місць пов'язаних з коробкою передач будуть під загрозою до 2030 року через електромобілі. Те саме відбудеться з половиною з 210 000 робочих місць, пов'язаних з виробництвом силового агрегату, тоді як самокеровані автомобілі безпосередньо впливатимуть

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

на три пов'язані галузі промисловості, а саме: транзит, логістику та автоперевезення. Можливо також, що самокеровані автомобілі впливатимуть на звички водія та попит на транспортні засоби; проте докази цього все ще не є остаточними.

Між описаними вище двома крайнощами Сполучених Штатів і Німеччини лежить ціла низка трудових перебудов. Те ж саме стосується робочих місць, які створюються в нових службах мобільності. Відстеження цих ринків праці та виробничих відносин, реконфігурації – це ще одна програма досліджень, яку слід продовжити.

1.2. «Розумний» автомобіль

Автомобільна промисловість впроваджує незліченні інновації з кінця 19 століття. Колись виробників автомобілів часто звинувачували що вони просто гнуть метал, але вони зазнали значних змін за останні 30–40 років. Так і сталось – сучасна електрика та електроніка стають невід'ємною частиною автомобіля для всіх. Вміти оптимально керувати компонентами це необхідна навичка для автомобільного виробника. У багатьох транспортних засобах впроваджені системи допомоги, які використовують датчики та камери, щоб автомобіль міг «відчувати» та бачити.

Зараз наближається нова ера, адже Інтернет в автомобілі дає можливість інтелекту автомобіля збільшитись в рази. Телематика як мистецтво слово з ІТ та телекомунікацій потрапило в автомобілі. Щоб автомобілі могли спілкуватися між собою, а також з іншими технологіями, які працюють через Інтернет.

1.2.1. Визначення ІТ автомобіля

Поки що всі марки автомобілів відрізняються потужністю двигуна, системою приводу або особливим дизайном. Клієнт має зрозуміти саме ці особливі характеристики автомобіля, а потім приймати рішення про покупку.

Однак не так давно була ще одна важлива опція покупки: оснащення автомобіля сучасними інформаційними технологіями. Особливо молодше

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

покоління хотіло б використовувати такі функції під час водіння мати доступ до смартфонів і, наприклад, відповідати на електронні листи або SMS безпосередньо в дорозі, їздити за допомогою карт Гугл і вибирати та відтворювати особистий музичний вміст. Водночас слід стежити за тим, щоб увага залишалася на дорозі.

Той виробник автомобілів що не може запропонувати ці функції сьогодні з точки зору покупця більше не приваблює і швидко втрачає частку ринку.

Звичайно, на перший погляд, є також старший клас покупців з більшими фінансовими можливостями, але для цієї цільової групи автомобіль з Інтернетом стане дуже привабливим. Якщо, наприклад, автомобілем можна дистанційно керувати через додаток то його можна буде і припаркувати через додаток, тому це величезне полегшення для водіїв. Також різноманітні функції безпеки, які пропонує нове авто, зацікавлять не тільки молодшу аудиторію – повністю автоматичні системи попередження про виїзд зі смуги руху, так званий екстрений виклик, який автоматично викликає рятувальників у разі аварії або постійно доступна інформація про стан транспортного засобу не тільки в автомобілі, а й з відстані через додаток або через портал клієнта у веб-браузері.

Тим часом у всіх західних країнах компанії виробники стільникових телефонів створюють всебічне, швидке та стабільне покриття стільникової мережі. Це означає не тільки те, що ви можете здійснювати телефонні дзвінки з будь-якого місця, а й те що підключення до Інтернету можна встановити практично з будь-якого місця, це стосується також автомобіля. Наприклад, якщо автомобіль оснащений SIM-карткою або з'єднання між смартфоном і автомобілем за допомогою радіопередачі (переважно через Bluetooth) встановлюється режим, так би мовити, «онлайн».

Функції мережевого транспортного засобу більше не базуються виключно на електриці чи електроніці всередині автомобіля, але також і на ІТ через мережу інтернет.

Одне з можливих визначень ІТ автомобіля може бути таким: термін ІТ автомобіль враховує всі потоки інформації, які надходять у транспортний засіб

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

або з нього, або які є в самому транспортному засобі. Водії виступають як безпосередні одержувачі та постачальники інформації у розширених бізнес-процесах та бізнес-моделі, незалежно від часу та місця розташування транспортного засобу. [6]

В цілому, однак, слід зазначити, що загальновизнане і комплексне визначення термінів ІТ автомобіля або під'єданого автомобіля ще не утвердилося. Ця тематика ще надто динамічно розвивається та надто молода для конкретних можливих визначень та розмежувань.

1.2.2. Сфери застосування ІТ автомобіля

Перш за все, повинно бути зрозуміло, які зацікавлені сторони присутні в середовищі ІТ автомобіля: з одного боку, це виробники автомобілів та їх постачальники, а з іншого боку, є користувачі транспортних засобів. Крім того, є ще одна зацікавлена сторона, яка відіграватиме важливу роль: держава повинна визначити в рамках законодавства, як використовувати майбутню цифрову транспортну інфраструктуру, особливо стосовно довгострокової мети автономного водіння.

Для того, щоб показати сильні та слабкі сторони, а також можливості та ризики для вищезазначених зацікавлених сторін у сфері ІТ автомобіля, для кожного зацікавленого боку був підготовлений SWOT-аналіз.

Таблиця 1.1 представляє SWOT-аналіз для виробників та постачальників автомобілів, таблиця 1.2 для користувачів транспортних засобів та таблиця 1.3 для державного сектору або держави.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

SWOT-аналіз для виробників та постачальників автомобілів

Сильні сторони	Слабкі сторони
<ul style="list-style-type: none"> • Прямий контакт з користувачем автомобіля • Збір даних про поведінку користувача транспортного засобу • Створення привабливого досвіду користувача • Підвищення комфорту та простоти використання • Полегшення для водія за допомогою інтелектуальних інформаційних систем водія 	<ul style="list-style-type: none"> • Малий досвід роботи з новими ІТ - технологіями в автомобілях • Брак персоналу, який важко набрати
<ul style="list-style-type: none"> • Встановлення прямих і довгострокових відносин з клієнтами • Розвиток нових адитивних бізнес - моделей 	<ul style="list-style-type: none"> • Робота з потоком інформації (великі дані, дані про автомобілі та клієнтів) • ІТ-безпека (хакерські атака на автомобіль)
Можливості	Загрози / ризики

SWOT-аналіз для користувачів транспортних засобів

Сильні сторони	Слабкі сторони
<ul style="list-style-type: none"> • Більший комфорт та легше керування автомобілем • Підвищена безпека під час водіння • Доступ до автомобіля та управління ним з будь -якого місця через додаток або веб -портал 	<ul style="list-style-type: none"> • Малий досвід роботи з новими ІТ - технологіями в автомобілі (труднощі в роботі та розумінні)
<ul style="list-style-type: none"> • Підвищення безпеки руху • Краще уникнення заторів і, таким чином, швидше прибуття до пункту призначення 	<ul style="list-style-type: none"> • Безпека особистої інформації • Чи все ще гарантується анонімність? • Доступ третьої сторони до вашого власного автомобіля?
Можливості	Загрози / ризики

SWOT-аналіз для державного сектору або держави

Сильні сторони	Слабкі сторони
<ul style="list-style-type: none"> • Контроль руху можна активно розробляти • Кращий моніторинг дорожнього руху 	<ul style="list-style-type: none"> • Малий досвід використання нових можливостей, особливо з автомобіль – інфраструктура • Правова ситуація щодо майбутнього автоматизованого управління дорожнім рухом неясна та складна з етичної точки зору
<ul style="list-style-type: none"> • Підвищення безпеки дорожнього руху (менше ДТП) • Можливе краще управління трафіком, адаптоване до поточної ситуації • Уникнення заторів (економічна шкода, заподіяна заторами зводиться до мінімуму) 	<ul style="list-style-type: none"> • Безпека даних • Анонімність користувачів транспортних засобів
Можливості	Загрози / ризики

Переваги та можливості, представленого ІТ автомобіля, також відображені в основних сценаріях застосування. У таблиці 1.1 наведено три основні сфери застосування, які роблять ІТ автомобілів однією з найважливіших тем ринку автомобільної промисловості.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

	<p>фактори можна збирати в майбутньому, і так на ранній стадії та протидіяти пробкам у режимі реального часу. Результат не тільки заощаджує великі гроші, але, перш за все, нерви і марну трату часу. Зокрема, для вантажних перевезень це означає краще планування та економічну ефективність. Крім того, можна заощадити багато палива, уникаючи заторів.</p>
<p>Інформаційно-розважальна система</p>	<p>Інформаційно-розважальна програма – сполучення слів, що складається з інформації та розваг. Це означає функції в транспортному засобі, які приносять у автомобіль так званий вміст, наприклад, новини, ціни на фондовому ринку, погода чи Facebook, Twitter та повідомлення електронної пошти, а також музику та розваги. Ці дані часто збагачуються поточним сигналом GPS, який показує місце, де ви знаходитесь. Це робить можливим розміщення реклами та місцевих новин, а також найдешевшу заправну станцію поблизу або рекомендовані ресторани чи готелі, а також найближчу зарядну станцію для електромобіля. Це означає, що часто непродуктивний час перебування в транспортному засобі може стати більш значущим у майбутньому за допомогою великої кількості нових інформаційно-розважальних функцій.</p>

Порівняння автомобільної та мобільної галузей зв'язку

Автомобільна промисловість	Індустрія мобільного зв'язку
Дуже довга історія і дуже стара (> 100 років)	Дуже молода промисловість (приблизно 15-20 років)
Дуже довгі цикли розвитку (приблизно 5–7 років)	Дуже короткі цикли розвитку (1-2 роки)
Історично немає прямого контакту з клієнтами (часто лише через майстерню або дилера)	Дуже тісний контакт з клієнтами
Дуже велика потреба в безпеці	Питання безпеки дуже важливі, але не настільки виражені, як у виробників автомобілів

Для того, щоб втілити в життя «мережевий» автомобіль із описаними варіантами застосування, мобільне з'єднання для передачі даних у транспортному засобі є абсолютно необхідним. Тут починають діяти нові партнери автомобільної промисловості: постачальники мобільного зв'язку та постачальники ІТ -послуг як нові гравці галузі. Два світи зростають разом, у яких раніше було мало або взагалі не було точок дотику. Таким чином, у таблиці 1.5 показано порівняння двох гравців у автомобільному та мобільному зв'язку чи секторі ІТ.

Тема циклів розвитку зокрема виділяється в таблиці 1.5. Для галузей мобільного зв'язку та ІТ, які ще молоді в порівнянні з автомобільною промисловістю, оновлення продуктів є нормальним у дуже коротких циклах. Клієнт звик до того, що кожні 1-2 роки на ринок надходить новий продукт із значно покращеними характеристиками. Швидкість інновацій надзвичайно висока. Нова модель автомобіля з'являється приблизно кожні 6-7 років. Можливо, фейсліфтинг був представлений на ринку між ними, але що стосується жорсткого інформаційно-розважального світу автомобіля, то клієнти все ще

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

значною мірою стикаються з інформаційно-розважальними пристроями, які не масштабовані та не можуть бути оновлені при покупці нових автомобілів. Оновлення можливе лише за допомогою компакт-диска, який є громіздкою покупкою, і успішна установка та використання якої часто доступні лише користувачам транспортних засобів, які знаються на технологіях.

1.2.3. Інтернет речей у контексті ІТ автомобіля

Через постійне підключення транспортного засобу до Інтернету, такі «речі» мають на увазі у значенні «Інтернет речей»:

- Зв'язок автомобіля з виробником автомобілів (ОЕМ) та дилером
- З державними органами (наприклад, для платних пунктів пропуску, офісу автотранспортних засобів або відділу дорожнього руху)
- Зв'язок з інфраструктурою, наприклад світлофорами, дорожніми знаками або гаражами
- Зв'язок з іншими транспортними засобами

Основна увага приділяється питанню, як автомобільний аспект в Інтернеті речей (ІоТ) впливає на виробників автомобілів, постачальників послуг, наглядові органи та споживачів щодо інтеграції інтелектуальних пристроїв, інформаційно-розважальних функцій, вбудованих рішень, великих даних, сумісності, контролю та безпеки? Залишається багато питань без відповіді, але ринок стрімко зростає.

«Очікується, що до 2023 року в дорозі з'явиться понад 285 мільйонів транспортних засобів, які мають можливість підключення до мережі інтернет, і враховуючи прогнози, що глобальні доходи від технологій та послуг Інтернету речей досягнуть 8 трильйонів доларів до 2022 року, автомобільний аспект Інтернету речей поширюється на автомобільну промисловість», - коментує Джоель Гофманн, директор з маркетингу Альянсу GENIVI та автомобільний стратег Intel. [8]

Мережа в автомобілі радикально змінює точку зору та критерії покупки користувачів автомобіля. Вся промисловість переживає потрясіння. Раптом знання ІТ потрібні виробникам та постачальникам автомобілів не лише для

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

оптимізації виробничих процесів чи впровадження систем ERP, а й безпосередньо у транспортному засобі.

Консультаційна компанія з питань управління МНР (Mieschke Hofmann and Partner) досліджувала це питання у дослідженні, проведеному спільно з Дослідницьким центром інформаційних технологій у Карлсруе. Загалом у 2013 році на цю тему було опитано 350 керівників ІТ-галузей автомобільної промисловості.

Технічний розвиток виробника автомобілів відповідає за подальший розвиток ІТ автомобіля. Це також означає інтеграцію вбудованих ІТ у процеси після продажного обслуговування, а також інтерфейс для постачальників послуг та вмісту (приклад: навігація Гугл). Розвиток електроніки та програмних рішень забезпечують спеціалізовані постачальники.

Надання рішень для керування мережею в майбутньому буде інтегровано в існуючу ІТ-організацію OEM. Таким чином, ІТ-організація візьме на себе відповідальність за ІТ-компоненти у майбутньому сценарії інтегрованого рішення «під'єднане водіння». Ці ІТ-компоненти включають рішення для таких областей: вбудованої системи, доступу, телематичного бекенду, корпоративної ІТ-інтеграції. Директор по технічній роботі несе відповідальність за всі ІТ-розробки та рішення в компанії та в транспортному засобі.

1.2.4. Ринок приєднаних автомобілів

У 2017 та 2018 роках тема ІТ автомобіля вважалась однією з популярних тем автомобільної промисловості з найбільшим потенціалом. Дослідницький інститут Gartner описує мережевий автомобіль як «найкращий мобільний пристрій». Матіас Віссманн як президент VDA цитується на конгресі на тему ІТ автомобіля: «Без створення приєднаних автомобілів – неможливо буде продавати транспортні засоби на світовому ринку». [9]

На рис. 1.1, показаному дочірньою компанією Deutsche Telekom, Detecon, підняття підключеного автомобіля показаний дуже чітко. Стає зрозуміло, що навіть під час буму Інтернету на рубежі тисячоліть деякі виробники автомобілів

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

все частіше боролися з темою приєднаних автомобілів. Після того, як бульбашка в Інтернеті тріснула, цю діяльність майже припинили. Головною причиною цього, ймовірно, була недостатня швидкість мобільного телефону (показана на рис. 1.1 як «швидкість передачі даних»). Оскільки новий стандарт мобільного радіомовлення UMTS, який тоді був надзвичайно відзначений, виявився недостатнім з точки зору пропускної здатності для застосувань у мережевих транспортних засобах.

Якщо подивитись на потенціал підключених автомобілів на основі транспортних засобів, які мають бути з'єднані в мережу на рис. 1.2, паралельно з розвитком ринку мобільного зв'язку та інтенсивно розширеною пропускною здатністю, ви можете побачити, наскільки великий ринок, і це в основному лише початок.

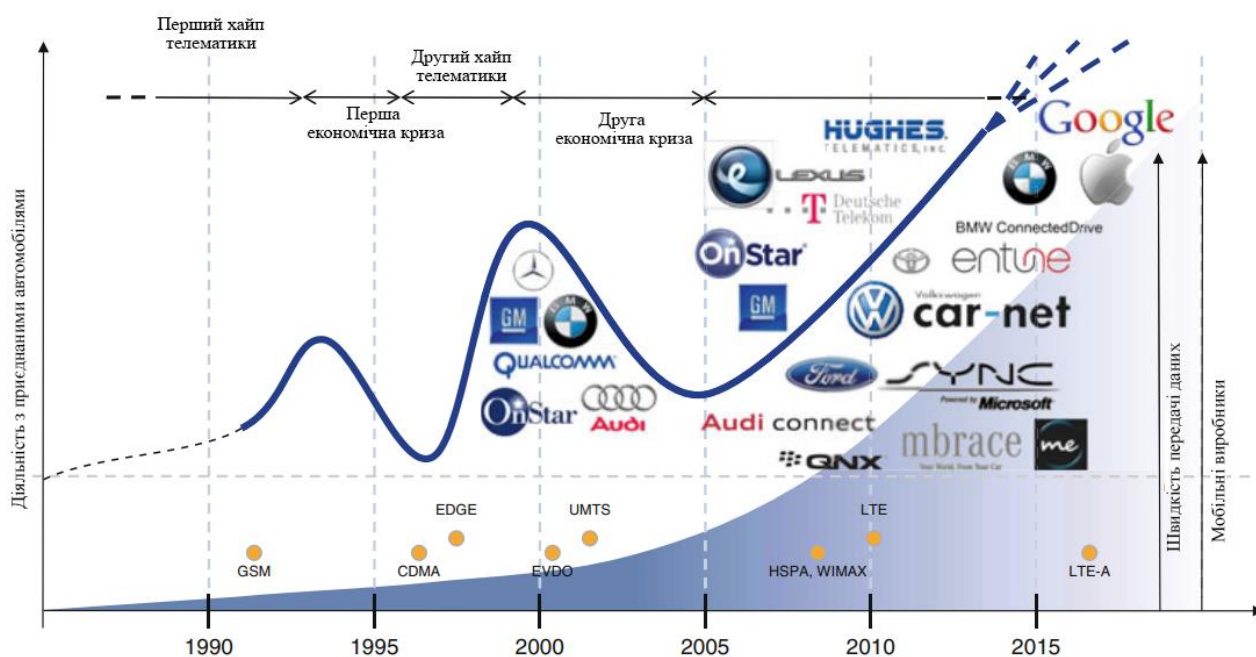


Рисунок 1.2.4.1 – Популяризація приєднаного автомобіля

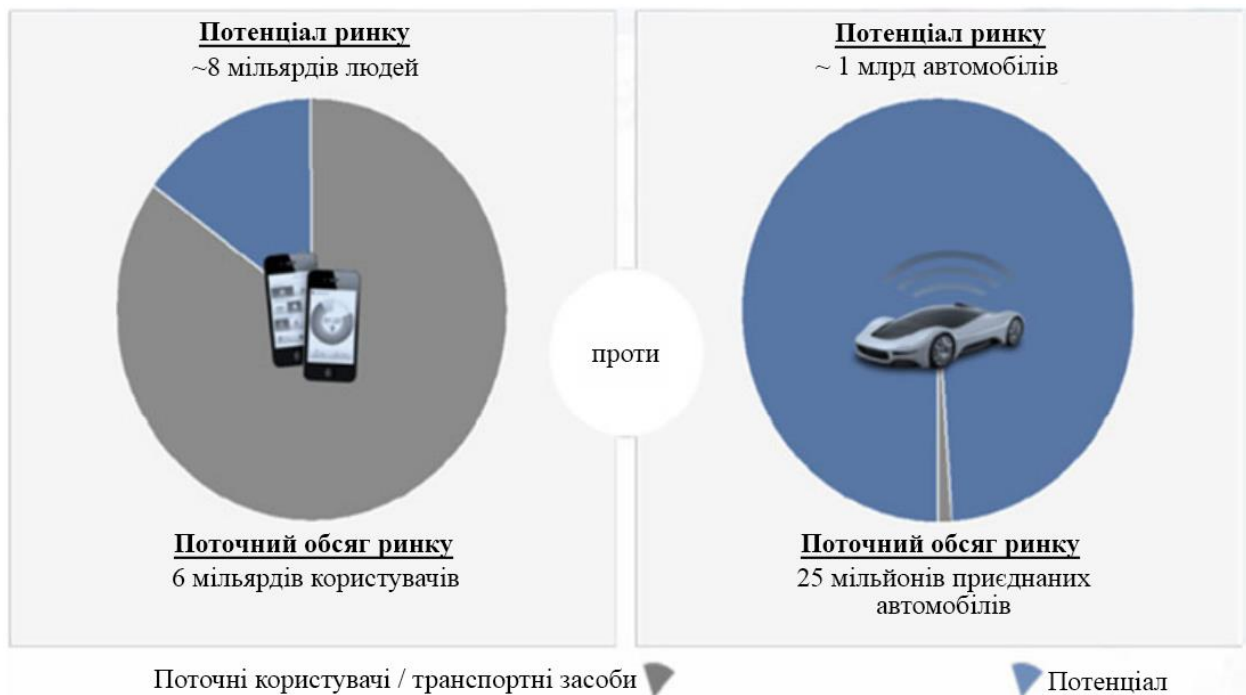


Рисунок 1.2.4.2 – Потенціал мобільного зв’язку проти потенціалу приєднаного автомобіля

1.2.5. Розвиток ІТ автомобіля

З одного боку, бізнес -модель повинна бути оцифрована в областях, які раніше були абсолютно нерелевантними, а з іншого боку, реальне мистецтво та якість автомобілебудування не повинні постраждати в результаті. Оцифрування та ІТ -технології для автомобілів не можна повністю передати на аутсорсинг, оскільки ці компетенції надто важливі для майбутнього і більш-менш вирішальні для ринку. Тим не менш необхідна міжгалузева співпраця. Виробникам автомобілів доведеться звертатися до постачальників бездротових послуг та програмного забезпечення, щоб мати можливість розвивати необхідні навички. Інститути, близькі до уряду, також незамінні для теми автономного водіння.

Наступні тези мають на меті показати, на що слід звернути увагу, щоб майбутній ринок ІТ автомобілів розширювався і не руйнував бізнес-модель автомобіля (на основі дослідження BVDW). [10]

ІТ автомобіль має бути основною компетенцією виробників автомобілів, оскільки в іншому випадку вони ризикують перейти від виробника до постачальника.

Прикладом такого сценарію є музична індустрія: відомій американській фірмі вдалося перетворити музичну індустрію на постачальника своїх продуктів та, насамперед, необхідної платформи.

Виробники автомобілів повинні стежити за тим фактом, що гравці з ІТ - індустрії, такі як Епл або Гугл, не беруть на себе ІТ-автомобіль, і що кращий дизайн, більш швидкі цикли продуктів та більш інноваційна робота стають настільки сильними, виходить що вони практично залучені до автомобілів. Купуйте виробників, оновлюйте їх технології та продавайте під своїм крилом. Тоді настільки важливий контакт з кінцевими клієнтами зникне, а активну лояльність до бренду більше не вдасться створити та підтримувати!

Значно різні цикли розвитку та життєвого циклу продукту між галуззю програмного забезпечення / мобільного зв'язку та автомобільною промисловістю створюють великий тиск на виробників автомобілів.

Цикли розвитку виробників автомобілів складають в середньому близько шести років і, отже, у багато разів довші, ніж цикл розвитку операційної системи для смартфонів, наприклад. Таким чином, ІТ-індустрія має важіль влади, який визначає тактову частоту для того, що клієнти вважають інноваційним і купують. Якщо з'явиться новий смартфон з новими функціями, автомобільній промисловості сьогодні потрібні роки для планування та впровадження подібних чи кращих функцій. В даний час ІТ -індустрія встановлює технічні стандарти, а автомобільній промисловості залишається лише роль «переслідувача». Сучасні клієнти не чекають роками функціональних можливостей автомобіля, які вони вже використовують на своїх смартфонах. Виробники автомобілів повинні, принаймні, досягти успіху у забезпеченні сумісності з цими споживчими пристроями, як з точки зору обладнання, так і програмного забезпечення. Оскільки динаміка розвитку більше не дозволяє довгострокового планування, вже зараз необхідно, щоб транспортні засоби залишалися гнучкими з точки зору технічних вимог, а також гарантували їх можливість модернізації та оновлення. Здатність діяти в мережі автомобільних та побутових пристроїв буде збережена лише в тому випадку, якщо автовиробнику вдасться погодити довгострокові

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

правила на ранній стадії. Якщо це не вдасться, майбутнє приєднаних автомобілів могли вирішувати не автомобільні компанії, а постачальники споживчих пристроїв.

Статус автомобіля втрачає цінність, а зв'язок та практична мобільність виходять на перший план.

Готовність молодих споживачів купувати транспортні засоби падає. Натомість у центрі уваги – спонтанна мобільність. Ця соціальна група, зокрема, використовує різні пропозиції – від ділення автомобілями до автобусів міжміського сполучення. У майбутньому простий і комплексний доступ до таких послуг та зв'язку, які набуваються у зв'язку з купівлею транспортного засобу та користуванням послугою мобільності вирішальний. Через зростаючий зв'язок все більшу роль відіграє зростання загальної пропозиції у сфері інформаційно-розважальної та розважальної сфери для виробників транспортних засобів та постачальників послуг мобільного зв'язку. На початку це вже проявляється у зростаючій доступності Вай-Фай у поїздах, літаках, автомобілях та автобусах. Крім того, постійне збільшення пропозицій окремого контенту, що супроводжує відповідне обладнання для мобільних пристроїв, набуває все більшого значення.

У майбутньому післяпродажне обслуговування та периферійні послуги принесуть більший дохід, ніж продаж самого автомобіля. У епоху цифрових технологій додана вартість все більше досягається за межі реального продукту за допомогою послуг (наприклад, надання контенту), які доступні на основі контексту використання. Автомобіль тоді служить доступом до екосистеми послуг мобільності. Моделі з оплатою за користування (оплата за відповідне використання) та передплата на послуги-це нові сфери бізнесу для цифрових пропозицій у сфері автомобілів та індивідуальної мобільності.

Правильний вибір партнерів є важливим фактором успіху для виробників автомобілів. Виробники транспортних засобів повинні визнати потенціал приєданого автомобіля та активно допомагати його формувати. Якщо цей варіант не використовується, його використовуватимуть компанії поза галуззю. Існує ризик того, що автомобіль стане просто платформою без значного

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

збільшення продажів для OEM. Вони повинні приймати правильні рішення щодо своєї ролі між контролем та інноваційним керівництвом. У разі власних розробок, повний контроль над позиціонуванням, витратами та якістю залишається в першу чергу за OEM, але це може призвести до довгих циклів розвитку, звичних для автомобільної промисловості, що уповільнює інноваційну силу. Частина аутсорсингу ланцюжка вартості може розвиватися швидше та ефективніше.

Просування екосистем проти втрати контролю: Якщо OEM-виробники вирішать працювати з цифровими експертами (підхід відкритого магазину), необхідно зважити можливості та ризики. З одного боку, можна заохочувати цифрові екосистеми, ефективно використовувати кошти та досягати високого рівня динамізму та інноваційної сили для OEM-виробників; Таким чином, управління та контроль суттєвих параметрів нових бізнес-моделей передаються на сторонніх підрядників.

Постачальникам контенту доводиться будувати нові медіа-мости в автомобілі. Сьогодні в автомобілях люди в основному слухають радіо або відтворюють власні списки музичних треків. Постачальникам контенту необхідно розвивати свої продукти таким чином, щоб споживання цифрових продуктів можна безперервно продовжувати в автомобілі та враховуючи конкретну ситуацію використання. Перш за все, мова йде про розумне перетворення текстового вмісту, такого як новини, RSS-канали, твіти, повідомлення про статус в соціальних мережах тощо у аудіопродукцію.

Приєднаний автомобіль відкриває доступ для компаній, які раніше не були в галузі, і пропонує платформу, яка допомагає формувати ринок автомобілів. Розробники та постачальники поза галуззю можуть стати активною частиною автомобільного ланцюжка створення вартості. Цифрова екосистема зростає і стає все більш і більш складною, але є різні точки зору на автомобіль: OEM виробники розглядають транспортний засіб насамперед як засіб пересування: Завдання транспортного засобу – це максимально ефективно, комфортно та швидко якомога більше доправити від точки А до В. Однак, оскільки це стає все

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

більш зрозумілим, це відходить на другий план. З іншого боку, телекомунікаційні компанії розглядають автомобіль насамперед як мобільний пристрій, що рухається: подібно до інших пристроїв, для забезпечення зв'язку потрібна SIM-карта (від відповідного постачальника). Таким чином, транспортний засіб / автомобіль відкриває потенціал як пасивного користувача, а також активного виробника мобільних потоків даних. Інноваційні цифрові компанії розуміють автомобіль як «п'ятий екран», тобто як еквівалент мобільних пристроїв. Дані відіграють вирішальну роль у перегляді доступного вмісту та його пошуку, а також у процесах взаємодії між людиною та машиною на вторинному рівні використання. Роздрібна торгівля (стаціонарна та електронна комерція) формує останню групу інтересів. Автомобіль розглядається тут як подальша точка дотику між клієнтами та дилерськими центрами і відповідно пропонує нові послуги та потенціал продажів – наприклад, підтримується Службами локації (LBS).

1.2.6. Нові моделі комунікації

Концепція ІТ автомобіля робить більше, ніж просто вводить Інтернет в автомобіль. Нові моделі спілкування з'являються між транспортним засобом та багатьма іншими речами. Мова йде про зв'язок Автомобіль-Х, за допомогою якого “Х” може означати автомобіль, інфраструктуру чи інші речі, як буде детальніше пояснено нижче.

1.2.6.1. Комунікація Автомобіль-Автомобіль

Зв'язок «автомобіль-автомобіль»-скорочено «А-А» або в англійських країнах під назвою «Vehicle2Vehicle, скорочено V2V-це прямий обмін інформацією між транспортними засобами. Мета зв'язку А-А – якомога раніше попередити пасажирів автомобіля, щоб уникнути аварій та запобігти критичним маневрам водіння. Інша мета-оптимізувати рух транспорту за рахунок швидкого та раннього обміну інформацією про затори, рух «зупинити-йти» або

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

несприятливі погодні умови, такі як сильний дощ з ризиком аквапланування або утворення льоду.

Принцип роботи Автомобіль-Автомобіль ґрунтується на поєднанні інформації з електричних та електронних датчиків, а мережа автомобіля з Інтернетом може обмінюватися інформацією між транспортними засобами. Типовий приклад – раптове гальмування транспортного засобу через раптову перешкоду на дорозі (наприклад, під час перетину оленів або коли діти грають м'яча). Наступні транспортні засоби негайно повідомляються про цю перешкоду і можуть негайно на неї реагувати.

1.2.6.2. Комунікація Автомобіль-Інфраструктура

Транспортні засоби спілкуються з інфраструктурними об'єктами, такими як системи контролю руху або системи світлофорів (в англомовних країнах це називається «Vehicle2Roadside» - скорочено V2R). Новою бізнес-моделлю в цьому середовищі є, наприклад, «платіть за їзду», в якому страховка оплачується лише за пройдені кілометри. Автоматичний дисплей і в майбутньому паркування на вільних місцях, паркування також є частиною спектру Автомобіль-Інфраструктура.

Подальшими варіантами застосування Автомобіль-Інфраструктура можуть бути такі моделі зв'язку зі світлофорами або системами управління дорожнім рухом:

- Раннє розпізнавання та зчитування фаз світлофора та відповідна реакція на них
- Інформація з «розумних» дорожніх знаків, таких як знаки зупинки
- Інформація про затори чи аварії на певних маршрутах
- Дані датчиків про температуру, вологість та видимість на маршрутах, включаючи відповідне попередження до відповідних систем допомоги у транспортному засобі
- Ранні попередження, коли наближаються автомобілі швидкої допомоги або пожежні машини, щоб якнайшвидше розчистити для них аварійну смугу

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.6.3. Комунікація Автомобіль-Дім

Завдяки комунікації Автомобіль-Дім автомобіль та всі існуючі домашні програми вступають у контакт. Зокрема, засоби масової інформації, які використовуються вдома та в дорозі, тепер також доступні у транспортному засобі. Це включає, наприклад, потокове передавання музики, аудіокниг, фільмів або фотографій з домашньої колекції. Маршрут, запланований на домашньому комп'ютері чи планшеті під час подорожі, також можна передати безпосередньо на навігаційний пристрій у транспортному засобі. Деякі виробники автомобілів уже пропонують так звані «Інтернет-журнал реєстрацій». Кожну поїздку можна автоматично записати у фоновому режимі та зберегти в онлайн -журналі реєстрацій. І навпаки, також можна дистанційно запитувати стан автомобіля: за допомогою програми або порталу для клієнтів можна дізнатися про рівень палива чи масла, а також температуру всередині та зовні.

1.2.6.4. Комунікація Автомобіль-Підприємство

Комунікація Автомобіль-Підприємство охоплює зв'язок між транспортними засобами та інфраструктурами, які експлуатуються приватно та комерційно (на відміну від Автомобіль-Інфраструктура, де задіяна державна та суверенна інфраструктури). Інфраструктура Автомобіль-Підприємство включає автосервіси, автостоянки, автозаправні станції, готелі та ресторани. Можливими застосуваннями є, наприклад, зв'язок між транспортним засобом та гаражем. Перевіряється, чи є, і якщо так, де ще є вільне місце для паркування на багатоповерховій автостоянці. Потім транспортний засіб направляється туди за допомогою навігації. Процес оплати за гараж можна також автоматизувати, оскільки точно відомо, який автомобіль був припаркований і як довго.

1.3. Розвиток інтерфейсів користувача в автомобілі

За останні десятиліття салон автомобілів постійно змінювався. Хоча основні пристрої взаємодії – кермо та педалі – залишалися незмінними, інші частини розвивалися. Адаптуючись до таких подій, як автоматизація водіння, у

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

автомобіль були інтегровані нові вхідні та вихідні пристрої. Перспективною, дослідженою раніше модальністю є великі стереоскопічні панелі 3D. Замінивши традиційну приладову панель автомобіля великим дисплеєм та застосувавши бінокулярні глибини, такий інтерфейс користувача може дати нові можливості для досліджень та промисловості. Враховуючи попередню еволюцію інтерфейсу автомобіля та прогрес у техніці автомобіля, особлива увага повинна приділятися відволіканню водія, керування автомобілем вручну, довіра, маневри із заволодінням керування та завдання, не пов'язані з водінням.

З моменту винайдення першого бензинового автомобіля в 1880-х роках спосіб їхнього користування водіями постійно змінювався. [11] На рисунку 1.3.1 висвітлено цей розвиток подій, ілюструючи звичайний транспортний засіб з первинними елементами управління поряд з наступними поколіннями салонів автомобіля. В той час як ранні автомобілі мали мало елементів управління, пізніше інтерфейси користувачів мали безліч ручок і кнопок. Сучасні автомобілі демонструють небагато з них, але пропонують багато функцій, інкапсульованих у цифрові інтерфейси користувача, такі як сенсорні екрани. У той час як основні елементи інтерфейсу користувача – кермо та педалі – здається, сьогодні значно покращені, решта елементів керування змінилися, стали складнішими та еволюціонували. Оскільки спочатку майже не було вторинного управління, сучасні автомобілі часто мають складні пристрої введення та виведення. Наприклад, модель Тесла 3 (див. Рис. 1.3.1, в) пропонує кілька механічних кнопок, але досить великий сенсорний екран з високою якістю графіки та складною структурою меню, навіть з повітряними жестами. Сучасні автомобілі також обладнані різними пристроями виведення, такими як аналогові датчики, різні цифрові екрани та покажчики. Ці вдосконалені умови були необхідні з різних причин: у автомобілях були введені розширені системи допомоги водінню, які пропонують варіанти налаштування, такі як спеціальні режими водіння (наприклад, економічне водіння) або різні налаштування підвіски (наприклад, спортивний режим). Крім того, все більше і більше функцій комфорту та розваг було інтегровано в автомобілі. Наприклад, системи

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

кондиціонування та мультимедіа, включаючи радіо, а також екрани для перегляду фільмів пасажирами. Оцифрування та мобільний зв'язок ще більше доповнюють цілий ряд функцій. Наприклад, у багатьох автомобілях телефон можна підключити до бортового комп'ютера, щоб просто пересилати сповіщення чи навіть використовувати як розширення інтерфейсу користувача, наприклад для навігації або дзвінків. Можна стверджувати, що цей розвиток триватиме, і що природа інтерфейсів користувача в автомобілі буде розвиватися далі.



Рисунок 1.3.1 – Розвиток інтерфейсів користувача автомобіля

Згадані приклади показують, що під час керування автомобілем все ще є основною метою автомобіля, дослідження та промисловість розробили салон автомобіля як простір, що пропонує затишне середовище та дає доступ до різних мультимедійних функцій. До цих пір основною метою проектування автомобільних інтерфейсів користувача в автомобілі була мінімізація відволікання водія та максимальна безпека. Поки водіння стає дійсно безпечнішим – принаймні, згідно зі статистикою Німецької ради безпеки дорожнього руху – проблема ще далеко не вирішена, як показує все ще велика кількість 3,275 смертельних аварій на німецьких дорогах. [12] Таким чином, безпека все ще залишатиметься питанням у майбутньому розвитку автомобільного інтерфейсу користувача. Тут однією з основних проблем є безпечне управління інтерфейсом користувача під час руху. Щоб уникнути аварій, водії повинні повністю зосередитися на дорожньому русі. Однак вони також хочуть керувати інтерфейсом автомобіля, щоб керувати радіо, регулювати кондиціонер тощо. Крім того, їм потрібно контролювати критично важливу

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

інформацію, таку як навігація або швидкість автомобіля. Ця взаємодія відволікає їх від керування автомобілем і є основною причиною ДТП. Наприклад, 8.9% смертельних катастроф у США були пов'язані з відволіканням водіння. [13] Отже, хороший інтерфейс повинен підтримувати та полегшувати взаємодію та зменшувати відволікання: пристрої введення повинні дозволяти швидку, точну, інтуїтивну взаємодію та – у кращому випадку – не потребуватимуть довгих поглядів на елементи управління. Вихідні пристрої повинні демонструвати інформацію таким чином, щоб її можна було легко і швидко сприймати та розуміти. Таким чином, пристрої введення та виведення можуть сприяти безпеці водія під час керування транспортним засобом.

Окрім питання безпеки, останні розробки в галузі автоматизації водіння становлять ще одну проблему для інтерфейсів автомобіля: умовні, високоавтоматизовані та повністю автоматизовані транспортні засоби вимагають користувацьких інтерфейсів, спеціально створених для цих режимів водіння. Наприклад, якщо автомобіль їздить сам по собі, важливо, щоб користувач довіряв системі автоматизації, щоб відчувати себе комфортно та безпечно. Низький рівень довіри до автоматизації може взагалі запобігти використанню системи або призвести до зловживання. Користувальницькі інтерфейси можуть сприяти довірі, наприклад шляхом поведінки системи комунікацій. Крім того, згідно з класифікацією Товариства інженерів автомобільної промисловості, під час керування транспортним засобом 3-го рівня автоматизації водії повинні бути готові брати на себе завдання керування у будь-який час, поки автомобіль керує самостійно. У кращому випадку водії завжди стежать за дорогою та наглядають за автомобілем. Таким чином, вони здатні правильно реагувати і при необхідності взяти на себе контроль. Однак було показано, що люди виконують завдання нижчого рівня при виконанні завдань моніторингу. Вони легко втрачають усвідомлення ситуації, що може призвести до потенційно небезпечних реакцій у критичних ситуаціях. Розширені інтерфейси користувача, такі як навколишнє освітлення або тактильні сигнали, можуть підтримувати цей процес поглинання. На додаток до цього, автоматизація автомобіля дозволяє

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

виконувати завдання, не пов'язані з керуванням автомобілем. Наприклад, за допомогою автоматизації 4-ого рівня автомобіль може їздити сам, і ніякий нагляд не потрібен. Після цього автомобіль повністю виконує завдання за кермом, і не потрібна раптова реакція. Під час керування таким транспортним засобом водій-пасажир завжди матиме відповідний час, щоб взяти на себе завдання керування, але цей процес некритичний. Таким чином, інтерфейс користувача може бути повністю присвячений завданням, не пов'язаним з керуванням автомобілем та при необхідності повернутися до режиму водіння. Наприклад, водії-пасажир майбутніх високоавтоматизованих транспортних засобів могли переглядати Інтернет у вільний час, але також могли необхідно виконувати певні робочі завдання, наприклад брати участь в онлайн-тренінгах або обробляти дані. Сучасні транспортні засоби не були розроблені для такого роду ситуації і часто перешкоджають ергономічному виконанню завдань, не пов'язаних з керуванням автомобілем.

В цілому можна припустити, що інтерфейси користувача в автомобілі відіграватимуть ще більш важливу роль у майбутньому, і що вони повинні розвиватися, щоб відповідати (або йти в ногу з) обговорюваними проблемами: крім аспекту безпеки, цілі майбутнього розвитку інтерфейсу – це серед іншого, спроектувати їх таким чином, щоб автоматизація довіри користувачів була достатньо добре побудованою та підтримуваною, щоб вони підтримували критичні ситуації користувачів, коли автоматика виходить з ладу, і щоб вони підтримували завдання, не пов'язані з керуванням автомобілем, коли автомобіль виконує завдання керування автомобілем.

Можна стверджувати, що традиційні інтерфейси користувача не можуть виконати ці вимоги. З одного боку, зменшення відволікання уваги водія є невід'ємною частиною інтерфейсу автомобільних користувачів, але це практично не вирішується, як показує статистика. З іншого боку, сучасні користувацькі інтерфейси не були розроблені для нових викликів, які викликає автоматизація, згадана раніше. Хоча сучасні автомобілі пропонують відносно безпечний досвід, вони не обов'язково сприяють довірі до автоматизації або підтримують процеси

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

поглинання. Крім того, салони транспортних засобів ще не спроектовані таким чином, щоб вони могли виконувати роботу або пов'язані з благополуччям завдання, не викликаючи дискомфорту або несуть загрозу безпеці. На додаток до цього, проблема безпечної видачі та обробки надзвичайно важливого часу поглинання досі не вирішена. Потрібні такі користувальницькі інтерфейси, які можуть відповідати цим вимогам: підвищують безпеку, підтримують прийняття керування на себе, дозволяють виконувати завдання, не пов'язані з керуванням автомобілем, та сприяють розвитку довіри.

Можливим вирішенням цих проблем може бути третій вимір: стереоскопічний. Інтеграція бінокулярних сигналів глибини в дисплеї автомобіля може потенційно забезпечити нові можливості дизайну для досліджень та промисловості. Додавши ще один вимір до дизайнерського простору, інформація може бути представлена таким чином, щоб бути ближче до реальності. Тим самим це могло б додати значення до інтерфейсу користувача: наприклад, бінокулярна глибина могла б виступати як новий елемент дизайну та сприяти розумінню навігаційної інформації. Або, якщо такий дисплей S3D забезпечує велику інтерактивну робочу область (подібну до настільної), інтерфейс автомобіля може дозволити користувачам виконувати різні завдання не пов'язані з керуванням автомобілем. Тоді користувачі могли б виконувати розважальні завдання, такі як перегляд фільму або виконувати професійні завдання, такі як науково-дослідницька діяльність, онлайн-навчання або повноцінне виконання наукових робіт. Попередні дослідження стосувалися двох невід'ємних аспектів цього підходу: з одного боку, малі та неінтерактивні стереоскопічні 3D-дисплеї, а з іншого боку, великі 2,5D-інтерактивні поверхні. [14, 15]

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. СИСТЕМА ТА АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ

2.1. CAN шина

Controller Area Network - локальна мережа контролерів – CAN шина - це протокол зв'язку, розроблений компанією Р.Бош GmbH на початку 1980-х років. Основною метою проекту було підтримка надійних програм в автомобілях. Потім протокол був представлений на ринку у співпраці Бош та Інтел. У 1990 році Mercedes Benz був першим виробником автомобілів, який застосував протокол мережі контролерів у серійному виробництві, в автомобілі S-класу для створення мережевої електроніки. Першими постачальниками модулів для мережі контролерів на той час були Інтел та Моторола. В даний час майже всі виробники напівпровідників незначних і великих розмірів мають у своєму портфоліо продукцію для мережі контролерів. У 1997 році за 1 рік було створено 24 мільйони інтерфейсів, через 2 роки їх було вже втричі більше. Нині, ймовірно, буде більше ніж мільярд модулів для мережі контролерів на рік.

На етапі впровадження нової шини наприкінці 80-х та на початку 90-х були запропоновані переважно так звані «автономні» рішення, які можна було легко підключити до будь-якого типу мікроконтролера. Ця стратегія сприяла широкому застосуванню нової шини, оскільки застосування мережі контролерів не обмежувалося лише сімействами мікроконтролерів першого покоління у виробництві напівпровідників. Починаючи з 90-х років на ринок виходило все більше і більше інтегрованих рішень – мережа контролерів разом з мікроконтролером на одному єдиному чіпі. Однак так званий трансивер - інтерфейс між аналоговими сигналами на лініях шини та цифровими сигналами контролера мережі контролерів - навіть досі, як правило, реалізується в окремій мікросхемі через свою різну напівпровідникову технологію. Інтегрована версія мережі контролерів та мікроконтролера дозволяє економити розміри кристала та з'єднувальних елементів, підвищує надійність та зменшує вартість. Ці фактори та сама якість є деякими з причин переважного поширення цієї шини на ринку. В

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

даний час комунікаційні мережі контролерів широко застосовуються в автомобілях, а також у авіатехніці, поїздах, військовому обладнанні, промисловому управлінні тощо. Великий інтерес до мережі контролерів відображається величезною кількістю публікацій, що стосуються цього інтерфейсу.

2.1.1. Історія мережі контролерів

Рушійним моментом в автомобільній промисловості для розвитку мережі контролерів (CAN) та інших так званих протоколів шин стала необхідність пошуку нових рішень для проблемної сфери, що призвело до зростання потреби в комунікації в транспортних засобах - граничною умовою пов'язаного з цим неминучого збільшення складності та кількості проводки в автомобілі. Електропроводка в автомобілях може мати довжину кабелю в декілька кілометрів і може бути важчою сотні кілограм. Зрозуміло, що такі джгути проводів складно встановлювати.

До того ж, наприклад, сімейний автомобіль вимагає для покриття варіантів лише однієї моделі сотні різних типів джгутів проводів. Це дає досить чітке розуміння, яка проблема існує з кабелями, також, у поєднанні з вчасною логістикою і, звичайно, з питанням зниження витрат. З початку 1980-х років майже всі виробники автомобілів почали робити інтенсивні зусилля щодо пошуку або розробки відповідних протоколів зв'язку. Одна програма - так звана програма класу C - спрямована на зв'язок між критичними блоками управління в режимі реального часу для управління двигуном, трансмісією, гальмами, стабілізацією автомобіля тощо. Будь-яка інша програма - так звана програма класу A - звертається до області електроніки кузова автомобіля, наприклад для управління дзеркалами, електросклопідійомниками, замками дверей, регулюванням сидінь, клімат-контролем, освітленням тощо. Остання область є менш критичною в режимі реального часу, отже, вона потребує меншої

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

пропускної здатності зв'язку, але, однак, надзвичайно чутлива до витрат. У відповідь на потреби різних класів додатків з'явилися різні протоколи шини, такі як мережа контролерів, локальна мережа взаємозв'язку (LIN - Local Interconnect Network), мережа транспорту орієнтованого на медіа (MOST- Media Oriented Systems Transport) і FlexRay.

FlexRay - це протокол комунікації автомобільних мереж, розроблений Консорціумом FlexRay для управління бортовими автомобільними обчисленнями. Він розроблений, щоб бути швидшим і надійнішим, ніж мережа контролерів та ТТР(протокол, що запускається за часом), але також він є дорожчим. Консорціум FlexRay розпався у 2009 році, але тепер стандарт FlexRay є набором стандартів ІСО від ІСО 17458-1 до 17458-5. FlexRay - це комунікаційна шина, призначена для забезпечення високої швидкості передачі даних, відмовостійкості, роботи в часовому циклі, розділеної на статичні та динамічні сегменти для зв'язку, що викликається подіями та запускається за часом.

З 1994–1995 рр. мережа контролерів є найпоширенішим автомобільним протоколом і описаний у стандартах Міжнародної організації зі стандартизації (ІСО) 11898-1 до 11898-5. У першій частині ІСО 11898 (ІСО 11898-1) рівень каналу передачі даних та фізичний рівень встановлюються відповідно до еталонної моделі ІСО ІСО/ІЕС 7498-1. [16] Доступ до високошвидкісної мережі контролерів (до 1 Мбіт/с) визначено в ІСО 11898-2 і в основному використовується для руху автомобіля. Мережа контролерів «Низької швидкості» (40... 125 кбіт/с) використовується для секції комфорту та описаний в ІСО 11898-3. ІСО 11898-4 дозволяє здійснювати зв'язок за часом, щоб забезпечити безперебійну передачу даних з високим трафіком зв'язку. Доступне з 2007 року розширення ІСО 11898-2 описує поведінку високошвидкісного вузла мережі контролерів у режимі енергозбереження. З початку 1990-х років мережа контролерів спочатку використовувалась в автомобілях серійного виробництва, а також у промисловому управлінні, зазвичай у поєднанні з вищим протоколом CANopen. CANopen зазначено у Європейському стандарті EN 50325-4 і

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

підтримується організацією користувача СіА (мережа контролерів в автоматизації). Причинами використання в серійному виробництві, безумовно, є високі стандарти роботи, низькі витрати та широкий асортимент виробників напівпровідників, які пропонують модулі цієї шини. Цей протокол зв'язку був вперше використаний в Мерседес S-класу в 1992 році і служив там як високошвидкісна мережа для зв'язку між управлінням двигуном, управлінням трансмісією та щитком приладів. Друга мережа контролерів - на низькій швидкості - використовувалася для розподіленого клімат-контролю. Незабаром після цього виробники BMW, Porsche, Jaguar тощо ввели цей інтерфейс в свої серійні автомобілі.

Типові застосування мережі контролерів:

- Транспортування із застосуванням мережі контролерів: легкові автомобілі, вантажівки, літаки, поїзди, кораблі, сільськогосподарська техніка, будівельна техніка тощо.
- Технологія автоматизації промислового управління з комунікацією мережею контролерів, такі як: програмовані контролери, автоматичне обладнання для роботи, інтелектуальні контролери двигунів, інтелектуальні датчики/приводи, гідравлічні системи, інтелектуальні лічильники для різних застосувань, таких як споживання води, споживання енергії тощо, текстильне застосування, таке як прядіння, ткацтво тощо, медичні технології, автоматизація лабораторій, будівельні системи, ліфти, банкомати, іграшки, механічні інструменти та багато іншого.

Нині все більш важливим є шлях посилення законодавчих норм щодо скорочення викидів CO₂ та споживання палива нових транспортних засобів. Водночас із скороченням термінів розробки слід підвищити ходові якості та комфорт автомобіля, зменшити витрати та покращити якість. Таким чином, у розвитку електронних систем автомобілів завжди виникають нові виклики, деякі з них можливо вирішити лише за допомогою мережі високого рівня. Багато

					<i>123.УДК: 004.42[629.33/.34:53.083.7]</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		50

сучасних стандартних технік були б неможливі без використання та розповсюдження промислових мереж.

2.1.1.1. Стандарти мережі контролерів

Р.Бош GmbH починався як виробник електронних блоків управління (ЕБУ) в 1983 році з розробки протоколу зв'язку для систем різних шин, які спочатку були призначені лише для використання в автомобільному секторі. Результатом стала мережа контролерів. Перший чіп мережі контролерів був запропонований виробникам автомобілів наприкінці 1980-х років компанією Інтел. Мережа контролерів, нарешті, була стандартизована у листопаді 1993 року як ISO 11898 (Дорожні транспортні засоби - Обмін цифровою інформацією - Мережа контролерів (CAN) для високошвидкісного зв'язку).

Сімейство ISO 11898 [17] - стандарт описує архітектуру мережі контролерів з точки зору рівнів моделі Міжнародної організації стандартизації взаємозв'язку відкритих систем (ISO/OSI). Він визначає як частини фізичного рівня, так і частини шару зв'язку для швидкості передачі до 1 Мбіт/с.

Перша частина і друга частина разом відповідають першій версії стандарту:

- ISO 11898-1 Дорожні транспортні засоби - Мережа контролерів (CAN)

Перша частина: Рівень каналу передачі даних та фізична сигналізація.

- ISO 11898-2 Дорожні транспортні засоби — Мережа контролерів (CAN)

— Друга частина: Високошвидкісний блок середнього доступу.

Пізніше були розроблені наступні частини:

- ISO 11898-3 Дорожні транспортні засоби - Мережа контролерів (CAN) —

Третя частина: Низька швидкість, надійність, середньозалежний інтерфейс.

- ISO 11898-4 Дорожні транспортні засоби - Мережа контролерів (CAN) —

Четверта частина: Зв'язок, що запускається за часом.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

- ISO 11898-5 Дорожні транспортні засоби - Мережа контролерів (CAN) – П'ята частина: Високошвидкісний блок середнього доступу з режимом низької потужності.

Крім сімейства ISO 11898, в інших організаціях зі стандартизації існують інші стандарти для мережі контролерів. Прикладами цього є стандарти J2284-1 bis-3 Товариства автомобільних інженерів (SAE), які визначають фізичний рівень для мережі контролерів з різною швидкістю передачі даних 125 кбіт/с (J2284-1), 250 кбіт/с (J2284- 2) і 500 кбіт/с (J2284-3).

2.1.2. Технічні характеристики мережі контролерів

Першою публікацією мережі контролерів став документ на конференції SAE у Детройті (SAE 860391) у лютому 1986 р. Першою опублікованою версією всього протоколу була версія 1.1 специфікації. Далі протокол був розроблений з версією 1.2, а потім версія 2.0А та версія 2.0В, версії сумісні. Модулі мережі контролерів різних версій протоколів можуть, з невеликими обмеженнями, працювати в одній мережі. Мережа контролерів версії 1.2 дозволяє отримати більший допуск на осцилятор, ніж мережа контролерів версії 1.1. Мережа 2.0А ідентична версії 1.2, тоді як специфікація мережі 2.0В вводить, на додаток до тих повідомлень з 11-розрядними ідентифікаторами, визначеними в специфікації мережі 2.0А кадри з 29-розрядними ідентифікаторами, які називаються "Розширений кадр".

Протокол мережі контролерів був стандартизований ISO, у стандартах ISO 11519 для швидкості передачі даних до 125 кбіт/с та ISO 11898 для швидкості передачі даних до 1000 кбіт/с.[18] Пізніше ці стандарти були об'єднані в ISO 11898. ISO 11898-1 описує вищий рівень ISO/OSI протоколу до рівня каналу передачі даних, ISO 11898-4 описує параметр запуску часу в мережі (TTCAN), а інші частини стандарту описують різні варіанти нижчих шарів ISO/OSI, приймачів мережі контролерів та лінії шини мережі.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

2.1.2.1. Бітове кодування

Бітовий потік на лінії шини мережі контролерів кодується відповідно до методу неповернення до нуля (NRZ) з заповненням бітів. Це має перевагу, коли для передачі сигналу потрібна лише мінімальна пропускна здатність. З іншого боку, кодування неповернення, де бітовий рівень постійний протягом бітового часу, не містить інформації про тактовий сигнал. При великій довжині даних це може спричинити проблеми з синхронізацією і, отже, призвести до виявлення помилкових бітів, оскільки кожен вузол у мережі має власний генератор тактових частот.

Тому до основної частини потоку розрядів додаються додаткові біти. У тій частині потоку бітів, де застосовується метод заповнення, передавач після того, як надіслав п'ять послідовних бітів ідентичного значення, вставить додатковий біт зворотного значення в потік бітів. Приймач розпізнає послідовність з п'яти послідовних бітів однакового значення і відкине наступний біт. Через механізм заповнення бітів відстань між краями в потоці бітів становить не більше п'яти бітів. Краї використовуються для синхронізації локальних бітових годинників усіх вузлів мережі. Вибрана ширина інформації є компромісом між обмеженням довжини кадрів та діапазоном допусків тактових генераторів.

2.1.2.2. Кадри мережі контролерів

У протоколі мережі контролерів існує чотири типи кадрів: кадр даних, кадр помилок, віддалений кадр і кадр перевантаження. Тільки кадр даних транспортує дані повідомлень, інші кадри призначені для усунення несправностей, запуску та синхронізації.

Кожне повідомлення мережі контролерів поділено на різні поля певної довжини. Цими полями є, наприклад, кінець кадру, поле перевірки циклічної надмірності (CRC), поле даних та поле арбітражу. Арбітражне поле поєднує в ідентифікаторі пріоритет повідомлення з логічною адресою інформації.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

Кадр даних CAN 2.0A (11-ти бітний ідентифікатор)



Кадр даних CAN 2.0B (29-ти бітний ідентифікатор)



Рисунок 2.1.2.2.1 - Структура кадру даних мережі контролерів

Заповнення бітів застосовується до полів від початку кадру до поля CRC, у кадрах даних та віддалених кадрах. В інших полях немає набивання бітів, а також немає в кадрах помилок або фреймах перевантаження.

У мережі контролерів є два формати ідентифікаторів. Перший має довжину 11 біт, зазначений у мережі контролерів 2.0A, дозволяючи 2048 різних логічних адрес (0–2047). Це означає, що максимум 2048 різних повідомлень можна надіслати в мережу, якщо не проводиться додаткове розрізнення, наприклад, у байтах даних.

У деяких додатках це число може бути занадто маленьким. Через це специфікація була розширена, тому можливі повідомлення з загальною кількістю 29 біт ідентифікатора. Це дозволяє 2^29 (= 536.870.912) різних повідомлень, які називаються розширеними кадрами відповідно до специфікації мережі контролерів 2.0B.

Точна структура кадрів показана на рисунку 2.1.2.2.1.

Окремі бітові поля, спочатку для кадрів з 11-розрядними ідентифікаторами (також звані стандартними кадрами), описані нижче:

- Початок кадру = 1 біт = низький (домінантний):

Позначає початок кадру. Після простою на шині нижній край використовується для фазової синхронізації всіх вузлів мережі.

- Арбітражне поле = 12 біт:

Поле арбітражу містить ідентифікатор довжиною 11 біт, що є логічним варіантом оголошення та пріоритетом повідомлення. Чим меншим є числове значення Ідентифікатора, тим вище пріоритет.

- Після ідентифікатора слідує біт Запиту на віддалену передачу (RTR). Біт запиту на віддалену передачу ідентифікує це повідомлення як таке, що містить або самі дані, що належать до логічної адреси (RTR = низький = домінуючий), або як кадр, який сам по собі не містить даних, а лише запускає фактичний передавач цієї інформації для надсилання кадру з поточними даними. Біт запиту на віддалену передачу - це останній біт поля арбітражу. Запит (який називається віддалений кадр) та відповідь відповідного передавача даних - це два окремих повідомлення. Це означає, що одне або кілька повідомлень з більш високим пріоритетом можуть передаватися по мережі контролерів між запитом та відповіддю.

- Поле керування = 6 біт:

Перший біт у полі керування - це біт прапорця розширення ідентифікатора (IDE). У 11-розрядних кадрах ідентифікатора він низький (домінуючий), що означає, що ідентифікатор завершено. Наступний біт r0 зарезервований. Останні чотири біти поля керування містять код довжини даних (DLC) для наступного поля даних.

- Поле даних = 0–8 байт даних:

Містить фактичні дані повідомлення.

- Поле перевірки циклічної надмірності = 16 біт:

Містить контрольну суму для попередніх бітів кадру. Контрольна сума перевірки циклічної надмірності довжиною 15 біт (послідовність CRC) використовується лише для виявлення помилок, а не для виправлення помилок. Поліном CRC для мережі контролерів призводить до відстані Хеммінга 6, що означає, що в повідомленні може бути виявлено до п'яти однорозрядних помилок. Крім того, так звані пакетні помилки - безпосередньо послідовні помилки бітів - можна розпізнати до довжини послідовності CRC. Після контрольної суми слідує один біт, роздільник перевірки циклічної надмірності (високий = рецесивний).

- Поле підтвердження = 2 біта:

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

Усі вузли, які побачили синтаксично правильне повідомлення по мережі контролерів, підтверджують, посылаючи домінуючий рівень у так званому слоті АСК. Передавач посилає рецесивний рівень у слоті АСК і очікує, що цей рецесивний рівень переписеться домінуючим рівнем. Передавач повідомлення вважає відсутній домінуючий рівень у слоті АСК помилкою підтвердження. Виявлення цього домінуючого рівня не означає, що кадр був отриманий усіма одержувачами цієї інформації! Він лише стверджує, що кадр був розпізнаний принаймні одним активним вузлом як правильний кадр мережі контролерів. Після слота АСК слідує один біт, роздільник АСК (високий = рецесивний).

- Кінець кадру (EOF) = 7 біт (високий = рецесивний):

Кінець кадру (EOF) характеризується навмисним порушенням кодування. Відповідно до механізму заповнення бітів, такий біт буде вставлений після п'ятого рецесивного біта. У кінці кадру цього не відбувається, EOF вказує на кінець кадру даних.

- Міжкадровий простір (IFS) = 3 біта (високий = рецесивний):

Міжкадровий простір (IFS) відокремлює поточний кадр від наступного кадру. У середині вузла CAN цей період також використовується для передачі правильно прийнятого повідомлення від контролера протоколу у відповідний простір буфера прийому або для передачі повідомлення від буфера передачі до контролера протоколу.

- Бездіяльність ≥ 0 біт (високий = рецесивний):

Шина не використовується. Кожен вузол мережі контролерів може, починаючи з біту "Початок кадру", надсилати повідомлення до шини.

Кадри з 29-розрядними ідентифікаторами (також звані розширеними кадрами) відрізняються від кадрів з 11-розрядними ідентифікаторами в полі арбітражу та в першому біті поля керування:

- Арбітражне поле = 32 біта:

Арбітражне поле кадрів з 29-розрядними ідентифікаторами складається з трьох частин. Він починається з 11 найбільш значущих бітів (MSB) ідентифікатора

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

(базового ідентифікатора), після чого йдуть два рецесивних (високі) біти: біт замітника віддаленого запиту (SRR) та біт прапорця розширення ідентифікатора (IDE). Потім ідуть 18 найменш значущих бітів (LSB) ідентифікатора (розширений ідентифікатор). З'єднані значення базового ідентифікатора та розширеного ідентифікатора є логічною адресою та пріоритетом повідомлення. Останній біт арбітражного поля - це біт RTR.

- Поле керування = 6 біт:

На відміну від кадрів з 11-розрядними ідентифікаторами, поле управління кадрами з 29-розрядними ідентифікаторами починається не з прапора IDE та одного зарезервованого біта, а з двох зарезервованих бітів, r1 та r0. Останні чотири біти поля керування містять DLC для наступного поля даних.

Дійсні значення для DLC - 0... 15. Значення DLC 8... 15 розглядаються як DLC 8, тобто максимальна довжина поля даних кадру становить 8 байт. Зарезервовані біти (r1 і r0) передаються домінантно (низько); приймачі приймають як домінуючі, так і рецесивні значення для r1 і r0.

Старі контролери мережі, розроблені до випуску CAN 2.0B, виявляли помилки в розширених кадрах і знищували їх за допомогою фреймів помилок, але ці контролери мережі більше не виробляються. Усі контролери мережі, які можуть обробляти розширені кадри, також можуть обробляти кадри з 11-розрядними ідентифікаторами. Таким чином, можна використовувати обидва формати кадрів в одній мережі контролерів. Обмежені реалізації мережі контролерів, які можуть надсилати та отримувати кадри з 11-розрядними ідентифікаторами, але ігнорувати розширені кадри, можуть бути використані для конкретних програм.

Біт IDE розрізняє формати кадрів. Цей біт є рецесивним у розширених кадрах; отже, розширений кадр програє арбітраж іншому фрейму з 11-розрядним ідентифікатором, ідентичним базовому ідентифікатору розширеного кадру.

Кадри даних, показані на рис. 2.1.2.2.1, є звичайними повідомленнями, які поширюють дані про роботу мережі. За звичайних обставин таке повідомлення

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

надсилається з ініціативи прикладної програми у передавальному вузлі. Крім того, мережа контролерів також надає можливість, що одержувач певної інформації може запропонувати відправнику передати фактичні значення даних. Це робиться за допомогою віддаленого кадру. Такий кадр характеризується рецесивним бітом RTR, а його поле даних - незалежно від DLC - не містить даних. Віддалений кадр запропонує відправнику передати відповідний кадр даних. Ця відповідь може бути виконана автоматично контролером мережі (з контролером "ПовнийCAN") або під управлінням прикладної програми (контролер "БазовийCAN"). Слід зазначити, що декілька повідомлень з більш високим пріоритетом можуть надсилатися на шину CAN між "чергою", тобто віддаленим кадром, і відповіддю, тобто кадром даних. Віддалений кадр може, наприклад використовуватись, коли одержувач був тимчасово неактивним, тобто не брав участі в шинному зв'язку, і потребує поточного значення даних до того, як його відправник досягне свого наступного запланованого часу передачі.

Структура віддаленого кадру показана на рис. 2.1.2.2.2. Структура кадру ідентична для мережі контролерів версії 2.0A та мережі контролерів версії 2.0B, за винятком довжини арбітражного поля.

Віддалений кадр завжди повинен надсилатися з тим самим DLC, що і відповідний кадр даних. Якщо більш ніж один вузол мережі контролерів одночасно запускатиме віддалені кадри з тим самим ідентифікатором, але з різними DLC, ці кадри знищуватимуть один одного.

Віддалений Кадр CAN 2.0A (11-ти бітний ідентифікатор)



Віддалений Кадр CAN 2.0B (29-ти бітний ідентифікатор)



Рисунок 2.1.2.2.2 - Структура віддаленого кадру мережі контролерів

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

2.1.2.3. Арбітраж

Доступ до мережі контролерів регулюється методом неруйнівного порозрядного арбітражу. Неруйнівний означає, що кадр, який є переможцем арбітражу, тобто, повідомлення з вищим пріоритетом - не порушується і його не потрібно перезапускати. Цей механізм вимагає, щоб відповідні фізичні драйвери були реалізовані певним чином: два логічних рівня на шині CAN повинні бути домінуючими та рецесивними, тобто, один вузол, посилаючи домінуючий рівень, перезаписує всі інші вузли, які посилають рецесивний рівень. У протоколі CAN логічний посилається рецесивним, а логічний нуль - домінуючим.

У багатомасштабних мережах «філософія» розподілу прав доступу на шині є вирішальним чинником, що характеризує пропускну здатність, затримку передачі і, отже, можливості мережі в реальному часі. Мережа контролерів показує дуже хороші показники в цьому плані: коли кілька вузлів одночасно намагаються отримати доступ до шини, автоматично вибирається найважливіший конкурент. Коли вузол мережі з очікуючим запитом на передачу виявляє, що шина вже зайнята, цей запит затримується до тих пір, поки шина не повернеться у стан очікування.

Приклад ілюструє арбітраж між двома конкуруючими вузлами мережі контролерів (див. Рис. 2.1.2.3.1).

Два вузли мережі контролерів А і В одночасно починають передачу. Відповідно до методу з багаторазовим доступом за чутливістю транспортера з виявленням зіткнення та арбітражем за пріоритетом повідомлень (CSMA/CD+AMP), обом вузлам довелося чекати, поки шина стане вільною (перевірка несучої). Коли це виявляється, вони обидва надсилають свій домінуючий біт початку кадру (множинний доступ). У всьому кадрі кожен вузол мережі за допомогою свого трансивера зчитуватиме логічне значення, що виникає на шині, і порівнюватиме його з переданим логічним значенням (виявлення зіткнення). Два домінуючих біта початку кадру накладаються на шину, а домінуючий рівень шини зчитується обома вузлами. Далі надсилається MSB ідентифікатор. У прикладі MSB є рецесивним в обох ідентифікаторах. І тут

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

рецесивний рівень з'являється на шині і знову розпізнається обома вузлами. Тому жоден вузол не помічає конкуруючий вузол до першої різниці в ідентифікаторі.

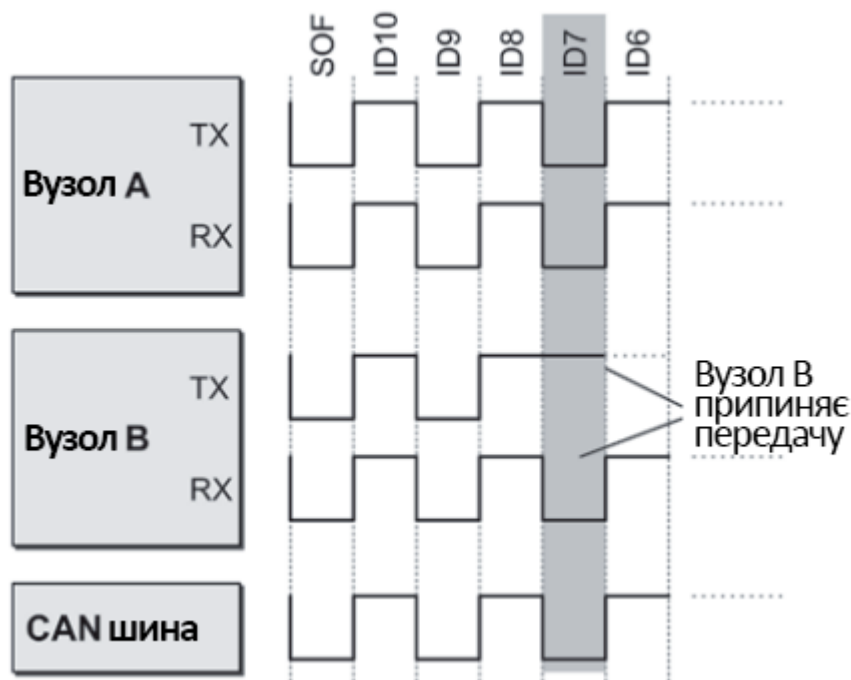


Рисунок 2.1.2.3.1 - Арбітраж в мережі контролерів

У прикладі перша відмінність між цими двома кадрами знаходиться в біті ідентифікатора 7. Вузол А посилає домінуючий рівень, вузол В - рецесивний. Відповідно до специфікації протоколу, на шині з'являється домінуючий рівень. Вузол А зчитує рівень, який він надіслав, отже, він не бачить зіткнення. Вузол В також зчитує домінуючий рівень і порівнює його з рецесивним рівнем, який він надіслав, тому він бачить бітову помилку. У цей момент вузол В визнає, що програв арбітраж (арбітраж щодо пріоритету повідомлення) і негайно припиняє передачу власного кадру. Крім того, вузол В стає одержувачем кадру вузла А, оскільки кадр, який виграв арбітраж, може містити дані, які необхідно обробити вузлом В.

Бітова помилка виявляється, коли рівень зчитування з шини відрізняється від рівня, що надсилається на шину. Якщо бітова помилка виявлена поза арбітражним полем (ідентифікатор і біт RTR) або слотом АСК, або якщо бітова

помилка є на домінуючому біті всередині арбітражного поля, це помилка, яка обробляється логікою управління помилками.

2.1.2.4. Керування помилками

Крім вищезгаданих бітових помилок, мережа контролерів також виявляє інші помилки і відповідно їх обробляє. У наведеному далі описі йдеться про управління помилками на рівні каналу даних. Це управління помилками також є частиною стандартизації в ІСО 11898, що означає, що кожна інтегральна схема мережі контролерів (ІС) має, реалізовану в кремнії, таку саму обробку помилок.

Принцип мережі контролерів передбачає, що якомога більше помилок слід виявляти та "лікувати" всередині інтегральної схеми мережі контролерів. Будь-яка помилка, виявлена мережевим вузлом, негайно повідомляється всім іншим вузлам. Після цього повідомлення про помилку всі учасники мережі відкидають поточне повідомлення. Виправлення помилок здійснюється за допомогою автоматичної повторної передачі, яка є функцією шару передачі даних.

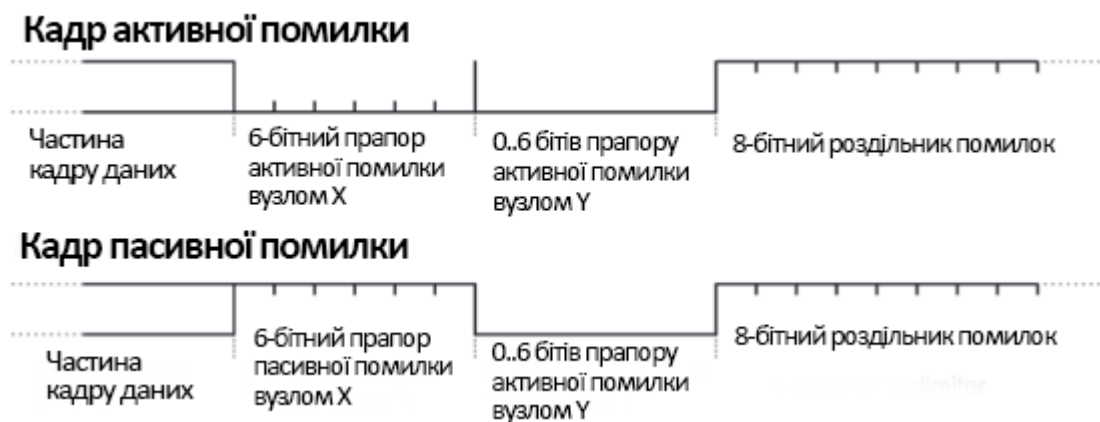


Рисунок 2.1.2.4.1 - Кадр помилки мережі контролерів

Повідомлення про виявлену помилку всім іншим учасникам здійснюється за допомогою фрейму помилок, який, як і кінець кадру, характеризується навмисним порушенням правил заповнення бітів. Структура такого кадру помилок показана на рис. 2.1.2.4.1.

Прапор активної помилки переписує всі інші поточні рівні шини з серією з шести або більше послідовних домінуючих бітів. Ця модель, порушуючи правила заповнення бітів, розпізнається усіма іншими вузлами мережі.

Щоб запобігти постійним локальним порушенням вузла мережі або групи вузлів мережі контролерів від постійного паралізування шини CAN з домінуючими прапорами активних помилок, уражені вузли мережі - відповідно до певного алгоритму - поступово виходять із діяльності шини CAN. Після першого етапу виведення з шини CAN вузол мережі контролерів може надсилати лише так звані прапори пасивних помилок (див. Рис. 2.1.2.4.1). Обмеження несправностей складається з трьох частин: виявлення помилок, обробка помилок та їх усунення.

Виявлення помилок. Управління помилками може визначити п'ять різних типів помилок:

- Біт-помилка

Переданий біт не приймається з тим самим логічним значенням, з яким він був надісланий. Виключені поля арбітражу та слот АСК.

- Помилка доповнення

Було виявлено більше п'яти послідовних бітів одного рівня. Виключені кінці кадру та міжкадровий простір (IFS).

- Помилка перевірки циклічної надмірності

Обчислена контрольна сума CRC не збігається з отриманою контрольною сумою CRC.

- Помилка форми

Сталося порушення формату кадру, наприклад, CRC розмежувач або АСК розмежувач не розпізнано як рецесивний біт, або кінець кадру порушено.

- Помилка підтвердження

Передавач не виявив домінантний біт у слоті АСК, що означає, що повідомлення не було розпізнано як безвідмовне будь-яким іншим вузлом мережі.

Обробка помилок. Після того, як одна з описаних вище помилок розпізнається, кожен вузол мережі повідомляється кадром помилки. Кадр

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

помилки перезаписує інші кадри і виявляється через порушення правил заповнення бітів. В результаті прапора активної помилки, який складається насправді лише з шести домінуючих бітів, інші вузли мережі контролерів бачать помилку доповнення і тому також починають надсилати фрейм помилки. Цей ефект виникає, коли помилка не є глобальною (виявляється одночасно всіма вузлами мережі), а є локальною помилкою (виявляється лише одним вузлом мережі або групою вузлів мережі контролерів) і призводить до можливого накладання прапорів помилок довжиною до 12 домінуючих бітів. Кадр помилок завершується роздільником помилок (серія з восьми рецесивних бітів).

Обробка помилок здійснюється в такій послідовності:

- Виявлено помилку.
- Кадр помилки надсилається будь-яким вузлом мережі контролерів, який виявив помилку.
- Поточне повідомлення відхиляється усіма вузлами мережі.
- На лічильники помилок у кожному вузлі мережі впливають відповідно до правил обмеження несправностей.
- Порушене повідомлення повторюється.

Ізоляція проблем. Протокол мережі контролерів визначає - з метою ізоляції несправностей - специфічні методи запобігання місцевим перешкодам, що порушують роботу шини CAN з кількома домінуючими прапорами помилок. Для цього для вузлів мережі контролерів визначено три стани:

- Помилка активна

Це нормальний стан вузла мережі контролерів, в якому повідомлення можна надсилати та приймати. У разі несправності передається прапор активної помилки, що складається з домінуючих бітів.

- Пасивна помилка

Цей стан досягається після того, як на шині було виявлено кілька помилок. У цьому стані вузол мережі контролерів може продовжувати надсилати та отримувати повідомлення, але у разі помилки надсилається лише прапор

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

пасивної помилки, що складається з рецесивних бітів. Отже, вузол мереж, що є пасивною помилкою, не може перешкоджати іншому трафіку мережі контролерів. Він буде позначати лише помилки у власних переданих повідомленнях. Це може статися, коли вузол бачить локальні порушення. Такий вузол мережі, який завершив перемикування власних переданих кадрів - після роздільника помилок - на 8 бітів у стан призупинення, де він не може розпочати кадр, але може приймати кожне повідомлення.

- Шина вимкнена

У цьому стані вузол мережі контролерів повністю відключений від шини. Тому він не може надсилати або отримувати повідомлення або прапори помилок. Цей стан досягається при тривалих перешкодах на шині або часто повторюваних порушеннях.



Рисунок 2.1.2.4.2 - Стани помилок мережі контролерів

Перехід від активної помилки до пасивної помилки і навпаки відбувається автоматично за допомогою інтегральної схеми мережі контролерів. Стан "шина вимкнена" може вийти лише відповідними діями хост-контролера (програмне або апаратне скидання). Переходи між станами показані на рис. 2.1.2.4.2.

Переходи стану помилок контролюються так званими лічильниками помилок. Кожен вузол мережі має лічильник помилок при отриманні помилок (Лічильник помилок прийому - REC) та лічильник помилок для помилок передачі (Лічильник помилок передачі - TEC).

Стани характеризуються такими значеннями підрахунку:

- Помилка активна

Підрахунок лічильника помилок прийому та лічильника помилок передачі менший або дорівнює 127.

- Пасивна помилка

Принаймні один з двох підрахунків помилок перевищує 127, а лічильник помилок передачі - менше 256.

- Шина вимкнена

Лічильник помилок передачі більше 255.

Лічильники помилок збільшуються за виявлених помилок або зменшуються за успішних повідомлень відповідно до правил обмеження помилок. Існують алгоритми управління помилками, які дозволяють вузлам мережі контролерів розрізняти локальні та глобальні помилки. Вузли, які бачать більше локальних помилок, досягають стану пасивних помилок раніше, ніж інші вузли мережі.

Правила обмеження несправностей пояснюються нижче (Лічильник помилок прийому - REC; Лічильник помилок передачі - TEC):

- Приймач виявляє помилку $\rightarrow REC = REC + 1$

за винятком того, що помилка - це бітова помилка, яка виявляється, коли одержувач надсилає прапор активної помилки або прапор перевантаження.

- Приймач виявляє домінуючий біт після надсилання прапору помилки $\rightarrow REC = REC + 8$.

- Передавач запускає прапор помилки $\rightarrow TEC = TEC + 8$

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

Виняток 1: Коли передавач знаходиться в стані «Пасивна помилка» і виявляє помилку підтвердження (немає домінантного біта в слоті АСК), і він не виявляє домінантний біт під час надсилання свого прапора пасивної помилки, ТЕК не змінюється. Виняток 2: Якщо передавач виявляє бітову помилку на рецесивному бітовому матеріалі в полі арбітражу перед бітом RTR, ТЕС не змінюється.

- Передавач виявляє бітову помилку під час надсилання прапора активної помилки або прапора перевантаження $\rightarrow \text{ТЕС} = \text{ТЕС} + 8$.

- Приймач виявляє бітову помилку під час надсилання прапора активної помилки або прапора перевантаження $\rightarrow \text{РЕС} = \text{РЕС} + 8$.

- Будь-який вузол мережі допускає до семи послідовних домінуючих бітів після надсилання прапора активної помилки, прапора пасивної помилки або прапора перевантаження. Після виявлення 14-го послідовного домінантного біта (у разі прапора активної помилки або прапора перевантаження) або після виявлення восьмого послідовного домінантного біта, що слідує за прапором пасивної помилки, і після кожної послідовності додаткових восьми послідовних домінантних бітів кожен передавач збільшується $\rightarrow \text{ТЕС} = \text{ТЕС} + 8$ і кожен приймач збільшується $\rightarrow \text{РЕС} = \text{РЕС} + 8$.

- Передавач з $\text{ТЕС} > 0$ зменшується після кожної успішної передачі кадру даних або віддаленого кадру $\rightarrow \text{ТЕС} = \text{ТЕС} - 1$.

- Приймач з $\text{РЕС} > 0$ зменшується після кожного прийому безвідмовного кадру даних або віддаленого кадру $\rightarrow \text{РЕС} = \text{РЕС} - 1$.

Перший виняток з третього правила було введено, щоб уникнути ускладнення фази ініціалізації мережі. Після запуску системи час ініціалізації окремих вузлів мережі контролерів у розподіленій мережі управління - за звичайних обставин - буде іншим. Якщо тепер припустити, що перший вузол мережі контролерів, який завершив свою ініціалізацію, надсилає кадр, він завжди отримає помилку підтвердження, оскільки немає іншого вузла мережі, готового дати підтвердження, при цьому передача за шаблоном повторюється автоматично. Це означатиме, що лічильник помилок передачі збільшується на 8

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

при кожній спробі, поки він, нарешті, не досягне порогу відключення шини і операція мережі не припиниться. Можна легко уявити, що ініціалізація мережі була б дуже складною за цих обставин. Тому ця помилка підтвердження трактується по-різному.

Після досягнення стану пасивної помилки лічильник помилок передачі більше не збільшується у разі помилки підтвердження якщо немає іншого вузла, що надсилає прапор активної помилки. Це означає, що перша спроба передачі буде повторюватися, поки другий вузол не завершить свою ініціалізацію і не дасть підтвердження.

Одна окрема помилка може спричинити застосування більш ніж одного правила, наприклад за таких обставин: якщо, наприклад, вузол мережі А, який наразі отримує фрейм, виявляє локальну помилку, а потім надсилає прапор активних помилок, цей прапор помилок не буде повністю перекриватися з прапорами помилок інших вузлів мережі. Інші вузли мережі не бачили локальної помилки, але прапор активної помилки змушує їх бачити помилку (або помилку форми), і тому вони починають відправляти свої прапори помилок пізніше. Це змушує вузол А виявляти більш домінуючі біти після надсилання свого шостого домінуючого біта прапора активної помилки. Перше Правило застосовується до всіх приймачів, а їх лічильники помилок прийому збільшуються на одиницю. Друге правило додатково застосовується до вузла А мережі, який бачив локальну помилку, отже, його лічильник помилок при отриманні збільшується загалом на 9. В результаті вузол мережі А досягне стану пасивної помилки раніше. Після того, як REC перевищив поріг пасивності помилок, він більше не збільшується.

Процес підрахунку помилок та зміна станів помилок показані на рис. 2.1.2.4.3. Прапор попередження про помилки в реєстрі стану мережі контролерів ІС встановлюється - і якщо цей параметр увімкнено, спрацьовує переривання - при досягненні порогового значення попередження 96 (отримати лічильник помилок або передати лічильник помилок), безпосередньо перед досягненням стану пасивної помилки.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

Коли лічильник помилок передачі досягає порогу відключення шини 256, прапор шини відключена виставляється в реєстрі стану мережі ІС, що, у свою чергу, може викликати переривання. Вузол мережі контролерів зараз перебуває у стані вимкненої шини і не буде брати участі в діяльності шини CAN до того, як він буде перезапущений хост-контролером.

Два вищезазначених прапорця попередження про помилки та вимкнення шини є мінімально необхідними засобами для хост-контролера для виявлення наявності шуму чи інших порушень. Їх перевага в тому, що хост-контролеру не потрібно дбати про нечасті несправності.

Більшість варіантів мікросхем мережі контролерів також дозволяють контролеру хосту мати прямий доступ для читання до лічильників помилок, а також інформацію про тип виявленої помилки; отже, для усунення несправностей програми можна вжити ранні заходи.

Описані заходи повністю є функціями рівня каналу передачі даних, тому вони реалізовані в апаратних засобах інтегральних схемах мережі контролерів. Це може, наприклад, означати, що навіть один несправний вузол мережі може вимагати повного вимкнення всієї системи в цілому, або це може вимагати переведення системи в безпечний стан.

У системах з меншим потрібним рівнем безпеки можуть бути впроваджені відповідні заходи, які принаймні дозволять екстрену операцію. Існують, наприклад, додатки, в яких вимірювання певного несправного датчика можуть бути повторені іншими датчиками, і таким чином загальна функція все ще виконується. Так само функцію виходу з ладу стоп-сигналу у транспортному засобі може взяти на себе задній протитуманний світловий сигнал. Усі ці заходи можливі лише завдяки використанню мережі.

Несправності в мережі контролерів можуть мати кілька причин, таких як електромагнітні перешкоди, коротке замикання, дефектні контакти або несправність самої інтегральної схеми мережі контролерів. Однією із заходів обробки помилок у додатку є повне вимкнення вузла мережі, який у розглянутому періоді занадто часто перемикається на стан вимкненої шини.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

Якщо першопричиною стала локальна помилка у відключеному вузлі мережі, цей захід дозволить решті мережі - і хостовому контролеру - продовжувати роботу, принаймні в аварійному режимі.

Обробка перевантажень. Іншим засобом обробки винятків є так званий кадр перевантаження. Структура фрейму перевантаження точно така ж, як і кадру помилок. Єдина відмінність полягає в тому, що кадр помилок перезаписує і знищує звичайні повідомлення, тоді як кадр перевантаження запускається виключно в IFS (див. Рис. 2.1.2.4.4).

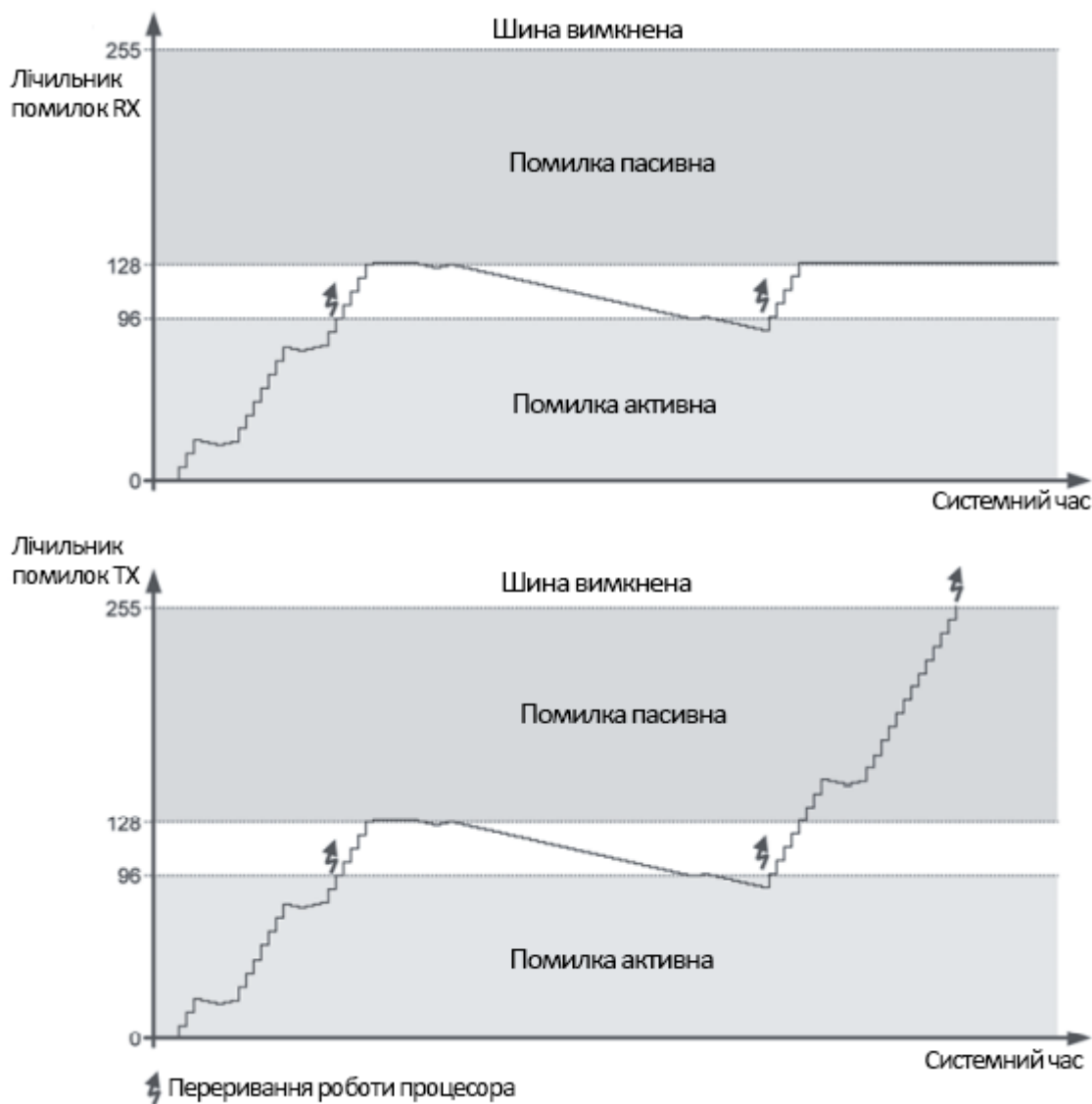


Рисунок 2.1.2.4.3 - Підрахунок помилок мережі контролерів

Існує дві умови для надсилання фрейму перевантаження. По-перше, кадр перевантаження надсилається, коли домінуючий біт бачиться в перших двох бітах IFS. Тут кадр перевантаження використовується для синхронізації всіх вузлів мережі. З іншого боку, контролер мережі може використовувати фрейм перевантаження, щоб повідомляти всі інші вузли мережі, що він перевантажений (не може обробляти нове повідомлення негайно), викликаний внутрішніми часами затримки (інтегральної схеми). Інші вузли мережі побачать перший домінуючий біт прапора перевантаження всередині IFS і відповідатимуть власними кадрами перевантаження, прапори перевантаження будуть перекривати один одного. У цьому випадку початок наступного кадру затримується на довжину кадру перевантаження. В обох випадках стан перевантаження не впливає на лічильники помилок. Сучасні контролери мережі досить швидкі, вони здатні обробляти весь трафік шини без додаткової затримки кадрів перевантаження.

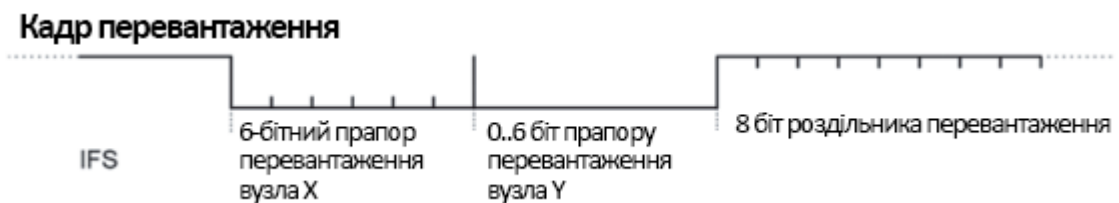


Рисунок 2.1.2.4.4 - Кадр перевантаження мережі контролерів

2.2. Вибір одноплатного комп'ютера

Одноплатні комп'ютери (SBCs – Single Board Computers) зазвичай складаються з того, що мають усі варіанти обробки та зв'язку на одній друкованій платі, в основному, застосовуються для цілей низької потужності та портативності. [19] Нині широко використовувані одноплатні комп'ютери такі як Arduino(Ардуіно), Raspberry Pi(укр. Малина Пай, РПай), Beagle Bone, одноплатний комп'ютер Nvidia, плата Asus Tinker board тощо. Ці плати працюють за допомогою Wi-Fi, а в деяких випадках - за допомогою комунікацій на основі порту COM. Одноплатні комп'ютери працюють за допомогою ОС на базі Linux, розміщеної на SD-картах.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		70

Образи операційних систем для плат зазвичай завантажуються з відповідних веб-сайтів і вставляються на SD-карту за допомогою програм для зчитування дисків Win32 або програм для записування образів. Плата живиться від 5 Вольт з потребою струму більше 2 Ампер, щоб забезпечити її повну працездатність. Потрібна обережність для живлення одноплатних комп'ютерів за допомогою відповідних зарядних пристроїв з належним обмеженням живлення. У деяких випадках ці пристрої підтримуються за допомогою дротового з'єднання ірv4. Плати, такі як Малина Пай 3, 3b+, Асус Тінкер і NVIDIA, мають опції підключення до Інтернет по кабелю зі швидкістю передачі даних до 100 Мбіт/с. Такі плати, як Малина Пай, Нвідіа, Beagle bone та Асус Тінкер, потребують аналого-цифрових перетворювачів для зчитування даних у реальному часі з датчиків. [20]

Рисунок 2.2.2 ілюструє одноплатні комп'ютери в режимі реального часу, що працюють від джерела живлення 5 В. Ці пристрої мають дуже малі розміри і здатні виконувати операції портативно(рис. 2.2.1).

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						71
Зм.	Арк.	№ докум.	Підпис	Дата		

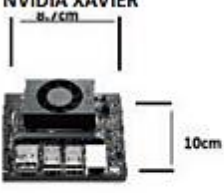
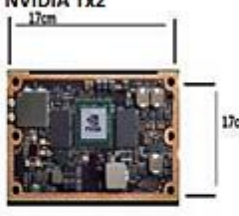


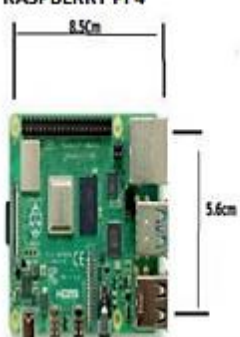
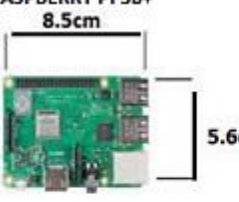

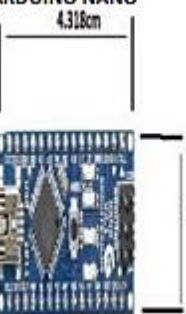
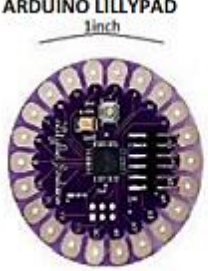
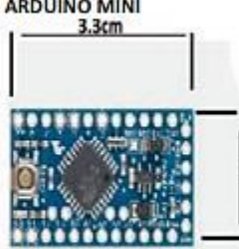

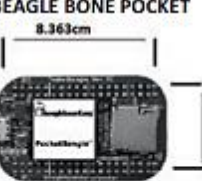
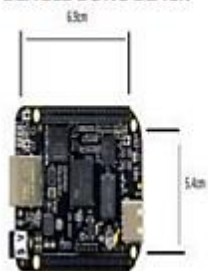
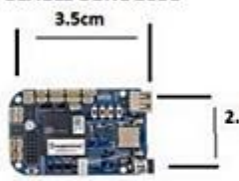
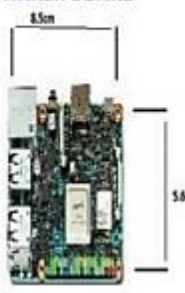

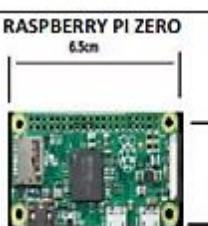



NVIDIA XAVIER  8.7cm 10cm	NVIDIA Tx2  17cm 17cm	NVIDIA JETSON NANO  10cm 5.5cm	RASSBERRY PI ZERO W  6.5cm 3cm
RASPBerry PI 4  8.5cm 5.6cm	RASPBerry PI 3b+  8.5cm 5.6cm	ARDUINO UNO  6.8cm 11cm	ARDUINO NANO  4.318cm 1.7cm
ARDUINO LILYPAD  1inch	ARDUINO MINI  3.3cm 1.8cm	NXP  1.7cm 6cm	BEAGLE BONE POCKET  8.363cm 5.461cm
BEAGLE BONE BLACK  6.5cm 5.4cm	BEAGLE BONE BLUE  3.5cm 2.15cm	TINKER BOARD  8.5cm 5.8cm	RASPBerry PI 3  6.5cm 3cm
RASPBerry PI ZERO  6.5cm 3cm	RASPBerry PI 3  6.5cm 3cm	ARDUINO MEGA:  10.16cm 5.33cm	ARDUINO LEONARDO:  6.858cm 5.33cm

Рисунок 2.2.1 - Опис розмірів одноплатних комп'ютерів, які широко використовується

Зм.	Арк.	№ докум.	Підпис	Дата

123.УДК: 004.42[629.33/.34:53.083.7]

У таблиці 2.1 наведено операційні дані плат, включаючи операційну систему, живлення, пам'ять, швидкість роботи системи та підтримки мов програмування.

Таблиця 2.1

Опис інтерфейсів введення\виведення одноплатних комп'ютерів та інші необхідні для вибору характеристики

Плата	GPIO	ОС	Живлення	Пам'ять	Продуктивність системи	Підтримка мов програмування
Raspberry Pi 3	40 конт.	Ubuntu Mate, Snappy Ubuntu Core, Kodi-базовані медіа центри OSMC на LibreElec, не-Linux-базовані Risc ОС, також Windows 10 IoT Core	5 Вольт, мін. 2 А	1 Гб ПДД	RISC процесор, частота від 700 МГц до 1.4 ГГц	C, C++ та Python
Rasp. Pi 3b+	40 конт.	Raspbian, Ubuntu Retropie Manjaro, OSMC Lakka Kali Linux, Kano OS, GentooRecalBox, DietPi, LibreELEC Fedora, OpenMediaVault, CentOS	2.5 А	512 Мб ПДД	Частота процесора 1.4 ГГц	C, C++ та Python
Rasp. Pi 4	40 конт.	Raspbian, Ubuntu MATE, Ubuntu Core, Ubuntu Server, Windows 10 IoT Core, Open Source Media Center.	3 А	2, 4, 8 Гб ПДД	Частота процесора 1.5 ГГц	C, C++ та Python
Rasp. Pi Zero	40 конт.	Raspbian Jessie, Kodi (OpenElec або OSMC) RetroPie, Kali Linux, MusicBox, MotionEyeOS,	1.2 А	512 Мб ПДД	Частота процесора 1 ГГц	C, C++ та Python

123.УДК: 004.42[629.33/.34:53.083.7]

Арк.

73

Зм. Арк. № докум. Підпис Дата

		RuneAudio, ArchLinux.				
Rasp. Pi Zero W	опц. 40 конт.	Raspbian Jessie, Kodi (OpenElec або OSMC)RetroPie, Kali Linux, MusicBox, MotionEyeOS, RuneAudio, ArchLinux.	1.2 А	512 Мб ПДД	Одноядерний АРМ11 процесор працюючий на частоті 1 ГГц	С, С++ та Python
Плата Асуе Тінкер	40 конт.	Tinker OC	5 В, 3 А	2 Гб DDR3 ПДД	Турбо тактова частота до 2,6 ГГц	С, С++ та Python
Плата Нвідіа JETSON NANO	40 конт.	Лінукс	5 В, 2 А	4 Гб ПДД	1600 МГц	С, С++ та Python
Rasp. Pico	26 GPIO	Micropython	5В	264 кб, 16 Мб флеш- пам'яті поза чіпом	Процесор Arm Cortex- M0+, 133 МГц	Python

Доступні різні одноплатні комп'ютери, однак найчастіше використовуваний економічно вигідний одноплатний комп'ютери - це РПай для прототипування. На рисунку 2.2.2 показано набір контактів РПай GPIO, починаючи з 3,3 В.

Входи/виходи загального призначення в РПай класифікуються як:

- Цифровий вхід/вихід
- Контакти зв'язку
- ШІМ-виходи

Контакти цифрового входу здатні зчитувати стани 0/1, однак порт Малини Пай не оснащений своїм АЦП, і для зчитування аналогової напруги потрібна окрема інтегральна схема.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

У останніх версіях Малини Пай близько 40 контактів. Ці контакти здатні забезпечувати вихід 3.3 В. Доступні два контакти 5 В, ці контакти управляються мовами програмування С, С ++, Python та node-red. Команда для зчитування всієї конфігурації контакту полягає у введенні такої команди у вікні терміналу Пай.

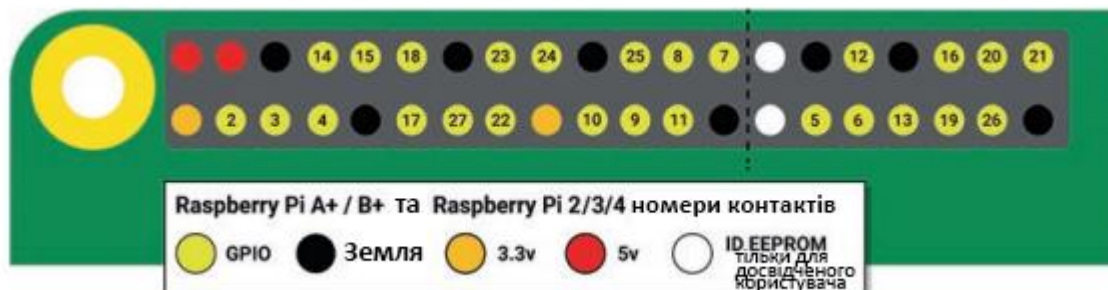


Рисунок 2.2.2 - Нумерація контактів GPIO у РПай

Рисунок 2.2.3 ілюструє вивід команди `gpio readall`. Середній струм, який можна отримати від контакту gpio, становить близько 16 мА.

```

pi@raspberrypi: ~
File Edit Tabs Help
Running at boot
^C
pi@raspberrypi:~ $ clear
pi@raspberrypi:~ $ gpioreadall
bash: gpioreadall: command not found
pi@raspberrypi:~ $ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |      | 3.3v     |      |   | 1 | 2 |      | 5v       |      |      | |
| 2    | 8    | SDA.1    | ALT0 | 1 | 3 | 4 |      | 5v       |      |      |
| 3    | 9    | SCL.1    | ALT0 | 1 | 5 | 6 |      | 0v       |      |      |
| 4    | 7    | GPIO. 7  | IN   | 1 | 7 | 8 | 1    | ALT5    | TxD  | 15  | 14  |
|      |      | 0v       |      |   | 9 | 10 | 1    | ALT5    | RxD  | 16  | 15  |
| 17   | 0    | GPIO. 0  | IN   | 0 | 11 | 12 | 0    | IN      | GPIO. 1 | 1  | 18  |
| 27   | 2    | GPIO. 2  | IN   | 0 | 13 | 14 |      |          | 0v     |      |      |
| 22   | 3    | GPIO. 3  | IN   | 0 | 15 | 16 | 0    | IN      | GPIO. 4 | 4  | 23  |
|      |      | 3.3v     |      |   | 17 | 18 | 0    | IN      | GPIO. 5 | 5  | 24  |
| 10   | 12   | MOSI     | IN   | 0 | 19 | 20 |      |          | 0v     |      |      |
| 9    | 13   | MISO     | IN   | 0 | 21 | 22 | 0    | IN      | GPIO. 6 | 6  | 25  |
| 11   | 14   | SCLK     | IN   | 0 | 23 | 24 | 1    | IN      | CE0    | 10  | 8   |
|      |      | 0v       |      |   | 25 | 26 | 1    | IN      | CE1    | 11  | 7   |
| 0    | 30   | SDA.0    | IN   | 1 | 27 | 28 | 1    | IN      | SCL.0  | 31  | 1   |
| 5    | 21   | GPIO. 21 | IN   | 1 | 29 | 30 |      |          | 0v     |      |      |
| 6    | 22   | GPIO. 22 | IN   | 1 | 31 | 32 | 0    | IN      | GPIO. 26 | 26 | 12  |
| 13   | 23   | GPIO. 23 | IN   | 0 | 33 | 34 |      |          | 0v     |      |      |

```

Рисунок 2.2.3 - Gpio Readall інформація у вікні терміналу

Під час використання цих контактів gpi0 слід дотримуватись певних правил і застережень, деякі з цих контактів gpi0 безпосередньо пов'язані з процесором. Будь-яке неправильне з'єднання з цими контактами може повністю вивести ОС з ладу і апаратне забезпечення. У деяких одноплатних комп'ютерах схема захисту від напруги відсутня, заборонено підключати джерело більше 3,3 В до будь-яких контактів GPIO. Керуючі ємнісні навантаження не рекомендується використовувати з цими контактами, систему слід підключати лише до навантажень резистора-конденсатора.

Програма Hardwire - це в основному програмування на основі логічних вентилів швидше, ніж мікропрограмування. Це дуже складно, оскільки всі реалізації повинні виконуватися за допомогою логічних елементів управління та сигналів управління. Така програма складно модифікується і можливо обмежена в певній мірі. Для швидшого виконання доцільно використовувати програмування на мові C.

2.2.1. Інтегровані середовища розробки для одноплатних комп'ютерів

32-розрядні одноплатні комп'ютери можуть працювати з операційними системами, які базуються на Linux. У деяких випадках плати запрограмовані для серверних програм. Легка операційна система також доступна для плат, таких як РПай, щоб скоротити час завантаження на шарі програм. Таблиця 1.3 ілюструє тип плати та її інтегроване середовище розробки(IDE).

Таблиця 2.2

Широко застосовувані IDE для одноплатних комп'ютерів

Тип плати	Інтегроване середовище розробки
Arduino	Arduino IDE, MATLAB, Keil C
Raspberry Pi	Python, Node-red, C, C++ тощо.
NXP	mbed, keil c programming, C, C++...
Тінкер	Python, node-red, C, C++ тощо.
NVIDIA	Python, Node-red, CUDA, тощо.

В даний час мікроконтролери, які програмуються кеїлом С також називаються вбудованими С. Малюнок 2.2.1.1 пояснює кроки програмування, які використовуються за допомогою Keil С. Компілятор перетворює програму С на шістнадцяткові коди та програмує частину програми на цільовий мікроконтролер.

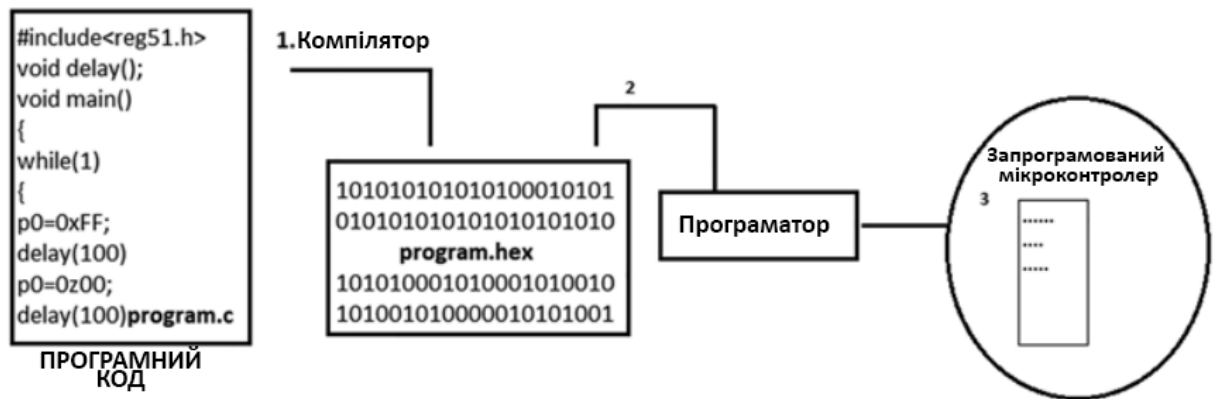


Рисунок 2.2.1.1 - Програмування вбудованої системи

2.2.2. Одноплатний комп'ютер Raspberry Pi

Можливості одноплатного комп'ютера майже нескінченні: мультимедіа, сервери та бази даних, всілякі засоби управління та регулювання, навіть обробка текстів та ігри можливі з їхньою допомогою.

Вбудовані системи стали невід'ємною частиною нашого життя. Це комп'ютери або вбудовані комп'ютери в технічній пристрій, які непомітно виконують свою роботу. Багато хто використовує їх, навіть не підозрюючи про це, наприклад у вигляді мобільного телефону, як DSL-модем або для домашньої автоматизації. Оскільки ці системи можуть працювати вдень і вночі цілодобово, вони характеризуються насамперед низьким споживанням електроенергії. Найекономічніші системи мають під час роботи споживання електроенергії всього 2,5 Вт.

Серцем систем є процесор, який зазвичай розробляється як архітектура ARM (Advanced RISC1 Machines). Поряд з цим працює графічний процесор- GPU

(Graphics Processing Unit). Він відповідає за графічний вихід, а також, серед іншого, за прискорене і, таким чином, рідке відтворення відео. Тут є великі відмінності не тільки з точки зору продуктивності, але і з точки зору програмної підтримки, що надається виробником. Хоча виробник Broadcom надає приклади програмування свого пристрою BCM2837 (РПай), інші драйвери мають бути вилучені у вигляді двійкових файлів із систем Андроїд, щоб вони могли з більшим чи меншим успіхом використовуватися в інших операційних системах (Cubieboard). Це пояснюється тим, що багато вбудованих плат призначені як плати розробників для виробників мобільних пристроїв і не розроблені як пристрої для кінцевих користувачів.

РПай найпопулярніша вбудована плата, яка була продана понад 17 мільйонів разів. РПай (рис. 2.2.2.1) - це одноплатний комп'ютер у форматі кредитної картки, який у версії 3В+ має всі необхідні інтерфейси: 1 Гб ПДД, 4 порти USB 2.0, виходи HDMI та FBAS для підключення монітора або телевізора, роз'єм 3,5 мм для аналогового виведення звуку (цифровий вихід звуку через HDMI) або гніздо Ethernet зі швидкостями 10/100/300 Мбіт/с, який недоступний у варіанті А. Живлення подається через гніздо мікро-USB (мінімум 5,1 В, 2,5 А).



Рисунок 2.2.2.1 - РПай (модель 3В+) - це повноцінний комп'ютер у форматі кредитної картки

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		78

У той час як USB-з'єднання старих моделей РПай забезпечували максимум 100 мА, модель 3В+ забезпечує до 1,2 А для всіх портів USB. У будь-якому випадку цього достатньо для роботи з клавіатурою або мишею, але не достатньо для роботи з жорстким диском. Якщо ви хочете підключити додаткові пристрої до РПай, ми рекомендуємо використовувати для USB -пристрою активний концентратор USB або зовнішній блок живлення. Водночас рекомендується використовувати блок живлення принаймні 2,5 А.

РПай забезпечує різні інші з'єднання, які можна використовувати, наприклад, для підключення камери або дисплея.

Графічний процесор Broadcom "VideoCore IV" використовується як графічний процесор, який підтримує фільми в роздільній здатності Full HD (1080p30 H.264). Ліцензії на апаратне прискорене відтворення фільмів, кодованих MPEG-2 та VC1, можна замовити у Фонді РПай (приблизно 3 євро). Присутній слот для карт пам'яті мікро-SD, який може використовувати формати SDHC і SDXC, карта пам'яті служить сховищем даних.

Звичайна версія Пай може не підходити по різним критерія тому існує також менша версія РПай Зіро. Ви можете отримати маленький одноядерний РПай, включаючи вільне гніздо мікро-USB, 512 Мб робочої пам'яті та міні- Порт HDMI. Другий роз'єм мікро-USB потрібен для живлення. 40-ка контактний роз'єм, який потрібно припаяти самостійно (за винятком версії Н), - повністю сумісний зі своїм старшим братом, а також є підключення до камери. За додаткову плату існує Пай Zero у версії W з бездротовою локальною мережею та Bluetooth - це потрібно, якщо невелика плата вимагає підключення до мережі. Існує також безліч аксесуарів для Zero W. Витрата енергії на крихітну річ становить всього 0,5-0,7 Вт. Однак це все не зовсім приносить продуктивність її старшого брата, тому що встановлена старіша версія системи на кристалі.

Всі ці фактори роблять Пай Zero (WH) особливо придатною для проектів, які мають небагато місця для встановлення і вимагають хороше з'єднання WLAN. Адаптери потрібні для багатьох потреб (мікро-USB до USB, міні-HDMI до HDMI, концентратор USB).

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						79
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.2.2.2 - РПай модель Zero W

Таблиця 2.1 узагальнює всі сучасні моделі Raspberry Pi та дає огляд їх властивостей. Окрім плат для домашнього використання, існує також так званий “обчислювальний модуль” (рис. 2.2.2.3), який обладнаний стандартним роз’ємом DDR2 SODIMM. Цей роз’єм також використовується для підключення модулів пам’яті ПК до материнської плати. Не думайте про використання обчислювального модуля РПай на материнській платі ПК, це не спрацює, і є велика ймовірність, що не тільки зруйнується модуль у процесі, але і материнська плата. Існують також відповідні плати для тестування з відповідним гніздом, яке вміщує обчислювальний модуль. Обчислювальний модуль РПай в основному призначений для компаній, які розробляють власну друковану плату



Рисунок 2.2.2.3 - Обчислювальний модуль Raspberry Pi

Як вибрати оптимальну РПай для необхідного використання? Для всіх додатків, які вимагають максимальної продуктивності, РПай 4 є основним

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						80
Зм.	Арк.	№ докум.	Підпис	Дата		

вибором: він пропонує найшвидше мережеве з'єднання і має найвищі тактові частоти процесора.

Встановлений тут чіп WLAN не тільки підтримує діапазон 2,4 ГГц, але також передає на частоті 5 ГГц і набагато швидше, ніж у старих моделей. Він також підтримує PoE (живлення через Ethernet). Процесор працює на тактовій частоті 1,4 ГГц, що теоретично на 17% швидше, ніж у старшого брата. Однак, якщо всі 4 ядра центрального процесора використовуються на повну потужність, Пай 3В+ зменшує свою тактову частоту з причини нагрівання, так що мало що залишається від переваги більшої частоти.

РПай 3В має більше половини споживаної потужності від 3В+, і тому його рекомендують, коли потрібні висока продуктивність процесора та низьке споживання енергії. Не рекомендується РПай 2, тому що він не має переваг перед РПай 3. РПай Zero (W) рекомендується там, де місце для встановлення обмежене і висока продуктивність не потрібна.

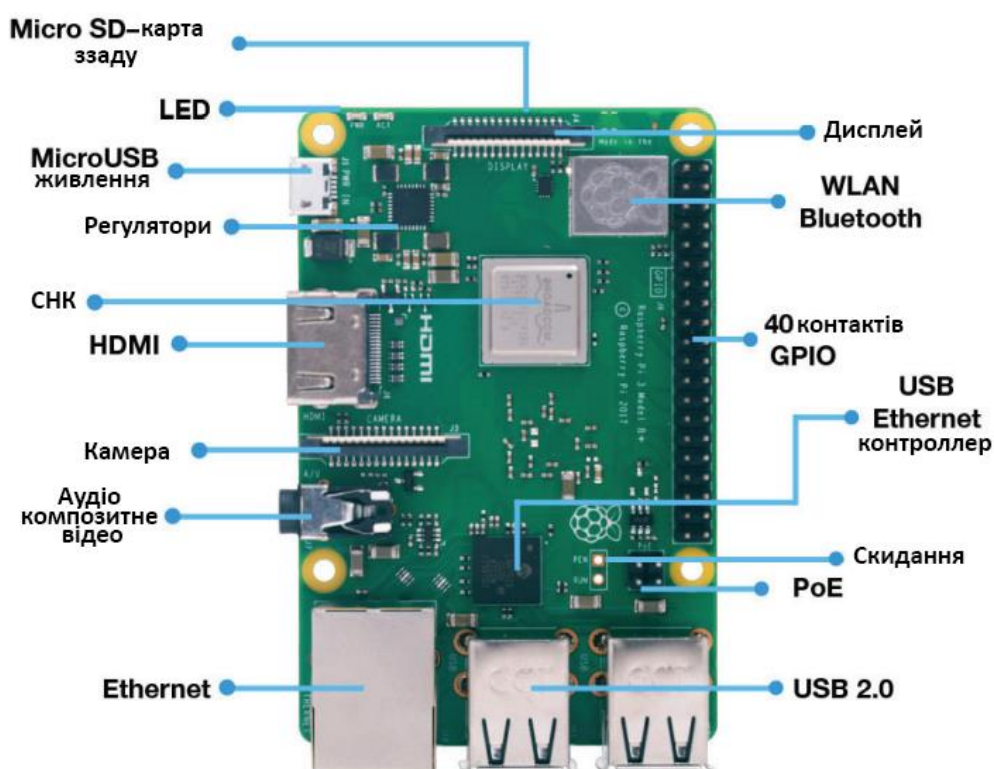


Рисунок 2.2.2.4 - З'єднання Raspberry Pi 3В+

На рисунку 2.2.2.4 показаний РПай 3В+ з усіма з'єднаннями та важливими компонентами плати. Гніздо для карти мікро-SD знаходиться на задній стороні друкованої плати. Карта мікро-SD вставляється в слот контактами, поверненими до друкованої плати. На відміну від Рі 3В, РПай 3В+ не має пружинного контакту. Всього доступно 4 порти USB2.0 та гніздо Ethernet. Контролер Ethernet підключається через USB 2.0. Система-на-кристалі BCM2837 знаходиться під кришкою з листового металу.

Камеру та зовнішні дисплеї можна з'єднати за допомогою стрічкового кабелю. Для цього необхідно відкрити білу клему, підключити стрічковий кабель і знову закрити білу клему. Менші дисплеї можна підключити через GPIO (вихід загального призначення). Пай має загалом 40 контактів, які забезпечують, наприклад, SPI, I2C, PWM та інші шини / сигнали.

З'єднання HDMI типу А дозволяє підключення до телевізора або монітора. У гніздо мікро-USB вставляється кабель під'єднаний до блоку живлення (мінімум 5,1 В, 2,5 А). Крім того, можна використовувати модуль PoE.

Модуль WLAN / Bluetooth також розташований під кришкою з листового металу. Існує також чотири просвердлені отвори, за допомогою яких Пай можна прикріпити, наприклад, до кріплення VESA. Для скидання Пай доступні 2 контакти.

2.2.3. Плата Асус Тінкер

Рух виробника в певному напрямі дав можливість звичайній людині скористатися невід'ємною людською стихією: виготовленням речей. Можливість створення роботів або метеорологічних станцій чи інших масштабних інтелектуальних пристроїв колись була зарезервована для найбільш освічених серед нас з доступом до дорогого обладнання та інструментів для виготовлення.

Плата Тінкер є частиною нової хвилі більш потужних інструментів для створення. Як одноплатний комп'ютер, він пропонує безліч варіантів для користувача. Він може керувати апаратним забезпеченням за допомогою коду,

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		82

він може запускати спеціалізовані операційні системи для нішевих проектів, або це може бути комп'ютер невеликого формату для особистого користування.

Відмінною рисою одноплатних комп'ютерів є те, що вони зазвичай мають систему на кристалі (СНК). СНК - це інтегральна схема (ІС), яка містить більшість основних компонентів комп'ютера в одному елементі, який займає одне місце на друкованій платі. Процесор часто є архітектурою на основі розширеної машини RISC (ARM), яка також зустрічається в сучасних телефонах на ОС Андроїд та Хромбук. Потужність і технології, що знаходяться в сучасних споживчих пристроях у зручному форм-факторі, доступному для виробників, мають багато можливостей. Творці можуть розширити межі свого проекту, не порушуючи бюджету.

Плата Тінкер, проілюстрована на рисунку 2.2.3.1, багато в чому схожа на інші одноплатні комп'ютери і має свою власну СНК; зокрема, Рокчип RK3288. Процесор RK3288 - це ARM Кортекс-А17, який має чотири ядра з тактовою частотою процесора 1,8 ГГц. Звичайно, більш досвідчені користувачі можуть спробувати розігнати чіп на свій страх і ризик.



Рисунок 2.2.3.1 – Плата Асус Тінкер

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

Його графічний процесор - ARM Малі-Т760, який також має чотири ядра і працює на частоті 600 МГц. Що стосується оперативної пам'яті, то вона має об'єм 2 ГБ ПДД LPDDR3, що дає хороший запас, щоб уникнути вузьких місць із характеристиками процесора та графічного процесора.

Для входів і виходів (I/O), як показано на рисунку 2.2.3.2, він має чотири порти USB 2.0 для підключення периферійних пристроїв та пристроїв HID. Існує також гніздо Ethernet для дротового підключення до Інтернету. Однак на платі також є можливість Wi-Fi, якщо вам потрібно, щоб плата була мобільною. Плата споживає живлення через з'єднання мікро-USB, але вимагає відносно великого джерела живлення щонайменше 5В та 2А-2,5А струму для оригінальної плати Тінкер та 5В і 3А для плати Тінкер моделі S, що є наслідком потужного RK3288.

HDMI 1.4 доступний на платі для відео із розширенням 1080p та 4K для деяких програм. Він також може передавати аудіо, але якщо ви віддаєте перевагу виділеному аудіовиходу, є 3,5-мм роз'єм для навушників, який може відтворювати до 24-бітної/192 кГц роздільної здатності.

Для зберігання, як і інші одноплатники, плата Тінкер пропонує слот для карт мікроSD для запуску ОС та власних файлів. Звичайно, якщо виявиться, що потрібно розширене сховище, є можливість скористатися згаданими раніше портами USB 2.0.



Рисунок 2.2.3.2 – Чотири порти USB 2.0 та порт Ethernet на платі Тінкер

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		84

Одна з найвідоміших і унікальних особливостей плати Тінкер - це набір входів/виходів загального призначення (GPIO). Контакти GPIO - це розривні контакти, які дозволяють зовнішнім електронним компонентам і схемам взаємодіяти з платою Тінкер.

Це дозволяє платі Тінкер вийти за рамки лише невеликої плати з тими ж функціями, що і ваш звичайний ПК. Маючи доступ до GPIO, ви можете створювати схеми з використанням різноманітних компонентів для створення унікальних проектів, над якими ви маєте повний контроль.

Присутні 40 окремих контактів, розташованих у двох горизонтальних рядах по 20 (рисунок 2.2.3.3). Кожен контакт ідентифікується номером, а плата Тінкер має колірне позначення заголовка GPIO, щоб ви могли з першого погляду легко визначити, яка саме спеціальна функція контакту.

Плюсом для плати є те, що вона поділяє свій набір входів/виходів загального призначення з серією одноплатних комп'ютерів РПай. Це означає, що існуючі схеми для РПай в більшості випадків можна без проблем перенести на плату Тінкер.

Існують також додаткові модулі для серії плат РПай, які називаються платами розширення HATs та Bonnets. Ці додаткові плати мають такий порт, який підходить безпосередньо до контактів GPIO, і мають вбудовані схеми для більш складних додатків, таких як управління серводвигунами, світлодіодні матриці, управління камерою тощо. Оскільки плата Тінкер має ідентично розведені контакти, ці надбудови у багатьох випадках можуть працювати з платою без проблем у багатьох випадках.

Є можливість використовувати різноманітні мови програмування для взаємодії з обладнанням, підключеним через контакти GPIO. Python і C - дуже популярні варіанти і мають багато бібліотек з відкритим початковим кодом.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		85

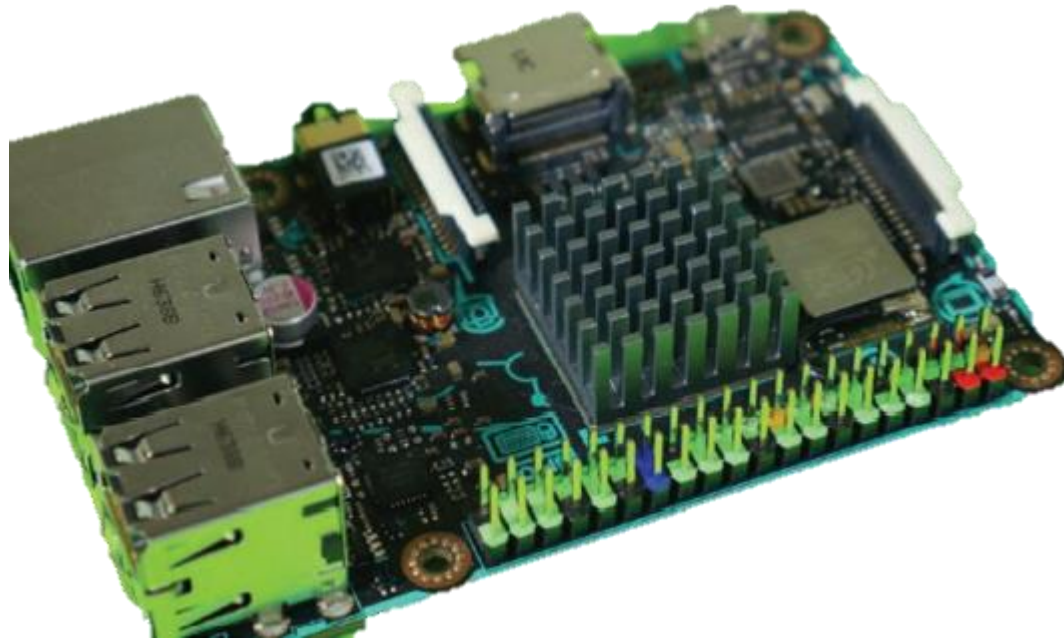


Рисунок 2.2.3.3 – Контакти GPIO на платі Тінкер

Останнє оновлення сімейства плат Тінкер - це плата Тінкер моделі S. На перший погляд плата Тінкер моделі S виглядає так само, як і оригінальна плата. Тим не менш, є деякі ключові відмінності, які можуть мати велике значення для продуктивності деяких проектів та варіантів використання.

Найбільш помітним доповненням до плати Тінкер моделі S є включення 16 ГБ вбудованої флеш-пам'яті eMMC (вбудована MultiMediaCard). Перевага eMMC в тому, що оскільки вона припаяна безпосередньо до плати, вона не використовує багато оперативної пам'яті, та може швидше передавати невеликі фрагменти даних між іншими вбудованими компонентами, ніж зовнішня пам'ять. Але коли ви працюєте над проектом, який доводить апаратне забезпечення до межі, вбудована eMMC може мати велике значення для запуску вашої операційної системи.

Дивлячись на GPIO, розробники внесли значні зміни у функціонал I2S (внутрішня IC звуку) на платі Тінкер моделі S. Як випливає з назви, I2S дозволяє передавати звук у форматі аудіоданих PCM між інтегральними схемами шляхом розділення часу та послідовних даних в окремі канали. Це ідеально підходить для аудіо додатків з високими потребами роботи із цифровими системами і де

використовується ЦАП (цифро-аналоговий перетворювач) для високоякісного прослуховування звуку. Традиційно тактова частота для I2S може працювати як у головному режимі, де вихід контакту встановлює годинник, так і в підпорядкованому режимі, де контакт приймає годинник від зовнішнього джерела. Оригінальні контакти I2S плати Тінкер дозволяли йому діяти тільки в головному режимі, але для плати моделі S, контакти можуть діяти як у головному режимі, так і в підпорядкованому режимі. Це дає більшу гнучкість для аудіопроєктів з платою моделі S і дозволить йому мати більшу сумісність зі спеціальними ЦАП та АЦП. Доступні додаткові плати ЦАП та АЦП, які підходять під контакти GPIO. Додатковим оновленням аудіо на платі моделі S є можливість автоматичного перемикання між 3,5-мм роз'ємом та HDMI для виведення звуку після підключення джерела. Це буде надзвичайно корисним для традиційної настройки робочого столу або проекту домашнього медіа-центру.

Останнім оновленням для плати моделі S є додавання двоконтактного роз'єму, який відповідає за живлення. Це може здатися тривіальним, але однією з функцій, яких бракує багатьом одноплатним комп'ютерам, є можливість дискретно вмикати плату. Оригінальну плату Тінкер, разом з багатьма іншими, можна увімкнути, лише підключивши блок живлення до плати. Виробники поділилися спільними модифікаціями, які використовують GPIO для підключення кнопки живлення, але вони жертвують цінними контактами GPIO для дуже простої функції. Для цього виділені контакти, відокремлені від GPIO на платі моделі S, є дуже корисними і простими для користувачів, яким потрібна можливість вставити свою плату в корпус або уникнути необхідності багаторазово від'єднувати та вставляти плату.

Крім цих змін, плата моделі S дуже схожа на оригінальну плату. Вона використовує ті ж СНК, графічний процесор, оперативну пам'ять, введення - виведення, GPIO (окрім оновленої функції I2S) та форм-фактор. Більшість користувачів можуть навіть не помітити відмінностей між цими двома платами, оскільки оновлення до моделі S мають спеціалізований характер. Однак ці оновлення також можуть бути відмінністю між зручним досвідом проекту та

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		87

розчаруванням. Вибираючи між двома варіантами, важливо врахувати сценарій використання та прийняти рішення.

Саме ці можливості та здібності виводять проекти на новий рівень із платою Тінкер та подібними одноплатниками. У порівнянні з більш базовою платформою, як-от платою Ардуіно Уно, можливості розширюються, коли мова йде про різні периферійні пристрої та технології, які можна бездоганно інтегрувати. Наприклад, можна забезпечити підключення до Інтернету проекту на базі Ардуіно, але під'єднання плати до мережі - це майже проект сам по собі. За допомогою плати Тінкер Wi-Fi вже є вбудованою функцією для плати, тому це так просто, як увійти у свою мережу або підключити кабель Ethernet. Єдине, що могло стримати від роботи із платою Тінкер це той факт, що RK3288 працює на архітектурі ARM версії 7-А, яка є 32-розрядною, а не більш сучасною 64-розрядною. Однак більшість операційних систем і процесів одноплатників наразі не використовують 64-розрядні версії, тому проблем із проектами виникати не повинно. Про цю архітектуру обладнання слід пам'ятати. RK3288 створений для спрощення більш важких процесів, оскільки має апаратне прискорення для складних відеофайлів та кодеків, що часто є перешкодою для продуктивності одноплатника. Він також використовувався у багатьох телефонах на ОС Андроїд та Хромбук, це свідчить про те, що він може використовуватись повсякденно, що є гарним знаком довіри щодо його використання у одноплатнику.

Навіщо використовувати плату Тінкер замість іншого одноплатника? Оскільки Тінкер має більш високі характеристики, ніж середньостатистичний одноплатник, він ідеально підходить для більш навантажених проектів, які в минулому перейшли б межі інших одноплатників. Це також чудовий вибір для виробників або ентузіастів технологій, які хочуть трохи глибше поглибитись у різні дистрибутиви ОС Лінукс.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						88
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРИСТРОЮ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Основне завдання полягає в створенні пристрою та додатку, що вміють зчитувати з мережі контролерів повідомлення та обробляти їх, і відображати оброблену інформацію в режимі реального часу. Проєкт може працювати, наприклад, на РПай з 7-ми дюймовим дисплеєм.

3.1. Реалізація апаратної частини

Технології апаратної частини, які можна використати для виконання поставлених завдань: Ардуїно(для емуляції автомобіля), шина автомобіля(мережа контролерів), СТМ32(для приймача повідомлень мережі контролерів), одноплатник Рпай(комп'ютер на який проходять повідомлення від СТМ32) та дисплей для відображення користувацького інтерфейсу.

Апаратна частина проєкту підключається до автомобіля через діагностичний роз'єм OBD2 (On-Board Diagnostics). [21] Це загальний термін для вбудованої (наданої виробником) системи діагностики транспортного засобу. Він надає можливості: самодіагностики, ідентифікації та зчитування діагностичної інформації, контрольованого усунення несправностей, моніторинг у режимі реального часу та аналіз характеристик та сигналів певних датчиків тощо.

Зазвичай цей роз'єм знаходиться в салоні автомобіля, але бувають і винятки, наприклад, у підкапотному просторі. Через цей роз'єм зчитуються фрейми мережі контролерів.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						89
Зм.	Арк.	№ докум.	Підпис	Дата		

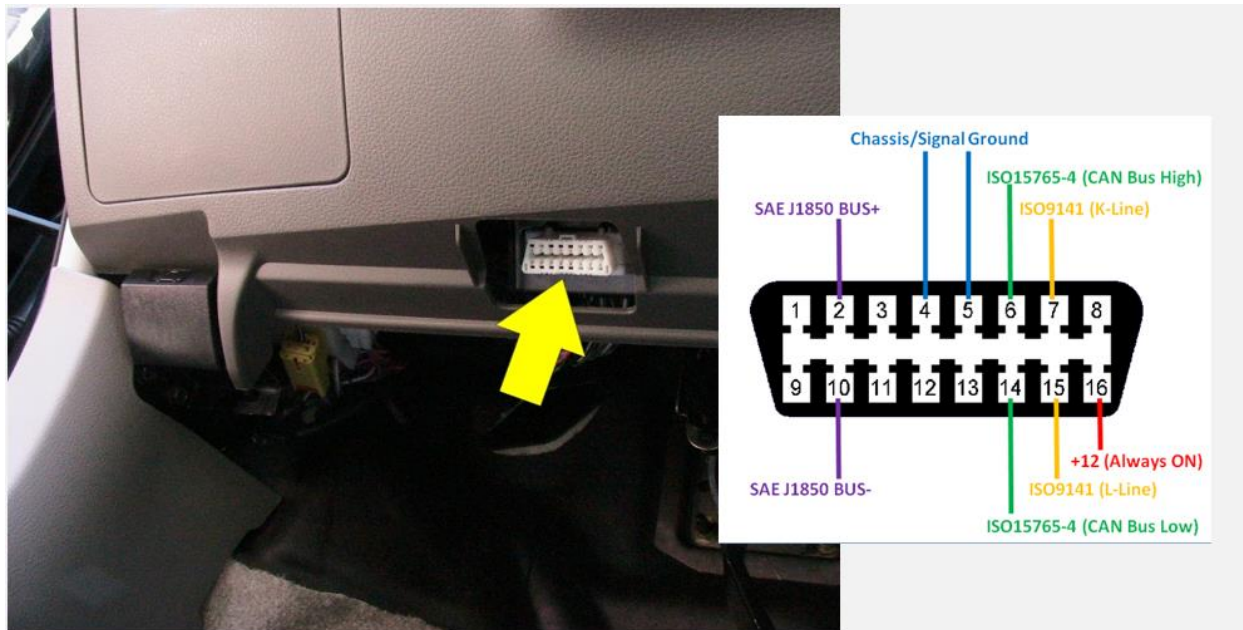


Рисунок 3.1.1 – Приклад роз'єму діагностування та його контакти

Продукти серії STM32F103x4 і STM32F103x6 включають високопродуктивне 32-розрядне ядро RISC, що працює на робочій станції Кортекс-M3 з частотою 72 000 кГц, високошвидкісну вбудовану пам'ять (до 256 кбіт флеш-пам'яті, до 48 кбіт СПДД), вдосконалену систему ввідів-виводів та периферійні пристрої, підключені до двох шин. Усі пристрої мають два 12-розрядних АЦП, три 16-розрядних таймери загального призначення та ШІМ таймер, а також стандартні та розширені інтерфейси комунікації: до двох I2C та SPI, три послідовних порти, USB та CAN.

Продукти серії STM32F103 з низьким навантаженням живляться джерелом живлення від 2,1 В до 3,6 В і можуть працювати в діапазоні температур від - 41°C до + 84°C і розширеному діапазоні температур від - 41°C до +104°C. [22] Повний набір режимів енергозберігання дозволяє розробляти мало навантажені програми.

Лінійка продуктів низької навантаженості STM32F103xx включає пристрої в чотирьох різних типах корпусів: від 36 до 64 контактів. Залежно від вибраного пристрою входять різні набори периферійних пристроїв. Ці функції роблять сімейство мікроконтролерів з низькою навантаженістю придатними для широкого спектру застосувань, таких як двигуни, керування застосунками,

медичне та портативное обладнання, ПК та ігрові периферійні пристрої, платформи GPS, промислові програми, інвертори, принтери, сканери, системи сигналізації, відеодомофони та опалення, вентиляція і кондиціонування повітря.

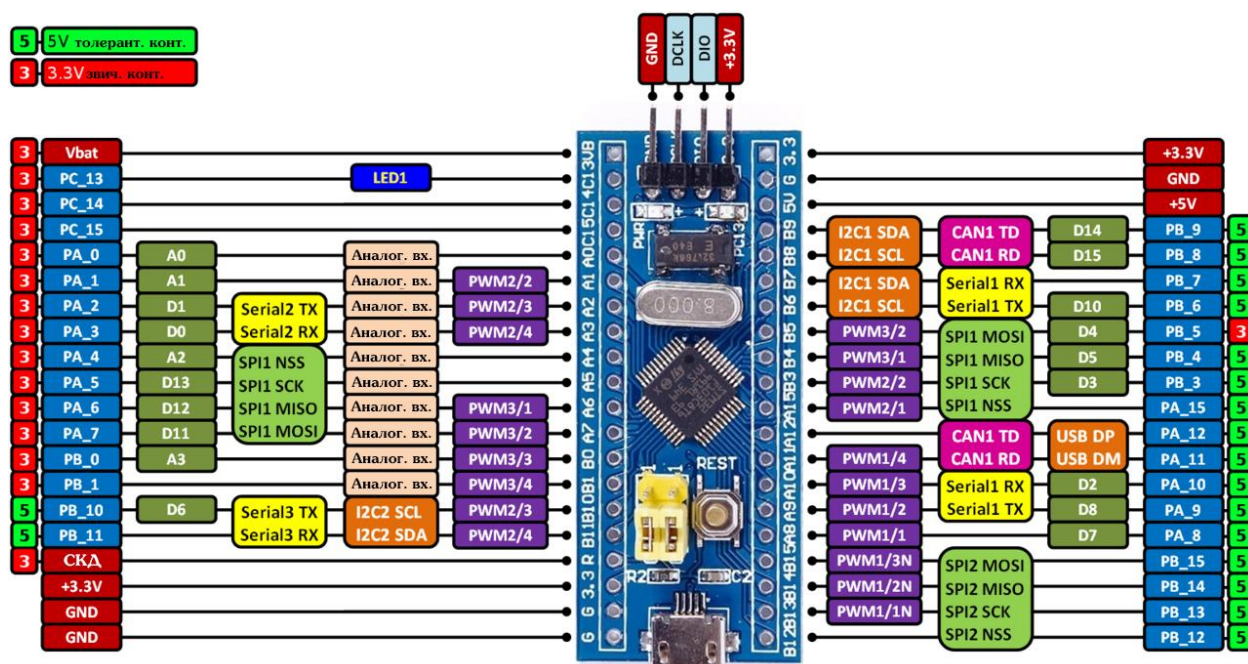


Рисунок 3.1.2 – Входи\виходи на платі з чіпом STM32F103

Процесор АРМ Кортекс-М3 - це доволі нове покоління процесорів АРМ для інтегрованих систем. Процесор був розроблений, щоб забезпечити платформу для невеликого бюджету, яка відповідає потребам впровадження блоку мікроконтролера, зі зменшеною кількістю контактів та низьким споживанням енергії, забезпечуючи при цьому достатню обчислювальну продуктивність та розширену реакцію системи на переривання. 32-розрядний процесор RISC має виключну продуктивність коду, забезпечуючи продуктивність високого рівня, очікувану від ядра АРМ, у розмірі пам'яті, зазвичай пов'язаному з 8- і 16-розрядними пристроями. Оскільки тут вбудоване ядро АРМ, тому воно ладнає з усіма застосунками та ПЗ АРМ.

Мережа контролерів відповідає стандартам 2.0А і В (активний) зі швидкістю передачі інформації до 0.125 МБ/с. Він може приймати і передавати стандартні фрейми з 11-розрядними ідентифікаторами, а також розширені кадри

з 29-розрядними ідентифікаторами. Він має три скриньки для передачі, дві приймальні черги(FIFO - ПЗПВ) з 3 етапами та 14 масштабованих банків фільтрів.

Лінійка продуктів STM32F103 має вбудований периферійний пристрій USB, сумісний із повношвидкісним USB 12 Мбіт. Інтерфейс USB реалізує повношвидкісний (12 Мбіт/с) функціональний інтерфейс. Він має настроювані програмним забезпеченням кінцеві точки та підтримку/призупинення підтримки. Виділені тактові частоти 48 МГц генеруються з внутрішнього основного ФАПЧ (джерело тактового сигналу має використовувати кристалічний генератор HSE).

ТЖА1050 є інтерфейсом між контролером протоколу мережі контролерів і фізичною шиною. [23] Він в першу чергу призначений для високошвидкісних автомобільних додатків із швидкістю передачі від 60 кбод до 1 Мбод. Він забезпечує диференціальну здатність передачі до шини та диференціальний приймач до контролера протоколу CAN. Він повністю сумісний зі стандартом ІСО 11898.

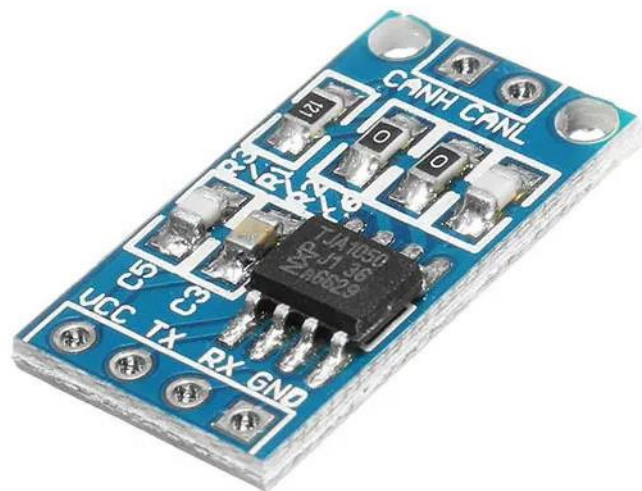


Рисунок 3.1.3 – Вигляд модуля мережі контролерів із ТЖА1050

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						92
Зм.	Арк.	№ докум.	Підпис	Дата		

Керуючий контакт S мікросхеми дозволяє вибрати два режими роботи: високошвидкісний режим або тихий режим. Високошвидкісний режим є нормальним робочим режимом і вибирається шляхом підключення контакту S до землі. Це режим за замовчуванням, якщо контакт S не підключений. Однак, щоб забезпечити продуктивність ЕМІ в додатках, що використовують тільки високошвидкісний режим, рекомендується, щоб контакт S був підключений до землі. У тихому режимі передавач вимкнено. Усі інші функції ІС продовжують працювати. Тихий режим вибирається шляхом підключення контакту S до живлення і може використовуватися для запобігання блокування мережевого зв'язку через контролер мережі, який вийшов з ладу.

Схема таймера «домінантного тайм-ауту TXD» запобігає переведенню шини у постійний домінуючий стан (блокування всього мережевого зв'язку), якщо контакт TXD постійно перебуває в низькому стані через збій апаратного та/або програмного забезпечення. Таймер запускається негативним фронтом на контакті TXD. Якщо тривалість низького рівня на висновку TXD перевищує значення внутрішнього таймера, передавач вимикається, що приводить шину в рецесивний стан. Таймер скидається позитивним станом на контакті TXD.

Автономний контролер мережі з інтерфейсом SPI. МСiPi2515 від Майкросхем Текнолоджі — це автономний контролер локальної мережі (мережа контролерів), який реалізує специфікацію мережі контролерів, версії 2.0В. Він здатний передавати і приймати як стандартні, так і розширені дані та віддалені кадри. У МСiPi2515 є дві маски приймання та шість фільтрів приймання, які використовуються для фільтрації небажаних повідомлень, тим самим зменшуючи накладні витрати вузла мікроконтролера. Інтерфейс МСiPi2515 з мікроконтролерами здійснюється через стандартний послідовний периферійний інтерфейс (SPI).

МСiPi2515 — це автономний контролер мережі, розроблений для спрощення програм, які потребують взаємодії з шиною мережі контролерів. [24] Пристрій складається з трьох основних блоків: 1. Модуль мережі контролерів, який містить механізм протоколу мережі контролерів, маски, фільтри, буфери

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		93

передачі та прийому. Логіка управління та регістри, які використовуються для налаштування пристрою та його роботи. Блок стандартного протоколу послідовного периферійного інтерфейсу. Приклад реалізації системи з використанням пристрою показаний на рисунку 3.1.4.

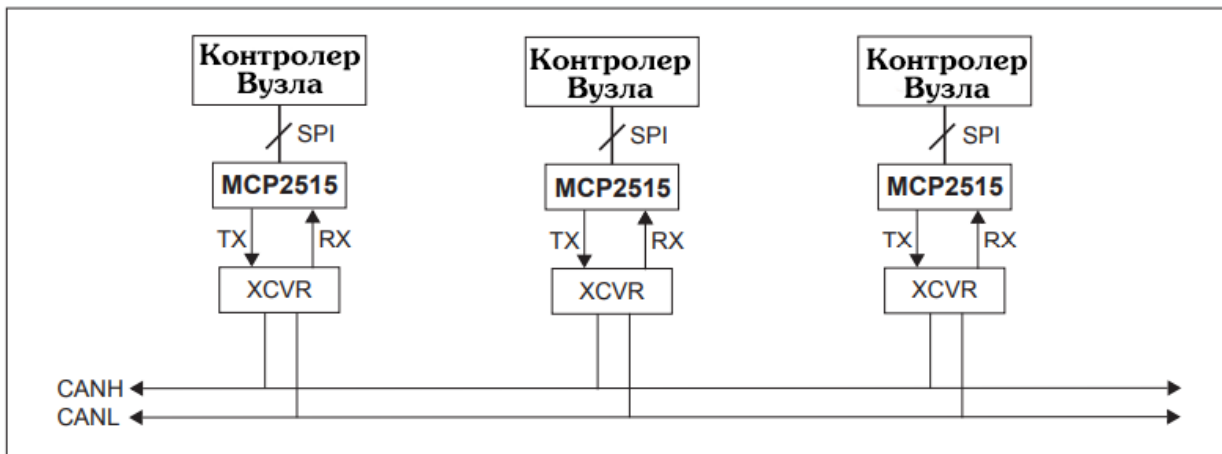


Рисунок 3.1.4 - Приклад реалізації системи

Максимальна напруга живлення VDD МСіПі2515 є 7.0В. Температура зберігання від -65°C до + 150°C. Зовнішня температура при під'єднаному живленні від -65°C до + 125°C. Більші значення будуть критичними для МСіПі.

Приймач мережі контролерів складається з плати STM32Ф103 та трансивера мережі контролерів ТЖА1050. Приймач мережі контролерів слухає всі вхідні CAN повідомлення і повторно відправляє їх послідовним портом.

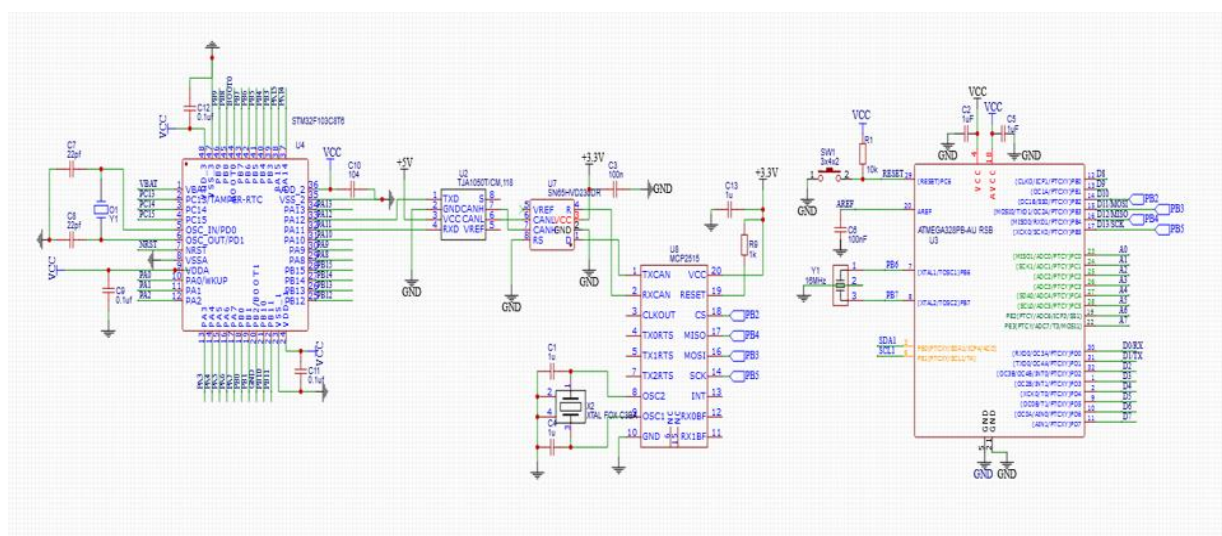


Рисунок 3.1.5 – Схема підключення приймача і емулятора шини мережі контролерів

Емулятор, який створює емуляцію автомобільної шини, складається з плати Ардуїно та контролера МСіПі2515. Він робить емуляцію CAN-інтерфейсу електронного блоку управління, надсилаючи з деякими інтервалами пакети мережі контролерів, що представляють емульовані параметри.

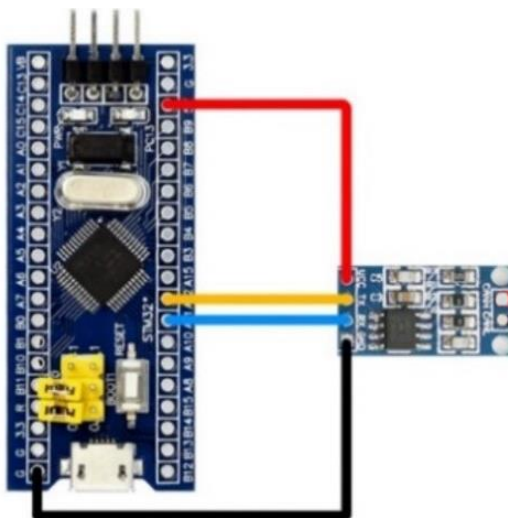


Рисунок 3.1.6 – Підключення приймача повідомлень мережі контролерів

Емулятор містить необроблене представлення емульованих параметрів та оновлює відповідні пакети мережі контролерів. Крім того, емулятор може виводити повідомлення мережі контролерів послідовним портом, замість того, щоб відправляти їх через МСіПі2515, для цілей тестування.

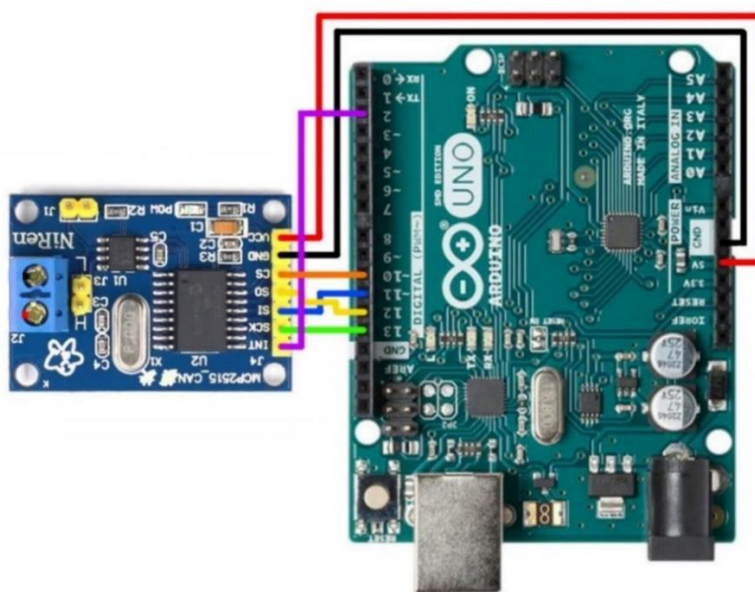


Рисунок 3.1.7 – Підключення емулятора мережі контролерів

3.2. Реалізація програмної частини

Технології які можна використати для виконання завдання: дот нет Блейзор (Програмна частина, фреймворк), Бутстрап (Інтерфейс користувача, бібліотека).

Програмна частина проєкту працює на Блейзор - безкоштовному веб-фреймворку з відкритим початковим кодом. Використано серверну версію фреймворку Блейзор.

Блейзор — це каркас веб-інтерфейсу, що дозволяє використовувати C# і .NET Core у фронті. Він дозволяє вам розвивати свою логіку інтерфейсу кількома різними способами, використовуючи мову програмування C#.

Відкидаючи технічні аспекти, у будь-якому стандартному проєкті веб-розробки знадобиться двоє спеціалістів, одна для JavaScript, а інша для бек-енду. Іноді також потрібен дизайнер для обробки елементів HTML і CSS та виконання інших завдань дизайну. Технологія Блейзор не усуне жодної залежності від дизайнера, але вона, безсумнівно, усуне залежність від JavaScript. Однак JavaScript все ще можна використовувати з технологією Блейзор, при потребі.

Блейзор — це комплексна технологія, що містить багато елементів, але Microsoft подбала про складність, надаючи три різні шаблони — на стороні сервера, на стороні клієнта та на хості. У цьому проєкті буде використано серверну частину.

Блейзор використовує синтаксис Рейзор (C# змішаний з HTML) тому будь-яке знайомство з синтаксисом Рейзор дасть перевагу під час розробки. Найголовніше, що Рейзор трапляється лише в одному, а Блейзор повторюватиметься знову й знову, а це означає, що ваша частина C# у Рейзор (файл .cshtml) виконуватиметься лише під час завантаження сторінки, але у Блейзор (файл .razor) код виконуватиметься на завантаженій сторінці для різних подій, таких як onclick, onchange та інші. Він використовує WebSocket для зв'язку з сервером, а також для роботи на стороні сервера, або використовує технологію WebAssembly, яка дозволяє створювати C# на клієнтській стороні. Тут вступають в гру різні типи технології Блейзор.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		96

На стороні сервера (див. рис. 3.2.1) Блейзор буде виконувати всю логіку, в основному використовуючи WebSocket для виконання завдання. Це дає можливість використовувати C# для написання інтерфейсного коду, це може бути не найефективнішим варіантом. За допомогою цієї опції ви усуваєте потребу у викликах API, оскільки ви просто вставите свої бібліотеки безпосередньо в передню частину.

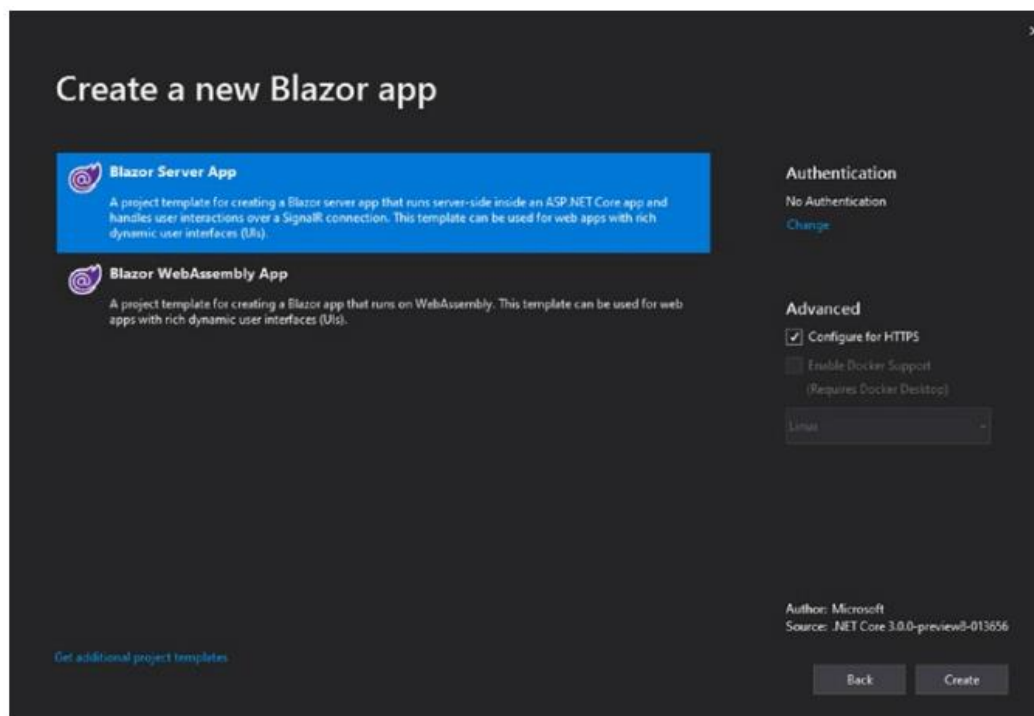


Рисунок 3.2.1 - Шаблони Blazor

Усі три види проєктів на Блейзор мають різні шаблони у Visual Studio, і завжди потрібно використовувати їх для своїх проєктів Блейзор, незалежно від того, який тип обрано. Як показано на рис. 3.2.1, потрібно буде вибрати тип проєкту Блейзор Додаток, а потім вибрати тип Блейзор після того, як ви вибрали місце розташування вашого проєкту.

Хоча серверна сторона проєкту на Блейзор виявляється зручною, все одно потрібно використовувати клієнт Блейзор, тобто Блейзор, що працює у браузері. На стороні сервера використовуватимуться ресурси сервера, тоді як клієнт заощадить ресурси або, принаймні, не витратить їх даремно.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						97
Зм.	Арк.	№ докум.	Підпис	Дата		

Коли виберете тип Блейзор у Visual Studio, потрібно буде вибрати «Додаток Блейзор Сервер».

Не існує найкращого типу Блейзор, кожна опція має власний варіант використання. Все залежить від того, що потрібно проєкту зараз і, що ще важливіше, що йому знадобиться в майбутньому. Якщо невідомо, просто перейдіть на сторону клієнта, оскільки це буде найрізноманітніший варіант.

Різниця між Рейзор та Блейзор у тому, що Рейзор буде виконуватись лише один раз під час «запуску сторінки», тоді як Блейзор буде працювати весь час. Цикли та логічні оператори будуть повторно переоцінено в Блейзор, тоді як у Рейзор це відбудеться лише один раз.

Як згадувалося раніше, при знанні синтаксису Рейзор, також буде відомо синтаксис Блейзор. Синтаксис Блейзор міститься у файлі розмітки з назвою .razor, який містить HTML, а також код C#.

Блейзор дозволяє прив'язати вхідне значення HTML до змінної і навпаки. Тому, здебільшого, можна назвати всі прив'язки двосторонніми. Якщо прив'язується текстове поле (текст типу введення) до рядкової змінної, відображене значення буде значенням цього рядка. Різні елементи працюватимуть по-різному, і для цього існує багато варіантів використання.

Використано такі додаткові пакети NuGet:

- Пакет від Майкрософт Конфігурація розширень - налаштування в АСП.НЕТ Кор здійснюється за допомогою одного або кількох надавачів конфігурації. Постачальники конфігурації одержують дані конфігурації в парах "ключ-значення" з різних джерел: файлів параметрів, таких як appsettings.json; Змінні середовища, сховище ключів Azure; конфігурація програми Azure; аргументів командного рядка; постачальники користувача, встановлені або створені; довідкових файлів; об'єктів . NET у пам'яті. Цей пакет використано для того, щоб конфігурувати налаштування СОМ-порту у файлі appsettings.json.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		98

- Системний пакет Порти введення\виведення(ІО) - містить класи для керування послідовним портом. Найважливіший з них, клас Послідовний порт, надає інструменти для синхронного та керованого подіями введення-виведення, доступу до відключеного стану пристрою та доступу до властивостей драйвера послідовного порту. Його можна використовувати для упаковки об'єкта Stream, відкриваючи таким чином доступ до послідовного порту класу, який використовує потік. Цей пакет використовується для зчитування даних із послідовного порту комп'ютера.

Формат повідомлень(фреймів) даних, які використані в роботі, мережі контролерів складається з 11 бітного ідентифікатора, який може позначати пристрій який відправив повідомлення, наступний байт – кількість байт даних, далі ідуть самі байти даних.

77E 05 62 F4 0C 0B C6 AA AA

- 77E – 11 бітний ідентифікатор, який може позначати пристрій який відправив повідомлення
- 05 – кількість байт даних
- 62 F4 0C 0B C6 – отримані дані
- AA AA – не використовуються

Рисунок 3.2.2 – Формат фрейму даних мережі контролерів

Щоб отримати параметр з повідомлення мережі контролерів потрібно ідентифікувати параметр та за певною формулою дані параметрі вирахувати. Наприклад, таке перетворення для отримання обертів двигуна на рис. 3.2.3.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		99

77E 05 62 F4 0C 0B C6 AA AA

- F4 0C – ідентифікатор параметру(Оберти двигуна)
- 0B C6 – дані параметру

Перетворення отриманих даних:

$$0BC6_{16} = 3014_{10}$$

$$3014/4 = 753 \text{ об./хв.}$$

Рисунок 3.2.3 – Приклад отримання параметру обертів двигуна з фрейму мережі контролерів

Для кожного параметру є свій ідентифікатор та своя формула перетворення даних параметру. Таблиця 3.1. демонструє приклади таких формул та ідентифікаторів.

Таблиця 3.1

Приклади ідентифікаторів та формул перетворення для фреймів мережі контролерів

Іденти-фікатор (hex)	Іденти-фікатор (Dec)	Повернено байтів даних	Опис	Мін. знач.	Макс. знач.	Одиниці	Формула
04	4	1	Пораховане навантаження двигуна	0	100	%	$\frac{100}{255}A$ (або $\frac{A}{2.55}$)
05	5	1	Температура охолоджуючої рідини двигуна	-40	215	°C	$A - 40$
06	6	1	Короткострокова адаптація палива — Бак 1	-100	99.2	%	$\frac{100}{128}A - 100$ (або $\frac{A}{1.28} - 100$)
07	7	1	Довгострокова адаптація палива — Бак 1				

08	8	1	Короткострокова адаптація палива — Бак 2				
09	9	1	Довгострокова адаптація палива — Бак 2				
0A	10	1	Тиск палива (манометричний тиск)	0	765	кПа	3A
0B	11	1	Абсолютний тиск впускного колектора	0	255	кПа	A
0D	13	1	Швидкість транспортного засобу	0	255	км/год	A
0F	15	1	Температура повітря на вході	-40	215	°C	A – 40
10	16	2	Датчик масової витрати повітря (MAF) швидкість потoku повітря	0	655.35	грами/сек	$\frac{256A + B}{100}$
11	17	1	Положення дросельної заслінки	0	100	%	$\frac{100}{255}A$

Налаштування програмного забезпечення перед використанням: вірно налаштувати appsettings.json відповідно до підказок у файлі для роботи з вибраним СОМ-портом – напишіть номер обраного порту(наприклад, СОМ3) на який у послідовний порт будуть приходити фрейми мережі контролерів, які надалі будуть проходити через обробку, швидкість в бодах (нормальна перевірена швидкість передачі інформації становить 115200), паритет(наприклад, None, але може бути також – Even, Mark, Odd та Space), стоп-біти(задає стандартну кількість стоп біт у байті), біти даних(задає стандартну кількість біт даних у байті), рукоствискання(задає протокол встановлення зв'язку передачі інформації послідовним портом з використанням значення рукоствискання,

					<i>123.УДК: 004.42[629.33/.34:53.083.7]</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		101

стандартно – None, але може бути також – RequestToSend, RequestToSendXOnXOff та XOnXOff), логування.

Інтерфейс повинен бути адаптивним(таким, щоб пристосовувався правильно до змін в розмірі дисплею), не відволікаючим під час їзди, простим(ненав'язливим), інформативним(максимум корисної інформації в мінімумі об'єму) та інтуїтивно зрозумілим.

Інтерфейс поділений на декілька значущих частин: поточний час, інформація про двигун(оберти двигуна в об\хв, температура охолоджувальної рідини, температура оливи), стан дверей(відкриті чи закриті і які саме) і освітлювальних приладів(вказівників повороту – право, ліво чи вимкнено), загальна інформація, загальний пробіг(одометр) та інші показники, наприклад стан ручного гальма.

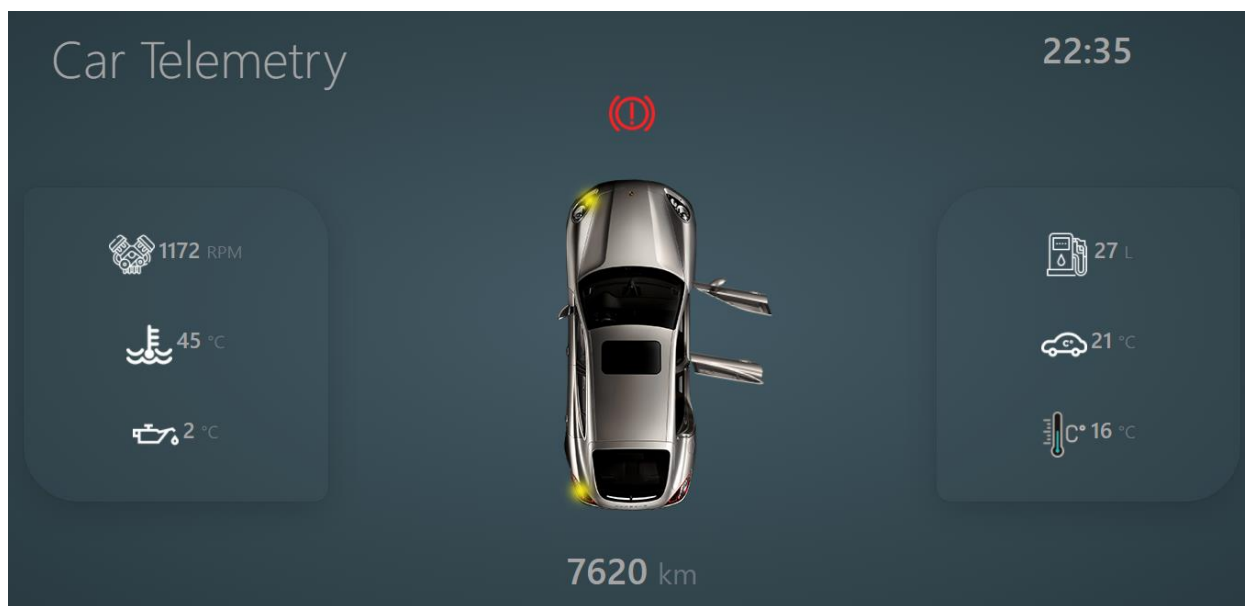


Рисунок 3.2.4 – Інтерфейс користувача

3.2.1. Опис алгоритму роботи

В роботі представлений такий алгоритм роботи програмного забезпечення:

- Відбувається підключення до вибраного в файлі конфігурації послідовного порту.
- Зчитування фреймів мережі контролерів із послідовного порту.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						102
Зм.	Арк.	№ докум.	Підпис	Дата		

- Розкладання зчитаного фрейму мережі контролерів на частини (ID та байти даних) для зручної роботи з фреймами.
- Аналіз фреймів мережі контролерів за допомогою набору обробників(хендлерів), кожен відповідає за свій тип фрейму мережі контролерів, цей набір можна розширювати, додавши обробник нового фрейму.
- Після цього оброблена інформація передається так, щоб інтерфейс користувача міг коректно відобразити отриману інформацію в режимі реального часу.

Програма обробки фрейму даних контролера пов'язана ланцюгом обов'язків – шаблон поведінки програмного забезпечення, який дозволяє послідовну передачу запитів через ланцюжок програми обробки.

Шаблон ланцюжка обов'язків використовується для досягнення слабкого зв'язку в розробці програмного забезпечення, коли запит від клієнта передається ланцюжку об'єктів для їх обробки. Пізніше об'єкт у ланцюжку самостійно вирішить, хто оброблятиме запит і чи потрібно надсилати запит наступному об'єкту в ланцюжку. Де та коли застосовується шаблон ланцюжка обов'язків: коли потрібно роз'єднати відправника запиту та одержувача, коли кілька об'єктів, визначених під час виконання, є кандидатами на обробку запиту, коли не потрібно явно вказувати обробника у своєму коді, коли потрібно надіслати запит до одного з кількох об'єктів, не вказуючи одержувача явно. Цей шаблон рекомендується, коли кілька об'єктів можуть обробляти запит, і хендлер не точно має бути якимось певним об'єктом. Крім того, обробник визначається під час виконання. Також є корисним те, що запит, який взагалі не обробляється жодним обробником, є дійсним випадком використання, це дуже зручно у цьому випадку, оскільки якщо щось піде не так та в послідовний порт попаде якийсь неправильне повідомлення, жоден з обробників не візьметься за нього, та він пройде через усі та буде «проігнорованим».

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						103
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.2.1.1 – Приклад шаблону ланцюжка обробників

Можуть бути представлені такі обробники фреймів мережі контролерів, які відповідають за стани важливих елементів та температури різних частин автомобіля.

Представлений такий набір обробників фреймів мережі контролерів:

- Стан дверей (відкриті або закриті, і які саме).
- Поточна витрата пального в літрах.
- Оберти двигуна в обертах за хвилину.
- Поточний рівень пального в баку.
- Стан стоянкового(ручного) гальма(ввімкнене чи вимкнене).
- Температура в салоні(в °С).
- Температура поза салоном(зовнішня), в °С.
- Температура оливи в двигуні, в °С.
- Одометр(загальний пробіг) в кілометрах.
- Стан покажчиків поворотів(право, ліво чи вимкнено).

Їх перелік можна розширити додаванням нових обробників, які будуть відповідати за інші види пакетів мережі контролерів, які вони зможуть обробити, та додавши їх гармонійно у інтерфейс користувача.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		104

РОЗДІЛ 4. ЕКОНОМІЧНА ЧАСТИНА

У економічному розділі цієї кваліфікаційної роботи приведена ціна на приклад готового виробу по тематиці роботи та розраховуються ціна на виготовлення та матеріали пристрою що розробляється. Приведені ціни були взяті з популярних інтернет-магазинів на момент написання даного розділу.

Для прикладу береться автомобільна магнітола виробника «Піонер» моделі ДМХ-3Ф93 з такими можливостями – сенсорний дисплей, підтримується пошук та приєднання до вай-фай мереж, вмонтований блютуз модуль, вхід для двох камер(наприклад, для передньої та задньої), юсб роз’єм(для пристроїв, яким потрібно споживати 5В для зарядки, чи для прослуховування музики чи перегляду відео матеріалів із картки флеш-пам’яті), навігація, керування кнопками з керма, та деякими іншими не значними можливостями.

Пристрій що розробляється складається із – корпусу із контактами для приєднання до роз’єму діагностики автомобіля, одноплатника Рпай із 10 дюймовим сенсорним дисплеєм, з плати СТМ32Ф103 та модуля ТЖА1050 та систему кріплення для одноплатника із дисплеєм в машині. За допомогою цього обладнання вже буде доступний такий функціонал – відображення пакетів із внутрішньої шини автомобіля, за допомогою яких можливо самому діагностувати наявні чи майбутні причини поломок транспортного засобу, вмонтований вай-фай та блютуз, отже надаються усі можливості пов’язані із безпроводними мережами, запуск будь-яких програм, які підтримуються операційно системою одноплатника - перегляд фото, відео, інтернет браузер та безліч інших можливостей. За допомогою інших модулів цей, і без того широкий функціонал, можна розширити – додати камеру, наприклад, заднього виду, щоб вона вмикалась при ввімкненні задньої передачі, модуль для використання навігації за допомогою системи глобального позиціонування, керування одноплатником можна реалізувати зчитавши з шини відповідний пакет, для прослуховування музики можна використати блютуз, оскільки він вмонтований у одноплатник, під’єднавшись до штатної магнітоли(якщо передбачений) або використати одноплатник як ЧМ-передавач і вибравши відповідну частоту у

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		105

магнітолі або вивести звук на зовнішній підсилювач потужності, який буде підключено до штатної акустики автомобіля.

Таблиця 4.1

Порівняння готового виробу та пристрою, що розробляється

Виріб	Компоненти	Ціна
Автомобільна магнітола, яка описана вище	-	28 499 гривень [25]
Пристрій, що розробляється в даній роботі	10 дюймовий дисплей – 3998 гривень[26], Рпай 4Б із 8Гб ПДД – 3500 гривень, модуль камери – 600 гривень, модуль системи глобального позиціонування – 285 гривень, плата СТМ32Ф103 – 256 гривень, модуль ТЖА1050 – 32 гривні, та корпус для під'єднання до роз'єму діагностики – 202 гривні.	8 873 гривні

Пристрій, що розробляється в даній роботі вийшов в ціні значно дешевший та зі значно більшим функціоналом, який можна самому розширювати.

ВИСНОВКИ

Автомобіль з кожним днем все більше перетворюється на центр обробки даних на колесах. Зміни, безумовно, відбуватимуться не по днях. Однак це не буде настільки ж далекоглядним моментом, як, наприклад, еволюція в телекомунікаціях від телеграфів до бакелітових телефонів з прокрутками до потужних смартфонів. Також необхідно попрацювати над зручністю використання нових систем. Найбільш технічно складний транспортний засіб мало придатний для користування, якщо технології здаються водієві занадто неосязними. До цього аспекту автовиробники повинні поставитися серйозно.

Для досягнення поставленої мети в роботі визначені і розв'язані такі завдання: було створено пристрій та додаток, які виконують наступні операції – для підключення до автомобіля було розроблено схему підключення приймача та емулятора шини та було налаштовано під'єднання до шини автомобіля через роз'єм діагностування, було налаштовано зчитування даних з внутрішньої шини автомобіля, для отримання даних було налаштовано отримання повідомлень формату фреймів мережі контролерів, була розроблена емуляція шини автомобіля, яка оновлює відповідні пакети мережі контролерів для розробки в умовах без під'єданого автомобіля, для аналізу фреймів мережі контролерів було створено алгоритм обробки, було використано набір обробників для аналізу обраних пакетів, який можна розширювати, було налаштовано отримання параметрів з CAN повідомлень, було розроблено та імплементовано інтуїтивно зрозумілий інтерфейс користувача, було налаштовано візуалізацію оброблених пакетів в режимі реального часу.

За програмну частину: фреймворк-бібліотека Блейзор відповідає за «невидиму» користувачу, «задню», частину програми, сам алгоритм обробки повідомлень мережі контролерів теж тут був описаний, як і набір обробників («хендлерів») також тут був описаний метод підключення програми до послідовного порту та налаштовано зчитування інформації з нього. В роботі були представлені такі обробники фреймів мережі контролерів, які відповідають за стани важливих елементів та температури різних частин автомобіля. Їх перелік

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						107
Зм.	Арк.	№ докум.	Підпис	Дата		

можна розширити додаванням нових обробників, які будуть відповідати за інші види пакетів, які вони зможуть обробити, в роботі були представлені такі обробники, які відповідають за такі види пакетів мережі контролерів: стан дверей (відкриті або закриті, і які саме), поточна витрата пального в літрах, оберти двигуна в обертах за хвилину, поточний рівень пального в баку, стан стоянкового(ручного) гальма, температура повітря в салоні(в °С), температура поза салоном(зовнішня), в °С, температура оливи в двигуні, в °С, одометр(загальний пробіг) в кілометрах, стан покажчиків поворотів(право, ліво чи вимкнено).

За вигляд інтерфейсу користувача була відповідальна фреймворк-бібліотека Бутстрап, яка дозволила поділити інтерфейс на декілька частин: поточний час, інформація про двигун, загальна інформація, загальний пробіг, відображення стану дверей та освітлювальних приладів, відображення інших показників, наприклад, стан стоянкового(ручного) гальма. Також ця бібліотека дозволила застосувати адаптивність для інтерфейсу користувача, тобто підлаштовуваність під зміну розмірів та пропорцій дисплею для його коректного відображення. Було використано таке поєднання кольорів, які не відволікатимуть водія під час поїздки. У інтерфейсі користувача для водія дуже важливі поняття простоти, структурованої інформації та інтуїтивно зрозумілих іконок та показів параметрів. Інформація в інтерфейсі мусить оновлюватись в режимі реального часу.

За апаратну частину відповідали: мережа контролерів в авто або в емуляторі, приймач мережі контролерів складався з плати STM32F103 разом з модулем ТЖА1050, емулятор шини автомобіля складався із плати Ардуіно разом з модулем МСПІ2515, в ролі одноплатника була плата РПай версії 4Б з 8Гб ПДД, інтерфейс користувача відображався на 10 дюймовому сенсорному дисплеї.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
						108
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Gereffi, Gary. (2018). Global Value Chains and Development. Redefining the Contours of 21st Century Capitalism. Cambridge University Press.
2. Energy Agency (IEA). 2018. Global EV Outlook 2018 [Online]. <https://www.iea.org/gevo2018/>.
3. Jullien, Bernard, and Tommaso Pardi. 2013. Structuring New Automotive Industries, Restructuring Old Automotive Industries and the New Geopolitics of the Global Automotive Sector. International Journal of Automotive Technology and Management 13 (2): 96–113.
4. Womack, John, Daniel Jones, and Daniel Roos. 1990. The Machine that Changed the World. New York: Rawson Associates.
5. Arbib, James, and Tony Seba. 2017. Rethinking Transportation 2020–2030. The Disruption of Transportation and the Collapse of the Internal Combustion Vehicle and Oil Industries. A RethinkX Sector Disruption Sector [Online]. <https://www.rethinkx.com/press-release/2017/5/3/new-report-due-to-major-transportation-disruption-95-of-us-car-miles-will-be-traveled-in-self-driving-electric-shared-vehicles-by-2030>.
6. Internet-Zeitschrift carit: „Das vernetzte Auto: Der Begriff Car-IT gewinnt in der Automobilindustrie massiv an Bedeutung – die künftige Rolle der Business-IT“, <http://www.car-it.com/der-begriff-car-it-gewinnt-in-der-automobilindustrie-massiv-an-bedeutung-die-kunftige-rolleder-business-it/id-0031942>.
7. VDA – Verband der deutschen Automobilindustrie: „Das vernetzte Fahrzeug“, <https://www.vda.de/de/themen/innovation-und-technik/vernetzung/das-vernetzte-fahrzeug.html>.
8. Internet-Zeitschrift carit: „GENIVI-Konferenz zum Internet der Dinge“, <http://www.car-it.com/genivi-konferenz-zum-internet-der-dinge/id-0039829>.
9. Verband der deutschen Automobilindustrie (VDA), „Vernetzung: Die digitale Revolution im Automobil“, <http://www.vernetzung-vda.de>.

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк. 109
Зм.	Арк.	№ докум.	Підпис	Дата		

10. BVDW: „10 Thesen zur Zukunft von Connected Cars“, <http://www.bvdw.org/medien/bvdwveroeffentlicht-thesenpapier-zu-connected-cars-digitalisierung-zwingt-autohersteller-zumumdenken?media¼6238>.
11. automobile - History of the automobile | Britannica [Електронний ресурс] – Режим доступу до ресурсу: <https://www.britannica.com/technology/automobile/History-of-the-automobile>.
12. Getötete und Verletzte in Deutschland [Електронний ресурс] – Режим доступу до ресурсу: <https://www.dvr.de/service/unfallstatistik2020/2020/2020/2020/getoetete-und-verletzte-in-deutschland>.
13. Research Note: Distracted Driving in Fatal Crashes, 2016 (dot.gov) [Електронний ресурс] – Режим доступу до ресурсу: <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812700>.
14. Sonja Rümelin. 2014. The Cockpit for the 21st century. Dissertation. Ludwig-Maximilians-Universität München.
15. Nora Broy. 2016. Stereoscopic 3D User Interfaces: Exploring the Potentials and Risks of 3D Displays in Cars. Dissertation. Universität Stuttgart. <https://doi.org/10.18419/opus-8851>.
16. ISO 7498-1 Information technology—Open Systems Interconnection—Basic Reference Model: The Basic Model, ICS 35.100.01, ISO, 1994.
17. ISO 11898 Road vehicles—Controller area network (CAN), ICS 43.040.15
18. ISO 11519 Road vehicles—Low-speed serial data communication, ICS 43.040.15, heute mit ISO 11898 zusammengefasst, ISO, 1994.
19. S. Rani, R. Maheswar, G.R. Kanagachidambaresan, P. Jayarajan, Integration of WSN and IoT for smart cities (Springer, Cham, 2020)
20. S.J. Johnston et al., Commodity single board computer clusters and their applications. Future Gener Comput Syst, 12 (2018).
21. OBD — Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/OBD>.

					<i>123.УДК: 004.42[629.33/.34:53.083.7]</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		110

22. STM32F103 - Arm Cortex-M3 Microcontrollers (MCU) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.st.com/en/microcontrollers-microprocessors/stm32f103.html>.
23. TJA1050 | High-Speed CAN Transceiver | NXP Semiconductors [Електронний ресурс] – Режим доступу до ресурсу: <https://www.nxp.com/products/interfaces/can-transceivers/legacy-can/high-speed-can-transceiver:TJA1050>.
24. MCP2515 | Microchip Technology [Електронний ресурс] – Режим доступу до ресурсу: <https://www.microchip.com/en-us/product/MCP2515>.
25. Автомагнітола Pioneer DMH-ZF93 [Електронний ресурс] – Режим доступу до ресурсу: https://rozetka.com.ua/pioneer_dmh_zf9350bt/p247859959/.
26. 10.1" дисплей сенсорний Waveshare 1280x800 [Електронний ресурс] – Режим доступу до ресурсу: <https://miniboard.com.ua/display/360-1-displej-sensornyj-waveshare-1280x800.html>.

					<i>123.УДК: 004.42[629.33/.34:53.083.7]</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		111

HARDWARE:

CAN-RECEIVER.INO:

```

/*
Prerequisites:
STM32duino core: https://github.com/stm32duino/Arduino_Core_STM32
exoCAN library: https://github.com/exothink/eXoCAN
*/
#include <arduino.h>
#include <eXoCAN.h>
#define bluePillLED PC13
int id, fltIdx;
uint8_t rxbytes[8];
eXoCAN can;
void setup() {
  Serial1.begin(115200);
  can.begin(STD_ID_LEN, BR250K, PORTA_11_12_XCVR);
  can.filterMask16Init(0, 0, 0x7ff, 0, 0);
  pinMode(bluePillLED, OUTPUT);
  Serial1.println("Start program");
}
uint32_t last = 0;
char output_str[28];
char output_format[] = "%.3X %.2X %.2X %.2X %.2X %.2X %.2X %.2X %.2X";
void loop() {
  if (can.receive(id, fltIdx, rxbytes) > -1) {
    digitalWrite(bluePillLED);
    sprintf(output_str, output_format, id, rxbytes[0], rxbytes[1], rxbytes[2], rxbytes[3],
    rxbytes[4], rxbytes[5], rxbytes[6], rxbytes[7]);
    Serial1.println(output_str);
  }
}

```

EMULATOR.INO:

```

/*
Prerequisites:
arduino-mcp2515 library: https://github.com/autowp/arduino-mcp2515
*/
#include <SPI.h>
#include <mcp2515.h>
#include "Emulator_parameters.h"
// uncomment line below to switch output mode to serial
// #define SERIAL_MODE
MCP2515 mcp2515(10);
can_frame* parameters[] = {&door_msg, &handbrake_msg, &outdoor_temperature_msg,
&fuel_level_msg, &coolant_temp_msg, &rpm_msg, &oil_temp_msg, &indoor_temp_msg,
&steering_switch_msg, &current_consumption_msg, &odometer_msg};
uint8_t n_parameters = 11;
#ifdef SERIAL_MODE
  char output_str[28];
  char output_format[] = "%.3X %.2X %.2X %.2X %.2X %.2X %.2X %.2X %.2X";
#endif
void setup() {
  Serial.begin(115200);
  mcp2515.reset();
  mcp2515.setBaudrate(CAN_500KBPS);
  mcp2515.setNormalMode();
  randomSeed(analogRead(0));
  //change_parameters_randomly();
  Serial.println("Start program");
}
void loop() {
  // send CAN messages
  for (uint8_t i = 0; i < n_parameters; i++) {
    #ifdef SERIAL_MODE
      mcp2515.sendMessage(parameters[i]);
    }
}

```

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		112


```

struct can_frame current_consumption_msg = {
    .can_id = 0x77E,
    .can_dlc = 8,
    .data = {0x05, 0x62, 0x22, 0x98, 0x00, 0x00, 0xAA, 0xAA}
};
float current_consumption_raw = 0;
struct can_frame odometer_msg = {
    .can_id = 0x77E,
    .can_dlc = 8,
    .data = {0x05, 0x62, 0x22, 0x03, 0x24, 0xC0, 0xAA, 0xAA}
};
uint32_t odometer_raw = 0;
// function for setting parameters value randomly
void change_parameters_randomly() {
    // invert state of random door
    door_i = random(4);
    door_state[door_i] = !door_state[door_i];
    update_door_msg();
    // set random handbrake state
    handbrake_raw = random(2);
    update_handbrake_msg();
    // set random outdoor temperature value
    outdoor_temperature_raw = random(0, 30);
    update_outdoor_temperature_msg();
    // set random fuel level
    fuel_level_raw = random(50);
    update_fuel_level_msg();
    // set random coolant temperature value
    coolant_temp_raw = random(150);
    update_coolant_temp_msg();
    // set random rpm
    rpm_raw = random(3000);
    update_rpm_msg();
    // set random oil temperature
    oil_temp_raw = random(50);
    update_oil_temp_msg();
    // set random indoor temperature value
    indoor_temp_raw = random(25);
    update_indoor_temp_msg();
    // set random steering switch value
    steering_switch_raw = random(3);
    update_steering_switch_msg();
    // set random current consumption value
    current_consumption_raw = random(200) / 10.0;
    update_current_consumption_msg();
    // set random odometer value
    odometer_raw = random(200000);
    update_odometer_msg();
}
// functions for updating CAN messages according to raw value
void update_door_msg() {
    door_msg.data[4] = 0;
    for (uint8_t i = 0; i < 4; i++) {
        door_msg.data[4] = door_msg.data[4] | door_state[i] << (i * 2);
    }
}
void update_handbrake_msg() {
    if (handbrake_raw) {
        handbrake_msg.data[4] = 0x21;
    } else {
        handbrake_msg.data[4] = 0x20;
    }
}
void update_outdoor_temperature_msg() {
    outdoor_temperature_msg.data[4] = outdoor_temperature_raw * 2 + 100;
}
void update_fuel_level_msg() {
    fuel_level_msg.data[4] = fuel_level_raw;
}
void update_coolant_temp_msg() {

```

					<i>123.УДК: 004.42[629.33/.34:53.083.7]</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		114

}

INTERFACES:

ICANMESSAGECOMPOSER.CS:

```
using CT.BusinessLogic.Entities;
namespace CT.BusinessLogic.Interfaces
{
    public interface ICanMessageComposer
    {
        CanMessage Compose(string rawMessage);
    }
}
```

IHANDLER.CS:

```
using CT.BusinessLogic.Entities;
namespace CT.BusinessLogic.Interfaces
{
    public interface IHandler
    {
        IHandler SetNext(IHandler handler);

        CanMessage Handle(CanMessage inCanMessage);
    }
}
```

ISERIALSERVICE.CS:

```
using System;
using System.Threading.Tasks;
namespace CT.BusinessLogic.Interfaces
{
    public interface ISerialService
    {
        event Func<string, Task> Notify;
    }
}
```

SERVICES:

CANMESSAGECOMPOSER.CS:

```
using CT.BusinessLogic.Entities;
using CT.BusinessLogic.Interfaces;
using System.Globalization;
namespace CT.BusinessLogic.Services
{
    public class CanMessageComposer : ICanMessageComposer
    {
        public CanMessage Compose(string rawMessage)
        {
            var data = rawMessage.Split(' ');
            return new CanMessage
            {
                Id = int.Parse(data[0], NumberStyles.HexNumber),
                Byte0 = byte.Parse(data[1], NumberStyles.HexNumber),
                Byte1 = byte.Parse(data[2], NumberStyles.HexNumber),
                Byte2 = byte.Parse(data[3], NumberStyles.HexNumber),
                Byte3 = byte.Parse(data[4], NumberStyles.HexNumber),
                Byte4 = byte.Parse(data[5], NumberStyles.HexNumber),
                Byte5 = byte.Parse(data[6], NumberStyles.HexNumber),
                Byte6 = byte.Parse(data[7], NumberStyles.HexNumber),
                Byte7 = byte.Parse(data[8], NumberStyles.HexNumber)
            };
        }
    }
}
```

						123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			117

SERIALSERVICE.CS:

```

using System;
using Microsoft.Extensions.Configuration;
using System.Threading.Tasks;
using System.IO.Ports;
using CT.BusinessLogic.Interfaces;
namespace CT.BusinessLogic.Services
{
    public class SerialService : ISerialService
    {
        public event Func<string, Task> Notify;
        private readonly IConfiguration Configuration;
        private readonly SerialPort _mySerialPort;
        public SerialService(IConfiguration configuration)
        {
            Configuration = configuration;
            var comPort = Configuration["ComPort"];
            var baudRate = int.Parse(Configuration["BaudRate"]);
            var parity = Configuration["Parity"];
            var stopBits = Configuration["StopBits"];
            var dataBits = int.Parse(Configuration["DataBits"]);
            var handshake = Configuration["Handshake"];

            _mySerialPort = new SerialPort(comPort)
            {
                BaudRate = baudRate,
                Parity = (Parity)Enum.Parse(typeof(Parity), parity, true),
                StopBits = (StopBits)Enum.Parse(typeof(StopBits), stopBits, true),
                DataBits = dataBits,
                Handshake = (Handshake)Enum.Parse(typeof(Handshake), handshake, true)
            };
            ;
            _mySerialPort.ReadTimeout = 500;
            _mySerialPort.WriteTimeout = 500;
            try
            {
                if (!_mySerialPort.IsOpen)
                {
                    SerialDataReceivedEventHandler(DataReceivedHandler);
                    _mySerialPort.Open();
                }
            }
            catch
            {
            }
        }
        public void DataReceivedHandler(object sender, SerialDataReceivedEventArgs e)
        {
            var myserialPort = (SerialPort)sender;
            try
            {
                var indata = myserialPort.ReadLine();
                if (Notify != null)
                {
                    Notify.Invoke(indata.ToString());
                }
            }
            catch
            {
            }
        }
    }
}

```

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		118

CANHANDLERS:

ABSTRACTHANDLER.CS:

```
using CT.BusinessLogic.Interfaces;
using CT.BusinessLogic.Entities;
namespace CT.BusinessLogic.Services.CanHandlers
{
    public abstract class AbstractHandler : IHandler
    {
        private IHandler _nextHandler;
        public IHandler SetNext(IHandler handler)
        {
            this._nextHandler = handler;
            return handler;
        }
        public virtual CanMessage Handle(CanMessage inCanMessage)
        {
            if (this._nextHandler != null)
            {
                return this._nextHandler.Handle(inCanMessage);
            }
            else
            {
                return null;
            }
        }
        protected void SetValue(CanProperties property, string value)
        {
            DataDictionary.aData[property] = value;
        }
    }
}
```

COOLANTTEMPHANDLER.CS:

```
using CT.BusinessLogic.Entities;
namespace CT.BusinessLogic.Services.CanHandlers
{
    public class CoolantTempHandler : AbstractHandler
    {
        public override CanMessage Handle(CanMessage canMessage)
        {
            if (canMessage.Id == 0x77E && canMessage.Byte3 == 0x05 && canMessage.Byte2 ==
0xF4)
            {
                var coolantTemp = canMessage.Byte4 - 40;
                SetValue(CanProperties.CoolantTemperature, coolantTemp.ToString());
                return canMessage;
            }
            else
            {
                return base.Handle(canMessage);
            }
        }
    }
}
```

CURRFUELCONSUMPTIONHANDLER.CS:

```
using CT.BusinessLogic.Entities;
namespace CT.BusinessLogic.Services.CanHandlers
{
    public class CurrFuelConsumptionHandler : AbstractHandler
    {
        public override CanMessage Handle(CanMessage canMessage)
        {
            if (canMessage.Id == 0x77E && canMessage.Byte3 == 0x98 && canMessage.Byte2 ==
0x22)
            {
                var currFuelConsumption = (double)canMessage.Byte5 / 10;
            }
        }
    }
}
```

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		119

```

        SetValue(CanProperties.CurrentFuelConsumption,
currFuelConsumption.ToString());
        return canMessage;
    }
    else
    {
        return base.Handle(canMessage);
    }
}
}

```

DOORHANDLER.CS:

```

using System;
using System.Collections.Generic;
using CT.BusinessLogic.Entities;
namespace CT.BusinessLogic.Services.CanHandlers
{
    public class DoorHandler : AbstractHandler
    {
        public override CanMessage Handle(CanMessage canMessage)
        {
            if (canMessage.Id == 0x77E && canMessage.Byte3 == 0x0D && canMessage.Byte2 ==
0x22)
            {
                var showByteInString = Convert.ToString(canMessage.Byte4, 2).PadLeft(8,
'0');

                //Byte4 in form 00000000, 0 mean open, 1 close. e.g. 01000100
                var doorIndex = AllIndexesOf(showByteInString, "1");
                if (Array.IndexOf(doorIndex, 7) == -1)
                {
                    SetValue(CanProperties.DoorFrontLeft, "Open");
                }
                else
                {
                    SetValue(CanProperties.DoorFrontLeft, "Close");
                };
                if (Array.IndexOf(doorIndex, 5) == -1)
                {
                    SetValue(CanProperties.DoorFrontRight, "Open");
                }
                else
                {
                    SetValue(CanProperties.DoorFrontRight, "Close");
                };
                if (Array.IndexOf(doorIndex, 3) == -1)
                {
                    SetValue(CanProperties.DoorBackLeft, "Open");
                }
                else
                {
                    SetValue(CanProperties.DoorBackLeft, "Close");
                };
                if (Array.IndexOf(doorIndex, 1) == -1)
                {
                    SetValue(CanProperties.DoorBackRight, "Open");
                }
                else
                {
                    SetValue(CanProperties.DoorBackRight, "Close");
                };
                return canMessage;
            }
            else
            {
                return base.Handle(canMessage);
            }
        }
    }
}

```

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		120


```

public static int[] AllIndexesOf(string str, string substr, bool ignoreCase =
false)
{
    if (string.IsNullOrEmpty(str) ||
        string.IsNullOrEmpty(substr))
    {
        throw new ArgumentException("String or substring is not specified.");
    }

    var indexes = new List<int>();
    var index = 0;

    while ((index = str.IndexOf(substr, index, ignoreCase ?
StringComparison.OrdinalIgnoreCase : StringComparison.Ordinal)) != -1)
    {
        indexes.Add(index++);
    }

    return indexes.ToArray();
}
}
}

```

HANDLERSSEQUENCE.CS:

```

using CT.BusinessLogic.Entities;

namespace CT.BusinessLogic.Services.CanHandlers
{
    public class HandlersSequence : AbstractHandler
    {
        public override CanMessage Handle(CanMessage canMessage)
        {
            var door = new DoorHandler();
            var inTemp = new InTempHandler();
            var outTemp = new OutTempHandler();
            var handbrake = new HandbrakeHandler();
            var fuelLevel = new FuelLevelHandler();
            var coolantTemp = new CoolantTempHandler();
            var engineRpm = new EngineRpmHandler();
            var oilTemp = new OilTempHandler();
            var turnSignals = new TurnSignalsHandler();
            var currFuelConsumption = new CurrFuelConsumptionHandler();
            var totalKm = new TotalKmHandler();
            door.SetNext(inTemp);
            inTemp.SetNext(outTemp);
            outTemp.SetNext(handbrake);
            handbrake.SetNext(fuelLevel);
            fuelLevel.SetNext(coolantTemp);
            coolantTemp.SetNext(engineRpm);
            engineRpm.SetNext(oilTemp);
            oilTemp.SetNext(turnSignals);
            turnSignals.SetNext(currFuelConsumption);
            currFuelConsumption.SetNext(totalKm);
            door.Handle(canMessage);
            return base.Handle(canMessage);
        }
    }
}

```

WEB:

INDEX.RAZOR:

```

@page "/"
@inherits IndexModel
@using CT.BusinessLogic.Entities;

<div class="container-fluid">
    <div class="row">
        <div class="col-md-10">

```

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		121

```

        <h1 class="display-4">Car Telemetry</h1>
    </div>
    <div class="col-md-2">
        <h1>@DateTime.Now.ToString("HH:mm")</h1>
    </div>
</div>
<div class="row">
    <div class="col-md-12">
        <div class="top-data size-box">
            @if (DataDictionary.aData[CanProperties.Handbrake] == "1")
            {
                <div class="icons-container">
                    
                </div>
            }
        </div>
    </div>
</div>
<div class="row align-items-center">
    <div class="col-md-3 side-info">
        <div class="row mt-5 mb-5">
            <div class="col-md-12">
                <div class="text-center">
                    <div class="">
                        
                        <span class="h4">@DataDictionary.aData[CanProperties.EngineRpm]
<small class="text-muted">RPM</small></span>
                    </div>
                </div>
            </div>
        </div>
        <div class="row mt-5 mb-5">
            <div class="col-md-12">
                <div class="text-center">
                    <div class="">
                        
                        <span
class="h4">@DataDictionary.aData[CanProperties.CoolantTemperature] <small class="text-
muted">&#8451</small></span>
                    </div>
                </div>
            </div>
        </div>
        <div class="row mt-5 mb-5">
            <div class="col-md-12">
                <div class="text-center">
                    <div class="">
                        
                        <span
class="h4">@DataDictionary.aData[CanProperties.OilTemperature] <small class="text-
muted">&#8451</small></span>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="col-md-6">
        <div class="car">
            
            @if (DataDictionary.aData[CanProperties.TurnSignal] == "Left")
            {
                
                
            }
            else if (DataDictionary.aData[CanProperties.TurnSignal] == "Right")

```

					<i>123.УДК: 004.42[629.33/.34:53.083.7]</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		122

```

        {
            
            
        }
        
            
                
                    
                </div>
            </div>
        <div class="col-md-3 side-info side-info-right">
            <div class="row mt-5 mb-5">
                <div class="col-md-12">
                    <div class="text-center">
                        <div class="icons-container">
                            
                            <span class="h4">@DataDictionary.aData[CanProperties.FuelLevel]
<small class="text-muted">L</small></span>
                        </div>
                    </div>
                </div>
            </div>
            <div class="row mt-5 mb-5">
                <div class="col-md-12">
                    <div class="text-center">
                        <div class="">
                            
                            <span
class="h4">@DataDictionary.aData[CanProperties.IndoorTemperature] <small class="text-
muted">&#8451</small></span>
                        </div>
                    </div>
                </div>
            </div>
            <div class="row mt-5 mb-5">
                <div class="col-md-12">
                    <div class="text-center">
                        <div class="">
                            
                            <span
class="h4">@DataDictionary.aData[CanProperties.OutdoorTemperature] <small class="text-
muted">&#8451</small></span>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <div class="row">
            <div class="col-md-12">
                <div class="text-center">
                    <h1 class="text-center">@DataDictionary.aData[CanProperties.TotalKm] <small
class="text-muted">>km</small></h1>
                </div>
            </div>
        </div>
    </div>

```

						123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
							123
Зм.	Арк.	№ докум.	Підпис	Дата			

INDEX.RAZOR.CS:

```
using Microsoft.AspNetCore.Components;
using System.Threading.Tasks;
using CT.BusinessLogic.Interfaces;
namespace CT.Web.Pages
{
    public class IndexModel : ComponentBase
    {
        [Inject] ICanMessageComposer CanMessageComposer { get; set; }
        [Inject] IHandler Handler { get; set; }
        [Inject] ISerialService Notifier { get; set; }
        protected override async Task OnInitializedAsync()
        {
            Notifier.Notify -= GetCurrentValue;
            Notifier.Notify += GetCurrentValue;
        }
        public async Task GetCurrentValue(string inCanCommand)
        {
            Handler.Handle(CanMessageComposer.Compose(inCanCommand));
            await InvokeAsync(StateHasChanged);
        }
    }
}
```

					123.УДК: 004.42[629.33/.34:53.083.7]	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		124