

ПРИЛАДИ

УДК 621.3

DOI <https://doi.org/10.32782/2663-5941/2023.2.1/06>**Дядюн Н.А.**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**Корнєв В.П.**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПРИСТРІЙ НИЗЬКОРІВНЕВОГО ТЕСТУВАННЯ АВТОМОБІЛЬНОЇ ШИНИ ДАНИХ CAN

Головна ціль пристрою, що описується, спростити процес діагностики автомобільної електроніки, а саме електронних блоків керування, які поєднані між собою за шиною даних CAN. Досвід показує, що можливо створити пристрій, який би відповідав певним вимогам практики. Розроблений пристрій буде доцільним при низькорівневій діагностиці електронної системи автомобіля. Перевагою розробки є повноцінний доступ до шини даних CAN, який майже завжди обмежений при використанні спеціалізованого діагностичного обладнання [1]. Окрім цього, функціонал девайсу розширюється наявністю двох незалежних контролерів CAN та підвищуючим перетворювачем живлення до 24V. Пряме підключення до комп'ютера і можливість використовувати розробку як HID (Human Interface Device) пристрій ще більше розширює його можливості і призначення. Все це разом робить прилад більш універсальним і придатним для застосування дослідження як легкової так і вантажної техніки, а процес діагностики досить ефективним, і в деяких випадках значно швидшим. Відмічено, що процедура проведення діагностики істотно ускладнюється не досконалістю, а часто і відсутністю документації на вузли електронної системи автомобіля. Особливо це стосується техніки, яка останнім часом потрапляє в Україну, особливі умови експлуатації якої спонукають до більш частого обслуговування і ремонту. Тому важливим і актуальним є створення засобів і методики проведення такої низькорівневої діагностики в умовах неповної інформації про електронну систему автомобіля. Розглянуті в роботі приклади тестування деяких вузлів конкретного вантажного автомобіля можна вважати початком розробки такої методики, основаної на принципах так званого «реверс інжинірингу» складних систем.

Ключові слова: автомобіль, шина даних CAN, низькорівнева діагностика, реверс інжиніринг, передача даних.

Постановка проблеми. Сучасний автомобіль пронизаний сотнями метрів дротів різноманітного призначення. З плином часу кількість технологій, що використовується одночасно тільки зростає, а це приводить в свою чергу до необхідності поєднувати блоки керування між собою додатковими провідниками. Однак навіть не це підштовхнуло на розробку CAN, а перспектива легкої реалізації нового функціоналу, зручного обміну даними між всіма блоками керування в рамках однієї мережі. З цією метою на початку 80х років 20 століття було розпочато роботу над розробкою нового мережевого стандарту.

Відома компанія Bosch в 1983 році провела аналіз існуючих систем послідовної передачі даних, проте ні одна з взятих не покривала

повністю вимог, які ставилися до автомобільного інжинірингу. Того ж року розпочалася робота над новою системою, яка потім отримала назву CAN (Controller Area Network). На самих початкових фазах проекту були залучені інженери з Mercedes-Benz для розробки специфікації, а також спеціалісти з Intel. В лютому 1986 року на конгресі SAE (Society of Automotive Engineers) світ побачив CAN [2]. Мережевий протокол був побудований на арбітражі доступу [3], коли повідомлення з найвищим пріоритетом опрацьовувалися в першу чергу без затримок. Вже на той час було реалізовано декілька механізмів відслідковування помилок та можливість автоматичного відключення несправного блоку від шини.

Першими протокол CAN в чіпі реалізували Intel в середині 1987 року – це був 82526, трохи згодом свою мікросхему випустила Philips з назвою 82C200. Чіпи мали деякі відмінності, проте фактично виконували одну й ту ж функцію – зняти необхідність в обробці, прийому і відправці, фільтрування пакетів з мікроконтролера або мікропроцесора. Першими ж, хто застосував CAN вже на практиці стали Mercedes-Benz та BMW, які з початку 1991 року розпочали встановлювати просту мережу для поєднання блоків керування двигуном та іншими критично важливими електронними вузлами [4]. Знайти шину даних CAN в ті роки можна було тільки в люксових моделях автовиробників, наприклад BMW 7 серії.

Аналіз існуючих рішень. Кожен автовиробник надає свій діагностичний пристрій і відповідне до нього програмне забезпечення для діагностики автомобіля, де враховуються особливості електронної системи транспортного засобу. В таких програмах, окрім виведення кодів помилок, можливості проведення параметрії, можна знайти ще й рекомендації по ремонту для тої чи іншої помилки, які зводяться до принципу «перевірити це, якщо ні – наступний крок». В продажі можна знайти також універсальні пристрої, або ж багато-маркові, які можуть працювати з широким спектром марок і моделей автомобілів. Наприклад, широко розповсюджений серед аматорів і навіть професіоналів справи Autocom [5]. Його вартість значно нижча за дилерський пристрій, але найголовнішою перевагою є те, що купивши такий девайс, можна діагностувати майже всі легкові автомобілі, так само, як і значну частину вантажних і комерційних транспортних засобів. Недоліком є універсальний доступ до електронної системи авто, тому деякі функції будуть недоступні. Також великий ризик втратити доступ до електронного блока керування у випадку неправильної послідовності дій, збою при передачі тощо.

Постановка завдання. Мета розробки пристрою є можливість діагностувати вузли автомобіля, що підключені до шини даних CAN, з безпосереднім доступом до них. Використання широко спеціалізованих або навіть дилерських діагностичних пристроїв може не забезпечити повноту даних, які можна побачити або змінити. Описаний девайс має велику перевагу – це перегляд всього потоку даних, що проходять в шині, можливість виявити недостачі, або некоректність в повідомленнях. Також, з допомогою такого пристрою відкривається можливість не тільки діа-

гностувати, а і симулювати роботу окремо взятого блоку керування або виконуючого пристрою.

Однією з цілей розробки є можливість одночасної роботи з двома незалежними мережами, для чого потрібно використовувати два контролери шини CAN. На відміну від 1990 років, вибір елементної бази зараз набагато ширший і більш доступний. В якості контролера слугує мікросхема MCP2515 виробника Microchip, а також CAN трансивер в мікросхемі TJA1050 виробника NXP. Контролери шини реалізують обмін даними з мікроконтролером з допомогою послідовної шини SPI. В якості мікроконтролера обрано досить поширений ATmega32u4, можливості якого покривають поставлені потреби в об'ємі пам'яті, кількості входів/виходів і швидкодії. Окрім того, мікросхема може працювати в режимі HID пристрою, що відкриває масу додаткових застосувань. Блок схема вузла, що підключений до шини даних CAN наведено на рис. 1 [4].

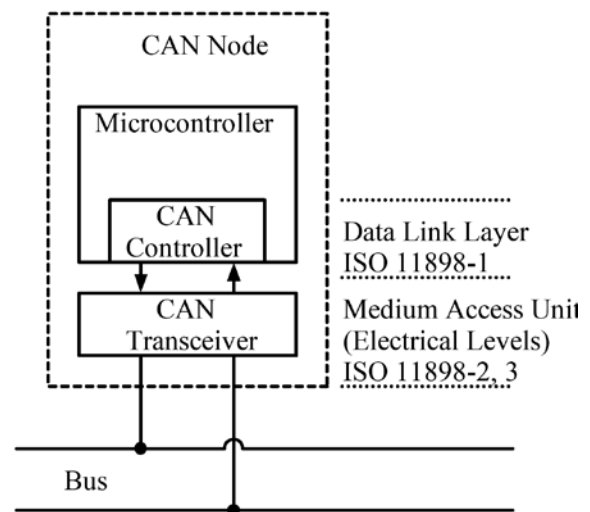


Рис. 1. Схема вузла, що підключений до шини CAN

Як видно, кожен вузол повинен мати:

- Мікроконтролер, або мікропроцесор, який би відправляв, отримував і опрацьовував дані;
- CAN контролер, який формуватиме пакети даних, займатиметься обробкою помилок, сигналізуванням про них, і багато чого іншого на рівні прийому-передачі;
- CAN трансивер, який буде контактувати з фізичною шиною даних, перетворювати рівні напруги, забезпечувати захист тощо.

Всі вузли фізично з'єднані між собою звичайною двопровідною шиною. Дроти являють собою скручену пару з номінальним опором 120 Ом. Стандарт ISO 11898 [6] розширює межі до 108 ÷ 132 Ом.

Опис функціонування та особливостей шини даних CAN. Мережевий протокол реалізує дуже цікавий механізм передачі повідомлень – так званий арбітраж доступу [3]. Оскільки шина даних є послідовною, при одночасній передачі можуть виникнути колізії, що призводить до затримок, втрат даних і інших неприємних явищ.

Один пакет CAN окрім корисних даних розміром до 8 байт, має системні та контрольні поля, і одним з важливих є поле арбітражу. Фактично поле арбітражу представляє собою ідентифікатор пакету. В залежності від призначення шини, поле може бути як 11 бітним, так і 29 бітним. Чим менший ID пакета, тим більший його пріоритет, це наочно зможемо побачити нижче на часовій діаграмі сигналів. Коли серед декількох вузлів, що передають дані буде обрано лише один – з найвищим пріоритетом, решта перейде в режим прослуховування. Це один з недоліків механізму – повідомлення з досить низьким пріоритетом можуть ніколи не відправитися.

На рисунку 2 наведено приклад часових діаграм сигналів, коли в шині, на якій підключено чотири вузли, одночасно починається передача пакетів даних.

Бачимо, що після передачі признаку початку пакета (Start Of Frame) на шині починається формування ідентифікаторів всіх чотирьох пакетів. Для цього застосовується механізм логічного «І». Кожен вузол, що передає дані, перед встановленням наступного рівня на шину перевіряє її стан. Якщо в мережі зараз встановлений домінуючий рівень (що є фактично «логічний нуль») – вузол

може встановити тільки такий же домінуючий рівень. Інакше – вузол переводиться в режим прослуховування.

Таким чином серед всіх пакетів як пакет найвищого пріоритету обирається той, кількість домінуючих бітів (коли на шині низький логічний рівень) в ідентифікаторі якого найбільша.

Специфікація CAN поділена на три абстрактних рівня, а саме [8]:

- **Об’єктний рівень (Object layer)** – представляє собою безпосередньо повідомлення (пакети). Забезпечує фільтрування, встановлення статусів передачі, обробку нових пакетів.

- **Рівень передачі (Transmit layer)** – більшість стандартів CAN опирається якраз на цей рівень, оскільки саме він представляє собою детектування помилок, підтвердження повідомлень, арбітраж доступу, формування нових пакетів, дотримання таймінгів та багато чого іншого, але не менш важливого. Цей рівень стоїть над фізичним, та проводить безпосередню обробку даних.

- **Фізичний рівень (Physical layer)** – фактично забезпечує дротове з’єднання вузлів між собою. Головними вимогами є загальний опір шини, який має становити 120 Ом, термінуючі резистори, які є методом подавлення шумів та відбивання сигналів. Окрім цього, рівень встановлює стандарти на кольори дротів, з’єднувачі (роз’єми), що застосовуються, інші вказівки на придушення шумів та завад, рівні напруги домінуючого і рецесивного станів. Час від часу фізичний рівень поповнюється рекомендаціями задля покращення

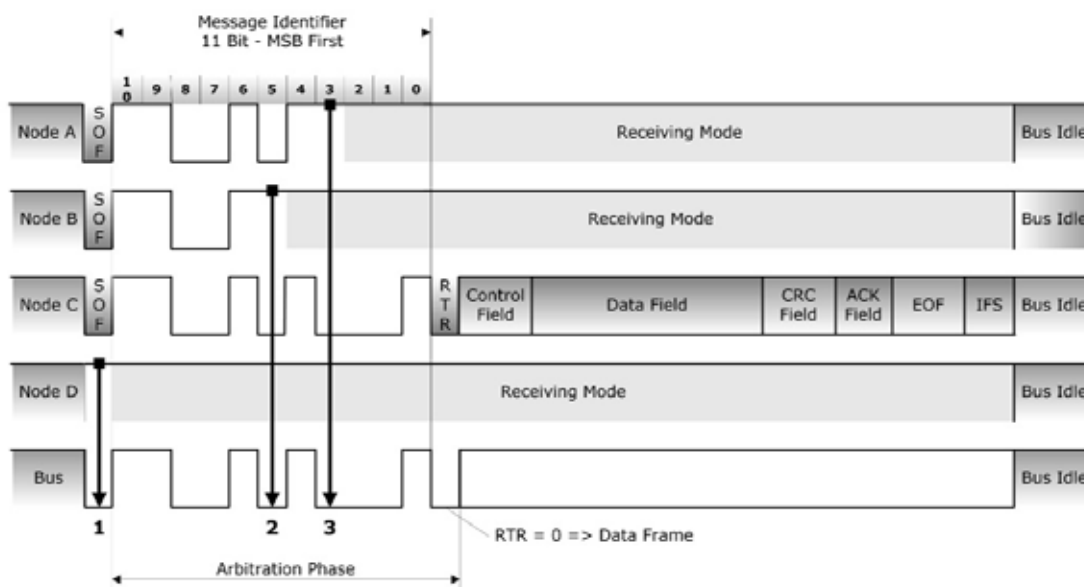


Рис. 2. Часова діаграма сигналів при одночасній відправці пакетів

передачі, зменшення ризику втрати повідомлення.

Мікросхеми-контролери CAN шини в проекті мають досить високу швидкість обміну даними до 1Mbps, чого більш ніж достатньо для використання майже з будь якою технікою.

До речі специфікація CAN ISO 11898 [6], що розроблена Bosch, пропонує два варіанти швидкостей шин даних [7]:

- **High Speed CAN** – найбільш розповсюджений стандарт, швидкість передачі даних в якому становить від 40 Кбіт/с до 1 Мбіт/с. Кожен кінець шини має термінуватися резистором номіналом 120 Ом.

Приклад часової діаграми передачі даних показано на рисунку 3.

- **Low Speed CAN, або Fault Tolerant CAN** – реалізує швидкість передачі даних від 40 Кбіт/с до 125 Кбіт/с. Характеризується можливістю продовжувати передачу інформації навіть при пошкодженні одного з провідників. Термінуючий резистор при цьому встановлюється у кожному з вузлів на шині. Приклад передачі даних на рисунку 4.

З наведених на рисунку діаграм стає чітко видно, як низько швидкісна шина CAN може продовжувати передачу даних навіть при пошкодженні одного з дротів [8].

Справа в тому, що кожен з сигнальних провідників (CANH і CANL) мають свої доміантний і рецесивний рівні, на відміну від High Speed CAN, де CANH представляє собою провідник тільки для встановлення доміантного рівня на шині, а CANL тільки рецесивного. Рівні сигналів перекривають один одного, що враховується трансиверами. При виникненні пошкодження, шина переходить на одно дрововий режим.

Деяко незвичними можуть показатися терміни доміантного і рецесивного стану. Специфікація CAN уникає використання станів шини як логічні 1 або 0, замість цього прийнято вживати слова рецесивний і доміантний рівні відповідно. Головна умова для фізичного рівня – це можливість доміантного рівня подавлювати рецесивний і не навпаки.

Термінуючі резистори [9] застосовують в мережі незалежно від швидкості передачі даних

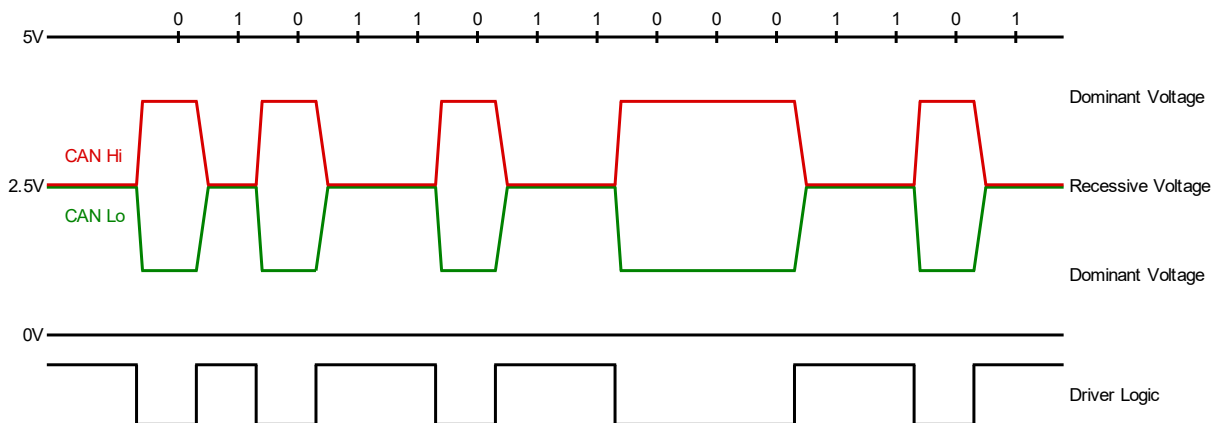


Рис. 3. Часова діаграма сигналу High Speed CAN

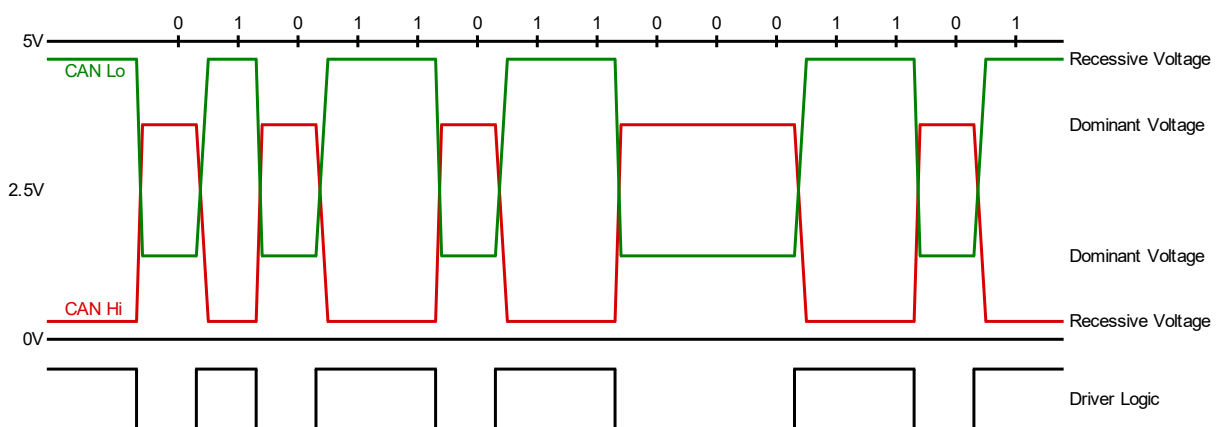


Рис. 4. Часова діаграма сигналу Low Speed CAN або Fault Tolerant CAN

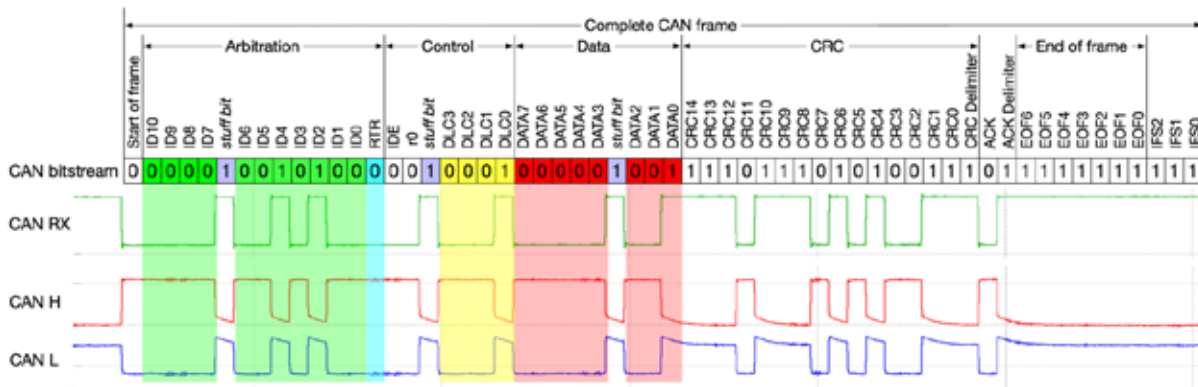


Рис. 5. Приклад передачі одного пакету даних в мережі CAN

в ній. Питання лише в якій кількості і де. Використання такого опору зумовлене можливістю відбивання сигналу на кінцях кабелю або впливом електромагнітних завад. Сила цих явищ тільки зростає зі збільшенням швидкості передачі даних і зростанням довжини кабелю. Описані проблеми можуть пошкоджувати сигнал, а гасіння напруги на кінцях шини забезпечує якісну передачу сигналу. На платі розробленого пристрою встановлені два термінуючих резистори, які підключаються до шини з допомогою перемикачів, що забезпечує швидке їх вмикання в електричне коло.

Завершуючи теорію, звернемо увагу на приклад передачі одного пакету даних на шині CAN зображений на рис. 5. На часовій діаграмі зображено логічний стан на шині (CAN RX), і напруги на лініях CANH та CANL. Зеленим кольором виділено передачу поля арбітражу, синім – біт закінчення поля арбітражу. Жовтим помічено передачу DLC, або довжини корисних даних. Червоним – корисні дані. Решта це чек-сума, і індикація закінчення пакету. При передачі даних видно, що стани на лініях CANH і CANL змінюються інверсно, разом з витою парою це забезпечує збільшену завадостійкість. До того ж бачимо, що у випадках, коли кількість бітів, що йдуть поспіль один за одним і мають нульове значення (тобто в лінії тривалий час не змінюється низький рівень), доходить до 5-ти, у бітову послідовність примусово вставляється одиничний біт (*stuff bit*), з ціллю зниження ймовірності помилок у наслідок можливої втрати синхронізації.

Схемотехнічні і конструктивні рішення. На рисунку 6 надано схему електричну принципову пристрою, яку виконано з поблочним поданням функціонально завершених елементів. Окрім мікроконтролера, і двох CAN інтерфейсів, на схемі також є два USB Type-C роз'єми для забезпечення

живлення і передачі даних до комп'ютера, дві тактові кнопки загального призначення (для завдання режимів роботи пристрою), декілька світлодіодів для індикації стану та DC-DC перетворювач для підвищення живлення до 24В.

На рисунку 7 показано вид монтажної сторони друкованої плати, що розроблена згідно схеми електричної принциповій пристрою, а на рис. 8 фото зібраної плати.

Розглянемо основні рішення, що стосуються програмної підтримки роботи пристрою. Оскільки специфікація CAN, вимагає виконання конкретних умов тільки для фізичного рівню, то реалізація верхніх шарів протоколу, а значить і їх програмна реалізація залишається більш вільною для розробника. А значить для пристрою, що розроблено, може бути або написана власна програма, яка б виконувала тільки певні алгоритми, підтримуючи завданий функціонал, або використане існуюче готове програмне забезпечення. Для першого тестування пристрою вирішено обрати другий варіант. Оскільки схематично і апаратно пристрій підтримуватиме протокол Lawicel [10], то це дає можливість використовувати існуюче програмне забезпечення для перегляду CAN шини.

Експериментальні випробування пристрою. Вирішено перевірити роботу пристрою в декількох режимах, а саме у *режимі прослуховування*, коли відбувається врізання в діючу шину для перегляду в неї реального потоку даних, і в *режимі трансивера*, коли генерування і передача даних в мережу здійснюється самим пристроєм.

Для першого режиму до моторної і інформаційної CAN мереж автомобіля DAF XF105 були підключені відповідні сигнальні дроти. Налаштувавши необхідні опції в програмі CAN Viewer, обравши потрібний USB порт і швидкість передачі даних, розпочато процес прослуховування.

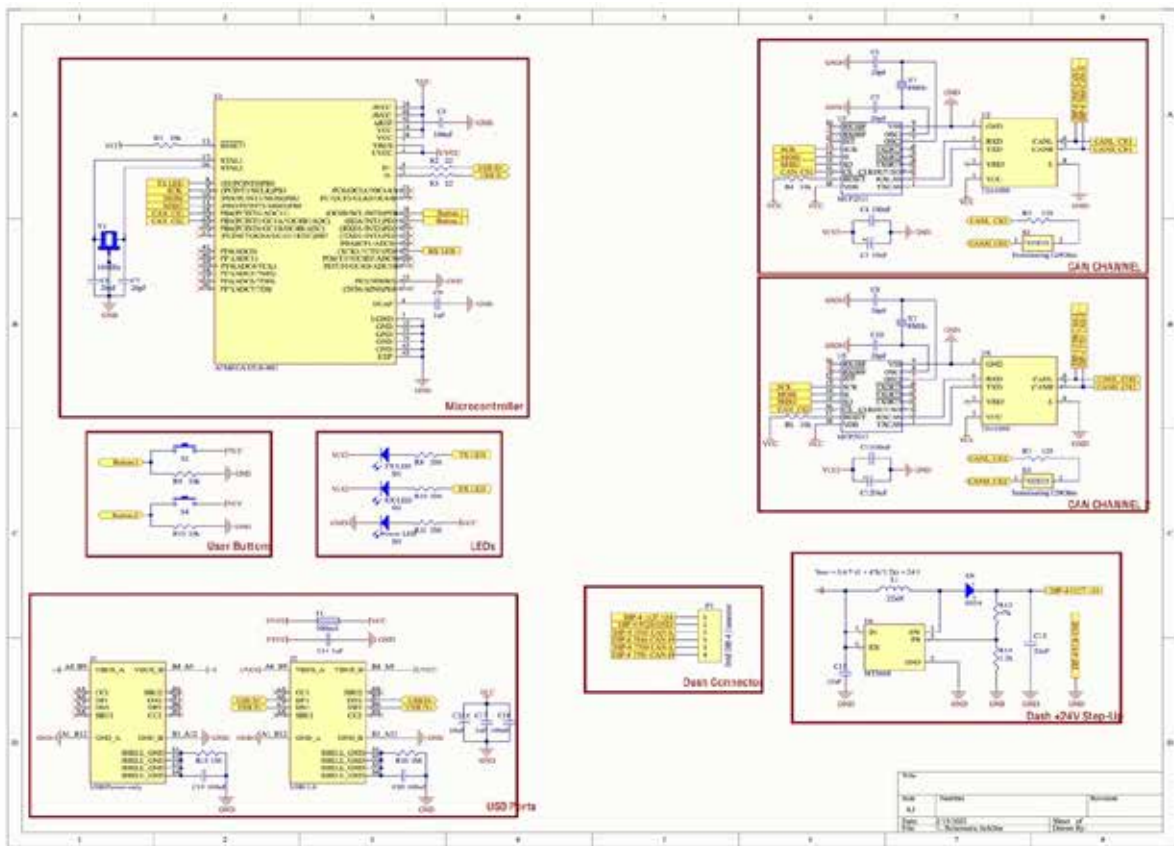


Рис. 6. Електрична принципова схема розробленого пристрою

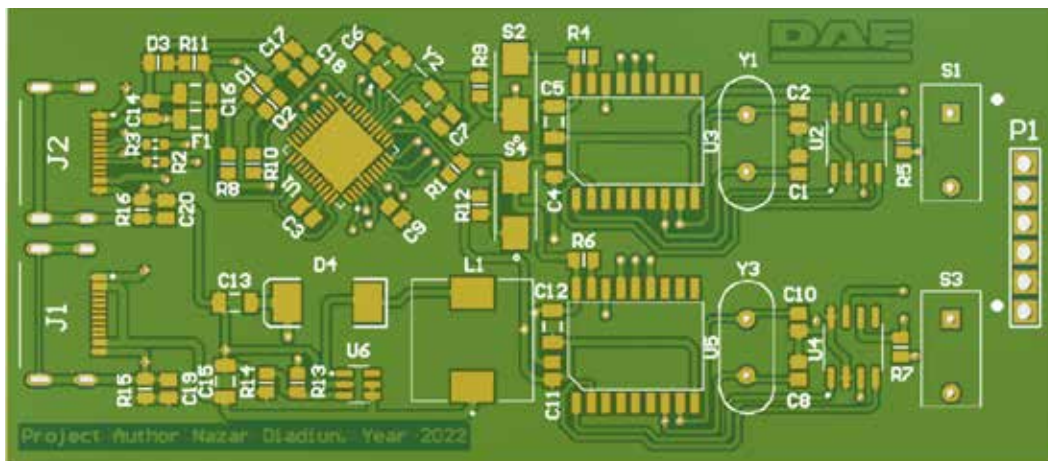


Рис. 7. Зовнішній вигляд друкованої плати

Оразу ж на екрані комп'ютера, підключеного до пристрою, відобразилися безліч повідомлень з різноманітними ідентифікаторами і даними всередині (рис. 9).

На рисунку видно декілька полів, перші три відповідають за відносний час з початку запису, ідентифікатор пакету і довжину даних в ньому. Решта – корисні дані в шістнадцятковій системі. Наочно видно щільність передачі даних. Інтервал становить 10-20 мс. Враховуючи, що на рисунку показано відфільтрова-

ний потік даних, які мають лише одну адресу пакетів, то у реальному потоці можуть бути сотні пакетів від різних пристроїв на шині.

Для тестування в режимі трансивера до пристрою була підключена панель приборів від того ж автомобіля. Оскільки на платі є підвищуючий перетворювач живлення, то піклуватися про забезпечення панелі напругою від зовнішнього джерела не потрібно. Достатньо просто підключити ще один USB кабель.

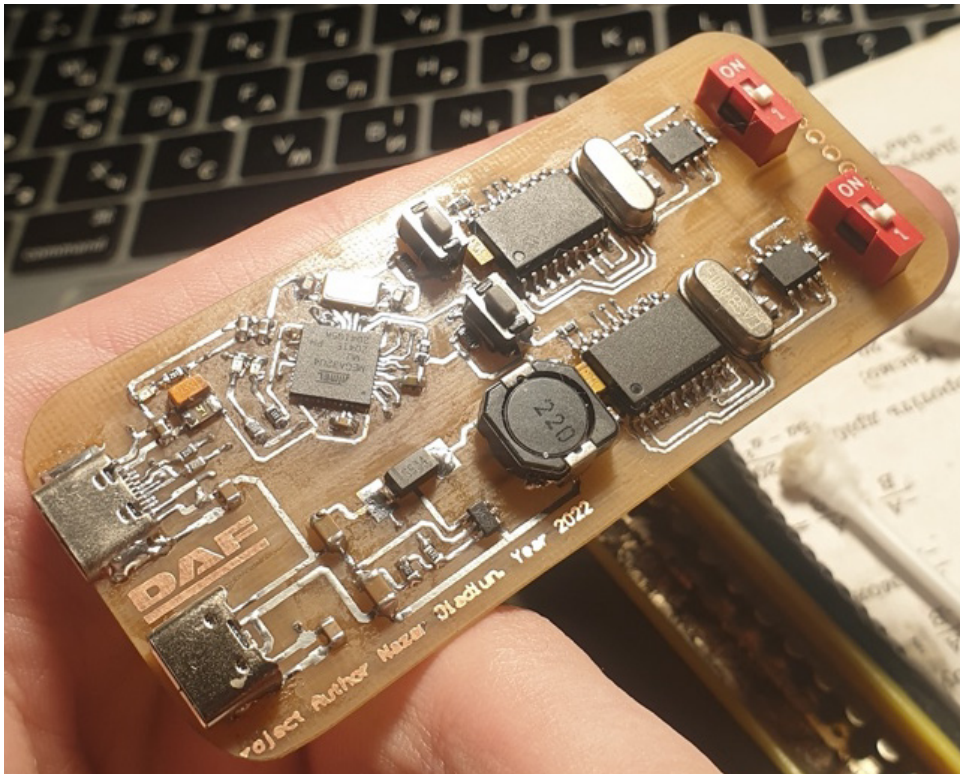


Рис. 8. Готовий вид друкованої плати з змонтованими на ній елементами

05.787	18FF3527	8	22	0	0	0	0	0	0	0
05.816	18FF3527	8	10 0F	0A	0A		1	0	0	0
05.826	18FF3527	8	21	0	0	0	0	0	0	0
05.836	18FF3527	8	22	0	0	0	0	0	0	0
05.866	18FF3527	8	10 0F	0A	0A		1	0	0	0
05.875	18FF3527	8	21	0	0	0	0	0	0	0
05.885	18FF3527	8	22	0	0	0	0	0	0	0
05.956	18FF3527	8	22	0	0	0	0	0	0	0
05.985	18FF3527	8	10 0F	0A	0A		1	0	0	0
06.000	18FF3527	8	21	0	0	0	0	0	0	0
06.007	18FF3527	8	22	0	0	0	0	0	0	0
06.034	18FF3527	8	10 0F	0A	0A		1	0	0	0
06.047	18FF3527	8	21	0	0	0	0	0	0	0
06.055	18FF3527	8	22	0	0	0	0	0	0	0
06.084	18FF3527	8	10 0F	0A	0A		1	0	0	0
06.099	18FF3527	8	21	0	0	0	0	0	0	0
06.107	18FF3527	8	22	0	0	0	0	0	0	0
06.133	18FF3527	8	10 0F	0A	0A		1	0	0	0
06.147	18FF3527	8	21	0	0	0	0	0	0	0
06.155	18FF3527	8	22	0	0	0	0	0	0	0
06.183	18FF3527	8	10 0F	0A	0A		1	0	0	0
06.199	18FF3527	8	21	0	0	0	0	0	0	0
06.206	18FF3527	8	22	0	0	0	0	0	0	0
06.223	18FF3527	8	10 0F	0A	0A		1	0	0	0
06.273	18FF3527	8	10 0F	0A	0A		1	0	0	0
06.283	18FF3527	8	21	0	0	0	0	0	0	0
06.294	18FF3527	8	22	0	0	0	0	0	0	0
06.305	18FF3527	8	10 0F	0A	0A		1	0	0	0
06.312	18FF3527	8	21	0	0	0	0	0	0	0
06.323	18FF3527	8	22	0	0	0	0	0	0	0
06.352	18FF3527	8	10 0F	0A	0A		1	0	0	0
06.363	18FF3527	8	21	0	0	0	0	0	0	0

Рис. 9. Отриманий файл за 500 мс часу

Спочатку у режимі прослуховування протягом декількох десятків хвилин записувався *Log-файл* даних з CAN шини реального автомобілю. Протя-

гом цього часу було проініційовано різноманітні важелі, що спричиняли ту чи іншу індикацію на панелі приладів (було запущено двигун, змінені



Рис. 10. Результат роботи пристрою в режимі трансивера в парі з панеллю приладів

його оберти і т.п.). Після запису файл було збережено. Для тестування в режимі трансивера, збережений файл відкрито знову, і всі дані, які було записано з визначеним інтервалом відправлялися через пристрій на панель приладів. Одразу після початку передачі даних на панелі приладів ввімкнулося підсвічування шкал приборів і екрану, здвинулися стрілки (рис. 10).

Причому індикатори вмикалися в такому ж хронологічному порядку, як і раніше вони вмикалися в автомобілі. На рисунку також видно, що швидкість руху, як це було і на реальному автомобілі, становить 50 км/г, а оберти двигуна близько 1300 об/хв. Це підтверджує дієвість очікуваної здатності пристрою симулювати реальні процеси, шляхом відправки відповідних пакетів у CAN шину.

Ще одне можливе застосування даного пристрою є певний реверс інжиніринг, коли при підключенні до шини «всліпу» знімається список пакетів за якийсь час, потім аналізується і робиться висновок про кількість і призначення даних, що передаються. Навіть без наявності опорного списку повідомлень можна проводити визначення належності пакетів, наприклад, для виконуючих пристроїв. В даному випадку панель приладів є таким пристроєм, а отже, простим спамом пакетів в її шину можна спостерігати вмикання індикаторів, здвиги стрілок, з чого робити певні висновки.

Слід відмітити, якщо на більш ранню електроніку автомобілів вже можна знайти якісь записи, технічну документацію і просто опис пакетів, напри-

клад, для блоку керування двигуном, то на пізніші моделі авто такого просто не буде. Використання описаного приладу і описаної методики зможе значно допомогти при визначенні несправності.

Додатковим, але не менш корисним призначенням можна назвати режим роботи шлюзу. Нерідкісні випадки, коли один або декілька агрегатів в автомобілі зазнають зміни на інші. І часто агрегати, що змінюються, мають інше покоління від решти електроніки автомобіля, що унеможливить їх роботу в парі. Кожен пакет даних несе в собі велику роль, і є незамінним при роботі всіх механізмів і електроніки, тому знаючи необхідні повідомлення, їх ідентифікатори і відповідні дані, можна всередині мікроконтролера видозмінювати пакети, надсилаючи їх далі в іншу шину. Таким чином, автомобіль буде «вважати», що електроніка агрегату рідна, а той же двигун – «вважатиме», що керується рідним блоком керування двигуна.

Висновки. Розглянуто історичні відомості щодо розробки та еволюції мережевого протоколу CAN. При аналізі декількох стандартів специфікації ISO 11898 приділено увагу особливостям фізичного рівня CAN шини. Це дало можливість сформулювати певні вимоги до структури пристрою, а саме, що крім стандартних функціональних блоків він має містити вбудований перетворювач напруги, для роботи з пристроями, живлення яких потребує 24В. Також необхідним стала наявність двох незалежних контролерів шини CAN. Все це сприяло прийняттю правиль-

них схемотехнічних і конструктивних рішень, що було експериментально підтверджено.

Проаналізовані джерела мережі інтернет на наявність існуючих та відкритих протоколів для роботи з шиною CAN, на які можна було б опиратися задля сумісності з існуючим зручним програмним забезпеченням.

Пристрій було протестовано в двох режимах роботи, а саме – в режимах прослуховування і трансиверу. Отримані результати навіть на такому простому розглянутому у статті прикладі, дають можливість стверджувати, що розроблений пристрій має великий потенціал для спрощення процесу діагностики автомобільної електроніки, і цілі, які покладалися на пристрій були досягнуті в повній мірі.

Також було доведено декілька додаткових можливих варіантів використання девайсу, які значно розширюють спектр його призначень. Результати випробування пристрою в режимі трансивера із опрацьованими спробами симулювання роботи

певних вузлів автомобілю, відкривають перспективи створення в наступному віртуальних повномасштабних тренажерів автомобілів, які можуть посприяти значному полегшенню і скороченню початкових етапів підготовки користувачів нової техніки.

Окрім використання в автомобілях, мережевий протокол приймає важливу роль в функціонуванні авіаційної та залізничної техніки, автоматизації будинків та індустріальних системах, медичних приладах та аграрній техніки. Маючи відповідні опорні записи, їх можна із застосуванням пристрою відтворювати у шині автомобілю або іншій технічній системі, чим самим спостерігати відмінності в поведінці у випадку несправностей. А це значить, що розглянуті в роботі приклади процедури тестування CAN шини, основаної на принципах реверс інжинірингу складних систем, можна вважати початком розробки методики низькорівневої діагностики різноманітних технічних систем із застосуванням розробленого пристрою.

Список літератури:

1. Діагностичне обладнання для роботи з автомобілями марки DAF. URL: <https://ad-instrument.com.ua/ua/p26071705-dilerskij-skaner-daf.html>. (Дата звернення 21.03.2023).
2. Історія виникнення CAN. URL: <https://www.can-cia.org/can-knowledge/can/can-history/>. (Дата звернення 19.03.2023).
3. Арбітраж доступу в мережі CAN. URL: <https://copperhilltech.com/blog/controller-area-network-can-bus-bus-arbitration/> (Дата звернення 19.03.2023).
4. Загальні відомості, опис шини даних CAN. URL: https://en.wikipedia.org/wiki/CAN_bus . (Дата звернення 19.03.2023)
5. Універсальний діагностичний пристрій Autocom. URL: <https://originaldiag.com/collections/autocom-diagnostics/products/autocom-cdp-flex-with-3-month-license> (Дата звернення 28.03.2023).
6. Опис стандарту ISO 11898. URL: <https://www.iso.org/standard/63648.html>. (Дата звернення 21.03.2023).
7. Опис фізичних рівнів CAN. URL: <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019LzHSAU> (Дата звернення 11.03.2023).
8. Опис фізичних рівнів і термінації шини CAN . URL: https://www.rpi.edu/dept/ecse/mps/CAN-LabVIEW_info/NI-Tutorial-9759-en.pdf. (Дата звернення 18.03.2023).
9. Використання і призначення термінуючих резисторів в мережі CAN . URL: <https://support.maxongroup.com/hc/en-us/articles/360009241840-CAN-bus-topology-and-bus-termination>. (Дата звернення 16.03.2023).
10. Опис протоколу Lawicel. URL: https://www.canusb.com/files/can232_v3.pdf (Дата звернення 28.03.2023).

Diadiun N.A., Kornev V.P. DEVICE FOR LOW-LEVEL DIAGNOSING OF AUTOMOTIVE CAN DATA BUS

The main purpose of the described device is to simplify the process of diagnosing automotive electronics in particular electronic control units, which are interconnected using a CAN data bus. Experience shows that it is possible to create a device that would meet certain requirements. The developed device will be appropriate for low-level diagnostics of the electronic system of the car. The advantage of development is full access to the CAN data bus, which is almost always limited when using specialized diagnostic equipment [1]. In addition, the functionality of the device is expanded by the presence of two independent CAN controllers and an up-converter of power supply to the 24V. A direct connection to a computer and the ability to use device as a HID (Human Interface Device) device further expands its capabilities and purpose. All this together makes the device more versatile and suitable for the study of both passenger and cargo equipment, and the diagnostic

process is quite effective, and in some cases much faster. The work also notes that the diagnostic procedure is significantly complicated not by perfection, but often by the lack of documentation for the electronic system of the car. This is especially true of equipment that has recently come to Ukraine, the special operating conditions of which encourage more frequent maintenance and repair. Therefore, it is important and relevant to create tools and methods for holding such low-level diagnostics in conditions of incomplete information about the electronic system of the car. The examples of testing some components of a particular truck considered in the work can be considered the beginning of the development of such a technique based on the principles of the so-called "reverse engineering" of complex systems.

Key words: *automotive, CAN data bus, low-level diagnosing, reverse engineering, data transfer.*