

ПРОГРАМНА МОДЕЛЬ ДАНИХ ДЛЯ ЗАБЕЗПЕЧЕННЯ РОБОТИ АВТОМОБІЛЬНОГО LIN-ПРОТОКОЛУ

© Павельчак А. Г., Яцук Ю. В., 2014

Запропоновано модель даних та механізми їхнього оброблення для забезпечення роботи LIN-протоколу на базі автомобільних мікроконтролерів AVR.

Ключові слова: LIN, AVR, мікроконтролер.

In this paper, the authors have proposed the data model and mechanisms for their processing that ensure functionality of LIN-protocol, based on automotive microcontrollers AVR.

Key words: LIN, AVR, microcontroller.

Вступ

Підвищення комфорту проїзду та безпеки водія є конкурентними факторами на автомобільному ринку. Для цього все частіше використовують сучасні технології з електронними блоками керування. З метою зменшення кількості кабелів, а відповідно і вартості, ваги, збільшення вільного простору в автомобілях використовують для обміну даними переважно лише протоколи з послідовною передачею даних. Найбільшого поширення на транспортних засобах набули послідовні інтерфейси CAN, LIN, MOST та FlexRay [1, 2].

CAN (Controller Area Network), який на міжнародному рівні стандартизований у 1994 р. є, без сумніву, найчастіше використовуваною системою послідовної шини. Ця система може бути застосована декілька разів навіть в межах одного транспортного засобу. По-перше, як система повільної CAN шини для побудови мережі з електронними блоками керування комфортом транспортного засобу з максимальною швидкістю передачі даних 125 Кбод (CAN low-speed). По-друге, як швидка CAN шина для організації мережі електронних блоків керування приводом та шасі транспорту з максимальною швидкістю передачі даних 1 Мбод (CAN high-speed).

LIN (Local Interconnected Network) використовується як дешева мережа для передачі простих даних від давачів та виконавчих елементів. Швидкість передачі даних обмежена всього 20 кбіт/с, але цього цілком достатньо для передачі некритичних до часу сигналів давачів та виконавчих елементів.

Для справді критичних до часу виконання завдань вже необхідна шинна система, що не залежить від завантаження шини та гарантує постійний реальний час виконання, а також є безпечною системою. FlexRay забезпечує максимальну швидкість передачі даних 20 Мбіт/с та гарантує рівновіддалену передачу даних, а також забезпечує детерміновані часові зв'язки. Другий канал зв'язку дає можливість передачі даних з резервуванням. Якщо передані дані на одному каналі пошкоджені, тоді вони зчитуються з іншого каналу. Інформаційно-розважальні пристрої, такі як навігація, радіо та телефон, потребують порівняно високої пропускної здатності, оскільки передаються не лише сигнали керування, але і відео- та аудіосигнали. Відповідно, MOST (Media-Oriented Systems Transport) використовується для організації мультимедійних мереж у транспортних засобах. MOST забезпечує порівняно велику пропускну здатність: для передачі бітових потоків (за частоти дискретизації 48 кГц) приблизно 23 Мбіт/с та 768 кбіт/с для передачі сигналів керування.

У системі послідовної шини електронні компоненти мають спільне середовище передачі та комунікації. Дані передаються біт за бітом і можуть бути прийняті, в принципі, всіма

учасниками шини, так званими шинними вузлами. У цьому випадку прокладення кабелів, на відміну від двоточкових з'єднань, є значно простішим. Це зменшує витрати, необхідний простір та вагу, а натомість покращує надійність та спрощує планування. Для послідовної передачі даних треба врахувати такі аспекти: кадрування (конструкція повідомлень), адресація (створення унікального розподілу між кадром та шинними вузлами), доступ до шини та дотримання часових меж, резервне копіювання даних та оброблення помилок, а також синхронізація.

Аналіз питання

Детальну інформацію про LIN-інтерфейс можна знайти в [3], а особливості функціонування апаратного LIN-контролера мікроконтролера ATmega32M1 в [4]. Специфікація LIN-інтерфейсу визначає не лише набір стандартів, необхідних для передачі даних, але також формулює єдиний технологічний процес, що дає змогу простіше і швидше розробляти мережі. Опис всієї LIN-мережі здійснюється за допомогою компоненти LDF. LDF визначає всі властивості мережі, що можна використати для автоматичної генерації програмних компонентів, які реалізують обмін даними. Крім того, LDF може застосовувати необхідну інформацію для різноманітного аналізу, вимірювання та інструментів тестування чи емуляторів шини. Специфікація мови конфігурування LIN описує синтаксис для створення LDF. Це достатньо просто, навіть щоб створити LDF вручну. Однак, звісно, це найкраще робити з використанням систем проектування. Починаючи з версії специфікації 2.0, введена додаткова єдина мова опису для вузлів. Ця мова, відома як мова опису вузлів LIN, дає можливість описувати стандартні вузли. Інформація зберігається у файлі NCF, що містить таку інформацію, як визначення повідомлень та сигналів, а також функції діагностики вузла.

Організація обміну даними в LIN кластері основана на Master-Slave архітектурі (рис. 1). Кластер містить Master-вузол (LIN Master) і щонайменше один Slave-вузол (LIN Slave). Задля здешевлення спеціалізований комунікаційний контролер не використовують. Натомість LIN комунікація реалізована програмними задачами у кожному вузлі, це так звані Slave-задачі. LIN Master також має Master-задачу, що використовується для координації комунікації кластера.

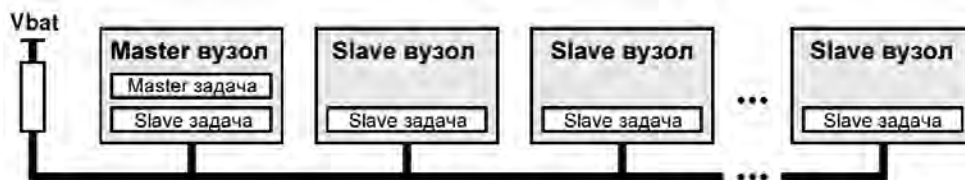


Рис. 1. Архітектура Master-Slave системи передачі даних

Координація забезпечується засобами періодичного виконання таблиці-розкладу LIN, що організована у вигляді кадрів у слотах. На початку кожного слота Master-задача посилає заголовок кадру з ідентифікатором кадру (ID), який всі Slave-пристрої аналізують у своїх Slave-задачах. Безпосередньо після заголовку кадру Slave-пристрій посилає відповідь, що асоційована з наданим ID. LIN кадр, що містить заголовок кадру та відповідь кадру, доступний для прийому кожним LIN-вузлом, завдяки адресації повідомлень на основі ID.

Структура LIN-повідомлення складається з двох частин (рис. 2): заголовка повідомлення (Frame Header), який формується Master-задачею; відповіді (Frame Response), яка формується Slave-задачею. Передача повідомлень в LIN-мережі відбувається лише з ініціативи Master-пристрою. Він відправляє на LIN-шину заголовок необхідного повідомлення. Заголовок повідомлення складається з: поля розриву синхронізації (Sync Break); поля синхронізації (Sync Field); захищеного ідентифікатора (PID, Protected Identifier).

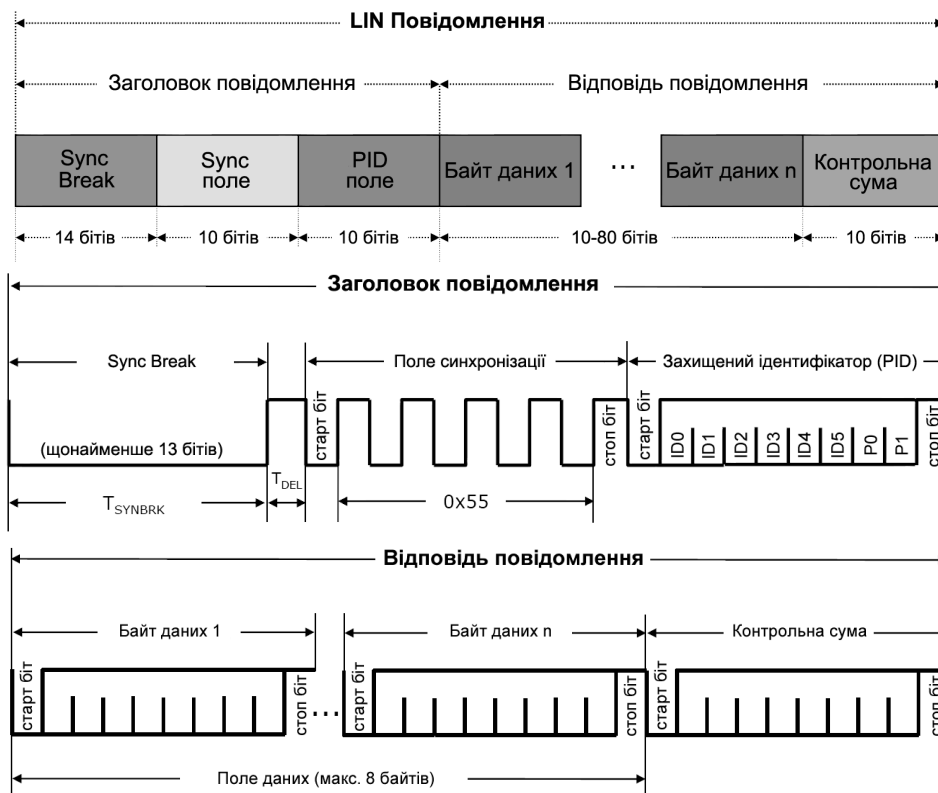


Рис. 2. Структура кадру LIN-повідомлення

Поле Sync Break використовується для сигналізування Slave-пристроїв, що починається новий кадр з повідомленням. Поле повинно містити щонайменше 13 бітових інтервалів основного (домінантного) рівня, після цього розділювач (break delimiter), який теж повинен мати інтервал не менше від 1 біта. Після цього Master посилає свою тактову частоту за допомогою поля синхронізації, відповідно до якої Slave-пристрої повинні синхронізуватися. Ці два поля, відповідно, призначені для ініціювання передачі даних та здійснення синхронізації з Master-пристроєм. Поле синхронізації містить значення 0x55. Для отримання тактового сигналу Slave-пристрої вимірюють час між першим і останнім спадним фронтом сигналу поля синхронізації та ділять його на вісім. Отриманий результат відповідає інтервалу одного біта.

Після поля синхронізації Master-пристрій передає ідентифікатор потрібного повідомлення. Ідентифікатори відображають комунікаційні зв'язки, що визначені у файлі конфігурації LDF. Також визначена поведінка відповідей окремо взятих Slave-пристроїв. Slave-пристрій може поводитися або пасивно (нічого не відповідати), або надіслати чи прийняти відповідь. Ідентифікатор має важливе значення, тому він захищений двома бітами парності P0 і P1. Ця комбінація ідентифікатора та асоційованих бітів парності і являє собою захищений ідентифікатор (PID).

Ідентифікатор повідомлення представляється шістьма бітами, відповідно, він може набувати значення від 0 до 63. Значення ідентифікатора повідомлення розділені на три категорії: значення від 0 до 59 (0x3B) використовуються для передачі робочих повідомлень; 60 (0x3C) та 61 (0x3D) використовуються для передачі діагностичних та конфігураційних даних; 62 (0x3E) та 63 (0x3F) зарезервовані для майбутнього вдосконалення протоколу.

Відповідь (message response) на вказаний ID відправляється Slave-задачею, що призначена для цієї цілі. У відповіді повідомлення максимально може бути передано до восьми байтів. Зазначимо, що передача починається з молодшого байта і молодшим бітом вперед. У принципі, відповідь повідомлення може бути отримана і сприйнята всіма Slave-задачами. Найчастіше кадр повідомлення містить заголовок від Master-пристрою і відповідь від Slave-пристрою. У файлі конфігурації LDF визначається, який вузол фактично використовуватиме дані, розміщені у відповіді.

Дані в повідомленні захищені контрольною сумою. Для LIN-протоколу визначено два типи контрольних сум: класична контрольна сума, де захищені лише байти даних; покращена контрольна сума, де захищені байти даних та PID ідентифікатор. Покращена контрольна сума використовується для ідентифікаторів від 0 до 59 та починаючи з версії 2.0 LIN-протоколу. Для збереження сумісності зверху вниз діагностичні повідомлення завжди захищені класичною контрольною сумою. Для формування контрольної суми окремі байти даних арифметично додаються за модулем 256. Це передбачає також додавання бітів переповнення до конкретного проміжного результату. Отриманий результат інвертується і Slave-задача передає його як контрольну суму. Приймач обчислює контрольну суму для отриманих байтів даних за таким самим алгоритмом, за винятком інверсії. Приймач виявляє помилку, якщо сума байтів даних та отримана з лінії контрольна сума не дорівнюють 0xFF.

Для забезпечення комунікації на LIN-шині протоколом визначено такі чотири категорії повідомлень: безумовні кадри; кадри, викликані подіями (ETF-кадри); спорадичні кадри; діагностичні кадри. З метою оптимізації LIN-мережі кожен з видів кадрів має своє функціональне призначення. При цьому окремо взятий вузол чи кластер загалом не зобов'язаний підтримувати усі перелічені типи кадрів.

Безумовні кадри (Unconditional frame). Як правило, використовуються для передачі корисних даних. Цей тип повідомлення відповідає стандартному кадру, який використовується без додаткової умови. Всі інші типи кадрів використовуються тільки в спеціальних випадках. Для безумовних кадрів використовуються ідентифікатори від 0 до 59. Безумовний кадр складається із заголовку та відповіді. Master-пристрій передає заголовок повідомлення на LIN-шину як запит. Тоді визначений slave-пристрій надсилає відповідь. Slave-задача на основі прийнятого ідентифікатора визначає, чи потрібно надсилати відповідь. Slave-задача Master-пристрою також може на основі значення ідентифікатора надіслати відповідь-повідомлення. У файлі конфігурації LDF описується, який саме вузол відправить асоційовану з ідентифікатором відповідь. Оскільки безумовний кадр містить унікальний зв'язок між заголовком та відповіддю, то в процесі комунікації не можуть виникати колізії під час передачі даних. Один слот кадру може бути використаний лише одним вузлом на шині. Відповідно, безумовні кадри ідеально підходять для періодичної передачі даних, оскільки час, коли повідомлення має з'явитися на шині, є передбачуваним.

Кадри, викликані подіями (ETF, Event triggered frame). Цей тип повідомлення використовується для передачі інформації на основі події, яку, за необхідності, потрібно передати вузлу LIN-мережі, тобто якщо Slave-пристрій має змінений (оновлений) сигнал, відповідно до цього запиту, то він передається. Насправді ETF-кадр еквівалентний стандартному безумовному кадру, з тією лише відмінністю, що декілька Slave-пристроїв може відповісти на заголовок від Master-пристрою. Для визначення автора відповіді його перший байт даних повинен містити адресу вузла (NAD). Якщо відповідь більше, ніж один Slave-пристрій, тоді виникає колізія, яку необхідно вирішувати за допомогою додаткової спеціальної таблиці вирішення цієї колізії. Ця таблиця активується на наступному кадрі після виникнення колізії. У таблиці вирішення колізій можуть міститися не лише асоційовані безумовні кадри, але й інші неасоційовані безумовні кадри, які до того ж можуть мати різну довжину даних.

Спорадичні кадри (Sporadic frame). Master-пристрій використовує спорадичні кадри для відправлення рідко використовуваної інформації (спорадичної інформації). Спорядичні кадри являють собою групу безумовних кадрів, які спільно використовують той самий слот. Коли спорадичний кадр підлягає передачі, приєднані безумовні кадри перевіряються на предмет оновлення сигналів. Якщо оновлених сигналів немає, тоді слот буде пустим (нічого відправлено не буде). Якщо є більше, ніж один кадр для передачі, то спершу передається кадр з найвищим пріоритетом. Решта кадрів кандидатів на передачу будуть передані в наступних сеансах (чергах передачі спорадичного кадру). У спорадичному кадрі тільки Master-пристрій передає безумовні

кадри (заголовки з відповіддю повідомлень). Спорадичним кадрам не присвоюються ідентифікатори, вони слугують своєрідними “заглушками” для передавання рідко використовуваних безумовних кадрів, кожен з яких має свій ідентифікатор.

Діагностичні кадри (Diagnostic frames). Вони завжди повністю забезпечують специфікацію транспортного рівня та мають завжди вісім байтів даних. Ідентифікатор діагностичного кадру є або 60 (0x3C), який називають кадром запиту Master-пристрою (master request frame), або 61 (0x3D), який називають кадром відповіді Slave-пристрою (slave response frame). Перед передаванням кадру запиту Master-пристрою Master-задача запитує свій діагностичний модуль, чи має кадр бути переданий, чи шина має мовчати. Заголовок кадру відповіді Slave-пристрою має бути посланий безумовно (беззастережно). Slave-задача публікується і підписується на відповідь, відповідно до свого діагностичного модуля.

Диспетчеризація. Master-пристрій контролює весь обмін даними у кластері. Відповідно, цей головний вузол має встановлену схему відправлення, яку проектує розробник системи і яка описана у файлі LDF. Це робить обмін даними в LIN-мережі передбачуваним, оскільки існує визначена послідовність фіксованих інтервалів часу. LIN-розклад (рис. 3) організований у слотах, які передбачені для передачі лише одного кадру для кожного слоту. Розмір слотів визначається за допомогою так званих міні-слотів. Міні-слот є періодичним інтервалом, з яким Master-задача опрацьовує LIN-розклад. Тривалість міні-слота формує базовий часовий інтервал для безперервної комунікації. Інакше кажучи, тривалість міні-слота – це той інтервал часу, через який спрацьовує налаштований для роботи з LIN-розкладом таймер мікроконтролера.

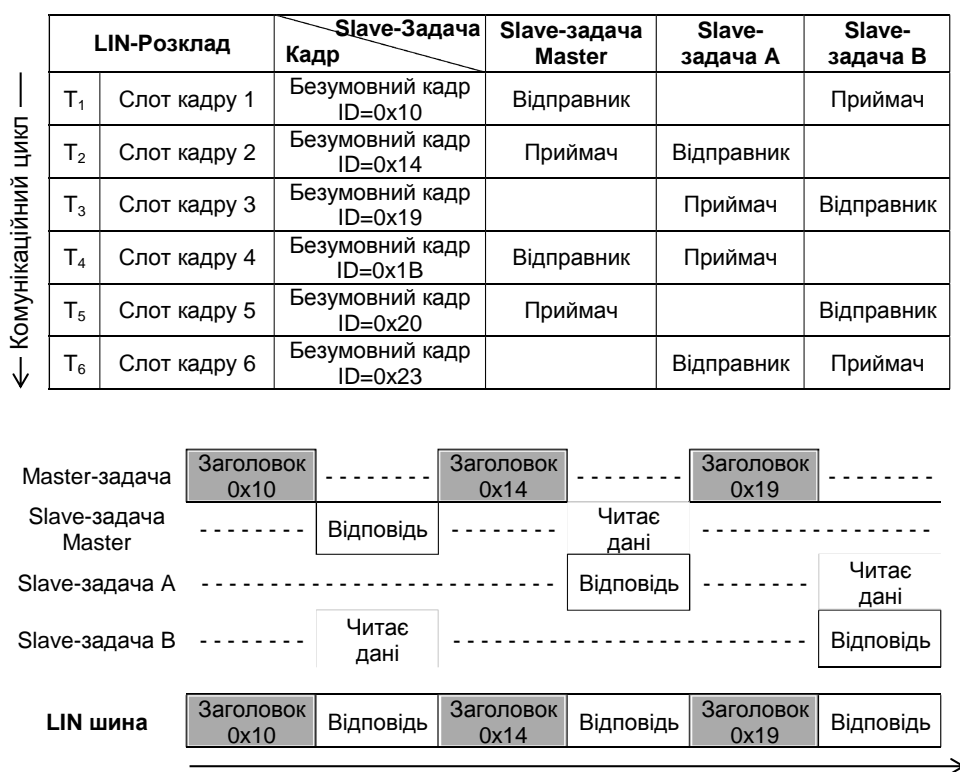


Рис. 3. Комунікаційна таблиця

Якщо розробник системи коректно обчислить часові інтервали для всіх кадрів, то ніяких колізій на LIN-шині під час обміну даними не виникне. Кожен слот повинен бути достатньо великим, щоб гарантувати передачу відповідного кадру. Оскільки часто використовуються не надто потужні мікроконтролери для організації LIN-мережі, то необхідно передбачати 40 % запаси часу для передавання повідомлень.

Постановка задачі

Програмна модель даних повинна забезпечувати ефективну роботу диспетчеризації повідомлень для LIN-мережі. Реалізація моделі орієнтована на використання апаратних контролерів LIN-протоколу мікроконтролерів AVR (на прикладі ATmega32M1).

Структура LIN моделі даних

Програмна LIN модель даних повинна забезпечувати всю функціональність щодо обміну даними на LIN-шині, початкову ініціалізацію Slave-вузлів, а також діагностику та тестування вузлів. Модель даних передбачає окремі структури як для Master-вузла (рис. 4), так і для Slave-вузлів (рис. 5). Своєю чергою, модель також має поділ на робочий та діагностичний модулі.

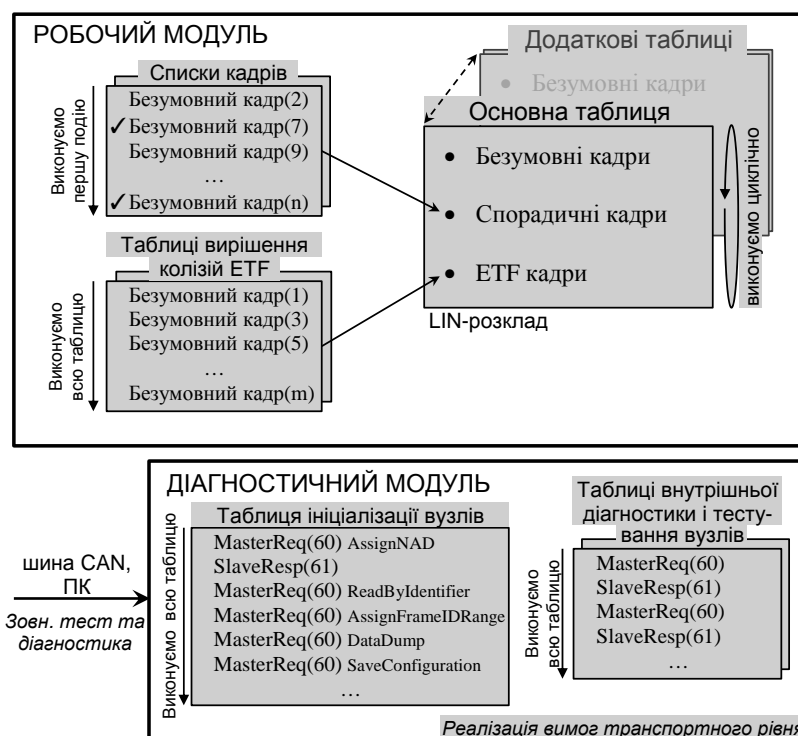


Рис. 4. Структура LIN моделі даних для Master-вузла

Master-вузол відповідає за весь процес обміну, що відбувається на LIN-шині. На початку своєї роботи (коли подано живлення) Master-вузол виконує ініціалізацію Slave-вузлів, тобто його діагностичний модуль повинен передати конфігурацію для кожного Slave-вузла, підключеного до LIN-шини, а також за необхідності може здійснити діагностику наявних Slave-вузлів. Надалі діагностику вузлів він виконуватиме вибірково у разі несправностей чи недотримання протоколу обміну даними з боку певних вузлів. Запит на проведення діагностики також може надійти з боку CAN-шини чи, якщо передбачено, з персонального комп'ютера, під'єданого до Master-пристрою.

Основна поточна робота виконується засобами робочого модуля Master-вузла. Цей модуль забезпечує диспетчеризацію робочих повідомлень, які, своєю чергою, можуть мати при цьому різну функціональність: несуть сигнали, байти даних чи вказівки від Master-вузла до визначеного Slave-вузла; несуть сигнали, байти даних чи вказівки від Master-вузла до декількох Slave-вузлів; несуть сигнали та байти даних від Slave-вузла до Master-вузла; несуть сигнали та байти даних від одного Slave-вузла до іншого Slave-вузла; несуть сигнали та байти даних від одного Slave-вузла до декількох вузлів.

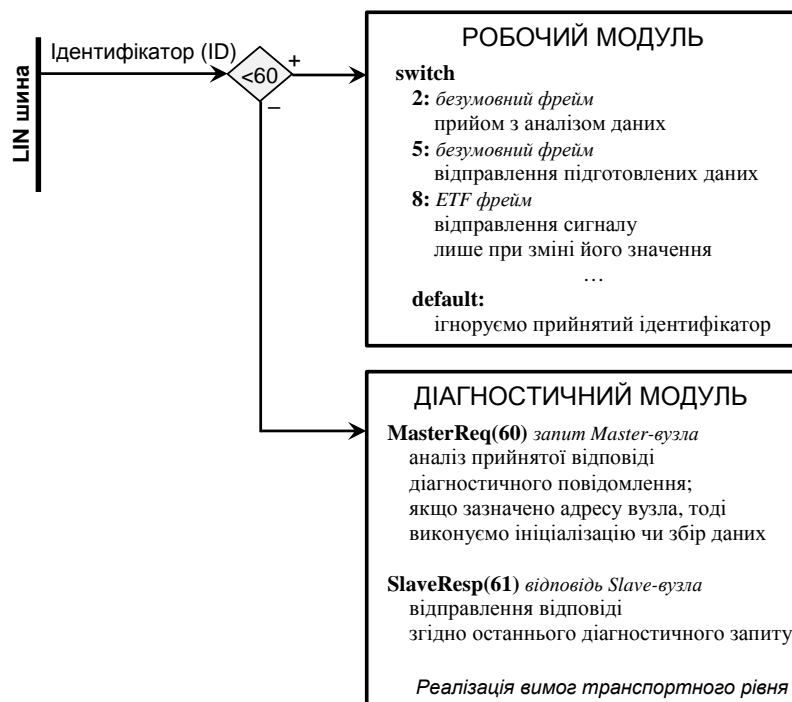


Рис. 5. Структура LIN моделі даних для Slave-вузла

Передавання повідомлень у ETF-кадрах обумовлене необхідністю надсилання рідкозмінних сигналів зі Slave-вузлів до Master-вузла. Тому передавання таких сигналів з різних Slave-вузлів може бути об'єднане в один часовий слот LIN-розкладу. Тут ми можемо розглядати декілька варіантів:

а) Master-вузол надіслав запит, однак жоден Slave-вузол не відповів. Це означає, що в Slave-вузлах сигнали, асоційовані з цим ETF-кадром, не змінювали свого значення. Master-вузол робить висновок про це на основі повідомлення тайм-аута, і отримана помилка про тайм-аут від периферійного LIN-модуля мікроконтролера у разі передачі ETF-кадрів є штатною ситуацією;

б) Master-вузол надіслав запит, і один зі Slave-вузлів відповів. Цей Slave-вузол, окрім зміненого значення сигналу (сигналів), обов'язково має надати у відповіді ще й адресу свого вузла (NAD), щоб основний вузол міг зорієнтуватися, від кого надійшла відповідь;

в) Master-вузол надіслав запит, і одразу декілька Slave-вузлів відповіли. Ця ситуація створює на LIN-шині колізію. Оскільки надіслані відповіді від різних Slave-пристроїв накладаються, то в кінцевому результаті Master-вузол фіксуватиме помилку контрольної суми, за якою і можна відслідковувати факт виникнення колізії. Одразу після виявлення колізії для ETF-кадру у роботі основної таблиці з LIN-розкладом буде зроблена пауза, а з наступного часового слоту розпочне свою роботу таблиця вирішення колізії для цього ETF-кадру. Ця таблиця складається з безумовних кадрів, які мають опитати усі Slave-вузли, що, ймовірно, могли надіслати відповідь. Крім них, можуть бути кадри й іншого призначення, що виконуватимуть збір чи передачу даних. Кадри, з яких складаються таблиці вирішення ETF-колізій, можуть при цьому мати різну довжину і не потребують передачі адрес вузлів (NAD).

На відміну від ETF-кадрів, де дані надходять з боку Slave-вузлів, спорадичні фрейми дають можливість Master-пристрою організувати відправлення рідкозмінюваних сигналів, що розміщуються в різних безумовних кадрах, в межах одного часового слоту. Це, звісно, зменшує таблицю LIN-розкладу та пришвидшує обмін даними. Для кожного спорадичного кадру має бути визначений список приєднаних безумовних кадрів, що можуть бути передані в межах одного часового слоту. Якщо на момент передачі спорадичного кадру декілька безумовних кадрів, що мають оновлені значення своїх сигналів, очікують на передавання, то в певний момент буде переданий лише кадр з найвищим пріоритетом. Решта безумовних кадрів з оновленими сигналами будуть передані згідно з

пріоритетом у наступній черзі в LIN-розкладі. Спорадичний кадр не має власного ідентифікатора (ID), передається ID безумовного кадру. Нормальною ситуацією на LIN лінії є, коли пустий слот спорадичного кадру. Формально, в специфікації не обумовлено, що довжина безумовних кадрів у спорадичному кадрі має бути однаковою, однак під час планування LIN-розкладу слід виділяти часові інтервали з розрахунку на найдовший кадр.

Структура LIN моделі даних для Slave-вузла (рис. 5) значно простіша за структуру моделі Master-вузла. Основне завдання полягає у прийманні ідентифікатора та, згідно з аналізом його значення і наявності у списку активних у робочому модулі Slave-вузла, обробленні прийнятих даних чи відправленні підготовлених даних.

Програмна реалізація LIN моделі даних

Для реалізації програмної LIN моделі даних необхідно передусім розробити структури даних, які б ефективно підтримували організацію комунікації. Ми не розглядатимемо реалізацію діагностичного модуля (через обмеження розміру статті), а зосередимо увагу лише на робочому модулі, що забезпечує передачу даних згідно з диспетчером завдань (LIN-розкладу).

Під час розроблення моделі було прийнято, що всі дані будуть зберігатися в RAM пам'яті. Якщо б необхідно було, з метою економії пам'яті, розміщувати структури даних у програмній пам'яті FLASH, то модель мала б дещо інший вигляд.

Основною цеглиною в моделі є окремих кадр. Тому для нього передбачена структура даних Frame (рис. 6). Елемент структури складається з двох полів: значення ідентифікатора та складового поля TypeLength, що забезпечує супровід кадру. Це дає нам можливість створити перелік необхідних кадрів. Зазначимо, що для спорадичних кадрів значення ідентифікатора має дорівнювати 0xFF. Це не має ніякого змістового навантаження і потрібно лише для візуальної зручності.

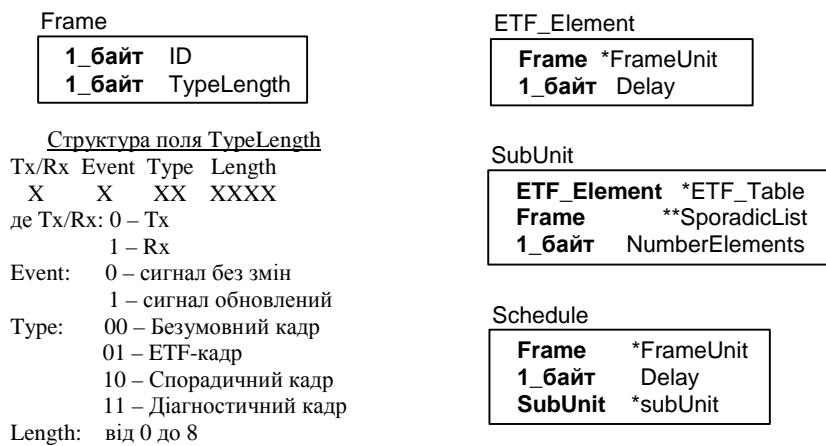


Рис. 6. Структури даних для реалізації робочого модуля LIN моделі даних

Таблиця LIN-розкладу формується у вигляді масиву елементів структури Schedule. Таблиця диспетчеризації являє собою фактично список циклічно повторюваних слотів, де для кожного слоту має бути визначений ідентифікатор кадру та часовий інтервал, а також, якщо це спорадичний кадр чи ETF-кадр, то ще й посилання на приєднаний список кадрів чи таблицю вирішення ETF-колізії. Відповідно, у структурі Schedule перший елемент – це посилання на потрібний кадр, що має виконуватися для вказаного слоту. Другий елемент визначає значення інтервалу часу для вказаного слоту. Це значення прийнято вказувати у кількостях міні-слотів, значення яких приймають переважно 5–10 мілісекунд. А от третє поле у випадку безумовних кадрів містить NULL значення, а у випадку спорадичних чи ETF-кадрів – адресу елемента структури SubUnit, в якому вже і вказано потрібний список безумовних кадрів для спорадичного кадру чи таблицю вирішення ETF-колізії

для ETF-кадру. Також елемент структури SubUnit повинен містити значення кількості елементів для зазначеного списку чи таблиці. Для таблиці вирішення ETF-колізій створена окрема структура даних ETF_Element. Вона, як і основна таблиця LIN-розкладу, повинна містити перелік безумовних кадрів та значення часових інтервалів у міні-слотах.

Програма роботи Master-вузла для організації обміну даними на LIN шині складається з п'яти основних процедур: 1) підпрограма переривання таймера/лічильника мікроконтролера. Тут виконується основна робота з диспетчеризації комунікації на LIN шині: виконується вибірка кадрів з таблиць, встановлюються часові інтервали слотів, здійснюються перемикання між основною таблицею LIN-розкладу та окремими таблицями вирішення ETF-колізій. Час циклічного спрацювання підпрограми дорівнює часовому інтервалу міні-слоту; 2) підпрограма переривання завершення LIN-передачі. У підпрограмі аналізуються такі ситуації: відправлено ідентифікатор, відповідно можемо надсилати чи приймати відповідь-повідомлення; відповідь прийнято; відповідь надіслано; 3) підпрограма переривання LIN-помилки. Тут виконується аналіз помилок, виявлених на LIN-шині. Нас тепер цікавить лише випадок виникнення помилки контрольної суми у разі отримання відповіді для ETF-кадру від декількох Slave-вузлів. У цьому випадку ми здійснюємо перемикання на таблицю вирішення ETF-колізій; 4) процедура підготовки даних для передачі. У FIFO-буфер LIN-контролера заносяться байти даних відповідно до значення ідентифікатора повідомлення; 5) процедура аналізу прийнятих даних. Відповідно до значення ідентифікатора повідомлення здійснюється читання з FIFO-буфера LIN-контролера прийнятих даних та виконується їхній подальший аналіз.

Модель даних робочого модуля Slave-вузла використовує той самий перелік кадрів, що й Master-вузол. На основі цього переліку для кожного вузла складають свій список підтримуваних кадрів. Програма роботи Slave-вузла для організації обміну даними на LIN-шині складається з трьох основних процедур: 1) підпрограми переривання завершення LIN-передачі. У підпрограмі аналізуються такі ситуації: прийнято ідентифікатор, відповідно, можемо надсилати чи приймати відповідь-повідомлення, для ETF-кадрів для відправлення сигналів перевіряється їхній стан (оновлений чи ні); відповідь прийнято; відповідь надіслано; 2) процедура підготовки даних для передачі. У FIFO-буфер LIN-контролера заносяться байти даних відповідно до значення ідентифікатора повідомлення; 3) процедура аналізу прийнятих даних. Відповідно до значення ідентифікатора повідомлення здійснюється читання з FIFO-буфера LIN-контролера прийнятих даних та виконується їхній подальший аналіз.

Експериментальна апробація

Для тестування розробленої програмної моделі даних для забезпечення роботи LIN-протоколу розроблено систему передачі даних (рис. 7). Розроблена система передачі даних складається з трьох LIN-модулів: Master-пристрій, який повністю відповідає за організацію передачі даних на LIN-шині, також керує увімкненнями/вимкненнями ламп на Slave-пристроях; два Slave-пристрої, які виконують прийом-передачу даних та керування виконавчими елементами згідно з вказівками Master-пристрою, також ці модулі можуть здійснювати керування виконавчими елементами один одного. Кожен модуль складається з мікроконтролера ATmega32M1 (ATMEL) та драйвера LIN-лінії MCP2025 (MICROCHIP), а також LCD-дисплею на базі контролера HD44780 для виведення інформації щодо обміну даними на LIN-шині.

У таблицю LIN-розкладу занесено типові задачі (табл. 1): передача байтів даних від Master-вузла до Slave-вузла (ID=1, ID=2); прийом Master-вузлом байтів даних від Slave-вузла (ID=3, ID=4); передача Master-вузлом байтів даних одразу двом Slave-вузлам (ID=11); передача вказівки окремому Slave-вузлу (ID=12, ID=13); передача вказівки одразу двом Slave-вузлам (ID=14); обмін даними між Slave-вузлами (ID=15, ID=16); передача ETF-кадру (ID=8); передача спорадичного кадру (ID=5). Для спорадичного кадру також вказується список безумовних кадрів (табл. 3), в якому часові інтервали не вказуються, оскільки кадри мають виконуватися в межах одного слоту, для якого часовий інтервал зазначений в таблиці LIN-розкладу. Для ETF-кадру додається на ви-

падок виникнення колізії на LIN-шині таблиця вирішення ETF-колізії (табл. 2). У таблицях комунікації у колонках з позначенням T зазначаються часові інтервали слоту в кількостях міні-слотів.

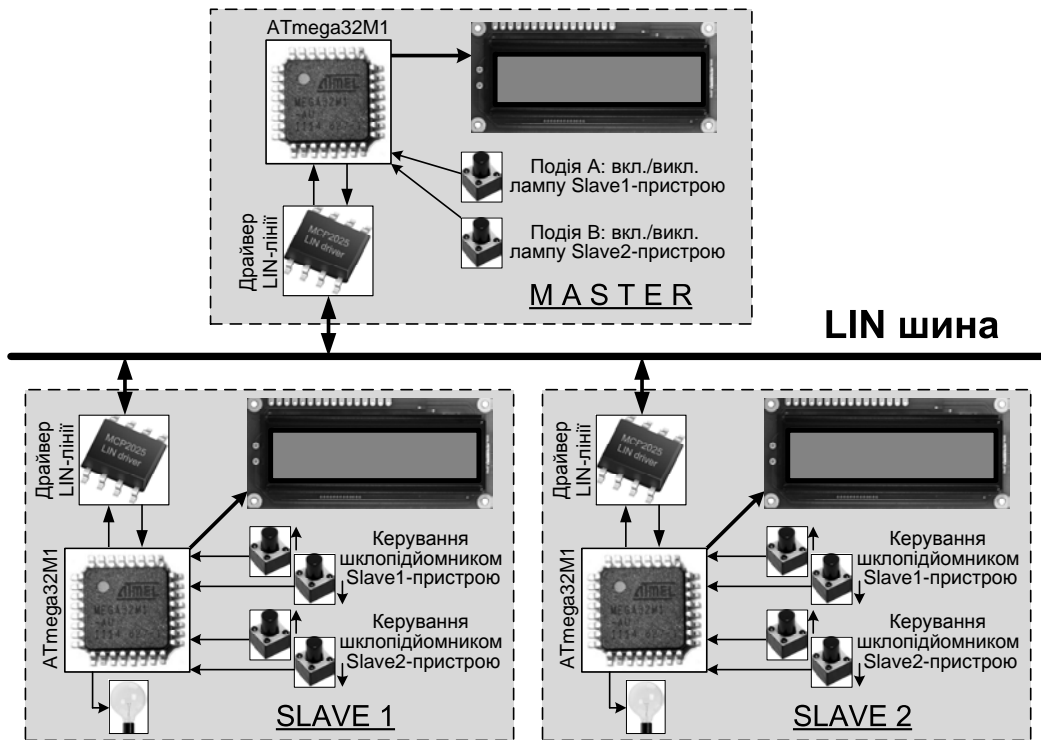


Рис. 7. Структурна схема системи передачі даних

Таблиця 1

LIN-розклад

| T | Slave-Задача Кадр | Кількість байтів, зміст | Slave-задача Master | Slave-задача 1 | Slave-задача 2 |
|---|-----------------------|----------------------------|------------------------|---------------------|---------------------|
| 3 | Безумовний кадр ID=1 | 4, поточні налаштув. | Відправник | Приймач | |
| 3 | Безумовний кадр ID=2 | 4, поточні налаштув. | Відправник | | Приймач |
| 5 | Спорадичний кадр ID=5 | 1, див. таблицю 3 | Відправник | ? | ? |
| | | | | Приймач | Приймач |
| 3 | Безумовний кадр ID=3 | 4, зовн+вн давач | Приймач | Відправник | |
| 3 | Безумовний кадр ID=4 | 4, зовн+вн давач | Приймач | | Відправник |
| 5 | ETF-кадр ID=8 | 3, NAD+вікно1+вікно2 | Приймач | ? | ? |
| | | | | Відправник | Відправник |
| 4 | Безумовний кадр ID=11 | 2, вікно1+вікно2 | Відправник | Приймач (вікно1) | Приймач (вікно2) |
| 2 | Безумовний кадр ID=12 | 0, вказівка | Відправник | Приймач | |
| 2 | Безумовний кадр ID=13 | 0, вказівка | Відправник | | Приймач |
| 2 | Безумовний кадр ID=14 | 0, вказівка | Відправник | Приймач | Приймач |
| 3 | Безумовний кадр ID=15 | 1, внутрішня інформ. | | Відправник | Приймач |
| 3 | Безумовний кадр ID=16 | 1, внутрішня інформ. | | Приймач | Відправник |

Таблиця 2

Таблиця вирішення ETF-колізії для ETF-кадру з ID=8

| T | Slave-Задача Кадр | Кількість байтів, зміст | Slave-задача Master | Slave-задача 1 | Slave-задача 2 |
|---|-----------------------|----------------------------|------------------------|----------------|----------------|
| 2 | Безумовний кадр ID=9 | 2, вікно1+вікно2 | Приймач | Відправник | |
| 2 | Безумовний кадр ID=10 | 2, вікно1+вікно2 | Приймач | | Відправник |

Список присланих безумовних кадрів для спорадичного кадру з ID=5

| Кадр \ Slave-Задача | Кількість байтів, зміст | Slave-задача Master | Slave-задача 1 | Slave-задача 2 |
|-----------------------|-------------------------|---------------------|----------------|----------------|
| Безумовний кадр ID=9 | 2, подія А | Відправник | Приймач | |
| Безумовний кадр ID=10 | 2, подія В | Відправник | | Приймач |

Апробація розробленої програмної моделі даних для забезпечення роботи LIN-протоколу та написаної алгоритмічної бібліотеки мовою Сі виконувалася безпосередньо на виготовленому нами фізичному макеті. Для нормальної для зору візуалізації даних вибрано часовий інтервал міні-слоту 1 секунда. Усі основні робочі режими відпрацьовані коректно.

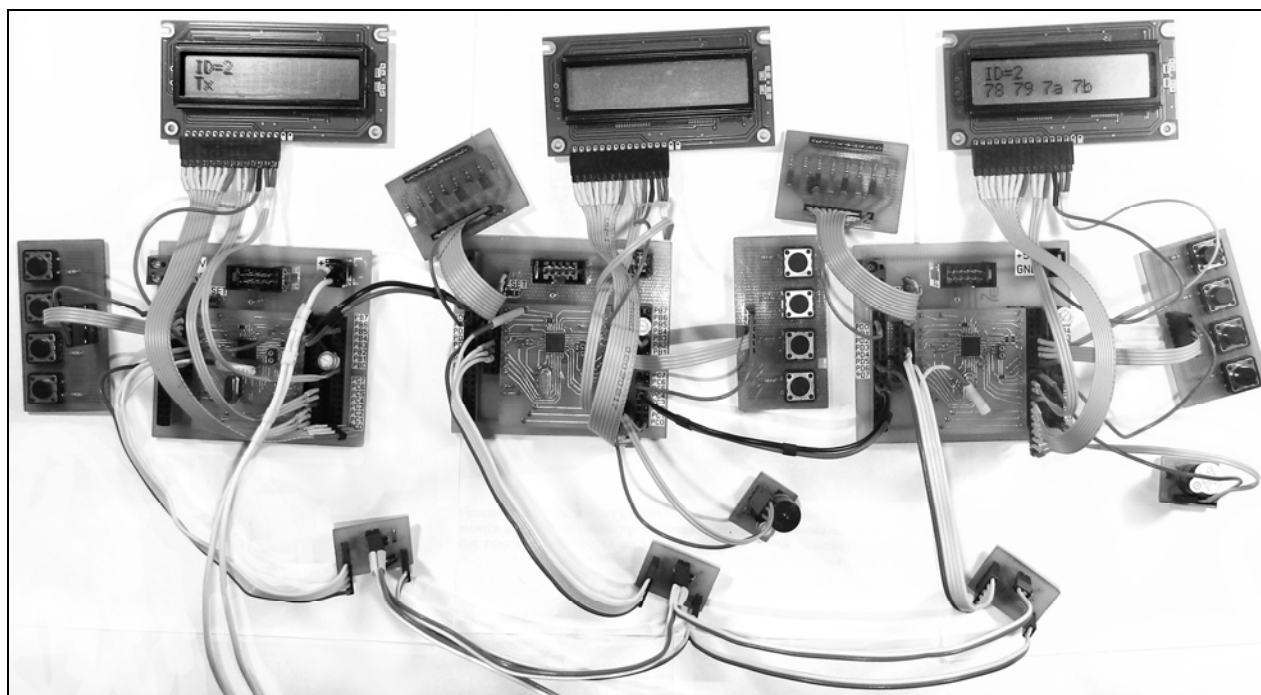


Рис. 8. Загальний вигляд макета для дослідження роботи LIN-мережі

Висновок

У результаті аналізу роботи автомобільного LIN-протоколу запропоновано програмну модель даних. Розроблена структура даних дає можливість повною мірою реалізувати усі функції диспетчеризації LIN-протоколу. Усі отримані результати експериментально апробовано на фізичному обладнанні.

1. Eugen Mayer. *Status quo and future of ECU communication in automobiles* // *AUTOMOTIVE, Vector Informatik*. – October, 2006. – P. 21–23. 2. *Technical Papers on Embedded Network Solutions*. – 5th Edition. – Vector Informatik GmbH, Stuttgart. – February 2014. – 419 p. 3. *LIN Specification Package, Revision 2.2A*. – LIN Consortium. – 2010. – 194 pages. 4. *ATmega32M1. Datasheets* – Atmel Corporation. – 2012. – 336 pages. 5. *MCP2025. LIN Transceiver with Voltage Regulator. Datasheets* – Microchip Technology Inc. – 2012. – 36 p. 6. *AVR286: LIN Firmware Base for LIN/UART Controller. Application Note* – Atmel Corporation. – 19 p. 7. *AVR308: Software LIN Slave. Application Note* – Atmel Corporation. – 12 p. 8. *AVR292: LIN “Break-in-Data” Feature of LIN/UART Controller. Application Note* – Atmel Corporation. – 8 p.