



УДК 004.057.4

ПОЛУДУПЛЕКСНАЯ СВЯЗЬ ПО ИНТЕРФЕЙСУ UART: АНАЛИЗ ДЕФЕКТОВ И СПОСОБЫ ИХ УСТРАНЕНИЯ

Шпынкова Виктория Дмитриевна,

Специалитет 5 курса направление «Системы управления летательными аппаратами»

Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)

Москва, Россия

shpynkova.viktoria@mail.ru

Деменев Дмитрий Андреевич,

Старший преподаватель кафедр ФН-7 (Электротехника и промышленная электроника),

ИУ-1 (Системы автоматического управления)

Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет),

Москва, Россия

demenev@bmstu.ru

Аннотация

В представленной работе исследуются практические вопросы построения полудуплексного канала связи по протоколу UART с применением стандарта RS-485. Проведен анализ ключевых проблем: появления ложных стартовых битов при смене режимов, коллизий драйверов на шине, утраты начальных байтов в ответе и несоответствия логических уровней. Для каждого случая предложены способы решения на аппаратном и программном уровнях. Также перечислены типичные ошибки при написании соответствующего кода.

Ключевые слова: UART, полудуплексный режим, RS-485, управление направлением, ложный стартовый бит, конфликт драйверов.

HALF-DUPLEX COMMUNICATION VIA THE UART INTERFACE: ANALYSIS OF DEFECTS AND WAYS TO ELIMINATE THEM

Shpynkova Victoria Dmitrievna,

5th-year specialist, majoring in "Aircraft Control Systems"

Bauman Moscow State Technical University (National Research University)

Moscow, Russia

shpynkova.viktoria@mail.ru

Demenev Dmitry Andreevich,

Senior Lecturer at the Departments FN-7 (Electrical Engineering and Industrial Electronics) and

IU-1 (Automatic Control Systems),

Bauman Moscow State Technical University (National Research University),
Moscow, Russia
demenev@bmstu.ru

ABSTRACT

The presented work examines practical issues related to building a half-duplex communication channel using the UART protocol with the RS-485 standard. An analysis of key problems has been carried out: the appearance of false start bits when changing modes, driver collisions on the bus, loss of initial bytes in the response, and mismatch of logical levels. For each case, solutions are proposed at the hardware and software levels. Typical errors when writing the corresponding code are also listed.

Keywords: UART, half-duplex mode, RS-485, direction control, false start bit, driver conflict.

Универсальный асинхронный приемопередатчик (UART) является одним из базовых и наиболее широко применяемых протоколов последовательного обмена в современной цифровой технике. По своей сути UART — это физический узел, который преобразует параллельные данные из системной шины в последовательную битовую последовательность для передачи по одной линии и выполняет обратную операцию при приеме [1].

Ключевой чертой UART выступает его асинхронная природа: в отличие от синхронных интерфейсов (например, SPI или I²C), для работы UART не требуется выделенная линия тактового сигнала. Вместо этого стороны предварительно договариваются о скорости передачи, а синхронизация пакетов обеспечивается использованием стартовых и стоповых битов [1].

При работе по стандарту RS-485, использующему дифференциальный (балансный) принцип передачи, UART функционирует в полудуплексном режиме. Такой метод обеспечивает высокую помехоустойчивость к синфазным искажениям: сигнал передается по паре проводников (А и В) в виде разности потенциалов, при которой наводки одинаково влияют на оба провода, не искажая полезный сигнал [2].

В полнодуплексном исполнении применяются отдельные линии TX и RX. UART можно представить как совокупность приемника (Receiver) и передатчика (Transmitter). В архитектуру UART входят: тактовый генератор (baud rate generator), регистры управления и статуса, буферы, а также сдвиговые регистры [3].

В полудуплексном режиме обмен данными идет по одной дифференциальной паре А/В с временным разделением. Устройства должны строго соблюдать очередность операций. В случае RS-485 полудуплекс реализуется через трансивер с управляющими выводами DE (Driver Enable) и RE (Receiver Enable). Часто их объединяют в один сигнал DIR: высокий уровень активирует передачу, низкий — прием [2].

Процесс переключения между этими режимами является основной причиной проблем, рассматриваемых в статье.

В момент перехода трансивера из режима передачи в режим приема на выводе R может возникнуть кратковременный провал напряжения. Из-за емкости линии напряжение на шине некоторое время остается в неопределенном состоянии, что заставляет приемник трансивера генерировать ложный импульс.

Для протокола UART такой скачок напряжения с высокого уровня на низкий распознается как стартовый бит. Стандартная передача начинается с установки нулевого уровня (стартовый бит), затем передаются данные (от младшего к старшему), и завершается процесс передачей стоп-бита (единичный уровень) [3]. Ложный импульс приводит к попаданию мусорных данных в приемный буфер.

Для борьбы с этим эффектом необходимо управлять направлением через GPIO, добавляя небольшую задержку после завершения передачи перед переключением в режим приема. После активации флага TC (Transmission Complete) следует сделать паузу в 1–2 битовых интервала, и только затем перевести DIR в режим приема.

Коллизия драйверов случается, когда два участника одновременно пытаются управлять состоянием шины, устанавливая разные логические уровни. Когда одно устройство «тянет» линию в высокий уровень, а другое — в низкий, на выходных каскадах трансиверов возникает короткое замыкание с превышением допустимых токов [2].

Чаще всего это происходит, когда устройство уже перешло в режим приема, а другое еще находится в режиме передачи (не начав ответ). В итоге первое устройство, не видя активности, может инициировать повторный цикл передачи.

Базовым способом предотвращения коллизий является введение пауз как перед началом передачи, так и после завершения текущей. Рекомендуемый алгоритм: отправка данных, ожидание флага TC, пауза на 3–4 битовых интервала, переход в режим приема и ожидание ответа по таймауту. Если ответ не получен, следует снова перейти в режим передачи для повторного запроса.

Аппаратной защитой служат токоограничивающие резисторы (100–200 Ом), устанавливаемые последовательно в линии А и В.

Дифференциальный метод RS-485 сам по себе устойчив к помехам. Однако при взаимодействии с логическими уровнями важно помнить, что порог чувствительности приемника составляет ± 200 мВ, и при слишком слабом сигнале распознавание может быть некорректным [2].

Потеря начальных байтов — это обратная сторона конфликта драйверов. Слишком длинные паузы перед переключением в режим приема приводят к тому, что ответ от собеседника приходит, когда трансивер еще не готов его принять. Эта проблема критична на высоких скоростях (от 115200 бод), где длительность бита исчисляется микросекундами.

Решением служит переход на аппаратное управление направлением, которое обеспечивает более точные временные интервалы, чем программные методы. Если используется программное управление, переключение DIR в режим приема должно происходить непосредственно в обработчике прерывания по флагу TC, чтобы минимизировать задержку.

При соединении компонентов с разными напряжениями питания (например, 3,3 В и 5 В) уровень 5 В может попасть на входы 3,3-вольтового контроллера, вызывая ток через защитные диоды. Это особенно опасно в схемах с транзисторной инверсией сигнала DIR, где возникают паразитные пути тока между цепями разного напряжения.

Для корректного согласования уровней применяют преобразователи (TXB0108, SN74LVC245) или NMOS-транзисторы с низким порогом (2N7002, BSS138). В промышленном оборудовании для этих целей применяется гальваническая развязка на базе изолированных трансиверов [4].

В процессе разработки полудуплексных интерфейсов часто встречаются следующие программные ошибки:

Использование флага TXE вместо TC. Флаг TXE (Transmit Buffer Empty) означает, что данные только перешли в буфер, но еще не ушли в линию. Переключение DIR по этому флагу «отрезает» последний бит данных.

Отсутствие критических секций при переключении. Прерывания по приему (например, RXNE) в момент смены режимов могут вызвать состояние «гонки» и сбой обработки.

Неверная настройка вывода DIR. Вывод должен работать в режиме push-pull; использование открытого коллектора или подтяжки к питанию недопустимо.

Отсутствие тайм-аутов. В полудуплексе важно обрабатывать отсутствие ответа; без тайм-аута система может зависнуть в ожидании при обрыве связи.

Игнорирование задержки распространения. На длинных линиях проявляются эффекты конечной скорости распространения сигнала. Максимальный кабель по RS-485 составляет 1200 м, а скорость может достигать 10 Мбит/с [5].

В данной статье проанализированы ключевые проблемы реализации полудуплексной связи UART через RS-485. Описаны физические причины возникновения дефектов и способы их устранения. Рассмотрены вопросы ложных стартовых битов, коллизий драйверов, потери первых байтов и несоответствия уровней. Выявлены типичные ошибки в коде.

Особый акцент сделан на управлении направлением передачи — критическом аспекте полудуплекса. Показано, что для точного переключения необходимо контролировать не только буфер, но и состояние сдвигового регистра. Аппаратное управление требует тщательного анализа временных диаграмм.

Систематизация этих проблем помогает оптимизировать разработку полудуплексных систем и минимизировать риск ошибок.

Список литературы:

1. UART (Universal Asynchronous Receiver/Transmitter) [Электронный ресурс] // Repka Pi Wiki. – Режим доступа: <https://repka-pi.ru/docs/94> (дата обращения: 04.07.2026).
2. Бень Е.А. RS-485 для чайников [Электронный ресурс] / Е.А. Бень. – 2003. – Режим доступа: https://relaylogic.ru/wp-content/uploads/2022/10/rs485_for_kettle.pdf (дата обращения: 01.08.2026).
3. Катукия Г.Т. Анализ и сравнение интерфейсов передачи данных на микроконтроллерах / Г.Т. Катукия, А.В. Дорофеев, Д.Э. Линтварев // Политехнический молодежный журнал. – 2020. – № 7. – DOI: 10.18698/2541-8009-2020-07-624.
4. Петров А.А. Протокол передачи данных для UART / А.А. Петров // Достижения науки и образования. – 2016. – № 1 (2). – С. 9-10.
5. Занина В.А. Программная реализация интерфейса UART / В.А. Занина, Е.А. Иванова // Современные научные исследования и разработки. – 2018. – Т. 2, № 11 (28). : С. 269–270.

References:

1. UART (Universal Asynchronous Receiver/Transmitter) [Electronic resource] // Repka Pi Wiki. – Access mode: <https://repka-pi.ru/docs/94> (accessed: 04.07.2026).
2. Ben E.A. RS-485 for Dummies [Electronic resource] / E.A. Ben. – 2003. – Access mode: https://relaylogic.ru/wp-content/uploads/2022/10/rs485_for_kettle.pdf (date of access: 08/01/2026).

3. Katukiya G.T. Analysis and comparison of data transmission interfaces on microcontrollers / G.T. Katukiya, A.V. Dorofeev, D.E. Lintvarev // Polytechnic Youth Journal. – 2020. – № 7. – DOI: 10.18698/2541-8009-2020-07-624.
4. Petrov A.A. Data transmission protocol for UART / A.A. Petrov // Achievements of Science and Education. – 2016. – No. 1 (2). – Pp. 9-10. – EDN: TDWWEB.
5. Zanina V.A. Software implementation of the UART interface / V.A. Zanina, E.A. Ivanova // Modern scientific research and development. – 2018. – Vol. 2, No. 11 (28). – Pp. 269-270. – EDN: YUKVJJ. – URL: <https://elibrary.ru/item.asp?id=36810343> (accessed: 20.08.2026).