

ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ
И ТЕЛЕКОММУНИКАЦИИ

INFORMATION
TECHNOLOGIES
AND TELECOMMUNICATIONS

УДК 004.415

DOI: 10.17212/2782-2001-2025-2-81-92

Проектирование программной реализации медиаконвертера для интерфейса передачи данных SBNI*

И.М. РЕБУС^{1,a}, А.Б. КОЛКЕР^{2,b}, Г.А. ФРАНЦУЗОВА^{1,c}

¹ 630087, г. Новосибирск, ул. Немировича-Данченко, 136, Новосибирский государственный технический университет

² 630099, г. Новосибирск, ул. Советская 30, Сибирский региональный научно-исследовательский гидрометеорологический институт

^a ilya.rebus42@gmail.com ^b alexk@sibnigmi.ru, ^c frants@ac.cs.nstu.ru

В настоящей статье рассматривается задача модификации существующего интерфейса связи для высокоскоростной передачи данных по медной витой паре в опасных и труднодоступных условиях подземных горнодобывающих предприятий в части разработки программной реализации медиаконвертера.

Для обеспечения связи и мониторинга оборудования в труднодоступных условиях горных выработок в конце 90-х годов XX в. был разработан специализированный сетевой интерфейс, названный «SBNI» (Siberian Board Network Interface – сибирский сетевой интерфейс). Он позволял передавать данные по одной витой паре на большие расстояния (до 9 км), при этом обеспечивая более высокие скорости (до 2 Мбит/с) по сравнению с широко известным стандартом физического уровня RS-485/EIA-485. Конфигурация интерфейса предполагает использование специального медиаконвертера, т.е. устройства, преобразующего кадры Ethernet в кадры SBNI. Существующий на данный момент вариант медиаконвертера реализован на устаревших, дорогостоящих и труднодоступных на рынке микроконтроллерах и программируемых логических интегральных схемах (ПЛИС). Развитие микроконтроллерной техники позволяет заменить аппаратный медиаконвертер на его программную реализацию без нанесения ущерба функционалу и совместимости с предыдущей реализацией интерфейса.

В статье представлены программные алгоритмы, спроектированные для работы с сопроцессорными ядрами реального времени для решения целевой задачи. Результаты, представленные в настоящей работе, позволяют использовать незадействованные ранее аппаратные ресурсы центрального процессора базовых станций для переноса на них функций медиаконвертера и тем самым удешевить и оптимизировать производство базовых станций без снижения их надежности и функциональности.

Ключевые слова: интерфейс передачи данных, медиаконвертер, витая пара, программное обеспечение, удаленные вычислительные ядра, вычисления в реальном времени, алгоритмы микропрограмм, язык ассемблера, SBNI, PRU-ICSS, Texas Instruments Sitara AM3358

* Статья получена 20 февраля 2025 г.

ВВЕДЕНИЕ

Базовые станции, обеспечивающие связь в подземной выработке, соединены между собой и с поверхностью с помощью оптоволоконных кабелей. На магистральных линиях связи их использование целесообразно ввиду большого количества трафика, однако на периферийных участках, где требования к пропускной способности канала не такие строгие, оптоволокно имеет больше недостатков, чем преимуществ. В частности, оно уязвимо к механическим повреждениям, поскольку обладает минимально допустимым радиусом сгиба и требует специального оборудования для ремонта (сращивания). Таким образом, эксплуатация оптоволоконных кабелей в условиях шахты целесообразна только в случае высоких требований к пропускной способности канала, в остальных случаях предпочтительна обычная витая пара [1–3]. Помимо шахтного оборудования существует множество сфер, где использование оптоволоконных кабелей для организации каналов связи нецелесообразно, а использование радиоканала не допускается, например, для аэродромных сетей телеметрии и систем метеорологических измерений.

Протокол передачи данных Ethernet ограничен длиной сегмента 100 метров, при превышении этого расстояния необходимо использовать специальные устройства-повторители. При этом длина подземных выработок может достигать многих километров, что делает Ethernet дорогостоящим и неудобным решением [4, 5].

Помимо Ethernet существуют и другие решения для обеспечения проводной связи на большие расстояния в промышленных условиях. В первую очередь это стандарт передачи данных RS-485, максимальная длина сегмента которого на скорости 62.5 кбит/с ограничена 1200 м [6], что зачастую оказывается недостаточным для удовлетворения требований к линиям связи SBGPS (Siberian Board Geopositioning System – сибирская система геопозиционирования) даже на периферийных участках. Существует также семейство технологий DSL (Digital Subscriber Line – цифровая абонентская линия), таких как ADSL (Asymmetric DSL – асимметричная DSL), VDSL (Very High Speed DSL – сверхскоростная DSL), HDSL (High-Speed DSL – высокоскоростная DSL) [7]. Все они позволяют добиваться относительно высоких скоростей на схожих расстояниях, однако требуют наличия специальных модемов и предъявляют особые требования к характеристикам витой пары. Помимо прочего, это оборудование класса SOHO (Small Office/Home Office – малый/домашний офис) – класс домашних и офисных сетей, которые не удовлетворяют условиям по искробезопасности и надежности.

Сетевой интерфейс SBNI позволяет решить эту проблему и обеспечивает скорость до 2 Мбит/с на расстоянии до 1200 м либо 62.5 кбит/с на расстоянии 8 км. В таком оборудовании применяется специальное устройство-медиаконвертер, который преобразует кадры Ethernet в кадры SBNI. Существующая аппаратная реализация использует устаревшие электронные компоненты, что затрудняет ее производство и сопровождение.

В настоящей статье описан способ программной реализации медиаконвертера для протокола SBNI, базирующийся на распределении задач между вычислительными сопроцессорными ядрами реального времени.

Медиаконвертер в текущей версии состоит из сигнального процессора, реализованного на ПЛИС (программируемая логическая интегральная схема), и микроконтроллера, реализующего канальный уровень SBNI и преобразую-

щего кадры Ethernet в SBNI и наоборот. Он встраивается в корпус базовой станции или технологического контроллера и подключается к одному из разъемов Ethernet на материнской плате. Второй вывод медиаконвертера соединяется с искрозащитным барьером, на котором есть гальваническая развязка и компаратор, являющиеся частью физического уровня SBNI, что обусловлено требованиями взрыво- и искробезопасности.

Вычислительные мощности и архитектура современных микропроцессоров позволяют реализовать программную реализацию интерфейса SBNI напрямую на процессоре базовой станции, что предоставляет возможность заместить устаревшие схемотехнические решения медиаконвертера. Актуальность настоящей работы обусловлена моральным устареванием компонентной базы существующего решения. Новизна работы заключается в использовании сопроцессорных ядер реального времени процессора базовой станции в качестве платформы для программной реализации алгоритмов.

1. ИНТЕРФЕЙС ПЕРЕДАЧИ ДАННЫХ SBNI

1.1. РАЗВИТИЕ ТЕХНОЛОГИЙ SBNI

Интерфейс SBNI был впервые разработан и поставлен на производство в 1996 г. Тогда он представлял собой широкополосный модем для выделенных физических линий SBNI12-02, выполненный в виде платы расширения персонального компьютера для ISA-шины. На физическом уровне он использовал модуляцию AMI, работал в диапазоне от 62.5 до 2000 кбит/с и позволял передавать данные на расстояние до 1 км с максимальной скоростью, и на расстояние 3600 м на скорости 250 кбит/с.

Применяемая в настоящее время версия SBNI 12-10 появилась в 2009 г. и использовала актуальные на тот момент технологии. В ней был применен медиаконвертер, схемотехнические решения которого в настоящее время морально устарели.

Соотношения длины / скорости SBNI12-10

Distance / speed proportions for SBNI12-10

Скорость передачи (кбит/с)	Параметр	Кабель ТПП50-0.5	Кабель ТПП50-0.4
2000	Длина линии (м)	1200	1000
	R (Ом) / α (дБ)	216 / 29	278 / 30
1000	Длина линии (м)	2000	1400
	R (Ом) / α (дБ)	360 / 34	389 / 31
500	Длина линии (м)	3200	2000
	R (Ом) / α (дБ)	576 / 40	556 / 32
250	Длина линии (м)	4600	3800
	R (Ом) / α (дБ)	828 / 42	1056 / 46
125	Длина линии (м)	7600	4800
	R (Ом) / α (дБ)	1368 / 57	1334 / 51
62.5	Длина линии (м)	8600	5000
	R (Ом) / α (дБ)	1548 / 57	1390 / 47

На физическом уровне используется модуляция FM0, являющаяся разновидностью манчестерского кодирования (рис. 1). Сигнал в линии асинхронный, дифференциальный, с амплитудой от -5 В до $+5$ В. Такой способ передачи данных обеспечивает очень высокую устойчивость к влиянию синфазной помехи.



Рис. 1. Диаграмма, иллюстрирующая принцип модуляции физического уровня SBNI

Fig. 1. A diagram illustrating modulation principles on SBNI physical layer

Здесь период передачи каждого символа имеет постоянную длительность, причем изменение уровня сигнала происходит на каждой границе бита, а при передаче сигнала логического нуля также и в середине бита.

1.2. СИСТЕМА PRU-ICSS

В базовой станции используется система-на-кристалле Sitara AM3358 от Texas Instruments. Отличительной особенностью данной SoC (System-on-Crystal – система-на-кристалле) является наличие двух сопроцессорных ядер Processing Real-Time Units или PRU, независимых от ядра ARM (Advanced RISC Machine, продвинутая RISC-машина), которые можно использовать для сравнительно ресурсоемких вычислений в реальном времени. Эти ядра являются частью так называемой системы PRU–ICSS (Programmable Real-Time Unit Subsystem and Industrial Communication Subsystem – программируемая подсистема вычислений в реальном времени и подсистема промышленной связи) (рис. 2) [8, 9]. Они удовлетворяют условиям поставленной задачи, поскольку снимают вычислительную нагрузку с основного ядра и позволяют реализовывать алгоритмы, требующие гарантии времени выполнения. Следовательно, система PRU–ICSS обеспечивает подходящий инструментарий для решения поставленной задачи.

Данная иллюстрация описывает внутреннее устройство системы PRU–ICSS. В частности, основными блоками являются два ядра PRU (PRU0 и PRU1), подключенные к 32-битной шине, на которой также располагаются индивидуальные разделы памяти для каждого ядра (RAM0 и RAM1) объемом в 8 кбайт и общая разделяемая память объемом 12 кбайт, конфигурационные регистры и периферийные интерфейсы, такие как UART и Ethernet.

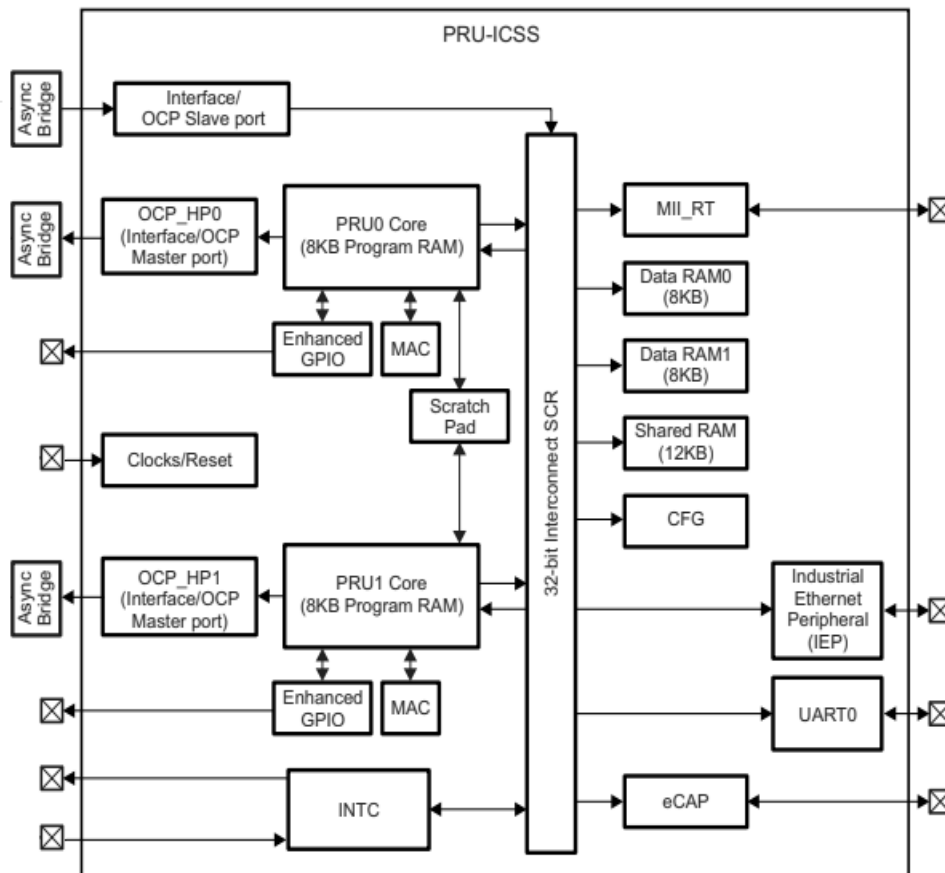


Рис. 2. Функциональная блок-схема PRU – ICSS

Fig. 2. The functional block diagram of PRU – ICSS

Основной функцией PRU – ICSS является разгрузка основного ядра ARM путем выполнения задач, строго фиксированных по времени исполнения. Такими задачами могут являться реализация пользовательских протоколов передачи данных или цифровая обработка сигналов в реальном времени. Контроль над временем выполнения ядер PRU обуславливается намеренным отсутствием конвейеризации, что гарантирует выполнение только одной инструкции за один такт процессора и позволяет гарантировать время выполнения той или иной функции. Помимо этого, контроллер прерываний PRU – ICSS не генерирует прерывания в традиционном понимании, то есть он не способен прерывать выполнение программы. Вместо этого он генерирует события, которые необходимо обрабатывать, непрерывно считывая соответствующие биты регистра внешних событий.

Такой подход ориентирован исключительно на осуществление полного контроля над временем выполнения кода, что делает такую архитектуру подходящим инструментом для реализации пользовательских протоколов передачи данных, функционирующих в режиме реального времени, на чем намеренно делает акцент разработчик процессора [10, 11].

2. РАЗРАБОТКА АЛГОРИТМОВ ОЖИДАНИЯ СОБЫТИЙ, ПРИЁМА И ПЕРЕДАЧИ ДАННЫХ

Точно гарантировать длину инструкций алгоритмических блоков кода возможно только при создании их на ассемблере, поскольку компиляторы языков высокого уровня могут по-разному представлять одинаковые директивы в зависимости от настроек оптимизации [12]. Таким образом, алгоритм может быть представлен следующими подзадачами.

2.1. АЛГОРИТМ ОЖИДАНИЯ ВНЕШНИХ СОБЫТИЙ

Алгоритм реакции PRU на внешние события представлен на рис. 3. Для отслеживания внешних событий (аналог механизма прерываний) существует регистр, биты которого характеризуют статус определенных событий [13].

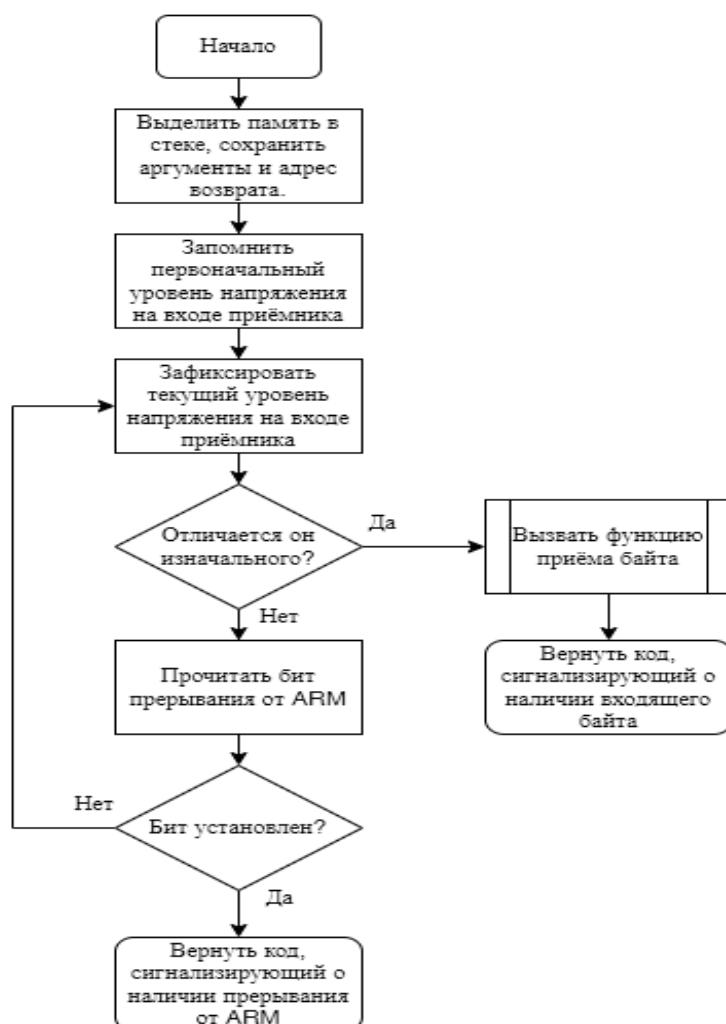


Рис. 3. Блок-схема алгоритма ожидания внешних событий

Fig. 3. A block diagram of the algorithm for waiting for external events

Данный алгоритм выполняет всего две задачи: отслеживание изменения уровня напряжения на входе приемника и наличие внешнего события, пришедшего от ядра ARM, что происходит на протяжении всего времени работы программы. Тело данного цикла занимает всего 3 машинные инструкции, что позволяет ему реагировать на изменения максимально быстро, с максимальным запаздыванием всего в 2 такта, что при частоте 200 МГц равняется 10 наносекундам.

2.2. АЛГОРИТМ ПЕРЕДАЧИ ДАННЫХ

Представленная на рис. 4 блок-схема иллюстрирует алгоритм, по которому работает функция передачи данных.

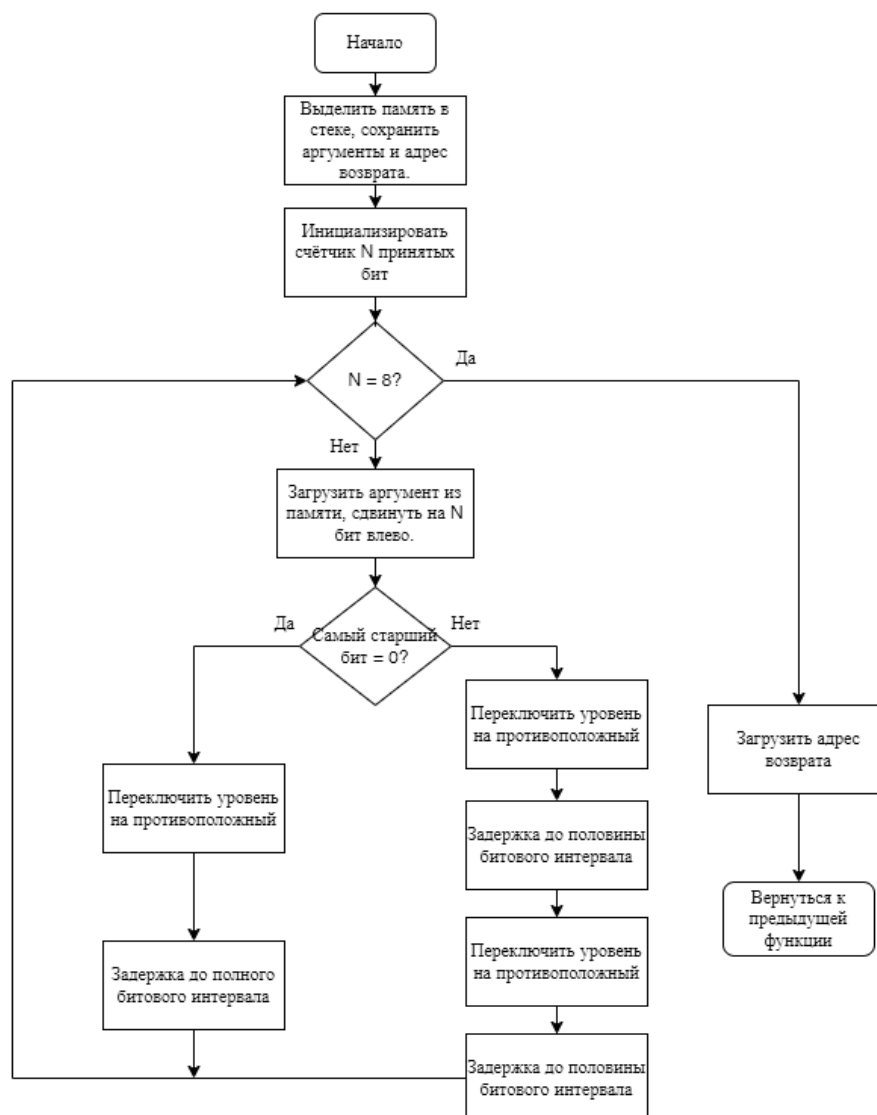


Рис. 4. Блок-схема алгоритма передачи данных

Fig. 4. A block diagram of the data transmission algorithm

Здесь реализован алгоритм, упомянутый при описании кодирования FMO и представленный на рис. 1. Алгоритм принимает в качестве аргумента байт данных и отправляет его в линию связи, начиная со старшего бита.

2.2. АЛГОРИТМ ПРИЕМА ДАННЫХ

Следующим этапом работы стала разработка алгоритма приема данных (рис. 5). Его реализация заключается в последовательном анализе уровня напряжения в начале и середине битового интервала в целях идентификации принятого бита.

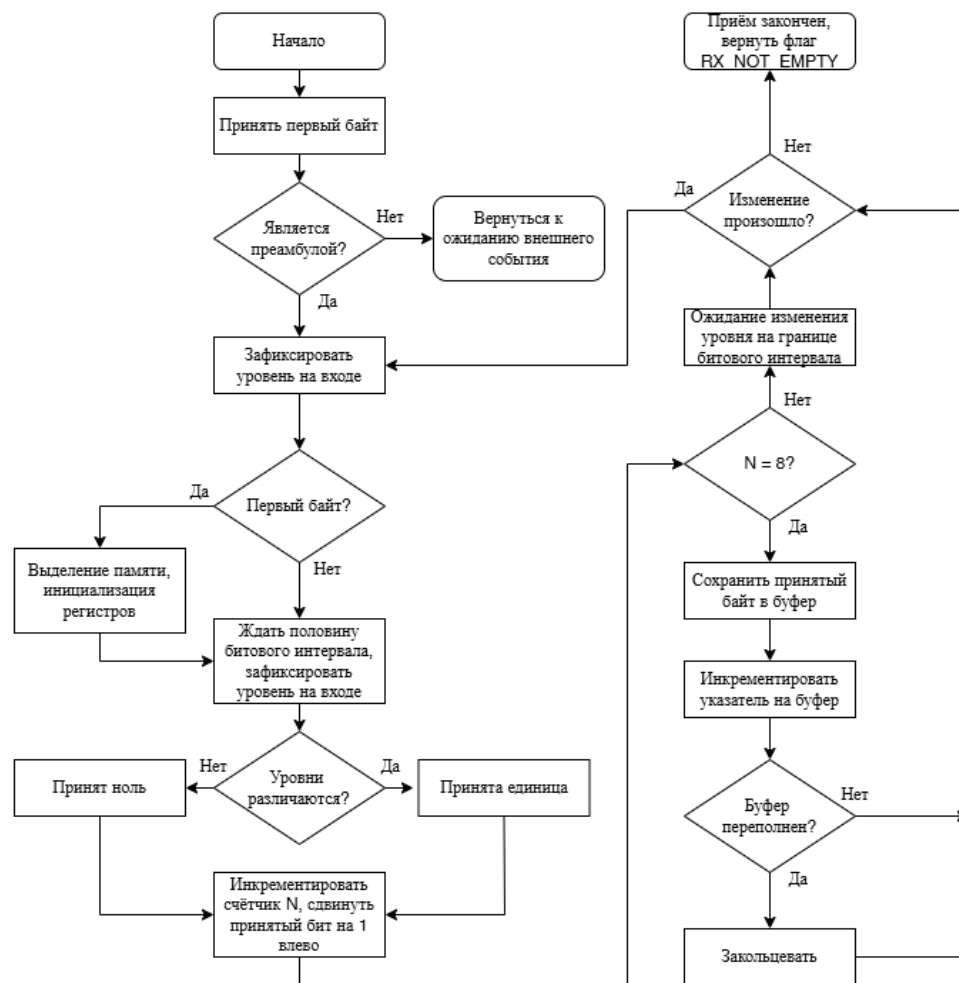


Рис. 5. Блок-схема алгоритма приема данных

Fig. 5. A block diagram of the data reception algorithm

Ключевым требованием к алгоритму является минимальная длина интервала между реакцией алгоритма ожидания внешних событий и его регистрацией в алгоритме приема. Именно поэтому инициализация регистров происходит после фиксации текущего уровня первого входящего бита.

Алгоритм FM0 требует выполнения процедуры «балансировки линии». Поскольку при передаче нуля уровень напряжения меняется только один раз – на границе битового интервала, то при передаче четного количества нулей уровень напряжения в линии связи вернется к исходному значению, а при передаче нечетного – не вернется. Таким образом, перед началом любого пакета данных должна идти преамбула в виде серии нулей, позволяющая алгоритму автоматически сбалансировать линию.

Реализованный алгоритм приема способен принимать данные, передающиеся со скоростью 4166 Мбит/с (с длительностью одного битового интервала 240 нс) и при доработке искрозащитного барьера позволяет повысить максимально возможную теоретическую пропускную способность SBNI.

ЗАКЛЮЧЕНИЕ

В работе представлены программные алгоритмы приема, передачи данных и ожидания событий на сопроцессорных ядрах реального времени, позволяющие использовать их для замены медиаконвертера SBNI. Показано, что при использовании разработанного подхода возможно отказаться от применения ПЛИС для обработки сигналов, реализовав все необходимые процедуры на программном уровне непосредственно на вычислительных мощностях CPU. Представленное решение также потенциально позволяет повысить скорость передачи данных SBNI. При этом решение позволяет сохранить функционал базовой станции, несмотря на изменение аппаратной конфигурации.

Планируемое развитие работы заключается во встраивании разработанных алгоритмов в операционную систему базовой станции и наладке взаимодействия удаленных сопроцессорных ядер реального времени PRU-ICSS с основным ядром ARM. В частности, планируется разработка драйвера сетевого интерфейса для операционной системы Linux.

СПИСОК ЛИТЕРАТУРЫ

1. Fundamentals of optical fiber communications / ed. by M.K. Barnoski. – 2nd ed. – Academic Press, 1981. – 351 p. – ISBN 0-12-079151-X.
2. Ребус И.М. Обсуждение возможности использования оптического волокна для передачи данных в сложных промышленных условиях // Наука. Технологии. Инновации: сборник научных трудов, г. Новосибирск, 2–6 декабря 2024 г.: в 8 ч. – Новосибирск: НГТУ, 2025. – Ч. 1. – С. 12–16. – ISBN 978-5-7782-5343-8.
3. Бабкин О.Э., Ильина В.В. Оптико-волоконные кабели – проблемы производства и перспективы развития // Актуальные вопросы развития индустрии кино и телевидения в современной России: материалы I Национальной научно-практической конференции. – СПб., 2018. – С. 33–35. – ISBN 978-5-94760-283-8.
4. IEEE Std 802.3ab-1999. IEEE Standard for Information Technology – Telecommunications and information exchange between systems – Local and Metropolitan Area Networks – Part 3. – IEEE, 1999. – 144 p. – DOI: 10.1109/IEEESTD.1999.90568.
5. Олифер В.Г., Олифер Н.А. Компьютерные сети: принципы, технологии, протоколы: учебник для вузов. – 3-е изд. – СПб.: Питер, 2006. – 960 с. – ISBN 5-469-00504-6.
6. Яшкардин В.Л. RS-485 – стандарт передачи данных по последовательному симметричному каналу // SoftElectro: сайт. – 2009. – URL: <http://www.softelectro.ru/rs485.html> (дата обращения: 20.05.2025).

7. Transmission and Multiplexing (TM); Access transmission systems on metallic access cables; Asymmetric Digital Subscriber Line (ADSL) – Coexistence of ADSL and ISDN-BA on the same pair / European Telecommunications Standards Institute. – ETSI, 1998. – 70 p.
8. AM335x Sitara™ Processors Technical Reference Manual. – Texas Instruments, 2014. – URL: <https://www.ti.com/lit/ug/spruh73q/spruh73q.pdf> (accessed: 20.05.2025).
9. AM335x Sitara™ Processors datasheet (Rev. L). – Texas Instruments, 2018. – URL: <https://www.ti.com/lit/ds/symlink/am3358.pdf> (accessed: 20.05.2025).
10. PRU Assembly Language Tools v2.3 User's Guide, rev. C. – Texas Instruments, 2018. – URL: https://www.ti.com/lit/ug/spruhv6c/spruhv6c.pdf?ts=1742287577782&ref_url=https%253A%252F%252Fwww.google.com%252F (accessed: 20.05.2025).
11. AM335x PRU-ICSS Reference Guide. – Texas Instruments, 2012. – 289 p. – URL: <https://mythopoeic.org/BBB-PRU/am335xPruReferenceGuide.pdf> (accessed: 20.05.2025).
12. Processor SDK Linux Software Developer's Guide for AM335X, rev. C. – Texas Instruments, 2021. – URL: <https://software-dl.ti.com/processor-sdk-linux/esd/docs/latest/devices/AM335X/linux/index.html> (accessed: 20.05.2025).
13. BeagleBone Black System Reference Manual. – BeagleBoard.org Foundation, 2025. – URL: <https://docs.beagleboard.org/beaglebone-black.pdf> (accessed: 20.05.2025).
14. Yoder M.A., Kridner J. BeagleBone cookbook. – 1st ed. – O'Reily Media, 2015. – 346 p. – ISBN 978-1-4919-0538-8.
15. Lockwood G.K. Programming the BeagleBone PRU-ICSS. – URL: <https://www.glennklockwood.com/embedded/beaglebone-pru.html> (accessed: 20.05.2025).
16. Molloy D. Exploring Beaglebone: tools and techniques for building with embedded Linux. – 2nd ed. – John Wiley & Sons, 2019. – 800 p. – ISBN 978-1-118-93512-5.

Ребус Илья Михайлович, аспирант кафедры автоматизации Новосибирского государственного технического университета. Основное направление научных исследований – подземные телекоммуникационные технологии и автоматизация. E-mail: ilya.rebus42@gmail.com

Колкер Алексей Борисович, кандидат технических наук, ведущий научный сотрудник, научный руководитель Сибирского регионального научно-исследовательского гидрометеорологического института, доцент кафедры автоматизации Новосибирского государственного технического университета. Основное направление научных исследований – системы искусственного интеллекта, вычислительные технологии, микропроцессорная техника. Имеет более 100 печатных работ. E-mail: alexx@sibnigmi.ru

Французова Галина Александровна, доктор технических наук, профессор кафедры автоматизации Новосибирского государственного технического университета. Основное направление научных исследований – анализ и синтез нелинейных систем автоматического регулирования. Имеет более 200 печатных работ. E-mail: frants@ac.cs.nstu.ru

Rebus Ilya M., postgraduate student at the Department of Automation, Novosibirsk State Technical University. The main field of his research is underground telecommunication technologies and automatization. E-mail: ilya.rebus42@gmail.com

Kolker Alexey B., PhD (Eng), lead researcher of Siberian Regional Hydrometeorological Research Institute, associate professor at the Department of Automation, Novosibirsk State Technical University. The main field of his scientific research includes artificial intelligence systems, computer technologies, and microcontroller devices. He has over 100 published works. Email: alexx@sibnigmi.ru

Frantsuzova Galina A., D.Sc. (Eng.), professor, Department of Automation, Novosibirsk State Technical University. The main field of her scientific research is analysis and synthesis of nonlinear automatic control systems. She has over 200 published works. E-mail: frants@ac.cs.nstu.ru

Designing a software implementation of a media converter for the SBNI data transfer interface*I.M. REBUS^{1,a}, A.B. KOLKER^{2,b}, G.A. FRANTSUZOVA^{1,c}¹ Novosibirsk State Technical University, 136, Nemirovitcha-Danchenko Street, Novosibirsk, 630087, Russian Federation² Siberian Research Hydrometeorological Institute, 30, Sovetskaya Street, Novosibirsk 630099, Russian Federation^a ilya.rebus42@gmail.com ^b alexk@sibnigmi.ru ^c frants@ac.cs.nstu.ru**Abstract**

This article covers the subject of making modifications to an existing high-speed data transmission interface meant for using in a dangerous environment of underground mining facilities, namely, the development of a software implementation of a media converter in order to replace it.

At the end of the 90-s of the last century, a specialized network interface was developed to provide communication and environmental monitoring in hazardous environments of mining facilities. It was called Siberian Board Network Interface (SBNI) and allowed long-range (up to 9 km) data transmission via single twisted pair, offering higher speeds (up to 2 Mbit/s) than interfaces based on a widely known physical standard RS-485/EIA-485. The current configuration of the interface suggests usage of a media converter which is a device that transforms Ethernet data packets into SBNI datagrams. This media converter includes electronic components that are now obsolete, expensive and unavailable on the local market. Continuous development of embedded electronics now allows replacing the media converter by a software implementation.

This article contains algorithms developed for work with remote processing units in order to solve the problem. The result of the work demonstrates a possibility of enabling previously unused remote processing units to implement the logic of a media converter on them, and simplify and optimize the production process without sacrificing reliability or functionality.

Keywords: network interface, media converter, twisted pair, software, remote processing units, program algorithm, assembler language, SBNI, PRU-ICSS, Texas Instruments Sitara AM3358

REFERENCES

1. Barnoski M.K., ed. *Fundamentals of optical fiber communications*. 2nd ed. Academic Press, 1981. 351 p. ISBN 0-12-079151-X.
2. Rebus I.M. [Discussion of the possibility of using optical fiber for data transmission in difficult industrial conditions]. *Nauka. Tekhnologii. Innovatsii: sbornik nauchnykh trudov* [Science. Technologies. Innovations: collection of scientific papers], Novosibirsk, 2–6 December 2024. Novosibirsk, 2025, pt. 1, pp. 12–16. ISBN 978-5-7782-5343-8. (In Russian).
3. Babkin O.E., Il'ina V.V. [Fiber-optic cables – the problems of production and development prospects]. *Aktual'nye voprosy razvitiya industrii kino i televideniya v sovremennoi Rossii* [Current issues in the development of the film and television industry in modern Russia]. Proceedings of the I National Scientific and Practical Conference]. St. Petersburg, 2018, pp. 33–35. ISBN 978-5-94760-283-8. (In Russian).
4. IEEE Std 802.3ab-1999. *IEEE Standard for Information Technology – Telecommunications and information exchange between systems – Local and Metropolitan Area Networks – Part 3*. IEEE, 1999. 144 p. DOI: 10.1109/IEEESTD.1999.90568.

* Received 20 February 2025.

5. Olifer V.G., Olifer N.A. *Komp'yuternye seti: printsipy, tekhnologii, protokoly* [Computer networks. Principles, technologies, protocols]. 3rd ed. St. Petersburg, Piter Publ. 960 p. ISBN 5-469-00504-6.
6. Yashkardin V.L. *RS-485 – standart peredachi dannykh po posledovatel'nomu simmetrichnomu kanalu* [Recommended Standard 485. Electrical Characteristics of Generators and Receivers for Use in Balanced Multipoint Systems]. 2009. (In Russian). Available at: <http://www.softelectro.ru/rs485.html> (accessed 20.05.2025).
7. *Transmission and Multiplexing (TM); Access transmission systems on metallic access cables; Asymmetric Digital Subscriber Line (ADSL) – Coexistence of ADSL and ISDN-BA on the same pair*. European Telecommunications Standards Institute. ETSI, 1998. 70 p.
8. *AM335x Sitara™ Processors Technical Reference Manual*. Texas Instruments, 2014. Available at: <https://www.ti.com/lit/ug/spruh73q/spruh73q.pdf> (accessed 20.05.2025).
9. *AM335x Sitara™ Processors datasheet (Rev. L)*. Texas Instruments, 2018. Available at: <https://www.ti.com/lit/ds/symlink/am3358.pdf> (accessed 20.05.2025).
10. *PRU Assembly Language Tools v2.3 User's Guide*, rev. C. Texas Instruments, 2018. Available at: https://www.ti.com/lit/ug/spruhv6c/spruhv6c.pdf?ts=1742287577782&ref_url=https%253A%252F%252Fwww.google.com%252F
11. *AM335x PRU-ICSS Reference Guide*. Texas Instruments, 2012. 289 p. Available at: <https://mythopoeic.org/BBB-PRU/am335xPruReferenceGuide.pdf> (accessed 20.05.2025).
12. *Processor SDK Linux Software Developer's Guide for AM335X, rev. C, part 3.5.1*. Texas Instruments, 2021. Available at: <https://software-dl.ti.com/processor-sdksdklinux/esd/docs/latest/devices/AM335X/linux/index.html> (accessed 20.05.2025).
13. *BeagleBone Black System Reference Manual*. BeagleBoard.org Foundation, 2025. Available at: <https://docs.beagleboard.org/beaglebone-black.pdf> (accessed 20.05.2025).
14. Yoder M.A., Kridner J. *BeagleBone cookbook*. 1st ed. O'Reilly Media, 2015. 346 p. ISBN 978-1-4919-0538-8.
15. Lockwood G.K. *Programming the BeagleBone PRU-ICSS*. Available at: <https://www.glenn-klockwood.com/embedded/beaglebone-pru.html> (accessed 20.05.2025).
16. Molloy D. *Exploring Beaglebone: tools and techniques for building with embedded Linux*. 2nd ed. John Wiley & Sons, 2019. 800 p. ISBN 978-1-118-93512-5.

Для цитирования:

Ребус И.М., Колкер А.Б., Французова Г.А. Проектирование программной реализации медиаконвертера для интерфейса передачи данных SBNI // Системы анализа и обработки данных. – 2025. – № 2 (98). – С. 81–92. – DOI: 10.17212/2782-2001-2025-2-81-92.

For citation:

Rebus I.M., Kolker A.B., Frantsuzova G.A. Proektirovanie programmnoi realizatsii media-konvertera dlya interfeisa peredachi dannykh SBNI [Designing a software implementation of a media converter for the SBNI data transfer interfacs]. *Sistemy analiza i obrabotki dannykh = Analysis and Data Processing Systems*, 2025, no. 2 (98), pp. 81–92. DOI: 10.17212/2782-2001-2025-2-81-92.