BULLETIN OF NATIONAL ACADEMY OF SCIENCES
OF THE REPUBLIC OF KAZAKHSTAN
ISSN 1991-3494
Volume 4, Number 356 (2015), 101 – 114

METHODS INTER-MODULE AND INTERBLOCK INTERACTIONS OF HARDWARE CONTROL GROUND INFRASTRUCTURE OF ROCKET AND SPACE COMPLEXES

D. V. Pashhenko[1], M. P. Sinev[1], V. V. Kutuzov[1], D. A. Trokoz[1], K. T. Sauanova[2]

[1] Penza State University, Russia,
[2]Almaty university of power engineering & telecommunications, Kazakhstan.
E-mail: klartag@mail.ru

Key words: мodular and block architecture, apparatus means the management, space-rocket technician.
Abstract. With the aim of alignment and providing the opportunities for up scaling, modular*and block type of ground-based Russia Federation infrastructure architecture construction control means are offered.

In the following article the typed block structure is presented, consisting of module and intellectual modules of enrty-exit, connecting to elements of information collection on technical facilities condition.

The methods of interaction organization between modules and blocks of control system, mentioned in the article allow not only to build up an effective highly stressed computation system with high security rate, but also enable provision of support of wide nomenclature of entry-exit modules. At the same time, the use of scandalized protocols in intra-block exchange gives opportunity to integrate into common adapter network of other producers.

The ground infrastructure objects are the main components of rocket-space complexes (RSC), the development and modernization of which are conducted in Russia. Management issues for specified objects are complex processes. In addition, for each class of products its automated system for monitoring and control of technological processes is used, which significantly increases the cost and complexity of their development, produuction and exploitation.

To solve this problem it is recommended to develop and use standardized and scalable hardware and software systems that will reduce the time and costs for development, production and exploitation. In order to unify and provide scalability in the article the use of block-modular principle of the hardware architecture control ground infrastructure RSC is suggested. At the same time the main goal is to develop ways to interconnect and inter-module interaction hardware control ground infrastructure rocket and space technology.

The article provides the structure of a typical block, consisting of a main module and the intellectual input-output modules, which connect the elements of the information collection on the state of technological equipment.

The results of the analysis of the above given structure show that the main problem with the inter-module and inter-unit cooperation is a large amount of network traffic and concludes that the development of hardware interaction methods and software ground infrastructure RSC control, which reduce network traffic, is an urgent task, which is being solved in this paper.

The above article are capable to organize the interaction between the module and the control system bocks can not only build an efficient heavy duty computing network is highly reliable, but also to provide support for a wide range of input-output modules, and the use of standardized protocols in the interconnect exchange provides the opportunity to integrate into a single network third-party manufacturers devices.

УДК 338.45:623

СПОСОБЫ МЕЖМОДУЛЬНОГО И МЕЖБЛОЧНОГО ВЗАИМОДЕЙСТВИЯ АППАРАТНЫХ СРЕДСТВ УПРАВЛЕНИЯ НАЗЕМНОЙ ИНФРАСТРУКТУРОЙ РАКЕТНО-КОСМИЧЕСКИХ КОМПЛЕКСОВ

Д. В. Пащенко[1], М. П. Синев[1], В. В. Кутузов[1], Д. А. Трокоз[1], К. Т. Сауанова[2]

[1] Пензенский государственный университет, Россия,
[2] Алматинский университет энергетики и связи, Казахстан

Ключевые слова: модульно-блочная архитектура, аппаратные средства управления, ракетно-космическая техника.

Аннотация. Объекты наземной инфраструктуры являются основными составляющими ракетно-космических комплексов (РКК), разработка и модернизация которых ведется в РФ. Задачи управления указанными объектами представляют собой сложные технологические процессы. При этом для каждого класса изделий используется своя автоматизированная система контроля и управления технологическими процессами, что существенно увеличивает стоимость и сложность их разработки, производства и эксплуатации.

Для решения этой проблемы предлагается разработать и использовать унифицированные и масштабируемые аппаратно-программные комплексы, что позволит снизить временные и материальные затраты на разработку, производство и эксплуатацию. С целью унификации и обеспечения возможности масштабирования в статье предлагается модульно-блочный принцип построения архитектуры аппаратных средств управления наземной инфраструктурой РКК. При этом основной целью работы является разработка способов межмодульного и межблочного взаимодействия аппаратных средств управления объектами наземной инфраструктуры ракетно-космической техники.

В статье приводится структура типового блока, состоящего из главного модуля и интеллектуальных модулей ввода-вывода, к которым подключаются элементы сбора информации о состоянии технологического оборудования.

В результате анализа приведенной структуры показывается, что основной проблемой при межмодульном и межблочном взаимодействии является большой объем сетевого трафика и делается вывод, что разработка способов взаимодействия аппаратно-программных средств управления наземной инфраструктурой РКК, позволяющих сократить сетевой трафик, является актуальной задачей, которая решается в данной работе.

Приведенные в статье способны организации взаимодействия между модулями и блоками системы управления позволяют не только построить эффективную высоконагруженную вычислительную сети, обладающую высокой надежностью, но и обеспечить поддержку широкой номенклатуры модулей ввода-вывода, а использование стандартизированных протоколов при межблочном обмене дает возможность интегрировать в общую сеть устройства сторонних производителей.

Введение. Одной из основных частей РКК являются объекты наземной инфраструктуры, включающие технологические и вспомогательные объекты и технические средства, основными функциями которых является заправка разгонных блоков и космических аппаратов компонентами топлива и газами с использованием заправочных станций, а также предпусковые работы, включающие управление хранилищами топлива, системами пожара-взрывотушения, водоводом, системами мониторинга и термостабилизации, техническими системами, входящими в состав кислородно-азотного завода [1, 2]. Задачи управления рассматриваемых объектов представляют собой сложные технологические процессы, уникальные для каждого класса изделия [3]. При этом для каждого из классов изделий существует своя автоматизированная система контроля и управления технологическими процессами заправки, слива, хранения топлива, подачи воды, и т.д., что существенно увеличивает стоимость и сложность их разработки, производства и эксплуатации [4].

Поэтому актуальным является решение проблемы создания единой базовой платформы управления наземной инфраструктурой ракетно-космической техники, которое обеспечит работу широкой номенклатуры технических, технологических и вспомогательных систем с использованием унифицированных и масштабируемых аппаратно-программных комплексов, что позволит снизить временные и материальные затраты на разработку, производство и эксплуатацию РКК [5]. При этом важным звеном являются встраиваемые устройства управления (контроллеры), которые должны соответствовать как функциональным требованиям, так и ряду нефункциональных, среди которых требования по надежности, управляемости, безопасности, реконфигурируемости, модернизации и ряду других [6]. В рамках работ по созданию унифицированных аппаратно-программных средств управления наземной инфраструктурой РКК предлагается архитектуру аппаратных средств строить на модульно-блочном принципе.

Цель работы: разработка способов межмодульного и межблочного взаимодействия аппаратных средств управления объектами наземной инфраструктуры ракетно-космической техники (РКТ).

Структура типового блока приведена на рисунке 1. Типовой блок состоит из главного модуля и интеллектуальных модулей ввода-вывода, к которым подключаются элементы сбора информации о состоянии технологического оборудования (датчики, реле давления, реле уровня, дозаторы, счетчики и т.п.) и исполнительные механизмы (клапаны, подогреватели, насосы и т.д.). При этом модуль ввода-вывода реализует полное информационно-логическое взаимодействие с объектами наземной инфраструктуры РКК, которые физически к нему подключены. Взаимодействие модулей ввода-вывода с главным модулем обеспечивается через внутреннюю шину блока.

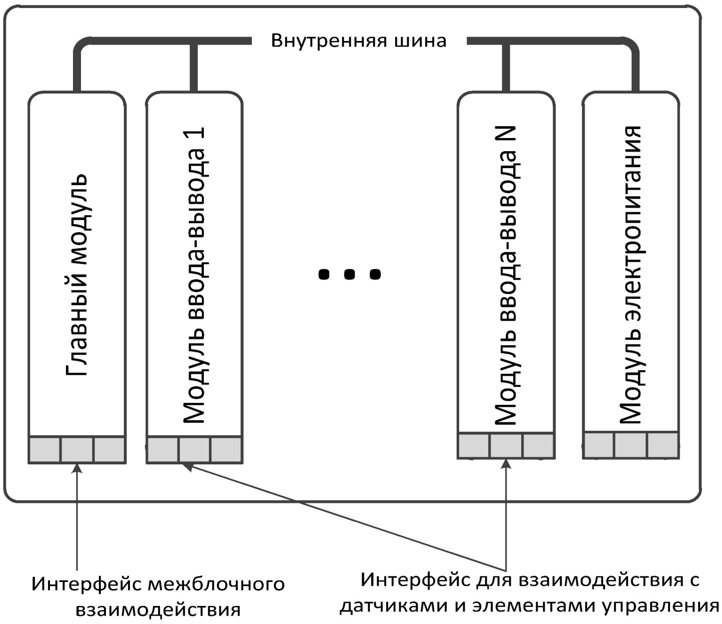

Рисунок 1 – Структура типового блока

В основе, как главного модуля, так и модулей ввода-вывода лежат современные 32-х разрядные микроконтроллеры российского производства серии 1886 [7]. Характеристики используемых микроконтроллеров приведены в таблице 1.

Данные микроконтроллеры обладают хорошими характеристиками, высокими тактовыми частотами и достаточным объемом внутренней памяти, а так же широким набором поддерживаемых интерфейсов.

Блоки взаимодействуют между собой через специальный интерфейс межблочного взаимодействия, реализованный в главном модуле каждого блока. Таким образом, имеется два уровня взаимодействия: межмодульное и межблочное (рисунок 2).

Таблица 1 – Сводная таблица характеристик микроконтроллеров

	Модуль ввода-вывода	Главный модуль
Название	1986ВЕ93У	1986ВЕ1Т
Корпус	Н16.48-1В	4229.132-3
Ядро	*ARM Cortex-M3*	*ARM Cortex-M1*
ПЗУ	128 Кбайт *Flash*	
ОЗУ	32 Кбайт	48 Кбайт
Питание	2.2...3.6 В	3.0...3.6 В
Частота	80 МГц	140 МГц
UART	2	
CAN	2	

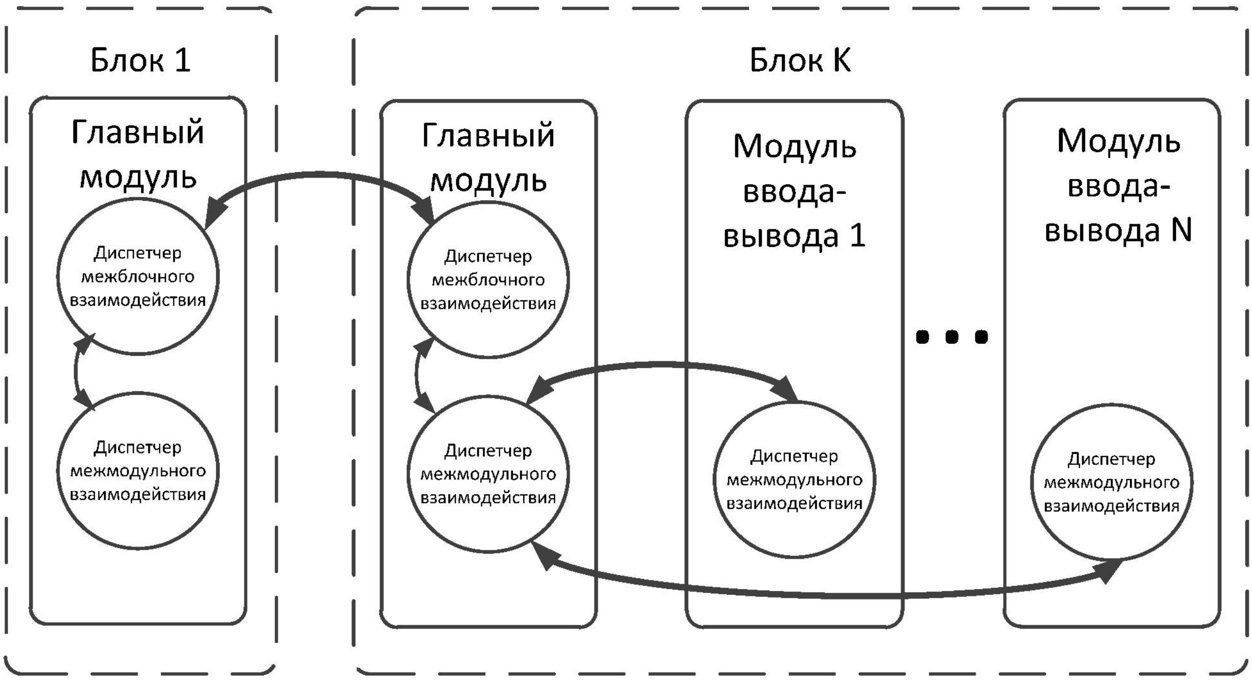

Рисунок 2 – Схема взаимодействия системы

Пусть $B = \{b_1, b_2, \ldots, b_N\}$ – множество блоков управления наземной инфраструктурой ракетно-космических комплексов, где N – общее количество. При этом B можно представить как множество, состоящее из элементов

$$b_i \equiv \{m_i, q_{i1}, \ldots, q_{iK_i}, p\}, \tag{1}$$

где m_i – главный модуль i-го блока, q_{ij} – j-й модуль ввода-вывода i-го блока управления, K_i – общее количество модулей ввода-вывода в i-м блоке, p – главный модуль i-го блока. Сетевой трафик межмодульного взаимодействия i-го блока может быть определен

$$V_i = \sum_{j=1}^{K_i} f_{traf}^{in}(m_i, q_{ij}), \tag{2}$$

где $f_{traf}^{in}(m_i, q_{ij})$ – функция определения сетевого трафика между главным модулем m_i и j-й модуль ввода-вывода i-го блока управления.

Аналогичным образом может быть определен сетевой трафик межблочного взаимодействия

$$V = \sum_{i=1}^{N} \sum_{z=1}^{N} f_{traf}^{out}(b_i, b_z), i \neq z, \tag{3}$$

где $f_{traf}^{out}(b_i, b_z)$ – функция определения сетевого трафика между главным модулем i-го блока управления и главным модулем z-го блока управления.

Способы взаимодействия. Межмодульное и межблочное взаимодействие можно реализовать следующими способами:

- взаимодействие на уровне общей памяти;
- взаимодействие на уровне команд.

Первый способ имеет ряд недостатков:

1. в каждом модуле необходимо иметь копию общей памяти, что при росте количества элементов системы является ограничивающим фактором возможности масштабирования средств управления;
2. требуются затраты процессорного времени на сканирование собственной копии общей памяти, для определения изменения информации;
3. проблемы синхронизации общей памяти для каждого блока.

Второй способ труднее реализовать на практике, так как он требует разработки более сложных алгоритмов работы диспетчера взаимодействия, создания унифицированного, расширяемого набора команд, зависящих от исполняемого алгоритма управления.

Межмодульное взаимодействие. На межмодульном уровне предлагается использовать способ взаимодействия на уровне общей памяти. В данном случае общая память содержится в главном модуле блока. Поддержание актуальности данных хранящихся в общей памяти производится самим главным модулем посредством последовательного опроса каждого модуля ввода-вывода и чтения данных об изменении состояния технологического оборудования. Аналогично происходит передача управляющих воздействий на технологическое оборудование.

Доступ к модулям ввода-вывода, как было написано ранее, производится по внутренней шине блока. При этом внутренняя шина может быть двух основных типов:

- n-разрядная параллельная шина;
- последовательная шина.

При всех своих достоинствах, это и высокая пропускная способность и малое время доступа и простота реализации протокола обмена, параллельная шина имеет ряд существенных недостатков, таких как трудности в разработке и производстве за счет большого числа сигнальных линий и низкую помехозащищенность.

С другой стороны последовательная шина во многом лишена этих недостатков при, часто, значительном снижение пропускной способности и значительном увеличении времени доступа.

Можно так же упомянуть вариант с доступом к модулям ввода вывода через специальный коммутатор (топология «звезда») [8]. При этом каждый модуль подключается отдельным каналом, канал может быть реализован как последовательный интерфейс, так и как параллельный, к специальному коммутатору, реализующему на аппаратном уровне протокол обмена. Такой способ предполагает высокую скорость обмена и низкие задержки доступа, но требует очень больших аппаратных затрат и сложен для реализации, поэтому в дальнейшем рассматривать его не будем.

Так же для межмодульного обмена большое значение имеет тип доступа к модулям ввода вывода. Можно выделить два типа доступа:

- непосредственный доступ, который обычно реализуется через n-разрядную параллельную шину;
- опосредованный доступ через промежуточный элемент, в роли которого может выступать сопроцессор ввода-вывода или микроконтроллер, в таком случае может использоваться как параллельная, так и последовательная шины.

В первом случае вся работа блока происходит в главном модуле и достаточно большое время тратится не только на опрос дулей ввода-вывода, но и на обработку полученных данных и преобразовании этих данных в состояние управляемых элементов.

Во втором случае всю предварительную работу по обработке данных полученных от управляемых элементов выполняет микроконтроллер, который расположен непосредственно на модуле ввода-вывода, это приводит к разгрузке главного модуля за счет распараллеливания задачи между всеми модулями блока.

На рисунке 3 представлено сравнение загрузки процессора центрального модуля при переключении реле в модуле ввода-вывода для случая с непосредственным доступом по параллельной шине и случая с опосредованными доступом по последовательной шине. Здесь видно, что во втором случае процессор посылает лишь команду на переключение и через какое-то время считывает уже готовое состояние, оставшееся процессорное время свободно для других задач.

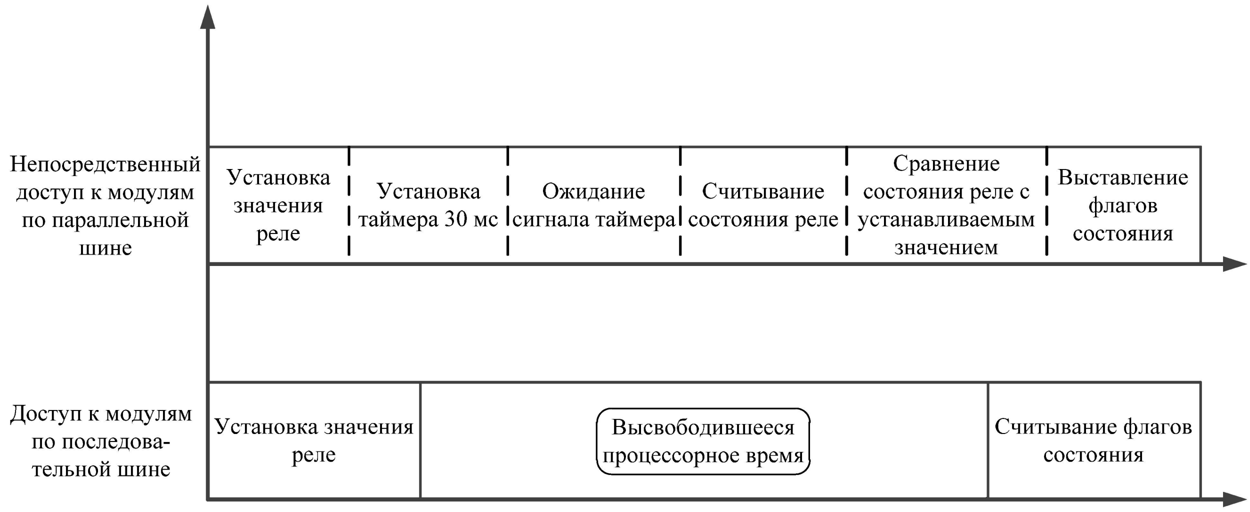

Рисунок 3 – Сравнение загрузки центрального модуля при переключении реле

Учитывая специфику области применения, наиболее важными свойствами являются надежность и скорость выполнения команд. Поэтому предлагается использовать доступ к модулям ввода-вывода по последовательному интерфейсу с обработкой большей части информации в самих модулях.

Как интерфейс межмодульного взаимодействия предлагается использовать *CAN* (*Controller Area Network*) [9]. *CAN* обеспечивает скорость передачи до 1 Мбит/с и обладает такими преимуществами как доступ с разрешением коллизий и контроль ошибок с восстановлением поврежденных бит [10]. В современных микроконтроллерах российского производства реализована аппаратная поддержка данного интерфейса [11].

Использование интерфейса *CAN* накладывает так же определенные ограничения [12]:

– устройства с низким приоритетом в определенных ситуациях могут никогда не получить доступ к шине;

– скорость до 1 Мбит/с может быть недостаточна при интенсивном обмене;

– длина поля данных каждого кадра ограниченна восьмью байтами.

На рисунке 4 представлена схема обмена главного модуля с модулями ввода-вывода.

Передача ведется кадрами. Опрос начинается с передачи одного или нескольких кадров содержащих информацию управляющих воздействиях на технологическое оборудование, если такие воздействия предусмотрены. Далее идет кадр тишины, в это время модуль ввода-вывода обрабатывает предыдущие кадры и передает управляющие воздействия на технологическое оборудование. Следом идет запрос на передачу, после которого модуль ввода-вывода передает кадры с информацией об изменении состояния технологического оборудования подключенного к нему. Последним передается кадр конца данных, сигнализирующий главному модулю о завершении обмена с текущим модулем ввода-вывода. Описанная процедура обмена происходит последовательно для каждого модуля ввода-вывода в блоке.

Последовательный опрос модулей ввода-вывода позволяет избежать проблем с доступом к шине. Сократить объем передаваемой информации можно путем обмена кадрами с данными об изменении состояния технологического оборудования.

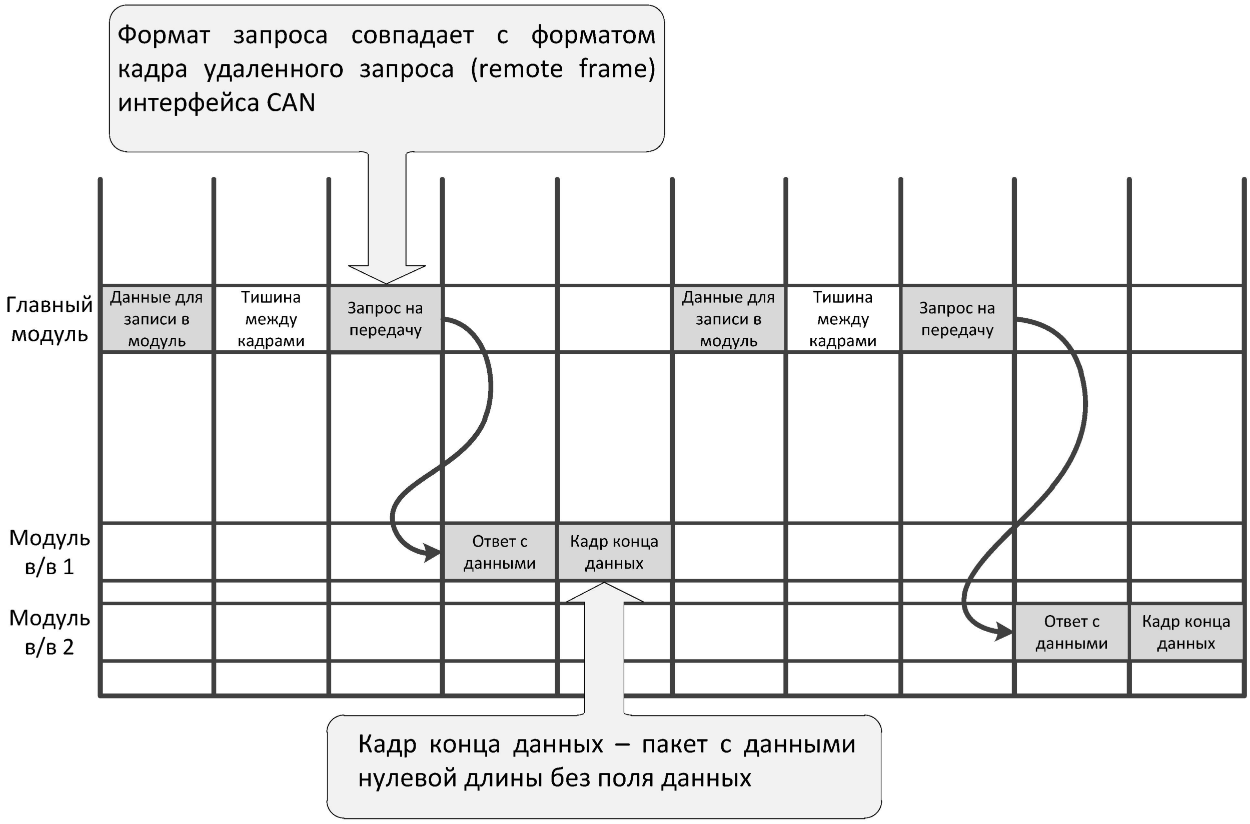

Рисунок 4 – Схема межмодульного обмена

Ограничение в восемь байт для поля данных не позволяет передавать за раз большие объемы данных, поэтому поле данных имеет специальный формат (рисунок 5).

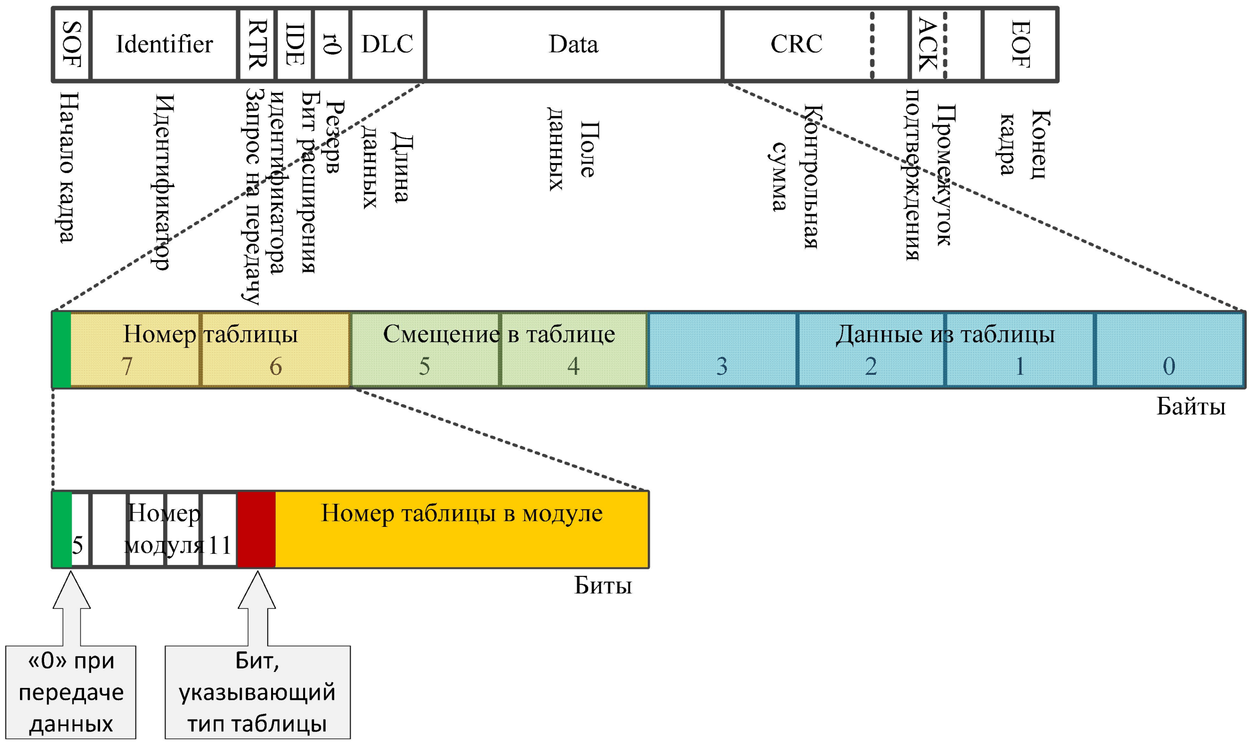

Рисунок 5 – Формат кадра с данными

Поле данных разбито на 4 части:

1. данные из таблицы – до четырех байт данных без описания типа;
2. смещение в таблице – два байта, указывающие на смещение в таблице, по которому данные должны быть записаны;
3. номер таблицы, который в свою очередь состоит из двух частей:
   - номер модуля;
   - бит, указывающий тип таблицы (таблица с флагами или с регистрами);
   - номер таблицы в модуле;
4. бит, указывающий тип кадра.

Такой формат поля данных позволяет передавать только изменения информации, указывая только ее смещение в таблице, адресуемой посредством её номера.

Так же имеется возможность передачи управляющих команд с несколькими аргументами. Формат управляющего кадра представлен на рисунке 6.

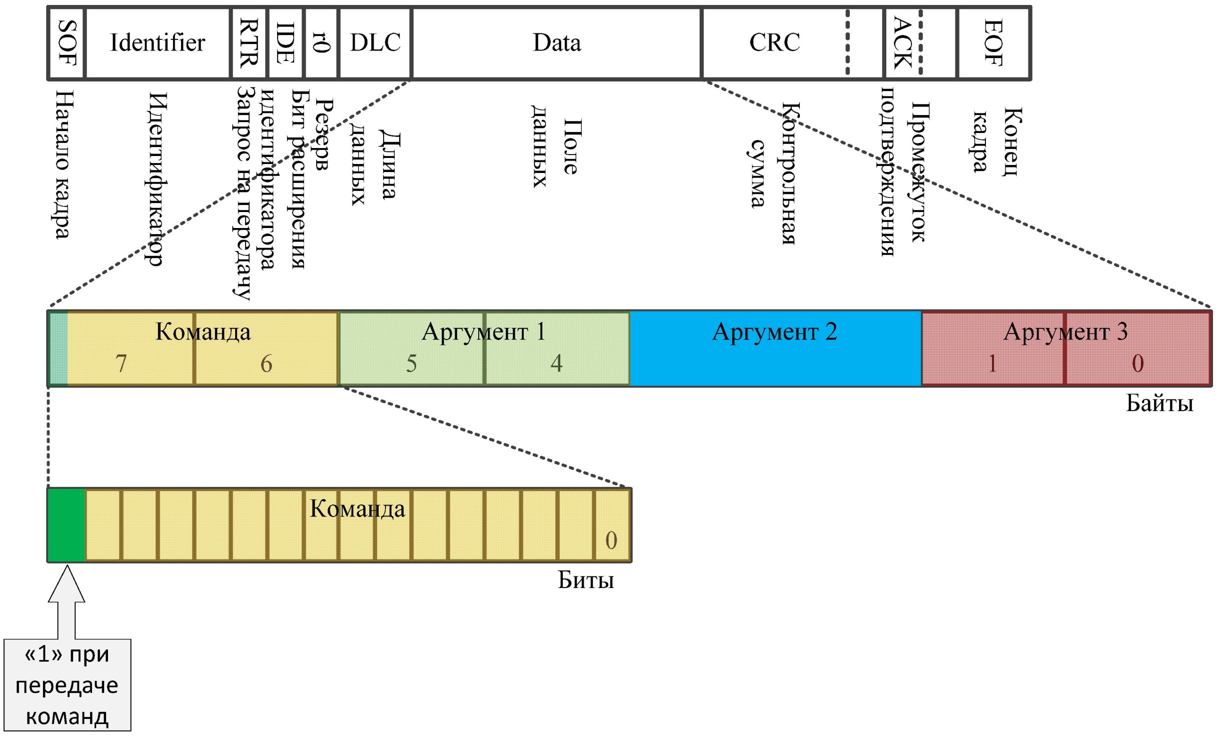

Рисунок 6 – Формат кадра команды

Поле данных состоит из следующих полей:

1. бит, указывающий тип кадра;
2. 15 бит - команды;
3. 2 байта - аргумент 1;
4. 2 байта - аргумент 2;
5. 2 байта - аргумент 3.

Такой формат позволяет передавать различные управляющие команды, в том числе команды настройки параметров модуля ввода-вывода и удаленной перепрошивки модуля ввода-вывода.

Межблочное взаимодействие. Для определения способа межблочного взаимодействия большое значение имеет сетевая топология системы управления. Существует множество способов соединения сетевых устройств[13]. Выделяют 3 базовые топологии:

– шина - все блоки взаимодействуют по общему каналу,

– кольцо - блок соединён линиями связи только с двумя другими: от одного он только получает информацию, а другому только передаёт,

– звезда - все блоки взаимодействуют с главным по отдельному каналу.

«Шина» представляет собой единый кабель, который выступает в роли магистрали, к которой посредством общей шины подключаются блоки системы. На концах шины расположены терминаторы (согласующий резистор), которые не позволяют сигналу отражаться, что исключает информационные шумы в сети.

Отправляемое блоком сообщение распространяется на остальные блоки сети. Каждый блок проверяет, кому адресовано сообщение и если ему, то обрабатывает информацию. Для того чтобы исключить одновременную посылку данных, применяется либо «несущий» сигнал, либо один из блоков является главным и «даёт слово» остальным блокам.

Достоинствами такой топологии является то, что её просто установить и настроить, а количество кабеля затрачивается гораздо меньше, чем в других топологиях, а так же при выходе из строя одного из блоков сеть продолжает функционировать. Недостатком такой топологии является проблемы синхронизации общей памяти для каждого блока.

Топология «кольцо». В такой топологии все блоки подключены последовательно друг к другу, чем образовывают замкнутую сеть. В «кольце», в отличие от других топологий («звезда», «шина»), не используется конкурентный метод посылки данных, компьютер получает данные от стоящего предыдущим в списке адресатов и перенаправляет их далее, если они адресованы не ему. Список адресатов генерируется блоком, являющимся генератором маркера. Сетевой модуль генерирует маркерный сигнал и передает его следующей системе Следующая система, приняв сигнал, не анализирует его, а просто передает дальше. Сигнал в такой связи всегда двигается в одну сторону.

Достоинствами такой связи является то, что ее просто установить и собрать с использованием минимального количества оборудования, а так же устойчивая работа связи. Но как только из строя выходит один из блоков, или происходит порыв на линии вся связь приходит в негодность и такие проблемы достаточно сложно найти и устранить, а конфигурирование и настройка такой связи вызывает массу хлопот. Так же недостатками данной топологии является то, что в главном модуле необходимо иметь копию общей памяти подчиненных, что при росте количества элементов системы является ограничивающим фактором возможности масштабирования средств управления, а так же требуются затраты процессорного времени на сканирование собственной копии общей памяти для определения изменения информации.

Топология «звезда» организована по принципу централизованного обращения, т.е. есть центральный блок или концентратор (хаб), к которому подключены все блоки сети. В такой топологии может быть очень много блоков, причём все они должны быть на удалении от «хаба» не более 100 метров (такова особенность принципа распространения сигналов). Для топологии «звезда» требуется не так много оборудования, а обслуживание происходит гораздо проще. Недостатком же является то, что при выходе из строя «хаба» все блоки остаются без подключения, а так же большие затраты на кабель и ограничение по количеству компьютеров в зависимости от количества выходов на концентраторе.

На рисунке 7 представлена топология соединения блоков при подключении типа точка-точка, где *Bm* – главный блок сети или подсети, *Bs* – подчиненный блок. По своему принципу эта топология схожа с топологией «звезда».

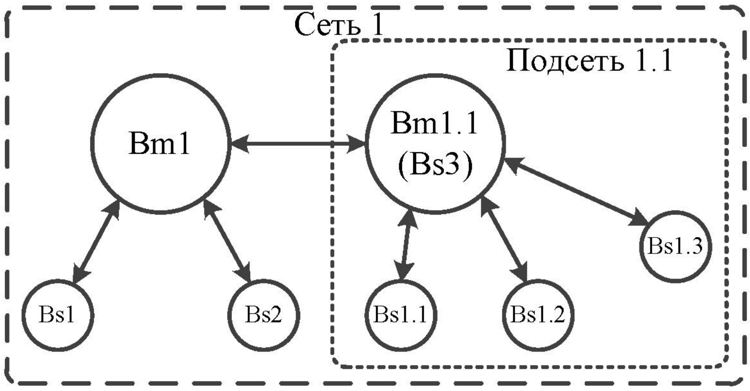

Рисунок 7 – Топология соединения блоков при подключении точка-точка

На рисунке 8 представлена топология соединения блоков при подключении к общей шине, где *Bm* – главный блок сети или подсети, *Bs* – подчиненный блок.

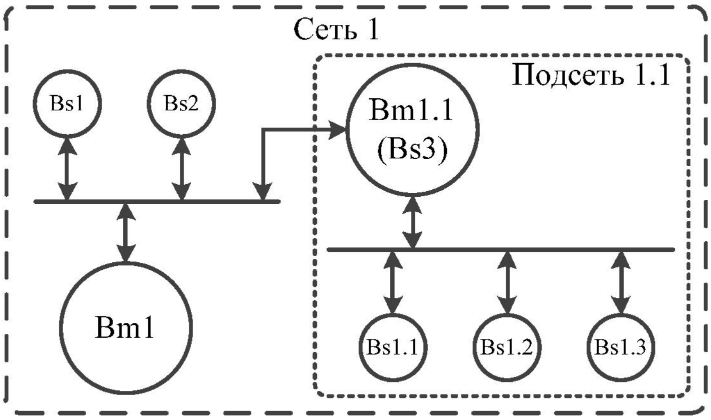

Рисунок 8 – Топология соединения блоков при подключении по общей шине

В качестве протокола передачи данных в топологии «Общая шина» можно использовать *Time-Triggered Protocol* [14-16]. Недостатком использования топологии является проблема синхронизации общей памяти для каждого блока.

В качестве протокола передачи данных в топологии «Звезда» можно использовать *MODBUS* [17, 15]. Недостатками использования топологии «Звезда» являются:

– в главном модуле необходимо иметь копию общей памяти подчиненных, что при росте количества элементов системы является ограничивающим фактором возможности масштабирования средств управления;

– требуются затраты процессорного времени на сканирование собственной копии общей памяти для определения изменения информации.

Рассмотрим протокол информационно-логического взаимодействия блока с сопряженным устройством, основанный на стандарте *MODBUS*.

Режим работы *MODBUS – RTU*.

Скорость передачи данных – 19200 бит/с.

Взаимодействие блока с сопряженным устройством организуется по принципу *master/slave*. Блок работает в режиме *slave*, а сопряженное устройство (ПЭВМ на основе промышленного ноутбука) работает в режиме *master*.

Данные по каналу *RS*-485 передаются 10-битными посылками, состоящими из:

– стартового бита;

– восьми битов данных;

– стопового бита.

Бит чётности не используется.

Признак конца сообщения - тайм-аут, длительность которого не менее чем время передачи 3,5 посылок.

В таблице 2 приведен список функций, поддерживаемых блоком.

Таблица 2 – Список функций, поддерживаемых блоком

Код функций (hex)	Назначение
0x01	Чтение флагов
0x03	Чтение регистров данных
0x0F	Установка флагов
0x10	Установка регистров данных
Примечание. В данной таблице и далее значения полей, представленные в шестнадцатеричной системе счисления, начинаются с символов 0x. Отсутствие символов обозначает представление значений полей в десятичной системе счисления.	

При взаимодействии блок выполняет установку/чтение 16-разрядных регистров данных, установку/чтение флагов и чтение идентификационных данных блока.

Разрядность регистра – 16 бит (формат *WORD* - 2 байта). При обмене данными старший байт следует первым.

Флаг занимает 1 бит.

В сообщениях, в которых передаются состояния флагов, длина поля данных кратна байту. В один байт записываются состояния восьми флагов, начиная с младшего бита по возрастанию адресов флагов. Неиспользованные биты в байте данных заполняются нулями.

Во всех полях сообщения используется шестнадцатеричный код.

Для хранения в блоке целочисленных данных в формате DWORD (4 байта) используются два последовательно расположенных регистра, адрес должен быть четным.

В таблицах 3 и 4 показан формат сообщений при выполнении чтения регистров данных (код функции 0x03).

Таблица 3 – Формат сообщения при запросе на чтение регистров данных

Функциональное назначение поля	Адрес блока	Код функции	Адрес первого регистра	Количество регистров	Контрольная сумма
Длина, в байтах	1	1	2	2	2
Значение	*A*	0x03	*AR*	*n*	*CRC*
Значение адреса (A) – от 1 до 247.					

Таблица 4 – Формат сообщения в ответе блока при выполнении чтения регистров данных

Функциональное назначение поля	Адрес блока	Код функции	Количество байт с данными	Данные (содержимое регистров)	Контрольная сумма
Длина, в байтах	1	1	1	2 x *n*	2
Значение	*A*	0x03	2 x*n*		*CRC*
n – количество регистров.					

Независимо от топологии и протокола обмена память модуля размечается на таблицы, имеющие уникальный диапазон адресов, по которым происходит доступ к ее элементам и модификатор типа данных – регистры или флаги. Межблочный обмен, как и межмодульный, происходит только блоками изменившейся информации. Для обеспечения такого обмена в описателе таблице имеется дополнительное поле, содержащее информацию о наличии изменений в данной таблице. При любом изменении вся таблица помечается как «измененная» и при выполнении следующего цикла обмена таблица будет передана в другой блок. При отсутствии же каких-либо изменений обмен вообще не будет происходить. Таким образом, сильно сокращается объем генерируемого трафика.

Рассмотрим различные варианты алгоритмов межблочного взаимодействия при разных топологиях построения сети. Каждый алгоритм может быть описан в виде графа [18], вершинами которого являются взаимодействующие блоки, а ребрами атомарные операции, выполняющиеся в определенном порядке при обмене. На рисунке 8 представлены алгоритмы взаимодействия для сети, состоящей из 4-х блоков.

Для топологии «звезда» при обработке всей информации на главном блоке b^m (рисунок 8*a*) каждый подчиненный блок b_i передает ему информацию об изменении состояния технологического оборудования $q_{send}(b^m, b_i)$. На главном блоке происходит выполнение алгоритмов управ-ления $q_{proc}(b^m)$ и последующая передача управляющих команд в подчиненные $q_{order}(b^m, b_i)$ для $\forall i \in N$.

При распределенной обработке всей информации для топологии «звезда» (рисунок 8*b*) на центральный блок накладывается дополнительная функция – маршрутизации команд $q_{route}(b^m, b_i)$

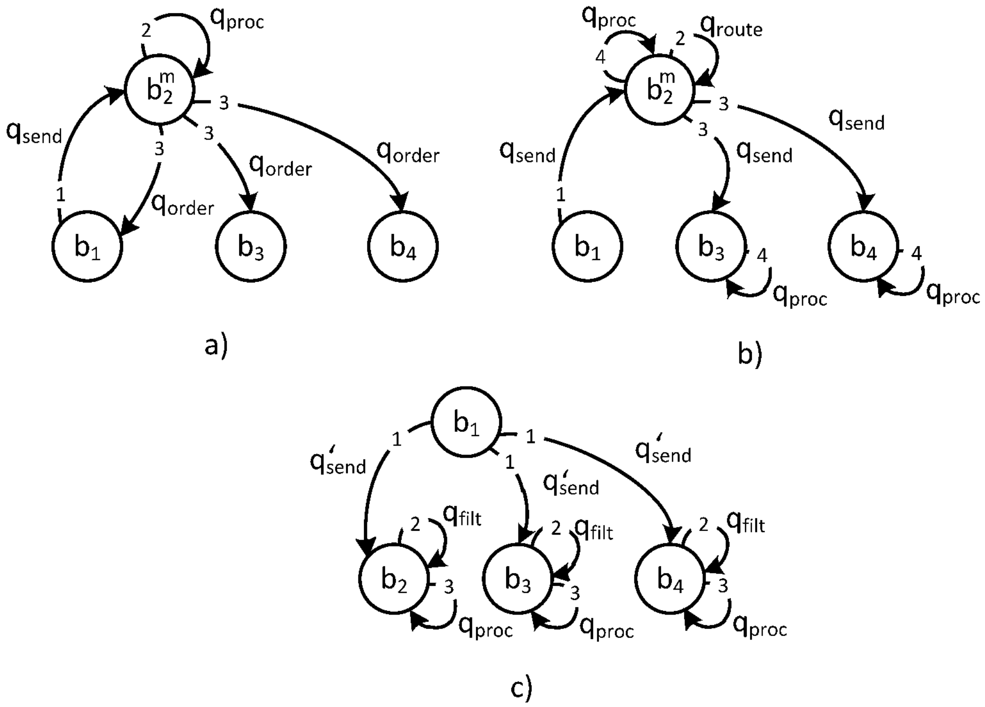

Рисунок 8 – Варианты алгоритмов межблочного взаимодействия при разных топологиях построения сети

для $\forall i \in N$. В данном случае по внутренним таблицам, хранящимся в памяти центрального блока, определяется какие именно подчиненные блоки заинтересованы в данной команде и для указанных блоков происходит ее ретрансляция $q_{send}(b^m, b_i)$ для $\forall i \in N$. После синхронизации памяти каждый блок выполняет свою часть распределенного алгоритма управления технологическим оборудованием $q_{proc}(b_i)$ для $\forall i \in N$.

При распределенной обработке всей информации для блоков организованных в сеть согласно топологии «Общая шина» (рисунок 8*c*) каждый блок b_i выдает широковещательную команду на общую шину $q'_{send}(b_i)$. Остальные блоки b_j , $i \neq j$ проверяют собственную принадлежность адресованному сообщению $q_{filt}(b_j, b_i), i \neq j$ и в случае успеха обрабатывают информацию.

Таким образом, можно составить математические модели приведенных на рисунке 8 алгоритмов с использованием логики предикатов [19, 20]:

– для топологии «звезда» при обработке всей информации на главном блоке

$$M_a(b_1, b_2^m, b_3, b_4) \equiv \left(q_{send}(b_1, b_2^m) \longrightarrow q_{proc}(b_2^m)\right) \longrightarrow \\ \longrightarrow \left(q_{order}(b_2^m, b_1) \,\&\, q_{order}(b_2^m, b_3) \,\&\, q_{order}(b_2^m, b_4)\right); \tag{4}$$

– для топологии «звезда» при распределенной обработке всей информации

$$M_b(b_1, b_2^m, b_3, b_4) \equiv \left(q_{send}(b_1, b_2^m) \longrightarrow q_{route}(b_2^m)\right) \longrightarrow \\ \longrightarrow \begin{pmatrix} \left(q_{send}(b_2^m, b_3) \longrightarrow q_{proc}(b_3)\right) \vee \\ \vee \left(q_{send}(b_2^m, b_4) \longrightarrow q_{proc}(b_4)\right) \vee q_{proc}(b_2^m) \end{pmatrix}; \tag{5}$$

– при распределенной обработке всей информации для блоков организованных в сеть согласно топологии «Общая шина»

$$M_c(b_1,b_2,b_3,b_4) \equiv q'_{send}(b_1) \longrightarrow \begin{pmatrix} \left(q_{filt}(b_1,b_2) \longrightarrow q_{proc}(b_2)\right) \vee \\ \vee \left(q_{filt}(b_1,b_3) \longrightarrow q_{proc}(b_3)\right) \vee \\ \vee \left(q_{filt}(b_1,b_4) \longrightarrow q_{proc}(b_4)\right) \end{pmatrix}. \quad (6)$$

Формализация данных алгоритмов позволяет говорить о непротиворечивости по Гилберту, а применение в объектах управления объектами наземной инфраструктуры о решаемости и адекватности.

Заключение. С использованием приведенных способов появляется возможность построения эффективной высоконагруженной вычислительной сети, обладающей высокой надежностью за счет аппаратного контроля целостности передаваемых данных. Благодаря унификации формата внутренних сообщений осуществляется поддержка широкой номенклатуры модулей ввода-вывода. Использование стандартизированных протоколов при межблочном обмене позволяет интегрировать в общую сеть устройства сторонних производителей, поддерживающих данных стандарт взаимодействия.

Исследования ведутся в рамках ФЦП «Исследования и разработка по приоритетным направлениям развития НТК России на 2014-2020 г.» (Соглашение № 14.574.21.0045 от 19.06.14, UID «RFMEFI57414X0045»).

ЛИТЕРАТУРА

[1] Fortescue P., Swinerd G., Stark J. (ed.). Spacecraft systems engineering. – John Wiley & Sons, 2011.

[2] Ingalls J., Cipolletti J. Standards and Specifications for Ground Processing of Space Vehicles: from an Aviation-based Shuttle Project to Global Application. American Institute of Aeronautics and Astronautics – 2011.

[3] М. П. Синев, В. И. Лапшин, Д. А. Левин, М. М. Пинский Автоматизация управления процессом заправки. //Надежность и качество: тр. междунар. симп. – 2011. – № 1. – С. 54–57.

[4] Майданович О. В. и др. Интеллектуальные информационные технологии наземно-космического мониторинга сложных объектов: состояние и перспективы развития //Материалы конференции" Интеллектуальные технологии в управлении"(ИТУ-2012)(9-11 октября 2012 г., Санкт-Петербург).-СПб: ГНЦ РФ ОАО" Концерн" ЦНИИ" Электроприбор. – 2012. – С. 38-52.

[5] Razani M. Information, Communication, and Space Technology. – CRC Press, 2012.

[6] Gaonkar R. Fundamentals of Micro Controllers and Applications in Embedded Systems. – 2012.

[7] сайт ЗАО «ПКК Миландр» [Электронный ресурс]. URL: *http://milandr.ru/en/* (дата обращения: 23.10.2013).

[8] В.В. Кутузов Реализация и сравнение производительности высокоскоростных способов передачи сообщений в многопроцессорных системах - Труды международного симпозиума «Надежность и качество», Издательство ПГУ, Пенза, 2013. – стр. 292-295.

[9] сайт CANopen Special Interest Group [Электронный ресурс]. URL: http://www.canopen.org/ (дата обращения: 23.10.2013).

[10] Lawrenz. W. CAN system engineering. – Springer, 2013..

[11] Шумилин,С. Новая серия отечественных 32-разрядных высокопроизводительных микроконтроллеров семейства 1986 на базе процессорного ядра *ARM CortexM3*/ С. Шумилин. - «Компоненты и технологии», №10, 2008. - стр. 60-64.

[12] Di Natale M. et al. Understanding and using the controller area network communication protocol: theory and practice. – Springer, 2012.

[13] Таненбаум, Э. Распределённые системы. Принципы и парадигмы / Э. Таненбаум, М.ван Стеен. — Санкт-Петербург: Питер, 2003.

[14] Śmieja M., Rapiński J. Prospects for the development of automotive networks based on ethernet //Journal of KONES. – 2011. – Т. 18. – С. 445-452.

[15] Fonseca J. A. Real-Time Communications: from Fieldbuses and Industrial Automation to Wireless and Vehicular Applications. – 2012.

[16] Клепиков,В.Распределённая архитектура перспективных встроенных систем управления / В.Клепиков, В.Федюкин, Л.Бондарев. - Электроника: Наука, Технология, Бизнес, № 6, 2007.

[17] Оригинальные спецификации протокола *MODBUS* [Электронный ресурс]. *URL: http://www.modbus.org/tech.php/* (дата обращения: 23.10.2013).

[18] Дистель, Р. Теория графов / Р. Дистель. - Новосибирск: изд-во Ин-та математики, 2002.

[19] Яблонский, С. В. Введение в дискретную математику / С. В. Яблонский. – М.: Высшая школа, 2003. – 384 с.

[20] Юдицкий, С. А. Проектирование дискретных систем автоматики / С. А. Юдицкий, А. А. Тогаевская, Т. К. Ефремова. – М. : Машиностроение, 1980.– 232 с.

REFERENCES

[1] Fortescue P., Swinerd G., Stark J. (ed.). *Spacecraft systems engineering. – John Wiley & Sons*, **2011**. (in Eng).

[2] Ingalls J., Cipolletti J. *Standards and Specifications for Ground Processing of Space Vehicles: from an Aviation-based Shuttle Project to Global Application. American Institute of Aeronautics and Astronautics* – **2011**. (in Eng)

[3] M. P. Sinev, V. I. Lapshin, D. A. Levin, M. M. Pinskij *Avtomatizacija upravlenija processom zapravki*. //Nadezhnost' i kachestvo: tr. mezhdunar. simp. – **2011**. – № 1. – S. 54–57.

[4] Majdanovich O. V. i dr. *Intellektual'nye informacionnye tehnologii nazemno-kosmicheskogo monitoringa slozhnyh ob#ektov: sostojanie i perspektivy razvitija //Materialy konferencii" Intellektual'nye tehnologii v upravlenii"(ITU-2012)*(9-11 oktjabrja 2012 g., Sankt-Peterburg).-SPb: GNC RF OAO" Koncern" CNII" Jelektropribor. – **2012**. – S. 38-52.

[5] Razani M. *Information, Communication, and Space Technology. – CRC Press*, **2012**. (in Eng)

[6] Gaonkar R. *Fundamentals of Micro Controllers and Applications in Embedded Systems*. – **2012**. (in Eng)

[7] *sajt ZAO «PKK Milandr» [Jelektronnyj resurs]. URL: http://milandr.ru/en/* (data obrashhenija: 23.10.**2013**).

[8] V.V. Kutuzov *Realizacija i sravnenie proizvoditel'nosti vysokoskorostnyh sposobov peredachi soobshhenij v mnogoprocessornyh sistemah/V.V. Kutuzov - Trudy mezhdunarodnogo simpoziuma «Nadezhnost' i kachestvo», Izdatel'stvo PGU, Penza*, **2013**. – str. 292-295

[9] sajt *CANopen Special Interest Group [Jelektronnyj resurs]. URL: http://www.canopen.org/* (data obrashhenija: 23.10.**2013**). (in Eng)

[10] Lawrenz. W. *CAN system engineering. – Springer*, **2013**. (in Eng)

[11] Shumilin,S. *Novaja serija otechestvennyh 32-razrjadnyh vysokoproizvoditel'nyh mikrokontrollerov semejstva 1986 na baze processornogo jadra ARM CortexM3/ S. Shumilin. - «Komponenty i tehnologii»*, №10, **2008**. - str. 60-64.

[12] Di Natale M. et al. *Understanding and using the controller area network communication protocol: theory and practice. – Springer*, **2012**. (in Eng)

[13] Tanenbaum, Je. *Raspredeljonnye sistemy. Principy i paradigmy / Je. Tanenbaum, M.van Steen. — Sankt-Peterburg: Piter*, **2003**.

[14] Śmieja M., Rapiński J. *Prospects for the development of automotive networks based on ethernet //Journal of KONES*. – **2011**. – T. 18. – S. 445-452. (in Eng).

[15] Fonseca J. A. *Real-Time Communications: from Fieldbuses and Industrial Automation to Wireless and Vehicular Applications*. – **2012**. (in Eng).

[16] Klepikov,V.*Raspredeljonnaja arhitektura perspektivnyh vstroennyh sistem upravlenija / V.Klepikov, V.Fedjukin, L.Bondarev. - Jelektronika: Nauka, Tehnologija, Biznes*, № 6, **2007**.

[17] *Original'nye specifikacii protokola MODBUS [Jelektronnyj resurs]. URL: http://www.modbus.org/tech.php/* (data obrashhenija: 23.10.**2013**).

[18] Distel', R. *Teorija grafov / R. Distel'. - Novosibirsk: izd-vo In-ta matematiki*, **2002**.

[19] Jablonskij, S. V. *Vvedenie v diskretnuju matematiku / S. V. Jablonskij. – M.: Vysshaja shkola*, **2003**. – 384 s.

[20] Judickij, S. A. *Proektirovanie diskretnyh sistem avtomatiki* / S. A. Judickij, A. A. Togaevskaja, T. K. Efremova. – M. : Mashinostroenie, **1980**.– 232 s.

РАКЕТА-ҒАРЫШТЫҚ КЕШЕНДЕРДІҢ ЖЕР БЕТІНДЕГІ ИНФРАҚҰРЫЛЫМЫН БАСҚАРАТЫН АППАРАТТЫҚ ҚҰРАЛДАРДЫН МОДУЛЬ АРАЛЫҚ ЖӘНЕ БЛОК АРАЛЫҚ ӨЗАРА ӘРЕКЕТТЕСУ ӘДІСТЕРІ

Д. В. Пащенко[1], М. П. Синев[1], В. В. Кутузов[1], Д. А. Трокоз[1], К. Т. Сауанова[2]

[1]Пенза мемлекеттік университеті, Ресей,
[2]Алматы Энергетика және байланыс университеті, Қазақстан

Тірек сөздер: модульно-блоктық архитектура, аппаратық басқару құралдар, ракета - ғарыштық техника.

Аннотация. Бұл мәселені шешу үшін бір қалыпқа, түрге келтірілетін, қорлары өскен сайын өнімділігін өсіре алатын аппаратты-программалық кешендер құру және қолдану жолы ұсынылады, әріне бұл жол құру, шығару және пайдаланудағы уақыт және заттық шығындарды азайтуға мүмкіндік береді. Бір қалыпқа келтірілу және қорлары өскен сайын өнімділігін өсіре алатын мүмкіндіктерді қамтамасыз ету үшін жер бетіндегі РҒК инфрақұрылымын басқару аппараттық құралдары архитектурасың құру қағидасы ретінде модульді-блоктық әдіс ұсынылып жатыр. Сонымен бірге жұмыстың негізгі мақсаты бұл ракета - ғарыштық техниканың жер бетіндегі инфрақұрылымының объекттілерін басқаратын аппараттық құралдардын модуль аралық және блок аралық өзара әрекеттесу әдістердің құру.

Мақалада енгізу-шығару және негізгі модульден тұратын үлгілі блоктың құрылымы келтірілген, мұнда технологиялық жабдықтардың күйі туралы мәліметтерді жинау элементтері интелектуалды енгізу-шығару модуліне қосылған.

Келтірған құрылымды талдау нэтижесінде модуль аралық және блок аралық өзара әрекеттесудегі негізгі мәселе бұл желідегі трафиктің үлкен көлемі ал қорытындылауда желідегі трафикті қысқартуға мүмкіндік беретін РҒК жер бетіндегі инфрақұрылымын аппаратты-программалық басқару құралдарының өзара әрекеттесу әдістердің құру бұл осы жұмыста ұйғарылған өзекті есеп.

Мақалада келтірген басқару жүйесінің модуль және блоктар арасында өзара әрекеттесулерді ұйымдастыру әдістері сенімділігі, тиімділігі жоғары есептеуіш желілерді құруға мүмкіндік берумен бірге енгізу-шығару модулдерінің кең тізімін қолдауды қамтамасыз етеді, ал блок аралық ақпаратпен алмасуда стандартталған хаттамаларды қолдану ортақ желіге бөтен өндірушілердің құралдарың интеграциялауға мүмкіндік беріді.

Поступила 22.05.2015 г.