

УДК 621.31

П.Ю. Огеенко

(Україна, Дніпропетровськ, Національний горний університет)

ОРГАНИЗАЦИЯ ПЕРЕДАЧИ ИНФОРМАЦИИ В СИСТЕМЕ УПРАВЛЕНИЯ ЭЛЕКТРОСНАБЖЕНИЕМ ШАХТЫ

При разработке АСУТП для горной промышленности одной из нерешенных проблем является отсутствие систем передачи информации, предназначенных для работы в подземных условиях. Характерный объект, для которого есть необходимость разработки подобной системы, – это система энергоснабжения шахты. Проводя анализ технических схем электроснабжения, можно отметить, что они имеют древовидную структуру связей между подстанциями разных уровней: ГПП, ЦПП, РПП. Таким образом, первое требование, предъявляемое к системе передачи данных, – это возможность наиболее близко повторить топологию шахтной сети. Второе требование можно определить на основе анализа расстояний между объектами – протяженность некоторых ветвей сети достигает 3 км [1].

Система автоматизированного управления электроснабжением шахты в общем случае должна соответствовать приведенной на рисунке 1 структурной схеме.

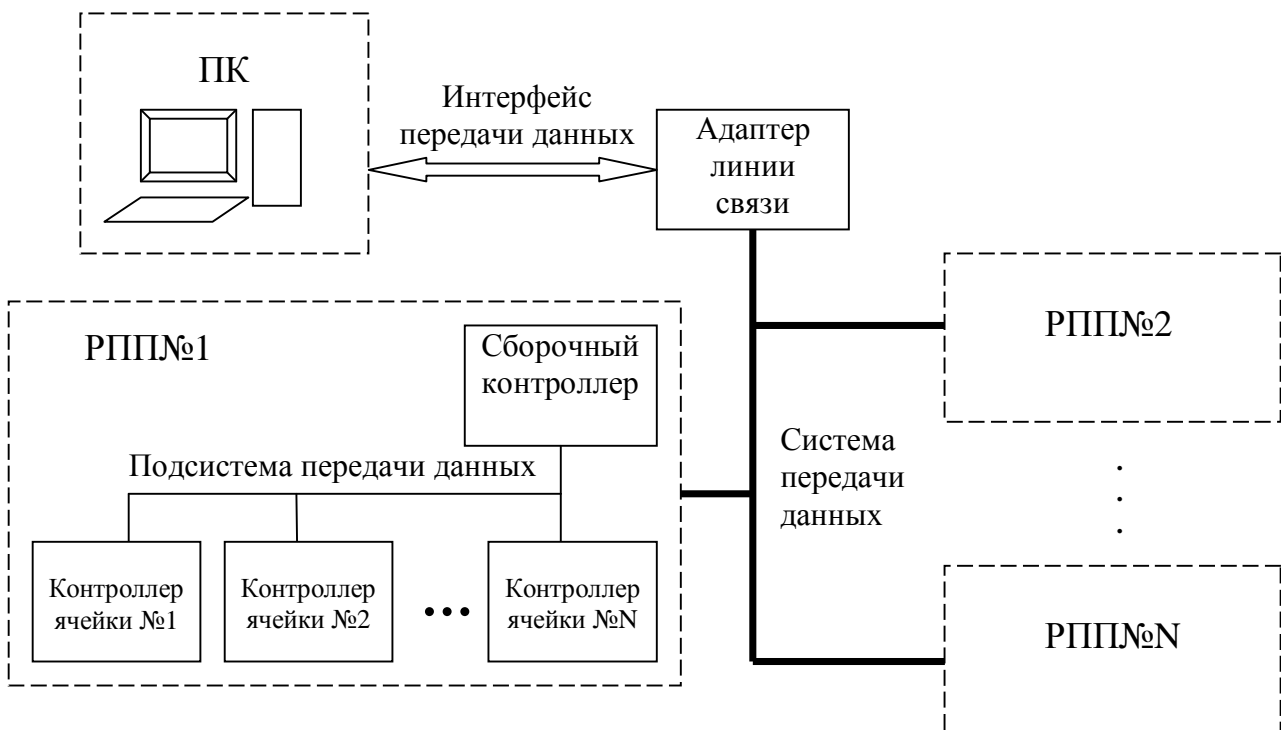


Рис. 1. Структурная схема проектируемой САУПЭШ

Связь с персональным компьютером (ПК) такой системы организована на основе использования специального устройства согласования или существующих и дополнительных возможностей внешних интерфейсов. Таким образом, адаптер линии связи может быть реализован как внешнее, так и как встроенное устройство. Связь между РПП, на которых будут установлены сборочные контроллеры, выполняющие функцию контроля всех устройств, находящихся на уровне ниже контроллеров ячеек, и обеспечивающие сбор и хранение информации о них, производится через общую систему передачи информации. Такая система служит для транспортных целей, выполняя задачу маршрутизации к указанным подсистемам (сегментам сети). Контроллеры ячеек подключены непосредственно к сборочному контроллеру и вместе с ним организуют сегмент сети. Сбор информации с них выполняется согласно выбранной дисциплине, например, циклического опроса в системе мастер-слэйв или обновления данных на сборочном контроллере путем передачи сообщения по событию при мульти-мастерной системе. Таким образом, к системе в целом предъявляются следующие требования:

- гибкость топологии – легкость изменения как самой структуры сети, так и добавления в нее новых сегментов и интеллектуальных устройств;
- высокая скорость обмена информацией, что обеспечит своевременную реакцию устройств системы на запросы и управляющие команды от пульта управления;
- большая пропускная способность канала линии связи, что позволит уменьшить возможность возникновения коллизий и обеспечит предпосылки для дальнейшего расширения функций системы в целом;
- адресация и (или) маршрутизация в сети – запросы сверху поступают только определенному сегменту, в котором находится запрашиваемое устройство, что уменьшит информационную загрузку сети;
- высокая помехозащищенность передачи информации – система должна гарантировать доставку данных в течение определенного интервала времени.

Сравнительный анализ имеющихся технологий для передачи данных [1] позволяет сделать вывод о том, что наиболее приемлемым решением для разработки подобных систем выступает полевая шина CAN. Она имеет высокую помехоустойчивость, обусловленную особенностью физического уровня, характеризуется устойчивостью против электрических и информационных перегрузок и имеет внутреннюю систему установки приоритетов. С точки зрения организации обмена информацией в CAN реализован механизм борьбы за канал [2]. К достоинствам можно отнести растущую популярность CAN-интерфейса в системах управления и, как следствие, выход в свет целых серий дешевых специализированных контроллеров, совместимых с ней. Единственной особенностью классической CAN-шины, ограничивающей ее применение для шахтной автоматизации, является отсутствие в сетевом и канальном уровнях возможности построения разветвленных сетевых решений. Однако это ограничение может быть снято при помощи дополнительных аппаратно-программных средств.

Главное отличие CAN-интерфейса от существовавших стандартов заключается в том, что передаваемое сообщение (кадр) не содержит уникальный адрес устройства приемника, а несет только идентификатор пакета, определяющий объект, которому предназначаются данные. Один и тот же пакет может быть одно-

временно прочитан и использован многими устройствами, включающими соответствующий объект.

Согласно требованиям к системе, можно утверждать, что реализовать ее на одном сегменте CAN-шины является практически невозможным из-за ограничений по длине. Таким образом, возникает необходимость осуществить формирование системы из множества сегментов. Подобное сегментное конструирование позволит сформировать необходимую древовидную топологию [1].

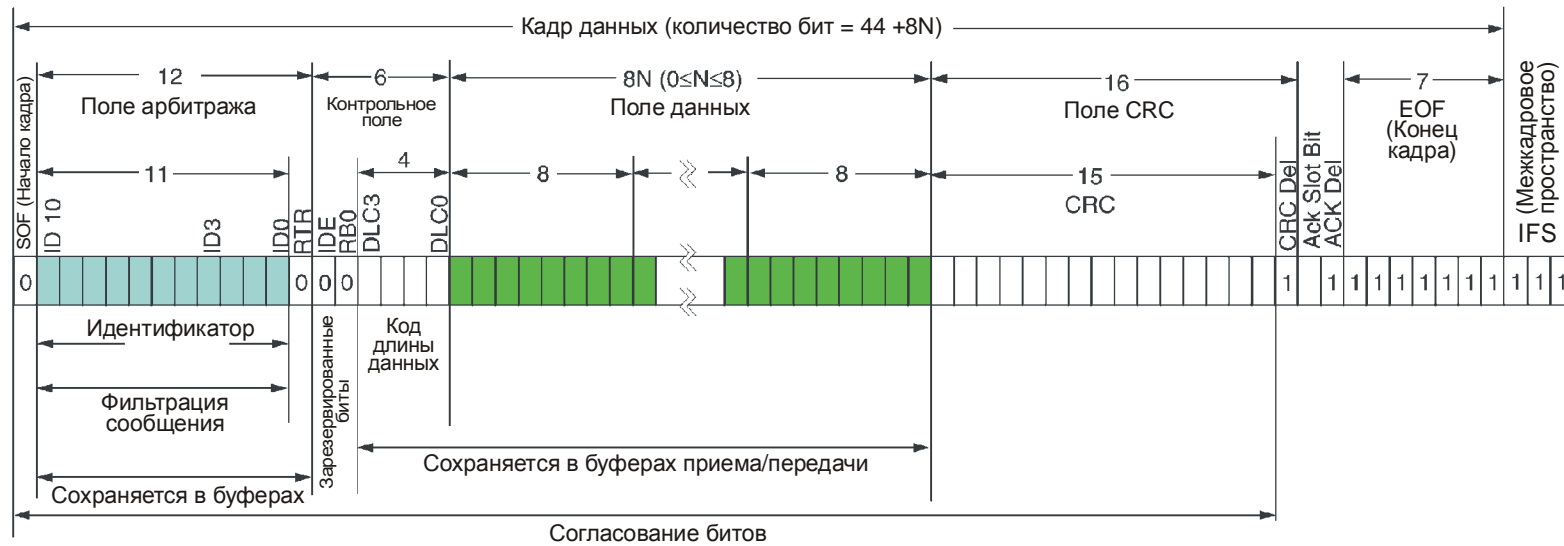
В основу топологии организованной таким образом системы принимаем древовидную, абстрактную структуру "бинарное дерево". Это – принцип организации множества объектов в единую систему при помощи связей между ними.

Выбор сегментированной CAN-сети и организации узлов в бинарное дерево позволяет организовать дисциплину маршрутизации посылок, т.е. сформировать транспортную сеть передачи данных, к которой можно будет легко выполнить подключение новых сегментов и устройств. Использование такой иерархии удобно тем, что точка входа в систему равноудалена от внешних сегментов (в идеальном случае). В такой организации легко организовать потоки данных с периферии в центр и наоборот.

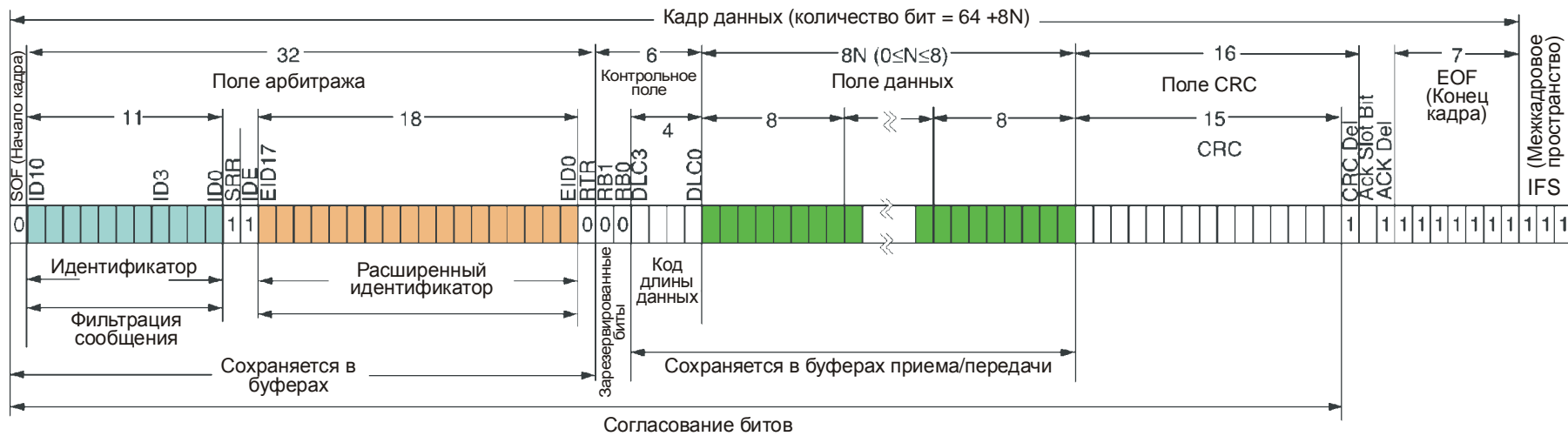
Транспортная сеть строится на использовании расширенного CAN-кадра, несущего в себе информацию о маршруте и полный стандартный кадр, который будет распакован при достижении указанного согласно маршруту сборочного контроллера и отправлен далее в соответствующий сегмент. Формат расширенного и стандартного кадров приведен на рис. 2.

Кадр данных согласно CAN содержит следующие поля:

1. Начало кадра Start of frame (1 бит): Кадр начинается старт-битом (SOF – Start of frame).
2. Поле управления доступом Arbitration Field (12 бит):
 - Одиннадцать бит идентификатора сообщения (ID10-ID0).
 - Бит RTR (Remote Transmission Request) – определяет сообщение как кадр данных (RTR=0 "dominant") или кадр запроса данных (RTR=1 "recessive").
3. Поле управления Control Field (6 бит):
 - Идентификатор расширенного кадра (IDE).
 - Резервный бит (RBO).
 - Четыре бита (DCL3...DCL0) – длины поля данных в байтах (N).
4. Поле данных Data Field (8*N бит): N=0,...,8.
5. Поле контрольной суммы CRC Field (16 бит):
 - 15 бит контрольной суммы.
 - Бит разделителя (CRC Del) по определению "recessive".
6. Поле подтверждения (2 бита):
 - ACKSlot. Во время передачи узел формирует "recessive" бит в поле ACKSlot. Любой узел, принявший кадр без ошибок, подтверждает корректность приема кадра установкой значения "dominant". Бит устанавливается при корректном приеме кадра вне зависимости от того, удовлетворяет он условиям фильтров или нет.
 - ACKDel. Разделительный бит по определению: сохраняет значение "recessive".



а)



б)

Рис. 2. Форматы CAN кадров: а – стандартного, б - расширенного

7. Поле окончания кадра End of Frame (7 бит) – состоит из семи разделительных "recessive" битов.

Для проверки описанного выше подхода организации топологии сети разработана приведенная на рисунке 3 структурная схема физической модели системы контроля параметров по CAN-шине (рис. 3).

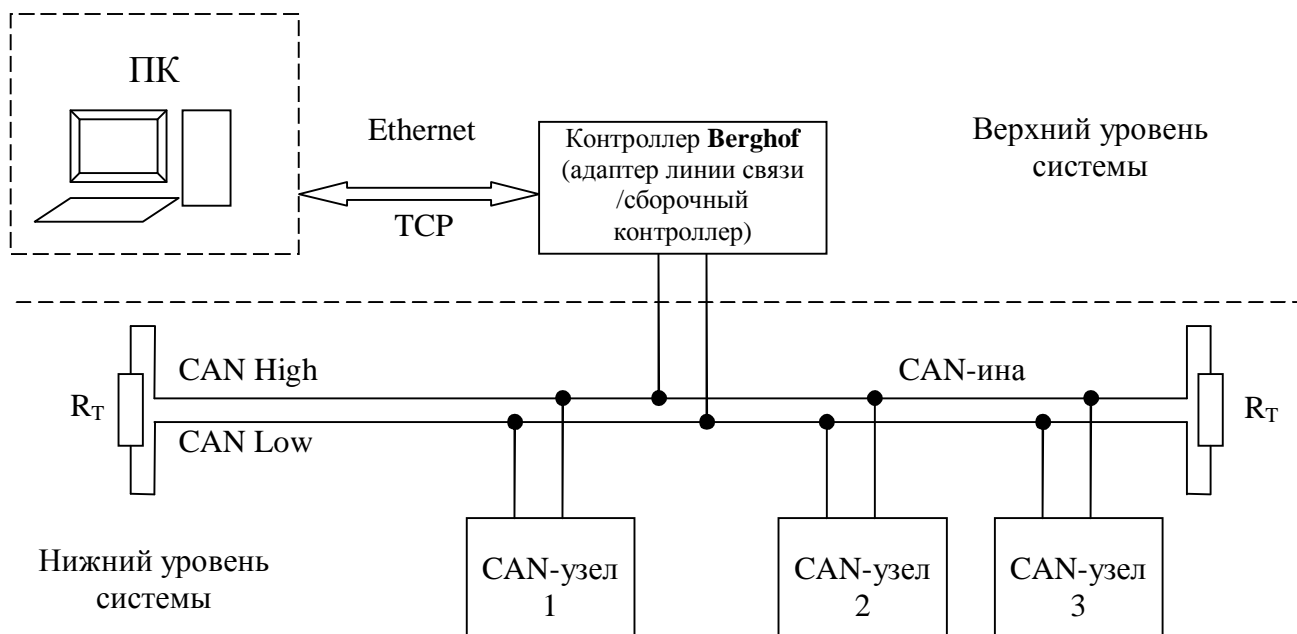


Рис. 3. Структурная схема модели для проведения экспериментов

В обмене информацией по CAN участвуют четыре CAN-узла: промышленный контроллер, в котором реализованы функции адаптера линии связи для подключения ПК и сборочного контроллера, и три однотипных CAN совместимых устройства, которые моделируют работу с ячейками определенного РПП. Взаимодействие между ПК и адаптером линии связи осуществляется посредством интерфейса Ethernet 100BaseT на основе протокола TCP.

Согласно данной структурной схеме на ПК запускается программное обеспечение, которое позволяет выполнить мониторинг информационных потоков в CAN-сети, аналогичной CAN-сегменту, приведенному на рисунке 1. Приложение пульт мониторинга с возможностью управления разработано специально для проведения экспериментов с физической моделью системы. Программное обеспечение написано в среде Visual C++ на базе SDI-приложения.

Функциональное назначение программы заключается в выполнении задач пульта управления или мониторинга системой в качестве TCP-клиента для реализованного на Berghof контроллере TCP сервера.

Приложение позволяет выполнить подключение через механизм WinSock по Ethernet к удаленному устройству, указав его уникальный в сети IP адрес и порт. При запуске клиент выполняет функцию мониторинга. Переход в режим управления выполняется через меню, команду "Control Request". При этом разрешение перехода в режим запрашивается у TCP сервера.

CAN-узлы, согласно структурной схеме рис. 1, соответствуют контроллерам ячеек, подключенным к CAN-сегменту. Это достигнуто путем разработки

полнофункциональных CAN-устройств на базе микроконтроллера производителя Microchip серии PIC18F458 со встроенным CAN-модулем. Принципиальная схема устройства приведена на рис. 4.

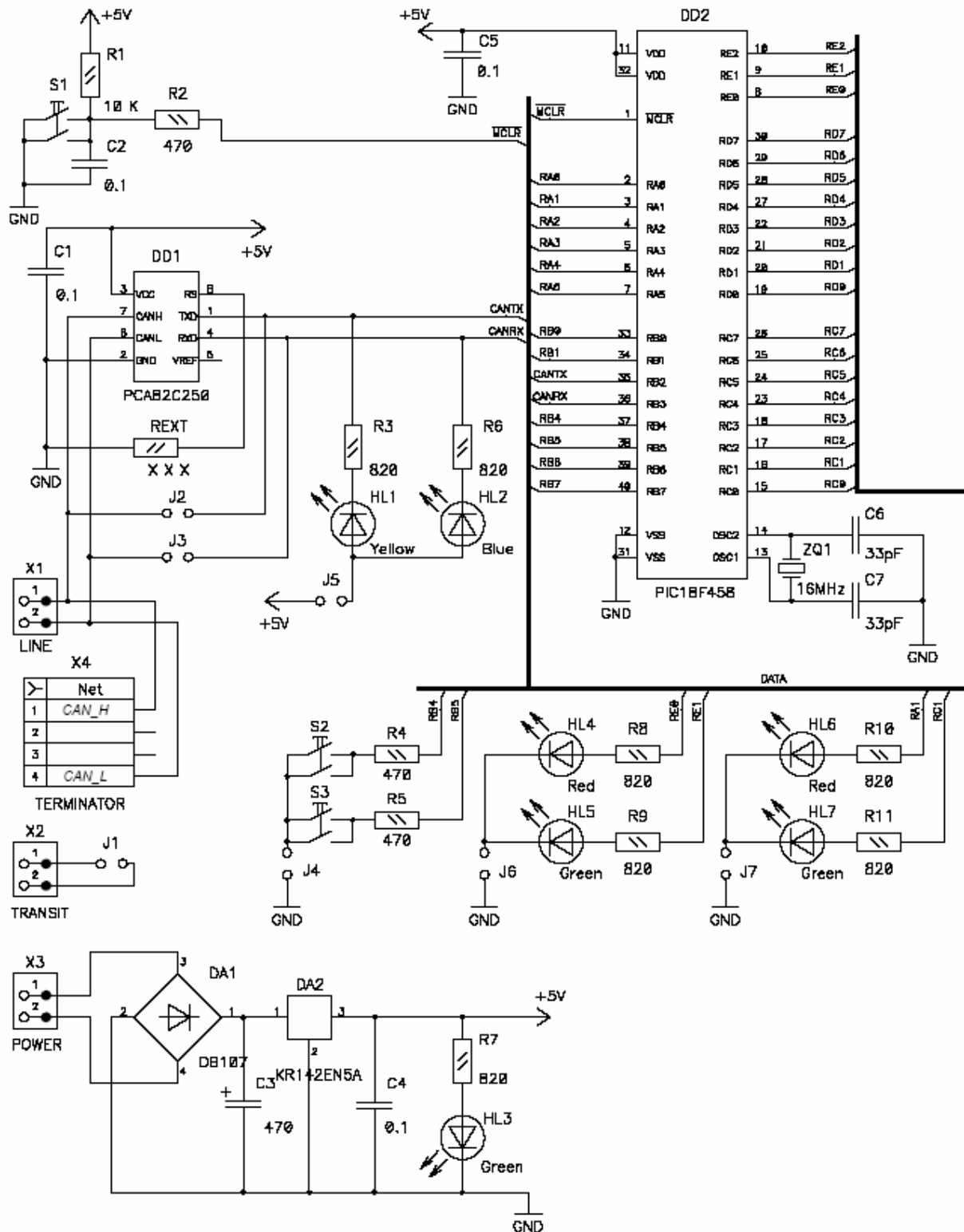


Рис. 4. Принципиальная схема CAN-узла

Основную задачу по выполнению интеллектуальных задач выполняет микроконтроллер PIC18F458 (DD2). Его тактирование производится от кварцевого резонатора ZQ1. Питание платы обеспечивается через схему, собранную

на микросхемах DA1 (диодный мост) и DA2 (крен KR142EN5A). Данная схема преобразует напряжение с 12 до 5 В ТТЛ логики, необходимых для питания встроенных модулей PIC18F458. CAN-линия подключается через разъем X1, соответственно 1 – CAN_H, 2 – CAN_L. Согласование сигналов, передаваемых с микроконтроллера PIC18F458 в CAN-линию, с уровнями CAN-шины осуществляется через формирователь уровней PCAB2C250 (DD1). Цепочка сброса схемы собрана на элементах R1, R2, C2 и кнопке S1. Кнопки S2 и S3 служат для инициации события на CAN-узле – в рамках системы в целом подобное событие может быть проассоциировано с получением данных от контроллера ячейки. Светодиоды HL1 и HL2 служат для индикации передачи/приема по CAN-шине. Для индикации режимов работы программного обеспечения узла используются светодиоды HL4-HL7.

В рамках экспериментальной модели CAN-узлы выполняют задачу эмуляции ячеек на РПП.

Работа программы CAN-узла представлена в виде схемы алгоритма на рис. 5.

При запуске устройства выполняется инициализация портов, таймеров и встроенного CAN-модуля (блоки 1 и 2). Затем происходит переход в рабочий цикл контроля событий (блок 3). Цикл опроса вечный, завершение работы может быть выполнено либо по команде от пульта управления (блоки 12 и 13), либо в случае аппаратного сброса устройства. Во время работы выполняется контроль двух событий: нажатие кнопки на устройстве и наличие данных в буфере приема по CAN (блоки 4 и 7). При замыкании контактов кнопкой программно при помощи встроенного CAN-модуля формируется стандартный кадр с уникальным идентификатором объекта события и производится его отправка в CAN-сегмент (блоки 5 и 6). Контроль получения сообщения по CAN выполняется через циклическую проверку заполнения буфера приема. Когда в буфере присутствуют данные, то происходит их сравнение с уникальными идентификаторами запроса "Эхо" и команды отключения (блоки 8 и 12). Если идентификаторы не соответствуют, то данные игнорируются, очищается приемный буфер и цикл переходит к следующей итерации (блок 14). В ином случае, при получении запроса "Эхо" выполняется извлечение данных из полученного кадра, формирование с ними кадра ответа с уникальным идентификатором объекта события и отправка его в CAN сегмент (блоки 9-11).

Для выполнения совмещенных функций адаптера линии связи и сборочного контроллера может быть использован широкий ряд промышленных контроллеров разных производителей. Основные требования к таким контроллерам: наличие совместимого с персональным компьютером интерфейса и CAN-канала, оперативной памяти, позволяющей запускать сложные программные приложения, а также легкость подключения устройства к системе. Для проведения эксперимента выбран промышленный контроллер немецкой фирмы Berghof, в котором реализованы интерфейсы RS232, CAN и Ethernet.

Разработка программного обеспечения для контроллера осуществлена в среде CP1131 (CoDeSys) с использованием технологических языков программирования стандарта IEC 61131-3 и соответствующих библиотек.

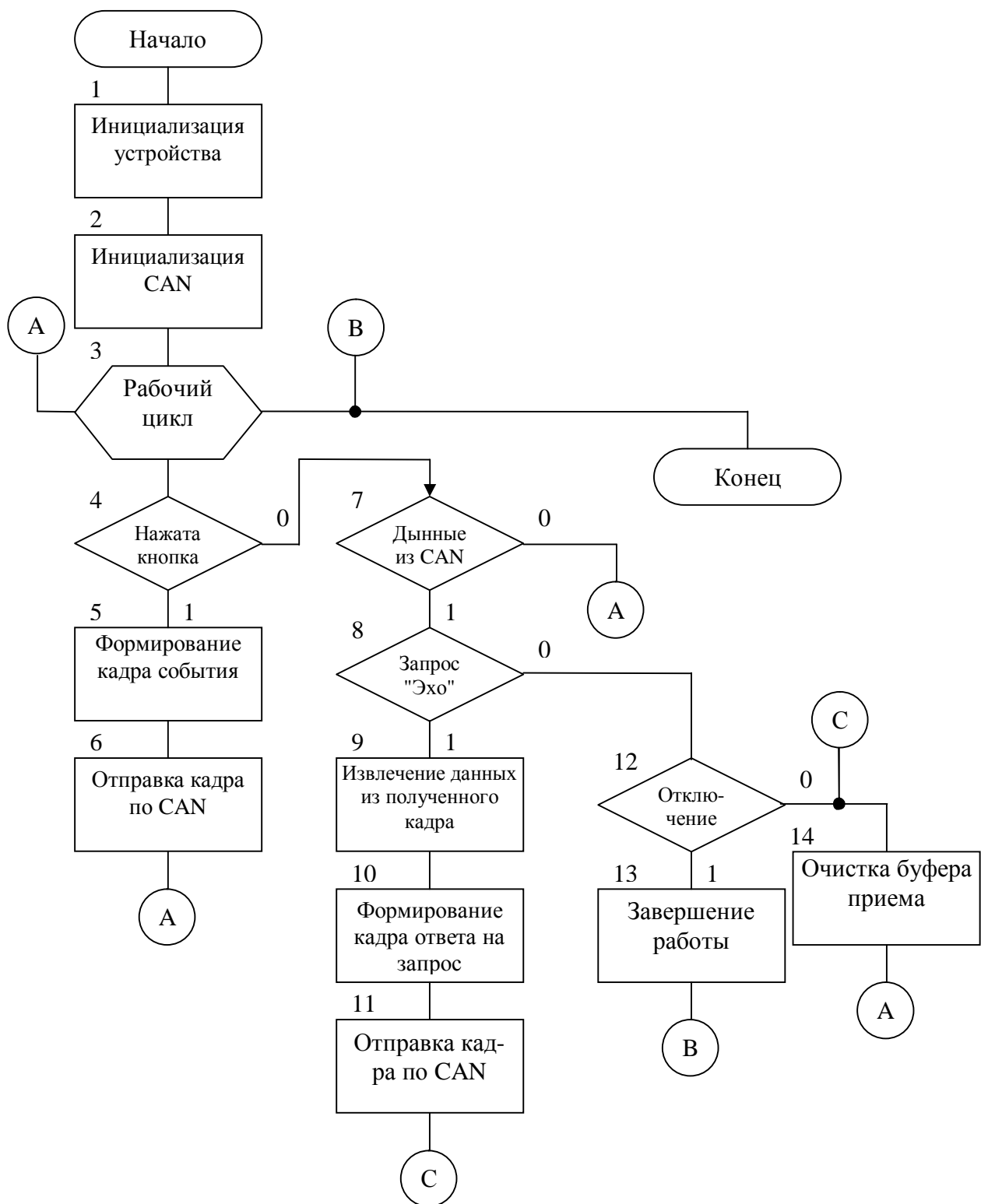


Рис. 5. Схема алгоритма принципа работы CAN-узла

Приложение представляет собой полноценный TCP-сервер с возможностью подключения до пяти клиентов и алгоритмом контроля информационных потоков в CAN-сегменте.

Программное обеспечение написано таким образом, чтобы не было необходимости фиксировать на контроллере Berghof, какой именно клиент находится в режиме управления.

Проведенный эксперимент был по смыслу разбит на следующие этапы:

1. Отображение информации о событии, произошедшем на CAN-узле, на интерфейсе пульта мониторинга;
2. Переход приложения клиента в режим управления и отправка широко-вещательного запроса "Эхо";
3. Подключение к системе нескольких клиентов, один из которых находится в режиме управления, и повторение этапов 1 и 2;
4. Отправка от клиента в режиме управления команды отключения устройств от линии;
5. Отправка с трех CAN-узлов блоков данных, состоящих из 1000 сообщений;
6. Подключение к разработанной модели пульта управления САУПЭШ через буферный файл и повторение этапов 1-5.

При проведении эксперимента доказана возможность контроля возникновения события на CAN-узле с пульта мониторинга. Система допускает контроль информационных потоков сразу с нескольких клиентов, выступающих в роли пультов мониторинга. Возможность управления системой обеспечивается лишь для одного приложения клиента. В режиме управления клиентское приложение способно производить опрос всех CAN-узлов через адаптер линии и осуществлять отключение устройств сети. При высокой загрузке канала первыми приходят кадры с меньшим значением идентификатора, т.е. обладающие более высоким приоритетом. При этом посылки от узлов с большими значениями идентификаторов не теряются, а доставляются после того, как были переданы все кадры с более высоким приоритетом. Также подтверждена возможность выполнения мониторинга и управления через разработанный пульт для САУПЭШ.

В целом, исследования доказали возможность реализации подсистемы передачи информации САУПЭШ на базе CAN-шины при использовании транспортной сети для маршрутизации сообщений к указанным сегментам.

Список литературы

1. Ткачев В.В. и др. Разработка системы передачи информации для подземных условий // Сб. науч. тр. Национального горного университета. – 2004. – № 19, т. 2. – С. 20-27.
2. Ткачев В.В. и др. Исследование возможности применения полевой шины CAN протокола CANopen для создания систем передачи информации в шахтных условиях // Сб. науч. тр. Национального горного университета. – 2004. – № 19, т. 2. – С. 50-59.