

Keywords: *moment characteristics of the distribution of traffic, traffic analyzer, the program Wireshar, the average waiting time in the queue.*

Тарасов Вениамин Николаевич, д.т.н., профессор, заведующий Кафедрой программного обеспечения и управления в технических системах (ПОУТС) Поволжского государственного университета телекоммуникаций и информатики (ПГУТИ). Тел. (8-846) 228-00-13, 8-960-827-22-33. E-mail: vt@ist.psati.ru.

Бахарева Надежда Федоровна, д.т.н., профессор, заведующая Кафедрой информатики и вычислительной техники ПГУТИ. Тел. (8-846) 228-00-59. E-mail: bahareva-nf@psuti.ru.

Горелов Глеб Александрович, ассистент Кафедры ПОУТС ПГУТИ. Тел. (8-846) 228-00-13, 8-927-721-89-79. E-mail: gleb_fox@bk.ru.

УДК 681.3

ОРГАНИЗАЦИЯ ОБМЕНА ИНФОРМАЦИЕЙ В ПРОМЫШЛЕННОЙ СЕТИ НА БАЗЕ МОДУЛЕЙ СЕРИИ ADAM-4000

Кангин В.В., Ложкин Л.Д.

Статья посвящена вопросам организации обмена информацией между компьютером-хостом и модулями удаленного и распределенного ввода-вывода серии ADAM-4000 (фирма Advantech, Тайвань) в компьютерной распределенной системе сбора данных и управления (КРССДУ). Хост и рабочие модули объединяются с помощью промышленной сети на базе интерфейса RS-485 с использованием протоколов ASCII или Modbus. Показано, что проблема обмена информацией между компьютером-хостом и рабочими модулями ввода-вывода серии ADAM-4000 КРССДУ на базе промышленной сети может быть решена средствами среды визуального программирования Delphi с использованием структур DCB и CommTimeOuts. Выполнен анализ ошибок, которые могут возникнуть при обмене информацией между хостом и рабочими модулями, проведена их классификация. На основании проведенных исследований разработано универсальное приложение, позволяющее изменять параметры настройки COM(USB)-порта и сетевой адрес модуля ADAM-4055.

Ключевые слова: промышленная сеть, хост, интерфейс RS-485, Modbus, ASCII, COM-порт, USB-порт, тайм-аут, DCB.

Введение. Постановка задачи

В настоящее время практически все системы управления технологическими процессами и оборудованием являются распределенными, то есть представляют набор локальных управляющих устройств, объединенных промышленной сетью. Локальные управляющие устройства, называемые также узлами сети, выполняют свои локальные подзадачи управления. Совокупность этих локальных подзадач образует одну большую задачу управления объектом или группой объектов.

Децентрализация задачи управления на ряд локальных подзадач с последующим построением на этой основе децентрализованных распределенных систем управления имеет два глобальных положительных эффекта:

- децентрализация глобальной задачи управления на ряд более простых подзадач приводит к упрощению программного кода управляющих программ, а следовательно, к их более простой отладке и тестированию;

- децентрализация системы управления на ряд подсистем позволяет приблизить управляющие устройства непосредственно к объектам управления, что снижает затраты на соединительные проводники и повышает помехоустойчивость системы управления.

В настоящее время наука и промышленность предлагают целый ряд вариантов компоновок КРССДУ. Однако все они основываются на двух концепциях:

- на основе модулей удаленного и распределенного ввода-вывода [1-4];

- на базе промышленных программируемых контроллеров [5-6].

Оба решения используют промышленную сеть для связи локальных управляющих устройств. На практике применяется смешанное решение, когда КРССДУ состоит и из модулей удаленного ввода-вывода, и из программируемых контроллеров.

Продукция фирмы Advantech широко используется в России под маркой ADAM.

Классификация оборудования марки ADAM на категории по функциональному назначению показана на рис. 1.



Рис. 1. Классификация оборудования марки ADAM

Имеется четыре категории оборудования марки ADAM (см. рис. 1):

- модули удаленного и распределенного ввода-вывода; к ним относятся модули серий: ADAM-4000, ADAM-5000, ADAM-4100, ADAM-6000;
- программируемые контроллеры серий ADAM-4500, ADAM-4501, ADAM-6500, ADAM-6501, ADAM-5510, ADAM-5550KW;
- коммуникационные модули серий ADAM-4500, ADAM-6500, предназначенные для решения коммуникационных задач при построении промышленных сетей;
- модули нормализации сигналов серии ADAM-3000 и модуль ADAM-4914V; эти модули являются вспомогательными и интеллектуальных функций не выполняют.

На рис. 2 приведен пример организации КРССДУ на базе модулей серий ADAM-4000 и ADAM-5000. Кроме рабочих модулей, в состав сети входят:

- Host PC – хост компьютер, который по сети опрашивает рабочие модули ввода информации, математически и логически обрабатывает принятую информацию и по сети передает воздействия в определенные модули вывода информации; в задачи хоста может входить визуализация на экране технологической информации, архивирование результатов, подготовка необходимых документов и т. д.;
- преобразователь интерфейса RS-232 в RS-485 типа ADAM-4520.

Промышленная сеть на рис. 2 показана штриховой линией: она содержит всего два проводника, число узлов в ней легко увеличивается и уменьшается. В качестве протокола обмена информацией может быть использован ASCII (American Standard Code for Information Interchange) – американский стандартный код для обмена информацией и Modbus/RTU.

Система сбора данных на базе RS-485



Рис. 2. КРССДУ на базе интерфейса RS-485

Следует отметить, что весь «интеллект» КРССДУ сосредоточен в одном месте – хосте. Модули удаленного и распределенного ввода-вывода предназначены лишь для приема информации с разнообразных датчиков, передачи информации хосту, приема управляющей информации от хоста и выдачи ее в исполнительные механизмы и устройства объекта управления.

Обмен информацией между модулями и хостом регламентируется хостом и осуществляется в режиме диалога: запрос-ответ. Запрос реализуется в виде команды, которая содержит адресную часть – адрес модуля, которому она предназначена. Даже выдача управляющей информации из хоста в модуль вывода по сети осуществляется в режиме запрос-ответ. Таким образом, процесс управления в КРССДУ такого типа сводится, по сути, к формированию хостом команд, отправлению их в промышленную сеть и приему ответов от адресуемых модулей.

В ОС Windows с портами компьютера (хоста) работают точно так же, как и обычными файлами: запись информации в файл, чтение информации из файла. Однако имеются и некоторые отличия, связанные с тем, что порт представляет собой физическое устройство и имеет ряд особенностей. Целью данной статьи является анализ программных средств среды Delphi, позволяющих работать с коммуникационными портами хоста и применение их на примере изменения сетевого адреса модуля ADAM-4055.

Открытие порта

Работа с портом начинается с его открытия. Для этого используется функция CreateFile, которая объявляется следующим образом:

```
function CreateFile (lpFileName: PChar;
dwDesiredAccess: DWORD;
dwShareMode: DWORD;
lpSecurityAttributes:
```

```
LPSECURITY_ATTRIBUTES;
dwCreationDistribution: DWORD;
dwFlagsAndAttributes: DWORD;
hTemplateFile: THandle); THandle;
```

Эта функция имеет семь параметров. Рассмотрим их.

Параметр `lpFileName` типа `PChar` представляет собой указатель на строку-имя COM-порта, записанное, например, в виде `'\\.\COM5'`.

Параметр `dwDesiredAccess` типа `DWORD` определяет тип доступа к порту: только запись, только чтение, чтение и запись-`GENERIC_READ` or `GENERIC_WRITE`.

Параметр `dwShareMode` типа `DWORD` – параметр совместного доступа. Для коммуникационных портов он равен 0.

Параметр `lpSecurityAttributes` типа `LPSECURITY_ATTRIBUTES`. Для коммуникационных портов он равен `nil`.

Параметр `dwCreationDistribution` типа `DWORD` – режим автосоздания. Для коммуникационных портов он равен `OPEN_EXISTING`.

Параметр `dwFlagsAndAttributes` типа `DWORD` – атрибут режима обработки. Для коммуникационных портов, работающих в асинхронном режиме, он равен `FILE_FLAG_OVERLAPPED`.

Параметр `hTemplateFile` типа `THandle` – описатель файла – шаблона. Для коммуникационных портов он равен 0.

При успешном открытии порта функция возвращает его описатель – дескриптор типа `THandle`, который необходим для вызова других функций работы с портом. В случае ошибки функция возвращает значение `INVALID_HANDLE_VALUE`.

Настройка параметров порта

Существует несколько способов настройки порта для работы. Одним из самых простых и наглядных является способ настройки структуры `DCB`. Структурой `DCB` описываются основные параметры последовательного порта. Она содержит несколько полей, каждое из которых соответствует определенному параметру порта:

- `DCB.BaudRate` – скорость передачи данных. Возможно указание констант – `CBR_100`, `CBR_300`, `CBR_600`, `CBR_1200`, ..., `CBR_256000`;
- `DCB.Parity` – метод контроля четности. Может содержать одно из следующих значений: `EVENPARITY`, `MARKPARITY`, `NOPARITY`, `ODDPARITY`, `SPACEPARITY`;
- `DCB.ByteSize` – число информационных битов в передаваемых и принимаемых байтах.

- `DCB.StopBits` – количество стоповых бит. Может быть `ONESTOPBIT`, `ONE5STOPBIT`, `TWOSTOPBIT`.

Существуют две функции для работы со структурой `DCB`:

- чтение текущих настроек порта:
function `GetCommState` (`hPort: THandle; DCB: TDCB`): `boolean`;
- установка новых настроек порта:
function `SetCommState` (`hPort: THandle; DCB: TDCB`): `boolean`;

Здесь:

- `hPort: THandle` – описатель открытого функцией `CreateFile` порта,
- `DCB: TDCB` – указатель на структуру `DCB`.

Настройка тайм-аутов порта

Под тайм-аутом понимается время ожидания выполнения операций чтения или записи информации. В нашем случае речь идет о последовательном порте ввода-вывода информации. Для установки значений тайм-аутов используется структура `COMMTIMEOUTS`. Описание структуры `COMMTIMEOUTS` на языке `Delphi` имеет вид:

```
TCommTimeouts = ^ TCommTimeouts;
_COMMTIMEOUTS = record
  ReadIntervalTimeout: DWORD;
  ReadTotalTimeoutMultiplier: DWORD;
  ReadTotalTimeoutConstant: DWORD;
  WriteTotalTimeoutMultiplier: DWORD;
  WriteTotalTimeoutConstant: DWORD;
end;
```

```
TCommTimeouts = _COMMTIMEOUTS;
```

Структура `COMMTIMEOUTS` имеет следующие поля.

- `ReadIntervalTimeout` – задает допустимый интервал ожидания в миллисекундах между получением двух символов.
- `ReadTotalTimeoutMultiplier` – задает множитель в миллисекундах, используемый для вычисления полного времени ожидания операции чтения. Для каждой операции чтения это значение умножается на количество запрошенных байтов.
- `ReadTotalTimeoutConstant` – задает константу, добавляемую к произведению, полученному при вычислении времени из `ReadTotalTimeoutMultiplier`.

Поля `WriteTotalTimeoutMultiplier` и `WriteTotalTimeoutConstant` аналогичны полям `ReadTotalTimeoutMultiplier` и `ReadTotalTimeoutConstant` соответственно, но служат для передачи. Задание нулевого значения всех констант

ReadXXX указывает на отсутствие тайм-аутов чтения, а всех WriteXXX – отключение тайм-аутов записи.

Для получения текущих значений тайм-аутов порта используется функция GetCommTimeouts, а для их установки функция SetCommTimeouts. Эти функции объявляются следующим образом:

```
- function GetCommTimeouts (hPort: THandle;
IpCommTimeouts:TCommTimeouts): boolean;
- function SetCommTimeouts (hPort: THandle;
IpCommTimeouts: TCommTimeouts): boolean;
```

Здесь:

- hPort: THandle – описатель-дескриптор открытого порта;
- IpCommTimeouts: TCommTimeouts – указатель на структуру TCommTimeouts.

Настройки порта желательно производить сразу после его открытия.

Чтение данных из порта

Синхронное или асинхронное чтение данных из порта выполняется с помощью функции ReadFile. Формат заголовка ReadFile на языке Delphi имеет вид:

```
function ReadFile (
    hPort: THandle;
    lpBuff: pointer;
    lpCnt: DWORD;
    var lpCntByte: DWORD;
    lpOver: Poverlapped
): BOOL;
```

Здесь:

- hPort: THandle – описатель – дескриптор открытого с помощью функции CreateFile порта;
- lpBuff: pointer – указатель на буфер для записи принятых байт;
- lpCnt: DWORD – размер буфера в байтах, то есть число байт, ожидаемых к приему или предназначенных для передачи;
- var lpCntByte: DWORD – число фактически принятых или переданных байт;
- lpOver: POverlapped – адрес структуры OVERLAPPED, используемой для асинхронного чтения или записи.

Функция завершается, если прочитано необходимое число байт или произошла ошибка. Если чтение произошло успешно, то функция возвращает ненулевое значение. Перед чтением необходимо установить тайм-ауты ожидания с помощью функции Set-CommTimeouts.

Передача данных в порт

Синхронная или асинхронная передача информации в порт выполняется функцией WriteFile.

Формат заголовка ReadFile на языке Delphi имеет вид:

```
function WriteFile (
    hPort: THandle;
    lpBuff: pointer;
    lpCnt: DWORD;
    var lpCntByte: DWORD;
    lpOver: Poverlapped
): BOOL.
```

Параметры этой функции и их назначение точно такие, что и у функции ReadFile. В случае удачного завершения работы функции переменная lpCntByte содержит количество байт, записанных в порт из lpBuff, а сама функция принимает ненулевое значение.

Проект. Изменение сетевого адреса модуля

Целесообразно сконцентрировать внимание на таких важных вопросах, как открытие и настройка порта, задание нужного сетевого адреса модуля. Предполагается, что фрагмент сети, содержащий модуль конвертора ADAM-4561 и модуль ввода-вывода ADAM-4055, собран и подключен к порту COM или USB хоста. Кроме этого, на модуль ADAM-4055 подано рабочее напряжение +10 В...+30 В.

Анализ ошибок, возникающих при работе с портами

При работе приложения (программы) могут возникнуть ошибки, которые не позволят сети нормально функционировать. Первый тип ошибок связан с невозможностью открыть COM-порт. Другие ошибки могут быть такими:

- не был запущен драйвер работы с портами, о котором говорилось в предыдущем разделе,
- неверно указан номер порта или сеть подключена к другому порту,
- имеются проблемы в работе аппаратуры COM-порта, сетевого кабеля и т.д.

Второй тип ошибок связан с невозможностью нормальной работы модуля ADAM-4055. Примеры таких ошибок: модуль не настроен; модуль не подключен к сети или неисправен. На рис. 3 приведен внешний вид диалогового окна, формируемого приложением.

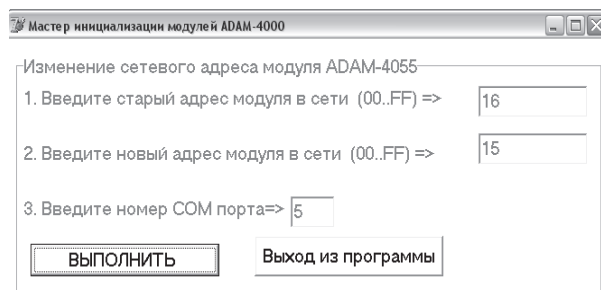


Рис. 3. Внешний вид диалогового окна

Необходимо ввести старый, новый адрес модуля в сети и щелкнуть по кнопке «ВЫПОЛНИТЬ». Если не появилось сообщений об ошибках, можно закрывать приложение, щелкнув по кнопке «Выход из программы». Модуль получает новый сетевой адрес; адреса необходимо записывать в шестнадцатеричной системе счисления – 00...FF.

Рассмотрим процедуру, которая выполняет настройки порта, тайм-аутов, открывает порт. Она выполняется при щелчке по кнопке «Выполнить»:

```

procedure TForm1.Button4Click(Sender:
TObject);
begin
//номер порта возьмем из окна Edit2.
PortString := '\\.\COM'+Edit2.Text ;
//пытаемся открыть порт
CommHandle:=CreateFile(PChar(PortString),
GENERIC_READ or GENERIC_WRITE,
0, nil, OPEN_EXISTING, 0, 0);
if (CommHandle = INVALID_HANDLE_
VALUE) then
begin //возникли проблемы с открытием порта
MessageDlg('Не могу открыть порт', mtError,
[mbOK], 0);
end
else
begin//проблем с портом нет, он открыт
//читаем старые настройки DCB
GetCommState(CommHandle, DCB);
//подготавливаем новые настройки DCB
// скорость обмена 9600 бод
DCB.BaudRate:= CBR_9600;
// бит четности не используется
DCB.Parity:= NOPARITY;
// обмен идет байтами (8 бит)
DCB.ByteSize:= 8;
// стоп-бит не используется
DCB.StopBits:= OneStopBit;
//готовим новые настройки CommTimeOuts
CommTimeOuts.ReadIntervalTimeout:= 0;
CommTimeOuts.ReadTotalTimeoutMultiplier:= 0;
CommTimeOuts.ReadTotalTimeoutConstant:=
1000; // ждем 1 сек
CommTimeOuts.WriteTotalTimeoutMultiplier := 0;
CommTimeOuts.WriteTotalTimeoutConstant :=
1000; // ждем 1 сек
//устанавливаем новые настройки DCB и
//CommTimeOuts
SetCommTimeouts(CommHandle,
CommTimeOuts);
SetCommState(CommHandle, DCB);
// подготавливаем команду в модуль; в конце –//коды
13 и 10 символов возврата каретки и // перевода строки

```

```

Buf:='%'+Edit1.Text+Edit3.Text+'400600'+
#13+#10;
// отправляем команду через порт в модуль
WriteFile (CommHandle,PChar(Buf)^,
Length(Buf), NumberWritten, nil);
Sleep(20); // ждем 20 миллисекунд
// читаем из модуля ответ в буфер ответа
ReadFile(CommHandle, BinBuf, 3,
NumberWritten, nil);
if BinBuf[0]<>'!' then
begin
//пришел плохой ответ, выдаем сообщение об
//ошибке и закрываем приложение
MessageDlg('Ответа нет! Начнем сначала',
mtError, [mbOK], 0);
Application.Terminate;
end;
end;
end;

```

Выводы

1. Показано, что проблема обмена информацией между компьютером-хостом и рабочими модулями ввода-вывода серии ADAM-4000 КРССДУ на базе промышленной сети может быть решена средствами среды визуального программирования Delphi с использованием структуры DCB и CommTimeOuts.

2. Выполнен анализ ошибок, которые могут возникнуть при обмене информацией между хостом и рабочими модулями, проведена их классификация.

3. На основании проведенных исследований разработано универсальное приложение, позволяющее изменять параметры настройки COM(USB)-порта и сетевой адрес модуля ADAM-4055.

Литература

1. Кангин В.В., Ложкин Л.Д., Ямолдинов Д.Н. Обмен информацией в промышленной сети PlcNet // ИКТ. Т.8, №3, 2010. – С.49-54.
2. Кангин В.В., Ложкин Л.Д., Ямолдинов Д.Н. Программная реализация межсетевого шлюза сетей ETHERNET и PLCNET// ИКТ. Т.9, №2, 2011. – С.36-41.
3. Кангин В.В. Промышленные контроллеры в системах автоматизации технологических процессов: учебное пособие. Старый Оскол: Изд-во ТНТ, 2013. – 408 с.
4. Кангин В.В., Козлов В.Н. Аппаратные и программные средства систем управления. Промышленные сети и контроллеры. М.: Изд-во БИНОМ. Лаборатория знаний. 2010. – 418 с.

5. Кангин В.В. Средства автоматизации и управления. Аппаратные и программные решения: учебное пособие. Старый Оскол: Изд-во ТНТ, 2014. – 520 с.
6. Кангин В.В. Персональные компьютеры в системах автоматизации технологических процессов. М.: Изд-во Книга по требованию, 2013. – 224 с.

ORGANIZATION OF THE EXCHANGE BY INFORMATION IN INDUSTRIAL NETWORK ON BASE OF THE MODULES TO SERIESES ADAM-4000

Kangin V.V., Lozhkin L.D.

The Article is dedicated to questions to organizations of the exchange by information between host-computer and module removed and portioned entering-conclusion to serieses ADAM-4000 (the company Advantech, Taiwan) in computer portioned system data acceptance and management (KRSSDU). Host and worker to modules unite by means of industrial network on the base of the interface RS-485 with use protocol ASCII or Modbus. It Is Shown that problem of the exchange by information between host-computer and workers module entering-conclusion to serieses ADAM-4000 KRSSDU on the base of the industrial network can be solved facility of the ambience of the visual programming Delphi with use the structures DCB and CommTimeOuts. The Executed analysis mistake, which can appear when changing by information between host and workers module, is organized their categorization. On the grounds of called on studies is designed universal exhibit, allowing change the parameters of the adjustment COM (USB)-port and network address of the module ADAM-4055.

Keywords: industrial network, host, RS-485, Modbus, ASCII, COM-port, USB-port, timeout, DCB.

Кангин Владимир Венедиктович, д.т.н., профессор Кафедры информационных технологий Российского экономического университета им. Г.В. Плеханова. Тел. 8-906-352-11-91; (8-831) 472-52-24; E-mail: king202008@yandex.ru

Ложкин Леонид Дидимович, к.т.н., программист Кафедры основ конструирования и технологии радио- и телекоммуникационных систем Поволжского государственного университета телекоммуникаций и информатики. Тел. (8-846) 310-69-93. E-mail: leon.lozhkin@yandex.ru

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 535.012.2

ФОРМИРОВАНИЕ ИЗОБРАЖЕНИЙ МОДОВЫХ СВЕТОВЫХ ПУЧКОВ ПРИ РАСПРОСТРАНЕНИИ В КРИСТАЛЛИЧЕСКИХ СТРУКТУРАХ

Кораблин М.А., Пономарев М.Ю.

Рассматривается прохождение модовых световых пучков через пластины кристаллического кварца. Экспериментальные исследования показывают различия формируемых распределений интенсивности в выходной плоскости и выявляют изменения в симметричной структуре пучков в зависимости от ориентации пластины кристалла. Предлагаемый эксперимент позволяет увидеть преимущества структурных пучков Гаусса-Лагерра и Бесселевых пучков высоких порядков перед пучками Бесселя низких порядков при прохождении через кристалл. Результаты работы существенны для исследования механизмов распространения световых пучков в природных кристаллах, разработки новых методик определения оптических параметров кристаллического кварца, а также теоретических основ передачи оптических сигналов в различных средах

Ключевые слова: оптические параметры, кристаллический кварц, структурные световые пучки.

Введение

Исследования в области получения изображений распределения интенсивности когерентного излучения при распространении в различных средах находят практическое применение при решении задач анализа неизвестных оптических свойств анизотропных материалов (диэлектрической проницаемости) по спектрофотометрическим данным.

Структурные световые пучки, характеризующиеся определенными модами лазерного излучения [1], находят применение в разнообразных областях благодаря своим особым свойствам. Одни типы мод