УДК 681.3

ОБМЕН ИНФОРМАЦИЕЙ В ПРОМЫШЛЕННОЙ СЕТИ PLCNET

Кангин В.В., Ложкин Л.Д., Ямолдинов Д.Н.

Статья посвящена вопросам организации обмена информацией между уровнями распределенной системы управления (РСУ), построенной на основе промышленной сети PlcNet. Показано, что проблема обмена информацией по промышленной сети является центральной при проектировании SCADA-систем. Решение этой проблемы лежит в плоскости организации клиент-серверных отношений между SCADA-системой и OPC-сервером, поставляемым разработчиком сетевой аппаратуры. В этом случае проблемы обмена сводятся к программному информационному обмену между SCADA-системой, выполняющей роль OPC-клиента, и OPC-сервером (выбран Fastwel PlcNet OPC Server). Разработан алгоритм просмотрщика тегов, позволяющего визуализировать значения 32 тегов, принятых по сети из контроллеров нижнего уровня. Разработаны алгоритмы процедур и функций, обеспечивающих следующие возможности просмотрщика тегов: запуск OPC-сервера Fastwel PlcNet OPC Server, добавление в него имени сегмента и имен тегов, определение типа тегов и вывода значений тегов в окна на форме просмотрщика тегов, остановка OPC-сервера.

Ключевые слова: распределенная система управления, промышленная сеть, SCADA-система, клиент-серверные отношения, OPC-клиент, OPC-сервер, тег.

В настоящее время наблюдается качественный скачок в развитии систем управления технологическим оборудованием и технологическими процессами. Системы управления стали децентрализованными, распределенными. Таким образом, в настоящее время можно говорить о распределенных системах управления.

Промышленные сети представляют собой физическую среду, на базе которой реализуются РСУ. На нижнем уровне этой структуры находятся контроллеры – устройства, осуществляющие непосредственное управление технологическими процессами в реальном времени. Контроллер должен решать задачу управления быстро, в темпе протекания технологического процесса.

На верхнем уровне РСУ расположены рабочие станции, позволяющие выполнять задачи:

- сбор данных о параметрах технологического процесса, их обработка и архивирование принятой информации;
- визуализация информации о ходе технологического процесса, сигнализация о предаварийных и аварийных ситуациях;
- формирование сводок, журналов, отчетов и т.д.;
- формирование команд оператора по изменению параметров настройки и режимов работы контроллеров;
- автоматическое управление ходом технологического процесса (только в ряде случаев и для решения задач невысокого быстродействия).

Рабочие станции верхнего уровня представляют собой автоматизированные рабочие места (АРМ) производственного персонала, обслуживающего технологический процесс: операторы, технологи, наладчики, диспетчеры и т.д. Верхний уровень РСУ называют диспетчерским, или операторским уровнем. Диспетчерский уровень представлен персональными компьютерами (ПК), рабочими станциями, промышленными компьютерами и т.д.

Контроллеры, решая задачу непосредственного управления технологическим оборудованием, принимают сигналы с датчиков и выдают управляющие сигналы на исполнительные механизмы. Одновременно с этим они решают еще одну важную задачу – передают информацию о состоянии технологического процесса на компьютеры верхнего диспетчерского уровня. В ряде случаев контроллеры могут обмениваться данными и между собой. Промышленная сеть и есть та физическая среда, по которой происходит обмен информацией между активными компонентами (узлами) РСУ.

Каждый технологический параметр представлен в сети в виде некоторой переменной – тега, который характеризуется величиной, качеством передачи (приема) по сети, типом и т.д.

Выделение в РСУ двух уровней позволяет говорить о программном обеспечении (ПО) для верхнего и для нижнего уровня. На нижнем уровне широко используются специализированные языки программирования по стандарту IEC 61131-3. К таким языкам, принадлежащим к группе языков Functional Block Diagrams (FBD) – функциональных блоковых диаграмм, относится язык, входящий в систему программирования контроллеров UltraLogik. Этот язык позволяет легко выполнить проектирование управляющих программ высокой степени сложности. Какие бы алгоритмы ни реализовала управляющая программа контроллера, важно, чтобы она еще передавала определенную нужную информацию в сеть. В UltraLogik это делается просто. Достаточно той или иной переменной (параметр технологичес-

кого процесса, состояние того или иного датчика, управляющий сигнал и т.д.) присвоить статус «сетевая», как она гарантированно будет периодически передаваться контроллером в сеть.

ПО верхнего уровня реализуется в виде SCADA-системы (Supervisory Control and Data Acquisition – система сбора данных и оперативного диспетчерского управления). Задачи, которые выполняют SCADA-системы, были обозначены выше, но центральной является задача по приему тегов из сети. Для этих целей используются специальные приложения (программы), называемые OPC-серверами. OPC-сервер считывает значения тегов из сети и передает их приложению – клиенту. SCADA-системы являются клиентами OPC-серверов [1].

Высокая стоимость существующих SCADA-систем, отсутствие качественной технической документации, избыточность, за которую приходится платить пользователю, делают процесс внедрения SCADA-систем в производство весьма непростым делом. Однако проектирование SCADA-систем можно выполнить с помощью таких программных средств, как Delphi, Visual C++, VBA. Они имеют в своем составе все необходимые средства для работы с OPC-серверами, графикой, мультипликацией, базами данных и т.д.

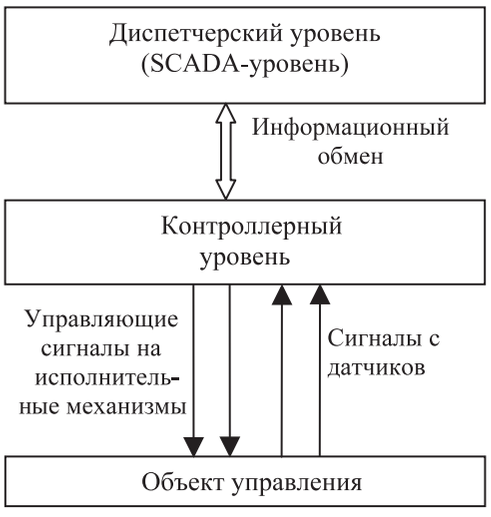


Рис. 1. Многоуровневая модель современной РСУ

Многоуровневая модель современной РСУ приведена на рис. 1. Ей можно поставить в соответствие модель сетевого обслуживания диспетчерского и контроллерного уровней РСУ (рис. 2).

Контроллерный уровень обслуживается одной из промышленных сетей на основе протоколов CAN, PLCNET, PROFIBUS и других. Верхний, диспетчерский уровень, как правило, обслуживается локальной сетью на основе протоколов Ethernet, TokenRing, ARCNet и других.

Один или несколько ПК в такой модели должны выполнять роль шлюза, то есть обеспечивать передачу информации из сети одного типа в сеть другого типа.

На рис. 3 приведена еще более подробная реализация РСУ.

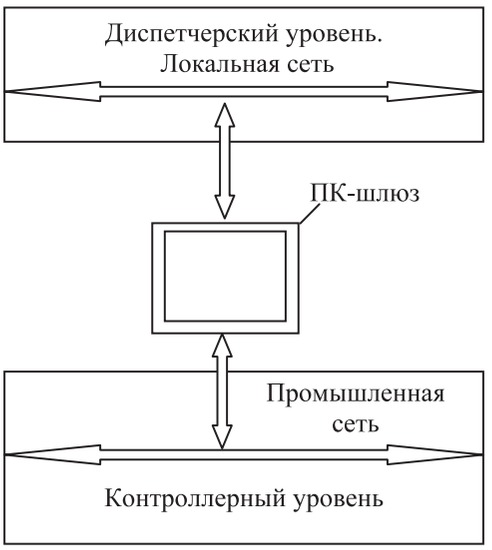


Рис. 2. Модель сетевого обслуживания диспетчерского и контроллерного уровней РСУ

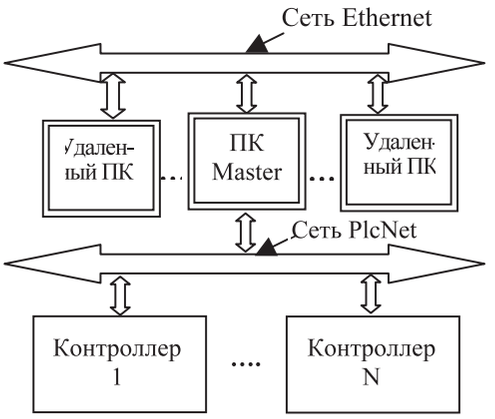


Рис. 3. РСУ на основе промышленной сети PlcNet

Нижний, контроллерный уровень объединяет N контроллеров с помощью промышленной сети PlcNet. Верхний диспетчерский уровень объединяет ряд ПК с помощью сети Ethernet. Среди этих ПК один должен выполнять роль мастера (Master) для сети PlcNet, то есть он должен иметь возможность принимать теги из сети PlcNet от контроллеров.

Для сети типа PlcNet в качестве контроллеров можно использовать PC-base контроллеры ADAM-5510. Сеть PlcNet организована на базе интерфейса RS-485 и реализуется в виде сегментов. Каждый из сегментов связан с одним из COM-портов ПК-мас-

тера. Для преобразования сигналов интерфейса RS-232 в сигналы интерфейса RS–485 используется преобразователь интерфейсов ADAM-4520.

Максимальное количество контроллеров, подключенных к одному сегменту сети, равно 32. ПК-мастер принимает информацию с контроллеров по сети PlcNet и затем может ее передавать по сети Ethernet удаленным ПК, образующим диспетчерский уровень.

На рис. 4 показано взаимодействие программного обеспечения компьютеров диспетчерского уровня в части передачи информации, поступившей от контроллера в ПК-мастер по сети PlcNet.

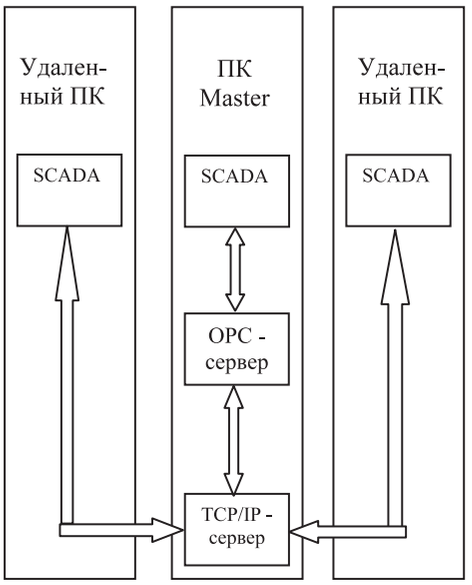


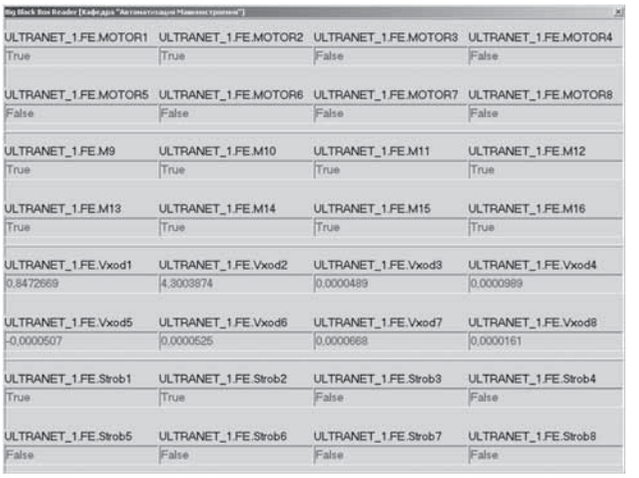
Рис. 4. Взаимодействие программного обеспечения компьютеров диспетчерского уровня

Прием этой информации из сети выполняется с помощью специальной программы OPC-сервера.

SCADA-система, расположенная на ПК-мастере, взаимодействует с OPC-сервером напрямую. SCADA-системы, расположенные на других удаленных ПК сети Ethernet, получают информацию из OPC-сервера через другую программу (TCP/IP-сервер), которая должна располагаться на ПК-мастере. Для сети PlcNet широко используется OPC-сервер фирмы Fastwel-Fastwel PLCNet OPC Server. Этот сервер позволяет не только читать информацию из контроллеров, но и передавать ее в обратном направлении – из SCADA-системы в контроллеры РСУ.

Рассмотрим более подробно процесс приема информации из контроллеров РСУ в SCADA-систему ПК-мастера. Для этого SCADA-систему реализуем в виде просмотрщика тегов – переменных сети PlcNet [2]. Внешний вид рабочего окна просмотрщика тегов показан на рис. 5.

Алгоритм работы просмотрщика тегов приведен на рис 6. На рис. 7 приведен алгоритм функции GetItemType, которая возвращает тип тега по его значению ItemValue.



ULTRANET_1.FE.MOTOR1	ULTRANET_1.FE.MOTOR2	ULTRANET_1.FE.MOTOR3	ULTRANET_1.FE.MOTOR4
True	True	False	False
ULTRANET_1.FE.MOTOR5	ULTRANET_1.FE.MOTOR6	ULTRANET_1.FE.MOTOR7	ULTRANET_1.FE.MOTOR8
False	False	False	False
ULTRANET_1.FE.M9	ULTRANET_1.FE.M10	ULTRANET_1.FE.M11	ULTRANET_1.FE.M12
True	True	True	True
ULTRANET_1.FE.M13	ULTRANET_1.FE.M14	ULTRANET_1.FE.M15	ULTRANET_1.FE.M16
True	True	True	True
ULTRANET_1.FE.Vxod1	ULTRANET_1.FE.Vxod2	ULTRANET_1.FE.Vxod3	ULTRANET_1.FE.Vxod4
0,8472669	4,3003874	0,0000489	0,0000989
ULTRANET_1.FE.Vxod5	ULTRANET_1.FE.Vxod6	ULTRANET_1.FE.Vxod7	ULTRANET_1.FE.Vxod8
-0,0000507	0,0000525	0,0000668	0,0000161
ULTRANET_1.FE.Strob1	ULTRANET_1.FE.Strob2	ULTRANET_1.FE.Strob3	ULTRANET_1.FE.Strob4
True	True	False	False
ULTRANET_1.FE.Strob5	ULTRANET_1.FE.Strob6	ULTRANET_1.FE.Strob7	ULTRANET_1.FE.Strob8
False	False	False	False

Рис. 5. Окно просмотрщика тегов

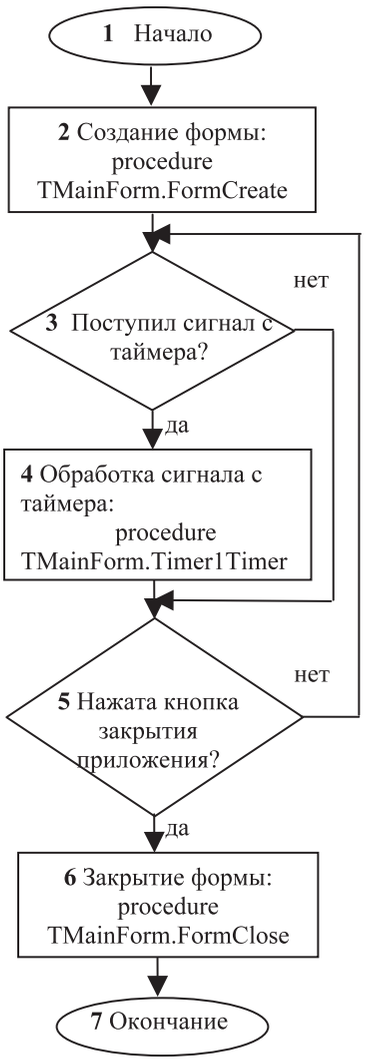


Рис. 6. Алгоритм работы просмотрщика тегов

Важным обстоятельством при работе всего просмотрщика тегов является то, что вся информация, которую выдает OPC-сервер, представлена в строковом формате string. Функция GetItemType используется в процедуре TMainForm.Timer1Timer.

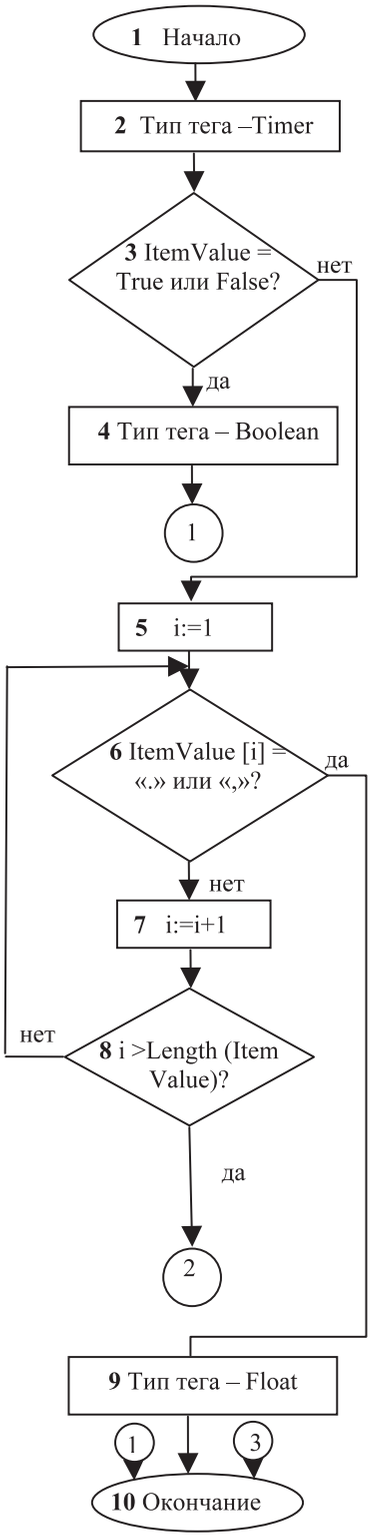


Рис. 7. Алгоритм функции GetItemType (начало)

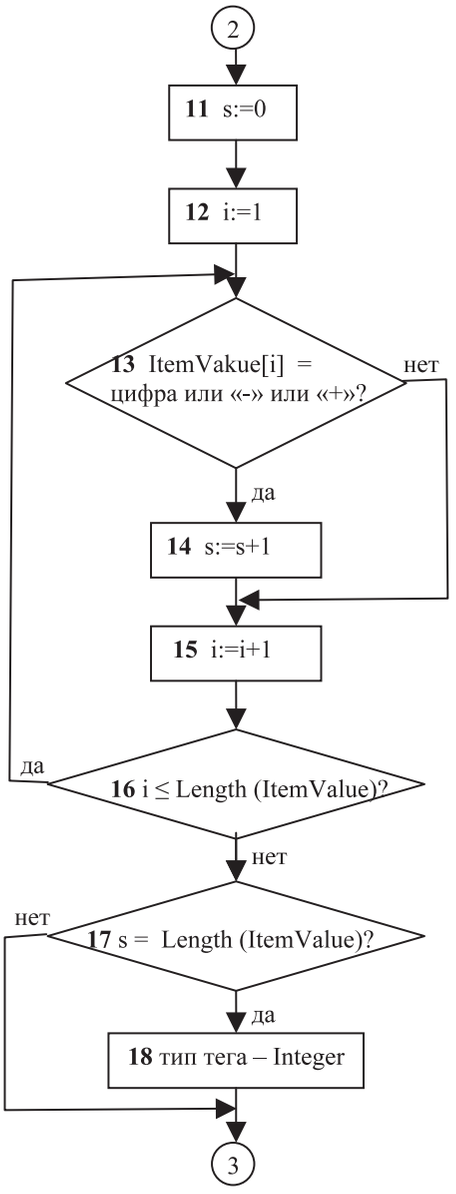


Рис. 7. Алгоритм функции GetItemType (окончание)

Сначала (блок 2) типу тега присваивается значение Timer – функция формулирует результат в виде строки. Затем (блок 3) анализируется значение ItemValue на равенство True или False. Эти два значения может принимать только переменная или константа типа Boolean. При выполнении одного из этих условий, типу тега присваивается значение Boolean.

Если тег не имеет тип Boolean, то начинается посимвольная проверка ItemValue (блоки 5-8). Если в результате этой проверки оказывается, что i-ый символ принимает значение «.» или «,», то типу тега сразу присваивается значение Float – действительное число (блок 9). Точка или запятая являются обязательными разделителями в действительных числах.

Если точка или запятая в значении ItemValue отсутствуют, то начинается его проверка на тип Integer – целое число (блоки 11-17). Признаком целого числа является наличие в нем лишь цифр от 0 до 9 и знаков «+» или «–». Естественно, что все эти знаки представлены символами в строковом выражении.

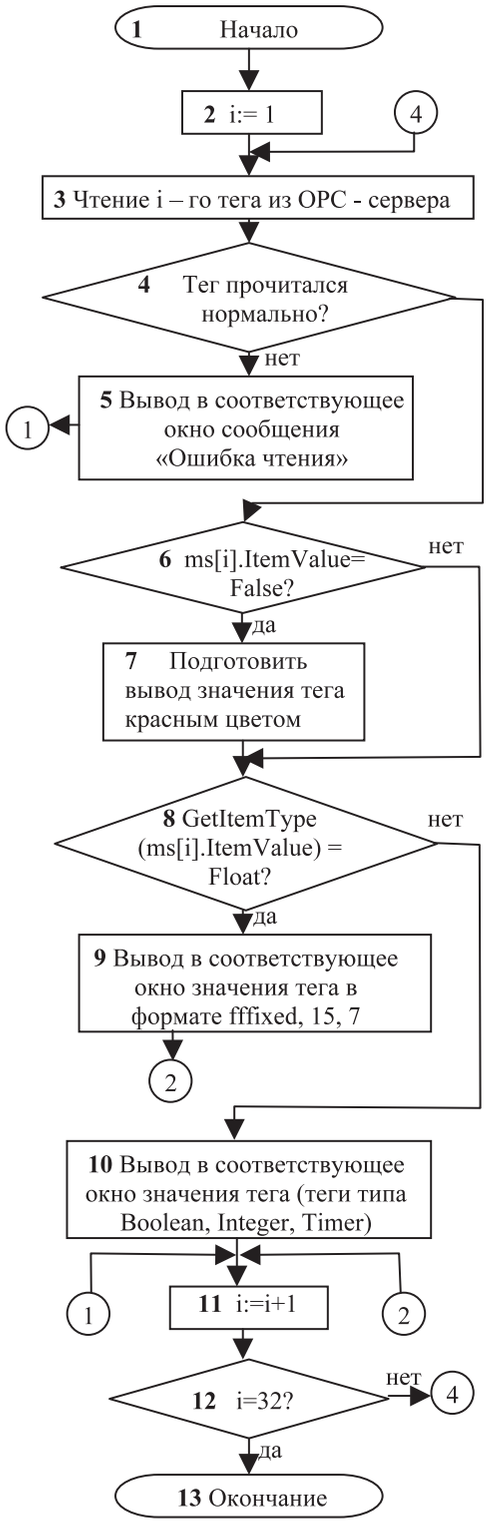


Рис. 8. Алгоритм процедуры TMainForm. Timer1Timer

В данном фрагменте алгоритма: i – номер символа в строке, s – число искомых символов в строке. Если в результате поиска оказалось, что все символы в строке относятся к цифрам или к знакам «+» или «–», то делается вывод, что тег имеет тип Integer (блок 18).

Процедура TMainForm.Timer1Timer является основной в приложении-просмотрщике тегов. Алгоритм процедуры приведен на рис. 8.

Чтение тегов здесь выполняется циклически. После чтения i-го тега из OPC-сервера сразу выполняется проверка на правильность его чтения, что выполняется с помощью функции Succeded(HR) (блок 4). Если тег прочитался неправильно, то в соответствующее окно формы выводится сообщение «Ошибка чтения» (блок 5).

Далее проверяется (блок 6), равен ли тег значению False (тип Boolean). Если он имеет значение False, то подготавливается вывод его значения красным цветом (блок 7). Затем (блок 8) тип тега проверяется на Float –действительное число. Для проверки используется функция GetItemType. Если тег имеет тип Float, то его значение выводится (блок 9) в соответствующее окно в формате fffixed, 15, 7. Если тег не относится к типу Float, то его значение выводится в соответствующее окно (блок 10).

Блоки 1; 11 и 13 предназначены для организации циклического вывода значений тегов в соответствующие окна формы просмотрщика тегов. Здесь параметр i = 1 … 32 – номер тега.

Процедура TMainForm.FormClose содержит всего один оператор Server: = nil, что приводит к выключению OPC-сервера.

Выводы

1. Показано, что проблема передачи информации из промышленной сети PlcNet на диспетчерский уровень РСУ может быть решена путем использования клиент-серверной архитектуры. В качестве OPC-клиента выступает SCADA-система, а в качестве OPC-сервера может быть использован Fastwel PLCNet OPC Server.

2. С использованием среды визуального программирования Delphi разработана SCADA-система, реализованная в виде просмотрщика тегов типов Boolean, Float, Integer, Timer.

3. Рассмотрены механизмы взаимодействия SCADA-системы и OPC-сервера: запуск OPC-сервера, добавление в него имени сегмента и имен тегов, определение типа тегов и вывода значения тегов в окна на форме просмотрщика тегов. Приведены алгоритмы соответствующих процедур и функций.

Литература

1. Анашкин А.С., Кадыров Э.Д., Харазов В.Г. Техническое и программное обеспечение распределенных систем управления. СпБ: Изд. П-2, 2004. – 368 с.
2. Кангин В.В., Кангин М.В., Ямолдинов Д.Н. Программные аспекты проектирования SCADA-систем. Н. Новгород: Изд. НГТУ, 2007. – 259 с.

EXCHANGE TO INFORMATION IN INDUSTRIAL NETWORK PlcNet

Kangin V.V., Lozhkin L.D., Yamoldinov D.N.

The Article is dedicated to questions to organizations of the exchange by information between level portioned managerial system, built on base of the industrial network PlcNet. It Is Shown that problem of the exchange by information on industrial network is central when designing SCADA-a systems. The Decision of this problem lies in planes of the organizations client-server relations between SCADA-a system and OPC-a server, delivered by developer of the network equipment. In this case problems of the exchange are reduced to programmed information exchange between SCADA-a system, executing role OPC-a client, and OPC-a server (is chose Fastwel PlcNet OPC Server). The Designed algorithm observer tegs, allowing observe importance's 32-h тегов, taken on network from controller lower level. The Designed algorithms of the procedures and function, providing watching possibility observer tegs: start OPC-a server Fastwel PlcNet OPC Server, accompaniment in it name of the segment and names tegs, determination of the type tegs and conclusion of importance's tegs in window on the form observer, stop OPC-server.

Keywords: *portioned managerial system, industrial network, SCADA, client-a server relations, OPC-a client, OPC-a server, teg.*

Кангин Владимир Венедиктович, д.т.н., профессор, заведующий Кафедрой «Автоматизация машиностроения» (АМ) Арзамасского политехнического института – филиала Нижегородского государственного технического университета (АПИ НГТУ). Тел. 8-906-352-11-91. E-mail: king202008@yandex.ru

Ложкин Леонид Дидимович, к.т.н., заведующий лабораториями Кафедры «Радиосвязь, радиовещание и телевидение» Поволжского государственного университета телекоммуникаций и информатики. Тел. (8-846) 339-11-74. E-mail: astro76@rambler.ru

Ямолдинов Дмитрий Николаевич, аспирант Кафедры АМ АПИ НГТУ. Тел. (8-83147) 252-24. E-mail: king202008@yandex.ru

ТЕХНОЛОГИИ КОМПЬЮТЕРНЫХ СИСТЕМ И СЕТЕЙ

УДК 004.942

ПРОГРАММНАЯ СИСТЕМА АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ КОМПЬЮТЕРНЫХ СЕТЕЙ НА ОСНОВЕ АППРОКСИМАЦИОННОГО ПОДХОДА

Бахарева Н.Ф.

В статье описывается структура программной системы, включающей несколько программ. Приведен краткий анализ математического аппарата, который использован для описания моделей трафика компьютерной сети. В качестве математической модели трафика получены уравнения равновесия потоков на уровне первых двух моментов распределений интервалов времени в потоках. В статье также приведены примеры расчета конкретных сетевых структур.

Ключевые слова: поток и его математическое представление, моментные характеристики распределений, математические операции агрегирования и вероятностного разрежения потоков, уравнения равновесия потоков.

Введение

Для решения задачи анализа производительности сети, заключающейся в определении всех основных узловых и сетевых характеристик, ее модель должна быть декомпозирована на отдельные узлы с вычислением характеристик входных и выходных потоков в каждом