

Заключение

Предложенный метод построения эталонов состояния компьютерной сети позволяет, за счет применения методов математической теории распознавания образов, более точно адаптировать значения эталонов для условий конкретной КС. Уровень точности зависит от объема доступной для анализа базы данных наблюдения за сетью в устойчивых состояниях.

Метод ориентирован на применение автоматизированных средств анализа и, при наличии достаточных вычислительных мощностей, позволяет организовать динамическую модификацию эталонов, используемых для выполнения диагностики состояния.

Литература

1. Кульгин М. Технологии корпоративных сетей. Энциклопедия – СПб: Питер, 2000. – 704 с.
2. Шиндер Д.Л. Основы компьютерных сетей. : Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 656 с.
3. Бигелоу С. Сети: поиск неисправностей, поддержка и восстановление : Пер. с англ. – СПб.: БХВ-Петербург, 2005. – 1200 с.

ПРОЕКТИРОВАНИЕ СЕТИ ДАТЧИКОВ С БЕСПРОВОДНЫМ ДОСТУПОМ

С.И. Сиротко, А.В. Дайнеко, С.В. Юзефович
Беларусь, г.Минск

Одним из перспективных направлений внедрения беспроводных технологий являются сети беспроводных датчиков различного назначения в системах управления, мониторинга, передачи данных. Беспроводный интерфейс для входящих в них устройств обеспечивает не только мобильность компонентов системы, но и повышение оперативности и снижение стоимости ее развертывания, а также работоспособность в условиях, когда проводные линии и контактные соединения неприменимы. Совместное функционирование устройств в одной сети требует унификации средств взаимодействия, а это, в свою очередь, приводит к потребности разрабатывать по возможности универсальные решения.

Рассматривая датчики, можно прийти к выводу, что, несмотря на разнообразие их типов, число различных режимы работы ограничено. Наиболее характерны следующие варианты:

- 1) непрерывное преобразование данных с оперативной (on-line) передачей их (возможна буферизация для оптимизации трафика и компенсации влияния задержек);
- 2) преобразование данных с малым периодом, накопление данных и последующая их передача большим массивом;

3) длительная автономная (off-line), большой период взятия отсчетов, накопление данных и лишь затем передача массива.

Следовательно, необходимо совмещать следующие характеристики:

- пропускная способность канала, достаточная для поддержки большого числа соединений, но не слишком избыточная с учетом стоимости и энергетической эффективности решения;
- минимальное энергопотребление в состоянии ожидания;
- наличие гибкого управления системой, желательно – открытость и расширяемость применяемых протоколов и интерфейсов;
- небольшая стоимость в условиях серийного производства.

Очевидно, создание единственного устройства, удовлетворяющее всем требованиям, затруднено. Более целесообразно разрабатывать семейство совместимых между собой устройств с отличающимися параметрами, которые бы и перекрывали весь спектр применений.

С точки зрения разработки программного обеспечения, наиболее гибким и удобным было бы использование готовых аппаратных платформ широкого применения под управлением распространенных полнофункциональных ОС, что характерно для разного рода мобильных и "встраиваемых" (embedded) систем. В частности, это означало бы поддержку стандартного стека протоколов TCP/IP. Однако ресурсы такого устройства были бы в большинстве случаев избыточными, а размеры, стоимость и энергопотребление – явно завышенными. Следовательно, требуются специализированные решения, наиболее полно отвечающие всему комплексу требований. В то же время, на них желательно перенести основные принципы и технологии существующих сетей – как с точки зрения использования имеющихся наработок, так и для упрощения последующей интеграции.

ZigBee – основанный на спецификациях IEEE 802.15.4 открытый отраслевой стандарт построения беспроводных сетей, разработанный и развиваемый ZigBee Alliance [1]. Это низкоскоростная (до 250 кбит/с) экономичная технология, оптимизированная для сетей устройств мониторинга и управления, в настоящее время внедряется различных областях. Используемая частота – 2,4 ГГц, дальность связи может составлять от нескольких метров до нескольких сот метров, реже километров. Продукты, использующие ZigBee, выпускаются рядом производителей и широко представлены на рынке, причем в виде как отдельных приемопередатчиков, так и законченных модулей со стандартными интерфейсами. Многие модули предусматривают подключение внешних датчиков (например, MaxStream XBee, [2]), но достижимые при этом характеристики для многих применений недостаточны.

В целом, можно сделать вывод о целесообразности построения беспроводной сети датчиков на основе технологии и продуктов ZigBee, характеристики которых для данного применения близки к оптимальным

ZigBee предполагает наличие стека сетевых протоколов, напоминающего стек TCP/IP, но реализуемый на ограниченных аппаратных ресурсах. Поддерживаются как простейшие соединения "точка-точка", так и разветвленные сети с выделенными узлами-"координаторами", для которых решаются задачи мар-

шрутизации, управления передачей сообщений и так далее [1]. Однако реализации стека протоколов в продуктах различных производителей либо версий лишь частично совместимы друг с другом, что ограничивает возможности построения неоднородной сети.

Для рассматриваемого применения – обслуживание группы датчиков – целесообразным представляется ограничиться одноранговой звездообразной сетью, одно из устройств которой является концентратором данных, передаваемых остальными устройствами – датчиками. В свою очередь, концентратор связан с ЭВМ, которая осуществляет сбор и обработку информации, а также управление сетью и конфигурирование посредством передаваемых через концентратор команд (см рис. 1).

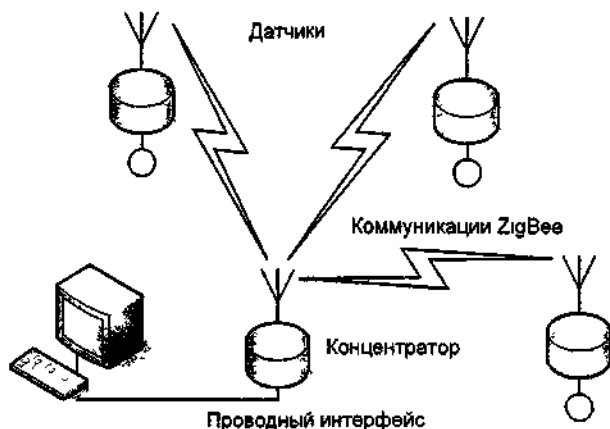


Рис. 1. Структура беспроводной сети датчиков

Таким образом, каждый датчик связан с концентратором соединением "точка-точка", что снижает зависимость от особенностей реализации ZigBee. Все устройства равноправны на уровне собственно ZigBee, и лишь на прикладном уровне различны по выполняемым функциям.

Разумеется, подобный подход требует самостоятельной реализации некоторых сетевых функций, например, обнаружения новых подключений и контроля их состояния. Также ограничиваются масштабы сети и возможности оперативного переключения радиочастотных каналов и оптимизации их загрузки. С другой стороны, жесткое разграничение занимаемых каналов позволяет организовать не разделенные территориально, но не зависящие друг от друга группы оконечных устройств (датчиков), обслуживаемых отдельными концентраторами.

Оконечные устройства выполняют следующие основные функции:

- управление датчиком требуемого типа и преобразование его сигнала в форму, подходящую для дальнейшей обработки;
- первичная обработка полученных данных;
- промежуточное хранение данных;

- связь с управляющим узлом посредством ZigBee;
- интерпретация команд протокола прикладного уровня, конфигурирование устройства и управление его работой.

Как отмечалось выше, устройство должно быть в достаточной степени универсальным, то есть поддерживать различные типы датчиков и режимы. Практически безальтернативное в настоящее время решение – использование достаточно мощного, но экономичного микроконтроллера с соответствующим набором периферийных устройств, под управлением сменяемого "встроенного" программного обеспечения и развитой системой описания конфигурации. Но даже в этом случае перекрыть все разнообразие возможных применений единственным вариантом устройства вряд ли возможно, необходимо изначально предусмотреть в его архитектуре модульность и возможность переконфигурации.

Разрабатываемая архитектура призвана стать основой для создания универсальных оконечных устройств беспроводной сети (см. рис. 2). Отметим, что хотя здесь обсуждается в основном один тип устройства – модуль беспроводного датчика, аналогичным образом должен быть организован и концентратор сети.

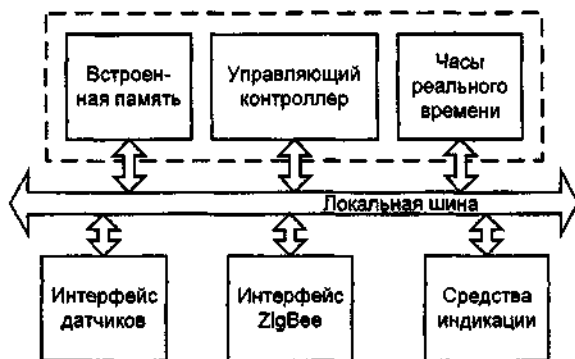


Рис. 2. Архитектура оконечного устройства

Архитектура относится к типу "общая шина" (здесь названа локальной, так как вывод ее за пределы устройства не планируется). Шину составляют: один или нескольких последовательных каналов передачи данных; асинхронные сигналы управления (арбитража шины); сигналы внешних прерываний (событий); линии питающего напряжения. Последовательные каналы SPI, UART или I2C служат для обмена командами и данными между компонентами устройства, их набор зависит от используемого микроконтроллера и периферии. Наличие более чем одного канала полезно для параллелизации операции ввода-вывода и раздельного управления периферией. Для снижения энергопотребления следует предусматривать раздельное питание узлов и их групп с возможностью избирательного отключения. Использование общей шины с унифицированными интерфейсами упрощает проектирование версий устройства с различной аппаратно конфигурацией и способствует их программной совместимости.

Для достижения конструктивной модульности и упрощения изготовления и обслуживания устройство может состоять нескольких плат, каждая из которых является функционально законченным изделием с определенным набором интерфейсов. Наиболее вероятно следующее разделение: контроллер и его ближайшее окружение (в первую очередь память) объединяются на одной плате, а основные периферийные узлы располагаются на отдельных платах. Для подключения к шине будут служить унифицированные разъемы на всех платах.

Наиболее существенными компонентами устройства, выбор которых во многом определяет его характеристики, являются центральный управляющий контроллер и интерфейс ZigBee. В настоящее время в этом качестве выбраны Texas Instruments MSP430 и MaxStream XBee.

16-разрядные микроконтроллеры MSP430 [3] используется достаточно долго и хорошо зарекомендовали себя. Им свойственны весьма низкое энергопотребление, достаточная для данного применения производительность, разнообразная поддерживаемая периферия, а также программная модель, оптимизированная для использования языка программирования C. Продукт имеет хорошую поддержку, включая примеры использования и библиотеки. В семействе присутствуют образцы с большим количеством портов ввода-вывода (в корпусах с 64 и даже 100 выводами), что позволяет реализовать управление шиной без дополнительных внешних дешифраторов, мультиплексоров и т.п.

Радиомодули XBee и XBee-PRO [2] представляют собой законченные сборки, объединяющие компоненты радиочастотного интерфейса под управлением собственного микроконтроллера. Внешним интерфейсом служит стандартный UART. Протокол обмена с модулем двухуровневый: подмножество AT-команд, типичных для модемов, и собственный пакетный. Интерфейсы всех версий модулей совместимы, различаются только параметры радиочастотной части. Для выбранной конфигурации сети возможности этих модулей избыточны, и в дальнейшем возможна замена их на более простые решения других производителей, но сейчас важна предоставляемая возможность подключения их непосредственно к ЭВМ и программная эмуляция функций устройства для тестирования и исследования возможностей системы.

Управление устройствами предусматривает наличие соответствующего протокола для передачи команд и данных. Фактически необходимо реализовать верхний (прикладной) уровень стека протоколов системы. На данном этапе целесообразно разрабатывать оригинальный набор команд и сообщений, однако в дальнейшем не исключается переход к одному из существующих протоколов аналогичного назначения. Среди команд необходимо предусмотреть также и служебные, отвечающие за функции более низкого уровня – например, установление соединения (ассоциации) между устройствами.

Необходимым условием жизнеспособности архитектуры устройств и сети в целом является наличие эффективной подсистемы конфигурирования, обеспечивающей адаптацию устройств к различным режимам работы и составам периферийных узлов по возможности без полной замены кода программ. Образцом может служить изложенная в [4] концепция датчиков, содержащих исчерпывающее описание своих характеристик, функциональных возможностей и

форматов данных. Сейчас планируется хранение конфигурации каждого устройства в виде структур данных в энергонезависимой памяти и изменение ее соответствующими командами протокола. Также в протоколе предусматривается описание каждого передаваемого элемента данных.

Заложенная в архитектуре устройства гибкость, конфигурируемость, и переменные состав и характеристики узлов требуют также строгого следования модульному подходу при проектировании программного обеспечения, например опираясь на рекомендации [5]. В частности, целесообразно выделять модули – своего рода "драйверы", реализующие программные модели компонентов и узлов и обеспечивающие за унифицированное взаимодействие с ними. Это облегчит управление версиями программного обеспечения либо оперативную настройку на текущую конфигурацию, что сокращает затраты на сопровождение системы.

В настоящее время работы продолжают в направлении уточнения конфигурации аппаратных средств и проектирования программного обеспечения как устройства, так и управляющей ЭВМ.

Литература

1. ZigBee specification. – ZigBee Alliance, 2006
2. XBee/XBee-PRO OEM RF Modules. Product Manual v1.xAx – MaxStream Inc., 2006.
3. SLAU049F: MSP430x1xx Family. User's Guide. – Texas Instruments Inc., 2006.
4. IEEE 1451.4: Transducer Electronic Datasheet. – IEEE Standard Association, 2004
5. SPRA 701: A Software Modularity Strategy for Digital Control Systems. – Texas Instruments Inc., 2001.

РЕАЛИЗАЦИЯ СИСТЕМЫ КОНТРОЛЯ ИСПОЛЬЗОВАНИЯ ПАМЯТИ В РАМКАХ ВИРТУАЛЬНОЙ МАШИНЫ JAVA

С.П. Маяк, А.А. Сокольский
Беларусь, г. Минск

На сегодняшний день ни одна из коммерчески успешных операционных систем общего назначения не предоставляет должного механизма управления ресурсами для программ, класса серверов приложений масштаба предприятия. В силу этого, сервер приложений не может ограничивать выделение ресурсов контролируемыми приложениями, что делает невозможным обеспечение необходимого уровня робастности и управления.

Для реального решения подобных задач, еще с начала 60-х годов фактически используется только один метод: запуск сервера в рамках ограничивающей виртуальной среды, а не предоставление серверу необходимых механизмов контроля со стороны операционной системы. Среда создавалась либо через полную виртуализацию (виртуализация оборудования) операционной системы (например, IBM CP/CMS и z/VM, VMWare, VirtualPC - как ее современное раз-