

УДК 338.467

Л.И. Миронова, канд. тех. наук, доцент,
О.М. Литовских, магистрант,¹
Екатеринбург, Россия

АВТОМАТИЗАЦИЯ ДОСТАВКИ БЛЮД И НАПИТКОВ В ЗАВЕДЕНИЯХ ОБЩЕСТВЕННОГО ПИТАНИЯ

В статье рассмотрен подход к обучению студентов в экономическом университете по направлению «Математическое обеспечение и администрирование информационных систем», реализующий технологию междисциплинарного проектирования. В рамках междисциплинарных проектов осуществляется информатизация производственных процессов в предметных областях, непрофильных по отношению к информатике. В данной статье рассмотрена сфера общественного питания. Предметом исследования является организация оригинальной подачи блюд и напитков в заведениях общественного питания. Цель исследования – разработка программно-аппаратного комплекса (ПАК), осуществляющего автоматизированную подачу блюд. На основе анализа существующих решений предложено цифровое управление моделями поездов масштаба 1:22, производства компаний LGB и Piko. Для реализации ПАК использован микроконтроллер Ардуино, интегрированной средой разработки которого является кроссплатформенное приложение на Java. Для разработки программного обеспечения комплексного управления железной дорогой использован .NET Framework и язык программирования C#. Для создания пользовательского интерфейса использована технология *Windows Presentation Foundation* (WPF), позволяющая создавать клиентские приложения Windows с визуально привлекательными возможностями взаимодействия с пользователем, являющаяся графической (презентационной) подсистемой в составе .NET Framework (начиная с версии 3.0), использующая язык XAML. В результате проведенной работы сформулированы базовые требования, реализация которых позволила разработать программно-аппаратный комплекс автоматизации доставки блюд и напитков, обладающий комфортным интерфейсом, а именно: требования к управлению периферией, управлению моделями поездов, общие требования к системе, требования к пользовательскому интерфейсу. Система прошла опытную эксплуатацию и внедрена в ресторане «Воксхолл» г. Екатеринбурга, о чем имеется соответствующий акт. Описанная в статье технология разработки программно-аппаратного комплекса, позволяющего автоматизировать процесс доставки блюд и напитков в заведениях общественного питания, позволяет перейти на качественно новый уровень организации менеджмента в системе общественного питания, что способствует привлечению клиентов, уменьшению времени их обслуживания и сокращения штата персонала заведения общественного питания.

Ключевые слова: междисциплинарное проектирование; программно-аппаратный комплекс; ресторанный бизнес; автоматизированная подача блюд и напитков; пользовательский интерфейс; микроконтроллер Ардуино.

Актуальность исследования

Активное развитие информационных технологий приводит к тому, что их повсеместное внедрение можно заметить практически в любой отрасли экономики.

В рамках обучения бакалавров Института менеджмента и информационных технологий по направлению подготовки

«Математическое обеспечение и администрирование информационных систем»

¹ *Миронова Людмила Ивановна* – кандидат технических наук, доцент кафедры гидравлики Строительного института Уральского федерального университета имени первого Президента России Б.Н. Ельцина, г. Екатеринбург, Россия (620002, г. Екатеринбург, ул. Мира, 19); e-mail: mirmila@mail.ru.

в Уральском государственном экономическом университете (УрГЭУ) разрабатываются междисциплинарные проекты, направленные на информатизацию производственных процессов в предметных областях, непрофильных по отношению к информатике, таких как торговля, банковское дело, менеджмент, маркетинг, макро- и микроэкономика, экономика и право, ресторанный и туристический бизнес, технология общественного питания, товароведение и экспертиза товаров и т. п.

Применение технологии междисциплинарного проектирования в экономическом вузе оказалось возможным, поскольку базовая профессиональная подготовка бакалавров в области математического обеспечения и администрирования информационных систем (МО и АИС) представляет собой сочетание алгоритмическо-программистского (Computer Science) и инженерного (Computer Engineering) подходов в обучении. Такой симбиоз двух направлений подготовки позволяет формировать у будущих бакалавров МО и АИС одновременно с навыками оценки, отбора, упорядочения и обработки информации творческий подход к созданию новых методов обработки информации и средств информатизации.

Одновременно с этим будущая деятельность бакалавров МО и АИС относится к информационно-технологической сфере, поскольку, согласно ФГОС ВО академического бакалавриата, объектами их деятельности являются «математические и алгоритмические модели, программы, программные системы и комплексы, методы их проектирования и реализации, способы производства, сопровождения, эксплуатации и администрирования в различных

областях, в т. ч. и междисциплинарных. Бакалавр МО и АИС готовится к следующим видам деятельности: научно-исследовательской, проектно-конструкторской, организационно-управленческой, эксплуатационно-управленческой, педагогической».

Согласно ФГОС ВО необходимо формировать у будущих бакалавров МО и АИС компетентность в области разработки различного рода алгоритмов и математических моделей, а на их основе программного обеспечения, его внедрения, разработки сопроводительной документации, способности к оформлению и публичному представлению результатов своей деятельности.

Таким образом, разработка программного обеспечения, в состав которого входят электронные учебные пособия, электронные учебно-методические комплексы по дисциплине, обучающие программные средства, программные средства (системы) – тренажеры, контролирующие программные средства, информационно-поисковые, информационно-справочные программные средства, имитационные программные средства (системы), моделирующие программные средства, автоматизированные рабочие места, демонстрационные программные средства, учебно-игровые программные средства и пр. для направлений подготовки экономического университета, непрофильных по отношению к информатике, является актуальной и вполне посильной задачей для бакалавров, обучающихся по направлению подготовки «Математическое обеспечение и администрирование информационных систем».

Объектом исследования является процесс внедрения информационных технологий в процедуру обслуживания клиентов заведений общественного питания. *Предметом исследования* является автоматизация процесса доставки заказов клиентам.

Целью исследования является разработка программно-аппаратного комплекса

Литовских Олег Михайлович – магистрант Института менеджмента и информационных технологий Уральского государственного экономического университета, г. Екатеринбург, Россия (620144, г. Екатеринбург, ул. 8 Марта, 62); e-mail: kateha007@bk.ru.

автоматизации доставки блюд и напитков в заведениях общественного питания.

Степень разработанности проблемы

В данной статье пойдет речь о сфере общественного питания и оптимизации процесса подачи блюд. Предприятиям в сфере общественного питания и ресторанного бизнеса, ввиду высокой конкуренции, необходимо придумывать различные способы привлечения клиентов. Кроме того, существует проблема уменьшения времени обслуживания и сокращения штата персонала. Автоматизированная доставка заказов решает названные проблемы. Оригинальная подача блюд и напитков привлечет в заведение больше клиентов, а автоматизация позволит сократить время обслуживания клиентов и штат официантов. Данное направление в ресторанном бизнесе новое, и по всему миру лишь несколько заведений в мире реализует подобную технологию. Данные факторы также обуславливают *актуальность* тематики данной статьи.

Описываемая технология была ориентирована на ресторан «Воксхолл» (г. Екатеринбург) – ресторан для семейного отдыха. Поэтому при разработке концепции ресторана «Воксхолл» было решено использовать для доставки блюд модели поездов масштаба 1:22, производства компаний LGB и PIKO. К каждому столику прокладывалась садовая железная дорога, по которой ездят поезда. Доставка блюд осуществляется в вагонах этих поездов.

На основе данной концепции к системе управления поездами сформировались следующие минимальные требования: централизованное управление всей железной дорогой; возможность управления каждой моделью поезда по отдельности; осуществления плавного разгона и торможения; переключение стрелок; мониторинг положения конкретного поезда.

В ходе анализа продукции компаний LGB и PIKO было найдено готовое решение, удовлетворяющее большинству поставленных требований, *DIGITAL COMMAND CONTROL* (далее DCC) – существующая система цифрового управления моделями поездов.

Основной предпосылкой создания DCC явилось естественное желание управлять самим поездом (локомотивом), нежели напряжением на рельсах под ним. Корни DCC в США восходят к сороковым годам XX в., когда компания Lionel Trains представила на рынок 2-канальную систему с управлением по частоте. К сожалению, эта система, несмотря на ее очевидную новизну и технологический приоритет, не являлась в той степени надежной, как хотелось бы.

В ранних 60-х гг. прошлого века General Electric (GE) представила 5-канальную систему ASTRAC для управления более чем одним поездом на одном изолированном участке.

В 1979 г. системы Dynatrols Carrier Control и CTC-16 были представлены журналом Model Railroader. Они являлись первыми системами, позволявшими управлять отдельным поездом с достаточной степенью надежности. Проблемы, связанные с этими двумя системами, имели другой источник, нежели качество управления – обе были не совместимы друг с другом и с другими системами.

Начало 80-х гг. ознаменовалось прорывными решениями в микроэлектронике, что резко снизило себестоимость применения цифровых технологий и расширило на порядки сферу их применения.

Тогда за дело взялась компания National Model Railroad Association (NMRA), предложив стандарт сигнала цифрового управления и объединив процесс разработки дальнейших стандартов в рамках собственной организации. С 1994 г., когда первый бюллетень о DCC был опубликован, на-

чалась новая эра в жизни домашних магистралей. Рабочая группа DCC в рамках NMRA продолжает работать над улучшением существующих стандартов и разработкой новых.

Однако сегодняшний конечный результат не был бы возможен, не будь в процесс его реализации вовлечены такие мощные компании индустрии хобби в США, как Digitrax, Quantum Technologies, Soundtraxx и др. Определяющую роль в становлении цифровых систем управления в моделировании железных дорог сыграли факты наличия емкого потребительского рынка в США и давние тамошние традиции этого семейного хобби длиной чуть ли не в столетие.

DCC состоит из пяти основных блоков: пульта управления; станции команд; усилителя; цепи рельсового пути с поданным на него DCC сигналом; декодера команд.

Основная идея DCC состоит в том, что на путь подается кодированный цифровой сигнал, который одновременно является источником тока для привода подвижного состава. Таким образом, DCC сочетает в себе цифровую сеть с прямой адресацией и систему электропитания одновременно.

Первым очевидным следствием этого является отсутствие полярности тока на рельсах, т. е. искать клеммы «+» или «-» не имеет смысла.

Работает система так: пульт управления позволяет выбрать по уникальному номеру декодер и указать команду для передачи, станция команд транслирует это в цифровой сигнал, который передается как пакет или их последовательность на рельсовый путь, с которого и считывается адресатом-декодером, который исполняет эту команду. На рис. 1 показана типовая схема DCC.

Вид сигнала DCC приведен на рис. 2. На рисунке видно, что прямоугольный сигнал двух разных длительностей передает два состояния сигнала, которые являются еди-

ницей или нулем соответственно. Кроме того, прямоугольная форма сигнала приводит к тому, что на рельсах постоянно есть напряжение одинаковой амплитуды, но разной полярности. Этот факт используется для схем питания декодеров и других компонентов, т. е. в одной цепи реализуются и функции питания компонентов, и передачи цифрового сигнала. Параметры сигнала приведены на рис. 3.

Такая организация цепи отлично подходит к существующему много десятиков лет классическому управлению моделями через цепь с использованием двух рельсов и токосъемниках на тележках потребителей, таких как локомотивы, вагоны с электропроводкой и стационарные устройства с управлением через цепь рельсов.

Основной единицей передачи информации в DCC служит пакет. Пакет состоит из последовательности байтов, описывающих текущую команду, выбранный адрес и данные самой команды. Порядок следования информации жестко предопределен стандартом DCC и называется форматом пакета. Не существует какого-то специального механизма в стандарте DCC, гарантирующего подтверждение о доставке пакета. Аппаратура DCC «надеется», что данный пакет будет получен адресатом.

В результате изучения принципов и особенностей работы системы DCC были выявлены ее несомненные достоинства, к которым можно отнести возможность управления каждым поездом по отдельности; наличие возможности регулирования плавности хода; помехоустойчивость, основанная на том, что сигнал поступает на декодер полностью либо не поступает вообще; управляющий сигнал передается по рельсам, следовательно, не требуется радиосвязь; существуют специальные модули позволяющие определять занятость отрезка пути.

К числу недостатков системы DCC можно отнести следующее: поскольку сигнал

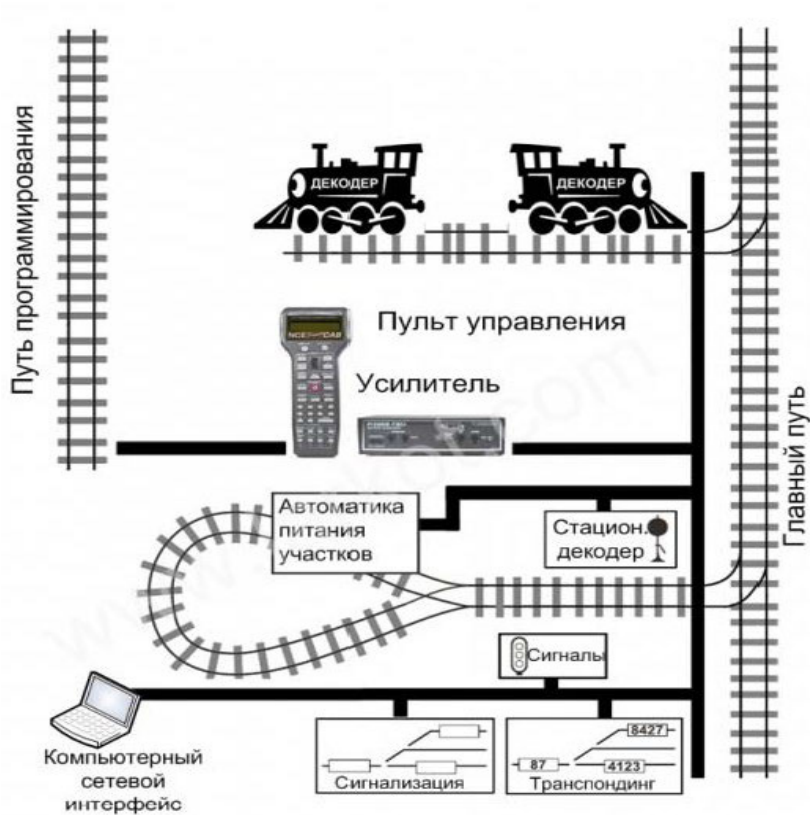


Рис. 1. Типовая схема DCC

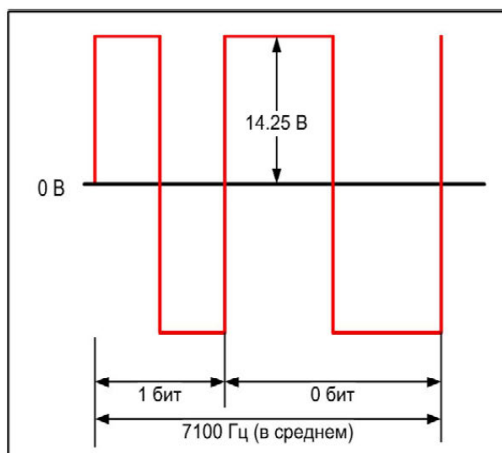


Рис. 2. Диаграмма сигнала DCC

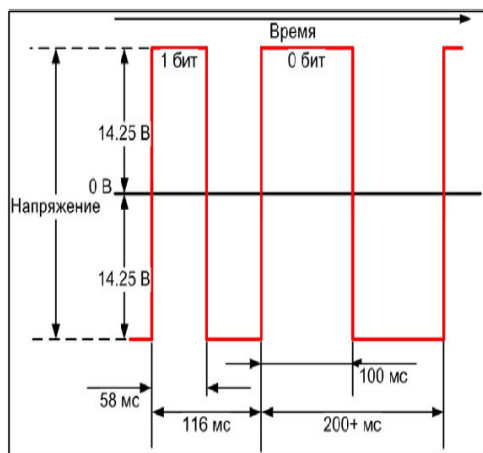


Рис. 3. Параметры сигнала DCC

передается вместе с питанием, нет возможности усилить мощность просто добавлением блока питания. Приходится ставить дорогостоящие усилители (boosters); отсутствие обратной связи между поездом и командной станцией; система позволяет определить лишь занятость участка пути, но не то, какой именно поезд на нем находится.

На основе проведенного анализа положительных и отрицательных характеристик системы DCC было принято решение о разработке собственной системы цифрового управления моделями поездов, которая позволит реализовать техническое задание наилучшим образом.

Технология реализации проекта

Комплексная система управления железной дорогой требует целого набора программных и аппаратных средств. Непосредственно поезда и периферия (стрелки, семафоры и т. п.) должны управляться при помощи микроконтроллеров.

Микроконтроллер – микросхема, предназначенная для управления электронными устройствами. Типичный микроконтроллер сочетает на одном кристалле функции процессора и периферийных устройств, содержит ОЗУ и (или) ПЗУ. По сути, это однокристалльный компьютер, способный выполнять простые задачи [1, 2, 4, 9, 13, 18].

При проектировании микроконтроллеров приходится соблюдать баланс между размерами и стоимостью, с одной стороны, и гибкостью и производительностью – с другой. Для разных приложений оптимальное соотношение этих и других параметров может различаться очень сильно. Поэтому существует огромное количество типов, отличающихся архитектурой процессорного модуля, размером и типом встроенной памяти, набором периферийных устройств, типом корпуса и т. д. В отличие от обычных компьютерных микропроцессоров, в ми-

кроконтроллерах часто используется гарвардская архитектура памяти, то есть раздельное хранение данных и команд в ОЗУ и ПЗУ соответственно [1, 2, 4].

Наиболее простым для программирования являются микроконтроллеры семейства AVR, поддерживающие программирование на языках Assembler, C, C++ [15, 16, 20]. AVR – семейство микроконтроллеров фирмы Atmel. Микроконтроллеры этого семейства (впрочем, как и все микроконтроллеры AVR фирмы «Atmel») являются 8-разрядными микроконтроллерами, предназначенными для встраиваемых приложений [20].

В последнее время все большую популярность набирает плата Arduino, которая представляет собой микроконтроллер AVR с необходимой периферией. [13, 18]. Основными компонентами Arduino-аппаратной вычислительной платформы являются простая плата ввода/вывода и среда разработки на языке Processing/Wiring. Arduino может использоваться как для создания автономных интерактивных объектов, так и подключаться к программному обеспечению, выполняемому на компьютере. Рассылаемые в настоящее время версии могут быть заказаны уже распаянными. Информация об устройстве платы (рисунок печатной платы) находится в открытом доступе и может быть использована теми, кто предпочитает собирать платы самостоятельно [10].

Платы Arduino позволяют использовать большую часть I/O выводов микроконтроллера во внешних схемах. Например, в плате Diecimila доступно 14 цифровых входов/выходов, 6 из которых могут выдавать ШИМ сигнал (широко-импульсная модуляция – процесс управления мощностью, подводимой к нагрузке, путем изменения скважности импульсов, при постоянной частоте), и 6 аналоговых входов. Эти сигналы доступны на плате через контактные площадки или штыревые разъемы. Также доступны несколько видов внешних плат рас-

ширения, называемых «shields» («щиты»), которые присоединяются к плате Arduino через штыревые разъемы.

Интегрированная среда разработки Arduino – это кросс-платформенное приложение на Java, включающее в себя редактор кода, компилятор и модуль передачи прошивки в плату.

Из-за удобства программирования именно микроконтроллер Arduino был выбран для организации низкоуровневого управления моделями поездов и периферией в разработанном программно-аппартном комплексе.

Для разработки программного обеспечения комплексного управления железной дорогой, было решено использовать .NET Framework [12, 16, 21] и язык программирования C# [14, 15, 16, 19]. Для создания пользовательского интерфейса используется технология WPF[8].

Windows Presentation Foundation (далее WPF) – система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML. В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учетом возможностей современного графического оборудования. WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (Extensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трехмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

Программная часть решения построена по *клиент-серверной модели*. Это вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка

распределены между поставщиками услуг (сервисов), называемыми серверами, и заказчиками услуг, называемыми клиентами. Нередко клиенты и серверы взаимодействуют через компьютерную сеть и могут быть как различными физическими устройствами, так и программным обеспечением [6,11].

Преимущества клиент-серверной модели: отсутствие дублирования кода программы-сервера программами-клиентами; так как все вычисления выполняются на сервере, то требования к компьютерам, на которых установлен клиент, снижаются; все данные хранятся на сервере, который, как правило, защищен гораздо лучше большинства клиентов. На сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа; имеется возможность объединить различных клиентов. Использовать ресурсы одного сервера часто могут клиенты с разными аппаратными платформами, операционными системами и т. п.; имеется возможность разгрузить сети за счет того, что между сервером и клиентом передаются небольшие порции данных.

При этом клиент-серверная модель обладает рядом *недостатков*, таких как: неработоспособность сервера может сделать неработоспособной всю вычислительную сеть. Неработоспособным сервером следует считать сервер, производительности которого не хватает на обслуживание всех клиентов, а также сервер, находящийся на ремонте, профилактике и т. п.; поддержка работы данной системы требует отдельного специалиста – системного администратора; высокая стоимость оборудования.

Связь между клиентом и сервером будет проходить по протоколу *Transmission Control Protocol* (далее TCP). Это один из основных протоколов передачи данных Интернета, предназначенный для управле-

ния передач данных в сетях и подсетях ТСР/IP [5, 10].

Проведенный теоретический анализ существующих систем цифрового управления позволил оценить их достоинства и недостатки. В результате рассмотрения средств организации межпроцессорного взаимодействия, а также поиска инструментов для создания системы цифрового управления моделями поездов и периферией для автоматизированной подачи блюд были отобраны: платы Arduino с микроконтроллерами AVR; платформа .NET Framework, язык программирования C# и технология WPF для разработки пользовательского приложения.

На основе анализа рабочего цикла ресторана, с учетом основных целей для системы автоматизации, были сформулированы следующие базовые требования к системе. Система должна обеспечивать:

1. *Управление периферией*: переключать стрелки и контролировать их положение; управлять питанием на рельсах; отключать питание по короткому замыканию.

2. *Управление моделями поездов*: обеспечить возможность задавать скорость и направление поездов; обеспечивать обратную связь, позволяющую определять, дошла ли команда до поезда; обеспечивать возможность отслеживания позиции конкретного поезда на рельсах с заданной точностью.

Программно-аппаратный комплекс должен удовлетворять следующим *общим требованиям*: формировать маршруты к конкретной точке пути, исходя из заданной заранее схемы; автоматически переключать стрелки и обеспечивать движение поезда по маршруту; осуществлять остановку поезда по достижению окончания маршрута; обеспечивать управление поездом из нескольких точек, к примеру, одна станция управления находится на кухне заведения, другая – у барной стойки.

Программно-аппаратный комплекс должен удовлетворять следующим *требованиям к пользовательскому интерфейсу*: обеспечивать возможность отправки поезда по маршруту одним кликом; обеспечивать отображение схемы путей в виде понятной карты; обеспечивать мониторинг позиции конкретного поезда на карте; быть ориентированным на сенсорное управление.

Для реализации требования, чтобы система должна управляться несколькими людьми из разных точек, была использована клиент-серверная модель. Клиенты напрямую взаимодействуют только с сервером. Сервер обрабатывает запросы клиентов и отдает команды подключенным устройствам (стрелкам и поездам).

Наиболее наглядно это видно на рис. 4. Штриховыми линиями обозначена беспроводная связь (Wi-Fi), сплошными – проводная. Клиенты связываются с сервером через сокет по TCP порту 7020. Сервер, в свою очередь, связывается с подключенными устройствами по последовательному (serial) порту, если устройство подключено через проводное (USB) соединение (модули управления питанием, стрелки); по TCP порту через сокет, если устройство беспроводное (модели поездов).

В любом случае формат сообщения одинаков (рис. 5).

Длина команды всегда один байт. Длина значения возможна от 0 до 255 байт. К примеру, для задания скорости поезда сервер отправляет на устройство следующую команду (рис. 6):

Чтобы определять положение поезда на путях с заданной точностью, использована оптическая пара следующей конфигурации: на путях, в тех местах, где нужно знать положения поезда, лежат светодиоды; на поезде закреплен фоторезистор (меняющий сопротивление в зависимости от освещения) [3]. Когда поезд проезжает светодиод, посылается сообщение на сервер. Учитыв-

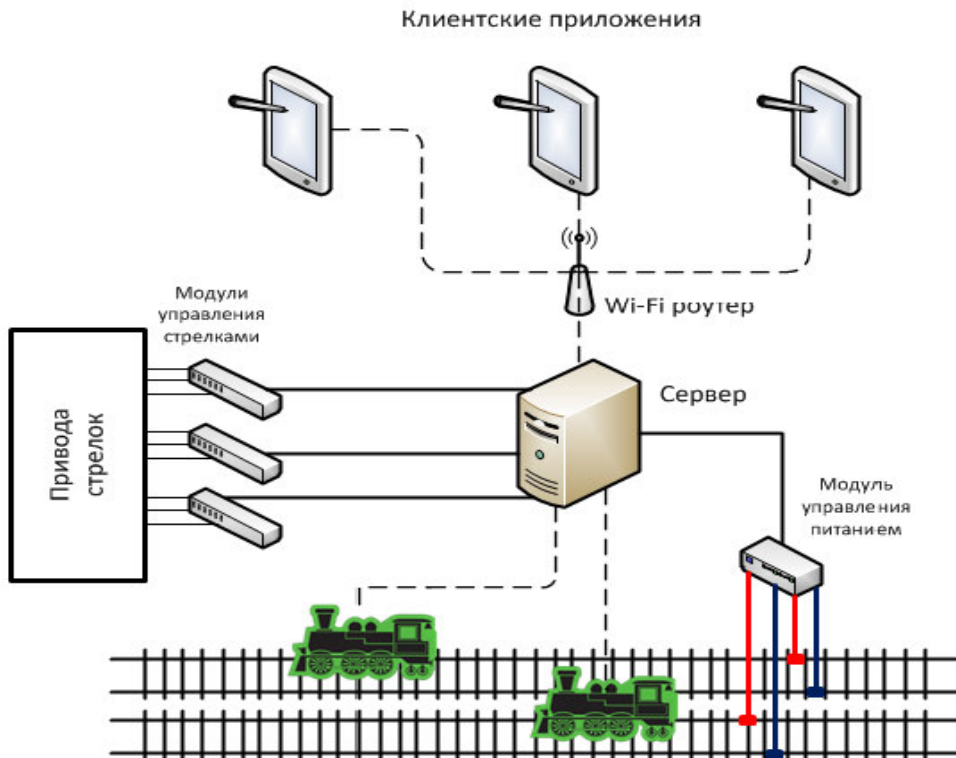


Рис. 4. Общая схема решения

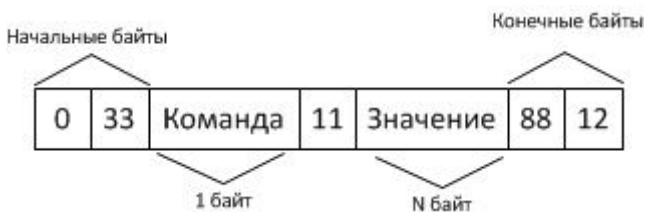


Рис. 5. Формат сообщения сервер-устройства

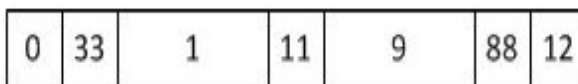


Рис. 6. Команда поезду об изменении скорости

вая начальное положение поезда и текущее состояние стрелок, сервер может точно определить, на каком светодиоде находится поезд.

Для того чтобы сделать управление поездами максимально независимым от питания и обеспечить возможность обратной связи, использованы Wi-Fi-сети. В каждый поезд был добавлен роутер, выступающий в роли Wi-Fi-клиента. Общая схема аппаратной части модели поезда можно представлена на рис. 7.

В качестве Wi-Fi-роутера выступает TP-LINK MR3020 с операционной системой OpenWrt (на основе Linux). С Arduino роутер связан по последовательному порту. Проброс данных с микроконтроллера на сервер происходит при помощи программы Ser2Net, установленной на роутере. Она считывает данные, полученные через последовательный порт, и через TCP порт отправляет их серверу.

Для перевода стрелок использовались электрические переключатели фирмы Tilig. Основной их особенностью является обратная связь, которая позволяет определить, в каком положении сейчас находится стрелка.

Для управления этими переключателями был разработан специальный стрелочный

модуль, представляющий собой Arduino Nano с необходимой обвязкой. Этот модуль поддерживает подключение до шести стрелок. С сервером соединяется через USB.

В соответствии с требованиями к системе, необходимо иметь возможность включать и отключать питание на рельсах и отслеживать короткое замыкание. Для этих целей был разработан *модуль управления питанием*. Он также основан на Arduino. Связь с сервером идет по проводному соединению. В случае короткого замыкания на рельсах срабатывает защита, и нагрузка отключается, при этом отправляется сообщение на сервер. Серверное и клиентское приложения представляют собой прикладные программы, связанные протоколом TCP, через порт 7020.

Сообщения между клиентом и сервером имеют формат, представленный на рис. 8.

Байты 1 и 3, идущие подряд, признак начала сообщения, 2 и 4 – конец. Сообщение имеет удобный в использовании формат JSON [17].

Сообщения клиент-сервер делятся на два типа: команды и запросы. Запросы, в отличие от команд, ждут от сервера ответа.

Схема подключения клиента к серверу изображена на рис. 9. Сначала клиент от-

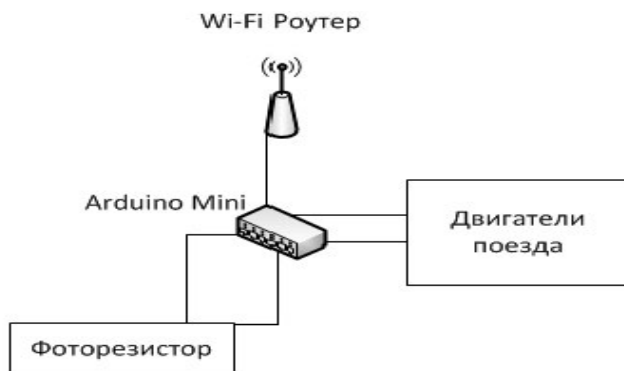


Рис. 7. Схема аппаратной части модели поезда

правляет запрос на подключение, и в ответ получает уникальный числовой идентификатор.

Алгоритм ответа на запросы пользователей реализован следующим образом. Клиент формирует сообщение. Каждому запросу присваивается уникальный идентификатор (ИД); полученный ИД добавляется в список запросов, ожидающих ответа; запрос отправляется на сервер; сервер после выполнения команды отправляет ответ с тем же идентификатором клиенту; клиент ищет полученный ИД в списке запросов, если он найден, выполняется процедура получения ответа.

Обмен сообщениями между клиентом и сервером выполняется в асинхронном режиме. Таким образом, ожидание ответа не останавливает работу клиента.

Весь контроль за поездом во время движения по маршруту ведется со стороны сервера. На карте в программе лишь показывается, когда поезд проезжает очередную точку пути. Поиск кратчайшего пути происходит при помощи алгоритма Дейкстры [7]. Движение поезда по маршруту является блокируемой командой, что означает невозможность выполнения других команд для этого поезда, кроме прерывания. То есть пока маршрут не завершен, возможно лишь прервать маршрут и остановить поезд.

Панель управления поездами представляет собой список расположенных в столбец подключенных к серверу поездов. Для каждого поезда отображается его название, направление и скорость.

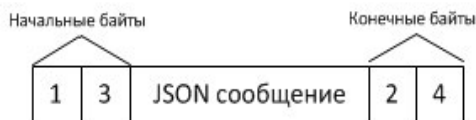


Рис. 8. Формат сообщения клиент-сервер

Методика расчета экономической эффективности программно-аппаратного комплекса

Для расчета экономической эффективности внедрения программно-аппаратного комплекса (ПАК) для автоматизированной подачи блюд в ресторане были выполнены следующие шаги:

- 1) для расчета себестоимости ПАК вычислены стоимость машинного часа и заработная плата программиста – разработчика ПАК;
- 2) определена цена ПАК с учетом ожидаемой прибыли;
- 3) для расчета экономической эффективности вычислены затраты на обслуживание клиентов при ручной раздаче блюд и напитков за год и затраты на обслуживание клиентов при использовании ПАК за год;
- 4) определен срок окупаемости ПАК;
- 5) вычислена экономическая эффективность внедрения ПАК.

Рассмотрим выполненные шаги более детально.

1) Для расчета себестоимости ПАК были использованы данные, представленные в табл. 1.

Стоимость машинного часа работы рассчитывается по формуле:

$$C_{м/ч} = Am_k + C_{эл}, \tag{1}$$

где $C_{м/ч}$ – стоимость машинного часа работы, руб; Am_k – амортизация компьютера за 1 м/ч, руб; $C_{эл}$ – стоимость электроэнергии за 1 ч работы, руб.

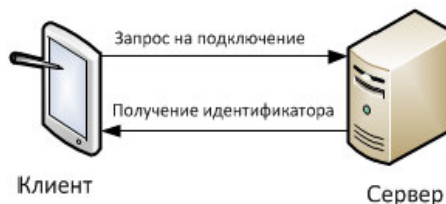


Рис. 9. Схема подключения клиента

Миронова Л.И., Литовских О.М.

Амортизация компьютера за 1 м/ч вычисляется по следующей формуле

$$AM_k = \frac{C_k \cdot N_A}{\Phi_B \cdot 100\%}, \quad (2)$$

где C_k – стоимость компьютера, руб; N_A – норма амортизации компьютера; Φ_B – фонд рабочего времени в год, ч.

Следовательно, амортизация компьютера за 1 м/ч равна

$$AM_k = 30\,000 \times 15/2000/100 = 4500/2000 = 2,25 \text{ руб.}$$

Стоимость электроэнергии за 1 ч работы вычисляется по следующей формуле:

$$C_{эл} = M_k \cdot C_{кв}, \quad (3)$$

где M_k – мощность компьютера, кВт/ч; $C_{кв}$ – стоимость 1 кВт электроэнергии, руб.

$$C_{эл} = 0,25 \times 2,5 = 0,625 \text{ руб.}$$

Используя теперь полученные значения, можно рассчитать стоимость 1 машинного часа (м/ч):

$$C_{м/ч} = 2,25 + 0,625 = 2,9 \text{ руб.}$$

Для расчета заработной платы программиста-разработчика ПАК были использованы данные, представленные в табл. 2.

Затраты на оплату труда программиста:

$$C_{тр} = Z_{пр} \cdot (1 + \text{Отч}) \cdot T_n, \quad (4)$$

где $Z_{пр}$ – зарплата программиста за час, руб.;

Отч – отчисления с зарплаты, %;

T_n – время написания программы, ч.

Заработная плата программиста за час определяется по следующей формуле:

Таблица 1

Данные для расчета себестоимости ПАК

Наименование показателя	Единицы измерения	Обозначение	Значение
Норма амортизации компьютера	%	N_A	15
Стоимость компьютера	Руб.	C_k	30000
Стоимость 1 кВт электроэнергии	Руб.	$C_{кв}$	2,5
Мощность компьютера	кВт/ч	M_k	0,25
Ставка программиста	Руб.	$C_{пр}$	30000
Норма отчислений на дополнительную заработную плату	%	$N_{доп}$	30
Фонд рабочего времени в год	Час	Φ_B	2000

Таблица 2

Затраты времени на написание программы

Наименование этапов работ	Время выполнения
Разработка программного продукта	240 часов
Тестирование и исправление ошибок	48 часов
Опытная эксплуатация	24 часов
Доработка программы по результатам эксплуатации	10 часов
Разработка документации	100 часов
Всего	422 часа

$$З_{\text{пр}} = \frac{С_{\text{т пр}}}{\Phi_{\text{вм}}}, \quad (5)$$

где $С_{\text{т пр}}$ – ставка программиста, руб.; $\Phi_{\text{вм}}$ – фонд рабочего времени в месяц, ч.

$$З_{\text{пр}} = 30000/167 = 180 \text{ руб.}$$

Заработная плата дополнительная определяется по следующей формуле:

$$З_{\text{доп}} = \frac{З_{\text{пр}} \cdot Н_{\text{доп}}}{100 \%}, \quad (6)$$

где $З_{\text{пр}}$ – заработная плата программиста, руб.; $Н_{\text{доп}}$ – норма отчислений на дополнительную зарплату.

$$З_{\text{доп}} = 180 \times 30 \% / 100 \% = 54 \text{ руб.}$$

Зарплата общая вычисляется по следующей формуле:

$$З_{\text{общ}} = З_{\text{пр}} + З_{\text{доп}}, \quad (7)$$

где $З_{\text{общ}}$ – зарплата общая, руб.

$$З_{\text{общ}} = 180 + 54 = 234 \text{ руб.}$$

Отчисления в Пенсионный фонд России (ПФ, 22 % от $З_{\text{общ}}$), в Фонд социального страхования (ФСС, 3,1 % от $З_{\text{общ}}$) и в Федеральный фонд обязательного медицинского страхования (ФФОМС, 5,1 % от $З_{\text{общ}}$), что в сумме составляет 30,2 %, вычислялись по формуле:

$$\text{Отч} = O_{\text{фсс}} + O_{\text{ффомс}} + O_{\text{пф}} \quad (8)$$

где $O_{\text{фсс}}$ – отчисления в ФСС, руб.; $O_{\text{ффомс}}$ – отчисления в ФФОМС, руб.; $O_{\text{пф}}$ – отчисления в ПФ, руб.

$$\text{Отч} = 13 \% \times 234 / 100 \% = 30,42 \text{ руб.}$$

Затраты на оплату труда программиста:

$$С_{\text{тр}} = (234 + 30,42) \times 422 = 111585 \text{ руб.}$$

Теперь себестоимость ПАК вычислим по формуле:

$$С_{\text{пр}} = C_{\text{м/ч}} \cdot T_{\text{н}} + С_{\text{тр}}, \quad (9)$$

где $C_{\text{м/ч}}$ – стоимость машинного часа работы, руб.; $T_{\text{н}}$ – время написания программы, ч; $С_{\text{тр}}$ – затраты на оплату труда, руб.

$$С_{\text{пр}} = 2,9 \times 422 + 111585 = 112809 \text{ руб.}$$

2) При определении цены на ПАК мы исходили из того, что ПАК является узко-

специализированным программным продуктом, поскольку он разрабатывался для нужд одной конкретной организации – ресторана «Воксхолл». Деятельность данной организации имеет свою специфику и, поэтому для нужд других организаций данный программный продукт неприменим. Следовательно, рассчитать цену можно только исходя из размера желаемой прибыли.

Размер желаемой прибыли определяло руководство ресторана, и она составила 25 %. Отсюда следует, что цена разработанной программы будет следующей:

$$Ц_{\text{пр}} = C_{\text{пр}} + \frac{C_{\text{пр}} \cdot \text{Пр}}{100 \%}, \quad (10)$$

где $Ц_{\text{пр}}$ – цена ПАК, руб.; $C_{\text{пр}}$ – себестоимость ПАК, руб.; Пр – ожидаемая прибыль, %.

$$Ц_{\text{пр}} = 112809 + (112809 \times 25) / 100 = 141011 \text{ руб.}$$

3) для расчета экономической эффективности ПАК необходимо рассчитать затраты на обслуживание клиентов ресторана при использовании ПАК за год. Для этого надо рассчитать стоимость часа работы компьютера (потребляемая электроэнергия, амортизация), время работы за год. При использовании ПАК необходим оператор ЭВМ, при этом обслуживание клиентов ресторана стали осуществлять два официанта, зарплата которых стала в два раза ниже, чем до автоматизации, из-за уменьшения их функционала. К затратам следует отнести стоимость микроконтроллера Ардуино, а также стоимость локомотива и трех вагонов, которые развозят блюда и напитки к столам клиентов. При работе с ПАК предполагается, что компьютер будет обслуживаться одним оператором в течение 12-часовой смены и заменит шесть официантов. Необходимые для расчета данные представлены в табл. 3.

Амортизация компьютера за 1 м/ч вычисляется по следующей формуле:

Миронова Л.И., Литовских О.М.

$$A_{M_k} = \frac{C_k \cdot N_{AK}}{T_{фг} \cdot 100\%}, \quad (11)$$

$$A_{M_k} = 30000 \times 15/4380 / 100 = 450000/4380/100 = 0,34 \text{ руб.}$$

Рассчитаем основную заработную плату за час оператора ЭВМ по следующей формуле:

$$Z_{осн} = \frac{C_{оп}}{\Phi_b}, \quad (12)$$

$$Z_{осн} = 7500/167 = 45 \text{ руб.}$$

Дополнительная заработная плата оператора за час вычисляется по следующей формуле:

$$Z_{доп} = \frac{Z_{осн} \cdot N_{доп}}{100\%}, \quad (13)$$

$$Z_{доп} = 45 \times 30 / 100 = 13 \text{ руб.}$$

Общая часовая заработная плата оператора получается суммированием основной и дополнительной зарплат:

$$Z_{общ} = 45 + 13 = 58 \text{ руб.}$$

Отчисления в Пенсионный фонд, Фонд социального страхования и в Федеральный фонд обязательного медицинского страхования вычисляются по формуле:

$$Отч = 13\% \cdot \frac{Z_{общ}}{100\%}, \quad (14)$$

где $Z_{общ}$ – общая заработная плата оператора, руб.; 13% – процент отчислений в бюджет.

$$Отч = 13\% \times 58 / 100\% = 7 \text{ руб.}$$

Стоимость машинного часа равна:

$$C_{м/ч} = 0,34 + 0,625 + 45 + 13 + 7 = 66 \text{ руб.}$$

Экономический эффект от внедрения программного продукта рассчитывается по формуле:

$$\Xi = C_1 - C_2, \quad (15)$$

где C_1 – стоимость не автоматизированной раздачи блюд и напитков официантами;

C_2 – стоимость автоматизированной раздачи блюд и напитков с использованием ПАК.

Таблица 3

Исходные данные для расчета экономической эффективности

Наименование показателя	Единицы измерения	Обозначение	Значение
Затраты машинного времени на автоматизированную подачу блюд (с 12.00 до 00.00)	Ч	T_m	12
Затраты времени на ручную подачу блюд в день	Ч	T_p	12
Стоимость компьютера	Руб.	C_k	30 000
Стоимость микроконтроллера Ардуино	Руб.	$C_{ард}$	254
Стоимость локомотива и три вагонов	Руб.	$C_{лок}$	18000
Ставка оператора ЭВМ	Руб.	$C_{оп}$	7500
Ставка официанта до автоматизации	Руб.	$C_{оф/д}$	40000
Количество официантов до автоматизации подачи блюд	Чел.	N_d	8
Ставка официанта после автоматизации	Руб.	$C_{д оф1}$	20000
Количество официантов при автоматизированной подаче блюд	Чел.	$N_{п}$	2
Фактическое время работы компьютера за год	Ч	$T_{фг}$	4380

Стоимость ручной раздачи блюд в год рассчитывается по формуле:

$$C_1 = (C_{\text{оф/д}} \cdot T_p \cdot N_d / \Phi_B) \cdot 12, \quad (16)$$

где $C_{\text{оф/д}}$ – ставка официанта до автоматизации, руб.;

T_p – затраты времени на ручную раздачу блюд, ч;

Φ_B – фонд рабочего времени в месяц, ч;

N_d – количество официантов, занятых на раздаче блюд до автоматизации, чел.;

$$C_1 = (40000 \cdot 8 \cdot 12/167) \cdot 12 = 275928 \text{ руб.}$$

Стоимость обслуживания посетителей ресторана в год с использованием ПАК вычисляется по формуле:

$$C_2 = (C_{\text{оп}} \cdot T_M / \Phi_B + C_{\text{м/ч}} \cdot T_M + C_{\text{оф/п}} \cdot N_{\text{п}} \cdot T_M / \Phi_B) \cdot 12 + C_{\text{пр}} + C_{\text{ард}} + C_{\text{лок}}, \quad (17)$$

где $C_{\text{оп}}$ – ставка оператора ЭВМ, руб.; Φ_B – фонд рабочего времени в месяц, ч; T_M – затраты времени на автоматизированную подачу блюд, ч; $C_{\text{м/ч}}$ – стоимость одного машинного часа, руб.; $C_{\text{пр}}$ – стоимость программного продукта, руб.; $C_{\text{ард}}$ – стоимость микроконтроллера Ардуино, руб.; $C_{\text{лок}}$ – стоимость локомотива и трех вагонов, руб.

$$C_2 = (7500 \times 12 / 167 + 66 \times 12 + 3740) \times 12 + 112809 + 254 + 6000 + 3 \times 4000 = (539 + 792 + 3740) \times 12 + 112809 + 254 + 18000 = 191915 \text{ руб.}$$

Экономический эффект от внедрения данного программного продукта составляет:

$$\Delta = 275928 - 191915 = 84013 \text{ руб.}$$

4) Определим срок окупаемости капиталовложений:

$$T_{\text{ср}} = \frac{C_k + C_{\text{пр}}}{\Delta}, \quad (18)$$

где C_k – стоимость компьютера, руб.; $C_{\text{пр}}$ – стоимость программного продукта, руб.; Δ – экономический эффект, руб.

$$T_{\text{ср}} = (30000 + 112809) / 84013 = 1,7 \text{ года.}$$

5) Определим экономическую эффективность от вложенных средств:

$$\text{Эф} = \frac{1}{T_{\text{ср}}}, \quad (19)$$

$$\text{Эф} = 1/1,7 = 0,6.$$

Анализ полученных результатов

В ходе проведенного исследования были решены следующие задачи: произведен сбор и анализ требований к программно-аппаратному комплексу для автоматизации подачи блюд в заведении общественного питания; изучены возможности микроконтроллеров, спроектирована и разработана аппаратная и программная части программного продукта, детально описан интерфейс системы и принципы работы серверной и клиентской части решения. Программно-аппаратный комплекс внедрен в ресторане “Воксхолл” г. Екатеринбурга. Проведенные расчеты показали, что общая себестоимость разработанного программно-аппаратного комплекса для автоматизации подачи блюд и напитков в ресторане составила 112 809 руб. Экономия от внедрения данного продукта составит 84 013 руб. в год. При установленной цене ПАК в 141 011 руб. срок окупаемости составит около 1,7 года. Таким образом, экономическая эффективность от внедрения ПАК составляет 0,6.

Основные выводы

В статье описана технология разработки программно-аппаратного комплекса, позволяющего автоматизировать процесс доставки блюд и напитков в заведениях общественного питания, разработанного в рамках междисциплинарного проектирования в Уральском государственном экономическом университете. Программно-аппаратный комплекс разработан на основе использования микроконтроллера Ардуино, позволяющего реализовать инновационные подходы для организации менеджмента в системе общественного питания; сформулированы базовые тре-

бования, реализация которых позволила разработать программно-аппаратный комплекс автоматизации доставки блюд и напитков, обладающий комфортным интерфейсом. Проведена опытная эксплуатация и система внедрена и успешно эксплуатируется в ресторане «Воксхолл», о чем имеется соответствующий акт.

Практическая значимость результатов исследования состоит в том, что внедрение в работу ресторана «Воксхолл» разработанного решения, позволило привлечь новых клиентов благодаря новизне и оригинальности решения; сократить штат официантов за счет автоматизации процесса подачи блюд и в целом повысить качество обслуживания клиентов.

Список использованных источников

1. Белов А.В. Микроконтроллеры AVR в радиолобительской практике. СПб.: БХВ-Петербург, 2007. 352 с.
2. Васильев А.Е. Микроконтроллеры: разработка встраиваемых приложений. СПб.: БХВ-Петербург, 2008. 152 с.
3. Гейтс Д.Э. Введение в электронику. М.: Энергия, 2000. 640 с.
4. Евстифеев А.В. Микроконтроллеры AVR семейства Classic фирмы Atmel. 5-е изд. М.: Издательский дом «Додэка XXI», 2008. 560 с.
5. Камер Д. Сети TCP/IP. Принципы, протоколы и структура. М.: Вильямс, 2011.
6. Коржов В.А. Многоуровневые системы клиент-сервер. М.: Открытые системы, 2009.
7. Кормен Т.Х. Алгоритмы: построение и анализ. М.: Вильямс, 2007. 1296 с.
8. Мак-Дональд М. WPF: Windows Presentation Foundation в .NET 3.5 с примерами. М.: Вильямс, 2010.
9. Микушин А.В. Занимательно о микроконтроллерах. М.: БХВ-Петербург, 2006. 432 с.
10. Ногл М. TCP/IP. Иллюстрированный учебник. М.: ДМК Пресс, 2007. 480 с.
11. Оглтри Т. Модернизация и ремонт сетей / пер. с англ. 4-е изд. М.: Вильямс, 2005. 1328 с.
12. Просиз Д. Программирование для Microsoft .NET. М.: Русская редакция, 2003. 704 с.
13. Соммер У. Программирование микроконтроллерных плат Arduino/Freeduino. СПб.: БХВ-Петербург, 2011. 256 с.
14. Стиллмен Э. Изучаем C#. СПб.: Питер, 2012. 696 с.
15. Троелсен Э. Expression Blend 4 с примерами на C# для профессионалов. М.: Вильямс, 2012. 384 с.
16. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. 6-е изд. М.: Вильямс, 2013. 1311 с.
17. Фримен А. jQuery для профессионалов. М.: Вильямс, 2013. 960 с.
18. Фрунзе А.В. Микроконтроллеры? Это же просто! : в 3-х т. М.: ИД СКИМЕН, 2002. Т. 1. 336 с.
19. Шилдт Г. C# 4.0: полное руководство. М.: Вильямс, 2010. 1056 с.
20. Шонфелдер Г., Шнайдер К. Измерительные устройства на базе микропроцессора ATmega / пер. с нем. СПб.: БХВ-Петербург, 2012. 288 с.
21. McClure W.B. Professional Programming with Mono and .NET/C#. Wrox, 2010.

Mironova L.I., candidate of technical sciences, associate professor,
Ural Federal University
named after the first President of Russia B.N. Yeltsin,
Litovskih O.M., master student,
Urals State University of Economics,
Ekaterinburg, Russia

AUTOMATION OF DISHES AND DRINKS GIVING IN PUBLIC CATERING

The article considers an approach to training of students at an economic university in the “Software and Administration of Information Systems” domain that employs the interdisciplinary design technology. The introduction of IT technologies in production processes is taking place in domains other than computer sciences. In this article, the public catering industry is the focus of research. The object of research is the organization of unusual serving of dishes and drinks in catering organizations. A research objective is the development of a hardware-software system (HSS) which will serve dishes automatically. On the basis of the analysis of existing decisions a digital control system for models of trains of scale 1:22 made by LGB and PIKO companies is proposed. For building the HASS, an Arduino’s microcontroller is used which is built upon integrated environment of a cross-platform application on Java. For the development of the software of integrated management of the railroad .NET Framework and the C# programming language is used. For the creation of the user interface the Windows Presentation Foundation (WPF) technology is used, allowing one to create the Windows client applications with visually attractive opportunities of interaction with the user, being a graphic (presentation) subsystem as a part of .NET Framework (starting with version 3.0) using the XAML language. As a result, basic requirements were formulated which led to the creation of the hardware-software system of automation of food and drink service that possesses a user-friendly interface. The requirements include the management of the periphery, management of the models of trains, general requirements to the system, requirements to the user interface. The system passed trial operation and is introduced at Vauxhall restaurant in Yekaterinburg, which is confirmed by an appropriate certificate. The technology of development of the hardware-software system making it possible to automate the process of the delivery of dishes and drinks in catering facilities that is described in the article makes it possible take the organization of management in catering business to a whole new level, to attract customers, reduce the time of service and reduce catering staff.

Key words: interdisciplinary project based learning; a hardware-software system; restaurant business; automated service of dishes and drinks; user interface; Arduino microcontroller.

References

1. Belov, A.V. (2007). *Mikrokontrollery AVR v radioliubitel'skoi praktike (AVR microcontrollers for radio amateurs)*. St Petersburg, BKhV-Peterburg.
2. Vasil'ev, A.E. (2008). *Mikrokontrollery: razrabotka vstraivaemykh prilozhenii [Microcontrollers: Development of embedded applications]*. St Petersburg, BKhV-Peterburg.

Миронова Л.И., Литовских О.М.

3. Gates, E. (2011). *Introduction to Electronics*. Delmar Cengage Learning (Rus. ed.: Geits, D.E. (2000). *Vvedenie v elektroniku*. Moscow, Energiia).
4. Evstifeev, A.V. (2008). *Mikrokontrollery AVR semeistva Classic firmy Atmel [Classic family of Atmel AVR microcontrollers]*. Moscow, Dodeka XXI Publishing House.
5. Comer, D. (2000). *Internetworking with TCP/IP Vol.1: Principles, Protocols, and Architecture*. Prentice Hall (Rus. ed.: Kamer, D. (2011). *Seti TCP/IP. Printsipy, protokoly i struktura*. Moscow, Vil'iams).
6. Korzhov, V.A. (2009). *Mnogourovnevye sistemy klient-server [Multi-level client-server systems]*. Moscow, Otkrytye sistemy.
7. Cormer, T., Leiserson, C.E., Rivest, L.R., Stein, C. (2009). *Introduction To Algorithms*. The MIT Press (Rus. ed.: Kormen, T.Kh. (2007). *Algoritmy: postroenie i analiz*. Moscow, Vil'iams).
8. MacDonald, M. (2010). *Pro WPF in C# 2010: Windows Presentation Foundation in .NET 4*. Apress (Rus. ed.: Mak-Donal'd, M. (2010). *WPF: Windows Presentation Foundation v .NET 3.5 s primerami*. Moscow, Vil'iams).
9. Mikushin, A.V. (2006). *Zanimatel'no o mikrokontrollerakh [Interesting facts about microcontrollers]*. Moscow, BKhV-Peterburg.
10. Naugle, M. (1998). *Illustrated TCP/IP*. Wiley (Rus. ed.: Nogl, M. (2007). *TCP/IP. Illiustrirovannyi uchebnik*. Moscow, DMK Press).
11. Ogletree, T. (2003). *Upgrading and Repairing Networks*. Que (Rus. ed.: Ogltri, T. (2005). *Modernizatsiia i remont setey*. Moscow, Vil'iams).
12. Prosis, J. (2002). *Programming Microsoft® .NET*. Microsoft Press (Prosis, D. (2003). *Programmirovanie dlia Microsoft .NET*. Moscow, Russkaia redaktsiia).
13. Sommer, U. (2013). *Arduino Mikrocontroller-Programmierung mit Arduino/Freduino*. Franzis Verlag (Rus. ed.: Sommer, U. (2011). *Programmirovanie mikrokontrollernykh plat Arduino/Freduino*. St Petersburg, BKhV-Peterburg).
14. Stellman, A. (2007). *Head First C#*. O'Reilly Media (Rus. ed.: Stillmen, E. (2012). *Izuchaem C#*. St Petersburg, Piter).
15. Troelsen, A. (2011). *Pro Expression Blend 4 (Expert's Voice in Expression Blend)*. Apress. (Rus. ed.: Troelsen, E. (2012). *Expression Blend 4 s primerami na C# dlia professionalov*. Moscow, Vil'iams).
16. Troelsen, A. (2012). *Pro C# 5.0 and the .NET 4.5 Framework*. Apress. (Rus. ed.: Troelsen, E. (2013). *Iazyk programirovaniia C# 5.0 i platforma .NET 4.5*. Moscow, Vil'iams).
17. Freeman, A. (2013). *Pro jQuery 2.0*. Apress. (Rus. ed.: Frimen, A. (2013) *jQuery dlia professionalov*. Moscow, Vil'iams).
18. Frunze, A.V. (2002). *Mikrokontrollery? Eto zhe prosto! [Microcontrollers for dummies]*. ID SKIMEN.
19. Schildt, H. (2010). *C# 4.0 The Complete Reference*. McGraw-Hill Education; (Rus. ed.: Schildt, G. (2010). *C# 4.0: polnoe rukovodstvo*. Moscow, Vil'iams).
20. Schneider, C., Schoenfelder, G. (2010). *Messtechnik mit dem ATmega*. Franzis. (Rus. ed.: Shonfelder G., Shnaider K. (2012). *Izmeritel'nye ustroistva na baze mikroprotssora ATmega*. St Petersburg, BKhV-Peterburg).
21. McClure, W.B. (2010). *Professional Programming with Mono and .NET/C#*. Wrox.

Information about the authors

Mironova Ludmila Ivanovna – Candidate of Technical Sciences, Associate Professor, Department of Hydraulics, Institute of Civil Engineering, Ural Federal University named after the first President of Russia B.N. Yeltsin, Ekaterinburg, Russia (620002, Ekaterinburg, Mira street, 19); e-mail: mirmila@mail.ru.

Litovskih Oleg Mihailovich – Master Student, Institute of Management and Information Technology, Ural State University of Economics, Ekaterinburg, Russia (620144, Ekaterinburg, 8 March street, 62); e-mail: kateha007@bk.ru.