

Белякова Ю.С.

*Научный руководитель: д.т.н., профессор, заведующий каф. Ю.А. Кропотов
Муромский институт (филиал) федерального государственного образовательного
учреждения высшего образования «Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
E-mail: kaf-eivt@yandex.ru*

Разработка программного обеспечения автоматизированной системы определения статистических показателей медицинского учреждения

Основой любой деятельности являются данные, т.е. информация в цифровом или символьном виде, которая зафиксированная на оптических, электронных или бумажных носителях. Такие сведения могут быть обработаны и переданы с использованием средств вычислительной техники и каналов связи.

Относительно систем управления в здравоохранении использование данных более актуально для получения первичной статистической информации, которая применяется для формирования аналитических отчетов, справочников и других статистических материалов, необходимых для принятия управленческих решений. Данные, сгруппированные и представленные в определенном виде, являются исходным материалом для статистического анализа.

Все возрастающая потребность получения информации в системе здравоохранения объясняется необходимостью постоянного мониторинга здоровья населения, деятельности системы здравоохранения, отдельных медицинских организаций и служб.

При разработке программного обеспечения автоматизированной системы в медицинском учреждении в качестве исходных данных необходимо осуществить следующие действия:

- разработка структуры базы данных с использованием CASE технологий;
- разрабатываемое ПО предназначено для руководителей, врачей и экономистов медицинских учреждений;
- ведение статистического учета и отчетности;
- проведение расчета посещаемости специалистов и их фактической нагрузки;
- формирование оценки количественных показателей деятельности врачей медицинского учреждения.

При этом разработанное программное обеспечение должно выполнять функции:

- ввод и хранение информации;
- формирование статистических отчетов по заболеваемости пациентов и посещаемости специалистов;
- формирование отчетной документации по произвольному запросу;
- информация вводится в компьютер с амбулаторного талона пациента.

Разрабатываемое приложение должно иметь понятный любому пользователю интерфейс, оно должно использовать в основе своего построения диалоговый режим, который выражается в диалоге программы с пользователем при редактировании данных, удалении, фильтрации, обработке ошибок ввода.

Вносимые в БД данные могут содержать сведения, потеря которых может привести к неправильным итоговым данным. Поэтому при разработке БД необходимо учесть возможность вывода сообщений о неправильном вводе данных, иначе вносимые данные могут быть безвозвратно утеряны.

Функциональное назначение разрабатываемой программы должно состоять в представлении возможностей работать с большим набором однотипных данных, а эксплуатационное назначение - в способе реализации возможностей работать с большим объемом информации.

Большакова В.А., Паршин К.Н., Морозов Н.Д., Чижов М.А.
Научный руководитель: д.т.н., профессор, зав. каф. Ю.А. Кропотов
Муромский институт (филиал) федерального государственного образовательного
учреждения высшего образования «Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
E-mail: kaf-eivt@yandex.ru

Развитие программирования как науки

История программирования берёт своё начало ещё в Древней Греции. Механическое устройство, поднятое в 1901 году с древнего судна, оказалось своеобразным калькулятором, использовавшим шестерни различных размеров и конфигурации, обуславливавших его работу. Антикитерский механизм (так его в последствие назвали) датируется 100 годом до н.э.

Первое программируемое вычислительное устройство, Аналитическую машину, разработал Чарлз Бэббидж (но не смог её построить). 19 июля 1843 года графиня Ада Августа Лавлейс, дочь великого английского поэта Джорджа Байрона, написала первую в истории человечества программу для Аналитической машины. Российское программирование имеет богатую историю, ведущую свой отчет от 50-х годов. В декабре 1948 г. И. С. Брук и Б. И. Рамеев была изобретена первая в СССР ЭВМ. А первая советская ЭВМ с сохраняющейся программой создана в 1951 г. под руководством С. А. Лебедева (МЭСМ – малая электронная счетная машина). В современной России компьютерный рынок быстро увеличивался вплоть до кризиса 1998 года. Отечественная компьютерная индустрия была одной из наиболее пострадавших из-за своей зависимости от западного рынка; лишь совсем недавно Россия снова вышла на уровень 1997 года по количеству продаваемых компьютеров. Итак, программирование — процесс создания компьютерных программ. В узком смысле под программированием понимается написание инструкций на конкретном языке программирования. Задача каждого программиста – создать хорошую программу, т.е. программу, которая: надежна, универсальна, совместима, удобно читаема, эффективна.

Языки программирования – искусственные языки, созданные человеком, для описания алгоритмов решения задач. Языки программирования разделяются на 2 вида: 1. Языки низкого уровня – средства записи алгоритмов при помощи команд, понятных ЭВМ (Assembler); 2. Языки высокого уровня – средства записи алгоритма в наглядном виде (Pascal, Basic, Delphi и т.д.). Сейчас существует практически любой софт, для любых целей. Есть множество программ для офисной работы, для создания сайтов, для работы с графикой, видео и звуком. И тем не менее, программирование необходимо в любое время и для разных целей. Это нужно и при создании сайтов, и в организации работы офиса, и ещё много, когда и где.

Профессия программиста имеет свои плюсы и минусы. К безусловным плюсам можно отнести уже перечисленные выше: востребованность и высокий доход. Кроме этого, такая специальность считается престижной и даёт возможность широкого выбора места работы, как в России, так и за рубежом. В большинстве случаев характер данной работы – это кропотливый труд, долгое сидение за компьютером.

Так как компьютеры всё больше и больше вступают в нашу повседневную жизнь, то и создание программного обеспечения для них в будущем имеет большие перспективы развития.

Литература

1. Кропотов Ю.А. Вычислительная техника, микропроцессорные системы и программное обеспечение. История развития. Учебное пособие по дисциплине Введение в специальность для студентов образовательной программы 230100.62 Информатика и вычислительная техника. – Муром: Изд.-полиграфический центр МИ ВлГУ, 2013. – 117с.
2. Роберт У. Себеста. Основные концепции языков программирования. — 5-е изд. — М.: Вильямс, 2001. — 672 с.

Воронин Е.Р.

Научный руководитель: к.т.н., доцент А.А. Колпаков
Муромский институт (филиал) федерального государственного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
E-mail: kaf-eivt@yandex.ru

Разработка локальной вычислительной сети поликлиники.

ЛВС - это сети, предназначенные для обработки, хранения и передачи данных, и представляет из себя кабельную систему объекта (здания) или группы объектов (зданий). На сегодняшний день трудно представить работу любого крупного объекта без локальной вычислительной сети (ЛВС, LAN – Local Area Network). Назначение локальной вычислительной сети – обеспечить доступ к разделяемым или сетевым (общим) ресурсам (компьютеров, серверов, факсов, сканеров, принтеров и т. п.), данным и программам.

Правильно построенная ЛВС, отвечающая современным стандартам безопасности, позволяет получать доступ к необходимой информации, обеспечивает защиту от несанкционированного доступа к данным, обеспечивая в поликлинике стабильное информационное взаимодействие. ЛВС обеспечивает следующие преимущества:

- распределение данных (Data Sharing). Данные в ЛВС хранятся на сервере и могут быть доступны для чтения и записи на рабочих станциях пользователей;
- совместное использование элементов сети, доступ к локальным сетевым устройствам (принтеры, сканеры, факсы и другие внешние устройства);
- возможность быстрого доступа к необходимой информации;
- распределение программ (Software Sharing). Все пользователи ЛВС могут совместно иметь доступ к программам поддерживающим сетевой режим;
- надежное хранение и резервирование данных;
- защиту информации;
- использование ресурсов современных технологий (доступ в Интернет, системы электронного документооборота и проч.).

Packet Tracer — симулятор сети передачи данных, выпускаемый фирмой Cisco Systems. Позволяет делать работоспособные модели сети, настраивать (командами Cisco IOS) маршрутизаторы и коммутаторы, взаимодействовать между несколькими пользователями (через облако).

Успешно позволяет создавать даже сложные макеты сетей, проверять на работоспособность топологии.

Так выглядит разработанная ЛВС в симуляторе сети передачи данных Packet Tracer (рис. 1).

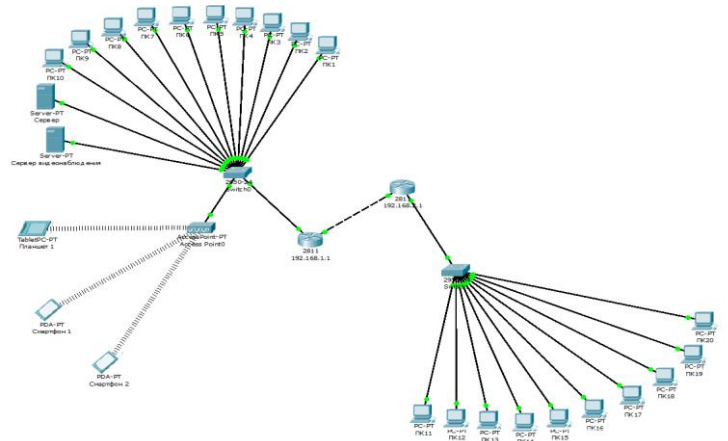


Рис. 1. Схема разработанной локальной вычислительной сети в симуляторе сети передачи данных Packet Tracer

Данная ЛВС выполняет такие задачи:

- Доступ к сети Интернет;
- Видеонаблюдение;
- Обмен данными между ПК.

При разработке в данном проекте реализована топология типа «полно связная» на основе витой пары категории 5е. Связь между зданиями обеспечивается оптическим соединением. В проекте предоставлены необходимые расчеты и чертежи, спецификация оборудования и материалов, необходимых для построения ЛВС.

В результате работы была произведена: компьютеризация рабочих мест с объединением их в локальную вычислительную сеть, с наличием сервера, сервера видеонаблюдения и доступом к сети Интернет. Выполнение данной работы обеспечило наиболее скоростную и производительную работу рабочего персонала.

Литература

1. Таненбаум Э., Уэзеролл Д. Компьютерные сети 5-е изд. – Санкт-Петербург: Питер, 2014, 960с.
2. Кандаурова Н.В. Вычислительные системы, сети и телекоммуникации.- Москва: Флинта, 2013г. – 344с.
3. Поляк-Брагинский А. Администрирование сети на примерах, 2 изд. Санкт-Петербург: БХВ-Петербург, 2010г., 432с.
4. Бройдо В., Ильина О. Вычислительные системы, сети и телекоммуникации: Учебник для вузов 4-е изд. Санкт-Петербург: Питер, 2010г., 560с.
5. Сети и телекоммуникации: методические указания по выполнению практических работ для студентов образовательной программы 09.03.01 Информатика и вычислительная техника/ сост. Белов А.А. [Электронный курс]. – Электрон. Текстовые дан.(3,3Мб) – Муром.:МИ (филиал) ВлГУ, 2016
6. Сети и телекоммуникации: методические указания по выполнению лабораторных работ для студентов образовательной программы 09.03.01 Информатика и вычислительная техника/ сост. Белов А.А. [Электронный курс]. – Электрон. Текстовые дан.(5,8Мб) – Муром.:МИ (филиал) ВлГУ, 2016

Гусенков С.В.

*Научный руководитель: ведущий электроник каф. ЭиВТ Д.В. Бейлекчи
Муромский институт (филиал) федерального государственного образовательного
учреждения высшего образования «Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
E-mail: kaf-eivt@yandex.ru*

Разработка программно-аппаратного комплекса управления устройствами IoT

В данном проекте проводятся исследование и разработка программно-аппаратного комплекса управления устройствами IoT.

«Интернет вещей» (IoT) – концепция вычислительной сети физических предметов («вещей»), оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой, рассматривающая организацию таких сетей как явление, способное перестроить экономические и общественные процессы, исключаящее из части действий и операций необходимость участия человека.

Целью данного проекта является создание сервера центрального управления и коммутатор для взаимодействия с устройствами IoT.

Разрабатываемый комплекс обеспечивает контроль и объединение в одну сеть устройств, устройств находящихся в разных точках планеты и подключенных к Интернету. Были разработаны протокол обмена, сервер и коммутатор, вместе обеспечивающие взаимодействие между устройствами IoT.

Коммутатор представляет собой программное обеспечение написанное на языке программирования C++ и C с использованием кроссплатформенной библиотеки QT 5 и обеспечивающие связь сервера и устройств IoT.

Сервер представляет собой программное обеспечение, написанное на языке программирования Python с использованием кроссплатформенного фреймворка Flask.

Особенностью данного сервера являются:

- данный проект является проектом типа OpenSource [1], поэтому любой разработчик может создать устройство для данного сервера;

- протокол FSM v2.0 [2] основан на JSON формате хранения информации и защищено цифровой подписью с использованием алгоритма ed25519[3] на основе эллиптической кривой.

Благодаря этим особенностям данный комплекс позволит обеспечить управление устройствами IoT более безопасным и простым способом.

В настоящий момент сервер обеспечивает:

- взаимодействие между устройствами, работающими по протоколу FSM v2.0;
- адресацию по 26-символьному ID;
- запуск программы устройства на сервере FSM, что позволяет перенести часть вычислительных задач устройств на сервер;
- организацию базы исходных кодов и набора инструкций для быстрой сборки программного и аппаратного обеспечения устройств IoT конечными пользователями.

Взаимодействие данного сервера тестировалось с программным обеспечением устройств на процессорах импортного и отечественного производства следующих типов:

- микроконтроллер ESP8266;
- микроконтроллер 1986BE1T;
- микроконтроллер 1986BE3T.

Разрабатываемая система управления устройствами IoT состоит из одного сервера, коммутационного оборудования и устройств. Устройство при включении посылает коммутатору пакет со своим идентификатором (id) на сервере по протоколу FSM v2.0. Коммутатор отправляет серверу HTTPS команду и добавляет в список соединений текущее соединение. Сервер определяет соответствие между id устройства и id соединения и передает сигнал подтверждения регистрации. Дальнейшие команды передаются с использованием TCP-сокета или Web-сокета защищенного SSL шифрованием. Запросы подписываются цифровой

подписью, которая проверяется на устройствах. Устройства также подписывают ответное сообщение.

Устройства комплекса делятся на следующие типы:

- устройства и ПО управления и контроля, обеспечивающие взаимодействие с пользователем и управление системой (ПО для персонального компьютер, смартфона, клавиатурная панель, дистанционный пульт);
- датчики, обеспечивающие сбор информации о состоянии «умного дома» (датчик освещенности, датчик дыма, датчик освещенности);
- исполнительные устройства, обеспечивающие управление «умным домом» (реле, блок управления питанием).

Для коммутации данных используется протокол FSM v2.0. Схема взаимодействия устройств комплекса по разработанному протоколу представлена на рис. 1.

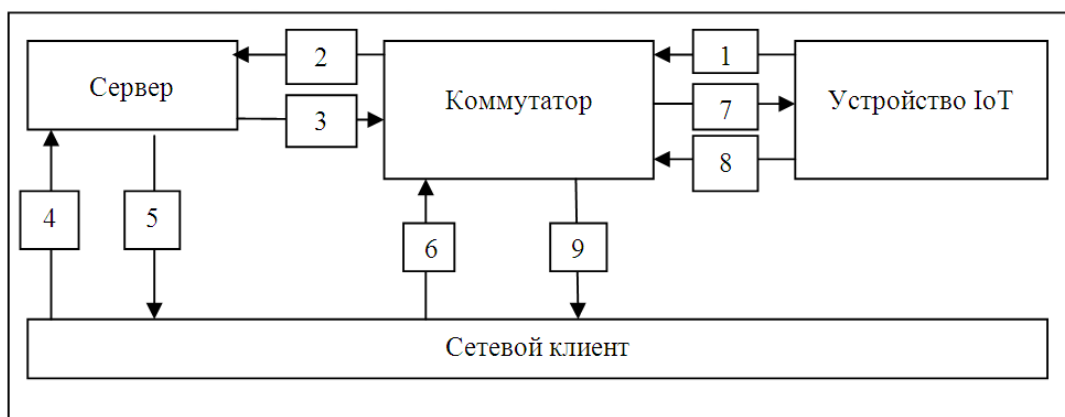


Рис. 1. Схема взаимодействия устройств комплекса по протоколу FSM v2.0.

На рисунке обозначена следующая последовательность взаимодействий:

1. Подключение и установка ID устройством.
2. Регистрация устройства на сервере коммутатором.
3. Подтверждение регистрации сервером.
4. Запрос данных устройства и коммутатора сетевым клиентом.
5. Ответ сервера на запрос данных устройства и коммутатора.
6. Отправка пакета на коммутатор с ID устройства.
7. Отправка коммутатором пакета устройству.
8. Отправка ответного пакета коммутатору.
9. Пересылка устройству ответного пакета.

Таким образом, был разработан сервер управления и коммутатор, обеспечивающие быстрый доступ к устройствам IoT через Интернет, а так же была разработана система помощи, позволяющая упростить сборку аппаратно-программного обеспечения разработчиками устройств IoT.

Литература

1. Проект коммутатора управления устройствами IoT. Открытый сервер управления разработкой ПО GitLab. [Электронный ресурс] Режим доступа: <https://git.isaynet.ru/device.isaynet.ru/FSMv2HomeSwitch>.
2. Мобильная ОС для умного дома FSM. [Электронный ресурс] Режим доступа: <http://fsmos.ru>.
3. High-speed high-security signatures [Электронный ресурс] Режим доступа: <http://ed25519.cr.yp.to/ed25519-20110926.pdf>

Жирунов А.А.

*Научный руководитель: к.т.н., доцент А.А. Белов
Муромский институт (филиал) федерального государственного образовательного
учреждения высшего образования «Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
E-mail: kaf-eivt@yandex.ru*

Исследование, разработка и оптимизация структур интернет магазина по реализации мебельных изделий

Главной отличительной чертой современных интернет-магазинов от традиционных является тип торговой площадки, на которой будет осуществлено представление информации о предлагаемых товарах, способах оплаты и доставки. Если привычному оффлайн магазину необходим торговый зал с витринами, информацией о ценах, а также персонал, включающий продавцов, кассиров и опытных консультантов, то у его «онлайн-аналога» вся вышеописанная инфраструктура будет реализована программно. Таким образом, интернет-магазин представляет собой совокупность программных модулей, функционирующих на Web-сайте интернет представительства компании, с помощью которых пользователи имеют возможность дистанционно выбрать товар из каталога и оформить на него заказ. Презентативные функции витрин и торгового зала в интернет-магазине выполняют «интерактивные страницы» с иллюстрированными каталогами товаров, а роль консультантов выполняют инструкции, описания и FAQ. Все остальные функции интернет-представительства полностью повторяют обычный магазин. Даже в интерфейсной части разрабатываемого интернет-магазина могут сохраниться привычные для большинства покупателей элементы, например такие, как виртуальная «корзина», в которую мы можем добавить выбранные товары, для дальнейшего их приобретения.

В рамках создания интернет магазина была использована бесплатная и многофункциональная система управления контентом Open Cart. OpenCart это бесплатная и современная платформа для создания индивидуального интернет - магазина. Структура OpenCart достаточно проста, однако, на базе OpenCart можно создать интернет-магазина любой сложности. Open Cart включает в себя минимальный набор инструментов при начальной установке, который дополняется по мере необходимости. Это снижает загромождение административной панели ненужными элементами, а также снижает нагрузку на сервер и экономит место на хостинге. Так же при разработке будут использованы дизайнерские приёмы, которые улучшат процесс поиска информации и оптимизируют внешний вид WEB системы.

OpenCart построен по принципу MVC и может быть установлен на любом веб-сервере с поддержкой PHP и MySQL. Как преимущества программного обеспечения официальный сайт отмечает следующие пункты:

- 1) Открытость исходного кода.
- 2) Документированность ПО.
- 3) Неограниченность категорий, продуктов и их производителей.
- 4) Неограниченность информационных страниц.
- 5) Поддержка мультиязычности и перевода интерфейса.
- 6) Возможность устанавливать собственные темы.

В результате разработки должен получиться многофункциональный интернет ресурс в виде каталога мебельных изделий с описанием и отзывами на каждый товар, а также с каталогом материалов.

Интернет магазин будет наполнен уникальным контентом, включающим фотографические изображения мебельных изделий, описание товаров и услуг, предлагаемых компаний.

В настоящее время проект интернет магазина по реализации мебельных изделий находится в стадии разработки. Дальнейшая информация и результат проделанной работы можно увидеть, пройдя по ссылке <http://мастер-про100.рф/>.

Козлов А.В.

*Научный руководитель: к.т.н, доцент А.А. Белов
Муромский институт (филиал) федерального государственного образовательного
учреждения высшего образования «Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
e-mail: kaf-eivt@yandex.ru*

Разработка сервиса для автоматизации публикации постов в Telegram

Маркетинг в социальных сетях — процесс привлечения внимания через социальные платформы. Это комплекс мероприятий по использованию социальных медиа в качестве каналов для продвижения компаний и решения других бизнес-задач. Продвижение в социальных сетях позволяет точно воздействовать на целевую аудиторию, выбирать площадки, где эта аудитория в большей степени представлена, и наиболее подходящие способы коммуникации с ней, при этом в наименьшей степени затрагивая не заинтересованных в этой рекламе людей.

Сегодня социальные сети и мессенджеры стали неотъемлемой частью нашей жизни. Много людей ведут свои каналы с различными тематиками. Чем больше становится публикаций, тем сложнее ими управлять. Существует много различных сервисов для планирования публикаций в социальных сетях и мессенджерах, большинство из которых ориентированы сразу на несколько сетей и мессенджеров. Это накладывает некоторые ограничения на структуру публикации: размер текста публикации, количество и размер вложений, ссылки, а также сильно повышает сложность сервиса, его поддержку и понижает стабильность работы. Большинство сервисов ориентированы на пользователя и не имеют публичного API для взаимодействия. Таким образом, использование сервиса при разработке другого сервиса становится невозможным, как и написание дополнительного клиентского приложения для него.

Цель данной работы - разработка сервиса "TelePost" для автоматизации публикаций в мессенджере Telegram. Сервис будет иметь следующие преимущества:

- Ориентированность на конкретный мессенджер;
- Наличие публичного API для взаимодействия и написания клиентских приложений;
- Разделение на серверную и клиентскую часть;
- Клиентское приложение – одностраничное приложение (SPA);
- Высокая производительность (использование асинхронного серверного фреймворка).

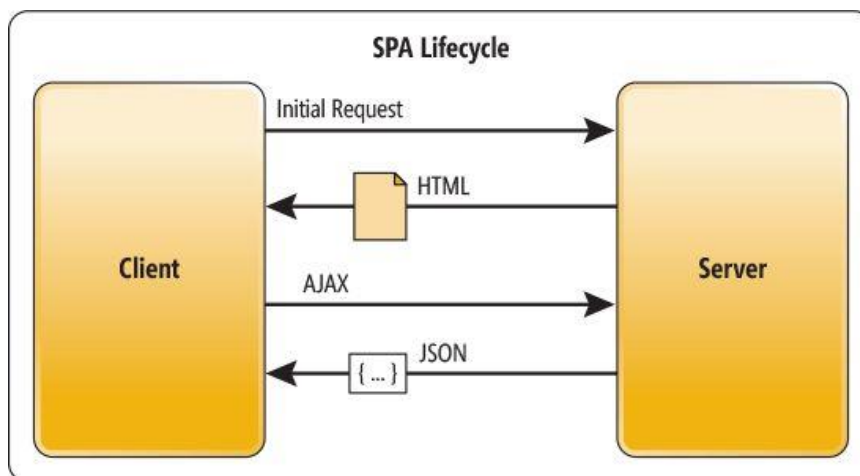


Рис. 1. Жизненный цикл SPA

К основным функциям разрабатываемого сервиса можно отнести:

- Планирование времени публикации поста;
- Выбор нескольких каналов для публикации;
- Возможность прикрепить вложения (документы, изображения, и т. п.).

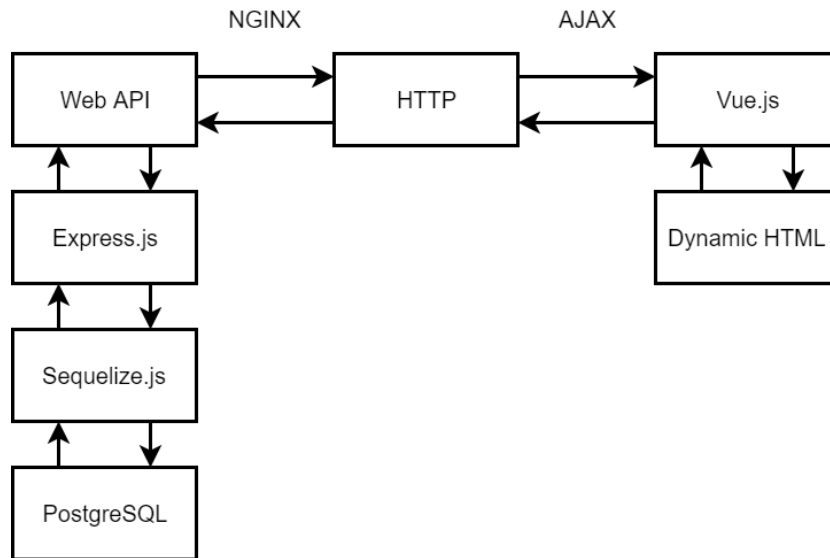


Рис. 2. Схема взаимодействия клиентского приложения с сервером

Меньшов И.С.

Научный руководитель: кандидат технических наук, А.Ю. Проскуряков
Муромский институт (филиал) федерального государственного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
e-mail: menshov.ivan1998@gmail.com

Статистический анализ тестирования прототипа торгового робота в задачах цифровой экономики

Для повышения эффективности выполнения торговых операций при любых изменениях рынка был разработан прототип автоматизированного торгового робота для спотового рынка криптовалют. Это позволяет исключить человеческий фактор, минимизировать убытки и работать постоянно без необходимости остановки с возможностью масштабирования без существенных затрат.

Основа данной системы – алгоритмы и торговые стратегии. Чтобы их реализовать используется JL-Script [4], упрощённый язык программирования, который полностью поддерживает функции и методы JavaScript [3]. Он разработан для создания сложных алгоритмических стратегий и автоматизации торговли. Создание скриптов JL Script производится внутри программы QtBitcoinTrader [5].

Для достижения максимальной эффективности системы, необходимо произвести ее установку на облачную виртуальную машину VPS (virtual private server) с бесперебойным, круглосуточным режимом работы, управляемым с компьютера из любой точки мира через Интернет.

В алгоритме робота существует три важных и определяющих его работу параметра-вектора. Были приняты обозначения X, Y, Z:

X – (Шаг цены) изменения стоимостного показателя следующего ордера относительно выполненного.

Y – (Коэффициент Объема) задействованная доля торгового актива относительно депозита.

Z – (Коэффициент профита) размер доходности по набранному торговому объему.

Наиболее сложной задачей при разработке данной системы является подбор оптимальных значений этих параметров [2].

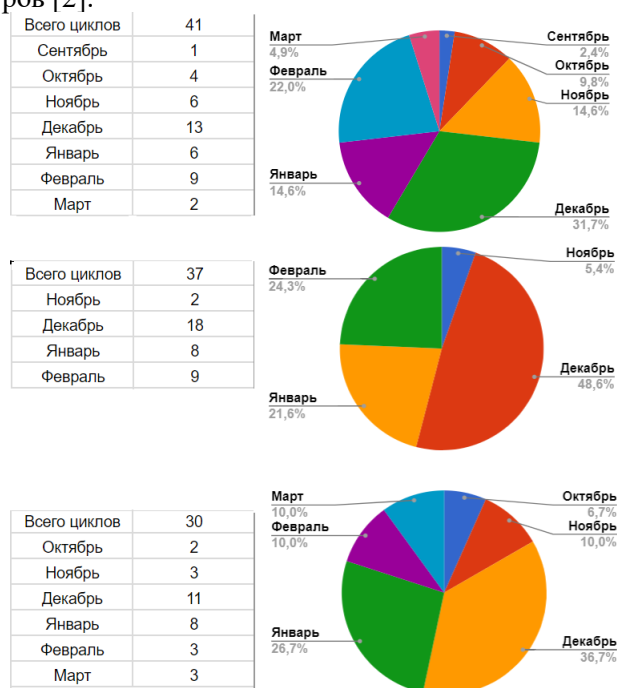


Рис. 1. Общее количество торговых циклов и их распределение по месяцам.

Для проверки стабильности кода и эффективности подобранных параметров, проводилось тестирование на трех валютных парах. На валютной паре LTC/USD тестирование проводилось 6 месяцев, за это время выполнялся 41 торговый цикл (первая диаграмма на рисунке 1). Начальные параметры, выставленные на роботе в течение сентября, показали низкую эффективность. После оптимизации параметров, было запущено тестирование на паре BTC/USD. На этой паре за 5 месяцев тестирование выполнялось 30 торговых циклов (третья диаграмма на рисунке 1). К концу октября, после тестирования новых параметров на двух торговых парах, вновь было принято решение редактировать параметры и сделать их более сглаженными. После обновления параметров, было запущено тестирование на паре ETH/USD. За 4 месяца тестирования на этой паре было закрыто 37 циклов (вторая диаграмма на рисунке 1). Таким образом, всего на период тестирования прототипа робота на всех торговых парах было выполнено успешно 108 циклов, что показало прирост в среднем около 27% от депозита (рисунок 2).

Если производить подборку параметров эмпирически в режиме реального времени, то уйдут годы. За это время рынок изменится, и настройки потеряют актуальность. Следовательно, необходимо разработать программную платформу для тестирования торговых алгоритмических стратегий, которая будет задействовать базу данных стоимостных показателей максимальной точности (минимального таймфрейма) и вычислять наиболее оптимальные параметры экспериментально для конкретного инструмента.

2017 год				
0,007129788811	BTC	171,1	USD	17% с 3.10 по 31.12
2018 год				
0,004204513075	BTC	87,55	USD	5,48%
2017 год				
0,32803795	ETH	678,02	USD	20% с 25.11 по 31.12
2018 год				
0,2217254765	ETH	928,31	USD	10,70%
2017 год				
2,9303142	LTC	575,13	USD	36% с 25.09 по 31.12
2018 год				
0.8519214515	LTC	417.75	USD	10.61%

Рис. 2. Статистика профита за 6 месяцев.

Статистический анализ в ходе тестирования системы при различных параметрах показал, что подбор оптимальных параметров – это чрезвычайно важный и сложный шаг разработки. Несмотря на то, что робот, при текущих параметрах стабильно функционирует и приносит прибыль, что отражено на рисунке 2, необходимо создание программной платформы для тестирования и проведения исследований для поиска оптимальных параметров.

Литература

1. A. Proskuryakov. Intelligent System for Time Series Forecasting. XII International Symposium Intelligent Systems 2016, INTELS 2016, 5-7 October 2016, Moscow, Russia. Procedia Computer Science. [<http://dx.doi.org/10.1016/j.procs.2017.01.122>] Volume 103, 2017, Pages 363–369.
2. Кобзарь, А.И. Прикладная математическая статистика. Для инженеров и научных работников/ А.И. Кобзарь. ФИЗМАТЛИТ. – 2006. – 816 с.
3. Дэвид Флэнаган. JavaScript. Подробное руководство, 6-е издание. Символ-Плюс 2012.
4. JL Script - Скриптовый язык для автоматизации торгов [Электронный ресурс]. URL: <http://forum.centraibit.com/viewtopic.php?p=4181> (дата обращения 26.03.2018).
5. Qt Bitcoin Trader - программа для управления счетами на биржах. URL: <https://github.com/JulyIGHOR/QtBitcoinTrader> (дата обращения 26.03.2018).

Миловидов А.Е.

Научный руководитель: к.т.н., доцент А.Ю. Проскуряков
Муромский институт (филиал) федерального государственного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
e-mail: alekseimilovidov@mail.ru

Анализ и обработка временных рядов стоимостных показателей с применением вейвлет-преобразования

В настоящее время некорректное принятие инвестиционных решений приводит к увеличению рисков и экономическим потерям. Причинами этому являются несвоевременное получение информации и значительная погрешность анализа. И если крупные компании с высоким оборотом могут позволить себе иметь аналитический отдел, то средний и малый бизнес, а также частные лица, нуждаются в эффективном инструменте для анализа и оценки рынка и получения максимальной отдачи от вложенных инвестиций.

Для эффективной работы с временными рядами показателей финансовых систем необходимо понимать, что изменение динамических параметров в этих рядах подчиняется определенным законам, правилам, теориям, циклам. Со стороны инвесторов это баланс спроса и предложения. С точки зрения фундаментального анализа это новостной фон, политическая и экономическая обстановки, настроение рынка. В свою очередь, со стороны математики такие ряды можно анализировать с помощью волновой теории Эллиотта, чисел Фибоначчи, трендовых направлений и т.д. [2]

Сложность применения волнового анализа состоит в том, что временные ряды изменений стоимостных показателей зашумлены всплесками и флуктуациями, которые могут быть вызваны влияниями крупных инвесторов, эмоциональным восприятием социума, техническими сбоями бирж, а также внебиржевыми сделками. Для устранения данной проблемы в работе рассматривается обработка временных рядов финансовых показателей с помощью вейвлет преобразования. Применение вейвлет обработки позволяет аппроксимировать временной ряд, представляя его в более удобном для дальнейшего основного анализа виде. [3]

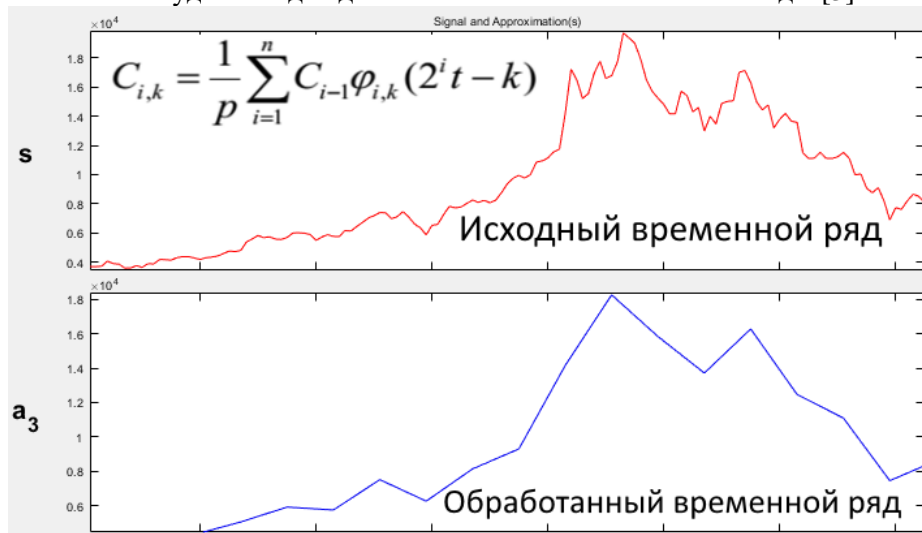


Рис. 1. Применение вейвлет-обработки

Вейвлет фильтрация применяется для обработки исходного временного ряда с целью повышения качества представления сигнала. Для этого осуществляется обработка детализирующих коэффициентов с помощью трешолдинга. Были проведены исследования различных методов выбора оптимальных параметров пороговой обработки. Сравнительный анализ различных критериев показал, что наиболее эффективным методом обработки

детализирующих коэффициентов при максимальном отношении сигнал/шум является применение критерия адаптивного штрафного порога. В результате обработки на полученный временной ряд можно нанести волновую разметку согласно правилам и алгоритмам циклической теории Эллиотта.



Рис. 2. Нанесение волновой разметки на график валютной пары

В результате применения данных алгоритмов на график валютной пары наносится волновая разметка с обозначением основных уровней, как показано на рисунке 2.

Литература

1. Роберт Пректер, Альфред Фрост Волновой принцип Эллиотта. Ключ к пониманию рынка / Elliott Wave Principle: Key to Market Behavior. — М.: «Альпина Паблишер», 2012. — 270 с. — ISBN 978-5-9614-1837-8.
2. A. Proskuryakov. Intelligent System for Time Series Forecasting. XII International Symposium Intelligent Systems 2016, INTELS 2016, 5-7 October 2016, Moscow, Russia. Procedia Computer Science. [http://dx.doi.org/10.1016/j.procs.2017.01.122] Volume 103, 2017, Pages 363–369.
3. Миловидов А.Е., Ярошенко А.И., Меньшов И.С., Проскуряков А.Ю. Анализ и обработка временных рядов финансовых показателей // Topical areas of fundamental and applied research XII: Proceedings of the Conference. North Charleston, 5-6.06.2017, Vol. 1—North Charleston, SC, USA: CreateSpace, 2017, p. 211,138-140 p.

Морозов Н.Д.

*Научный руководитель: ст. преподаватель Н.Е. Холкина
Муромский институт (филиал) федерального государственного образовательного
учреждения высшего образования «Владимирский государственный университет имени
Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
E-mail: mnardia1996mor@gmail.com*

Разработка игры «Веселая история»

Игра представляет собой 3D платформер, в котором игроку предстоит добраться из пункта А в пункт В. Действие происходит на летающих платформах в небе. Чтобы дойти до конца уровня игроку необходимо выполнить задание советника, сразиться с противниками, и преодолеть ловушки.

Разработка игры проходила на бесплатном движке Unity 3D. Unity 3D – межплатформенная среда разработки компьютерных игр. Она позволяет создавать приложения, работающие под различными операционными системами, в том числе под Windows, Android, IOS.

Основные программы, использованные при разработке игры: Blender и 3D Coat. Blender – freeware программное обеспечение для создания трехмерной компьютерной графики, включающее в себя средства создания 3D моделей, анимации, рендеринга, постобработки и монтажа видео со звуком. С его помощью можно добавлять визуальные эффекты на созданные модели: подсветку, эффект матовости (глянцевости) поверхности и др. 3DCoat предоставляет набор инструментов для создания скульптуры, топологии, UV-карт и текстурирования модели; позволяет проводить визуализацию статичных сцен и круговую анимацию модели.

Для создания игровых объектов и персонажей использовалась программа Blender. Создание скелета персонажей, настройка весов для последующего анимирования, создание материалов модели, настройка и рендеринг сцены так же происходил в Blender. Создание материалов и послойное текстурирование моделей происходило в программе 3DCoat. Сборка сцены, создание анимации персонажей, создание системы частиц, настройка освещения, запекание карты путей, оптимизация и сборка готового приложения производилась в Unity3D.

В результате удалось разработать игровое приложение под Windows, включающее один игровой уровень, основного персонажа, советника и два типа противников. Персонаж, под управлением игрока, перемещается по сцене, ищет и собирает предметы, защищается от нападения противников и, обходя ловушки, приходит в конечный пункт. Поставленная перед разработчиком задача успешно выполнена.

Паршин К.Н.

Научный руководитель: к.т.н., доцент А.Ю. Проскуряков
Муромский институт (филиал) федерального государственного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
e-mail: c.parshin2013@yandex.ru

Разработка простейших игровых приложений на примере создания игры «Три в ряд»

В работе исследуется вопрос создания и разработки простейших игровых приложений на примере «Три в ряд». Приведена реализация этапов прорисовки поля, возможности сохранения и загрузки.

Ключевые слова: подпрограмма, компоненты Delphi, диалоги, разработка, компьютерные игры

Разработку игрового приложения в первую очередь необходимо спланировать. Механика игры, написание кода и отрисовка должны быть заранее predeterminedены и представлены наглядно в плане работ. Наиболее удобный и репрезентативный способ представить план – это графическое представление. Графическое представление в этом случае может содержать блок-схемы, графики и таблицы. Затем идёт отрисовка поля и произвольное расположение «рун». В процессе разработки игры эта идея реализуется с помощью следующих функций:

```
for x:=0 to 11 do  
for y:=0 to 11 do  
if (a[x,y]=0) and (ObjectClick=true) then  
begin  
randomize;  
z:=random(4);  
a[x,y]:=z;  
Form1.Imagelist1.GetBitmap(z,BMP);  
Form1.Image1.Canvas.Draw(y*32,x*32,BMP);
```

Результат этих функций – игровое поле, которое представлено на рис.1. Если на игровом поле не будет возможности хода, то «руны» перетасуются.

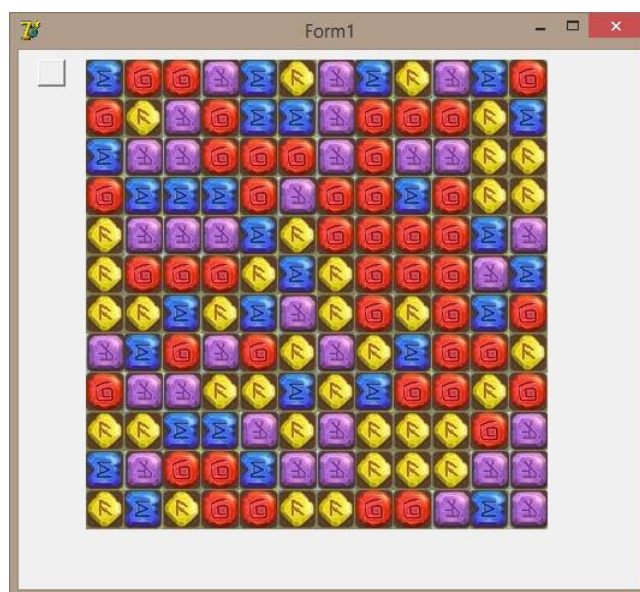


Рис.1 Игровое поле

После реализации игрового поля необходимо добавить возможность сохранения и загрузки игры. Для этого используются два компонента TOpenDialog и TSaveDialog, которые отвечают

за открытие и сохранение файлов. Процесс реализации сохранения игры представлен следующими функциями:

```
SaveDialog1.Filter := 'Сохраненная игра|*.sav';  
SaveDialog1.DefaultExt := '*.sav';  
if SaveDialog1.Execute then begin  
  if FileExists(SaveDialog1.FileName) then  
    if Application.MessageBox('Такой файл уже существует. Перезаписать?',  
'Вопрос', mb_YesNo) <> 6 then Exit;  
  AssignFile(f, SaveDialog1.FileName);  
  {$I-} ReWrite(f, 1); {$I+}  
  if IOResult <> 0 then begin  
    Application.MessageBox('Ошибка записи файла!', 'Ошибка', mb_ok);
```

Графическое отображение результата данных функций и возможность сохранения и загрузки игры приведено на рис.2.

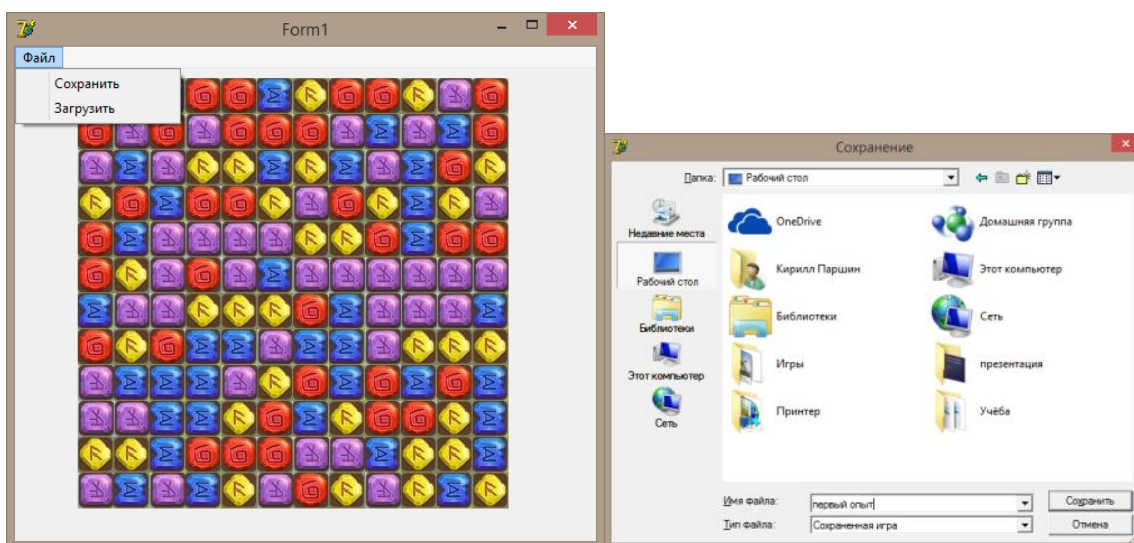


Рис.2. Сохранение и загрузка игры

Разработка простейших игровых приложений позволяет совершенствовать навыки программирования. Знание свойств компонентов Delphi или любой другой системы программирования помогают лучше ориентироваться в среде разработчика, выбирать более подходящий компонент. В дальнейшем будут добавлены новые уровни сложности и рейтинговые таблицы, подключение по сети Интернет к базе рейтингов и результатов других игроков.

Литература

1. Потопахин В.В. Turbo Pascal. Решение сложных задач. – М.: «БХВ-Петербург», 2006.
2. Демидов Д.В. Основы программирования в примерах на языке Паскаль. –М.: «МИФИ», 2010.
3. Конгер Д. Физика для разработчиков компьютерных игр. –М.: «БИНОМ», 2010. – 520 с.

Томчик А.О.

*Научный руководитель: ст. преподаватель Н.Е. Холкина
Муромский институт (филиал) федерального государственного образовательного
учреждения высшего образования «Владимирский государственный университет имени
Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
E-mail: kaf-eivt@yandex.ru*

Разработка автоматизированной системы для детского образовательного учреждения

При поступлении малыша в детский сад или ясли родители пишут заявление, заключают договор на предоставление образовательных услуг, заявление о компенсации части средств за обучение, заполняют другие нужные образовательной организации бумаги. Когда ребенок посещает садик, ведется документ (табель), отражающий пребывание в саду и потом, на его основании, выставляют родителям счет на оплату; этот же документ позволяет составлять ежедневные заявки на питание воспитанников. Отдельная задача – подготовка методических материалов, поурочных планов, сценариев, материалов для родителей.

Разрабатываемая автоматизированная информационная система «Детское образовательное учреждение» поможет вести учет посещений, формировать счета и необходимые бланки, облегчая работу персонала.

Основные функции разрабатываемой системы:

- Ведение личных дел воспитанников.
- Контроль посещаемости.
- Подготовка счетов на оплату.
- Формирование документов и их бланков.
- Анализ успехов малышей: развития речи, логического мышления, готовности к школе и др.
- Организация хранения методических наработок коллектива.

Программа позволяет вводить и корректировать данные, формировать стандартные отчеты и бланки (табель посещаемости, договор на предоставление услуг, счета на оплату, бланки для ввода различных показателей др.)

В докладе представлены результаты моделирования предметной области, приведена модель данных, рассматриваются алгоритмы реализации некоторых функций и основные отчеты.

Ярошенко А.И.

*Научный руководитель: к.т.н., доцент А. А. Колпаков, аспирант С.В. Савинов
Муромский институт (филиал) федерального государственного образовательного
учреждения высшего образования «Владимирский государственный университет имени
Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
e-mail: artawower@protonmail.com*

Автоматизированный органайзер на Raspberry PI3

Raspberry PI3 (рис. 1) представляет собой бюджетный одноплатный компьютер небольших размеров, оснащенный процессором BCM2837, 1 Гб оперативной памяти и обладающий такими интерфейсными разъемами как: 3.5 мм Jack, HDMI, Ethernet, Wi-fi, USB 2.0 x4, GPIO, Micro USB, Bluetooth 4.1 (рис. 2). Данный микрокомпьютер был выбран в качестве основного устройства благодаря своей относительно невысокой стоимости, низким энергопотреблением, поддержкой по умолчанию необходимых для разработки языков высокого уровня, UNIX подобную операционную систему – Raspbian.



Рис. 1. Raspberry PI3

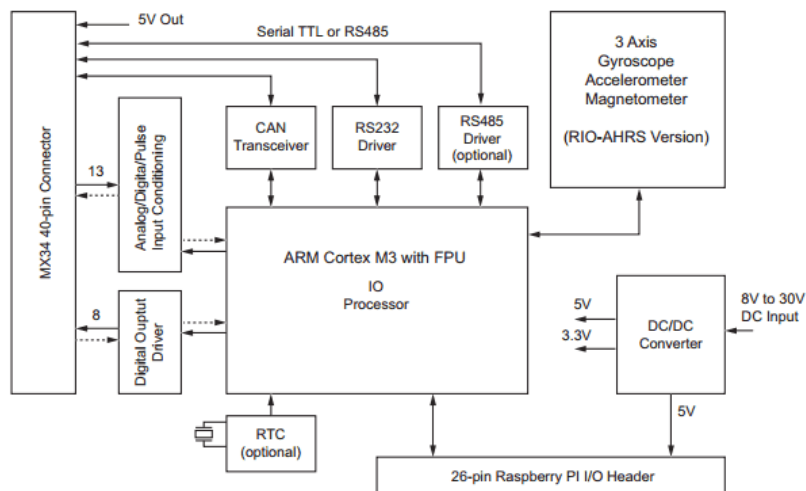


Рис. 2. Схема Raspberry PI3

Разработка программного обеспечения производилась через командную строку с помощью среды разработки VIM. В качестве языка программирования был использован Python3.6 и Golang. Для запуска задач с периодичным выполнением используется встроенная в операционную систему служба Cron, в качестве службы ведения логов используется пакет Mutt, все уведомления о внутренних ошибках поступают через программный интерфейс на почту. Для отправки email писем используется Gmail API.

Актуальность данной работы заключается в автоматизации управления рабочего процесса и личного времени, а также в решении фоновых задач с длительным периодом времени. Важным преимуществом перед другими аналогами является круглосуточная работа Raspberry PI. В отличие от работы ПК это возможно благодаря крайне низкому энергопотреблению.

Для разработки умного будильника, определяющего выходные и праздничные дни используется публичное API сервиса isdayoff.ru, модуль requests на языке Python3.6. Скрипт запускается каждый день в определенное пользователем время, перед запуском будильника идет сбор информации о том выходной день или нет. В качестве отказоустойчивости присутствует скрипт собирающий данные заранее, в случае отсутствия доступа в интернет сигнал будильника срабатывает из кэшируемых данных. При наличии подключения к сети интернет в качестве звукового сигнала для будильника выступает случайное звуковое сопровождение загруженное через Spotify API, при его отсутствии случайный звуковой трек, удовлетворяющий условие регулярного выражения берется из файловой системы. При срабатывании сигнала подается напряжение на 16 pin GPIO, приводящий к срабатыванию реле и включающий лампу дневного света. В необязательной конфигурации скрипта указывается количество повторений сигналов будильника, а также интервал через который он будет срабатывать.

Организованная рассылка рабочей почты достигается с помощью планировщика задач Cron, почтового клиента Mutt, программного обеспечения написанного на Bash, конвертора html файлов в PDF – Wkhtmltopdf. Данные для формирования отчетов берутся с приватного репозитория BitBucket, после чего формируется html файл, текстовые данные берутся из commit сообщений на удаленном сервере, графические изображения создаются с помощью Python библиотеки Matplotlib. Отчеты отправляются с указанной в скрипте периодичностью.

Для автоматизации сбора информации с помощью парсинга данных с публичных сайтов используется язык высокого уровня Golang. Для изменения ip адреса используется скрипт написанный на данном языке, переключение осуществляется запущенным в фоне процессом через openvpn, данные о доступных VPN серверах собираются при каждом 10ом запуске, что позволяет избежать простоев. Данные собираются с помощью библиотеки goquery, позволяющему искать по вложенным тегам внутри DOM дерева загруженной страницы.

Литература

1. Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
2. Хахаев И.А. Практикум по алгоритмизации и программированию на Python. – М.: Альт Линукс, 2010. — 126 с.
3. Марк Саммерфильд. Программирование на языке GO. МСК.: ДМК Пресс, 2016. – 556 с