

Правительство Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
**«Национальный исследовательский университет
«Высшая школа экономики»**

Факультет информационных технологий и вычислительной техники

Вычислительные машины, системы, комплексы и сети
(Специализация)
Информационно-коммуникационные технологии
(кафедра)

ДИПЛОМНАЯ РАБОТА

Разработка аппаратно-программного комплекса для передачи оповещений в
системе контроля доступа.

Выполнил

Студент группы № 104

Медведев Сергей Михайлович
(Ф.И.О.)

Научный руководитель

Сафонов Сергей Николаевич
(должность, степень, звание,
Ф.И.О.)

Консультант

Сафонов Сергей Николаевич
(должность, степень, звание,
Ф.И.О.)

Москва, 2013

Аннотация

В дипломном проекте был разработан аппаратно-программный комплекс для передачи оповещений в системе контроля доступа. В ходе выполнения были рассмотрены существующие системы «Умный дом», составлены функциональные требования, разработана структурная схема. Для передачи оповещений по SMS было решено использовать SMS-шлюз, а для отправки оповещений по электронной почте библиотеку smtplib языка Python. Для реализации кнопки дверного звонка использовалась аппаратная плата Arduino micro. Аппаратная плата Arduino была запрограммирована, на языке Python была написана управляющая программа. Также был настроен веб-сервер Apache и на нем запущен демонстрационный сайт. Клиент и сервер были соединены с помощью виртуальной частной сети. Полученный комплекс был протестирован на соответствие функциональным требованиям.

Оглавление

Оглавление

Аннотация.....	2
1. Введение.....	4
1.1 Актуальность.....	4
2. Обзорно-аналитическая часть.....	5
2.1 Постановка задачи.....	5
2.2 Обзор аналогов разрабатываемого комплекса.....	7
2.3 Выбор программных и аппаратных компонентов.....	9
2.3.1 Оценка требований.....	9
2.3.2 Выбор модели платы Arduino.....	11
2.3.3 Выбор веб-сервера.....	13
2.3.4 Выбор CMS.....	14
2.4 Выбор поставщиков сервисов, используемых в проекте.....	17
3. Разработка.....	19
3.1 Установка Apache.....	19
3.2 Установка и настройка Joomla.....	21
3.2.1 Установка Joomla.....	21
3.2.2 Ограничение доступа стандартными средствами Apache.....	22
3.3 Построение виртуальной частной сети.....	27
3.4 Реализация кнопки на Arduino.....	31
3.5 Программирование Arduino.....	33
3.6 Создание управляющей программы.....	35
3.6.1 Структура программы.....	35
3.6.2 Функция mail.....	37
3.6.3 Функция send_sms.....	37
3.6.4 Функция button_state.....	37
3.6.5 Функция parse.....	38
3.6.6 Функция capt_f.....	38
3.6.7 Функция start_sending.....	39
3.6.8 Демон.....	39
4. Экспериментальная часть.....	40
4.1 Функциональное тестирование прототипа.....	40
5. Охрана труда.....	41
5.1 Введение.....	41
5.2 Вредные факторы электромагнитного излучения, влияющие напрямую на здоровье человека.....	43
5.2.1 Общая характеристика РЧ – воздействий.....	44
5.2.2 Воздействие на здоровье РЧ-полей от 1 МГц до 10 ГГц.....	45
5.3 Методы и средства защиты пользователей от воздействия РЧ-излучения.....	46
5.4 Методы и средства защиты от поражения электрическим током.....	47
5.5 Выводы.....	48
6. Заключение.....	49
6.1 Перспективы развития.....	50
6.2 Выводы.....	51
Список литературы.....	52

1. Введение

1.1 Актуальность

Система "Умный дом" – это централизованное управление такими системами как:

1. Система освещения
2. Система безопасности
3. Система аудио/видео, домашний кинотеатр (мультирум)
4. Система кондиционирования
5. Система отопления
6. Система теплых полов
7. Система водоснабжения
8. Управление шторами, жалюзи
9. Система распределения электронагрузок.

Стоимость таких систем не по карману многим потребителям, да и потребность во всех функциях системы «Умный дом» возникает не у всех. Одной из наиболее интересных функций данных систем является «умный домофон», позволяющий оповещать о посетителях даже тогда, когда хозяина нет дома. Кроме того, данная функция может оказаться полезной не только дома, но и на малых или средних предприятиях.

Задачей дипломного проекта стоит создание аппаратно-программного комплекса для передачи оповещений в системе контроля доступа, представляющий из себя бюджетную и ограниченную функционально альтернативу системам «Умный дом». Центром системы будет персональный компьютер, к которому подключены устройства, за которыми осуществляется наблюдение. Управление будет вестись через вебсайт. В качестве примера

системы доступа будет использоваться веб камера и контроллер usb с кнопкой, условно расположенные на входной двери.

2. Обзорно-аналитическая часть

2.1 Постановка задачи

В данной дипломной работе необходимо разработать комплекс, оповещающий администратора о событии, происходящем в системе контроля доступа, а именно о людях, проходящих или использующих систему. В качестве устройства, с помощью которого пользователь будет принимать оповещения, планируется использовать смартфон или коммуникатор.

Центром комплекса предполагается сделать ПК, на котором будет запущена программа-служба, выполняющая фиксирование события и отправку оповещений. Конфигурирование работы службы будет выполняться путем занесения изменений в конфигурационный файл.

Необходимо добавить возможность оповещения несколькими способами, в зависимости от требований администратора, а именно по электронной почте и по SMS. Короткие сообщения SMS передаются по сетям GSM, использование которых возможно в местах, где недоступен широкополосный доступ в интернет или мобильный интернет. SMS, как правило, доставляются в течение нескольких секунд, в одно сообщение можно поместить до 160 латинских символов, чего вполне хватает для размещения веб-ссылки. Просмотр захваченных снимков будет выполняться с помощью веб-браузера, что позволит избежать установки дополнительного программного обеспечения на коммуникатор. Необходимо также добавить возможность удаления ненужных материалов с помощью веб-браузера.

В ходе разработки приоритетом является себестоимость комплекса при сохранении требуемой функциональности, поэтому при проектировании приветствуется использование свободных и бесплатных программных

продуктов с открытым исходным кодом.

Данная система окажется полезной в домашних условиях и на малых предприятиях в тех случаях, когда они не могут позволить себе полноценные системы SmartHouse или не нуждаются во всех функциональностях данных систем. При разработке необходимо обратить внимание на надежность системы, простоту настройки и эксплуатации, стоимость. Разрабатываемый комплекс должен соответствовать всем нормам и ГОСТам РФ.

2.2 Обзор аналогов разрабатываемого комплекса

Проводя обзор аналогов разрабатываемого устройства, стоит рассмотреть наиболее распространенные системы.

Система Умный Дом SHS состоит из контроллеров (базовых устройств). Контроллеры системы SHS - это устройства управления, разработанные специально для монтажа системы "Smart Home System", с помощью которых осуществляется подключение и управление внешних устройств.

Каждый контроллер (модуль) имеет необходимый набор функций, которые определяют значимость и функциональность устройства в рамках сети. Базовые настройки, заложены в автономные возможности каждого контроллера. То есть совсем не обязательно делать все функции в квартире "умными". Можно начать с любого объекта, подключив его не через обычный выключатель или розетку, а создать управление через контроллеры системы SHS.

Интеллект в доме появляется, когда все эти устройства начинают обмениваться информацией друг с другом. Те минимальные недостающие знания для каждого устройства становятся доступными. Объединение модулей в систему расширяет возможность отдельных устройств, и создает принципиально новые возможности, которые нельзя получить, используя автономные устройства.

Все контроллеры в сети равноправны и нет главного управляющего контроллера. В случае отказа любого контроллера, все остальные продолжают свою работу. Тем самым достигается достаточно высокая "живучесть" сети.

Каждый модуль SHS системы в отдельности, строго выполняет свою автономную функцию. Контроллер имеет свою узкую программу управления. Микропрограмма контроллера построена изначально для обеспечения максимального комфорта и удобства управления конкретным устройством. В связи с этим контроллеры не имеют "запасных" не используемых ресурсов. Начальная функциональность устройств работает без подключения к сети

(автономно). Стоимость от 2500 €

LanDrive2 — это универсальная платформа для построения классических шинных распределённых систем управления в системах умный дом, в автоматизации зданий. Разрабатывается компанией INSYTE на базе протокола Modbus/RTU. Предназначена для управления внутренним и уличным освещением, силовыми нагрузками, электроприборами, а также такими системами как отопление, кондиционирование, вентиляция, охранная сигнализация, контроль доступа, контроль протечек воды. Также возможно управление: аудио и видеоаппаратурой, жалюзи, рольставнями, шторами, воротами, насосами, двигателями.

Структура системы.

Система состоит из центрального контроллера и исполнительных модулей, связанных между собой полевой шиной (сетью). К исполнительным модулям подключаются управляемое оборудование. Вместо центрального контроллера может быть использован ПК с установленным Modbus OPC сервер, в этом случае управление системой производится с компьютера. Также возможна интеграция в любую SCADA систему. Стоимость заказа и установки системы от 1000 €.

2.3 Выбор программных и аппаратных компонентов

2.3.1 Оценка требований

Для решения задачи необходимо следующее оборудование:

- Персональный компьютер
- Веб камера
- Микроконтроллер с кнопкой
- Соединение с интернет

Наиболее распространенными и универсальным интерфейсом для подключения периферийных устройств к ПК на данный момент является USB, поэтому веб-камера и микроконтроллер должны иметь USB интерфейс. Обычно для решения подобных задач используются ip-камеры, однако цена их достаточно велика, цена на самую простую ip-камеру приблизительно в 10 раз превышает цену на самую простую веб-камеру. Недостаток простых веб-камер с интерфейсом USB, собственно, заключается в расстоянии, на которое можно подключить камеру. В спецификации USB указана максимальная длина кабеля 5 метров. Для увеличения длины кабеля придется использовать повторители. Еще одним из решений проблемы длины кабеля может стать использование активного USB-удлинителя фирмы Omix, позволяющего обеспечить удлинение до 30 метров без применения блока питания и до 60 метров с использованием внешнего блока питания. Активный удлинитель USB 1.1 Omix представляет собой два блока. Один из них подключается к USB-разъему компьютера, а ко второму блоку подключается USB-устройство. Между двумя блоками удлинителя подключается кабель - стандартная витая пара категории не ниже 5е, обязательно экранированная.

Для избежания проблем с пайкой и программированием микроконтроллера проще всего использовать одну из плат Arduino или ее аналогов.

Оповещения будут происходить с помощью sms и e-mail сообщений,

пользователь сможет выбирать нужный вариант по своему желанию либо одновременно оба. В качестве операционной системы для управления комплексом будет использоваться GNU/Linux, т. к. он бесплатен, не тратит много ресурсов и удобен для разработки и использования функций разрабатываемой системы. Кроме того, понадобится и веб сервер.

Ядром разрабатываемого комплекса будет являться демон (в системах класса UNIX – служба, работающая в фоновом режиме без прямого общения с пользователем).

Данный демон запускается при старте системы (или вручную администратором) и ждет сигнал, поступающий с кнопки на микроконтроллере USB.

После получения сигнала демон считывает конфигурационный файл, в котором указаны необходимые пользователю параметры работы системы, делает снимок с камеры, помещает его в каталог на веб-сайте, формирует ссылку на него и выполняет функции отправки, заданные в конфигурационном файле. Пользователю в сообщении приходит информация о времени совершения события и ссылка на фото.

Для демонстрации фотографий и конфигурации параметров системы нужен веб сервер. Пользователь сможет перейти по ссылке в сообщении и загрузить фотографию, если ему это необходимо. Для предотвращения нежелательной утечки информации, для доступа к фотографиям необходимо авторизоваться на сайте. Для изменения параметров системы, администратор может использовать форму на сайте, параметры из которой заносятся в конфигурационный файл демона.

Демон написан на языке программирования Python. Отправка сообщений по электронной почте производится с помощью библиотеки `smtplib`, которая позволяет выполнять функции почтового клиента. Отправку сообщений по sms можно производить с помощью мобильного телефона или с помощью SMS-шлюза (интерфейс, позволяющий отправлять и получать SMS-сообщения без

использования мобильного телефона). В случае с телефоном необходимо, чтобы телефон был в зоне обслуживания оператором сотовой связи, батарея заряжена и на балансе было достаточно средств. Отправка через SMS-шлюз производится через интернет, и не зависит от состояния мобильного телефона, кроме того компании, обслуживающие SMS-шлюзы предоставляют разработчикам API (application programming interface) для интеграции их сервиса в программу. С помощью API сообщение отсылается на данный сервис, который перенаправляет его оператору сотовой связи.

2.3.2 Выбор модели платы Arduino

Рассматривая оригинальные платы Arduino можно столкнуться с проблемой выбора нужной, ведь ассортимент моделей достаточно велик. Главные отличия между ними – используемый микроконтроллер, а значит функциональность, внешние интерфейсы, а также объем памяти и физические размеры. Приоритетными критериями в выборе платы для данного проекта становятся:

1. Наличие usb порта
2. Габариты
3. Цена

Модель	Цена, руб	Наличие встроенного USB	Габариты, мм	Микроконтроллер	Флеш память, кб	Дополнительно
Mini	990	Нет. Нужен USB-Serial адаптер	43x18	ATmega328	32	Является аналогом Uno в компактном варианте. Для использования нужна пайка.
Uno	1150	Есть	68.6 x 53.3	ATmega328	32	Самая популярная версия базовой платформы Arduino

						USB.
Leonardo	1150	Есть	68.6 x 53.3	Atmega32u4	32	Аналог Uno, но на микроконтроллере Atmega32u4, который также выполняет ф-цию USB-UART преобразователя для прошивки.
Micro	1190	Есть	48 x 17.7	Atmega32u4	32	Является аналогом Leonardo в компактном варианте. Нет гнезда для питания.
Nano	1990	Есть	43 x 18	ATmega328	32	Является аналогом Uno в компактном варианте. Нет гнезда для питания.
Ethernet	2290	Нет. Нужен USB-Serial адаптер	68.6 x 53.3	ATmega328	32	Uno со встроенным Ethernet адаптером
Mega 2560	2290	Есть	101.6 x 53.3	ATmega2560	256	Расширенная версия Arduino Uno. Больше контактов и аппаратных serial-портов.
Duo	2590	Есть 2	101.6 x 53.3	AT91SAM3X8E	256	Мощная плата с новым процессором и большим числом контактов и входов

В ходе сравнения различных моделей Arduino была выбрана Arduino Micro, т.к. она имеет встроенный USB, обладает малыми размерами, приемлемой ценой и при этом тем же функционалом, что и другие платы той же ценовой категории, но больших габаритов.

2.3.3 Выбор веб-сервера

Основными критериями сравнения веб-серверов стали следующие

параметры:

- Операционные системы
- Лицензия (проприетарное или свободное ПО)
- Имеется опыт работы с данным сервером

Основные параметры выбора это:

- Надежность
- Работа под Linux
- Бесплатность
- Наличие хорошей документации
- Наличие личного опыта работы

Название	Разработчик	Лицензия	ОС	Особенности
Apache HTTP Server	Apache Software Foundation	Apache	Кроссплатформенное	Упор на надёжность и гибкость.
nginx	Игорь Сысоев	BSD	Кроссплатформенное	Разрабатывался для испытывающих большую нагрузку серверов. Включает в себя почтовый прокси-сервер.
Internet Information Services	Microsoft	EULA	Microsoft Windows	Является частью пакета IIS. Поддерживает .NET
XSP	Mono	GPL	Кроссплатформенное	Является частью пакета Mono. Поддерживает .NET

Как видно из представленной таблицы всем требованиям удовлетворяет только Apache.

2.3.4 Выбор CMS

CMS – Content Management System (Система управления сайтом) программное обеспечение, позволяющее управлять содержимым и структурой

сайта.

Основными критериями сравнения CMS стали следующие параметры:

- Лицензия (проприетарное или свободное ПО)
- Сложность
- Имеется опыт работы с данной CMS

Основные параметры выбора CMS это:

- Надежность
- Бесплатность
- Наличие хорошей документации
- Наличие личного опыта работы

Название	Разработчик	Лицензия	Документация	Достоинства	Недостатки
Joomla!	The Joomla Project Team	GNU GPL	Предоставляется сообществом	Высокая надежность. Функционал расширяется с помощью многочисленных дополнений и расширений. Существует очень много готовых шаблонов.	Немного сложнее конкурентов.
WordPress	WordPress Foundation	GPLv2	Предоставляется сообществом	Функционал расширяется с помощью многочисленных плагинов.	Уязвимости и бедный функционал при стандартной

				Существует очень много готовых шаблонов, простота освоения.	установке.
Drupal	Дрис Бейтаерт и сообщество разработчиков	GNU GPL	Предоставляется сообществом	Широкий функционал, коммерческая поддержка.	Мало готовых шаблонов, сложна для неподготовленных пользователей.
1С-Битрикс	ООО «1С-Битрикс»,	Коммерческое ПО	Есть	Надежность, техническая поддержка	Потребляет много ресурсов, сложный код, функционал зависит от редакции продукта.
DataLife Engine (DLE)	SoftNews Media Group	Коммерческое ПО	Есть	Открытый код, возможность подключать сторонние модули.	Небезопасный алгоритм работы с параметрами запросов.

По представленным данным для осуществления проекта был выбран движок Joomla!

2.4 Выбор поставщиков сервисов, используемых в проекте

Для получения оповещений по электронной почте и SMS необходимо пользоваться соответствующими сервисами, которые предоставляют разные поставщики данных услуг. Так как в проекте используется библиотека `smtplib`, пользователь может пользоваться любым почтовым ящиком, в своем проекте я использую свой основной почтовый ящик на `gmail.com`.

С выбором поставщика SMS-шлюза несколько сложнее, выбор приходится делать на основе предоставляемых библиотек и тарифах. Необходима библиотека на Python, недорогой тариф либо отправка бесплатных SMS.

	Библиотеки	Цены	Возможность бесплатных SMS	Поставщик услуги
<code>sms24x7</code>	Php Python Ruby Perl javascript	31,5 коп/SMS, минимум 1 SMS	+ (с рекламой в SMS)	beeline
<code>smsimple</code>	Php	60 коп/SMS, минимальный пакет 200 SMS	-	?
<code>smspilot</code>	Php perl Python Delphi VBA	29 коп/SMS, минимум 1 SMS	-	?

Как видно, наиболее подходящим является `sms24x7`, так как их API позволяет отправлять бесплатные SMS, хоть и с рекламой, если у пользователя

закончатся средства на балансе.

Для отправки сообщений и для загрузки изображений сервер должен быть соединен с Интернет. Предоставлением доступа к Интернет занимается многочисленное число интернет-провайдеров. В зависимости от используемых провайдером технологий зависят параметры подключения. Многие провайдеры предоставляют клиенту выделенный ip за дополнительную плату, иные не предоставляют вовсе, что осложняет обращение к сайту из сети Интернет. Также компьютер пользователя может находиться за NAT (*Network Address Translation*), что осложняет вывод сайта в интернет. Для обеспечения простоты соединения между клиентом (коммуникатором) и веб-сервером можно соединить их с помощью виртуальной частной сетью (VPN), это позволит обращаться к серверу по заранее присвоенному в виртуальной сети статическому ip адресу, а также предотвратить несанкционированный доступ к серверу.

3. Разработка

3.1 Установка Apache

Веб-сервер Apache является кроссплатформенным ПО, он поддерживает операционные системы GNU/Linux, BSD, Mac OS, Microsoft Windows, Nowell NetWare, BeOS. В рамках данного проекта установка будет осуществляться на операционную систему Debian GNU/Linux. Как известно, семейство операционных систем GNU/Linux очень обширно, установка и настройка на разных дистрибутивах может несколько отличаться, однако основные моменты будут схожи во всех дистрибутивах.

Из официальных репозиторий необходимо установить пакеты apache2, mysql и php5. Пакетный менеджер установит необходимые пакеты и их зависимости. После этого нужно создать директорию /var/www/smartring/, в которой будет установлена Joomla!

Далее приведены настройки веб-сервера. На одном веб-сервере может находиться несколько сайтов.

Файл /etc/apache2/sites-available/smartring.conf

```
<VirtualHost *:80>
```

```
    ServerName smartring
```

```
    ServerAlias smartring
```

```
    ServerAdmin webmaster@localhost
```

```
    DocumentRoot /var/www/smartring
```

```
    <Directory /var/www/smartring>
```

```
        Options Indexes FollowSymLinks MultiViews
```

```
        AllowOverride None
```

```
        Order allow,deny
```

```
        allow from all
```

```
</Directory>  
</VirtualHost>
```

Включение сайта производится командой `a2ensite smarthring`, отключение производится `a2dissite smarthring`

3.2 Установка и настройка Joomla

3.2.1 Установка Joomla

Для установки Joomla необходимо иметь установленный веб-сервер Apache, PHP с поддержкой XML и Zlib. При установке php из репозитория debian мы получим готовый к использованию php, в случае с другими дистрибутивами операционной системы, например Gentoo, необходимо собрать пакет php с необходимыми USE флагами. После этого необходимо скачать дистрибутив Joomla с официального сайта.

<http://www.joomla.org/download.html>

Полученный архив перемещается в директорию с настроенным виртуальным сайтом и распаковывается. В данном случае это /vaw/www/smartring. После этого необходимо подключиться к сайту и произвести настройку через веб-браузер.

Joomla использует веб-инсталлятор. Установка состоит из нескольких шагов. На первом этапе Joomla проверяет наличие необходимых пакетов, параметры безопасности, а также рекомендуемые параметры php. Если все необходимые требования были выполнены на этапе настройки Apache, можно переходить к следующему шагу.

Далее Joomla предлагает ознакомиться с лицензией GNU/GPL, которую необходимо прочитать и принять. Данная лицензия разрешает копировать, модифицировать и распространять (в том числе на коммерческой основе) данный продукт, однако обязует распространять производные продукты на тех же условиях.

После принятия правил лицензии GNU/GPL начинается процесс настройки сайта.

На первом шаге производится настройка базы данных. Необходимо

ввести логин и пароль администратора базы данных, имя базы данных, префикс таблиц. Удаленный сервер указывать не нужно, так как он не используется. Галочками необходимо отметить пункты «Удалить существующие таблицы», «Создать резервные копии существующих таблиц», «Установить демонстрационные данные».

На втором шаге необходимо задать название сайта, которое будет отображаться в заголовке окна веб-браузера. В рамках данного проекта мы будем придерживаться названия smartring.

Шаг 3. Необходимо проверить правильность пути URL, задать электронный адрес администратора сайта и пароль для доступа к администраторской панели.

Шаг 4 уведомляет, что настройка проведена успешна. После этого Joomla автоматически удалит директорию installation с сайта, если для этого достаточно прав. В противном случае можно удалить этот каталог самостоятельно.

Теперь при обращении к сайту будет выводиться начальная страница с демонстрационными данными. Для доступа к администраторской панели используется ссылка вида <http://127.0.0.1/administrator/>

3.2.2 Ограничение доступа стандартными средствами Apache

Стоит отметить, что при размещении сайта в сети Интернет рекомендуется изменить имя администраторской директории с administrator на другую, например на unlimitedbladeworks, что уменьшит угрозу взлома, т. к. злоумышленникам необходимо не только подобрать пароль суперпользователя, но и знать путь к администраторской панели. Кроме того желательно ограничить доступ к директориям сайта с помощью .htaccess и .htpasswd .

Файл .htaccess помещается в директорию, доступ к которой необходимо

ограничить. Содержание файла приблизительно такое:

```
AuthType Basic
```

```
AuthName "Доступ ограничен"
```

```
AuthUserFile /var/www/smartring/administrator/.htpasswd
```

```
require user admin root
```

Файл `.htpasswd` помещается по адресу, указанному в `.htaccess`. Файл состоит из строк, каждая из которых соответствует паре логин/хеш пароля. Логин отделяется от хеша двоеточием. Для добавления строки с пользователем и хешем пароля используется команда `htpasswd`.

3.2.3 Установка и настройка дополнительных компонентов Joomla

CMS Joomla имеет огромное число дополнений, благодаря которым расширяется функционал сайта. Для удобного управления фотографиями необходима галерея, способная производить демонстрацию фотографий, расположенных в определенном каталоге, а также производить удаление, перемещение и скрытие материалов. Для этих целей подойдет компонент Admiror gallery.

Для установки компонента Admiror gallery необходимо скачать его с сайта разработчика. Затем в администраторской панели необходимо перейти в Расширения → Менеджер расширений выбрать скачанный файл и нажать «Загрузить и установить». Архив автоматически распакуется и установится, появится сообщение об успешной установке. Стоит добавить, что Admiror gallery распространяется по лицензии GNU GPL.

Для настройки Admiror gallery необходимо перейти на вкладку Компоненты → Admiror gallery. Прежде всего нужно задать папку, из которой будут демонстрироваться снимки. Пусть это будет `/images/webphotos/`. Средствами операционной системы данную директорию необходимо сделать доступной для чтения и записи, при желании можно ограничить доступ к ней с

помощью .htaccess. Настройки отображения можно изменять с помощью вкладок Templates и Popups.

3.2.4 Добавление необходимых модулей на сайт

Добавить компонент на страницу можно в меню Расширения → Менеджер модулей. Там указывается статус модуля, его позиция, и видимость (для всех, для зарегистрированных, для администратора). Так были добавлены следующие модули: главное меню (для всех), форма входа (для зарегистрированных пользователей).

Далее необходимо отредактировать пункты меню. Меню → Главное меню → Пункты меню → создать. На странице создания пункта меню указывается Заголовок и прочие параметры. В первой строке необходимо выбрать тип пункта меню, например для отображения страницы с галереей необходимо выбрать Gallery Layout, а для управления снимками Image manager. Для отображения пункта необходимо выбрать статус «Опубликовано», доступ можно ограничить, указав «Для зарегистрированных пользователей». Таким образом были добавлены пункты «Главная», ведущая на главную страницу с галереей и «Управление галереей» с соответствующей страницей. Оба пункта видны только для зарегистрированных пользователей. Пользователь, не прошедший аутентификацию, вместо галереи на главной странице увидит форму для входа.

Результат проделанных операций показан на скриншотах, приведенных ниже.

Главная страница для незарегистрированного пользователя:

smartring

Сообщение

- Пожалуйста, прежде пройдите авторизацию

Логин

Пароль

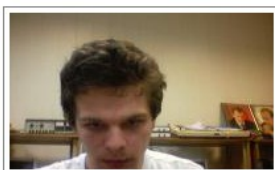
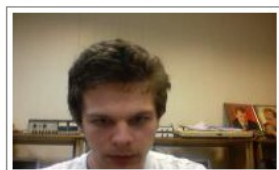
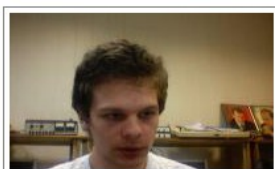
Запомнить меня

- [Забыли пароль?](#)
- [Забыли логин?](#)

©2013 smartring

Главная страница после совершения входа:

smartring



Главное меню

[Главная](#) [Управление галерей](#)

Форма входа

Здравствуйте, user,

Управление галереей:

smartring

Apply Reset

Show Sidebar

Current Folder:
/images/webphotos/

Operation With Selected Items: None

Upload images (jpg, jpeg, gif, png or zip) [Max 10 MB]: Browse...

Create folders: Add

Edit Folder Captions

2013-05-23-16-45-15	2013-05-23-16-45-38	2013-05-23-16-47-36
Visibile	Visibile	Visibile
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Priority: <input type="text"/>	Priority: <input type="text"/>	Priority: <input type="text"/>
<input checked="" type="checkbox"/> Folder Thumb	<input checked="" type="checkbox"/> Folder Thumb	<input checked="" type="checkbox"/> Folder Thumb

Главное меню

Главная Управление галереей

Форма входа

Здравствуйте, user,

Выйти

3.2.5 Добавление пользователей

Добавление пользователя проводится в администраторской панели в соответствующем пункте или через регистрацию на главной странице. В данном проекте самостоятельную регистрацию пользователей на сайте следует отключить.

Для добавления пользователя через администраторскую панель администратор должен указать имя, логин, email пользователя. Можно также задать пароль пользователя и добавить его в определенные группы.

3.3 Построение виртуальной частной сети

VPN (Virtual Private Network — виртуальная частная сеть) — обобщённое название технологий, позволяющих обеспечить одно или несколько сетевых соединений (логическую сеть) поверх другой сети (например). VPN может обеспечивать три вида соединений, узел-узел, сеть-сеть и узел-сеть в зависимости от используемых протоколов. В данном случае нам необходимо соединение типа узел-узел, одним из узлов является сервер, другим – клиент (Коммуникатор). Предоставлением услуги VPN занимается множество компаний, но лучше создать собственную сеть. Приоритетом в выборе протокола является его поддержка клиентом (Коммуникатором). Hamachi и OpenVPN не поддерживается многими коммуникаторами, либо для их работы нужно дополнительное ПО. Большинство устройств поддерживает PPTP и L2TP протоколы. В данном проекте будет использоваться PPTP в силу его меньшей сложности в настройке и требовательности в ресурсах.

Для организации VPN сети понадобится сервер, находящийся в интернете и имеющий статический адрес. В качестве VPN сервера используется имеющаяся виртуальная машина, расположенная на amazon, работающая под управлением Debian GNU/Linux. Веб-сервер и коммуникатор будут подключены к нему в качестве клиентов, таким образом будет получена виртуальная сети из трех узлов. Необходимо установить серверное ПО на VPN сервер, настроить его, а так же произвести настройку клиентов.

Для начала необходимо установить пакеты `ppp`, `pptpd` и их зависимости. Пакетный менеджер Debian сам удовлетворяет зависимости устанавливаемых пакетов. Далее необходимо настроить конфигурационные файлы. В файле `/etc/pptpd.conf` необходимо задать локальный адрес сервера в виртуальной сети, а так же диапазон адресов, выдаваемых клиентам:

```
localip 192.168.9.1
remoteip 192.168.9.102-238
```

Затем нужно добавить параметры об авторизации, шифровании и логировании в файл `/etc/ppp/pptpd-options`

```
nodefaultroute
require-mppe-128
auth
logfile /var/log/pptpd.log
```

И еще требуется добавить информацию о пользователях в файл `/etc/ppp/chap-secrets`. В этом файле можно также указать, какой ip адрес будет присвоен клиенту. Этот файл состоит из строк, в каждой строке в строгом порядке указывается имя пользователя, сервер, пароль и ip, присваиваемый клиенту. Ниже приведен пример файла для двух клиентов, первому (веб-серверу) присвоен статический адрес 192.168.9.112, а второму (Коммуникатору) будет выдаваться любой адрес из доступного диапазона.

```
tukilaptop * password1 192.168.9.112
tukidroid * password2 *
```

После этого необходимо перезапустить pptpd

```
/etc/init.d/pptpd restart
```

Чтобы VPN-сервер принимал соединения необходимо добавить 1723 порт в security group в панели управления виртуальной машиной amazon, а также добавить правила в межсетевой экран. Не стоит забывать о добавлении правил

не только в цепочки INPUT и OUTPUT, но и FORWARD, иначе подключенные клиенты не смогут общаться. При перезагрузке системы правила межсетевого экрана обнуляются, поэтому стоит сразу создать скрипт с правилами, сделать его исполняемым и поместить в /etc/init.d/netfilter

```
iptables -A INPUT -p tcp -s 0/0 --dport 1723 -j ACCEPT
```

```
iptables -A OUTPUT -p gre -m state --state RELATED,ESTABLISHED -j  
ACCEPT
```

```
iptables -A INPUT -p gre -s 0/0 -j ACCEPT
```

```
iptables -A FORWARD -i ppp+ -j ACCEPT
```

Далее описан процесс настройки vpn соединения на узле с веб-сервером.

Требуется к установке пакет pptp-linux и редактирование конфигурационных файлов. В /etc/ppp/options.pptp

lock

noauth

require-mppe-128

persist

В /etc/ppp/chap-secrets необходимо внести данные для аутентификации

```
tukilaptop * password1 192.168.9.112
```

Создать файл /etc/ppp/peers/Myvpn для подключения

```
pty "pptp 54.246.111.11 --nolaunchpppd"  
name tukidesktop  
remotename PPTP  
file /etc/ppp/options.pptp  
ipparam Myvpn
```

Подключение устанавливается по команде `pon Myvpn`

Для передачи пакетов между веб-сервером и клиентом необходимо добавить маршрут до виртуальной сети

```
route add -net 192.168.9.0 netmask 255.255.255.0 ppp0
```

Если необходимо устанавливать соединение при старте сервера, нужно дописать в `/etc/network/interfaces` строки

```
auto ppp0  
iface ppp0 inet ppp  
provider Myvpn  
up route add -net 192.168.9.0 netmask 255.255.255.0 ppp0
```

Далее приведена настройка VPN соединения на коммуникаторе на примере операционной системы Android. Меню → Параметры → Еще → VPN → Добавить VPN-сеть, в появившемся меню выбрать тип подключения PPTP, заполнить поля адрес сервера значением 54.246.111.11, включить MPPE шифрование и добавить маршрут 192.168.9.0/24

Таким образом, при получении сообщения с ссылкой на фотографию клиенту необходимо установить подключение к VPN и лишь тогда перейти по ссылке.

3.4 Реализация кнопки на Arduino

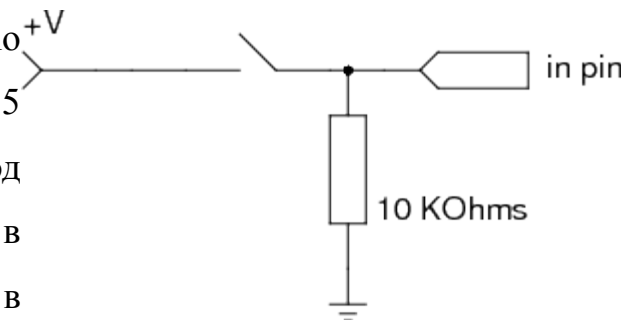
Задача собираемого устройства – реагировать на нажатие кнопки и посылать сигнал через USB интерфейс, который воспринимает программа.

В проекте используется аппаратная платформа Arduino Micro. Микроконтроллер Arduino Micro – плата микроконтроллера на базе ATmega32u4. Плата имеет 20 цифровых вход/выходов (из них 7 могут использоваться в качестве выходов ШИМ и 12 - как аналоговые входы), кварцевый генератор частотой 16 МГц, гнездо micro-USB, разъем ICSP и кнопку reset. На ней есть все, что необходимо для работы с микроконтроллером. Чтобы запустить Arduino Micro, просто подключите его к компьютеру с помощью кабеля micro-USB. Форм-фактор контроллера позволяет легко разместить его на макетной плате.

ATmega32u4 имеет встроенную поддержку USB-соединения, благодаря чему не требуется вспомогательный процессор. Это позволяет Micro появляться на подключенном компьютере в качестве мыши или клавиатуры в дополнение к виртуальному (CDC) последовательному порту (COM).

Каждый из 20 цифровых выводов Micro может использоваться как вход или как выход, используя функции pinMode(), digitalWrite() и digitalRead(). Работают при напряжении 5 В. Максимальный входной или выходной ток каждого вывода 40 мА. На каждом выводе имеется внутренний нагрузочный резистор 20-50 кОм (по умолчанию отключен).

Для подключения кнопки к Arduino необходимо подать через ключ на вход 5 вольт. Когда цепь разомкнута, вход фиксирует отсутствие напряжения. Но в данном случае вход находится в «никаком» состоянии. Причина тому — шумы, образующиеся вокруг. Чтобы



гарантировать отсутствие напряжения при разомкнутой цепи, рядом с входом ставится стягивающий резистор, через который нежелательный ток будет уходить в землю. Для стягивания используются резисторы больших сопротивлений (10 кОм и более). При замыкании цепи большое сопротивление резистора не даёт большей части тока идти в землю: сигнал пойдёт к входному контакту. Если бы сопротивление резистора было мало, при замкнутой цепи произошло бы короткое замыкание. Так как кнопка будет всегда подключена к ПК, ее питание будет осуществляться через USB.

Получившуюся схему удобно собрать на макетной плате, поместив туда устройство Arduino. В качестве цифрового входа удобнее всего использовать 4 пин, так как его удобнее всего соединить резистором с заземлением на макетной плате.

3.5 Программирование Arduino

Программирование Arduino осуществляется в специальной среде разработки Arduino IDE. Данное ПО является открытым и доступно для таких операционных систем как Windows, Mac OS, и GNU/Linux. Язык программирования Wiring схож с C. В начале программирования необходимо подключить аппаратную платформу к ПК, в среде разработки выбрать порт и платформу. ArduinoMicro является аналогом платформы Arduino Leonardo, поэтому выбрать нужно эту платформу. Выбор платформы влияет на: параметры (напр.: скорость ЦП и скорость передачи данных), используемые при компиляции и загрузке скетчей и на настройки записи загрузчика (Bootloader) микроконтроллера. Некоторые характеристики платформ различаются только по последнему параметру (загрузка Bootloader), таким образом, даже при удачной загрузке с соответствующим выбором может потребоваться проверка различия перед записью загрузчика.

В начале программы, как и в программах на C задаются переменные.

Программа (скетч) состоит из двух функций, `setup()` и `loop()`.

Функция `setup()` вызывается только один раз, после подачи питания или сброса платы, когда стартует скетч. Используется для инициализации переменных, определения режимов работы выводов, запуска используемых библиотек и т.д. В ней, с помощью функции `pinMode` задаем режим работы 4 пина платы как входа.

Функция `loop()` выполняется в цикле. В ней, с помощью функции `digitalRead` считываются значения состояния входа 4 платы (HIGH или LOW), и с помощью конструкции `if..else` выполняет действия, в зависимости от состояния входа. Если на вход подается высокий уровень (кнопка нажата), с помощью `Serial.println` на порт посылается строка символов, за которой следует перенос строки и устанавливается десятисекундный перерыв в работе программы. Если же на входе низкий уровень, программа не выполняет ничего

и цикл повторяется.

После написания программы ее необходимо скомпилировать и загрузить на устройство с помощью среды разработки Arduino. После успешной загрузки устройство нужно перезагрузить. Работу кнопки можно также проверить с помощью среды разработки Arduino, в которой имеется функция мониторинга порта. В появившемся окне при нажатии кнопки будет появляться строка, при этом между нажатиями должно пройти 10 секунд. Это нужно для избежания отправки нескольких строк при одном длительном или нескольких последовательных нажатиях на кнопку.

Блок-схема программы:



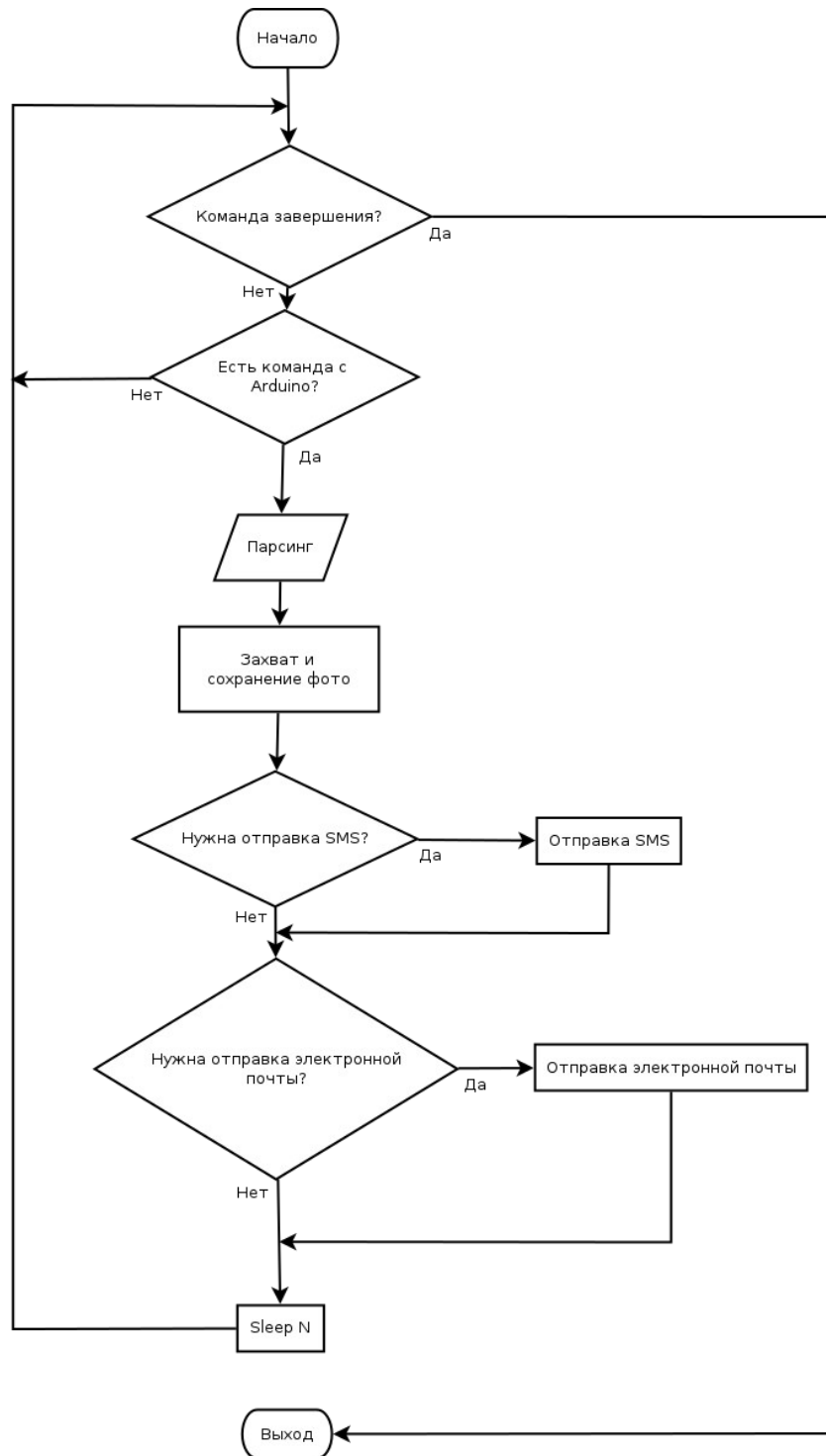
3.6 Создание управляющей программы

3.6.1 Структура программы

Задача программы – выполнять заданные действия при нажатии кнопки на микроконтроллере Arduino. При написании программы используется язык Python – высокоуровневый язык программирования, ориентированный на повышение читаемости кода. Большой плюс данного языка это множество функций, включенных в стандартную библиотеку. Кроме стандартной библиотеки в данной программе также используется библиотека, предоставляемая поставщиком SMS-шлюза, используемого в проекте.

Для удобства чтения и модификации кода программа разделена на функции. Программа представляет собой объявление функций и бесконечный цикл, в котором последовательно выполняются определенные функции. Переменные, используемые несколькими функциями целесообразно сделать глобальными. Для того, чтобы не хранить пользовательские данные и параметры (логин/пароль от почты и от сторонних сервисов) в коде программы, предусмотрен специальный конфигурационный файл `config.cfg`. Для запуска программы в режиме демона используется дополнительный `bash` скрипт.

Блок схема управляющей программы:



3.6.2 Функция mail

Данная функция служит для формирования и отправки пользователю сообщений по электронной почте. Функция использует библиотеку `smtplib`. Осуществляется подключение по протоколу `smtp` к почтовому серверу `gmail`, устанавливается шифрование, производится авторизация на сервере. Адрес и порт почтового сервера указаны в коде программы, поэтому пользователю необходимо использовать для отправки сообщений почтовый ящик на `gmail`. Ящик получателя может быть на любом сервере. Логин и пароль для авторизации на почтовом сервере, а также адрес получателя содержатся в конфигурационном файле. Сообщение формируется из пояснительного текста, даты, времени и ссылки на изображение. После отправки заголовка и сообщения соединение с сервером закрывается. Данная функция вызывается функцией `start_sending` при необходимости.

3.6.3 Функция send_sms

Эта функция использует библиотеку `sms24x7`, предоставляемую компанией `sms24x7`, которая позволяет отправлять короткие текстовые сообщения через их шлюз. Библиотека представляет собой программу на языке Python. При выполнении функция авторизуется на SMS-шлюзе с помощью `login` и посылает сгенерированное из времени и ссылки сообщение с помощью `push_msg`. Все необходимые переменные получаются из функции `parse`. Данная функция вызывается функцией `start_sending` при необходимости.

3.6.4 Функция button_state

Данная функция идет первой в цикле и выполняет простую функцию – ждет строку с кнопки на микроконтроллере Arduino. Для работы функции необходимо подключить в начале программы библиотеки `serial` и `time`. В коде функции задается порт, к которому подключена плата Arduino, устанавливается соединение с данным портом и ожидается входящая строка. После получения

строки функция фиксирует время нажатия кнопки в глобальную переменную `s_time`, которая в дальнейшем будет использоваться функциями `send_sms` и `mail` при формировании текста сообщения. После этого функция завершается до дальнейшего вызова в новом цикле.

3.6.5 Функция `parse`

Функция требует библиотеку `ConfigParser`. В ней указан путь к конфигурационному файлу, который она считывает, выполняет парсинг, берет из него значения параметров и записывает их в глобальные переменные. Считывание конфигурационного файла производится функцией `read` этой библиотеки. Библиотека `ConfigParser` поддерживает конфигурационные файлы со специальным синтаксисом. Названия секций пишутся в квадратных скобках, значения параметров идут после знака «`=`». Получение параметров функцией `get` этой библиотеки, которой первым аргументом указывается название секции, откуда считывать, а вторым имя параметра, который считывать.

Эта функция вызывается в цикле после завершения функции `button_state`, но перед выполнением других функций, благодаря чему пользователь может не перезапускать программу после редактирования конфигурационного файла: после нажатия на кнопку программа получает новые переменные из конфигурационного файла.

3.6.6 Функция `capt_f`

Функция использует библиотеки `time` и `cv`. Вызов этой функции происходит в цикле после функции `parse`. Задача этой функции – произвести снимок с веб-камеры, сохранить его в указанную директорию веб-сервера и сформировать ссылку на него, которая впоследствии будет использоваться в функциях `mail` и `sms_send`. Номер используемой камеры можно задать в конфигурационном файле.

3.6.7 Функция `start_sending`

Задача этой функции вызвать функции `mail` и `send_sms` в зависимости от параметров, указанных в конфигурационном файле в секции `notifications`. Эти параметры считываются функцией `parse`, а `start_sending` сравнивает их с определенными значениями, если параметр `sms_notification` соответствует `True`, `true` или `1`, вызывается функция `send_sms`, иначе она не вызывается. Такая же операция производится и с параметром `email_notification`, от которого зависит вызов или пропуск функции `mail`. После Функции `start_sending` следует пауза, длительность которой зависит от пользовательских настроек в конфигурационном файле. Пауза нужна для избежания повторных снимков и оповещений.

3.6.8 Демон

Демон в операционных системах Unix – процесс, не взаимодействующий с пользователем напрямую, они не имеют пользовательского интерфейса и терминала.

В операционной системе Debian GNU/Linux демоны находятся в папке `/etc/init.d/` . Запуск, остановки или перезапуск демона осуществляется с помощью его выполнения с параметром `start`, `stop` или `restart`.

4. Экспериментальная часть

4.1 Функциональное тестирование прототипа

Описание аппаратных и программных элементов, используемых при создании прототипа описаны в главе 3. Тестирование проводилось в домашних условиях, где были соблюдены все необходимые условия для запуска прототипа.

В ходе тестирования были успешно проверена работа следующих функций:

1. Редактирование параметров в конфигурационном файле в ходе работы демона.
2. Захват снимка с веб-камеры и размещение его на веб-сервере.
3. Работа веб-сервера и подключение к нему с помощью виртуальной частной сети.
4. Оповещение пользователя о размещении нового снимка с помощью SMS, отправленного через SMS-шлюз.
5. Оповещение пользователя о размещении нового снимка с помощью сообщения, отправленного на электронную почту пользователя.

5. Охрана труда

5.1 Введение

Охрана труда – это система законодательных актов, социально-экономических, организационных, технических, гигиенических и лечебно-профилактических мероприятий и средств, обеспечивающих безопасность, сохранение здоровья и работоспособности человека в процессе труда.

Основной задачей охраны труда является сведение к минимуму вероятности поражения или заболевания работающего с одновременным обеспечением комфорта при максимальной производительности труда.

Опасный фактор - это производственный фактор, воздействие которого на работающего в определенных условиях приводит к травме или другому резкому внезапному ухудшению здоровья.

Вредный фактор - производственный фактор, приводящий к заболеванию, снижению работоспособности или летальному исходу. В зависимости от уровня и продолжительности воздействия вредный производственный фактор может стать опасным.

Любой производственный процесс, в том числе работа с ЭВМ, сопряжен с появлением опасных и вредных факторов. Также при использовании разработанного аппаратно-программного комплекса используется коммуникатор со встроенным GSM-модулем, являющийся источником радиочастотного воздействия, поэтому необходимо рассмотреть влияние на человека мобильного телефона. Питание ПЭВМ производится от сети 220 В. Так как безопасным для человека напряжением является напряжение 40 В, то при работе на ПЭВМ опасным фактором является поражение электрическим током.

Исходя из анализа вредных факторов, видна необходимость защиты от них.

При эксплуатации перечисленных элементов вычислительной техники могут возникнуть следующие опасные и вредные факторы:

- Поражение электрическим током.
- Электромагнитное излучение.

5.2 Вредные факторы электромагнитного излучения, влияющие напрямую на здоровье человека.

ПЭВМ генерирует в окружающее пространство широкий спектр ЭМП различной интенсивности, в том числе:

- электростатическое поле;
- переменные низкочастотные ЭМП;
- электромагнитное излучение радиочастотного диапазона;
- электромагнитное излучение оптического (видимого) диапазона;
- ультрафиолетовое (УФ) и рентгеновское излучения устаревших ЭЛТ-дисплеев.

Кроме того, на рабочем месте пользователя всегда присутствует электромагнитный фон промышленной частоты, обусловленный как ПЭВМ, так и сторонними источниками.

Источником электростатического поля является экран дисплея, несущий высокий электростатический потенциал (ускоряющее напряжение ЭЛТ). Заметный вклад в общее электростатическое поле вносят электризующиеся от трения поверхности клавиатуры и мыши.

Электростатическое поле, помимо собственно биофизического воздействия на человека, обуславливает накопление в пространстве между пользователем и экраном пыли, которая затем с вдыхаемым воздухом попадает в организм и может вызвать бронхо-легочные заболевания и аллергические реакции. Кроме того, пыль оседает на клавиатуре ПЭВМ и, проникая затем в поры пальцев, может провоцировать заболевания кожи рук.

Следствием систематического воздействия переменных ЭМП с параметрами, превышающими допустимые нормы, являются функциональные нарушения нервной, эндокринной и сердечно-сосудистой систем. Указанные

нарушения проявляются в виде повышенной утомляемости, головных болей, нарушений сна, гипертонии, заторможенности рефлексов. В отдельных случаях отмечаются изменения состава крови, помутнение хрусталика, нервно-психические и трофические заболевания (ломкость ногтей, выпадение волос).

Указанные функциональные изменения, как правило, обратимы, однако при непринятии своевременных профилактических мер могут накапливаться в организме, причем порог необратимости определяется как интенсивностью и длительностью воздействия, так и индивидуальными особенностями организма.

Источником фоновых ЭМП промышленной частоты является, в первую очередь, электропроводка, независимо от того скрытая она или открытая, а также любое электрооборудование (щиты питания, розетки, выключатели) и бытовая электрорадиотехника (осветительные и нагревательные приборы, холодильники, кондиционеры, телевизоры и т. п.). При этом фон конкретного помещения формируется электрооборудованием всего здания и внешними источниками (трансформаторные подстанции, ЛЭП и др.).

5.2.1 Общая характеристика РЧ – воздействий

При обсуждении возможных неблагоприятных эффектов для здоровья при воздействии на человека радиочастотных (РЧ) полей важно не путать РЧ-поля с ионизирующим излучением – таким как, рентгеновские лучи, гамма-лучи или коротковолновое ультрафиолетовое излучение. В отличие от ионизирующего излучения, даже мощные РЧ-поля не могут вызвать ионизацию или радиоактивность в организме. По этой причине РЧ-поля называют неионизирующим излучением. Современные системы мобильной телефонной связи работают на частотах от 800 до 1900 МГц. Системы, использующие частоты 2100 МГц и выше, могут появиться вскоре. Эти частоты попадают в диапазон от 1 МГц до 10 ГГц (1 ГГц = 1 тысячи МГц). Неблагоприятные РЧ-воздействия для здоровья в этом диапазоне излучаются на протяжении 45 лет.

Исследования влияния электромагнитных полей на здоровье человека касались

в основном работников промышленных предприятий, имеющие контакт с генераторами электромагнитного излучения. Было выделено заболевание – радиоволновая болезнь. Она проявляется в функциональных нарушениях нервной системы, неврастеническом и астеническом синдромах. Люди, долгое время находящиеся в электромагнитном поле, жалуются на слабость, раздражительность, быструю утомляемость, ослабление памяти, нарушение сна.

5.2.2 Воздействие на здоровье РЧ-полей от 1 МГц до 10 ГГц

Как известно, РЧ-поля в этом диапазоне проникают сквозь экспонированные ткани и создают нагрев вследствие поглощения энергии. Глубина поглощения РЧ-поля в ткань зависит от частоты поля, и она выше при более низких частотах. Даже очень низкие уровни РЧ-энергии создают небольшое число тепла, но это тепло отводится в процессе нормальной терморегуляции организма. Однако международные технические стандарты, согласно которым производятся мобильные телефоны и их базовые станции, не позволяют им вызвать сколько-либо значительного нагрева (больше 1 градуса Цельсия)

По сообщениям ученых, воздействия РЧ-полей низкой интенсивности – слишком низкое для того чтобы вызывать значительный нагрев, - все же изменяет электрическую активность мозга кошек и кролик за счет подвижности ионов кальция. Также сообщалось о присутствии таково эффекта в изолированных тканях и клетках.

Другие исследования дают основание предполагать, что РЧ-поля изменяют скорость деления клеток, ферментативную активность или воздействуют на гены в клеточной ДНК. Эти эффекты не всегда фиксировались, а также не удалось достаточно хорошо понять их роль во влиянии на здоровье человека, с тем, чтобы создать основу для ограничения воздействия РЧ-полей низкой интенсивности людей.

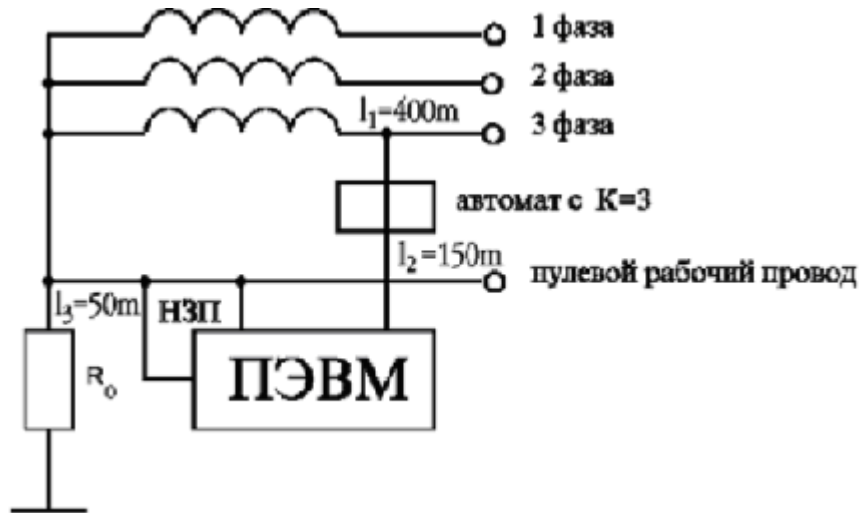
5.3 Методы и средства защиты пользователей от воздействия РЧ-излучения.

Есть три способа защиты от излучения:

- защита расстоянием
- защита экранированием
- минимизация времени воздействия

5.4 Методы и средства защиты от поражения электрическим током

Зануление — преднамеренное соединение нетоковедущих частей с нулевым защитным проводником.



Защитное зануление применяется в трехфазных сетях с глухо заземленной нейтралью, в установках до 1000В и является основным средством обеспечения электробезопасности.

Принцип защиты пользователей при занулении заключается в отключении сети НЗП — нулевой защитный проводник за счет тока короткого замыкания, который вызывает отключение ПЭВМ от сети.

5.5 Выводы

Таким образом, использование рассмотренных норм и способов защиты от опасных и вредных факторов, обусловленных массовым применением вычислительной техники в современном мире, обеспечивают защиту жизни и здоровья пользователей, работающих с вычислительной техникой.

6. Заключение

В процессе выполнения дипломной работы были успешно выполнены следующие задачи:

- произведен обзор рынка существующих систем «Умный дом» и их возможностей
- оценена возможная организация комплекса, способного выполнять одну из функций таких систем, а именно отправку оповещений
- подобраны необходимые аппаратные и программные средства для сбора комплекса
- написана программа для аппаратной платформы Arduino
- написан демон на языке программирования Python
- организована виртуальная частная сеть для соединения веб-сервера и клиента
- установлен и настроен веб-сервер Apache
- с помощью CMS Joomla был организован сайт для демонстрации фотографий и удаления ненужных материалов.
- проведено тестирование прототипа, в ходе которого была подтверждена работоспособность определенных в техническом задании функций

В результате проделанной работы был получен прототип бюджетного аппаратно-программного комплекса для передачи оповещений в системе контроля доступа, способный выполнять одну из функций коммерческих систем.

6.1 Перспективы развития

Полученный аппаратно-программный комплекс выполняет требуемые задачи, однако его структура позволяет расширить функциональность, добавив в код программы необходимые функции, такие как:

- Захват снимка по запросу пользователя.
- Трансляция видео потока с камеры на сайте.
- Указание параметров работы демона на сайте с последующим автоматическим внесением изменений в конфигурационный файл.

6.2 Выводы

В ходе выполнения проекта были сделаны следующие выводы:

- Аппаратные платы Arduino – отличное средство для ознакомления с миром микроконтроллеров и создания прототипов устройств, однако при изготовлении разработанных устройств партиями с финансовой точки зрения выгоднее собирать программировать собственные платы с микроконтроллером.
- С помощью CMS Joomla можно создать сайт, обладающий практически любой функциональностью.
- Стандартная библиотека Python обладает широким набором средств для решения разнообразных задач.
- При выборе протокола, используемого в VPN необходимо учитывать его поддержку используемыми в сети устройствами.

Список литературы

1. Боттс Т., Доусон Т., Перди Г.Н. Linux. Руководство администратора сети. 3-е издание, 2006, с 121-142, 152-153, 272-287
2. Скотт Граннеман. Linux. Карманный справочник, 2008, с 278-280
3. Wikipedia, LanDrive
<http://ru.wikipedia.org/wiki/LanDrive>
4. «Система умный дом SHS»
<http://sh-system.rusmarket.ru>
5. Официальный российский сайт Arduino
<http://arduino.ru/>
6. Руководство по CMS Joomla!
<http://joomla.ru/docs/administrator/dokumentatsiya-po-joomla-2-5>
7. Викиучебник Python
<http://ru.wikibooks.org/wiki/Python>
8. Девисиллов В.А. Охрана труда. М., 2009. с 168-177

Приложение: Исходный код управляющей программы

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
#необходимые пакеты python-opencv и python-numpy для захвата снимка
#в Gentoo необходимо собрать opencv с python и v4l!!!
import cv
import smtplib
import sms24x7
import os
import os.path
import ConfigParser
import serial
import time
from time import gmtime, strftime

config = ConfigParser.SafeConfigParser()

#кнопка
def button_state():
    global c_time
    ##Arduino работает:
    ser = serial.Serial('/dev/ttyACM0', 9600) # если не хочет, то нужно сделать chmod
для этого пользователя
    ser.readline()
    c_time = strftime("%H:%M:%S %d.%m.%Y", gmtime())
#парсинг
def parse():
    config.read("config.cfg")
    global sms_notification
    global email_notification
    global gmail_user
    global gmail_pwd
    global mail_to
```

```

global email
global password
global phone1
global cam_path
global cam_n
global pause
sms_notification = config.get("notifications", "sms_notification")
email_notification = config.get("notifications", "email_notification")
print sms_notification
print email_notification
#email отправка:
gmail_user = config.get("email", 'gmail_user')
gmail_pwd = config.get("email", "gmail_pwd")
mail_to = config.get("email", "mailto")
#sms:
email = config.get("sms_service", "email")
password = config.get("sms_service", "password")
phone1 = config.get("sms_service", "phone1")

#параметры камеры:
cam_path = config.get("camcfg", "camera_path")
cam_n = config.getint("camcfg", "cam_num")
print cam_n
#прочее
pause = config.getfloat("pause", "pause")

```

#захват снимка

```

def capt_f():
    global httpLink
    cname = strftime("%Y-%m-%d-%H-%M-%S", gmtime()) ##### new cname
    cameran = cam_n    #-1' -- любая, 0,1,2... номера камер
    camera_path = cam_path
    capturename = camera_path+cname+".jpg"

    capture = cv.CaptureFromCAM(cameran)

```

```
frame = cv.QueryFrame(capture) #щелк!!
frame = cv.QueryFrame(capture)
cv.SaveImage(capturename, frame)
httplink = "http://192.168.9.112/images/webphotos/"+cname+".jpg" ##### тут
додклать
print httplink
print "Конец ф-ции capt_f"
```

#отправка SMS.

```
def send_sms():
    smsapi = sms24x7.smsapi(email, password)
    smsapi.login()
    smsapi.push_msg("You have a guest! "+c_time+" "+httplink, phone1)
    print "Конец функции send_sms"
```

#отправка электронной почты

```
def mail():
    to = mail_to
    mailServer = smtplib.SMTP("smtp.gmail.com", 587)
    mailServer.ehlo()
    mailServer.starttls()
    mailServer.ehlo()
    mailServer.login(gmail_user, gmail_pwd)
    header = 'To:' + to + '\n' + 'From: ' + gmail_user + '\n' + 'Subject:К вам пришли! \n'
    print header
    msg = header + '\n'+ c_time+'\n' + httplink
    mailServer.sendmail(gmail_user, to, msg)
    print 'done!'
    mailServer.close()
    print "Конец функции отправки почты"
```

```
def start_sending(sms_notification, email_notification):
```

```
    if sms_notification == "1":
        print "Sms notification enabled"
        send_sms()
```

```
elif sms_notification == "true":
    print "Sms notification enabled"
    send_sms()
elif sms_notification == "True":
    print "Sms notification enabled"
    send_sms()
else: print "Sms notification disabled"
if email_notification == "1":
    print "Email notification enabled"
    mail()
elif email_notification == "true":
    print "Email notification enabled"
    mail()
elif email_notification == "True":
    print "Email notification enabled"
    mail()
else: print "Email notification disabled"
```

#бесконечный цикл

```
while True:
    #выполнение функций
    button_state()
    parse()
    capt_f()
    start_sending(sms_notification, email_notification)
# print httplink
time.sleep(pause)
print "В начало цикла"
```


Приложение: конфигурационный файл управляющей программы

[notifications]

sms_notification = 0

email_notification = 0

[email]

gmail_user = tikitukk@gmail.com

gmail_pwd = password

mailto = volkodav2@mail.ru

[sms_service]

email = volkodav2@mail.ru

password = password

phone1 = 79653890223

[camcfg]

camera_path = /home/tikituki/testphoto/

#camnum -- номер камеры, '-1' -- любая, 0,1,2... номера камер

cam_num = -1

[pause]

#Пауза после выполнения цикла.

pause = 1

Приложение: Исходный код для аппаратной платы Arduino

```
const int buttonPin = 4;
int buttonState = 0;
void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  } // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    Serial.print("Button pressed");
    Serial.println();
    delay(2000);
  }
  else {
    // Serial.print("Button released");
    // Serial.println();
  }
}
```