Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Кафедра «Телекоммуникационные системы»
Специальность: 6М071900 «Радиотехника, электроника и телекоммуникации»

ДОПУЩЕН К ЗАЩИТЕ
Зав. кафедрой
к.т.н., профессор Байкенов А.С.
(ученая степень, звание, ФИО)

_____
(подпись)
«_____»_____2015 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ
пояснительная записка

на тему: «Исследование лазерной системы безопасности»

Магистрант: Рыбалов Н.Б _____ группа МРЭн 13-1
(Ф.И.О.) (подпись)

Руководитель: профессор _____ Коньшин С.В
(ученая степень, звани (подпись) (Ф.И.О.)

Рецензент_____ _____ _____
(ученая степень, звание) (подпись) (Ф.И.О.)

Консультант по ВТ к.т.н., ст.препод _____ Ефремова Ю.И
(ученая степень, звание) (подпись) (Ф.И.О.)

Нормоконтроль: ст.препод _____ Демидова Г.Д
(ученая степень, звание) (подпись) (Ф.И.О.)

Алматы, 2015

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Факультет: «Радиотехники и связи»
Специальность: 6М071900 «Радиотехника, электроника и телекоммуникации»
Кафедра: «Телекоммуникационные системы»

ЗАДАНИЕ
на выполнение магистерской диссертации

Магистранту Рыбалову Н.Б.
(фамилия, имя, отчество)

Тема диссертации «Лазерной системы безопасности»
утверждена Ученым советом университета №142 от «31» октября 2013 г.
Срок сдачи законченной диссертации «___»
Цель эксперимента состоит в экспериментальном исследовании возможностей улучшения лазерной системы безопасности и повышения информативности её интерфейса. Добавление возможности логирования информации.

Перечень подлежащих разработке в магистерской диссертации вопросов или краткое содержание магистерской диссертации:
1. Основные направления развития современных систем безопасности
2.Физические параметры приминяемых сенсоров
3. Исследование способов сзаимодействия сенсоров и разработка системы оповещения

Перечень графического материала (с точным указанием обязательных чертежей)
Рисунок 18 – структурная схема системы безопасности .
Рисунок 20 – схематичное представление трасировщика.
Рисунок 26 – графический интерфейс написанный на Java .

Рекомендуемая основная литература

1. Зайцев А.П., Шелупанов А.А. - «Технические средства и методы защиты информации / Машиностроение» – 2009
2. Программирование микроконтроллерных плат Arduino/Freeduino - Улли Соммер

Г Р А Ф И К
подготовки магистерской диссертации

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
1. Информационный обзор согласно теме	05.10.2013	
2. Основные направления развития и проблемы современных систем безопасности	14.01.2013	
3. Исследование факторов влияющих на качество и достоверность детектирования лазерных систем безопасности	02.02.2014	
4. Проектирование лазерной системы ьезопасности	18.03.2015	
5. Анализ полученных экспериментальных данных	01.05.2015	

Дата выдачи задания_____

Заведующий кафедрой_____(Байкенов А.С.)
(подпись) (Ф.И.О.)

Руководитель диссертации_____(Коньшин С.В.)
(подпись) (Ф.И.О.)

Задание принял к исполнению
магистрант _____(Рыбалов Н.Б.)
(подпись) (Ф.И.О.)

Аңдатпа

Диссертациялық жұмыс қазіргі заманғы қауіпсіздік жүйелерінің мәселелеріне арналған. Бұл сала бүгінгі қоғамдағы криминогендік жағдайдың нашарлауына байланысты өзекті мәселе болып табылады. Осы жұмыста ұсынылған қауіпсіздік жүйесінің модульді болуы, оның даусыз артықшылығы болып табылады. Күрделі жүйелердің модульді болуы техникалық қолдаудың мүлдем жаңа саласының белсенді дамуына мүмкіндік берді, сондай-ақ еркін бағдарламалық қамтамасыздандырудың болуы бір мезгілде соңғы жаңартуларды шығаруға мүмкіндік береді.

Бірыңғай перифериялық датчиктер жиынтығымен жабдықталған лазерлік жүйесінің потенциялын белсенді пайдаланылуы, рұқсатсыз кірудің барлық түрлеріне қарсы тиімділігін айтарлықтай арттыруға мүмкіндік береді. Жанама анықтау әдісі ретінде және тұтас игерілген жуйенің тиімділігін арттыру мақсатында температура функциясының туындысы қолданылған.

Аннотация

Диссертационная работа посвящена рассмотрению проблемы современных систем безопасности. Неоспоримым преимуществом представленной в данной работе системы безопасности является её модульность. Именно модульность таких сложных систем позволила положить начало активному развитию совершенно новой сфере технической поддержки, а так же наличие программного обеспечения, носящего характер свободное ПО позволяет единовременно выпускать актуальные обновления.Активное использование потенциала лазерной системы объединённой с набором периферийных датчиков позволяет значительно повысить эффективность схемы ко всем видам несанкционированных проникновений. В качестве косвенного метода обнаружения и повышения продуктивности разработанной системы в целом была применена производная функция температуры.

Abstract

The thesis is devoted to the problems of modern security systems. This area remains a topical effect worsening crime situation in today's society. The undeniable advantage presented in this paper, the security system is its modularity. This modularity of complex systems has allowed to initiate active development of an entirely new field of technical support, as well as the availability of software in the nature of free software allows you to simultaneously release the latest updates.

The active use of the potential of the laser system with a set of unified peripheral sensors can significantly increase the efficiency of the scheme to all types of infiltrations.

Содержание

Введение

Общая характеристика работы. Диссертационная работа посвящена рассмотрению проблемы современных систем безопасности. Данное направление остаётся актуальным в силу обострения криминогенной обстановки в современном обществе. Неоспоримым преимуществом представленной в данной работе системы безопасности является её модульность. Именно модульность таких сложных систем позволила положить начало активному развитию совершенно новой сфере технической поддержки, а так же наличие программного обеспечения, носящего характер свободное ПО позволяет единовременно выпускать актуальные обновления.

В магистерской диссертации представлена концепция реализации принципов построения современных систем безопасности, продемонстрированы способы взаимодействия их модулей между собой, а так же описаны программные и аппаратные рекомендации, которые следует учитывать при разработке подобных систем.

Постановка проблемы и её актуальность. На сегодняшний день в области организации активной безопасности стремительно развиваются и внедряются модульные концепции. Данное направление было актуальным и раннее, однако настоящее развитие получило лишь в недавнее время. Связано это с выпуском большого числа сенсоров с открытыми интерфейсами. Модульность конструкции позволяет масштабировать и расширять функционал конечного устройства без полной замены аппаратной части, что может полностью покрыть спектр запросов потенциальных заказчиков. Согласно аналитическим данным, в будущем наиболее динамично будут развиваться те сервисы, которые полностью отвечают потребностям современных компаний. Они должны быстро и просто внедряться, не требовать больших инвестиций, иметь постоянную доступность.

Необходимо определить критерии системы, которые следует учесть при разработке модели. При этом система должна обеспечивать сочетание нескольких важных характеристик:

-управляемость;
-поддержка и сопровождение;
-масштабируемость;
-надёжность;
-высокая готовность;
-работоспособность.

Целью работы является решение части существующих проблем:

1 Организация протокола взаимодействия между аппаратной и программной частью;

3 Рассмотрение вариантов построения модульной конструкции;

4 Улучшение эффективности действия лазерных сенсоров;

5 Тестирование и анализ результатов исследования системы.

Объектом исследования является технология построения и взаимодействия элементов системы безопасности и улучшения функционирования лазерных рубежей защиты.

Методы исследования. Исследование проводилось на самостоятельно разработанной модели. За время работы над диссертацией было изучено множество материалов, представленных на специализированных интернет-порталах. Все базовые понятия и теория были выделены из литературы, указанной авторами статей, в том числе представленные в трудах зарубежных научных исследователей.

Необходимые практические навыки исследования были получены во время прохождения обязательной научной стажировки.

Над аппаратной частью системы проводились эксперименты путём симуляции всех возможных сценариев организации нападения, программная часть на первых этапах разработки эмулировалась посредством виртуализации в среде Proteus. В дальнейшем поведение системы исследовалось да полноценном физическом макете.

Научную новизну представляет разработка протокола взаимодействия между аппаратной частью и графическим интерфейсом, логированием данных в MySQL и исследованием возможности обнаружения проникновения в охраняемую зону базируясь на производной функции температуры (для особых случаев).

Практическая значимость результатов исследований представляет собой возможность дальнейшего активного развития данного направления и открывает перспективы выпуска на рынок конечного продукта в виде охранного комплекса. По результатам опросов, на сегодняшний день в Казахстане нет компаний, занимающихся разработкой подобных систем безопасности, в связи с чем все основные исследования проходят только в рамках университетов.

Личный вклад автора присутствует во всех звеньях работы, а именно: в проведении литературного обзора по теме диссертации, постановке проблемы, постановке и проведении экспериментов, разработку виртуальной модели системы на базе компьютера с применением систем автоматизированного проектирования, анализе результатов решения задач, создания рабочего прототипа.

Апробация

Рыбалов Н.Б Исследование лазерной системы безопасности//Сборник научных трудов магистрантов специальностей “Радиотехника, электроника и телекоммуникации”.

Структура и объем диссертации. Диссертационная работа содержит список обозначений и сокращений, введение, основная часть из пяти разделов, заключения, приложения и списка использованных источников. Объем диссертации составляет 73 страницы машинописи, включая 27рисунков, 4 формулы.

1 Основные понятия систем безопасности

1.1 Основы построения систем безопасности

Для построения сбалансированной системы информационной безопасности требуется первоначально провести анализ рисков в области информационной безопасности. Затем определить оптимальный уровень риска для организации на основе заданного критерия. Систему информационной безопасности (контрмеры) предстоит построить таким образом, чтобы достичь заданного уровня риска.

Анализ состояния дел в сфере защиты информации показывает, что уже сложилась вполне сформировавшаяся концепция и структура защиты, основу которой составляют:

- хорошо развитый ассортимент технических средств защиты информации, производимых на промышленной основе;
- значительное число, имеющих необходимые лицензии организаций, специализирующихся на решении вопросов защиты информации;
- четкая система взглядов на проблему защиты информации и наличие некоторого практического опыта.

Тем не менее, несмотря на это число злоумышленных действий над информацией не только не уменьшается, но и имеет достаточно устойчивую тенденцию к росту. Опыт показывает, что для борьбы с этой тенденцией необходима крепкая и целенаправленная организация процесса защиты информационных ресурсов. Причем в этом должны активно участвовать профессиональные специалисты, администрация, сотрудники и пользователи, что и определяет повышенную значимость организационной стороны вопроса. Опыт различных ведомств и организаций показывает, что:

- обеспечение безопасности информации не может быть одноразовым актом. Это непрерывный процесс, заключающийся в обосновании и реализации наиболее рациональных методов, способов и путей совершенствования и развития системы защиты, непрерывном контроле ее состояния, выявлении ее узких и слабых мест и противоправных действий;
- безопасность информации может быть обеспечена лишь при комплексном использовании всего арсенала имеющихся средств защиты во всех структурных элементах производственной системы и на всех этапах технологического цикла обработки информации. Наибольший эффект достигается тогда, когда все используемые средства, методы и меры объединяются в единый целостный механизм – Систему Защиты Информации (СЗИ). При этом функционирование системы должно контролироваться, обновляться и дополняться в зависимости от изменения внешних и внутренних условий;
- никакая система защиты информации не может обеспечить требуемого уровня информационной безопасности без надлежащей подготовки пользователей и соблюдения ими всех установленных правил, направленных на ее защиту.

Таким образом, можно определить систему защиты информации как организованную совокупность специальных органов, средств, методов и мероприятий, обеспечивающих защиту информации от внутренних и внешних угроз.

Анализируя вышеприведенные требования, необходимо выделить условия, которым должна удовлетворять система защиты информации:

- охватывать весь технологический комплекс информационной деятельности;
- быть разнообразной по используемым средствам, многоуровневой, с иерархической последовательностью доступа;
- быть открытой для изменения и дополнения мер обеспечения безопасности информации;
- быть нестандартной, разнообразной. При выборе средств защиты нельзя рассчитывать на неосведомленность злоумышленников относительно ее возможностей;
- быть простой для технического обслуживания и удобной для эксплуатации пользователями;
- быть надежной;
- быть комплексной, обладать целостностью, означающей, что ни одна ее часть не может быть изъята без ущерба для всей системы.

Кроме того к системе защиты информации должны быть предъявлены следующие требования:

- четкость определения полномочий и прав пользователей на доступ к определенным видам информации;
-предоставление пользователю минимальных полномочий, необходимых ему для выполнения порученной работы;
- сведение к минимуму числа общих для нескольких пользователей средств защиты;
- учет случаев и попыток несанкционированного доступа к конфиденциальной информации;
- обеспечение оценки степени конфиденциальности информации;
- обеспечение контроля целостности средств защиты и немедленное реагирование на их выход из строя;
- система защиты информации, как и любая другая система должна иметь определенные виды собственного обеспечения, опираясь на которые она будет выполнять свою целевую функцию.

С учетом вышеприведенных положений система защиты информации должна иметь:

- *правовое обеспечение.* Сюда входят нормативные документы, положения, инструкции, руководства, требования которых являются обязательными в рамках сферы их действий;
- *организационное обеспечение.* Реализация защиты информации осуществляется определенными структурными единицами – такими, как служба защиты документов, служба режима допуска охраны, служба защиты

информации техническими средствами, служба информационно-аналитической деятельности;

- *аппаратное обеспечение.* Предполагается широкое использование технических средств как для защиты информации, так и для обеспечения деятельности собственно системы защиты информации;

- *информационное обеспечение.* Включает в себя сведения, данные, показатели, параметры, лежащие в основе решения задач, обеспечивающих функционирование системы;

- *программное обеспечение.* К нему относятся различные информационные, учетные, статистические, и расчетные программы, обеспечивающих оценку наличия и опасности различных каналов утечки и путей несанкционированного проникновения к источникам конфиденциальной информации;

- *математическое обеспечение.* Предполагает использование математических методов для различных расчетов, связанных с оценкой опасности технических средств злоумышленников, зон и норм необходимой защиты;

- *лингвистическое обеспечение.* Сюда входят нормы и регламенты деятельности органов, служб, средств, реализующих функции защиты информации, различного рода методики, обеспечивающие деятельность пользователей при выполнении своей работы в условиях жестких требований защиты информации.

Таким образом, под системой безопасности понимают организованную совокупность специальных органов, служб, средств, методов и мероприятий, обеспечивающих защиту жизненно важных интересов личности, предприятия и государства от внутренних и внешних угроз. Структурная схема системы безопасности приведена на странице 35 рисунке 18.

1.1.1 Технические средства

Для создания охраняемого периметра используются различные системы безопасности. В основе их действия используются различные физические процессы, такие как лазерное излучение , электромагнитные поля, изменения давления и температуры окружающей среды.

Охранная сигнализация, как следует из самого названия, предназначена для немедленного реагирования установленной системы на нарушение границы подконтрольного объекта. В продвинутых системах сигнал тревоги в первую очередь подается хозяину или непосредственно на пульт охранной фирмы. Если конфигурация сигнализации подобрана с учетом особенностей дома и видом потенциальной угрозы, то со своей задачей она справиться эффективно.

Залог эффективности любой охранной сигнализации – возможность раннего обнаружения нарушителя. Чем раньше он будет замечен, тем больше остается времени для ответных мер. Однако раннее обнаружение всегда связано с увеличением затрат на оборудование и установку. Рынок

предлагает несколько вариантов выявления угрозы несанкционированного проникновения злоумышленников на территорию частной собственности:

- обнаружение в момент преодоления границ участка (забор или условный рубеж);
- обнаружение злоумышленника на пути к объекту;
- обнаружение в момент проникновение в здание через двери или окна;
- обнаружение в доме.

Каждый из этих вариантов реализуется при помощи определенного вида оборудования, в частности, специальными датчиками. Центральный пульт сигнализации при этом может быть одним и тем же. Различные охранные задачи обеспечиваются системами с разной степенью сложности. Наиболее простыми являются системы, обнаруживающие нарушителя уже непосредственно в здании.

Для каждого вида нарушителя соответственно свой вид средств охраны. Профессиональный подход к разработке системы сигнализации требует составления «портрета» потенциального нарушителя. При этом учитываются такие факторы, как уровень подготовки, набор навыков и возможные методы воздействия. Эти люди грубо взламывают запертые помещения, редко пользуются масками и перчатками, оставляют следы и имеют слабое представление о способностях современной электронной сигнализации. Профессионалы встречаются гораздо реже, но и о них не нужно забывать, особенно, если на объекте хранится имущество, которое потенциально может их заинтересовать.

Среднестатистический дом нуждается в защите от обычных непрофессиональных злоумышленников, в арсенале которых простые методы проникновения на охраняемую территорию. Степень сложности охранной системы всегда должна быть адекватной ценности охраняемого имущества. Если речь идет о дорогостоящей аппаратуре, предметах искусства и других материальных ценностях, то риск стать жертвой профессионала резко увеличивается. Поэтому в таком случае целесообразно воспользоваться помощью специалистов, которые установят соответствующее оборудование.

Если имущество можно определить как не выделяющееся на общем фоне, то, скорее всего, можно воспользоваться типовой сигнализацией, возможностей которой хватит, чтобы оповестить о проникновении на объект «низкопрофильных» воров. К примеру, в домах на дачных участках, где все ценное имущество представлено водяным насосом, холодильником и телевизором, устанавливать лазерную сигнализацию не рентабельно.

На выбор сложности системы сигнализации также влияет и окружение охраняемого объекта. Благоприятные факторы – наличие соседей, близость к охраняемым зданиям, освещенное и оживленное место. Если система защиты устанавливается от преступников не профессионалов, в данном случае будет действенной только звуковая сирена. Однако указанный способ будет эффективным только в том случае, если дом находится в населенном пункте, пусть даже с малой численностью населения. Дом на окраине

довольно трудно защитить от разорения только лишь средствами сигнализации.

Различают несколько типов звуковых сигнализаций:

- автономная сигнализация, представляющая собой оборудование, устанавливаемое внутри дома, элементы которой функционируют независимо друг от друга;
- классическая проводная сигнализация, расположенная внутри дома;
- беспроводная сигнализация, датчики которой связываются с центральным пультом радиосигналами;
- наружная проводная сигнализация с датчиками, обеспечивающими охрану периметра объекта.

Все эти системы работают по принципу сигнал – тревога. При невысокой вероятности проникновения такие сигнализации могут сэкономить средства, которые пришлось бы потратить на механическую защиту – противовзломные окна и двери. Если же вероятность проникновения высока, то понадобятся системы раннего обнаружения, элементы которых располагаются на внешнем ограждении и фасаде. Самостоятельно смонтировать такую систему без специальных навыков, скорее всего, не получится. Используемые электронные системы таких сигнализаций довольно сложны, а, кроме того, потребуется скрытый монтаж внешних датчиков.

Чтобы установить простую автономную сигнализацию, необходим пассивный ИК-датчик – он же датчик движения. Такие датчики бывают со встроенной сиреной и с возможностью дистанционного управления. Потребуется также мощная сирена, которая будет включаться по сигналу датчиков, и несколько контактных магнитных датчиков (контактронов) с маломощной сиреной и автономным питанием [4].

ИК-датчики могут питаться как от батарей, так и от сетевого блока питания. Такой блок необходим еще и при длительном отсутствии хозяев – больше недельного срока. Если батарея сядет, то датчик останется в рабочем состоянии.

Несложная автономная система обеспечит сигнализационную охрану мест, наиболее подверженных угрозе проникновения, причем, даже когда они находятся дома (например, в ночное время, когда все спят).

Автономные пассивные инфракрасные датчики питаются от батареи. Некоторые модели имеют собственную сирену, а также дополнительный выход подключения вспомогательной сирены. ИК-датчики устанавливаются в помещении для контроля их внутреннего пространства. При активированном датчике незаметное проникновение в охраняемое помещение невозможно. Но устанавливать их нужно так, чтобы траектория предполагаемого злоумышленника была перпендикулярна обзора датчика.

Кроме того, ИК-датчики выполняют функцию постановки сигнализации на охрану. Для этого, выходя из дома достаточно воспользоваться пультом дистанционного управления. При возвращении в дом сигнализацию выключают и тогда сирена не срабатывает. Постановка и

снятие сигнализации с охраны подтверждается коротким звуковым сигналом. Очень важно учитывать, что автономные датчики сработают даже при передвижении в помещении животных (кошка, собака). Если у вас в доме имеются таковые, то лучше купить датчики, реагирующие на движущиеся объекты определенного веса (объема), например от 20 кг и выше [7].

Автономные охранные сирены обычно укомплектованы длинным проводом, позволяющем установить их, примерно на удалении 10 м от датчика. Расположить сирену можно как снаружи дома, так и внутри. Корпус бюджетных сирен не имеет значительной механической прочности, поэтому устанавливать ее нужно в труднодоступном месте. В то же время желательно чтобы место установки должно было скрытым, иначе злоумышленник может сломать ее еще до того, как она сработает. Таким местом снаружи дома может стать ниша в стене на высоте более 3 м, а внутри дома – ниша в стене за гардиной. Эту нишу можно защитить вмурованной металлической решеткой. Однако сирены чаще просто устанавливают на высоте более 3 м, и этого бывает достаточно. Датчики делятся на типы. Пассивный инфракрасный извещатель является одним из наиболее распространенных детекторов устанавливаемых в домашних охранных сигнализациях и малом бизнесе, поскольку он предлагает доступные и надежные функциональные возможности. Этот термин означает, что пассивный детектор способен функционировать без необходимости генерировать и излучать свою энергию (в отличии от ультразвуковых и микроволновых объемных детекторов вторжения, которые являются "активными" в операции). Пассивный инфракрасный датчик способен отличить, если инфракрасный излучающий объекта присутствует, сначала обучение температуре контролируемого пространства, а затем выявление изменений температуры, вызванное присутствием объекта. Используя принцип дифференциации, который является проверкой наличия.

Пассивный инфракрасный датчик проверяет, есть ли злоумышленник или объект на самом деле. Создание отдельных зон обнаружения, где каждая зона включает в себя один или несколько слоев можно добиться дифференциации.

Между зонами существуют области не чувствительности (мертвых зон), которые используются датчиками для сравнения [5].

Установка датчика движения обеспечивает надежную защиту дома. Ультразвуковые датчики используют частоты от 15 кГц и 75 кГц, эти активные датчики передают ультразвуковые волны, которые не слышны для человека. Принцип доплеровский сдвиг является основным методом работы, в которых изменение частоты обнаружения за счет движения объекта.

Эффект Доплера — изменение частоты и длины волн, регистрируемых приёмником, вызванное движением их источника или движением приёмника. Это происходит, когда движущийся объект изменяет частоты звуковых волн вокруг себя. Два условия должны произойти, чтобы успешно обнаруживать событие доплеровского сдвига. Там должно быть движение объекта или к или от приемника. Движение объекта должно привести к изменению

ультразвуковой частоты приемника относительно частоты передачи. Ультразвуковой датчик работает передатчиком излучения ультразвукового сигнала в области которые должны быть защищены. Звуковые волны отражаются от твердых предметов (таких, как окружающий пол, стены и потолок), а затем обнаружить приемник. Потому что ультразвуковые волны, которые передаются через воздух, а затем встречаются с твердым покрытием, как правило, отражают большую часть ультразвуковой энергии, в то время как мягкие поверхности, как правило, поглощают большую часть энергии. Когда поверхность неподвижна, частота волн приемника будет равна передаваемым частотам. Тем не менее, изменение частоты будет происходить в результате принципа Доплера, когда человек или объект перемещается ближе или дальше от детектора. Такое событие инициирует сигнал тревоги. Эта технология считается устаревшей, многие специалисты по охранной сигнализации уже активно не устанавливает эти виды датчиков.

Микроволновые датчики. Это устройство испускает микроволны из передатчика и обнаруживает отражение микроволн или уменьшение интенсивности пучка с помощью приемника. Передатчик и приемник, как правило, сочетается в одном корпусе (моностатической) для использования внутри помещений, так и отдельных помещений (двухпозиционный) для наружного применения. Для уменьшения ложных срабатываний такого вида датчиков, как правило, они используются вместе с пассивным инфракрасным детектором. Микроволновые датчики реагируют на доплеровский сдвиг частоты отраженной энергии, фазовый сдвиг, или внезапным снижением уровня получаемой энергии. Любой из этих эффектов может указывать на движение нарушителя.

Фото-электрические пучки. Фотоэлектрические системы луч обнаруживают присутствие злоумышленника при пересечении видимого или ИК луча света через охраняемую область. Для улучшения зоны обнаружения, часто используются сразу два или более фото-электрических пучка. Однако, если злоумышленник знает о наличии этой технологии, он может ее избежать.

Эта технология эффективна для защиты большой территории, если они установлены в штабель из трех или более, где передатчики и приемники расположены в шахматном порядке, чтобы создать забор, как барьер. Системы доступны как для внутреннего и внешнего применения.

Датчики разбития стекла. Извещатель разбития стекла может быть использован для защиты внутреннего периметра здания. Если стекло разбивается, оно генерирует звук в широком диапазоне частот. Они могут варьироваться от инфразвуковых, что ниже 20 Гц и не может быть услышана человеческим ухом, через аудио диапазоне от 20 Гц до 20 кГц, который люди могут услышать, вплоть до ультразвуковой, что выше 20 кГц и опять же не может быть услышан. Детекторы разбития стекла устанавливаются в непосредственной близости от стекла и прослушивают звуковые частот связанные с разбития стекла.

Сейсмические датчики разбития стекла отличаются тем, что они устанавливаются на стекло. Если стекло разбивается, оно производит определенные частоты шока, которые проходят через стекло и часто через оконную раму и окружающие стены и потолок. Как правило, наиболее интенсивные частоты порожденных от 3 до 5 кГц, в зависимости от типа стекла и наличие пластиковой прослойки. Сейсмические датчики разбития стекла "почувствуют" эти ударные частоты и, в свою очередь сработает охранная сигнализация.

Более примитивные методы обнаружения разбития стекла состоит в приклеивание тонкой полоски фольги на внутреннею сторону стекла и и пропустить маломощный электрический ток через неё . При разбитии стекла практически гарантированно обрывается фольга и разрывается цепь. А там что подключите сами: сирена или что-то другое.

Датчики дыма, тепла и детекторы угарного газа. Система пожарной сигнализации с тепловыми извещателями. Большинство систем также может быть оснащены датчиками дыма, тепла или оксид углерода детекторами. Они также известны как 24-часовые зоны. Детекторы дыма и тепловые детекторы защитят от риска возникновения пожара и детекторы угарного газа защитят от риска задохнуться угарным газом.

Уличные охранные датчики. Эти типы датчиков будут установлены в большинстве случаев на заборах или устанавливаться по периметру охраняемой территории.

Датчики измерения вибрации. Эти устройства устанавливаются на барьеры и используются в основном для обнаружения нападения на территорию объекта. Технология основывается на неустойчивой механической конфигурации, которая является частью электрической цепи. При движении или вибрации происходит, неустойчивость части схемы движется и нарушения движения тока, который вырабатывает сигнал тревоги. Технология устройства варьируется и может быть чувствительным к различным уровням вибрации. Среды, передающее колебания должны быть правильно выбраны для конкретного датчика, поскольку они лучше всего подходят для различных видов конструкций и конфигураций.

Пьезоэлектрические датчики это новые технологий с недоказанной еще эффективностью в отличие от механических датчиков, которые в некоторых случаях имеют эффективный срок службы более 20 лет.

Обнаружение магнитного поля. Это система безопасности основана на принципе определения магнитных аномалий при пересечении определенной территории. Система использует генератор электромагнитного поля, питается от двух проводов, параллельно. Оба провода проходят по периметру и, как правило, прокладываются около 5 см друг от друга в верхней части стены или около 12 "/ 30 см под землей. Провода подключены к сигнальным процессорам, который анализирует любые изменения в магнитном поле.

Такого рода система безопасности, как обнаружение магнитного поля, может быть встроена в верхней части практически в любой стене и обеспечить регулярную способность обнаружения движения, или может

быть закопан в земле. Она обеспечивает достаточно низкий уровень ложных тревог, и у подобных систем очень высокая вероятность обнаружения реальных грабителей. Тем не менее, они не могут быть установлены возле высоковольтных линий, или радар передатчиков.

Активные оптико-электронные извещатели. Эта близкая система может быть установлена на здании периметра, заборы и стены. Она также имеет возможность установки на специальных столбах. Система использует генератора электромагнитного поля один провод питания, с другой чувствительный провод параллельно с ним. Оба провода проходят по периметру и, как правило, устанавливаются около 800 мм друг от друга. Чувствительный провод подключен к сигнальным процессорам, который анализирует:

- амплитуда изменения (масса злоумышленника);
- скорость изменения (движение злоумышленника);
- нарушение установленного времени (время злоумышленника в зоне безопасности).

Эти элементы определяют характеристики нарушителя и, когда все три тревоги обнаружены одновременно, сигнал тревоги генерируется. Барьер может обеспечить защиту от земли до около 4 метров в высоту.

Плюсы: скрытый, есть возможность закопать в землю.

Минусы: дорогой, короткие зоны, чем больше электроники (соответственно, больше затрат), повышенный уровень ложных срабатываний, поскольку он не может отличить кошку от человека.

Микроволновые барьеры. Работает микроволновой барьер очень просто. Этот тип устройства, создает электромагнитные пучки с помощью высоких частот, которые проходят от передатчика к приемнику, создавая невидимую, но чувствительный стену защиты. Когда приемник обнаруживает разницу в условиях пучка (и, следовательно, возможного вторжения), система начинает детальный анализ ситуации. Если система считает, что сигнал реального вторжения, она обеспечивает сигнал тревоги, которые могут рассматриваться в аналоговом или цифровом виде.

Плюсы: низкая стоимость, простота в установке, невидимый барьер по периметру, большая площадь охраны периметра от нарушителя.

Минусы: очень чувствительны к погоде, как дождь, снег и туман вызывают ложные срабатывания.

Микрофонные системы. Микрофонные системы отличаются по дизайну, но каждый, как правило, работает на обнаружении нарушителя пытающего перелезть через забор. Обычно микрофонные системы обнаружения устанавливаются и подключаются к жесткому забору, однако некоторые специализированные версии этих систем могут быть установлены и под землю. В зависимости от выбранной версии, они могут быть чувствительными к различным уровням шума и вибрации. Система основана на коаксиальном или электромагнитном датчике с контроллером.

Системы предназначены для обнаружения и анализа входящих электронных сигналов, полученных от датчиков микрофонной системы, а

затем генерировать сигналы тревоги от сигналов, которые превышают заданные условия. Системы имеют регулируемые электроникой разрешения инсталляторов для изменения чувствительности датчиков сигнализации с учетом конкретных условий окружающей среды. Настройка системы, как правило, осуществляется при вводе в эксплуатацию устройств обнаружения.

Плюсы: очень дешево, очень простая конфигурация, легко установить.

Минусы: некоторые системы имеют высокий уровень ложных срабатываний, так как некоторые из этих датчиков могут быть слишком чувствительными. Хотя системы с использованием DSP (Digital Signal Processing) будет в значительной степени исключать ложные тревоги в некоторых случаях.

Заборы охранные. Тугой провод натягивается по периметру системы безопасности, обычно устанавливается на заборе или стене. Кроме того, можно натянуть проволоку так , что нет необходимости в самом заборе. Эти системы предназначены для обнаружения любых физических попыток проникнуть на охраняемую территорию. Забор охранный может работать со множеством переключателей или детекторов, которые чувствуют движение на каждом конце натянутого провода. Эти коммутаторы или детекторы могут быть простым механическим контактом, статическим преобразователем силы или электронным тензометрическим. Нежелательные тревоги, вызванные животными и птицами, можно избежать путем изменения настройки датчика, и игнорировать объекты, которые оказывают небольшое количество давление на провода. Такая система является уязвимой для злоумышленников, можно подкопать под забором.

Плюсы: низкий уровень ложных тревог, очень надежные датчики и высокий уровень обнаружения.

Минусы: очень дорогие, сложные в установке и старые технологии.

Волоконно-оптические датчики. Волоконно-оптические датчики могут быть использованы для обнаружения вторжений путем измерения разницы в количестве света в волокне сердечника. Если кабель нарушается, свет будет "протекать", и приемник будет определять разницу в размере полученного света. Кабель может быть подключен непосредственно к забору или связаны в колючие ленты из стали, которые используется для защиты вершины стены и забора. Этот тип колючей ленты обеспечивает хорошее физическое сдерживания, а также дает немедленный сигнал тревоги, если ленту разрезать или сильно деформировать. Другие типы работы по обнаружению изменения поляризации, которая обусловлена изменением позиции волокна.

Плюсы: очень похоже на микрофонные системы, очень простая конфигурация, простота в установке.

Минусы: высокий уровень ложных срабатываний сигнализации.

Напряженность магнитного поля. Эта система использует электромагнитные поля нарушения, принцип основан на двух неэкранированных (или 'вытекающих') коаксиальных кабелях закопанных около 10-15 см в глубину и находится на расстоянии
около 1 метра друг от друга. Передатчик излучает непрерывную

радиочастотную (РЧ) энергию по одному кабелю и энергию получает другой кабель. При изменении поля в связи с наличием объекта когда достигает заданного нижнего порога включается охранная сигнализация. Систему необходимо установить правильно, и необходимо соблюдать и обеспечить окружающию почву хорошим дренажом для того, чтобы уменьшить риск ложных тревог.

Плюсы: не заметен в связи с расположением под землей.

Минусы: повышенный уровень ложных срабатываний, имеются трудности с установкой.

Виды датчиков для зашиты окон и дверей. Датчик штора - предназначен для защиты оконных и дверных проемов, датчик штора устанавливается над окном и образует занавесу при пересечении которой сразу срабатывает охранная сигнализация. Эти виды датчиков служат хорошей заменой устаревающих датчиков геркон.

Датчик геркон – также предназначен для защиты окон и дверей, этот датчик состоит из двух частей ,одна часть датчика крепится на не подвижной части рамы окна, а другая на подвижной. Когда окно открывается, разрывается цепь между двумя частями геркона и срабатывает охранная сигнализация.

1.1.2 Организационные меры

Организационная защита должна обеспечивать:

- охрану, режим, работу с кадрами, с документами;
- использование технических средств безопасности и информационно-аналитическую деятельность по выявлению внутренних и внешних угроз производственной деятельности.

Организационные мероприятия играют существенную роль в создании надежного механизма защиты информации, т.к. возможности несанкционированного использования конфиденциальных сведений в значительной мере обуславливаются не техническими аспектами, а злоумышленными действиями, небрежностью и халатностью пользователей или персонала защиты. Влияния этих аспектов практически невозможно избежать с помощью технических средств. Для этого необходима совокупность организационно-правовых и организационно-технических мероприятий, которые исключали бы несанкционированный доступ к конфиденциальной информации.

К основным организационным мероприятиям относят:

- организацию режима и охраны. Их цель – исключение возможности проникновения (как тайного, так и явного) на территорию и в помещения посторонних лиц; обеспечение удобства контроля прохода и перемещения сотрудников и посетителей; создание отдельных производственных зон по типу конфиденциальных работ с самостоятельными системами доступа; контроль и соблюдение временного режима и пребывания на территории персонала фирмы; организация и поддержание надежного пропускного режима и контроля сотрудников и посетителей:

- организацию работы с сотрудниками, которая предусматривает подбор и расстановку персонала, включая ознакомление с сотрудниками, их изучение, обучение правилам работы с конфиденциальной информацией, ознакомление с мерами ответственности за нарушение правил защиты информации;

- организацию работы с документами, включая организацию разработки и использования документов и носителей информации, их учет, исполнение, возврат, хранение и уничтожение;

- организацию использования технических средств сбора, обработки, накопления и хранения информации;

- организацию работ по анализу внутренних и внешних угроз конфиденциальной информации и выработке мер по обеспечению ее защиты;

- организацию работы по проведению систематического контроля за работой персонала с конфиденциальной информацией, порядком учета, хранения и уничтожения документов и технических носителей.

В каждом конкретном случае организационные мероприятия носят специфические для данной организации форму и содержание, направленные на обеспечение безопасности информации в конкретных условиях.

Очевидно, что организационные мероприятия должны четко планироваться, направляться и осуществляться каким-то специально созданным для этих целей структурным подразделением, укомплектованным соответствующими специалистами по безопасности, производственной деятельности и защите информации. Зачастую, таким структурным подразделением является служба безопасности предприятия, на которую возлагаются следующие основные функции:

- организация и обеспечение охраны персонала, материальных и финансовых ценностей и защиты конфиденциальной информации;

- обеспечение пропускного и внутриобъектового режима на территории, в зданиях и помещениях, контроль соблюдения требований режима сотрудниками, смежниками, партнерами и посетителями;

- руководство работами по правовому и организационному регулированию отношений по защите информации;

- участие в разработке основополагающих документов с целью закрепления в них требований обеспечения безопасности и защиты информации, а так же положений о подразделениях, трудовых договоров, соглашений, подрядов, должностных инструкций и обязанностей руководства, специалистов, рабочих и служащих;

- разработка и осуществление совместно с другими подразделениями мероприятий по обеспечению работы с документами, содержащими конфиденциальные сведения;

- изучение всех сторон производственной, коммерческой, финансовой и другой деятельности для выявления и последующего противодействия любым попыткам нанесения ущерба, ведения учета и анализа нарушений безопасности, накопление и анализ данных о злоумышленных устремлениях

конкурентных организаций, о деятельности предприятия и его клиентов, партнеров, смежников;

- организация и проведение служебных расследований по фактам разглашения сведений, утрат документов, утечки конфиденциальной информации и других нарушений безопасности предприятия;

- осуществление руководства службами и подразделениями безопасности подведомственных предприятий, организаций, учреждений и другими структурами в части оговоренных в договорах условий по защите конфиденциальной информации;

- организация и регулярное проведение учета сотрудников предприятия и службы безопасности по всем направлениям защиты информации и обеспечения безопасности производственной деятельности;

- ведение учета и строгого контроля выделенных для конфиденциальной работы помещений, технических средств в них, обладающих потенциальными каналами утечки информации и каналами проникновения к источникам охраняемых секретов;

- обеспечение проведения всех необходимых мероприятий по пресечению попыток нанесения морального и материального ущерба со стороны внутренних и внешних угроз.

Для защиты речевой информации во время проведения совещаний целесообразно и обоснованно применение следующих организационных мер:

- перед проведением совещания необходимо проводить визуальный осмотр помещения на предмет выявления закладных устройств. Осмотр должен проводиться систематизировано и тщательно, обращая внимание на любую мелочь, на первый взгляд незначительную: наличие посторонних предметов, бытовой аппаратуры или предметов интерьера. Осмотр по возможности, должен проводиться сотрудником службы безопасности и человеком, хорошо знакомым с обычной обстановкой зала, например, сотрудником охраны;

- непосредственно перед проведением совещания охрана должна осматривать прилегающие к комнате для совещаний помещения, на предмет удаления из них сотрудников или посторонних лиц, которые могут вести подслушивание непосредственно через систему вентиляционных коммуникаций или через стену при помощи специальной аппаратуры; закрывать эти помещения на время проведения совещания и вести контроль за доступом в них;

- количество лиц, участвующих в конфиденциальных переговорах должно быть ограничено до минимума;

- вход посторонних лиц во время проведения совещания должен быть запрещен;

- должна быть четко разработана охрана выделенного помещения во время совещания, а также наблюдение за обстановкой на этаже. Во время проведения закрытого совещания необходимо не допускать близко к дверям комнаты никого из посторонних лиц или сотрудников организации. Стоя под дверью, или поблизости от нее, злоумышленник может подслушать то, о чем

говориться в комнате. Особенно это вероятно в те моменты, когда кто-то входит или выходит из зала, так как на то время, пока дверь остается открытой, разборчивость речи резко повышается. Помимо этого, выходя или входя, человек может не совсем плотно закрыть за собой дверь, что также повысит разборчивость речи. Для реализации этой меры необходимо, чтобы в течение всего времени, которое длится совещание, у двери комнаты дежурил охранник, осуществляющий контроль за дверью и коридором около входа в комнату, а также следить за тем, чтобы никто из посторонних не проник внутрь;

- по возможности проведение совещаний переносить на нерабочее время;

- любые работы в комнате, проводимые вне времени проведения конфиденциальных совещаний, например: уборка, ремонт бытовой техники, мелкий косметический ремонт, должен проводиться в обязательном присутствии сотрудника службы безопасности. Наличие этих сотрудников поможет обезопасить помещение от установки подслушивающих устройств сотрудниками организации или же посторонними лицами (электромонтер, рабочие, уборщица и так далее);

- после проведения совещания комната должна тщательно осматриваться, закрываться и опечатываться;

- между совещаниями комната должна быть закрыта и опечатана ответственным лицом.

Ключи от комнаты должны быть переданы дежурной смене охраны под расписку и храниться в комнате охраны, доступ к ним должен быть регламентирован руководством;

Перечисленные меры организационной защиты информации в комнате для совещаний считаются весьма действенными и не несут серьезных материальных затрат или проблем с персоналом и могут применяться как отдельно, так и совместно, что позволит значительно повысить степень защиты информации рассматриваемого объекта.

1.2 Механизмы взаимодействия аппаратных модулей системы

Как правило, все компоненты любой распределённой вычислительной системы (вне зависимости от сложности) взаимодействуют между собой по заранее заданным и настроенным протоколам.

На рисунке 1 представлен COM port.

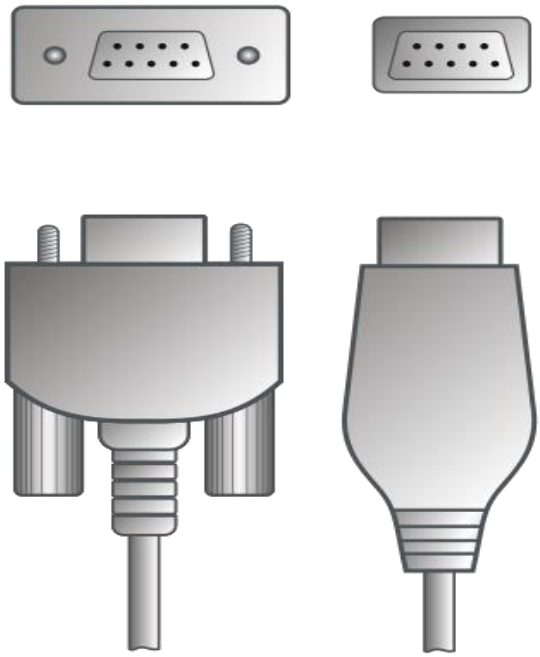

Рисунок 1 – Строение COM порта

Последовательный порт это устройства ввода-вывода (I/O device). Как устройство I/O это только путь для передачи данных из компьютера и в него. существует также множество других устройств ввода-вывода, таких как последовательные порты, параллельные порты, контроллеры дисков, сетевые карты, устройства универсальной последовательной шины USB, и т.п. Большинство компьютеров имеют один или два последовательный порта. Каждый имеет 9-ти контактный разъем (иногда 25-ти контактный). Программы могут отсылать данные (байты) через контакт отправки данных (output) и получать байты через другой контакт приема данных (input). Все остальные контакты служат для управления и земли.

Последовательный порт (serial port) это несколько больше чем просто разъем. Он преобразует, данные из параллельного представления в последовательное и меняет электрическое представление данных. Внутри компьютера, биты данных передаются в параллельном виде (используется несколько проводов для передачи данных одновременно). Последовательный поток данных это последовательность битов всего по одному проводу (такому как провод передачи и приема данных на разъеме последовательного порта). Для того и служит это устройство, чтобы создать такой поток данных из параллельного вида в последовательный и передать на контакт передачи данных (и соответственно наоборот).

Большинство электронных компонентов последовательного порта сосредоточено в одно компьютерном чипе (микросхеме) называемом UART.

Старые компьютеры используют 25-ти контактные разъемы, но только 9 контактов реально задействовано на сегодняшний день. Каждый из 9-ти контактов соединен обычно с проводом. за исключением двух проводов для передачи и приема данных, остальные используются для контроля и земли. Напряжение на каждом из контактов и проводов измеряется относительно

сигнальной земли. Поэтому минимальное количество проводом для двунаправленной передачи данных - 3. В редких случаях для работы может хватить и двух проводов (без сигнальной земли), однако это может привести к низкой производительности, и иногда к ошибкам при передаче данных.
Пример кросировки COM порта представлен на рисунке 2.

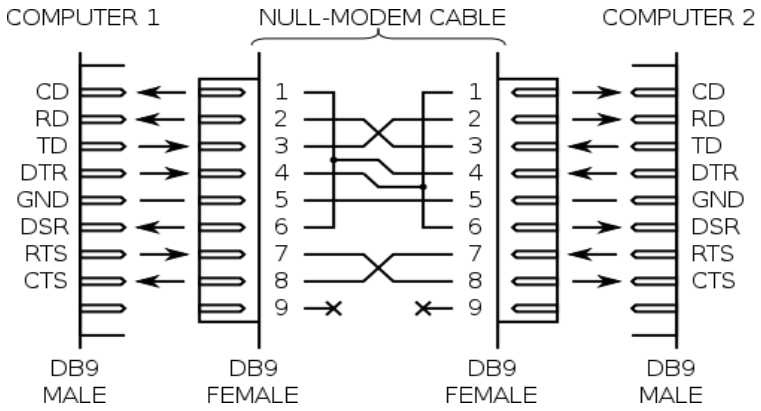

Рисунок 2 - Кроссировка кабеля

Остается еще несколько проводов, которые предназначены только для управления (контроля) и не используются для передачи данных. Все эти сигналы могли бы передаваться по одной линии, но вместо этого, для выделены отдельные провода. Некоторые (или все вместе) эти сигнальные линии называются "линии состояния модема". Линии состояния могут находиться в одном из двух состояний установленном (включено) +12 вольт или сброшенном (выключено) -12 вольт. Одни из этих проводов сигнализируют компьютеру о том, что нужно прекратить передачу данных через последовательный порт. Другие в свою очередь сигнализируют устройству, подключенному к последовательному порту, прекратить передачу данных в компьютер. Если подключено устройство это модем, то оставшиеся линии могут указывать модеме на то, что нужно занять телефонную линию или сигнализируют компьютеру о том, что соединение было установлено или что есть звонок на телефонной.

Выводы по первой главе

Открытые протоколы и библиотеки для open-source сред разработки позволяют объединить огромный ассортимент аппаратных сенсоров. Организовать передачу данных на верхние уровни представления информации. Хранение полученной информации с использованием MySQL server открывает новые горизонты. Вплоть до организации совместного доступа посредствам интернет.

2 Подбор аппаратной и програмной части, необходимой для реализации системы безопасности

2.1 Подбор программных модулей для построения комплексной системы

Так как одна из концепций open source solution подразумевает под собой использование не только самостоятельно разработанных программных, но и библиотек сторонних разработчиков, за время исследования был проведён подбор и настройка соответствующих библиотек в соответствии с требованиями к производительности вычислительной системы.

В качестве ядра системы был выбран Windows 7 — пользовательская операционная система семейства Windows NT. Программное обеспечение, необходимое для графического интерфейса выполнено в среде разработки Processing . Processing 1.0 — это бесплатное, открытое, кроссплатформенное ПО. Исходный архив включает в себя java-машину, сам интерпретатор, мини-IDE.

Основным связующим звеном между элементами системы и аппаратной частью является библиотека processing.serial. Используя эту библиотеку можно организовать проприетарный протокол, функционирующий только в рамках этого продукта или семейства продуктов.

За хранение данных отвечает MySQL server. MySQL — свободная реляционная система управления базами данных. MySQL является решением для малых и средних приложений. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей.

В роли WEB сервера выступает Apache. Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS. Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках. Ядро Apache включает в себя основные функциональные возможности, такие как обработка конфигурационных файлов, протокол HTTP и система загрузки модулей. Ядро (в отличие от модулей) полностью разрабатывается Apache Software Foundation.

2.2 Основные аспекты выбора оборудования для построения распределённой вычислительной системы

Выделение базовых требований к разрабатываемой системе является самой важной частью исследования. На данном этапе моделируется износоустойчивость модулей и компонентов, разрабатывается программное обеспечение, организующее корректную работу системы в целом при минимальном использовании памяти устройства. Как следствие, при

должном выполнении указанных требований достигается максимальная производительность всех компонентов.

2.2.1 Arduino UNO

Подбор модулей основывается на выделении основных критериев и требований:

Размер устанавливаемой платы должен быть как можно меньше, в силу ограниченности пространства.

Центральным узлом аппаратной части выступает Arduino UNO. Внешний вид платы представлен на рисунках 3 и 4.

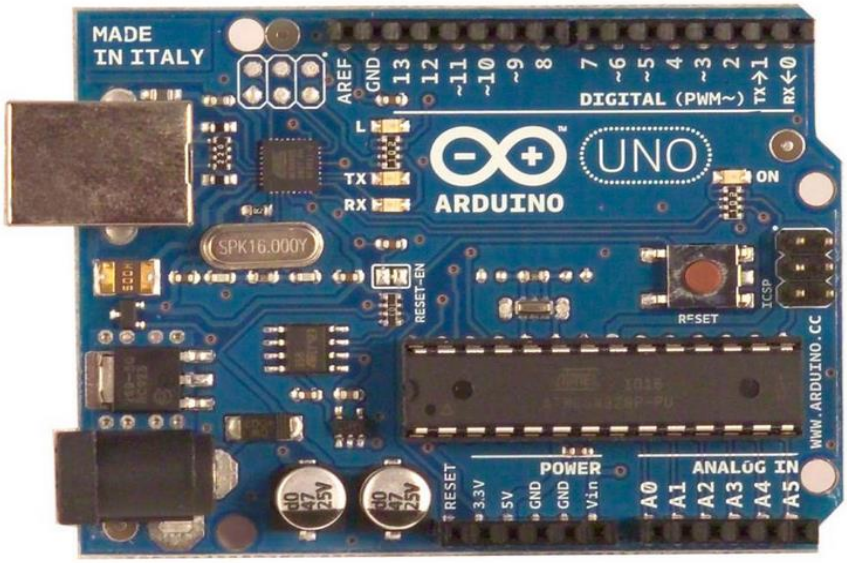

Рисунок 3 – Плата Arduino. Вид спереди

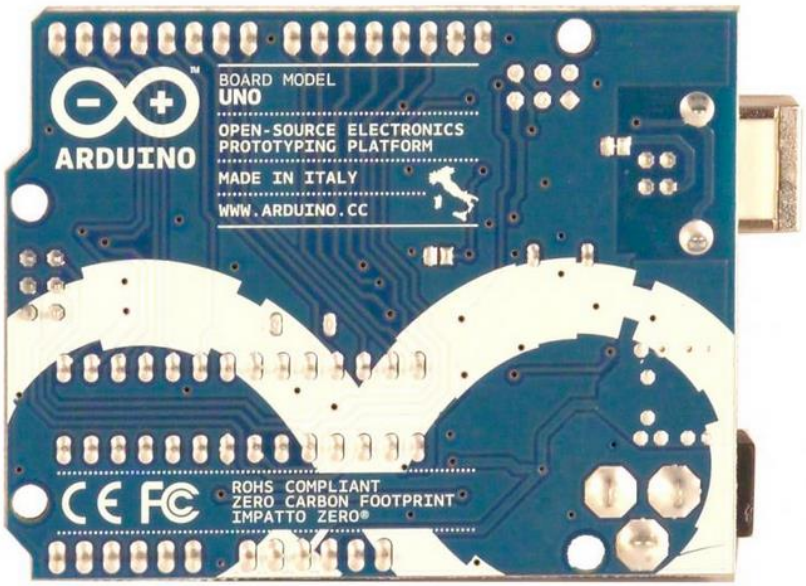

Рисунок 4 – Плата Arduino. Вид сзади

Arduino Uno контроллер построен на ATmega328 (техническое описание, pdf). Платформа имеет 14 цифровых вход/выходов (6 из которых могут использоваться как выходы ШИМ), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB, либо подать питание при помощи адаптера AC/DC или батареи.

Характеристики:

- микроконтроллер: ATmega328 ;
- рабочее напряжение: 5 В ;
- входное напряжение (рекомендуемое): 7-12 В ;
- входное напряжение (предельное): 6-20 В ;
- цифровые Входы/Выходы: 14 (6 из которых могут использоваться как выходы ШИМ) ;
- аналоговые входы: 6 ;
- постоянный ток через вход/выход: 40 мА ;
- постоянный ток для вывода 3.3 В: 50 мА ;
- флеш-память: 32 Кб (ATmega328) из которых 0.5 Кб используются для загрузчика ;
- ОЗУ: 2 Кб (ATmega328) ;
- EEPROM: 1 Кб (ATmega328) ;
- тактовая частота: 16 МГц.

Микроконтроллеры обычно не могут выдавать произвольное напряжение. Они могут выдать либо напряжение питания (например, 5 В), либо землю (т.е. 0 В). Но уровнем напряжения управляется многое: например, яркость светодиода или скорость вращения мотора. Для симуляции неполного напряжения используется ШИМ (Широтно-Импульсная Модуляция, англ. Pulse Width Modulation или просто PWM).

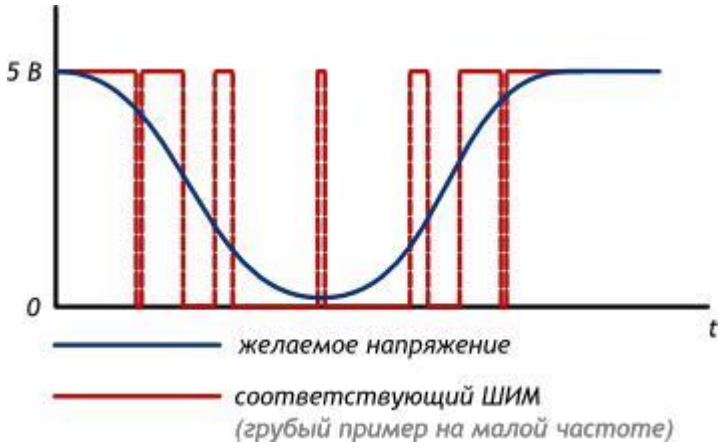

Рисунок 5 – Пример работы ШИМ

Выход микроконтроллера переключается между землёй и Vcc тысячи раз в секунду, то есть имеет частоту в тысячи герц. Человеческий глаз не

замечает мерцания более 50 Гц, поэтому нам кажется, что светодиод не мерцает, а горит в полсилы.

Аналогично, разогнанный мотор не может остановить вал за миллисекунды, поэтому ШИМ - сигнал заставит вращаться его в неполную силу.

Отношение времени включения и выключения называют скважностью (англ. duty cycle). При рассмотрении сценариев при напряжении питания Vcc равным 5 вольтам, можно получить следующие графики:

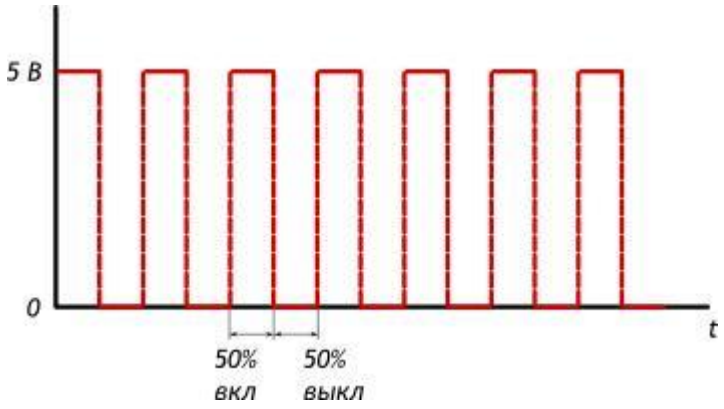

Рисунок 6 – Скважность при 2,5 В

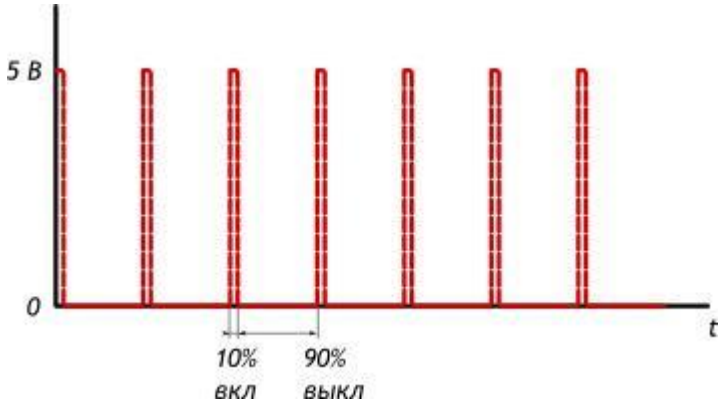

Рисунок 7 - Скважность при 0,5 В

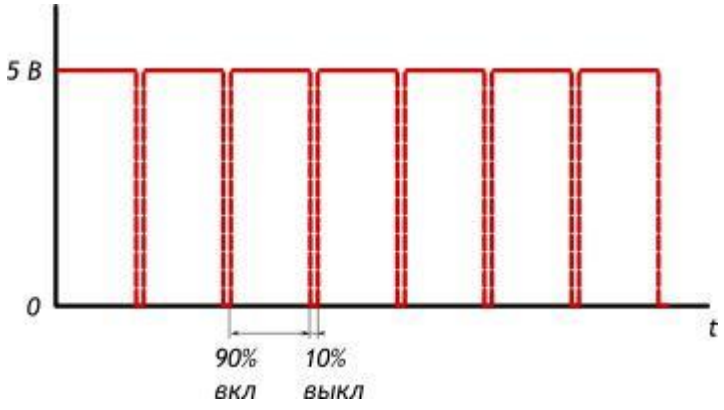

Рисунок 8 - Скважность при 4,5 В

2.2.2 Сервоприводы

В качестве устанавливаемых сервоприводов были выбраны моторы фирмы TowerPro MG90 . Данные сервоприводы представленны на рисунке 9. Они относятся к классу микро-сервоприводам и имеют следующие спецификации:

- размер: 23.2 × 12.5 × 22 мм ;
- вес:14 г ;
- скорость поворота: 0.12 сек/60 градусов (при питании 4.8В) и 0.10сек/60градусов (при питании 6В) ;
- рабочее напряжение: от 4.8В до 6В ;
- аналоговое подключение ;
- максимальный угол поворота: 180 градусов ;
- материал вала: металл ;
- материал шестерёнок: металл ;
- длина подключаемого провода: 20 см .

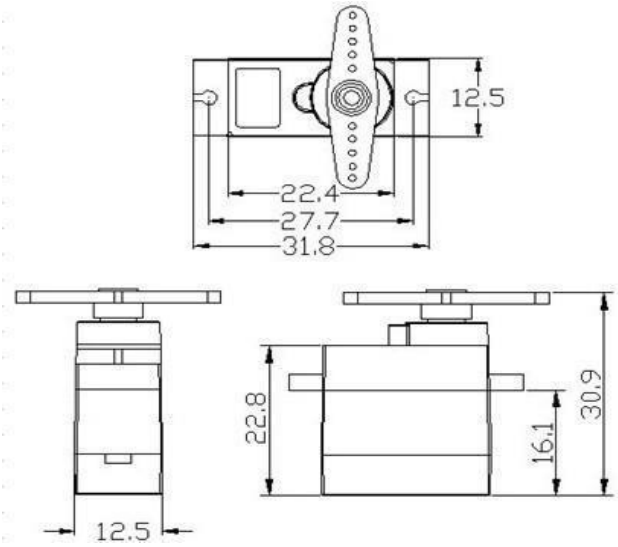

Рисунок 9 – Размер сервоприводов

Основным плюсом выбора являются металлические шестерни, которые имеют повышенную износоустойчивость, а так же наименьший люфт по сравнению с пластмассовыми.

2.2.3 Сонар

В качестве сенсора расстояния используется HC-SR04. Внешний вид представлен на рисунке 10.

Характеристики:

- напряжение питания: 5В ;
- ток покоя: <2мА ;
- эффективный угол обзора: <15° ;
- диапазон измерения дальности : 2–500 см ;
- разрешение датчика: 0.3см ;
- размеры (ДхШхВ): 45x20x15мм.

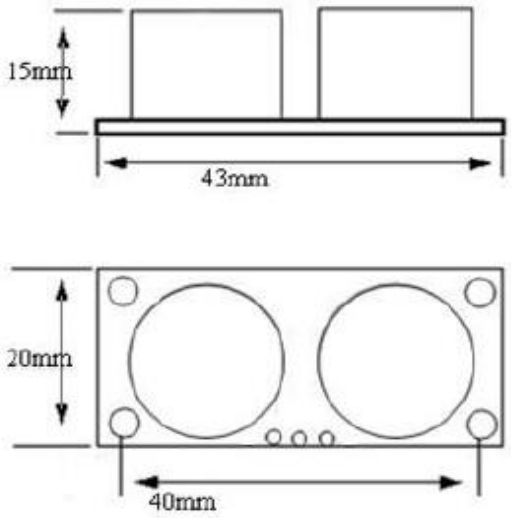

Рисунок 10 – Сонар

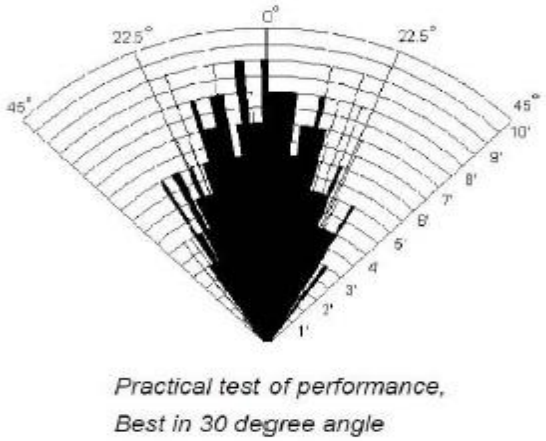

Рисунок 11 – Диаграмма направленности сонара

Сенсор излучает короткий ультразвуковой импульс (в момент времени 0), который отражается от объекта и принимается сенсором. Расстояние рассчитывается исходя из времени до получения эха и скорости звука в

воздухе. Осцилограмма работы сонара представленна на рисунке 12.

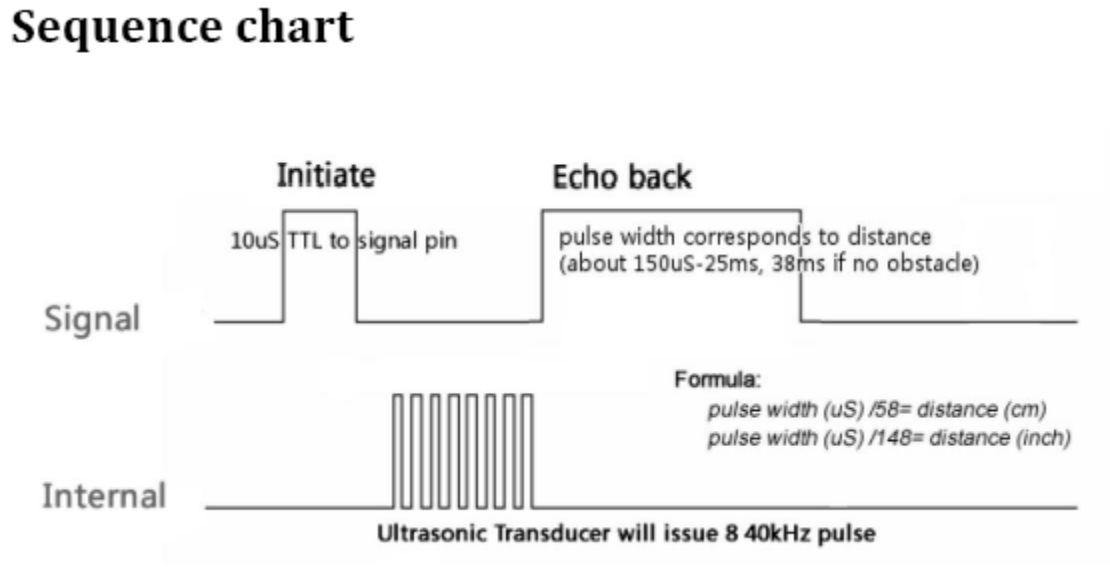

Рисунок 12 – Осциллограмма работы сонара

Сенсор получает сигнал эха, и выдаёт расстояние, которое кодируется длительностью электрического сигнал на выходе датчика (Echo).

Следующий импульс может быть излучён, только после исчезновения эха от предыдущего. Это время называется периодом цикла (cycle period). Рекомендованный период между импульсами должен быть не менее 50 мс. Если на сигнальный пин (Trig) подаётся импульс длительностью 10 мкс, то ультразвуковой модуль будет излучать восемь пачек ультразвукового сигнала с частотой 40кГц и обнаруживать их эхо. Измеренное расстояние до объекта пропорционально ширине эха (Echo) и может быть рассчитано по формуле, приведённой на графике выше.

PulseWidth (uS)/58 = distance (sm)

2.2.4 Инфракрасный датчик

Работа PIR-сенсора, т.е. пассивного ИК датчика основывается на измерении инфракрасного излучения от объектов. Датчик представлен на рисунке 13.

Работу можно разделить на два этапа:

1- Калибровка. При включении датчик измеряет инфракрасное излучение для получения эталонных значений.

2- Мониторинг. Датчик постоянно измеряет инфракрасное излучение и при отклонении от эталонного выдает единицу в порт.

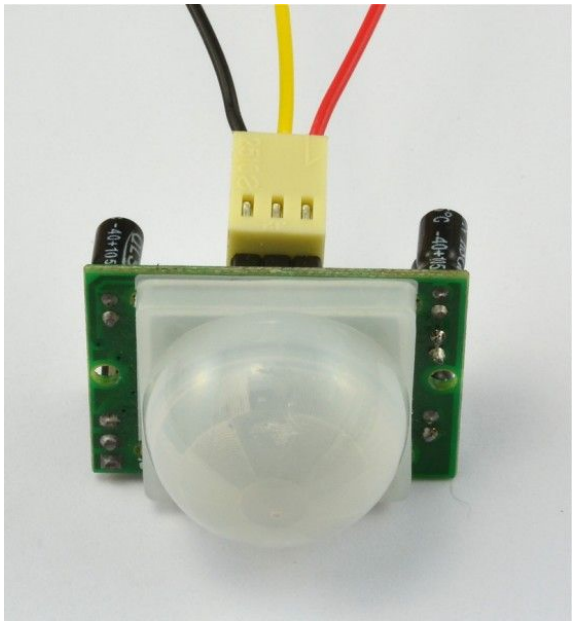

Рисунок 13 – Внешний вид инфракрасного сенсора

Основные технические характеристики:
- зона работы датчика: До 6 метров (110° x 70° область обнаружения) ;
- рабочее напряжение: 5 - 9В.

Модуль имеет 3 вывода (стандарта 2.54мм):
- GND: "-" питание;
- VCC: "+" питание;
- OUT: Вывод выходного сигнала.

Подключение датчика :
- GND на любой из GND пинов Arduino ;
- VCC на + 5 вольт на Arduino ;
- OUT на любой из цифровых входов/выходов платы Arduino.

2.2.5 Прецизионный датчик температуры и влажности

Измерение температуры и влажности происзодит с помощью прецизионного датчика DHT11. Внешний вид представлен на рисунке 14.

Представленный датчик имеет следущие параметры.

Относительная влажность:
- разрешение: 16Bit ;
- повторяемость: ± 1% относительной влажности ;
- точность: На 25 ± 5% относительной влажности ;
- взаимозаменяемость: полностью взаимозаменяемы ;
- долгосрочная стабильность: <± 0,5% RH .

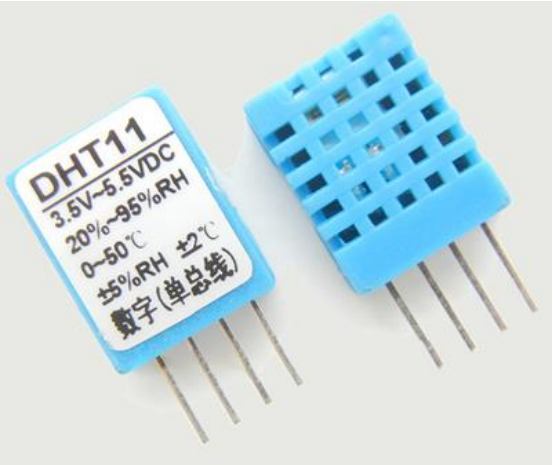

Рисунок 14 – Внешний вид DHT11

Температура:
- разрешение: 16Bit ;
- повторяемость: ± 0,2 ;
- диапазон: 0 - 50;
- электрические характеристики ;
- питание: DC 3,5 ~ 5,5 В ;
- ток питания: измерение 0.3mA ожидания 60μ ;
- период выборки: более 2 секунд.

Фотография датчика DHT11 в разрезе представленна на рисунке 15.

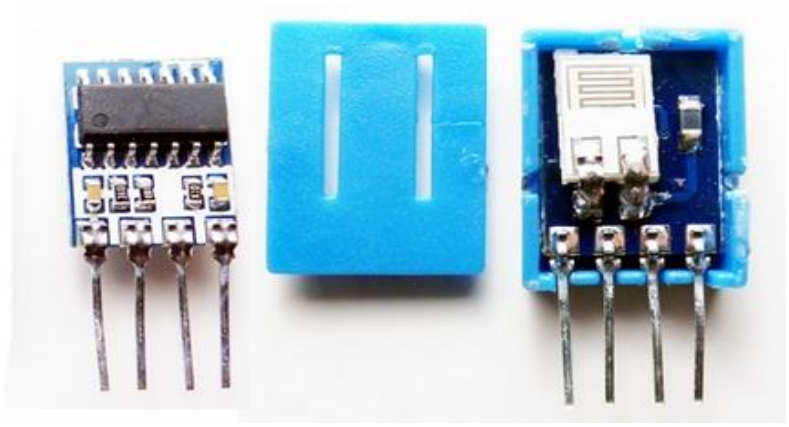

Рисунок 15 - Датчик DHT11 в разрезе

2.2.6 Фоторезистор

Фоторезистор – это датчик, электрическое сопротивление которого меняется в зависимости от интенсивности падающего на него света. Внешний вид представлен на рисунке 16.

Рисунок 16 – Фоторезистор

Чем интенсивней свет, тем больше создается свободных носителей зарядов и тем меньше становится сопротивление элемента. Два внешних металлических контакта фоторезистора идут через керамический материал основания к светочувствительной пленке, которая по своей геометрии и свойству материала определяет электрические свойства сопротивления. Так как фоточувствительный материал по природе с большим сопротивлением, то между электродами с тонкой извилистой дорожкой, при средней интенсивности света, получается низкое общее сопротивление элемента. Так же как и человеческий глаз, фоторезистор чувствителен к определенному диапазону длины волны света.

2.2.7 Датчик газа MQ2

Датчик газа, построенный на базе газоанализатора MQ2 позволяет обнаруживать наличие в окружающем воздухе углеводородных газов (пропан, метан, н-бутан), дыма (взвешенные частицы, являющиеся результатом горения), водорода и представлен на рисунке 17.

Рисунок 17 – Сенсор CO2

Датчик можно использовать для обнаружения утечек промышленного газа и задымления. Выходным результатом является аналоговый сигнал,

пропорциональный содержанию газов, к которым восприимчив газоанализатор. Чувствительность может быть настроена с помощью триммера на плате датчика.

Характеристики:

- напряжение питания: 5 В ;
- потребляемый ток: 160 мА ;
- диапазон измерений ;
- пропан: 0,2 – 5 промилле ;
- бутан: 0,3 – 5 промилле ;
- метан: 5 – 20 промилле ;
- водород: 0,3 – 5 промилле ;
- пары спиртов: 0,1 – 2 промилле.

Выводы по второй главе

При разработке системы носящий аппаратный характер необходимо учитывать возможности как каждого модуля по отдельности, так и при работе всей конструкции в целом. Любой модуль подбирается с учётом возможностей предыдущего: они должны либо дополнять друг друга, либо заменять (при необходимости). Функциональность программного обеспечения не менее важна – при достаточной оптимизации кода , даже простые библиотеки дают колоссальный результат. Обеспечивая надёжность и быстродействие.

3 Разработка и исследование системы безопасности

Общая схема системы представлена на рисунке 18 и представляет собой комплексное решение по организации безопасности, как от внешних угроз, так и от внутренних. О принципах функционирования аппаратной части пойдёт речь в главе 3.1 - Разработка принципов взаимодействия модулей аппаратной части. Непосредственно, стык между аппаратной и программной частях рассматривается в двух главах: в каждой со своей стороны соответственно. Способ реализации графического интерфейса, описание метода отправки данных в базу данных MySQL, организация почтовой рассылки и разработка WEB-интерфейса подробно описаны в главе 3.2 - Разработка принципов взаимодействия модулей программной части.

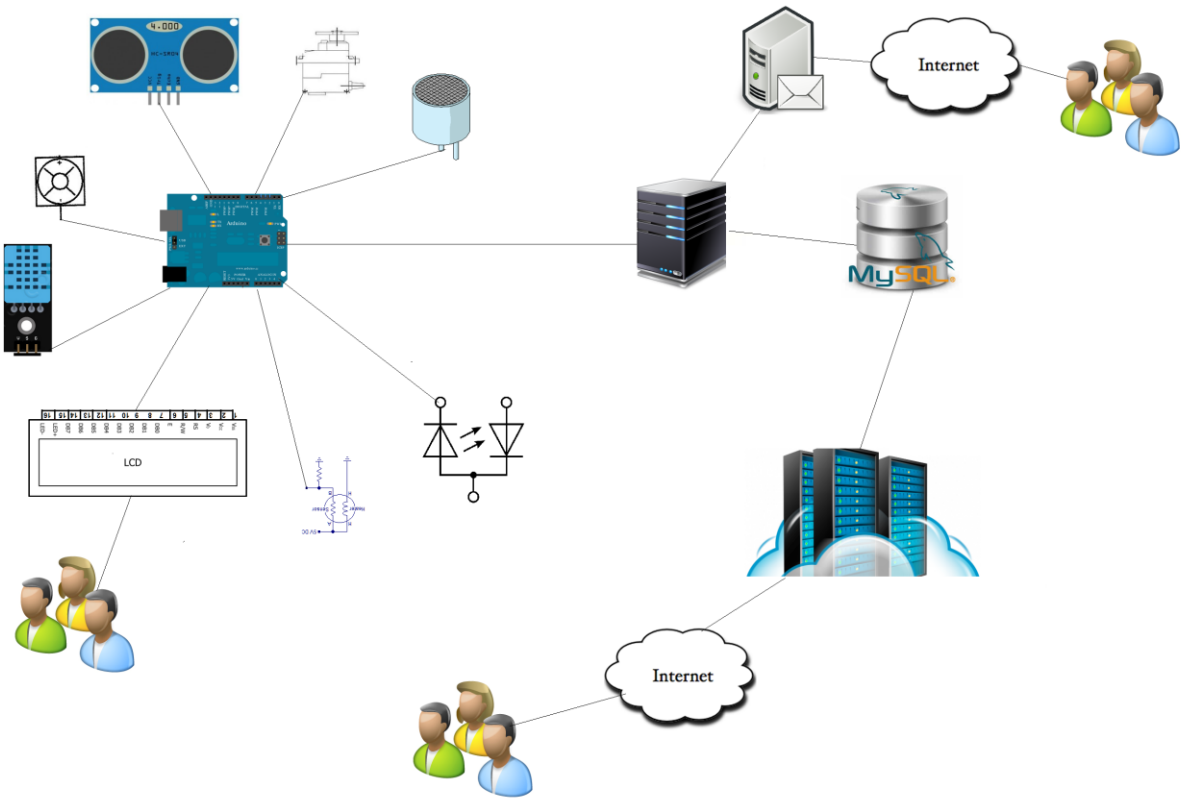

Рисунок 18 – Структурная схема системы безопасности

3.1 Разработка принципов взаимодействия модулей аппаратной части

Для обмена информацией между аппаратной и программной частью используется последовательный интерфейс на физическом уровне.

Контроллер посылает два типа сообщений:

Первый тип хранит информацию о лазерных приёмниках, данные сонара и угле поворота сервопривода.

Структура фрейма |L1|L2|L3|Sonar|ServoAngle| - Type 1

Второй тип сообщения хранит информацию о температуре, влажности, уровне шума, CO2, инфракрасного сенсора.

Структура фрейма |Noise|Temp|RH|CO2|PIR| - Type 2

При попадании фрейма на сторону графического интерфейса, производится разделение фреймов по признаку длины.

Длина Type 1 фрейма 74±5.

Длина Type 2 фрейма 37±5.

3.1.1 Подключение сонара

Схема подключения сонара представлена на рисунке 19.

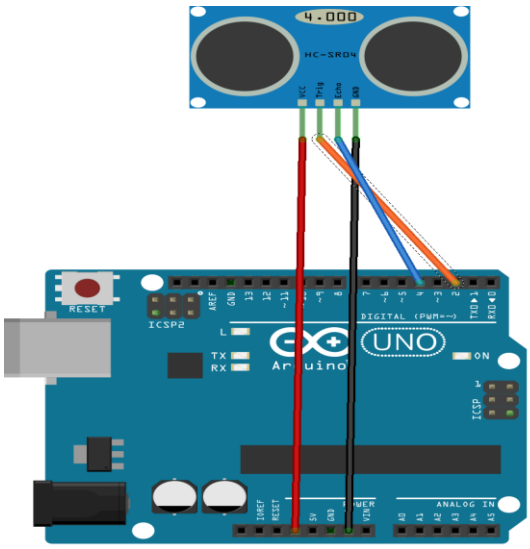

Рисунок 19 - Подключение сонара

Данная часть используется для определения расстояния до потенциального нарушителя. Применяется в связке с сервоприводом, работа которого будет рассмотрена позднее.

Для запуска измерения посылается короткий импульс (10 mS) на ножку Trig. Часть листанга программы работы сонара приведена ниже.

digitalWrite(distanseTrig, HIGH); // Подаем сигнал на выход микроконтроллера

delayMicroseconds(10); // Удерживаем 10 микросекунд

digitalWrite(distanseTrig, LOW); // Затем убираем

После замеряется длительности ответного сигнала, пропорциональная измеренному расстоянию. Програмно это реализованно следущим образом.

time_us=pulseIn(distanseEcho, HIGH); // Замеряем длину импульса

Так как у сенсора число миллисекунд в ответном сигнале не равно расстоянию в сантиметрах, то вводится поправочный коэффициент. Рекомендуемое значение 58, но оно может быть подправлено при необходимости. А так же при калибровке устройства.

distance_sm=time_us/53; // Пересчитываем в сантиметры

В случае моего датчика, этот коэффициент был выбран равным 53.

3.1.2 Организация слежения за границей периметра

На рисунке 20 представлен схематичный рисунок функционирования лазерного трассировщика. Расчёт происходит в первой и второй координатных четвертях.

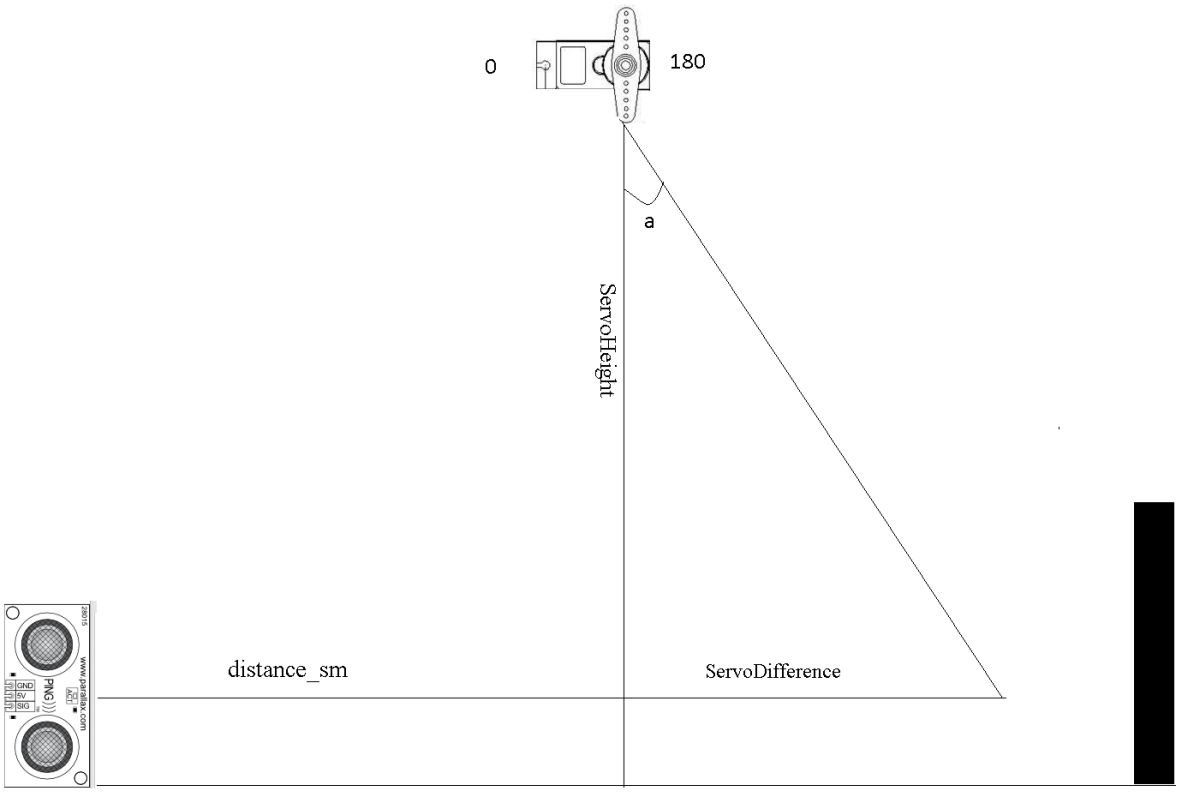

Рисунок 20 – Схематичное представление трасировщика

Первоначально плоскость разбивается на четыре зоны:

- за пределами контроля ;
- в первой координатной четверти ;
- ровно по центру ;
- во второй четверти.

В контроллере это реализовано по средствам нижеприведенного кода. Полный листинг приведен в приложении А.

```
if (distance_sm > 33) {Serial.print("33");} else {Serial.print(distance_sm);}
// Выводим на порт расстояние до объекта
Serial.print(" distanse,");
 if (distance_sm <= 16)
 {
   DistanseLess16();
 }
 else if (distance_sm >= 33)
 {
   DistanseOut(); // вышел за контролируемый предел
 }
 else if (distance_sm >= 18)
 {
```

```
    DistanseMore18();
  }
  else
  {
    DistanseBetween();
  }
}
```

В каждой четверти расчёт ведётся по принципу

digitalWrite(myservoLsr, HIGH); // Зажигаем лазер на сервоприводе
ServoDifference = distance_sm - MiddleDistanse; // Находим расстояние от центральной точки до объекта
ServoDivision = ServoDifference/ServoHeight; // Находим отношение противолежащего катета к прилежащему
ServoAtan = atan(ServoDivision); // Берём аркТангенс от отношения катетов
ServoAtaDeg = ServoAtan*180/3.1415; // Переводим из радиан в градусы
Serial.print(90+int(ServoAtaDeg)); // Выводим на серийный интерфейс угол поворота сервопривода с учетом координатной четверти
Serial.println(" RealAngle."); //
myservo.write(90+int(ServoAtaDeg)); // Задаем сервоприводу рассчитанный угол поворота
LcdDistanse (distance_sm); // Выводим на LCD экран угол поворота сервопривода
Для первой четверти убирается поправка в 90 градусов.

3.1.3 Инфракрасный датчик движения

Возможность раннего обнаружения живых объектов, является очень ценным свойством для систем безопасности. Оснащенный линзой Френеля PIR датчик реагирует на малейшие изменения в окружающем инфракрасном поле.

На рисунке 21 представлена схема интеграции датчика к основной конструкции.

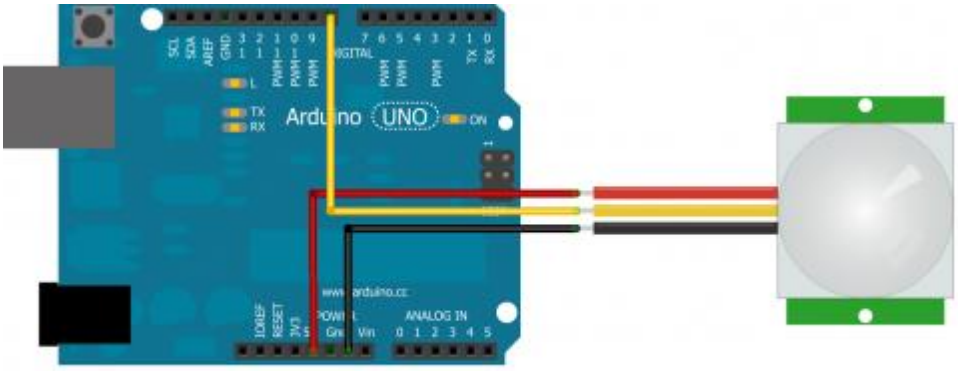

Рисунок 21 – Подключене инфракрасного датчика

На своем борту датчик имеет контроллер. Подстройка осуществляется двумя резисторами. Первый отвечает за длительность импульса при обнаружении нарушителя. Второй за растр угла зоны обнаружения.

Для каждого случая эти параметры могут быть подстроены индивидуально.

При детектировании микроконтроллером высокого уровня сигнала, по UART интерфейсу она незамедлительно передается в сервер обработки.

Обработка информации с PIR датчика происходит следущим образом:

```
    pirRes = digitalRead(pirPin); //read state of the PIR
    if (pirRes == LOW) {
      Serial.println("No motion."); //if the value read is low, there was no motion
      lcd.setCursor(11, 0);
      lcd.print("         ");
      lcd.setCursor(11, 0);
      lcd.print("No motion");
    }
    else {
      Serial.println("Motion."); //if the value read was high, there was motion
      lcd.setCursor(11, 0);
      lcd.print("         ");
      lcd.setCursor(11, 0);
      lcd.print("Motion!");
    }
}
```

3.1.4 Контроль температуры и влажности

За контролем температуры и влажности следит прецизионный датчик DHT11. Представленный на рисунке 22.Такой род контроля необходим , так как в зимний период времени во время проникновения возможны резкие перепады температур. Что с высокой вероятностью демаскируют злоумышленника.

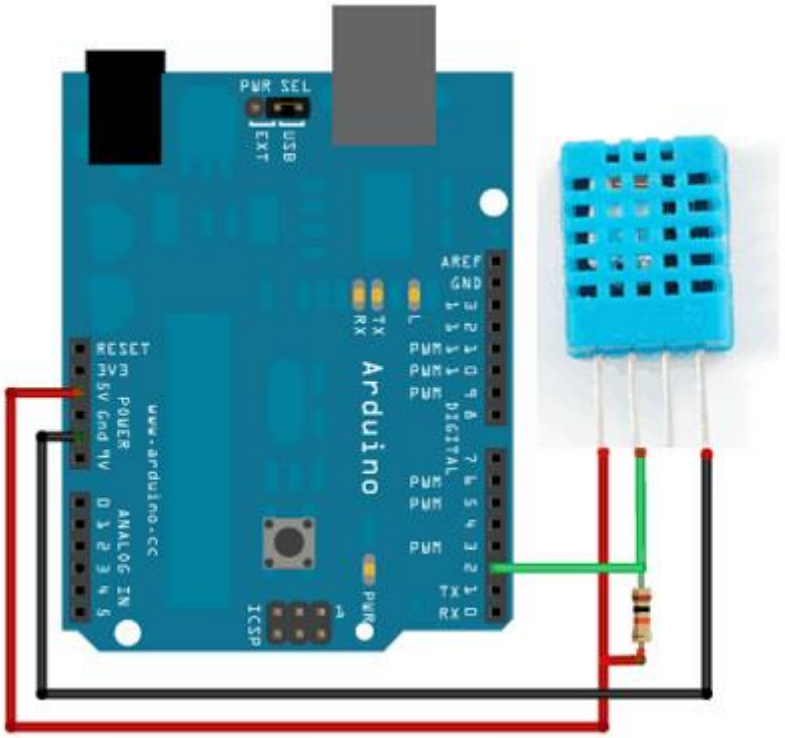

Рисунок 22 – Подключение сенсора температуры и влажности

Выбор пал на этот сенсор в силу высокой точности, экономии ножек контроллера, низкого энергопотребления, отсутствием необходимости использования АЦП.

Коммуникация осуществляется при помощи библиотеки взаимодействия с датчиками семейства DHT. Библиотека оснащена обработчиком ошибок. Под процессом взаимодействя понимается считывание показаний температуры и владности окружающей среды и передача этой информации на секрвер графического интерфейса и ЖД экран. Часть листинга взаимодействия приведена ниже.

```
#include <dht11.h>
 dhtRes = DHT.read(DHT11_PIN);    // Чтение данных с сенсора DHT11
switch (dhtRes){
case DHTLIB_OK: // Если всё хорошо то вывести данные
        LcdDhtTemp(DHT.temperature);
        LcdDhtRH(DHT.humidity);
        Serial.print(DHT.temperature);
        Serial.print(" C,");
        Serial.print(DHT.humidity);
        Serial.print(" RH,");
break;
case DHTLIB_ERROR_CHECKSUM:
        Serial.println("Checksum error, \t");
```

```
    break;
    case DHTLIB_ERROR_TIMEOUT:
        Serial.println("Time out error, \t");
break;
default:
    Serial.println("Unknown error, \t");
break;
  }
```

3.1.5 Датчик CO2

Сенсор MQ2 позволяет детектировать в воздухе уровни угарного газа, либо природного газа. Что дает возможность предотвратить утечки газа в трубопроводе, определять уровень безопасной концентрации CO2 после аварийного пожаротушения, включать сирены эвакуации при превышении концентрации угарного газа в жилых и рабочих помещеньях.

Схема подключения представлена на рисунке 23.

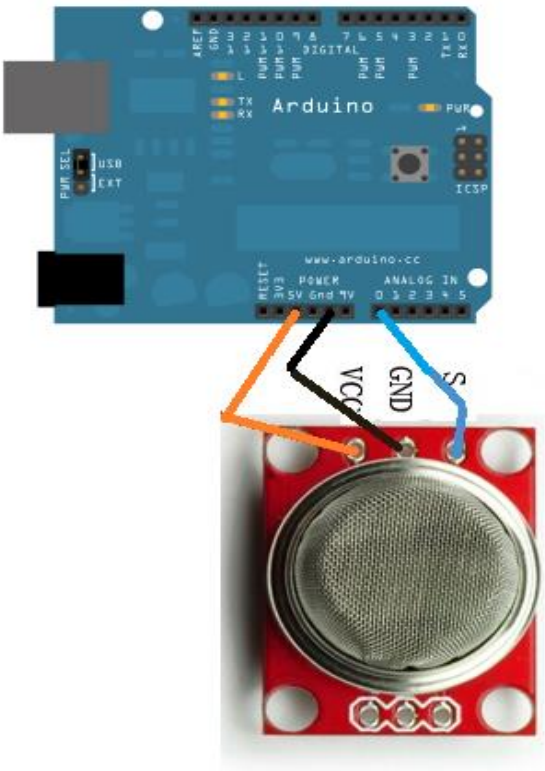

Рисунок 23 – Схема подключения датчика CO2

Подстройка датчика осуществляется при помощи переменного резистора, установленного на корпус. Данные мероприятия проводятся в момент пуско-наладки и калибровки системы. Пороги срабатывания могут быть настроены в зависимости от требований заказчика.

Коммуникация с сенсором проходит при помощи 10-и битного АЦП, входящего в состав микроконтроллера. Часть листинга взаимодействия микроконтроллера с датчиком дыми приведена ниже.

CO2_data = map(analogRead(MQ2_pin), 0, 1023, 0, 100); // Считываем значение с датчика CO2, и масштабируем по шкале от 0 до 100

Serial.print(CO2_data); // Выводим нормированные данные в серийный интерфейс

Serial.print(" CO2");

Serial.print(",");

LcdMQ2(CO2_data); // Выводим нормированные данные на LCD экран

Масштабирование данных происходит по шкале от 0 до 100.

3.1.6 Лазерные рубежи защиты

Защита границ вверенной территории и обнаружение злоумышленника на стадии раннего проникновения отводится лазерным датчикам пересечения.

Принципиальная схема представлена на рисунке 24.

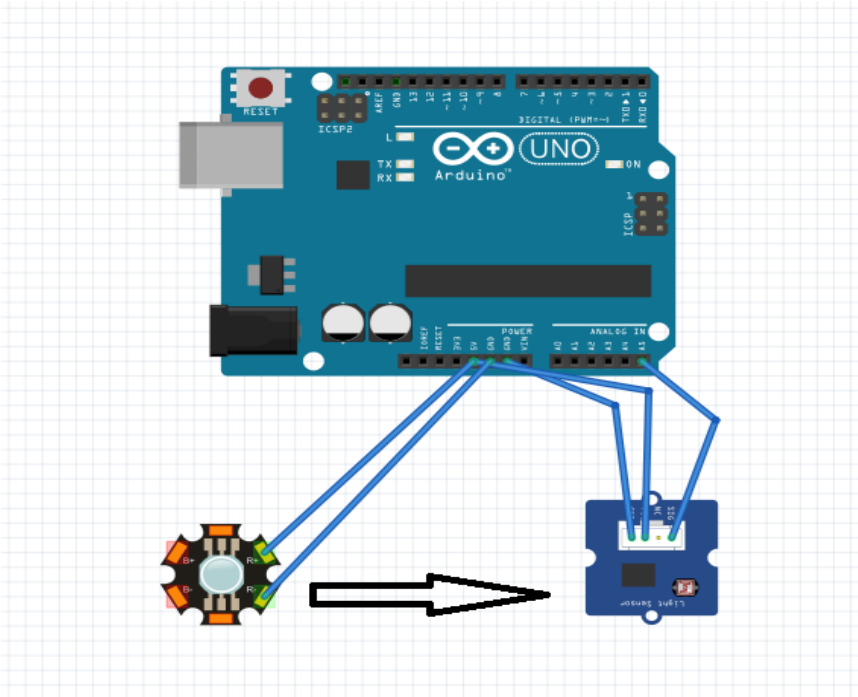

Рисунок 24 – Схема подключения сенсора оазерного рубежа

Особенностью датчиков на приемной стороне является их подключение не к цифровым, а к аналоговым входам. Эта особенность позволяет не только засечь пересечение, но и отслеживать флуктуации на приемной стороне.

Так как производится полномасштабное логирование информации, появляется возможность рассчитать джиттер. Так же выставить пороги срабатывания по джиттеру.

Все необходимые данные выводятся на LCD 20x4 экран.

Внешний вид экрана данной серии приведен на рисунке 25.

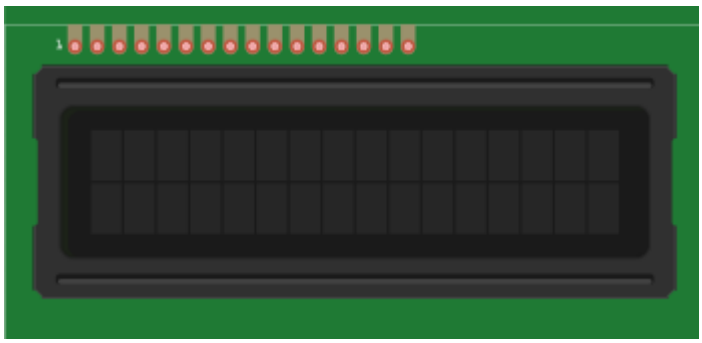

Рисунок 25 – Строчный экран LCD 16x2

3.2 Разработка принципов взаимодействия модулей программной части

3.2.1 Графический пользовательский интерфейс

Графический интерфейс реализован на компиляторе Processing на рисунке 26.

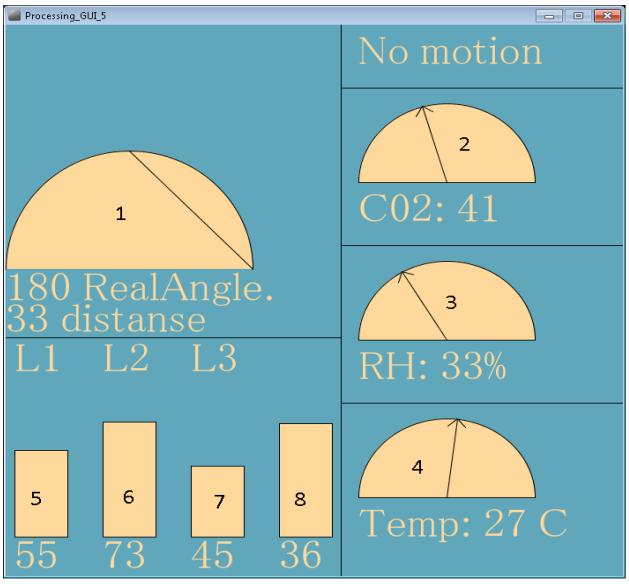

Рисунок 26 – Графический интерфейс написанный на Java

Анимированное меню состоит из следующих секций:

- индикатор положения лазера относительно горизонта;
- уровень CO2;
- относительная влажность воздуха;
- температура;
- уровень приёма первого лазера;
- уровень приёма второго лазера;
- уровень приёма третьего лазера;
- уровень шума.

Пороги срабатывания лазеров выставляются в контроллере и графическом интерфейсе отдельно. Пороги внутри контроллера отвечают за

изменения на LCD экране. В то время как пороги внутри кода графического интерфейса помимо дополнительной индикации инициируют отправку электронной почты. В теле письма содержится информация о номере лазерного рубежа, на который было совершено нападение.

3.2.2 Логирование данных в MySQL

В силу необходимости ведения лога и статистики об охране вверенного объекта, необходимо записывать историю состояний датчиков. Предоставленная система позволяет вести такую запись с частотой 10 записей в секунду.

Для этих целей идеально подходит MySQL server.

Настраивается интерфейс между графической оболочкой и MySQL как на рисунке 27.

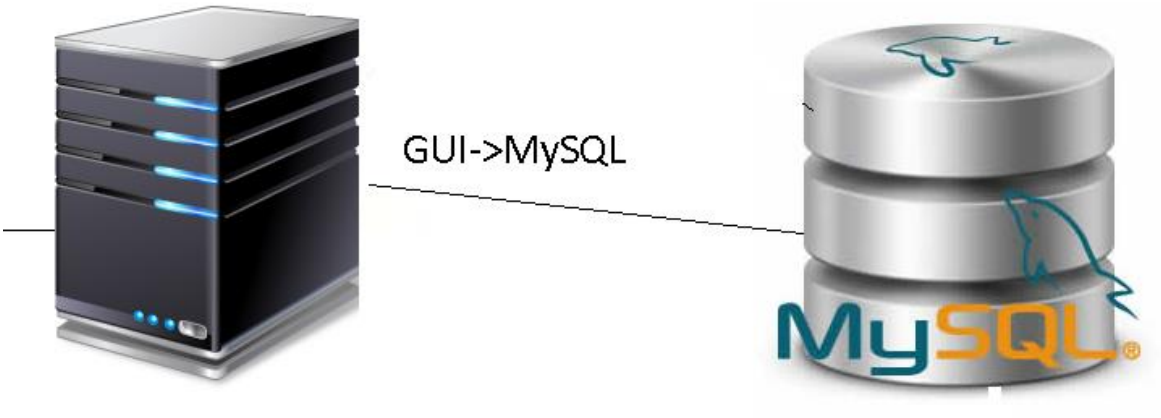

Рисунок 27 – Интерконнект между графической оболочкой и MySQL

Взаимосвязь MySQL и графическим интерфейсом происходит по средствам библиотеки import de.bezier.data.sql.*;

Данные узлы могут находиться на единой машине, так и быть разнесёнными и использовать для соединения Internet. Естественно при таком решении должен быть предоставлен IP Sec туннель. Так как передается частная информация.

Была написана функция, в аргументах содержащая имя таблицы и передаваемые туда данные.

Имя таблицы меняется в зависимости от характера данных.

Пример взаимодействия java оболочки и MySQL server приведена ниже.

```
MySQL msql; // Create object
String user     = "root";
String pass     = "";
String database = "Arduino";
```

```
String host = "localhost";
void Sql( String curTabl,String Col1,String Col2,String Col3, String Col4, String Col5) {
  if ( msql.connect() )
  {
   msql.query( "INSERT INTO " + curTabl + "  VALUES(NULL, " + Col1 + ", "+ Col2 + ", "+ Col3 + ", "+ Col4 + ", "+ char(34)+ Col5 + char(34)+", NULL);");
   //println("INSERT INTO " + curTabl + "  VALUES(NULL, " + Col1 + ", "+ Col2 + ", "+ Col3 + ", "+ Col4 + ", "+ Col5 +", NULL);");
  }
 }
```

3.3.3 Представление данных в WEB сервере

Бывают случаи, в которых требуется множественный доступ к информации.

Например для возможности мониторинга группой людей, или же если требуется иметь возможность мониторинга находясь в любой точке планеты.

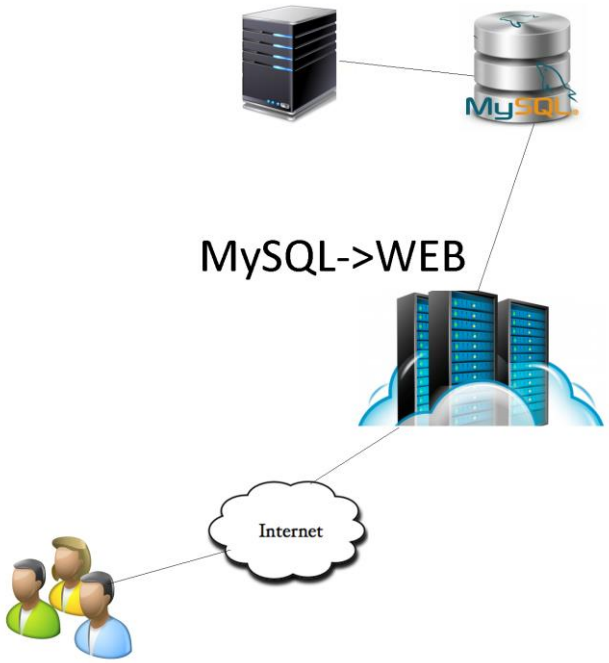

Рисунок 27 – WEB форма представления информации.

Для этих целей был использован WEB сервер Apache .

Стык между ним с базой данных был выполнен по средствам mysql_connect.

В следствии распределенности узлов системы, данные могут независимо обрабатываться на разных этапах следования .

WEB форма является наиболее приемлемой, с точки зрения её унифицированности и кросс-платформенности. Давая возможность для

просмотра информации как мобильным устройствам разных вендоров , так и стационарным персональным компьютерам с различными операционными системами.

Выводы по третьей главе

При разработке сложных охранных систем с модульной конструкцией требуется разделение процессов обработки информации. Нецелесообразно нагружать один блок всеми задачами. Перед выполнением макета объекта необходимо сделать модель , проверив её работоспособность в идеальных условиях. В данном деле помогает программный пакет ISIS Proteus и Fritzing. Fritzing, бесплатное ПО ориентированное на макетирование из плат Arduino и Arduino-совместимых продуктов.

Заключение

Изучение и последующее построение собственной лазерной системы безопасности позволяет более глубоко проникнуть в тонкости организации охранных рубежей и контроля территории. Учитывая актуальность данного направления в наши дни и высокую востребованность на рынке в связи с ростом уровня преступности, можно сказать, что подобные технологии будут активно развиваться и дальше, причём в самых разных областях: от разработки и поддержки комплексов организации безопасности для крупных компаний, до рынка мелкого и среднего бизнеса.

Базируясь на модульной конструкции и концепции взаимозаменяемости деталей, дополненной политикой открытого кода – достигается высокая надёжность и быстродействие системы при сохранении приемлемой стоимости. Достигается это тем, что минимизируются издержки на стоимость программного обеспечения.

В данной работе было проведено исследование, посвященное возможности объединения низкоуровневого программирования с верхними формами хранения и представления данных. Такой вид интеграции открывает новые горизонты в построении охранных систем. Так как появляется возможность наращивания функционала сенсоров, по мере их появления или по мере роста потребностей заказчика.

Перечень сокращений

В настоящей магистерской диссертации применяют следующие термины с соответствующими определениями:

IDE – Integrated Development Environment, интегрированная среда разработки

HTTP - Hypertext Transfer Protocol

PWM - Pulse Width Modulation, широтно-импульсная модуляция

UART - Universal Asynchronous Receiver-Transmitter

SQL - structured query language

WWW - world wide web

POP3 - Post Office Protocol Version 3

ИК - инфракрасный порт

СЗИ - средства защиты

Vcc - Collector supply voltage

ШИМ – широтно импульсная модуляция

Список литературы

1 Майк Шмидт. Arduino - 08.05.2014

2 Марголис М. Рецепты Arduino. Margolis M. – Arduino Cookbook – 2011

3 Долгий А. Разработка и отладка устройств на микроконтроллерах / Радио» – №5. – 2001.

4 Зайцев А.П., Шелупанов А.А. Технические средства и методы защиты информации / Машиностроение – 2009.

5 Р.Стюарт Болл – Аналоговые интерфейсы микроконтроллеров (Программируемые системы) 2007.

6 Кабатов Д.А. «Что такое облачные сервисы для небольших компаний?» URL: http://www.moysklad.ru/chto_takoe_oblachnye_servisy/ (Дата обращения: 21.02.2015)

7 Облачные технологии // Wikipedia.URL: https://ru.wikipedia.org/wiki/Облачные_вычисления / Дата обращения: 13.02.2015)

8 Оксер Дж., Блемингс Х. Практика для Arduino: Крутые проекты для доступных плат / Practical Arduino: Cool Projects for Open Source Hardware - Oxer J., Blemings H

9 Майк МакРобертс . Arduino для начинающих»// Arduino starter kit manual - Mike McRoberts

10 Улли Соммер. Программирование микроконтроллерных плат Arduino/Freeduino

11 Знакомство с Arduino (перевод книги "Getting Started with Arduino")

12 Чарльз Платт . Электроника для начинающих

13 В. Эванс . Ардуино Блокнот программиста

14 Полярная система координат. URL:https://ru.wikipedia.org/wiki/Полярная_система_координат (Дата обращения 5.04.2015)

Приложение А
Код микроконтроллера

```
//////////Список используемых библиотек
#include <LiquidCrystal.h>
#include <Servo.h>
#include <Math.h>
#include <dht11.h>
//Инициализация дальномера
const int distanseTrig = 6;
const int distanseEcho = 5;
unsigned int time_us = 0;
unsigned int distance_sm = 0; //Расстояние в см до объекта
//Инициализация LCD экрана
//RS, E, DB4, DB5, DB6, DB7
LiquidCrystal lcd(4, 2, 7, 8, 12, 13);
byte pipe[8] = {
        B00100,
        B00100,
        B00100,
        B00100,
        B00100,
        B00100,
        B00100,
        B00100,
};
byte smile[8] = {
        B00000,
        B10001,
        B00000,
        B00000,
        B10001,
        B01110,
        B00000,
};
byte degree [8] = {
        B11000,
        B11000,
        B00111,
        B01000,
        B01000,
        B01000,
```

```
        B01000,
        B00111,
};
byte AllOk [8] = {
        B00000,
        B00001,
        B00011,
        B10110,
        B11100,
        B01000,
        B00000,
        B00000,
};
byte NotOK [8] = {
        B00000,
        B11011,
        B01110,
        B00100,
        B01110,
        B11011,
        B00000,
        B00000,
};

//Инициализация PIR приемника
int pirPin = 3;
int pirRes;
//Инициализация сервопривода
Servo myservo;
#define myservoPin 9
#define myservoLsr 11
float ServoDifference, ServoDivision , ServoAtan , ServoAtaDeg , ServoAngle,
RealAngle  ;
int MiddleDistanse = 16;
int ServoHeight = 21 ;
//Инициализация DHT11
int dhtRes;
dht11 DHT;                                          // Объявление переменной
класса dht11
#define DHT11_PIN 10                          // Датчик DHT11 подключен к
цифровому пину номер 6
//Инициализация датчика громкости
```

```
int volumeThreshold = 35;                // Порог громкости
int volumeRes;                                    // Переменная содержащая результаты измерения громкости
#define volumnePin A0                    // Датчик громкости подключен к A0
//Инициализация газоанализатора
#define MQ2_pin A1
int CO2_data;
//Инициализация лазерных сенсоров
#define LaserSensor1_pin A4
#define LaserSensor2_pin A2
#define LaserSensor3_pin A3
int LaserSensor1; // Переменная в которой хранятся данные в лазерном рубеже защиты № 1
int LaserSensor2; // Переменная в которой хранятся данные в лазерном рубеже защиты № 2
int LaserSensor3; // Переменная в которой хранятся данные в лазерном рубеже защиты № 3
int LaserThreshold = 85; // Порог срабатывания лазерной системы безопастности

//Переменная таймера для проверки температуры
float DhtCurTime = 100; //Стартовое значение числа проходов , обнуляется в первом цикле
float DhtTimeOut = 100; //Число проходов - при достижении которого необходимо опросить датчики
//
void setup()
{
//Distance meter
pinMode(distanseTrig, OUTPUT);
pinMode(distanseEcho, INPUT);
//Servo
myservo.attach(myservoPin,800,2750); // Подключение серва и задание констант для мин и макс углов поворота
pinMode(myservoLsr, OUTPUT);
//LCD
lcd.begin(20, 4); // Задание размера дисплея
lcd.clear();
lcd.noBlink();
lcd.createChar(0, pipe);
lcd.createChar(1, smile);
```

```
lcd.createChar(2, degree);
lcd.createChar(3, AllOk);
lcd.createChar(4, NotOK);
Smileee(); // Print 4 smile
//UART
Serial.begin(9600);
//PIR
pinMode(pirPin,INPUT);
}

void loop()
{
QueryFromLaserSensor();
checkDistanse();
//if (distance_sm>= 33) {checkPIR();}

if (DhtCurTime > DhtTimeOut && distance_sm>= 33)
{GlobalCheck(); DhtCurTime=0;}
else {DhtCurTime =++ DhtCurTime;}          // инкрементируем счётчик
который периодически делает проверку общих датчиков
}


//Check Distance
void checkDistanse(void) {
digitalWrite(distanseTrig, HIGH); // Подаем сигнал на выход
микроконтроллера
delayMicroseconds(10); // Удерживаем 10 микросекунд
digitalWrite(distanseTrig, LOW); // Затем убираем
time_us=pulseIn(distanseEcho, HIGH); // Замеряем длину импульса
distance_sm=time_us/53; // Пересчитываем в сантиметры
if (distance_sm > 33) {Serial.print("33");} else {Serial.print(distance_sm);} //
Выводим на порт расстояние до объекта
Serial.print(" distanse,");
  if (distance_sm <= 16)
  {
     DistanseLess16();
  }
  else if (distance_sm >= 33)
  {
     DistanseOut(); // вышел за контролируемый предел
```

```
  }
  else if (distance_sm >= 18)
  {
     DistanseMore18();
  }
  else
  {
     DistanseBetween();
  }
}

//More than 18
void DistanseMore18(void) {
digitalWrite(myservoLsr, HIGH); // Зажигаем лазер на сервоприводе
ServoDifference = distance_sm - MiddleDistanse;  // Находим расстояние от центральной точки до объекта
//Serial.print("ServoDifference ");
//Serial.print(ServoDifference);  // Выводим расстояние от центральной точки до объекта
//Serial.print("  ");
ServoDivision = ServoDifference/ServoHeight; // Находим отношение противолежащего катета к прилежащему
//Serial.print("ServoDivision "); //
//Serial.print(ServoDivision); //  Выводим отношение противолежащего катета к прилежащему
//Serial.print("  ");  //
ServoAtan = atan(ServoDivision); // Берём аркТангенс от отношения катетов
//Serial.print("ServoAtan "); //
//Serial.print(ServoAtan);  // Выводим аркТангенс от отношения катетов
//Serial.print("  ");  //
ServoAtaDeg = ServoAtan*180/3.1415; // Переводим из радиан в градусы
//Serial.print("ServoAtaDeg ");  //
//Serial.print(ServoAtaDeg); // Выводим на серийный интерфейс угол поворота сервопривода без учета координатной четверти
//Serial.print("  ");  //
Serial.print(90+int(ServoAtaDeg)); // Выводим на серийный интерфейс угол поворота сервопривода с учетом координатной четверти
Serial.println(" RealAngle."); //
myservo.write(90+int(ServoAtaDeg));  // Задаем сервоприводу рассчитанный угол поворота
```

```
LcdDistanse (distance_sm);  // Выводим на LCD экран угол поворота сервопривода
}

//Less than 16
void DistanseLess16(void) {
digitalWrite(myservoLsr, HIGH);  // Зажигаем лазер на сервоприводе
ServoDifference = MiddleDistanse - distance_sm;  // Находим расстояние от центральной точки до объекта
//Serial.print("ServoDifference ");
//Serial.print(ServoDifference);  // Выводим расстояние от центральной точки до объекта
//Serial.print("  ");
ServoDivision = ServoDifference/ServoHeight; // Находим отношение противолежащего катета к прилежащему
//Serial.print("ServoDivision ");
//Serial.print(ServoDivision); //  Выводим отношение противолежащего катета к прилежащему
//Serial.print("  ");
ServoAtan = atan(ServoDivision); // Берём аркТангенс от отношения катетов
//Serial.print("ServoAtan ");
//Serial.print(ServoAtan);  // Выводим аркТангенс от отношения катетов
//Serial.print("  ");
ServoAtaDeg = ServoAtan*180/3.1415; // Переводим из радиан в градусы
//Serial.print("ServoAtaDeg ");
//Serial.print(ServoAtaDeg); //Выводим на серийный интерфейс угол поворота сервопривода без учета координатной четверти
//Serial.print("  ");
Serial.print(90-int(ServoAtaDeg)); // Выводим на серийный интерфейс угол поворота сервопривода с учетом координатной четверти
Serial.println(" RealAngle."); //
myservo.write(90-int(ServoAtaDeg));  // Задаем сервоприводу рассчитанный угол поворота
LcdDistanse (distance_sm);  // Выводим на LCD экран угол поворота сервопривода
}

//Between16and18
void DistanseBetween(void) {
digitalWrite(myservoLsr, HIGH);  // Зажигаем лазер на сервоприводе
```

```
ServoDifference = distance_sm;   // Объект расположен в пределах центральной точки, расстояние выводится до начала точки отсчёта
//Serial.print("ServoDifference "); //
//Serial.print(ServoDifference);  //Выводим расстояние от объекта до начала точки отсчёта
//Serial.print("  ");
Serial.print(90); // Выводим на серий порт угол поворота сервопривода
Serial.println(" RealAngle."); //
myservo.write(90);  // Задаем сервоприводу рассчитанный угол поворота
LcdDistanse (distance_sm); // Выводим на LCD экран угол поворота сервопривода
}

//DistanseOut - ПП выполняемая в случае если расстояние од объекта превышает 33 см
void DistanseOut(void) {
digitalWrite(myservoLsr, LOW);        // Тушим лазер на сервоприводе
ServoDifference = distance_sm; // Находим расстояние от точки начала координат до объекта
//Serial.print("ServoDifference "); //
//Serial.print(ServoDifference); // Выводим расстояние от точки начала координат до объекта
//Serial.print("  ");
Serial.print(180); // Выводим на серий порт угол поворота сервопривода
Serial.println(" RealAngle."); //
myservo.write(180);   // Задаем сервоприводу рассчитанный угол поворота
LcdDistanse (33);  // Выводим на LCD экран угол поворота сервопривода
}

//LCD print distance - функция печатает на LCD экране значение расстояния до объекта
void LcdDistanse (unsigned int Dist) {
lcd.setCursor(0, 1);
lcd.print("          ");
lcd.setCursor(0, 1);
lcd.print("Dist:");
lcd.print(Dist);
lcd.setCursor(8, 1);
lcd.print("cm");
lcd.write((uint8_t)0); // выводим пайп
}
```

```
//Check information from DHT11 sensor - проверка информации с термо-
датчика DHT11
void checkDHT()  {
  // Мониторинг ошибок
  dhtRes = DHT.read(DHT11_PIN);    // Чтение данных с сенсора DHT11
        switch (dhtRes){
        case DHTLIB_OK: // Если всё хорошо то вывести данные
                LcdDhtTemp(DHT.temperature);
                LcdDhtRH(DHT.humidity);
                Serial.print(DHT.temperature);
                Serial.print(" C,");
                Serial.print(DHT.humidity);
                Serial.print(" RH,");
        break;
        case DHTLIB_ERROR_CHECKSUM:
                Serial.println("Checksum error, \t");
    break;
    case DHTLIB_ERROR_TIMEOUT:
                Serial.println("Time out error, \t");
        break;
        default:
    Serial.println("Unknown error, \t");
        break;
  }
}

//LCD print temperature humidity - ПП вывода температуры на LCD экран
void LcdDhtTemp(unsigned int Temp) {
    lcd.setCursor(0, 3);
    lcd.print("          ");
    lcd.setCursor(0, 3);
    lcd.print("Temp  ");
    lcd.print(Temp);  // DHT.temperature
    lcd.setCursor(9, 3);
    lcd.write(byte(2));
        lcd.write(byte(0));
}

//LCD print  humidity  - ПП вывода влажности на LCD экран
void LcdDhtRH(unsigned int RH) {
```

```
   lcd.setCursor(0, 2);
   lcd.print("          ");
   lcd.setCursor(0, 2);
   lcd.print("RH    ");
   lcd.print(RH); // DHT.humidity
   lcd.setCursor(9, 2);
   lcd.print("%");
        lcd.write((uint8_t)0); // выводим пайп
}

//Check information from Volume sensor - считывание информации с датчика уровня звука
void checkVolume(void) {
volumeRes = analogRead(volumnePin);//map(analogRead(volumnePin), 0, 1023, 0, 100); // Считываем данные с датчика звука и масштабируем в диапазоне от 0 до 100
//Debug mode
Serial.print(volumeRes);
Serial.print(" noise,");
LcdVolume(volumeRes);

//if(volumeRes>=volumeThreshold){
//    LcdVolumeHigh(); //Turn ON Led
//  }
//  else{
//    LcdVolumeLow(); // Turn OFF Led
//  }
}

//LCD print distance
void LcdVolume(unsigned int Volume) {
lcd.setCursor(0, 0);
lcd.print("          ");
lcd.setCursor(0, 0);
lcd.print("Vol:");
lcd.print(Volume);
lcd.setCursor(10, 0);
lcd.write(byte(0));
}

//LCD Vol high "!!!"
```

```
//void LcdVolumeHigh(void) {
//lcd.setCursor(7,0);
//lcd.print("!!!");
//}

//LCD Vol low "   "
//void LcdVolumeLow(void) {
//lcd.setCursor(7,0);
//lcd.print("   ");
//}

//Check PIR sensor
void checkPIR(void) {
   pirRes = digitalRead(pirPin); //read state of the PIR
   if (pirRes == LOW) {
     Serial.println("No motion."); //if the value read is low, there was no motion
     lcd.setCursor(11, 0);
     lcd.print("         ");
     lcd.setCursor(11, 0);
     lcd.print("No motion");
   }
   else {
     Serial.println("Motion."); //if the value read was high, there was motion
     lcd.setCursor(11, 0);
     lcd.print("         ");
     lcd.setCursor(11, 0);
     lcd.print("Motion!");
   }
}

//GasSensor
int GasSensor(void) {
CO2_data = map(analogRead(MQ2_pin), 0, 1023, 0, 100);  // Считываем
значение с датчика CO2, и масштабируем по шкале от 0 до 100
Serial.print(CO2_data);  // Выводим нормированные данные в серийный
интерфейс
Serial.print(" CO2");
Serial.print(",");
LcdMQ2(CO2_data); // Выводим нормированные данные на LCD экран
}
```

```
//LCD print MQ2 data
void LcdMQ2 (unsigned int GasLvl) {
lcd.setCursor(11, 1);
lcd.print("        ");
lcd.setCursor(11, 1);
lcd.print("CO2: ");
lcd.print(GasLvl);
}

// QueryFromLaserSensor
void QueryFromLaserSensor(void){
LaserSensor1 = map(analogRead(LaserSensor1_pin), 0, 1023, 0, 100) ;
//Считываем значения с лазеров, и масштабируем по шкале от 0 до 100
LaserSensor2 = map(analogRead(LaserSensor2_pin), 0, 1023, 0, 100) ;
//Считываем значения с лазеров, и масштабируем по шкале от 0 до 100
LaserSensor3 = map(analogRead(LaserSensor3_pin), 0, 1023, 0, 100) ;
//Считываем значения с лазеров, и масштабируем по шкале от 0 до 100

// Выводим информацию в серийный интерфейс
Serial.print(LaserSensor1);
Serial.print(" LaserSensor1,");
Serial.print(LaserSensor2);
Serial.print(" LaserSensor2,");
Serial.print(LaserSensor3);
Serial.print(" LaserSensor3,");
//Выводим на LCD экран данных первого лазера
lcd.setCursor(11, 2);
lcd.print("     ");
lcd.setCursor(11, 2);
lcd.print("L1=");
if (LaserSensor1 <= LaserThreshold)
{ lcd.write(byte(3)); }
else {lcd.write(byte(4)); }
lcd.write(byte(0));

//Выводим на LCD экран данных второго лазера
lcd.setCursor(16, 2);
lcd.print("    ");
lcd.setCursor(16, 2);
lcd.print("L2=");
if (LaserSensor2 <= LaserThreshold)
```

```
{ lcd.write(byte(3)); }
else {lcd.write(byte(4)); }

//Выводим на LCD экран данных третьего лазера
lcd.setCursor(11, 3);
lcd.print("     ");
lcd.setCursor(11, 3);
lcd.print("L3=");
if (LaserSensor3 <= LaserThreshold)
{ lcd.write(byte(3)); }
else {lcd.write(byte(4)); }
lcd.write(byte(0));
}

//GlobalCheck
void GlobalCheck (void) {
checkVolume();
checkDHT();
GasSensor();
checkPIR();
}

//Don't worry be happy. Smile =)
void Smileee (void) {
lcd.setCursor(17, 3);
lcd.print("    ");
lcd.setCursor(16, 3);
lcd.write(byte(1));
lcd.write(byte(1));
lcd.write(byte(1));
lcd.write(byte(1));
}
```

Приложение В
Код графического интерфейса

```
import java.util.Properties;
import java.io.*;
import java.util.*;
import javax.mail.*;
import javax.mail.Flags.Flag;
import javax.mail.internet.*;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;


// A function to check a mail account - IMAP
void checkMail() {
  try {

    IMAPFolder folder = null;
    Store store = null;
    String subject = null;
    Flag flag = null;
    try
    {
      Properties props = System.getProperties();
      props.setProperty("mail.store.protocol", "imaps");
      props.put("mail.imaps.host", "imap.gmail.com");

      // Create authentication object
      Auth auth = new Auth();

      // Make a session
      Session session = Session.getDefaultInstance(props, auth);
      store = session.getStore("imaps");

      store.connect();

      folder = (IMAPFolder) store.getFolder("Inbox");
```

```
  if (!folder.isOpen())
      folder.open(Folder.READ_WRITE);

    Message[] messages = folder.getMessages();
    System.out.println("No of Messages : " + folder.getMessageCount());
    System.out.println("No of Unread Messages : " +
folder.getUnreadMessageCount());
    System.out.println(messages.length);

    for (int i=0; i < 10; i++)
     //for (int i=0; i < messages.length;i++)
    {


System.out.println("**********************************************************
**************************");
      System.out.println("MESSAGE " + (i + 1) + ":");
      Message msg =  messages[i];
      //System.out.println(msg.getMessageNumber());
      //Object String;
      //System.out.println(folder.getUID(msg)

      subject = msg.getSubject();

      System.out.println("Subject: " + subject);
      System.out.println("From: " + msg.getFrom()[0]);
      System.out.println("To: "+msg.getAllRecipients()[0]);
      System.out.println("Date: "+msg.getReceivedDate());
      System.out.println("Size: "+msg.getSize());
      System.out.println(msg.getFlags());
      System.out.println("Body: \n"+ msg.getContent());
      System.out.println(msg.getContentType());
    }
  }
  finally
  {
    if (folder != null && folder.isOpen()) {
      folder.close(true);
    }
    if (store != null) {
      store.close();
```

```
      }
    }
  }


  // This error handling isn't very good
  catch (Exception e) {
    System.out.println("Failed to connect to the store");
    e.printStackTrace();
  }
}


// Function to send email - IMAP

void sendMail( String MailToBeSend )

{
  try {

    Properties props = new Properties();

    props.put("mail.smtp.host", "smtp.gmail.com");
    props.put("mail.smtp.socketFactory.port", "465");
    props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
    props.put("mail.smtp.auth", "true");
    props.put("mail.smtp.port", "465");

    // Create authentication object
    Auth auth = new Auth();

    Session session = Session.getDefaultInstance(props, auth);

    try {

      Message message = new MimeMessage(session);
      message.setFrom(new InternetAddress("n00304844@gmail.com"));
      message.setRecipients(Message.RecipientType.TO,
InternetAddress.parse("1tempfile1@gmail.com"));
      message.setSubject("This message send by Your security system");
      message.setText(MailToBeSend);
```

```
      Transport.send(message);

      System.out.println("Sending done");
    }

    finally
    {
      //session.close();
    }
  }
  catch (MessagingException e)
  {
    throw new RuntimeException(e);
  }
  System.out.println("Session closed");
}


import javax.mail.Authenticator;
import javax.mail.PasswordAuthentication;

public class Auth extends Authenticator {

  public Auth() {
    super();
  }

  public PasswordAuthentication getPasswordAuthentication() {
    String username, password;
    username = "n00304844@gmail.com";
    password = "AZCVqpda1!";
    System.out.println("authenticating. . ");
    return new PasswordAuthentication(username, password);
  }
}


import java.util.Properties;
import java.io.*;
import java.util.*;
import javax.mail.*;
```

```
import javax.mail.Flags.Flag;
import javax.mail.internet.*;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;


// A function to check a mail account - IMAP
void checkMail() {
  try {

    IMAPFolder folder = null;
    Store store = null;
    String subject = null;
    Flag flag = null;
    try
    {
      Properties props = System.getProperties();
      props.setProperty("mail.store.protocol", "imaps");
      props.put("mail.imaps.host", "imap.gmail.com");

      // Create authentication object
      Auth auth = new Auth();

      // Make a session
      Session session = Session.getDefaultInstance(props, auth);
      store = session.getStore("imaps");

      store.connect();

      folder = (IMAPFolder) store.getFolder("Inbox");

      if (!folder.isOpen())
        folder.open(Folder.READ_WRITE);

      Message[] messages = folder.getMessages();
      System.out.println("No of Messages : " + folder.getMessageCount());
```

```
System.out.println("No of Unread Messages : " +
folder.getUnreadMessageCount());
    System.out.println(messages.length);

    for (int i=0; i < 10; i++)
     //for (int i=0; i < messages.length;i++)
    {


System.out.println("**************************************************************
**************************");
      System.out.println("MESSAGE " + (i + 1) + ":");
      Message msg =  messages[i];
      //System.out.println(msg.getMessageNumber());
      //Object String;
      //System.out.println(folder.getUID(msg)

      subject = msg.getSubject();

      System.out.println("Subject: " + subject);
      System.out.println("From: " + msg.getFrom()[0]);
      System.out.println("To: "+msg.getAllRecipients()[0]);
      System.out.println("Date: "+msg.getReceivedDate());
      System.out.println("Size: "+msg.getSize());
      System.out.println(msg.getFlags());
      System.out.println("Body: \n"+ msg.getContent());
      System.out.println(msg.getContentType());
    }
   }
   finally
   {
    if (folder != null && folder.isOpen()) {
     folder.close(true);
    }
    if (store != null) {
     store.close();
    }
   }
  }
```

```
  // This error handling isn't very good
  catch (Exception e) {
    System.out.println("Failed to connect to the store");
    e.printStackTrace();
  }
}


// Function to send email - IMAP

void sendMail( String MailToBeSend )

{
  try {

    Properties props = new Properties();

    props.put("mail.smtp.host", "smtp.gmail.com");
    props.put("mail.smtp.socketFactory.port", "465");
    props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
    props.put("mail.smtp.auth", "true");
    props.put("mail.smtp.port", "465");

    // Create authentication object
    Auth auth = new Auth();

    Session session = Session.getDefaultInstance(props, auth);

    try {

      Message message = new MimeMessage(session);
      message.setFrom(new InternetAddress("n00304844@gmail.com"));
      message.setRecipients(Message.RecipientType.TO,
InternetAddress.parse("1tempfile1@gmail.com"));
      message.setSubject("This message send by Your security system");
      message.setText(MailToBeSend);

      Transport.send(message);

      System.out.println("Sending done");
    }
```

```
 finally
    {
      //session.close();
    }
  }
  catch (MessagingException e)
  {
    throw new RuntimeException(e);
  }
  System.out.println("Session closed");
}

//Xpos - X position of Rouond bar in pix
//Ypos - Y position of Rouond bar in pix
//MinValue -  0 degree position
//MaxValue - 180 degree position
//ScaleL - lenth of bar
//ScaleH - height of bar
//SensorValue - sensor value to be drowen

void Round_bar(int Xpos, int Ypos, int MinValue, int MaxValue, int ScaleL, int
ScaleH, float SensorValue, float SensorValueMax) {
  pushMatrix();
  translate(Xpos, Ypos);
  arc(0, 0, ScaleL, ScaleH, PI, 2*PI, CHORD); // (center X , center Y, width ,height
, start, stop)
  // Kruglay shkala so strelkoi
  float SensorValue1Normalized = 180 / SensorValueMax * SensorValue;
  rotate(radians(SensorValue1Normalized-90));
  line(0, 0, 0, -ScaleH/2);
  line(0, -ScaleH/2, (-ScaleH/2)*0.1, (-ScaleH/2)*0.9);
  line(0, -ScaleH/2, (ScaleH/2)*0.1, (-ScaleH/2)*0.9);
  popMatrix();
}



void Sql( String curTabl,String Col1,String Col2,String Col3, String Col4, String
Col5) {
```

```
 if ( msql.connect() )
  {
    //println("|" + Col3 + "|" );
    msql.query( "INSERT INTO " + curTabl + "  VALUES(NULL, " + Col1 + ", "+
Col2 + ", "+ Col3 + ", "+ Col4 + ", "+ char(34)+ Col5 + char(34)+", NULL);");
    //println("INSERT INTO " + curTabl + "  VALUES(NULL, " + Col1 + ", "+
Col2 + ", "+ Col3 + ", "+ Col4 + ", "+ Col5 +", NULL);");
    }
```