МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГАОУ ВО «Крымский федеральный университет имени В. И. Вернадского»

Физико-технический институт

Кафедра компьютерной инженерии и моделирования

Отчет по основам проектной деятельности

Студентов 1 курса группы ИВТ-192(1)

Москаленко Алексей Константинович

Цыганков Андрей Алексеевич

Абибулаев Ремзи Энверович

**РАЗРАБОТКА СИСТЕМЫ ДИСТАНЦИОННОГО КОНТРОЛЯ И УПРАВЛЕНИЯ КЛИМАТИЧЕСКИМИ СИСТЕМАМИ, ОБЕСПЕЧИВАЮЩИМИ ТРЕБУЕМЫЕ УСЛОВИЯ ДЛЯ РАБОТЫ УСТРОЙСТВ ЭЛЕКТРОСВЯЗИ**

Направление подготовки: 09.03.01«Информатика и компьютерная техника»

Научный руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Корниенко А. Ю.

К ЗАЩИТЕ ДОПУСКАЮ:

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Милюков В. В.

Симферополь 2019

Реферат

Темой нашего проекта является: разработка системы дистанционного контроля и управления климатическими системами, обеспечивающими требуемые условия для работы устройств электросвязи.

Цель: разработать системы дистанционного контроля и управления климатическими системами, обеспечивающими требуемые условия для работы устройств электросвязи.

Объект исследования – управление системой кондиционирования.

Методы исследования. При исследовании данной темы использовались аналитические, теоретические и практические методы.

Использованные оборудования и аппаратура: персональные компьютеры, ноутбуки. NodeMcu V3 Amica, инфракрасный светодиод, датчик температуры, макетная плата, резисторы, соединительные провода.

Список сокращений и условных обозначений:

СУБД – структура управления базы данных.

ИК – инфракрасный.

ДУ – дистанционное управление.

ТЗ – техническое задание.

БД – база данных.

ПО – программное обеспечение.

Оглавление

1. Введение.
2. Основная часть.
   1. Теоретическая часть;
      1. Мульти-сплит система кондиционера.
      2. Система дистанционного управления.
      3. Способы передачи инфракрасного излучения.
   2. Вывод из теоретической части. Методы реализации.
   3. Практическая часть.
      1. Создание сервера.
      2. Создание терминала управления.
      3. Работа с платами для дистанционного управления климатическими системами.
3. Заключение.

Введение

Актуальность проектной работы дистанционного управления кондиционирования помещения через интернет связана с возможностью управления всеми имеющимися мульти-сплит системами кондиционирования. Такому дистанционному управлению можно подключить любой прибор имеющий ИК приёмник.

Цели и задачи работы:

Методы исследования. При исследовании данной темы использовались аналитические и практические методы.

Цель проектной работы: изучить какие системы контроля вообще бывают в промышленных, лабораторных масштабах. Далее выбрать одну из систем, понять какими методами в выбранной системе передаётся информация от управления до кондиционера, подробно изучить настройку прибора по приёму определённой команды через интернет и отправки её же путем инфракрасного излучения на систему кондиционирования, а также создание прибора для наглядного пособия работы управления кондиционерами в настоящее время через интернет.

Предметом исследования является анализ использования и настройка оборудования.

Теоретическая часть:

2.1.1. Мульти-сплит-система представляет собой обычную сплит-систему, которая разделена на две части (испаритель и конденсатор) и позволяет охладить или отопить несколько помещений с одним внешним блоком. В внешнем блоке этого кондиционера имеется более мощный компрессор, порты для подключения нескольких трасс и автоматика с запорными клапанами для регулирования объема хладагента, подаваемого на внутренние блоки, расположенные в помещении. Большая мульти сплит-система называется системой с переменным расходом хладагента и может использоваться вместо центральной системы кондиционирования воздуха, так как она позволяет повысить энергоэффективность, но ее дороже приобрести и установить.

Разница между сплит-системой и мульти-сплит-системой в нескольких внутренних блоках. Все они подключены к основному внешнему блоку, но принцип их работы аналогичен простой сплит-системе.

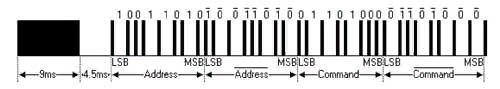
Ее уникальной особенностью является наличие одного основного внешнего блока, который подключен к нескольким внутренним блокам. Такие системы могут быть правильным решением для поддержания микроклимата в нескольких офисах, магазинах, больших жилых помещениях. Лишь немногие внешние блоки не ухудшают эстетический облик здания. Основной внешний блок может быть подключен к нескольким различным типам помещений: пол, потолок и так далее.

В рамках данного проекта мы выбираем мульти-сплит систему.

2.1.2. Дистанционное управление — это один из способов сообщения сигналов с управляемым устройством (в отличие от проводного, механического или при помощи оператора). Дистанционное управление чаще всего осуществляется методами инфракрасной, ультразвуковой, Bluetooth, GPS или Wi-Fi передачи данных. Чаще используемый метод дистанционного управления кондиционером является пульт ДУ для передачи инфракрасного сигнала. Дистанционное управление применяется там, где невозможно или сложно применить другой подход — например, беспилотники, космические аппараты, ракеты, квадрокоптеры. В случае климатических систем ДУ также используется для удобства.

2.1.3. Разберемся как вообще работает прием сигналов ДУ. Общий алгоритм работы аппаратуры, настроенной на прием сигналов определенного пульта инфракрасного дистанционного управления таков:  
1. Ожидание начала посылки.  
2. Прием преамбулы (измерение длительности первого импульса и паузы за ним). Преамбула является передовой частью пакета (кадра). Поле преамбулы занимает 8 байт и состоит из 1-байтной последовательности 10101011, которая указывает на начало кадра (пакета), и семи байтов, в которых чередуются 1 и 0.  
3. Определение "наша" преамбула или чужая. Если чужая - переход к началу.  
4. Прием информационных бит согласно "нашему" протоколу.  
5. Валидация принятых бит и преобразование принятых бит в более короткую форму, выделение адреса/команды, служебных бит, получение окончательного кода посылки.  
6. Сравнение окончательного кода посылки с кодами команд, при соответствии - установки кода команды.

Рассмотрим пункт 5 - преобразование в более короткую форму. Как описано выше - разные протоколы имеют разное количество информационных бит. Одни используют 12, другие 32, а Панасоник вообще 48. Некоторая информация передается дважды. Но нам интересно знать только адрес и команду. Если устройство рассчитано на работу с одним конкретным протоколом, то длину окончательного кода можно выбрать совпадающим с длиной адреса и команды. Если ж протокол заранее неизвестен - нужно выбирать максимально возможную. Некоторые протокол имеют длину не более чем word (два байта), другие укладываются в long (два word или четыре байта), а Panasonic (JAP) передает аж три слова (шесть байтов). Но много битные протоколы как правило, содержат избыточную информацию, которую можно использовать для валидации. Т.е. проверки правильности приема посылки. Например, в протоколе NEC:



Вначале идёт импульс АРУ (AGC) - Автоматическая регулировка усиления, он использовался в старых ИК приёмниках для настройки усиления. После него идёт пауза, по длительности начального импульса и паузы можно определить тип используемого протокола. После этого посылается адрес начиная с младшего бита. Здесь используется мультиплексор передачи данных, больше пауза = 1. Затем для надёжности передается инвертированный адрес. В какой-то момент оказалось, что 256 адресов недостаточно и решили отказаться от повторной передачи адреса, но взамен увеличить длину адреса до 2 байт. Адрес не зависит от нажатой кнопки, он всегда одинаков, он отличается от устройства к устройству (телевизор не реагирует на пульт от муз. центра даже если они одной фирмы). В конце концов передаётся команда и её инверсия, которые определяются нажатой на пульте кнопкой. Эта информация передаётся всего 1 раз. Если продолжать держать кнопку нажатой, то идёт передача кода повтора: Код повтора состоит из начального АРУ импульса с последующим коротким импульсом.

2.2. Выводы из теоретических сведений, методы реализации: Одной из поставленных задач проекта является бесперебойное, а главное быстрое соединение и передача информации от среды управление до конечного устройства - кондиционера.

При выборе схемы управление необходимо учитывать, что удаленность конечного устройства не определена каким-то конкретно установленным максимальным расстоянием.

Следовательно физическое местоположение элемента кондиционирования может быть значительно удалено от управляющего элемента.

Прежде всего именно этот фактор закрывает возможность подключения и управление кондиционером в непосредственной близости при помощи Bluetooth и Wi-Fi соединения.

Следующим немаловажным параметром является то, что такой способ управления на практике крайне неудобен из-за своей концепции.

Это означает, что необходимым условием является использование какой-либо локальной или глобальной сети, при помощи которой появляется возможность контролировать каждое устройство по отдельности.

На данный момент существует большое множество способов управления оборудованием, обеспечивающим контроль климата в помещении.

Современные системы оборудованы центральными пультами, ДУ пультами, Wi-Fi контроллерами.

Исходя из ТЗ использование центрального пульта невозможно без низкоуровневого вмешательства в его программную архитектуру и строение в целом, что является дорогостоящей операцией и не оправдывает усилий.

Использование Wi-Fi контроллера также недопустимо из-за вышеперечисленных выводов, а также в силу того, что устаревшие модели аппаратно не имеют данных возможностей.

Одним из немногих оставшихся вариантов остается ДУ пульт с инфракрасным излучателем.

Однако данный метод по-прежнему не выполняет условия ТЗ.

Для решения данной проблемы в проекте была выбрана следующая схема. Управление всеми элементами климатических систем осуществляется при помощи ИК-сигналов от микроконтроллера.

На момент выполнения данного проекта одноплатные контроллеры получили широкое распространение на рынке радиоэлектроники.

Вследствие распространенности на рынке стоимость используемых продуктов достаточно низкая, а доступность, наоборот, крайне высока.

Именно доступность обеспечивает ремонтопригодность всей системы.

Подключение каждого контроллера к единому центру управлению.

Создание сервера обеспечивает незатруднительную масштабируемость проекта.

Незатруднительное расширение количества микроконтроллеров на единицу потребителя является несомненным преимуществом такого решения.

Сервер также позволяет обеспечить безопасное, а, главное, удобное подключение к общей системе, разделенный доступ и легкое усовершенствование элементов общей системы.

Самое важное преимущество сервера - возможность подключения из любой точки, входящей в окружение сети сервера.

Подключение к серверу при помощи системы терминалов.

Подключение при помощи терминала позволяет каждому пользователю одновременно работать со всей системой, наблюдать динамическое отображение установленных параметров.

Терминалы, несомненно, должны быть кроссплатформенным (удобным для пользователя) решением. В идеале - работать с любого устройства, находящегося под рукой в данный момент времени.

2.3. Практическая часть:

2.3.1. Сервер

В качестве языка программирования для сервера был выбран C++.

Среда разработкии - Microsoft Visual Studio. Это мощная IDE с широкими возможностями использования, применяемыми не раз в ходе разработки.

Решение основано на несомненной скорости работы данного языка программирования из-за его низкоуровневой архитектуры.

C++ нередко используется для создание технически тяжелых программ в том числе и серверов. C++ является компилируемым языком, что означает отсутствие необходимости "таскать" за собой "хвост" в виде интерпретатора.

Огромное количество библиотек написано под язык C++ и достаточно крупные из существующих были применены в ходе разработки.

Для обмена информацией между узлами связи использовался текстовый формат данных JSON, используемая библиотека - JSON For Modern C++ от разработчика Nlohmann.

Для некоторой информации использована библиотека SQLite3 от официального разработчика данной структуры управления базы данных.

Для обмена данными библиотека Windows Sockets 2 выбранная на основании присутствия подробной документации.

Ипользуемые библиотеки:

* Nlohmann JSON For Modern C++.
* SQLite3

Общая схема подключения.

Создается общий сокет для подключения как терминалов, так и контроллеров. В отдельном потоке бесперебойно прослушиваются и принимаются все входящие подключения.

И далее обрабатываются в зависимости от того, какой именно клиент пытается подключиться к серверу.

Каждый микроконтроллер в сети (локальной или интернете) подключается к серверу, отправляя handshake-запрос, содержащий некоторую информацию о себе и ключ доступа (access\_key в СУБД).

Сервер обрабатывает запрос, проверяя на подлинность данные, присланные контроллером в целях защиты от несанкционированного доступа.

Если контролер проходит этап проверки, то его сокет отправляется в другой поток, что бесконечно обрабатывает список (внутреннюю БД) всех подключенных микроконтроллеров.

При необходимости, контролеру отправляется запрос и при получении ответа записывается в специальную переменную.

Во время простоя контролер отправляет пакеты с одинаковым содержимым, называемые alive-пакеты.

Ожидает в течение нескольких секунд, настраиваемых внутри программы и в случае, если не получает ответ, отключает клиента.

Каждый терминал, устанавливающий соединение с сервером, проходит точно такой же этап верификации и далее выполняется запрос, присланный терминалом.

После обрабатывается запрос терминала и отправляется пакет данных, содержащий ответ на данный тип запроса.

Еще одной важной частью является приложение экстренных оповещений.

В случае сбоев в работе микроконтроллер присылает код ошибки при опрашивании данных с него. Терминалы не отображают те контролеры, что присылают код ошибки.

В ходе работы контролером могут присылаться оповещения уровня: тревоги, предупреждения, замечаний.

Разделение этих двух функций в отдельные приложения позволяет следить также за состоянием сервера.

При возникновении ошибки в работе сервера и неожиданным его отключением (падением) приложение оповещений делает рассылку с соответствующим содержимым.

Данное ПО было написано на языке Python версии 3.8.

В качестве IDE использовался Visual Code с расширение Python от Microsoft.

Выбор языка обусловлен скоростью написания кода, низким требованиям к скорости работы и низкой загруженностью программы.

Приложение в свою очередь является второстепенным, вспомогательным, сервер вполне может функционировать без него.

Доставка сообщений производится пользователям зарегистрированным в социальной сети ВКонтакте.

Данный метод обусловлен тем, что на момент создания проекта большинство людей, проживающих в странах СНГ имеют учетную запись ВКонтакте.

Также данный сервис предоставляет доступ к своему функционалу через VKApi, позволяющим полноценно работать с каждым элементом сервиса.

Инструкция такова.

Создается закрытая группа, в которую допускаются лишь системные администраторы/пользователи, которым необходимо получать оповещения от микроконтроллера.

Регистрируется токен для взаимодействия с приложением оповещений.

Все входные данные записываются в конфигурационный файл, находящийся в одной директории с исполняемым файлом приложения.

Выбирается режим работы.

Обычный:

Рассылка оповещений проводится по участникам группы. Ограничение сообщений проводится кнопкой "Запретить сообщения" в диалоге с сообществом.

Расширенный:

Рассылка оповещений проводится через сайт, на котором лежат исходные файлы для работы с СУБД, хранящей пользователей для рассылки, а также допуск для рассылки оповещений.

И файл для callback'а вконтакте, позволяющий управлять настройками присылаемых оповещений и для ответа на сообщения от группы.

Требования:

* Python 3.8
* Учетная запись ВКонтакте
* Сообщество ВКонтакте
* Сервер с постоянным ip/доменом использующим PHP-скрипты.(В зависимости от режима работы)

2.3.2. Терминал.

Современные стандарты и функции веб-программирования позволяют создавать небольшие приложения прямо в браузере.

Также неотъемлемым и очевидным преимуществом использования веб-подхода является кроссплатформенность.

Многие современные браузеры используют движок chromium и поддерживают javascript стандарта 2015 года.

Именно поэтому была создана одна html-страница с элементами управления, которые строятся динамически в соответствии с запросом на сервер.

Работа терминала имеет вид

1) Один "GETDB" запрос на сервер, содержащий конфигурационную информацию о всех устройствах, подключенных к серверу, соответствующих ключу доступа.

2) Динамическое построение страницы в соответствии с ответом от сервера.

3) Последующие "GET" (название самого запроса, а не заголовка http) запросы на сервер, в которых содержится текущее состояние кондиционеров. Частота зависит от настроек в конфигурационном файле.

4) Сравнение с текущей страницей, обновление состояний на странице.

5) Запросы "PUT" по нажатию на кнопку на сайте.

Терминал работает средствами HTTP запросов на сервер, которые содержат JSON-данные.

Требования:

* Браузер, поддерживающий JavaScript 2015+
* Подключение к локальной сети с сервером / доступ в интернет

2.3.3. Микроконтроллер.

За основу был выбран контролер NodeMcu V3 Amica с управляющим чипом Esp8266 со встроенным Wi-Fi модулем(esp12).

К нему подключен температурный датчик DS18B20 для снятия реальных показания в помещении и инфракрасный светодиод, отправляющий сигналы на кондиционер.

Стоит учитывать, что плату можно закрепить в любом месте помещения, где есть физический доступ для отправки.

Контролер также хранит необходимые параметры и конфигурацию кондиционера.

Контролер беспрерывно прослушивает сокет с сервером, ожидает запроса и отправляет ответ.

В случае сбоев или оповещений контролер отправляет экстренное сообщение серверу с соответствующим содержимым.

Используемые библиотеки:

* ESP8266 WiFi
* ArduinoJson
* IRremoteESP8266
* OneWire
* DallasTemperature

Требования:

* Wi-Fi точка доступа.
* Постоянный источник питания 5-9V.
* Прямая видимость кондиционера.

1. Заключение.

В процессе выполнения работы было много трудностей, нам присылали бракованную модель NodeMCU, перегорела плата Arduino UNO. Была проделана большая работа по написанию серверной части, сайта и скетча. На данный момент мы получили работающую модель, которую можно при желании использовать в сфере бизнеса. Главные задачи проекта были сделаны, из дополнительных атрибутов добавлена рассылка пользователям о критической ошибке модуля, если, к примеру он перестаёт работать. Главная цель была достигнута – проектная деятельность сплотила нас как команду невзирая на все неудачи, мы достигли конечной точки этого предложения, но в ближайшем будущем мы опять будем начинать с нового предложения.