МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**Учреждение образования**

**«Гомельский государственный университет им. Франциска Скорины»**

# Факультет физики и информационных технологий

Кафедра общей физики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

**Разработки ботанического лабораторного комплекса для имитации климатических условий**

ГГУ 1-39 03 02КП

Исполнитель:

Студент группы МС-32: \_\_\_\_\_\_\_\_\_\_\_\_\_ Лапунов Г.А.

Научный руководитель:

старший преподаватель

кафедры общей физики \_\_\_\_\_\_\_\_\_\_\_\_\_ М.А. Подалов

Гомель 2022

**Содержание**

Введение3

1 Анализ предметной области ……...………………………………………..5

1.1 Постановка задачи……………………………………………………...…5

1.2 Общая информация……………………………………………………….5

1.3 Инструменты разработки……………………………………………...…6

1.3.1 Язык программирования C++……………………………………….…6

1.3.2 Среда разработки VisualStudioCode ………..…………………...…….8

1.3.3 Язык программирования Lua ..………………………………...…..…..9

1.3.4 Технология интернет вещей..…………………...................................10

2 Логическая и физическая структуры данных............................................12

2.1 Логическая и концептуальная структура данных..................................12

3 Разработка автоматизации системы на базе сервера Linux......................15

3.1 Постановка задачи15

3.2 Взаимодействия сервера с установкой 16

Заключение22

Список используемой литературы23

**Введение**

В современном информационном обществе с каждым днем стремительно увеличивается уровень развития информационных и коммуникационных технологий. За последние годы их интенсивное использование и глобальное распространение, а также неограниченный доступ общества к интернету привели к многократному увеличению информации объема информации. В связи с этим возникает необходимость представления информации в компактном, простом в использовании, комфортном и визуально приятном для пользователя виде. Это поможет пользователю среди огромного объема сведений быстро и удобно найти необходимую информацию, затратив при этом минимум времени и усилий.

Жизнь современного человека уже нельзя представить без различных устройств, таких как мобильный телефон или планшетный компьютер, они стали неотъемлемым атрибутом каждого. Интернет вещей становится глобальной тенденцией. Несмотря на то, что число подключенных к глобальной Сети устройств велико (на одного человека приходится более одного устройства), это лишь 0,06% всех вещей (в самом широком смысле), которые могут потенциально быть подключены к Интернету. По оценкам аналитиков CISCO, мировой рынок Интернета вещей составит от 8,9 до 20 трлн долларов. Кроме того, система присвоения уникального IP-адреса видоизменилась до того, что теперь его можно присвоить бесконечно великому множеству вещей. Интернет вещей становится не только средством оптимизации производства в индустриальном секторе, но и маркетинговым, а в частности, коммуникативным инструментом в секторе В2С (business-to-consumer). Активно развивается (и можно даже судить о его пресыщении) рынок так называемых wearables, т.е. «носимых» вещей — фитнес-трекеров различных ценовых категорий (Apple Watch, Jawbone UP, Nike Fuel Band). На данный момент подавляющее большинство исследований и научно-популярных статей рассматривает интернет вещей с коммуникационной точки зрения, т.е. с точки зрения его технических и технологических характеристик. Однако концепция не изучается как коммуникативный феномен. Это может объясняться значительным весом именно технических и технологических параметров, участвующих в процессе коммуникации.

Подобная проблема возникает и при попытках охарактеризовать Интернет: его многомерность и постоянная изменчивость не позволяют дать четкое определение как с коммуникационной, так и с коммуникативной позиций.

В данный момент технологии, претерпевшие в последнее время значительный скачок в развитии (во многом благодаря спонсированию профильных отраслей), детерминируют. Формируются новые технические и технологические феномены, которые существенно трансформируют социальную коммуникацию. Один из таких новых коммуникационных и коммуникативных феноменов — так называемый Интернет вещей. Концепция Интернета вещей, то есть коммуникационного пространства, возникла с появлением объектов, подключенных к всемирной Сети, что в свою очередь было обусловлено развитием радиочастотной идентификации (RFID) и беспроводных сенсорных сетей (БСС).

Особенность Интернета вещей заключается в его относительной новизне (термин появился в 1999 году, но датой фактического зарождения концепции считают 2010 год, когда количество подключенных к Всемирной сети устройств превысило численность населения Земли) и одновременно активному развитию (особенно в индустриальном секторе). В рамках феномена формируется принципиально новая, с точки зрения коммуникаций, модель — machine-to-machine (M2M), фактически исключающая человека из процесса коммуникации. В связи с этим, возникает необходимость пересмотреть модели интеракции в Интернет-среде.

**1 Анализ предметной области**

**1.1 Постановка задачи**

Основной целью данной работы является разработка комплекса для имитации климатических условий. Для осуществления цели служат следующие задачи:

* выполнить анализ работы установки.
* выбрать язык программирования и среду разработки.
* выбрать набор датчиков для использования в установке.
* разработать сервер для обеспечения взаимодействия с датчиками удаленно.
* выполнить теоретический анализ научной и методической литературы.

**1.2 Общая информация**

История IoT началась в 1999 году, когда британский технолог Кевин Эштон внес предложение по оптимизации логистики компании Procter & Gamble посредством радиочастотных меток. Для изучения данной концепции при Массачусетском технологическом институте был создан аналитический центр Auto-ID Center, который направил свои усилия на исследования в области сетевой радиочастотной идентификации. [1]

Временем рождения IoT называют период с 2008 по 2009 год, поскольку, по оценкам исследователей, именно в это время количество приборов, подключенных к Всемирной Паутине, превысило численность населения нашей планеты. Столь стремительное развитие технологии стало возможным благодаря массовому распространению смартфонов и беспроводных сетей, а также снижению стоимости электроники и систем обработки информации.

К IoT можно отнести не одну, а целый спектр технологий, призванных соединять Интернет и технические устройства. Сюда можно отнести интеллектуальные датчики, протоколы их взаимодействия, различные типы беспроводной связи, включая Bluetooth, RFID, BLE и др.

Область применения технологии охватывает практически все сферы человеческой жизни. Ее используют на производстве, в логистике, здравоохранении, образовании и даже в домашних условиях. [1]

Умный офис:

Система «умный офис» – это комплекс оборудования и программных решений, позволяющих контролировать перемещение персонала компании и обеспечивать взаимодействие между сотрудниками. Наиболее распространенными примерами внедрения систем навигации в офисе являются:

* внутренняя навигация с помощью приложения в смартфоне;
* трекинг персонала в режиме реального времени;
* сбор данных о перемещении сотрудников и активов;
* управление кондиционированием и отоплением и др.

Здравоохранение:

Многие государственные больницы еще далеки от внедрения современных технологий, но в частных клиниках уже сейчас можно встретить примеры технологии Интернета вещей. К ним относят и «кнопки жизни», и передачу результатов анализов на компьютер врача, и браслеты Embrace, которые способны предсказывать приступ у эпилептиков.

Особо стоит упомянуть внутреннюю навигацию в медицинских учреждениях, которая позволяет:

* быстро найти кабинет врача;
* построить маршрут до любого объекта в больнице;
* удаленно записаться на прием;
* отследить нагрузку персонала клиники;
* выполнить трекинг медицинского оборудования.

Важная деталь в использовании Интернета вещей – энергозатратность. Подключение каждой вещи к интернету требует колоссальных ресурсов. По этой причине сегодня особое внимание уделяется разработке протоколов передачи данных. Специалисты полагают, что будущее за LPWAN-сетями, в то время как настоящее за протоколами, аналогичными ZigBee. Его мы и выбрали при разработке системы управления зданиями Perenio. Протокол передачи данных ZigBee, не тратит много энергии, позволяет сделать работу устройств автономной, а также обладает оптимальной для Интернета вещей скоростью передачи данных. [1]

**1.3 Инструменты для разработки**

**1.3.1 Язык программирования C++**

C++ — компилируемый, статически типизированный язык программирования общего назначения, на котором можно создавать программы любого уровня сложности. С++ – это мультипарадигмальный язык (от слова парадигма – стиль написания компьютерных программ), включающий широкий спектр различных стилей и технологий программирования. Часто его причисляют к объектно-ориентированным языкам, но, строго говоря, это не так. В процессе работы разработчик получает абсолютную свободу в выборе инструментов для того, чтобы задача, решаемая с помощью того или иного подхода, была решена максимально эффективно. Иными словами, С++ не понуждает программиста придерживаться только одного стиля разработки программы (например, объектно-ориентированного).

Синтаксис C++ унаследован от языка C. Одним из принципов разработки было сохранение совместимости с C. Тем не менее, C++ не является в строгом смысле надмножеством C. Со временем, практическая совместимость между языками C и C++ постепенно будет утрачиваться, так как языки разрабатывают разные группы по стандартизации, не взаимодействующие друг с другом. [2]

C++ повлиял на многие языки программирования, в их числе: Java, C#, PHP, Perl, D, Lua, Rust.

Главные отличия от Си:

1. поддержка объектно-ориентированного программирования;
2. поддержка обобщённого программирования через шаблоны;
3. дополнительные типы данных;
4. исключения;
5. пространства имён;
6. встраиваемые (inline) функции;
7. перегрузка операторов;
8. перегрузка функций;
9. ссылки и операторы управления свободно распределяемой памятью;
10. дополнения к стандартной библиотеке.

Также хотелось бы отметить список чего в языке программирования С++ не разрешается:

1. вызывать функцию main() внутри программы, в то время как в C это действие правомерно.
2. неявное приведение типов между несвязанными типами указателей.
3. использовать функции, которые ещё не объявлены.

Критики С++ не противопоставляют ему какой-либо конкретный язык, а наоборот, утверждают, что для всякого случая применения С++ всегда существует альтернативный инструментарий, позволяющий решить ту же задачу более эффективно и качественно. В свою очередь, сторонники С++ считают некорректным сравнивать различные аспекты С++ с совершенно различными языками, так как общий набор средств и возможностей С++ существенно шире, чем в большинстве языков, с которыми проводится сравнение, и сама по себе широта возможностей, на их взгляд, является веским оправданием несовершенства каждой отдельно взятой возможности.

Более того, по их мнению, высокая совместимость с Си является одной из принципиальных черт языка, и потому все недостатки С++ оправданы преимуществами, предоставляемыми этой совместимостью. [2]

Достоинства языка:

1. Высокая совместимость с языком Си
2. Вычислительная производительность
3. Поддержка различных стилей программирования: структурное, объектно-ориентированное, обобщённое программирование, функциональное программирование, порождающее метапрограммирование.
4. Автоматический вызов деструкторов объектов (в порядке обратном вызову конструкторов) упрощает и повышает надёжность управления памятью и другими ресурсами (открытыми файлами, сетевыми соединениями, т. п.).
5. Перегрузка операторов
6. Шаблоны (дают возможность построения обобщённых контейнеров и алгоритмов для разных типов данных)
7. Возможность расширения языка для поддержки парадигм, которые не поддерживаются компиляторами напрямую
8. Доступность. Для С++ существует огромное количество учебной литературы, переведённой на всевозможные языки

Также у языка есть и недостатки, которые хотелось бы перечислить [2]:

1. Плохо продуманный синтаксис сужает спектр применимости языка
2. Язык не содержит многих важных возможностей
3. Язык содержит опасные возможности
4. Производительность труда программистов на языке оказывается неоправданно низка
5. Громоздкость синтаксиса
6. Тяжелое наследие
7. Необходимость следить за памятью

**1.3.2** **Visual Studio Code**

Visual Studio Code — это редактор исходного кода. Он имеет многоязычный интерфейс пользователя и поддерживает ряд языков программирования, подсветку синтаксиса, IntelliSense, рефакторинг, отладку, навигацию по коду, поддержку Git и другие возможности. Многие возможности Visual Studio Code недоступны через графический интерфейс, зачастую они используются через палитру команд или JSON-файлы (например, пользовательские настройки). Палитра команд представляет собой подобие командной строки, которая вызывается сочетанием клавиш.

VS Code также позволяет заменять кодовую страницу при сохранении документа, символы перевода строки и язык программирования текущего документа.

С 2018 года появилось расширение Python для Visual Studio Code с открытым исходным кодом. Оно предоставляет разработчикам широкие возможности для редактирования, отладки и тестирования кода.

Также VS Code поддерживает редактирование и выполнение файлов типа «Блокнот Jupyter» (Jupyter Notebook. Напрямую «из коробки» без установки внешнего модуля в режиме визуального редактирования и в режиме редактирования исходного кода.

На март 2019 года посредством встроенного в продукт пользовательского интерфейса можно загрузить и установить несколько тысяч расширений только в категории «programming languages» (языки программирования).

Также расширения позволяют получить более удобный доступ к программам, таким как Docker, Git и другие. В расширениях можно найти линтеры кода, темы для редактора и поддержку синтаксиса отдельных языков.

**1.3.3 Язык программирования Lua**

Главная особенность Lua заключается в том, что он может использоваться как в качестве отдельного, так и встроенного скриптового языка. Уже в процессе создания разработчики поставили себе цель – сделать компактный и простой инструмент, который мог бы с легкостью работать в различных исполняющих средах и обеспечивать при этом должный уровень производительности.

Помимо этого, Луа разрабатывался с учетом того, чтобы в последующем могли легко выучить его синтаксис и понять особенности. Причем простота языка делает его интересным даже непрофессиональным разработчикам.

Lua относится к языкам программирования ООП. Благодаря встроенным средствам параллельного программирования, разработчики могут создавать многопоточные приложения без обращения к внешним библиотекам или API. При этом, в языке эффективно реализованы возможности межъязыкового взаимодействия. [3]

Где применяется язык Lua? Несмотря на достаточно широкое применение в промышленности (не будем забывать, что язык изначально разрабатывался для нужд нефтяной компании Petrobras), более активно Луа сегодня используется при разработке компьютерных игр.

Связано это с тем, что Lua позволяет довольно легко запрограммировать поведение так называемых NPC (от англ. Non-playable characters). Также, с помощью этого языка программируются и другие персонажи, поведение которых можно впоследствии быстро менять, не оказывая влияния на движок.

Наиболее известным игровым продуктом, в котором применяется язык Lua является World of Warcraft. Здесь язык использовался при написании интерфейса. Также, Lua активно использовался и при создании других известных игровых шедевров, таких как Цивилизация 5, Crysis, Sim City, Far Cry, Stalker. [3]

Помимо игрового применения, Луа также подходит для написания ботов для Телеграм, например. Также, Википедия планирует использовать данный язык разработки для встраивания в MediaWiki.

Наконец, применяется Lua и в обсерваториях, которые занимаются исследованиями космоса. Язык также используется различными университетами. А в самой Бразилии его применяют повсеместно. То есть он стал чем-то вроде государственного языка программирования.

Язык программирования обладает следующими преимуществами:

1.Lua обладает хорошей портируемостью. Например, можно без особых изменений перенести программу из Windows на Linux.

2.Большое количество библиотек. Несмотря на то, что по этому параметру Lua проигрывает JavaScript, на официальном сайте языка можно найти все необходимые библиотеки для работы, которых более чем достаточно для выполнения различных задач.

3.Возможность быстрого добавления новых библиотек, написанных на языке С.

**1.3.4 Технология интернет вещей (IoT)**

Простыми словами это сеть, в которой между собой общаются устройства, а не пользователи. Британский технолог Кевин Эштон, ввел в оборот определение “Интернет вещей” в 1999 году и определяет его так: “если в двадцатом веке данные попадали в компьютер только от человека, использовавшего для их ввода дополнительные устройства, то в двадцать первом веке мы уже имеем дело с гаджетами, которые сами могут собирать и отправлять данные. Посмотрите, например, на свой смартфон. Можно сказать, что у него есть десять чувств. Он может определить, где находится, в каком направлении движется. У него есть представления об изменении давления, времени.  Он даже может предсказать погоду. Вам не нужно сообщать ему все эти данные посредством ввода, благодаря мобильным приложениям смартфон собирает их самостоятельно. В этом и состоит суть Интернета вещей. Данные собираются, обрабатываются и передаются устройствами без участия человека”.

IoT основан на сборе, обработке, передаче и хранении данных. Одним из основных направлений стало развитие технологий M2M — от машины к машине. Разработка протоколов передачи данных от одного устройства к другому началась задолго до изобретения термина и определения его специфики. Изначально было ясно, что передача данных должна быть энергоэффективной, безопасной и точной. [4]

Индустриальный Интернет вещей связан с темой, которой занимался Эштон — оптимизацией производства. В отличие от M2M, он также учитывает человеческий фактор, но в целом строится на принципах передачи данных от машины к машине.

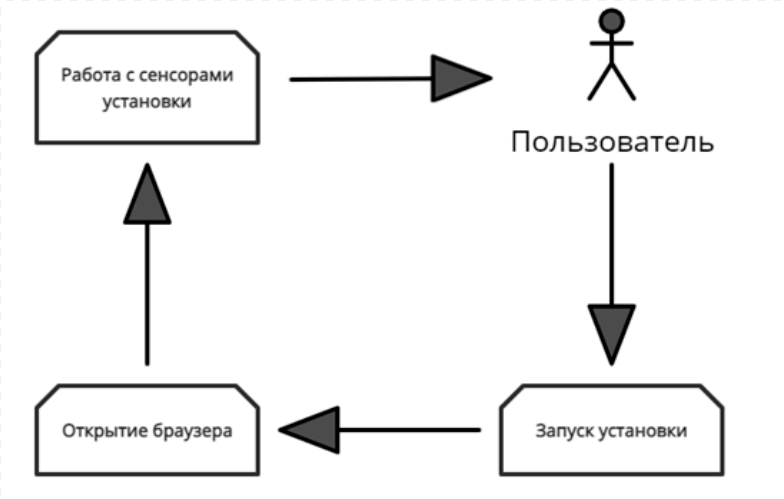
Также Интернет вещей подтолкнул к развитию новых концепций. Так появилась идея Всеобщего интернета или Internet of Everything (IoE). Эта технологическая концепция не видит разницы между человеком и устройством, полагая, что в первую очередь необходимо изобретать способы обмена данными. Не важно, происходит эта операция при участии человека или без него. Популярность набирает и идея Четвертой индустриальной революции, неотъемлемым элементом которой считается Интернет вещей. По мнению ее сторонников, IoT наравне с виртуальной и дополненной реальностью, большими данными (Big Data), 3D-печатью, блокчейном и квантовыми вычислениями полностью изменит не только производство, но и представление о мире. [4]

Сегодня Интернет вещей используется фактически во всех сферах жизни человека. Например, он очень востребован в медицине. Это привело к созданию нового термина – Интернета медицинских вещей (Internet of Medical Things). На нем основываются «умные города» (Smart Cities). Он также используется в разработке гаджетов для «умного дома».

**2 Логическая и физическая структуры данных**

**2.1 Логическая и концептуальная структура данных**

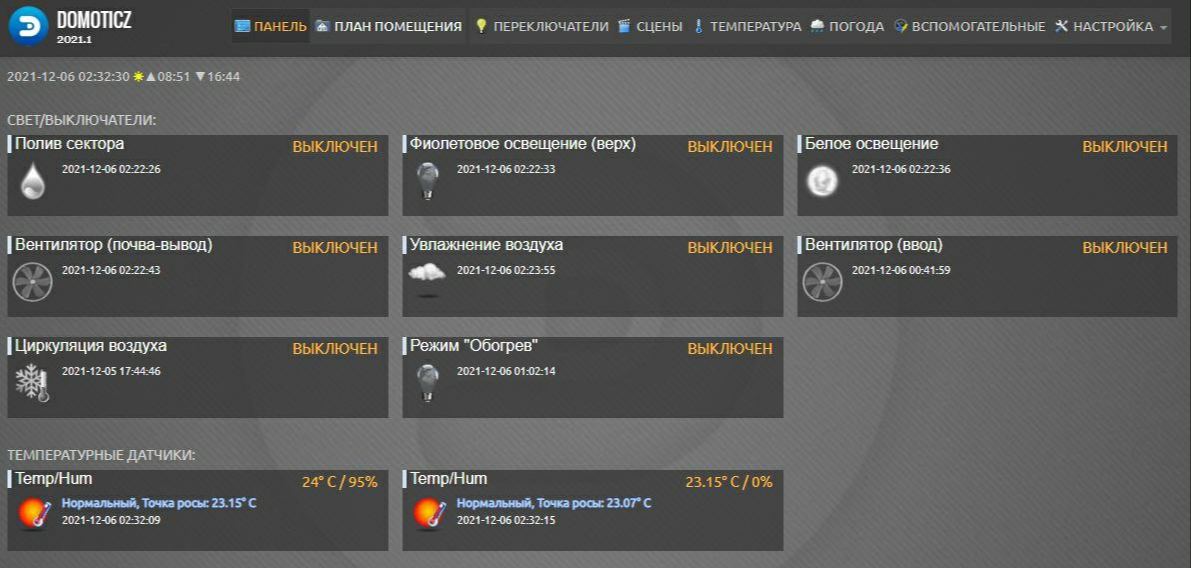
Для определения работы с сенсорами была определена следующая диаграмма претендетов. Для начала работы пользователь должен обратиться к физическому запуску электронной установки, произвести пусковые работы. Пусковые работы установки заключаются в: подключении установки к сети питания, включение тумблеров ограничения питания. После чего произвести проверку запуска системы. После успешного запуска системы и проверки всех узлов, необходимо запустить браузер для дальнейшего перехода в графическую систему управления. В работе с графической системой управления имеется возможность настройки автоматизации системы в соответствии с различными сенсорами. Диаграмма претендетов приведена на рисунке 1.



**Рисунок 1 – Диаграмма претендентов**

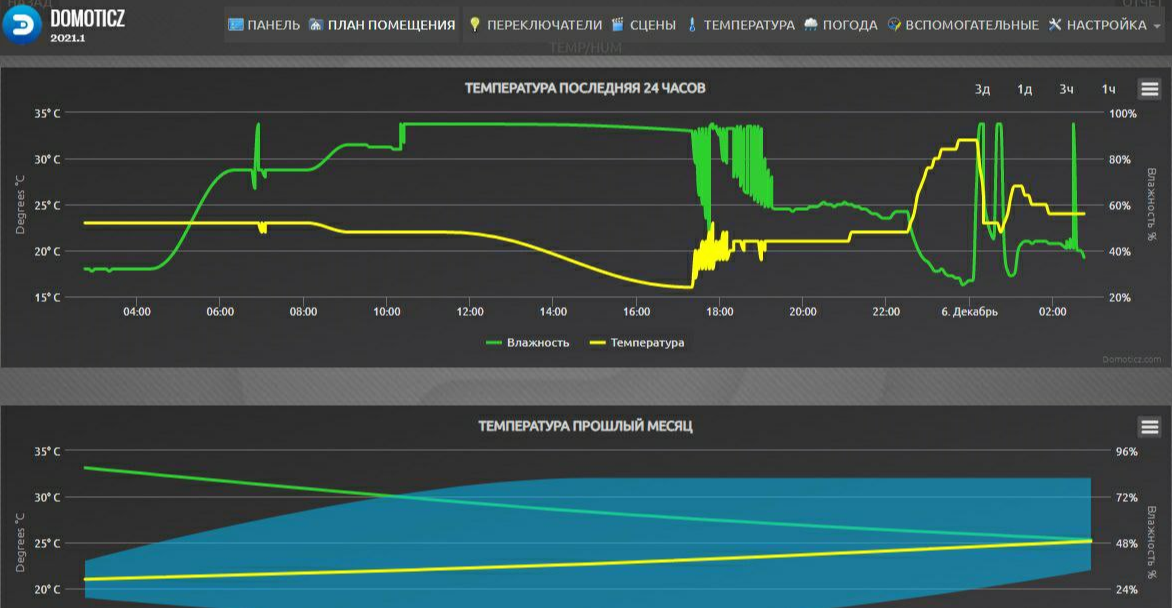
Для работы системы с сервером была выбрана система семейства Linux, а именно Armbian. Система была установлена на одноплатный компьютер OrangePi1 с следующими системными характеристиками: 265/HEVC 4 К; CPU – Mali400MP2 GPU (c поддержкой OpenGL ES 2.0), оперативная память – DDR3 512 Mb. На съемный носитель была установлена система Linux

Armbian 5.25, после чего смонтирован образ и установлен на одноплатный компьютер. После успешной установки операционной системы в качестве базового приложения сервера была установлена OpenSourse приложение domoticz. Это очень легкая система домашней автоматизации, которая позволяет контролировать и настраивать различные устройства. Данное приложение использовалось в качестве скелета системы с набором необходимых библиотек. Установленное приложение было доработано и адаптировано для выполнение параметров автоматизации. Концепт разработанного графического образа представлен на рисунке 2. [5]



**Рисунок 2 – Концепт дизайна взаимодействия с сенсорами**

Далее после активации датчиков можно перейти во вкладку с информацией о датчиках. К примеру, можно просмотреть на панели устройств параметрические данные температуры, влажности воздуха, а также влажность почвы. Данные представлены на рисунке 3.



**Рисунок 3 – Отображение датчиков**

**3** **Разработка автоматизации системы на базе сервера Linux**

**3.1 Постановка задачи**

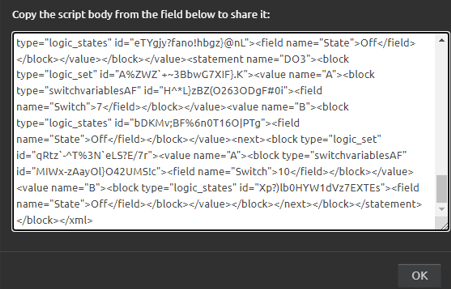
Цель данной курсовой работы является настройка разработанной системы климатической установки для выращивания растений и адаптация к ней установленного сервера. Будут разработаны системы автоматизации для общения сервера с датчиками, установленными в климатической камере. А также настройка датчиков на отправление обратной информации на сервер. После настройки взаимодействия датчиков с сервером делаем алгоритмизация данной установки, а именно: настройка событий при определенных условиях (полив почвы, подогрев воздуха при низких температурах) рисунок 4.



**Рисунок 4 – Внешний вид установки**

**3.2** **Взаимодействия сервера с установкой**

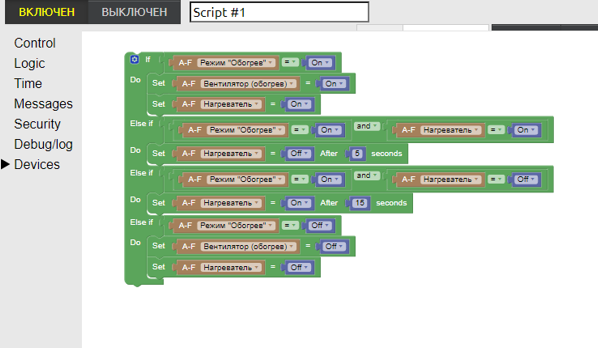
Для того чтобы управлять одним из элементов климатической камеры, в первую очередь нужно произвести на нее какую-то манипуляцию. Таким образом для управления реле, которые в свою очередь установлены в системе, подключен wifi модуль и каждый элемент климатической камеры является в свою очередь элементом интернет-вещей. То есть у каждого элемента управления есть свой ip-адрес. Посредством этого есть прямое взаимодействие с датчика установки. Чтобы считывать данные нужно создать объект, в котором разместить ip-адрес датчика, также создаем хранилище, в которое будут приходить данные (каждый 5 секунд) с данного сенсора. В свою очередь все данные работают на базе языка С++ и отправляются в хранилище с помощью интернет-вещей. [6]



**Рисунок 5 – Пример кода**

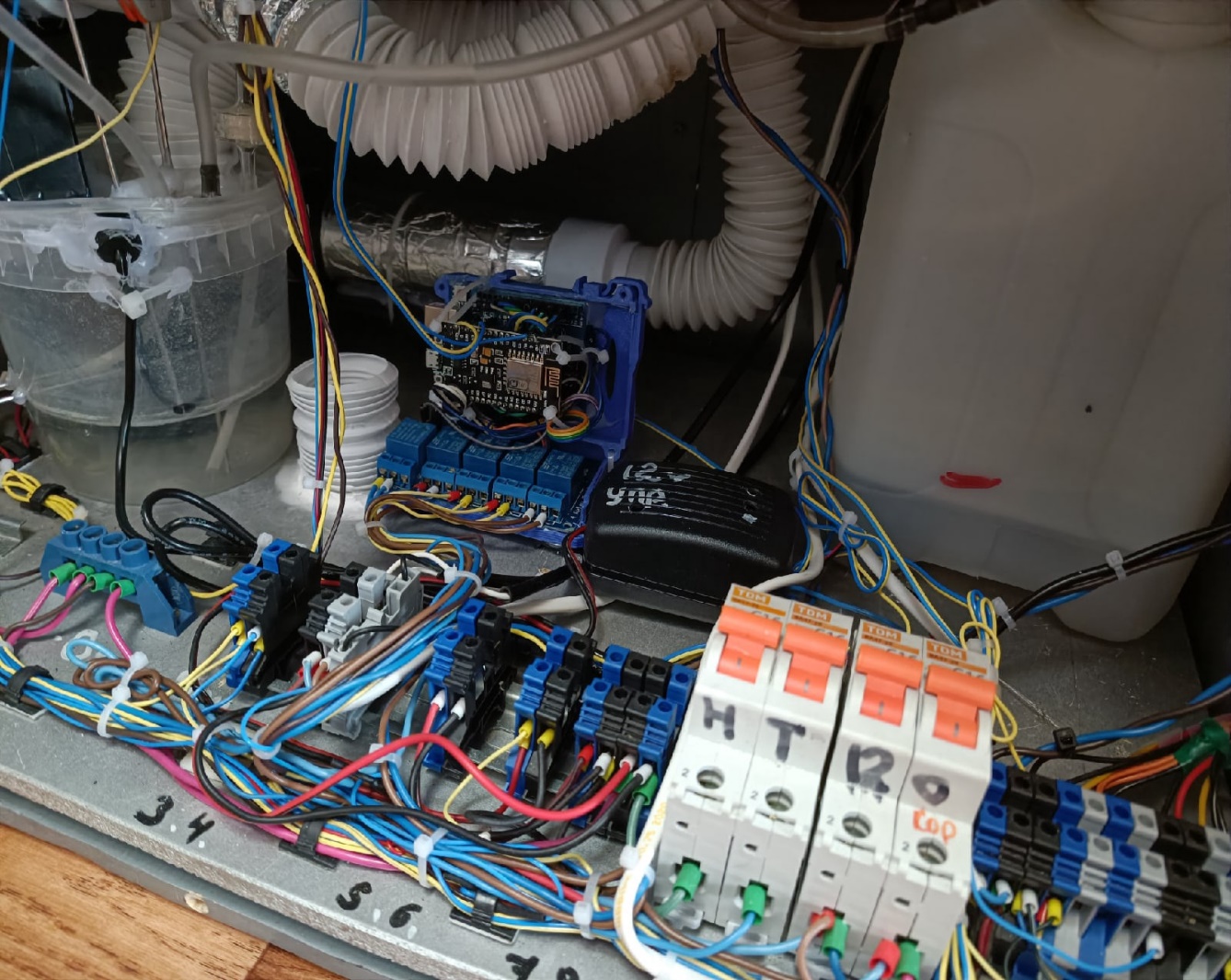
Как было выше сказано каждый элемент является элементом общей локальной сети и у каждого есть свой ip-адрес. Помимо всего этого у каждого датчика также есть свой уникальный id, которому в свою очередь прописывается свой протокол, по которому будут отправляться данные. Для реле принимается и отправляется логические ноль или единица (включение, выключение, соответственно). Для датчиков принимается протокол, на котором они отправляют свои данные телеметрии. [6]

Для того чтобы создать кнопку, которая будет взаимодействовать с датчиками, нужно прописать 2 кода, а именно: первый – это http запрос на отправку единицы (то есть включение реле), второй - http запрос по ip-адресу, который отправляется на реле логический ноль (для выключения реле). Каждая кнопка взаимодействует со своим реле, для того чтобы наладить между ними взаимодействие используются простые операторы условия, которые можно описать кодом, после чего система сама сгенерирует блок схему под определённую задачу рисунок 6.



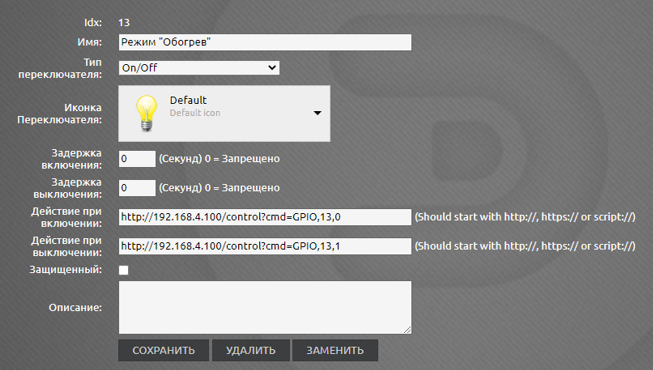
**Рисунок 6 – Блок схема кнопки**

Добились того, что у каждого реле и устройства свой ip-адрес тем, что под устройства был использован микроконтроллер ESP-8266 рисунок 7. Который в свою очередь дает возможность установить прошивку и сделать устройства объектом локальной сети имеющей какой-то адрес. Управлять ей можно с помощью wifi отправляя и получая с нее данные.



**Рисунок 7 – Внешний вид электроники**

После всего вышеописанного была создана кнопка “Обогрев”, которая отвечает за подогрев воздуха, при понижении температуры в камере рисунок 8.



**Рисунок 8 – Внешний вид кнопки**

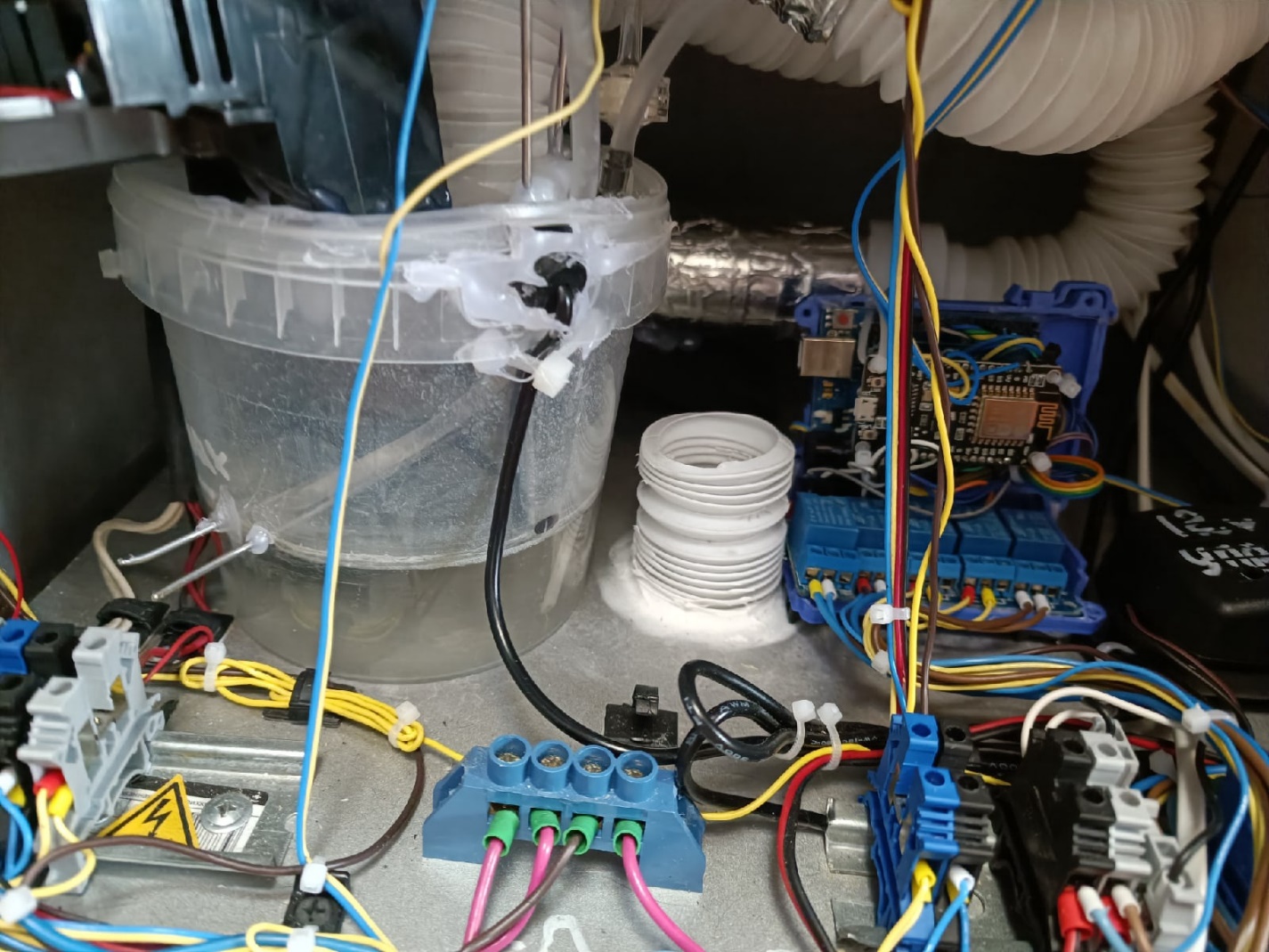
Ниже представлен внешний вид камеры установки, в которой и происходят все климатические изменения, задаваемые через приложение рисунок 9.

****

**Рисунок 9 – Внешний вид камеры**

Было настроены следующие функции: ультрафиолетовое излучение, белое освещение, обдув почвы, повышение/понижение температуры, и изменение влажности воздуха.

В свою очередь влажность почвы изменяется засчет предварительно залитой воды в установку, которая перетекает по трубам в резервуар с активной водой, где в свою очередь вода, испаряясь увлажнят камеру рисунок 10.



**Рисунок 10 – Внешний вид контейнера с водой.**

Резервуар с активной водой, в котором должна быть всегда вода, иначе электроника может сгореть.

**Заключение**

В первой главе была поставлена цель и задачи курсового проекта, а также подробно описана предметная область. В второй главе была проанализирована информация и составлена диаграмма претендентов. Также было положено начало разработке веб-приложения. В заключительной главе был подробно описан каждый шаг взаимодействия сервера с установкой, его структура и особенности каждой функции. Также были приведены примеры кодов некоторых функций и скриншоты программы.

**Список используемой литературы**

1 IoT технология [Электронный ресурс] – Режим доступа: https://nvgn.ru/blog/vse-ob-internete-veshey/– Дата доступа: 05.05.2022.

2 С++ [Электронный ресурс] – Режим доступа: http://web.spt42.ru/index.php/chto-takoe-c-plus-plus – Дата доступа: 05.05.2021.

3 Lua [Электронный ресурс] – Режим доступа: : <https://zserge.wordpress.com/2012/02/23/lua> – Дата доступа: 05.05.2021.

4 История IoT [Электронный ресурс] – Режим доступа: https://perenio.ru/blog/the-history-of-the-internet-of-things – Дата доступа: 05.05.2021.

5 Применение OpenSourse Domoticz [Электронный ресурс] – Режим доступа: https://habr.com/ru/post/535838/ – Дата доступа: 08.05.2021.

6 Взаимодействие сервера с Domoticz [Электронный ресурс] – Режим доступа: https://www.ixbt.com/live/chome/orange-pi-one---ustanovka-domoticz-servera-s-nulya.html – Дата доступа: 10.05.2021.