

## Purple Dragons представляет проект Purple City



Г. Москва

Участники:

Руководители:

АНО ДО «НОЦ МГТУ им. Н.Э.  
Баумана»

Назарова Полина Алексеевна  
Панюшкина Виола Сергеевна  
Фролова Ксения Николаевна  
Капитонов Даниил Дмитриевич  
Секисов Давыд Михайлович

## **Оглавление**

<b>1 Проблема.....</b>	<b>3</b>
<b>2 Идея проекта .....</b>	<b>3</b>
<b>3 Проведенные исследования .....</b>	<b>3</b>
<b>4 Используемые средства по накоплению и генерации энергии .....</b>	<b>6</b>
<b>5 Одна планета - один человек.....</b>	<b>6</b>
<b>6 Электронные и роботизированные системы в помощь людям .....</b>	<b>7</b>
<b>7 Электрический общественный транспорт .....</b>	<b>8</b>
<b>8 Используемые программные решения для работы нашей системы ..</b>	<b>9</b>
<b>8.1 Нейросети .....</b>	<b>9</b>
<b>8.2 Приложение.....</b>	<b>10</b>
<b>8.3 Сайт .....</b>	<b>10</b>
<b>8.4 Автопилот для автомобиля.....</b>	<b>12</b>
<b>8.5 Программный код для Arduino.....</b>	<b>13</b>
<b>8.6 ТСР сервер.....</b>	<b>14</b>
<b>9 Электронные компоненты и их использование .....</b>	<b>16</b>
<b>10 Механическая составляющая .....</b>	<b>17</b>
<b>11 Подробнее про модели и декорации .....</b>	<b>18</b>
<b>Дополнительные материалы .....</b>	<b>19</b>
<b>Приложение А.....</b>	<b>20</b>

## **1 Проблема**

Все больше и больше стран развивают свои технологии для выработки «чистой» энергии. Они вкладывают средства в разработки, направленные на увеличение вырабатываемой энергии из «чистых» и возобновляемых источников энергии, таких как ветрогенераторы, солнечные панели, ГЭС, волновые генераторы и другие. Но, к сожалению, большинство из этих источников не могут вырабатывать энергию постоянно (рисунок 2), и количество этой энергии также варьируется. При этом люди используют эту энергию в любое время суток, невзирая ни на какие погодные или другие условия, которые, к примеру, могут повлиять на выработку энергии. Эта проблема имеет несколько решений, которые уже активно применяются - накопление энергии в аккумуляторных батареях или, при нехватке «чистой» энергии, подключение резервных источников выработки энергии.

## **2 Идея проекта**

В своём проекте города, где энергия «умнее» потребителя Purple City, мы постарались решить проблемы с использованием «чистой» энергии, совместив уже имеющиеся методы и дополнив их своими идеями. Мы сделали модель города с несколькими кварталами - жилыми и социальным, зону выработки «чистой» электроэнергии и сеть дорог для движения по ним автомобилям. Оставшуюся часть проекта посвятили реализации своих и улучшению заимствованных идей по «оптимизации» использования и хранения энергии.

## **3 Проведенные исследования**

При выборе темы для проекта мы изучали проблемы, связанные с разными потребителями энергии в городской электросети. Исследования показали, в какое время человек больше всего тратит электроэнергии, и

несмотря на кажущуюся очевидность этих фактов были удивлены, что больше всего энергии дома человек тратит в период с 8 до 10 утра и с 18 до 23 часов вечера (рисунок 1). В остальное время трата энергии стремиться к минимуму, а выработка энергии с помощью возобновляемых источников к максимуму, что зачастую приводит к тому, что энергия тратится вхолостую, а было бы хорошо её сохранять или равномерно тратить. В этом и есть весь Purple City.

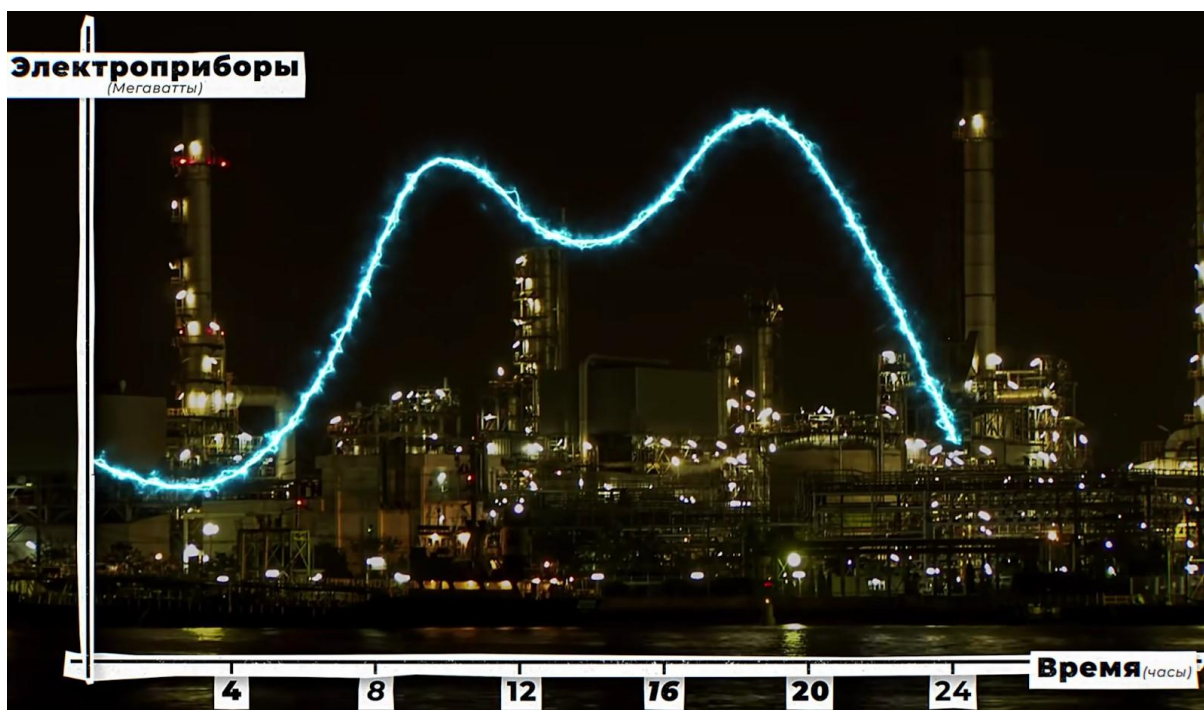


Рисунок 1 – График потребления электричества

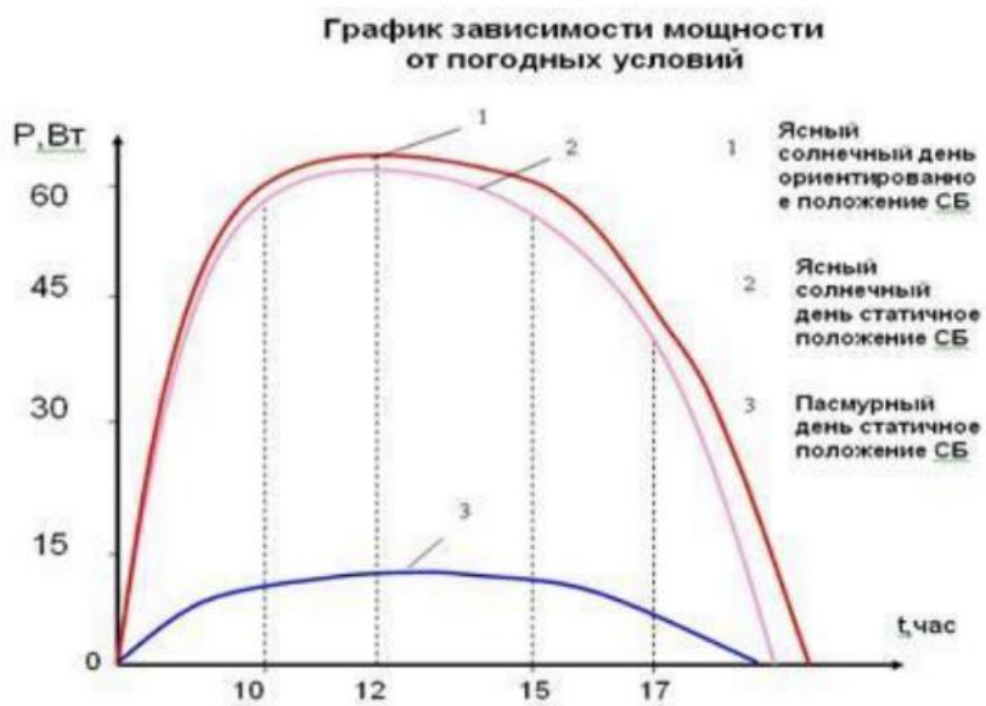


Рисунок 2 – График выработки электроэнергии в течении дня с помощью солнечных панелей

#### **4 Используемые средства по накоплению и генерации энергии**

В своём городе в качестве «чистых» источников энергии мы используем самые оптимальные и наиболее развитые устройства: солнечные панели - 6 штук и ветрогенераторы - 2 штуки.

В качестве накопления энергии мы используем стандартные аккумуляторные батареи, но исключительно в качестве временного хранилища энергии. В качестве основного хранилища энергии мы используем аккумулирующую ГЭС. При наличии «свободной» энергии система автоматически включает насосы и закачивает воду на уровень выше. При нехватке энергии открывается электрический клапан и вода, проходя через гидрогенератор, вырабатывает энергию обратно. Само собой, мы теряем около 15% электроэнергии, но это позволяет нам не использовать вредные в производстве аккумуляторные батареи, что очень важно в виду общемировой тенденции сокращения выбросов вредных веществ в атмосферу.

#### **5 Одна планета - один человек**

Кажется, что наша планета это нечто огромное и влияние одного человека на неё не значительно. А если так подумают все люди? Тогда ни один человек будет приносить вред природе, а несколько миллиардов, что существенно отразится на экологии, как и происходит в последнее время. Глобальное потепление, вымирание животных и растений, загрязнение всего, куда ступает нога человека... И поэтому каждый может все делать правильно, каждый из нас может внести хотя бы маленьких вклад в сохранение экосистемы нашей любимой планеты, планеты - Земля.

## **6 Электронные и роботизированные системы в помощь людям**

Мы понимаем, что менять привычный образ жизни для каждого человека - это трудно, но для сохранения нашего общего дома придётся потрудиться.

Команда Purple Dragons разработала программную систему по автоматическому контролю вырабатываемой и расходуемой энергии. Основная система, включающая в себя 2 нейронные сети, интерфейсы работы с пользователем (рисунок 3) и сопутствующие программы. Система составляет прогноз по выработке и расходу энергии, основываясь на актуальных погодных показателях, прогнозирует количество энергии, которое потратится и количество энергии, которое выработается, затем анализирует остаток энергии в аккумуляторной батарее и ГЭС, на конец следующего часа и формирует рекомендации по использованию электроэнергии на грядущий час. Помимо этого, система учитывает приоритеты социальных строений, таких как больница, МЧС и администрация, отдавая им приоритет в рекомендациях.

Простой житель города в свою очередь может выбрать хочет ли он просто получать рекомендации по включению и отключению электроприборов или система будет автоматически выключать из сети его приборы. За доверие системе предусмотрена скидка на оплату электрического тарифа. А в случае нехватки электроэнергии в системе город автоматически будет запрашивать необходимое количество электроэнергии у электростанций.

К примеру, в случае переизбытка энергии система будет рекомендовать пользователю поставить на зарядку робота пылесоса, включить посудомойку, зарядить электромобиль или система отправит на зарядку общественные электромобили под управлением автопилота. Все эти и подобные действия зачастую можно сделать на протяжении какого-то времени, именно это время и находит система.

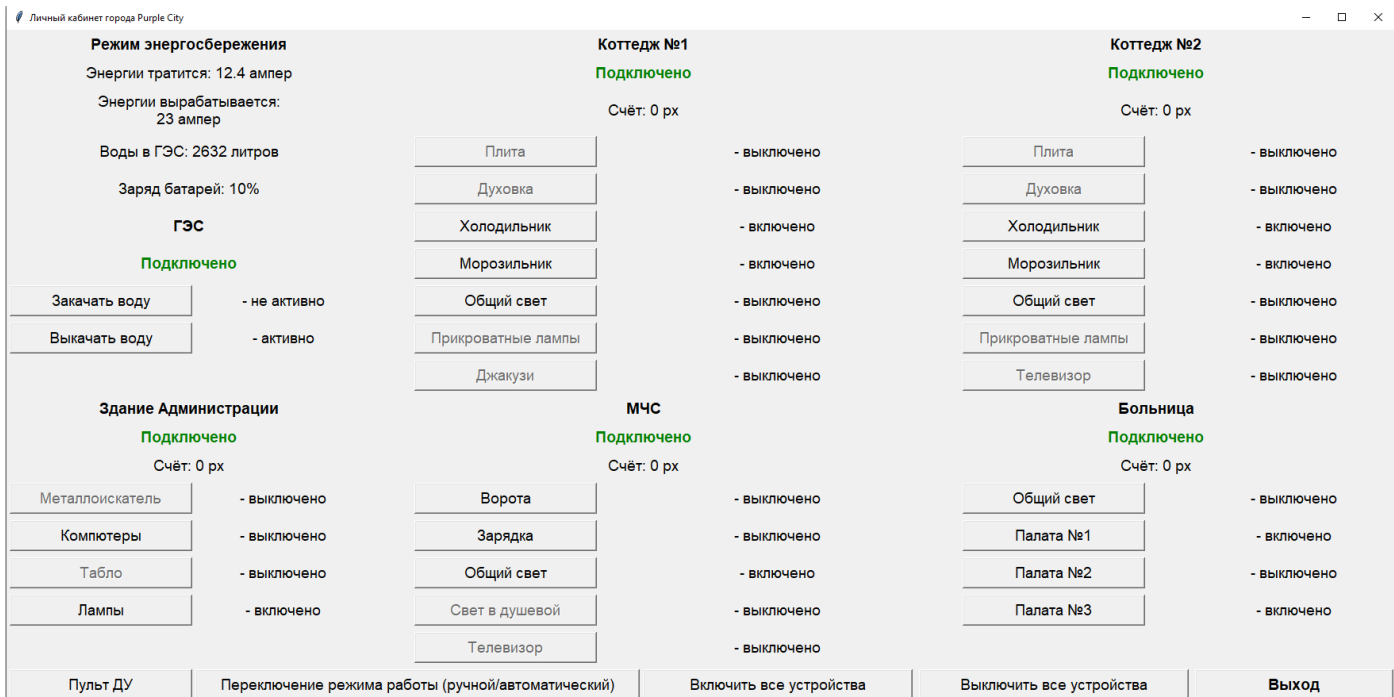


Рисунок 3 – интерфейс личного кабинета города

## 7 Электрический общественный транспорт

Жители города будут иметь возможность за небольшую плату перемещаться по городу на беспилотном электрическом автомобиле, работающем на “чистой” энергии. В нашем городе представлена сеть дорог по которой будут циркулировать общественные автомобили, и житель города по нажатию на кнопку сможет вызвать к своему дому такой автомобиль.

Такое решение позволит помочь людям отказаться от использования личного автотранспорта, не важно электрический он или на ДВС, что в первом случае позволит сократить выбросы при производстве аккумуляторных батарей, а во втором выбросы в атмосферу выхлопных газов.

В модели нашего города представлена сеть дорог, которые предназначены, как для беспилотных автомобилей, так и для обычных,



также мы расположили в нём автоматическую станцию зарядки, которая присоединяется к электромобилю через разъём на крыше и заряжает его.

## 8 Используемые программные решения для работы нашей системы

### 8.1 Нейросети

Нейросети написаны на языке программирования Python, с помощью библиотеки Keras, встроенной в библиотеку Tensorflow. Также использовались библиотеки numpy, pandas, matplotlib. Для предсказаний мы брали настоящую Московскую погоду с помощью библиотеки OWM.

Архитектура нейросети:

model - Sequential

BatchNormalization

Dense - 128, activation='relu'

Dense - 1000, activation='tanh'

Dense - 100, activation='relu'

Dense - 1, activation='linear'

optimizer=Adam

Sequential - это модель, в которой слои располагаются последовательно один за другим, подобно цепочке блоков.

Слой Batch Normalization нужен для приведения данных к нормальному распределению.

Dense (- количество нейронов, активационная функция) - полносвязный слой. Это слой, все нейроны которого связаны с нейронами предыдущего слоя.

Функция активации (activation) позволяет установить условия, которые будут определять: передавать выходное значение в следующий слой (активировать нейрон) или не передавать.

Механизм, который отвечает за процесс обновления весов в Keras называется оптимизатором(optimizer).

В нашем проекте используются две нейросети с почти одинаковыми архитектурами. Одна для дома, другая - города. Как датасет использовались файлы csv с данными собранными с Kaggle. В будущем мы добавим более точную систему, которая будет опираться на большее количество параметров.

## 8.2 Приложение

Приложение также написано на Python, интерфейс сделан с помощью библиотеки tkinter. Одновременно с приложением работают скрипты:

- Скрипт по определению разрешений, того, что можно включать в этом режиме и того, что включать нельзя
- Скрипт, который списывает деньги со счетов пользователей каждый час
- Скрипт, считывающий данные с ТСР- сервера
- Система (скрипт по определению режима работы)
- Телеграм бот
- Сайт

## 8.3 Сайт

Сайт (рисунок 4) проекта Purple City был написан для возможности пользователя просматривать нужную информацию того, или иного здания/дома через удобный и красивый интерфейс. Backend сайта был написан на языке программирования Python, с использованием известного фреймворка Flask и соответствующих для него дополнений flask\_cors и flask\_session. Flask\_cors нужен для разрешения агенту пользователя делать запрос с другого источника (cross-origin HTTP request), если источник

текущего документа отличается от запрашиваемого ресурса доменом, протоколом или портом.

Flask\_session нужен для реализации файлового хранения данных о пользователе, что в свою очередь является своеобразной базой данных, состоящей из информации о всех пользователях, заходивших на сайт.

Frontend сайта был написан на VueJS - JavaScript - фреймворк с открытым исходным кодом для создания пользовательских интерфейсов. Он легко интегрируется в проекты с использованием других JavaScript-библиотек. Может функционировать как веб-фреймворк для разработки одностраничных приложений в реактивном стиле.

Для реализации красивого интерфейса был использован модуль vuesax, который собрал в себя большое множество тегов и различных встроенных функций и методов, позволяющих быстро сверстать приятный для пользователей сайт.

Для взаимодействия бэкенда и фронтенда, пользователя и сайта, используются request-ы. Request - это определённое сообщение, отправляемое клиентом серверу

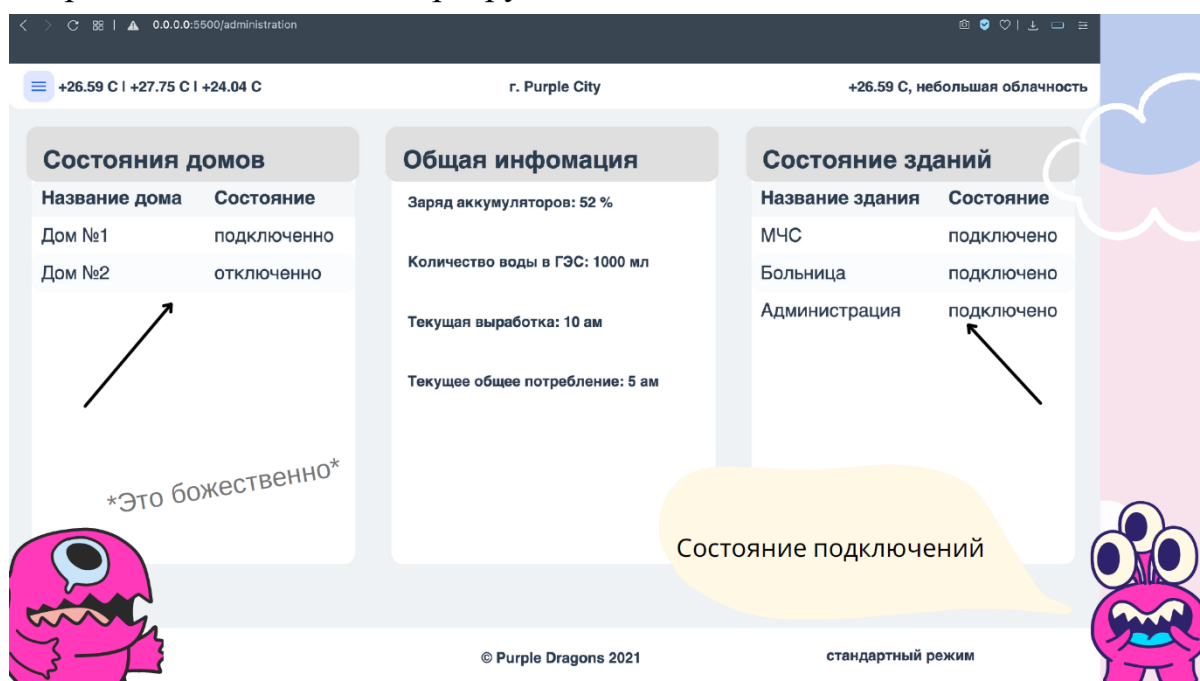


Рисунок 4 – Сайт

## 8.4 Телеграм бот

Для создания бота использовался BotFather. Мы добавили картинку, команды и описание. Добавили функционал, с помощью которого жители города смогут узнать основную информацию прямо в своем смартфоне!

Бот написан на Python библиотеки с открытым кодом-telebot. Взаимодействие с телеграмом происходит с помощью api. API (Application Programming Interface) - конструкции языка программирования, позволяющие разработчику строить сложную функциональность с меньшими усилиями. Они "скрывают" более сложный код от программиста, обеспечивая простоту использования. Проще говоря, api-переводчик с языка программиста на язык приложения. В нашем случае Telegram представляет своё собственное API для более удобной реализации тех, или иных задач, связанных с автоматизацией всевозможных процессов. На данном этапе проекта бот полностью переписывается, как и вся программная часть, поэтому он, пока, полностью не функционирует, но в будущем мы добавим больше "фич" и возможностей.

## 8.5 Автопилот для автомобиля

Мы перепробовали более 15 прототипов: использовать знаки, распознавание фигур, bounding box, езду по линии (сегментировать дорогу и пытаться понять куда она поворачивает), поиск контуров и у каждой такой идеи были разные возможности реализации.

Пока самым эффективным решением себя показала сегментация, но уже были найдены проблемы. К примеру, если количество машинок меняется, то программу надо менять, это не очень хорошо. Поэтому в будущем мы перейдем на индивидуальные камеры для каждой машинки.

Как работает сегментация? Это не нейросети, машинное обучение — это, конечно, здорово, но слишком медленно, поэтому мы остановились на алгоритме, подобрав вручную правильные параметры наша программа с высокой точностью “закрашивает” дорогу и, пока, не так хорошо-машинку, из-за того, что машинка неровная и свет на разные части наклейки падает по-разному. Это еще один аргумент “за” индивидуальные камеры, так как мы избавляемся от задачи сегментации.

Нам немного не хватило времени, чтобы дописать систему позиционирования, которая позволила бы машинке ехать, но в связи с тем, что данное решение не самое эффективное мы не будем реализовывать данную систему позиционирования в будущем. Вместо этого мы напишем принципиально новый автопилот, который будет выигрывать в точности и скорости.

## **8.6 Программный код для Arduino**

В нашем проекте используется 4 платы Arduino все они запрограммированы на языке C++ (рисунок 5) в среде программирования Arduino IDE, сами по себе они являются лишь исполнителями или считывателями какой-либо информации. В код каждой платы интегрирован механизм работы с TCP сервером через библиотеку Ethernet. Остальной программный код либо считывает полученные пакеты, либо отправляет показания с датчиков, например, с датчиков тока.

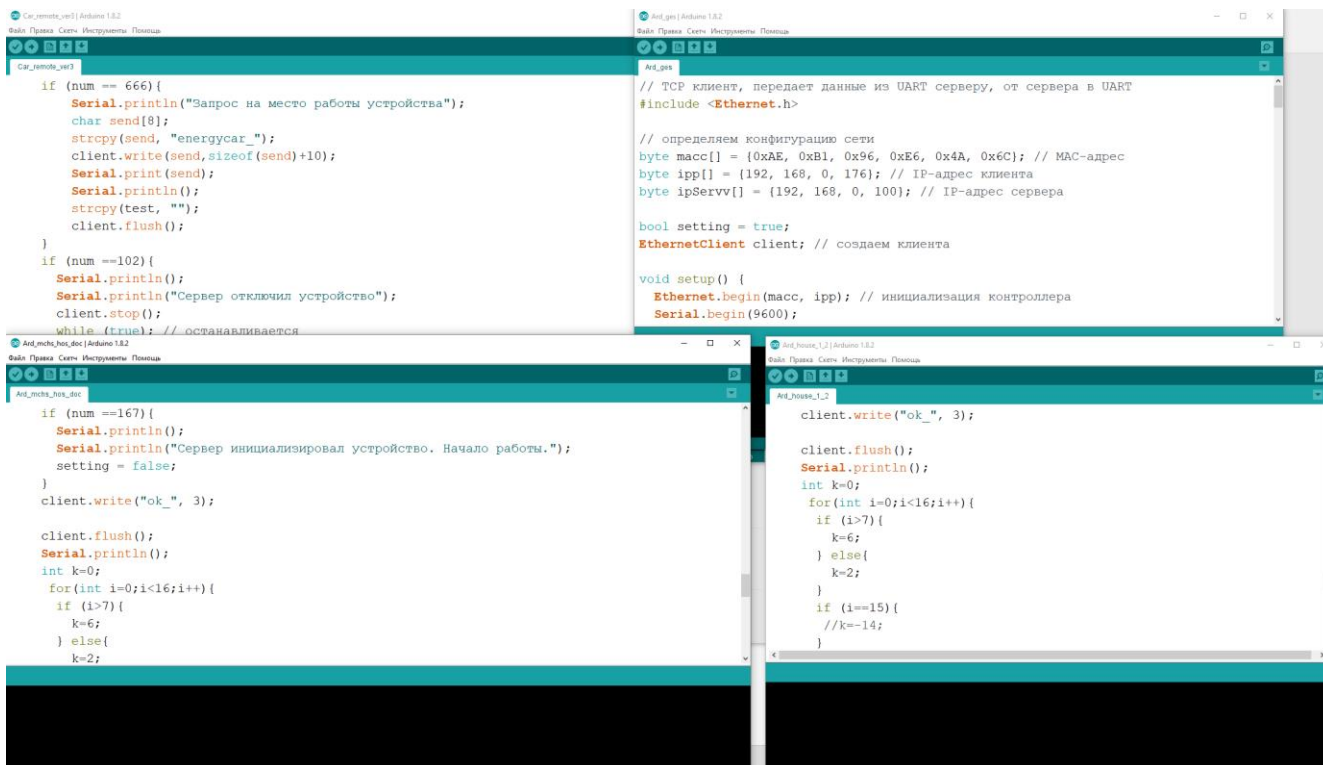


Рисунок 5 – Программы Arduino

## 8.7 TCP сервер

Такое большое количество информации между устройствами необходимо как-то передавать, усложняет ситуацию ещё и то, что все устройства имеют разную платформу, операционную систему и др. Серверу необходимо передавать данные между Arduino, Raspberry и Macbook.

Изначально мы хотели использовать Bluetooth, но отказались от этой идеи из-за сложности настройки на разных устройствах, затем планировали настроить Wi-Fi сеть, чтобы в беспроводном режиме все устройства могли “общаться”, эта идея нам понравилась больше, но возникла проблема того, что на крупных мероприятиях Wi-Fi легко “теряется” в виду большого количества других сетей, поэтому было принято решение использовать соединение по Ethernet кабелю (рисунок 6).

Для этого у нас в системе есть роутер с встроенным DHCP сервером и маршрутизатор для увеличения количества возможных подключаемых устройств.

Было принято решение поставить сервер на Raspberry. Сервер написан на языке Python, используя библиотеки Socket и Multiprocessing. Первая нужна для возможности создания сети устройств на нижнем уровне сети. Вторая была добавлена чтобы на каждое устройство выделить отдельный поток, а также чтобы сервер обрабатывал каждое устройство параллельно другим. Была разработана система специальных команд для общения устройств с сервером. Это было сделано для безопасности подключения устройств и для удобства работы с устройствами.

Ещё, одной немало важной частью, является устойчивость системы, а именно сервера по обмену данными, он должен уметь правильно реагировать на всевозможные команды и данные и в случае ошибки отключать только “сломавшегося” клиента. Эта особенность была успешно реализована посредством библиотеки multiprocessing, при экстренном завершении работы одного из процессов другие продолжают работать в штатном режиме.

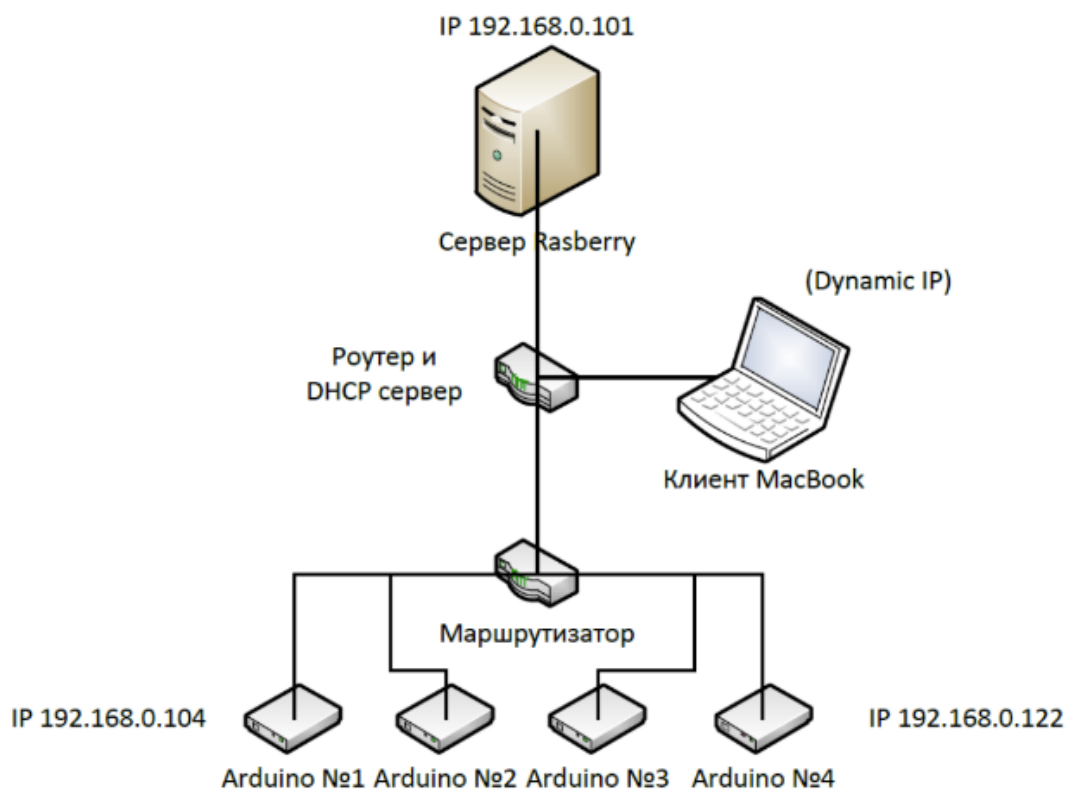


Рисунок 6 – Топология сети

## 9 Электронные компоненты и их использование

Наш город делится на 4 условных сектора. Для каждого используется по одной плате Arduino UNO с Ethernet модулем. Две из них находятся в жилых и социальном районах, и, вместе с двумя 16-ти канальными релейными модулями они переключают различные устройства внутри зданий. Понять, включен ли электроприбор и потребляет ли он энергию, можно с помощью состояния светодиода. Тогда можно услышать характерный щелчок реле и увидеть яркое свечение. Мы попытались максимально симитировать работу бытовой техники, чтобы наглядно показать потребление электричества в современной жизни. Питание для компонентов идёт от средств получения альтернативной энергии.

Вся электроника проекта спроектирована (приложение А) в модульном исполнении. Нет единой платы проекта. Проект может быть легко модернизирован или починен путем замены необходимого модуля.

Состав модулей:

- Большой модуль питания
- Малый модуль питания
- Модуль arduino с шилдом x 4 шт.
- Модуль гидроэлектростанции
- Модуль учета электроэнергии
- Модуль реле x 3 шт.

В качестве соединения проводников в проекте используется пайка, клемные колодки. Для удобства монтажа и демонтажа стенда все здания обеспечены разъемными электрическими соединениями.

Диаметры проводников подобраны с учетом удобства монтажа и с учетом величины максимального допустимого протекающего тока. Они составляют 0,6 мм и 1 мм для высокоточковых участков.



В качестве электроизолирующего материала использовалась термоусадочная трубка и изолирующая лента.

Питание проекта автономно и осуществляется сборкой из высокотоковых аккумуляторных батарей общим напряжением 11,2 - 12,6 Вольт. Для защиты батарей от перегрузки и просадки напряжения используется специальная BMS плата (battery management system). Питание устройств, работающих от 5 Вольт осуществляется через высокоэффективный импульсный dc-dc преобразователь напряжения.

Управление нагрузкой осуществляется через электромагнитные реле. Использование реле обусловлено отсутствием необходимости частого срабатывания. Также наличие реле позволяет при необходимости включить в цепь бытовые устройства напряжением 220 Вольт за счет гальванической развязки схемы через реле. Это в свою очередь позволяет легко перейти от схемы макета к схеме прототипов и готовых устройств.

## **10 Механическая составляющая**

Несмотря на большое количество программных решений мы также внедрили механические устройства в свой город.

- Аккумулирующая ГЭС, она использует два нормально-закрытых клапана для прерывания подачи воды, помпу, поплавков и гидрогенератор. Также для неё было приобретено два резервуара объемом около 30 литров каждый для перекачки воды из одного резервуара в другой.
- Модули реле - они имитируют работу электрощитка и показывают, как система может самостоятельно или по указанию пользователя включать или выключать различные устройства, в том числе зарядку для роботов пылесосов и других робототехнических устройств.

- Автомобиль с автопилотом. Мы взяли обычную машину на радиоуправлении, прикрепили к ней видеокамеру с usb свистком, а к пульту управления подключили плату Arduino. Система, получая сигналы о необходимости управления автомобилем включает или выключает реле, что имитирует нажатие клавиш на пульте управления, тем самым машина передвигается под управлением автопилота.

(фото платы)

## **11 Подробнее про модели и декорации**

На верхней части макета Purple City располагаются 5 различных трёхмерных моделей: 2 двухэтажных дома типа «Коттедж», двухэтажная больница, одноэтажное здание администрации и двухэтажное МЧС. Смоделировано всё в программе Autodesk Inventor и напечатано с помощью аддитивных технологий. Для печати был выбран пластик типа PLA, так как он биоразлагаем и не так вреден для окружающей среды. Для обработки макетов мы использовали несколько баллонов серого грунта-эмали и аэрозольную краску белого, синего, красного и зелёного цветов. Для наглядного различия зданий мы нанесли на них небольшие рисунки. Также мы добавили 3D модели из прозрачного пластика, такие как экраны для компьютеров, стулья, рамка для металлоискателя и торшеры. Благодаря им освещение становится мягче и приятнее глазам.

Из декораций на поле Purple City есть 8-образная дорога для машин, небольшой центральный парк и парковка в жилом квартале. Чтобы придать городу интересный и оживлённый вид, мы поместили туда различные фигурки из конструктора LEGO и не только.

## Дополнительные материалы

С дополнительными материалами по проекту вы можете ознакомиться на YouTube:

- Презентация на русском - <https://youtu.be/rF9yMwjOM14>
- Презентация на английском - <https://youtu.be/erjsI0yEnpI>
- Отдельные кадры работы наших систем - <https://youtu.be/5AQ8tA72Yi8>
- Интересные факты о проекте от участников:
  - Полина - <https://youtu.be/7C8pHxt4JV8>
  - Виола - <https://youtu.be/imPox3ciwAQ>
  - Ксения - [https://youtu.be/zWWbH\\_yI6fY](https://youtu.be/zWWbH_yI6fY)

Также вы можете ознакомиться с программными кодами наших систем. Для удобства мы выложили их на GitHub - [https://github.com/daniilkapitonov/purple\\_city\\_code\\_qualification.git](https://github.com/daniilkapitonov/purple_city_code_qualification.git).

## **Приложение А**

