

## Введение

Этот документ является специализированным руководством по проекту 5277(<http://5277.ru>) и предназначен большим образом для администратора решения(интегратора системы автоматизации).

Дополнительную информацию можно найти в похожих документах, например в руководстве для пользователя.

Также стоит обратить внимание на документы с описанием линейки устройств, мы постарались вложить в них богатый функционал и конечно система работает с этими устройствами максимально оптимально.

Далее рассматривается общая концепция, цели, которые мы ставили при реализации данного проекта и его возможности.

Основная часть документа описывает функционал, логику взаимодействия компонентов проекта и примеры его использования.

## Концепция

Существует большое количество различных систем автоматизации, на наш взгляд многие из них имеют существенные недостатки, в некоторых этот недостаток — цена, в других узкая направленность, в третьих — большая зависимость от сторонних узлов, решений или знаний специалиста.

Мы попытались учесть эти недостатки и создать что-то свое, независимое, при этом старались создать решение ориентируясь на поверхностный уровень знаний в области автоматизации.

И так, для создания решения Вам не потребуются какие-то особенные, специфические знания или опыт. Достаточно знать, к примеру, что такое RS485, как подключать такие устройства и какие у них параметры.

Мы старались сделать продукт таким, чтобы выбор модели решения автоматизации зависел от знания и оборудования, которое Вы используете, а не от функционала нашего продукта.

Мы считаем очень важным вопрос универсальности.

Вся структура проекта создавалась таким образом, чтобы можно было использовать различные типы устройств, не зависимо от их интерфейсов, шин и тому подобного. Система легко расширяемая, позволяет добавлять новые устройства, интерфейсы, контроллеры, менять параметры устройств отображаемых в клиенте, языки и многое другое. При этом, зачастую нет необходимости скачивать новую версию и выполнять переустановку ПО.

Также, большая часть ПО написана на Java(клиент использует GluonHQ), что позволяет развернуть решение на большом количестве устройств, в том числе и на мобильных. Вы сами решаете какое железо будете использовать, но не стоит забывать и о надежности.

Облако. Логика в облаке и надежность вещи не сочетаемые. Наше решение работает в нескольких режимах, максимальный функционал доступен с использованием выделенного сервера. Можно обойтись и без сервера, функционала будет меньше, но основные задачи доступны. Также можно использовать только клиент, но его функционал по явным причинам сильно ограничен. При этом, контроллер полностью автономен, для его основной работы сервер не нужен.

Финансовый вопрос. Большая часть решения распространяется бесплатно. У нас есть некоторые ограничения в контроллере и мы не распространяем серверную часть.

Основной статьей нашего дохода мы видим услуги и продажу собственных конечных исполнительных устройств, датчиков, модулей и тому подобное.

Также, мы считаем, очень важен принцип взаимодействия системы с пользователем. Это не аналог SCADA, хотя нет препятствий для создания графического интерфейса выполняющего задачи диспетчерского пульта.

Способы, причины и вид информирования задаются самим интегратором в редакторе сценариев.

При этом при создании системы мы придерживались принципа минимального участия пользователя. Решение не должно быть видно, оно не должно отвлекать пользователя, а информировать только в заранее заданном пользователем условии. Этот принцип особенно виден по разработанным нами устройствам.

## Цели создания проекта

1. Базовое решение для собственных задач по автоматизации различной области.

Мы сталкивались с задачами по автоматизации различного типа. Это была система управления освещением основных магистралей города, сбор данных центральных тепловых пунктов, сбор данных трансформаторных подстанций, пункты весового контроля, система автоматизации для гидропоники и многое другое. И почти везде, по некоторым причинам, разрабатывались решения заново. Также использовались дорогие зарубежные контроллеры, для программирования которых требовался специально обученный инженер. Теперь у нас есть комплексное решение с богатым функционалом, которое легко разворачивается для большинства задач. В том числе легко расширяется при необходимости.

2. Инструмент для сторонних специалистов автоматизации.

Мы предлагаем ознакомиться с нашим решением в надежде, что опытные специалисты данной области его положительно оценят и будут применять также как и мы в своей практике.

3. Практика для студентов, имеющих отношение к автоматизации, электронике и программированию.

Мы используем наше решение также в рамках работы со студентами в центре проектной деятельности ДВФУ.

4. Конструктор для любителей различной домашней автоматизации и электроники.

Существует огромное количество недорогих(Китайских) устройств типа датчиков, исполнительных устройств и т.п., которые можно применить в домашней автоматизации. Но зачастую нет полноценного ПО, которое могло бы этот весь зоопарк объединить в одно решение. К тому же любители схемотехники сами конструируют свои устройства и некоторые из них не готовы писать комплексное и легко настраиваемое решение, зачастую довольствуясь популярными бесплатными решениями с явными проблемами.

В том числе это касается и любителей торговой марки Arduino, наш широкий функционал будет им весьма полезен.

## Компоненты системы

1. Сервер - это облачное решение, предоставляет дополнительный функционал, такой как:
  - регистрация проекта, пользователя, контроллера(в целях обеспечения уникальности)
  - предоставление удаленного доступа клиентам к функционалу контроллера(управление конечными устройствами и т.п.)
  - информирование пользователей, в том числе о недоступности контроллеров
  - обеспечение различных точек подключения (интернет мессенджеры, голосовые помощники, электронная почта и т. п.)
  - сбор и хранение статистики, действий пользователей и прочее.
  - хранение данных о проекте и сценариях.
  - API, позволяет подключать сторонние наработки, типа SCADA к примеру.
  - обновление контроллера и табличных данных клиента.

Одной из полезных особенностей сервера заключается в том, что основные компоненты системы(клиент и контроллер) могут находиться в локальной сети(за NATом), т. е. не иметь внешнего статического IP адреса. Также немаловажен размер передаваемых данных, он значительно меньше чем у многих других систем использующих HTTP к примеру. При этом используемый протокол базируется на TCP и поддерживает шифрование.

2. Контроллер или Java контроллер c5277 - разворачивается на компьютере/мини компьютере/мобильном устройстве обычно непосредственно рядом с конечными устройствами(сервером также поддерживается облачный контроллер). Выполняет сбор данных с конечных устройств, управляет ими, в том числе выполняет сценарии.

Т.е. является основным локальным узлом управления. Не зависит от сервера, но передает на него статистику, различные действия и принимает различные запросы пользователей и изменения проекта.

3. Клиент g5277 - полноценное решение позволяющее как вести проект(создание, редактирование и т.п.) так и взаимодействовать к готовым проектам(управлять устройствами, наблюдать за показаниями и т. п.). Он содержит максимальный функционал предоставляемый нашим решением. Также может выполнять часть функционала контроллера, что позволяет, в некоторых случаях, использовать его без контроллера и сервера.

4. Библиотеки устройств, шлюзов и сервисов(папка или архив sip). Контроллер и клиент используют специально разработанные библиотеки устройств. Эти библиотеки написаны на Java и реализуют специальный интерфейс обеспечивающий полную совместимость с контроллером. По сути в них реализована работа с конечным устройством(формирование данных для опроса и других операций, парсинг ответов). Библиотеки могут быть разработаны сторонним разработчиком, и после нашей инспекции введены в систему для всех пользователей.

В основном библиотеки разрабатываются нами. Мы также можем предоставить услугу по разработке библиотек по предоставленной спецификации.

В будущем также планируется создать функционал и выделить пул идентификаторов для личных библиотек разработчика. Так, чтобы, он мог сам в используемый контроллер прописать свои библиотеки неизвестных нам устройств.

5. Дополнительные утилиты. Также в ходе разработки появляется необходимость в дополнительном ПО. Например, реализованы виджеты на Android устройство и консольная утилита, позволяющая получать показания для решений типа MRTG или CACTI.

## Основные дистрибутивы клиента

Для большинства действий Вам потребуется клиент.

Его можно скачать по адресу <http://5277.ru/distr/client/>.

Нам необходимо выбрать последнюю версию и файл в зависимости от Вашей операционной системы.

- g5277-android-X.XX.apk — для мобильных устройств на базе Android
- g5277-linux-x64-X.XX.deb — установочный пакет для Debian дистрибутивах GNU Linux(x86-64)
- g5277-linux32-X.XX.zip — архив для GNU Linux дистрибутивах(x86-32)
- g5277-linux64-X.XX.zip — архив для GNU Linux дистрибутивах(x86-64)
- g5277-macosx64-X.XX.pkg — установочный пакет для MacOSX(x86-64)
- g5277-macosx64-X.XX.zip — архив для MacOSX(x86-64)
- g5277-nojre-X.XX.zip — архив без Java SDK
- g5277-windows-x64-X.XX.msi — установочный пакет для Windows(x86-64)
- g5277-windows32-X.XX.zip — архив для Windows(x86-32)
- g5277-windows64-X.XX.zip — архив для Windows(x86-64)

Вам необходимо выбрать инсталлятор или установочный пакет согласно Вашей ОС.

Затем запустить ярлык g5277 или файл run.sh/run.cmd в каталоге g5277/bin.

Если такого инсталлятора нет, то Вы можете скачать zip архив согласно Вашей ОС.

Если же нет необходимого zip архива - качайте архив pojre, в нем нет Java машины, ее Вам придется установить самому, необходима будет Java8(JRE) с JFX, можно скачать по ссылке <http://5277.ru/distr/Java/>

## Первый запуск клиента

В данный момент формирование инсталляционных пакетов под различные ОС еще в процессе, есть недоработки.

Пожалуй лучше всего обстоит дело с инсталлятором под Windows x86-64, он формирует ярлык как на рабочем столе так и меню «Пуск», в Debian ярлык доступен в Меню → Утилиты. MacOSx — пакет распаковывается в «Библиотеки», в ланчере ярлыка пока нет.

Поэтому, если находим — запускаем ярлык g5277, либо заходим в установленные программы, находим g5277, и там, в папке bin запускаем run.sh или run.cmd(в зависимости от Вашей ОС)

Вот так выглядит начальная форма клиента:

Мы видим диалог GluonHQ, на котором основан клиент, что позволяет его запускать на различных операционных системах. Фактически мы распространяем клиент бесплатно, но в будущем планируем приобрести лицензию и убрать этот диалог.

Язык приложения зависит от настроек Вашей операционной системы, пока поддерживается только русский и английский.

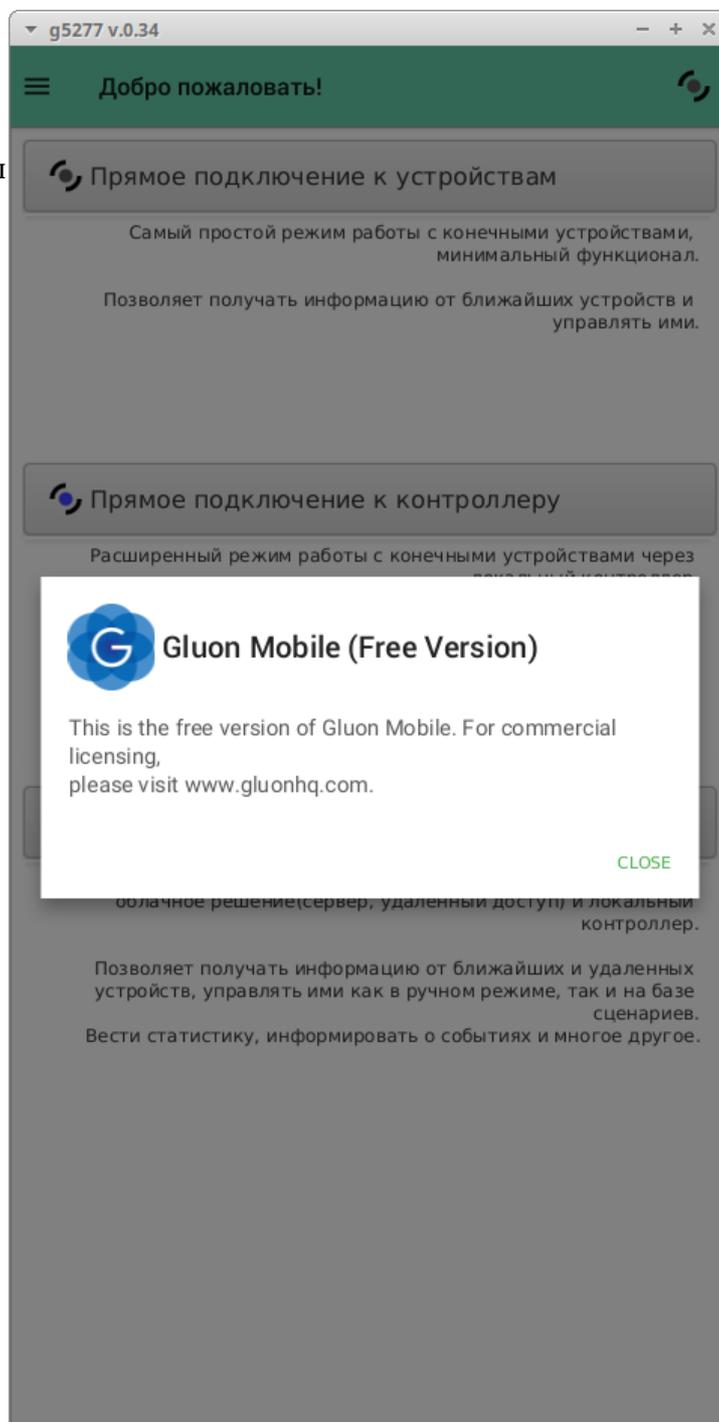
Итак, сейчас мы видим три кнопки с описанием режимов работы клиента. Расположены по мере увеличения функционала.

1. прямое подключение к устройствам - клиент сам, без посредников обращается к устройствам, например к Mi Smart Plug (Wi-Fi), выполняет их опрос и управляет ими. Проект хранится локально, никаких таймеров и сценариев здесь нет. Просто пульт управления и мониторинга.

2. прямое подключение к контроллеру - клиент подключается к контроллеру в пределах локальной сети, контроллер выполняет опрос и запросы от клиента, поддерживает сценарии. При завершении клиента контроллер продолжает работать без изменений. Проект хранится локально на контроллере.

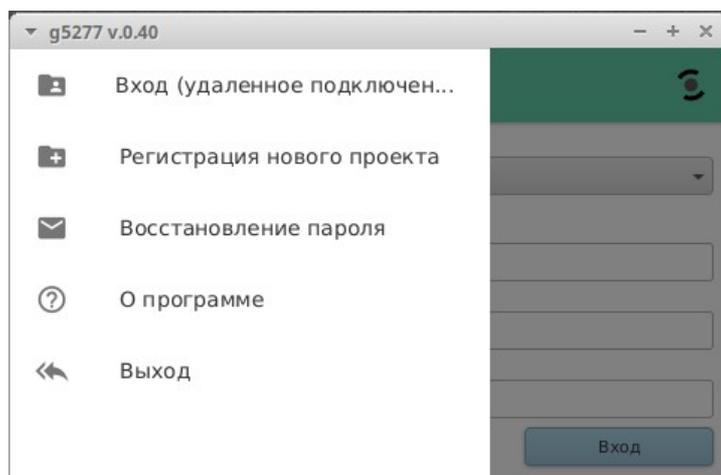
3. удаленное подключение к серверу - клиент подключается к серверу с любой точки мира. Сервер взаимодействует с контроллером, который как и в п.2 ведет опрос устройств и выполняет запросы от клиента. Кроме доступного функционала контроллера клиенту также доступен функционал сервера(статистика и прочее). Проект хранится как на контроллере так и на сервере.

Далее рассмотрим максимально функциональный режим - облако. Таким образом об остальных режимах достаточно будет описать ограничения и особенности.



## Облако - удаленное подключение к серверу

Все пункты, описанные далее находятся в меню, список элементов меню зависит от режима работы и активной формы на экране.



Если это первый запуск клиента и не был выполнен ни один вход, то мы увидим форму регистрации нового проекта. Иначе будет форма входа, при этом если ранее галка «Запомнить» была выбрана, то произойдет автоматический вход.

1. Регистрация - каждый проект привязывается к электронному почтовому адресу. Можно использовать личную почту, но все же лучше создать отдельную почту подходящую под тематику проекта. Главным образом почта нужна для идентификации проекта и информирования администратора проекта.

Площадка - на данный момент в систему заложено 3 площадки:

5277 - авторская площадка для любителей домашней автоматизации и электроники.

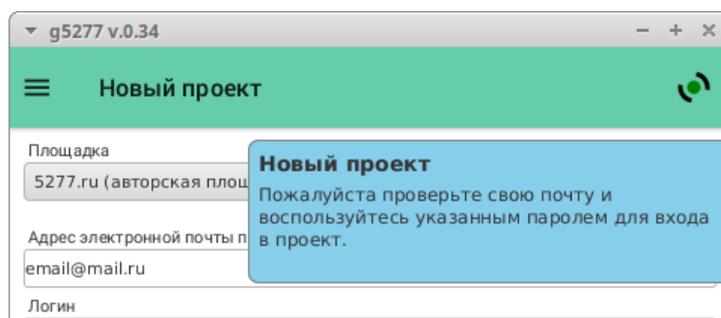
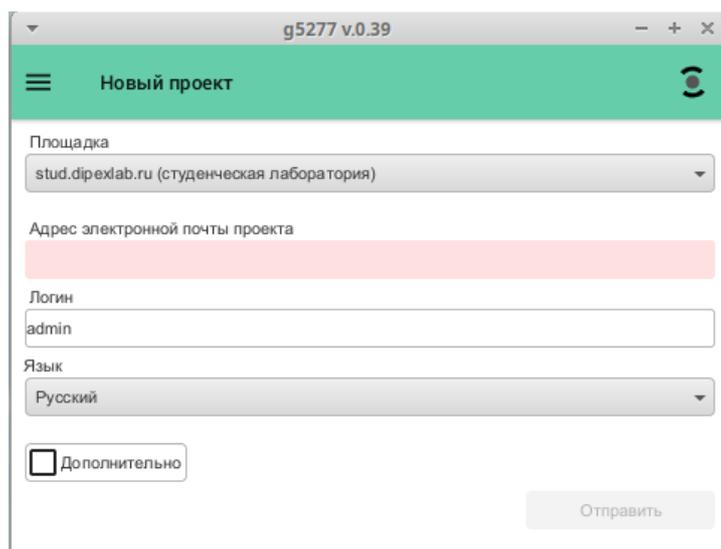
dipexlab.ru - официальная площадка для интеграторов автоматизации (пока не работает по причине отсутствия клиентов)

stud.dipexlab.ru — для студенческих проектов.

Каждая из них это отдельная площадка со своими серверами которые поддерживают и несут ответственность различные компании или люди. Кроме этого, Вы всегда можете сделать экспорт и импорт проекта.

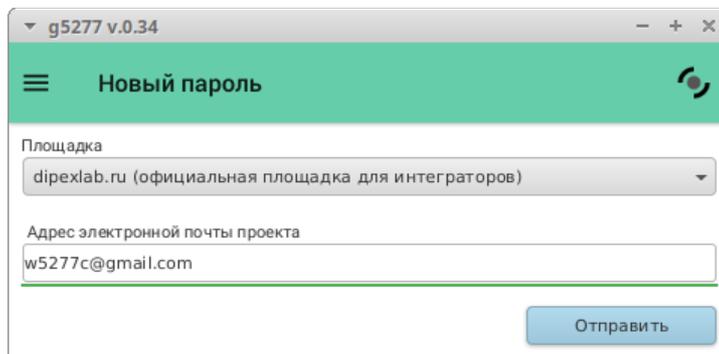
Теперь, Вам остается выбрать необходимую площадку, указать свой или выделенный под проект адрес электронной почты, выбрать язык и нажать кнопку отправить.

Вы увидите сообщение с ошибкой либо сообщение в котором будет сказано, что нужно проверить почту, там будет Ваш первый пароль для входа. Так Вы подтвердите, что адрес электронной почты принадлежит Вам. Пароль после входа можно изменить в профиле пользователя.



2. Восстановление доступа - здесь можно восстановить доступ администратору, если старый пароль утерян. Необходимо указать площадку и адрес электронной почты к которой привязан проект.

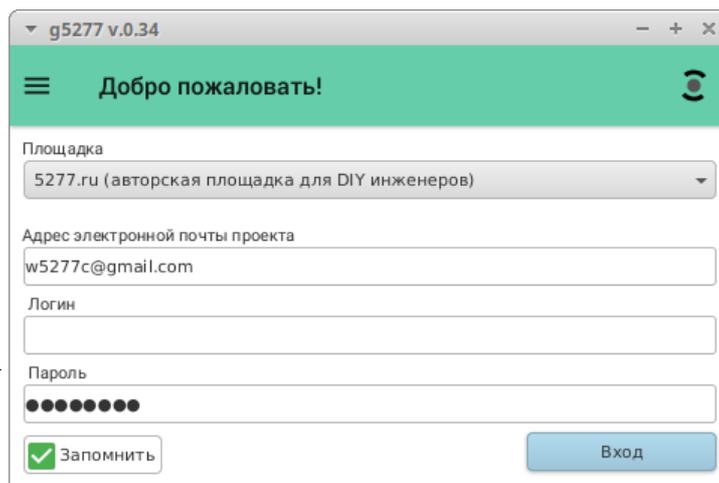
После нажатия кнопки «Отправить» Вы получите сообщение либо с ошибкой, либо сообщение о том, что создан новый пароль и отправлен на почту. Методика аналогичная регистрации. Взяли новый пароль из письма и выполнили вход. После чего пароль можно сменить. Вы может просто игнорировать подобные письма, если этим воспользуется злоумышленник, пароль без входа не меняется и не блокируется.



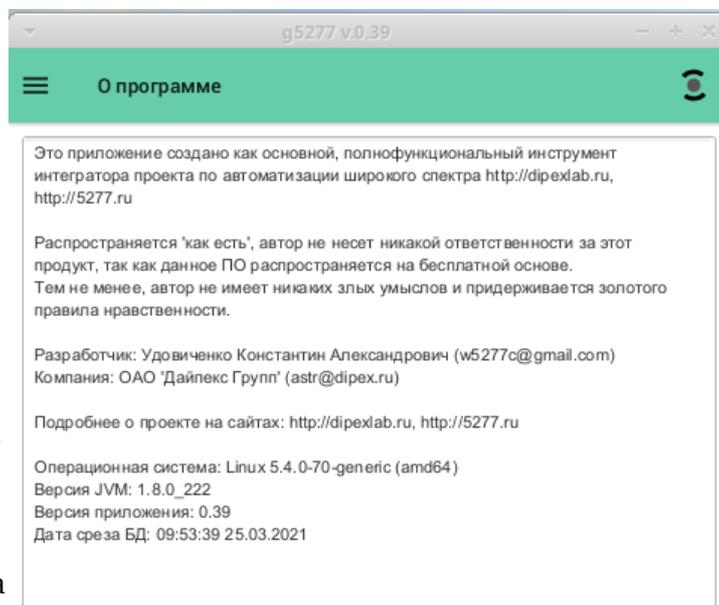
3. Вход — похожая форма на регистрацию. Здесь мы выполняем вход в проект(подключаемся к серверу). Обязательно нужно выбрать площадку, указать адрес электронной почты проекта и пароль.

Администратор поле «Логин» может оставить пустым, но пользователь здесь должен указать свой логин.

Установленная галка «Запомнить» позволяет выполнять следующие входы автоматически, без ввода данных. И наоборот, снятая галка после входа отменит последующий автоматический вход, необходимо будет ввести пароль.



4. О программе — форма, на которой можно почитать общую информацию о программе. В том числе увидеть информацию о операционной системе, версии JVM, версии приложения и дату среда базы данных с информацией о множественных параметрах системы и интерфейса. Эта информация будет крайне полезна если Вы решите обратиться к технической поддержке по обнаруженному сбою.



5. Заголовок формы — верхняя часть формы, имеет следующие элементы:

- кнопку меню, через которое осуществляется большинство переходов по различным формам,
- текстовый заголовок или элемент для выбора текущего помещения
- индикатор сети, показывает текущее состояние сети:
  - серый, нет активного соединения,
  - синий, прямое подключение к контроллеру(к устройству тоже?),
  - зеленый, активное соединение с сервером,
  - красный, сбой соединения.

\* при нажатии на индикатор сети будет переустановлено текущее соединение, а при удержании курсора на нем будет отображена подсказка с информацией о текущем соединении.

## Основная форма

После успешного входа в режиме подключения к серверу мы увидим основную форму для наблюдения и управления устройствами.

Структура проекта, детальнее будет описано позже, состоит из:

- локаций которые лучше всего ассоциируются с географическим объектом в котором все устройства объединены одним контроллером,
- помещением, отдельным объектом объединяющего собой устройства по каким-либо признакам, например тематическим либо физическому расположению.

В качестве примера можно привести квартиру, где локацией выступает сама квартира, а помещениями — ее комнаты. Либо, скажем гараж, где сам гараж выступает локацией а основное помещение и подвал — ее помещениями. Ну или скажем тепловые узлы, где локациями могут выступать города определенного края, а сами географические объекты(тепловые узлы) — помещениями.

В заголовке данной формы доступен выбор помещения. В нем можно выбрать отображение либо всех помещений сразу, либо отобразить одно определенное. Локация выбирается в меню, об этом позже, в описании самого меню.

Далее форма разделена на блоки — помещения, каждое помещение состоит тоже из блоков — конечных устройств.

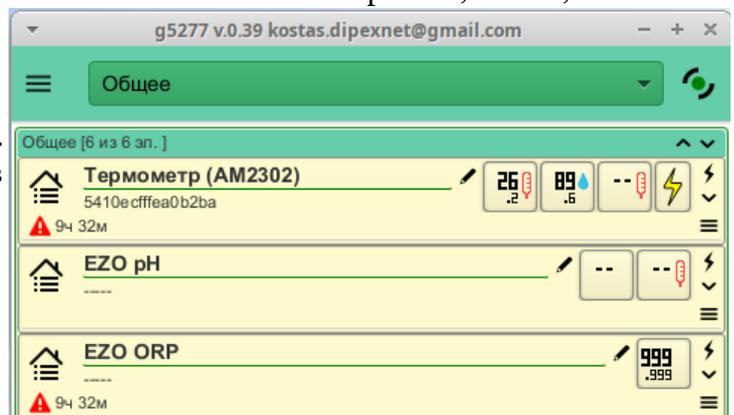
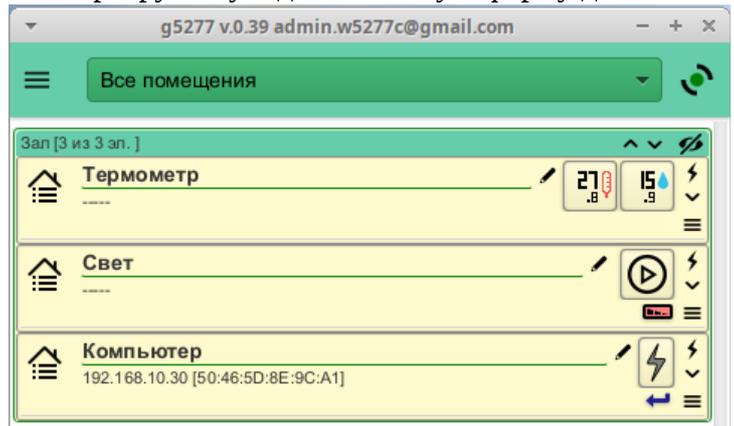
Помещение снабжено заголовком, в котором указано название помещения и количество доступных устройств(устройства, которые в проекте не заблокированы) из количества всех устройств данного помещения. С правой стороны заголовка есть стрелки позволяющие раскрыть или скрыть свойства всех устройств помещения. А также элемент 'глаз' который сворачивает или разворачивает блок помещения.

Также, при наличии специальных элементов группировки, доступен переключатель 'Группировать', который переводит помещение из обычного представления в представление в виде групп устройств. Дополнительно можно почитать описание элемента меню 'Управление группами'.

Далее рассмотрим элемент устройства. Устройства, кроме прочего (более подробно см. в описании конструктора) разделены на области применения, такие как: датчик, управление, приемник, передатчик и тому подобное.

В свернутом виде (без отображения свойств) элемент состоит из пиктограммы, имени, комментария, строки состояния(отображается информация о устаревших показаниях), динамических элементах с правой стороны и трех, вертикально расположенных элементов управления.

Набор динамических элементов отличается в зависимости от области применения устройства и от типа устройства(в котором описаны поддерживаемые показания и элементы управления). К примеру датчикам присущи только верхние элементы - отображают показания. Устройства управления имеют также нижнюю строку элементов для быстрого управления устройством (обычно выведена команда переключения).



Режим работы — системой учтено, что устройства управления могут поддерживать разные режимы управления.

— все запрещено, все источники управления запрещены, устройство не должно изменять свое состояние.

— только триггеры на устройстве, только внутренняя логика устройства может изменить его состояние

— ручное управление, состояние устройства

может быть изменено триггером или кнопкой/сенсором расположенным непосредственно на устройстве,

— удаленное управление, разрешены предыдущие пункты и управление через интерфейс устройства (например через шину),

— полное управление, кроме предыдущих пунктов также разрешено управление контроллером на базе сценариев.

• не все устройства поддерживают данные режимы, в некоторых случаях часть режимов обеспечивает контроллер.

Три вертикальных элемента управления элементом устройства с правой стороны (в порядке сверху вниз):

— подать запрос на обновление показаний (произвести новый опрос устройства),

— развернуть/свернуть свойства устройства,

— показать/спрятать историю действий над устройством.

Свойства устройства, здесь отображается более полная информация.

Мы видим помещение, производителя устройства, тип самого устройства, последнюю дату опроса устройства, его состояние.

Текущий список состояний:

— ожидание данных, отображается при применении проекта, устройство еще не было опрошено,

— сбой инициализации, при попытке опроса контроллер не смог инициализировать библиотеку устройства,

— недоступен, устройство не ответило на запрос,

— ошибка данных, устройство ответило, но формат данных не корректный,

— сетевой сбой, не удалось подключиться к устройству,

— сбой модуля, сбой связанного устройства,

— системный сбой, сбой в логике основной программы или библиотеки устройства,

— лицензионное ограничение, превышено количество разрешенных для опроса устройств, данное устройство не будет опрашиваться,

— ок, успешный опрос устройства.

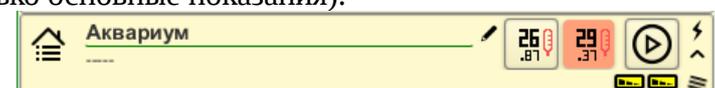
Далее могут быть выведены дополнительные параметры устройства, в зависимости от его типа и настроек проекта или БД, например может быть отображен уникальный идентификатор устройства.

Далее следует блок показаний, в отличие от списка показаний в заголовке элемента устройства, этот список полный (в заголовке отображаются только основные показания).

Показания могут быть выделены красным цветом, данный функционал см. в описании сценариев.



Зал [3 из 3 эл.]	
Термометр	27.6 14.8
Помещение	Зал
Производитель	5277
Тип устройства	Термометр (AM2301)
Дата/время опроса	26.03.2021 15:04:21
Доступность устройства	Ок
Температура	27.60 °C
Отн. влажность	14.80 %

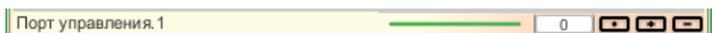


Температура (малый) 29.37 °C

Далее следует блок управления. Он также отличается в зависимости от опций управления устройством, описанных в БД.



– диммирование, бегунком (или числом в поле) можно задать проценты для диммирования (0-100), первая кнопка '\*' задает яркость диммирования на базе числа в поле, следующая '+' увеличивает яркость, следующая '-' уменьшает яркость (на значение, прописанное на конечном устройстве или в его библиотеке),



– розетка(или реле), три кнопки, выключить, переключить и включить. При этом кнопка переключить отображает текущее состояние розетки, красный — выключено, желтый — включено, прозрачный — неизвестное состояние(не получены данные после применения проекта),



– выполнить, действие не имеющее состояния, например, запустить сценарий, включить компьютер, передать текст.



– установить, пока не используется, задумано аналогично 'выполнить' с разницей в том, что 'выполнить' подразумевает какое-то действие, а 'установить' меняет какой-то параметр, которые не обязательно приведет к действию в ближайшее время.

– таймер, кнопка отображения/скрытия



элементов показана на последнем рисунке в виде пиктограммы времени «12:00». При ее нажатии появляются два текстовых поля с элементами выбора единиц измерения времени. В первом блоке указываем время, которое нужно выждать перед началом действия, во втором указываем время паузы между началом действия и выполнением противоположного действия. Начальное действие задается кнопками розетки(реле). Если противоположное действие не нужно, то второй элемент выбора единицы показания оставляем пустым. Таймер может поддерживаться на стороне конечного устройства (в этом случае доступны миллисекунды, иначе функционал таймеров реализует сам контроллер, в этом случае минимальная единица времени — секунда (контроллер не гарантирует точные тайминги при большой нагрузке на шину).

Многие элементы управления поддерживают анимацию (мигание), сигнализирует о процессе выполнения команды. Команда может выполняться долгое время из-за сбоев в работе сети, шины, либо из-за большой очереди заявок и медленной пропускной способности шины или шлюзов через которые проходит заявка. Повторное нажатие на кнопку в данном режиме сгенерирует контркоманду для ее отмены.

История действий, сверху расположены кнопки выбора периода отображения списка действий. Далее идет таблица с действиями. Таблица выглядит интуитивно понятно и думаю не нуждается в описании. Снизу, справа дополнительная кнопка 'x3' она увеличивает/уменьшает высоту блока в 3 раза. Данные таблицы зависят от настроек в проекте и типе действия.

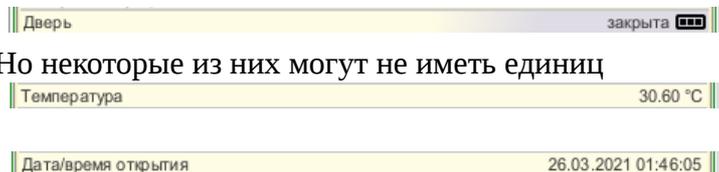
время	канал	действие	инфо
03.25/22:00:00	большой	выключить	сценарий "Выкл свет, аквариумы" (не выполн
03.25/22:00:00	малый	выключить	сценарий "Выкл свет, аквариумы" (не выполн
03.26/10:00:00	большой	включить	сценарий "Вкл свет, аквариумы" (не выполнен
03.26/10:00:00	малый	включить	сценарий "Вкл свет, аквариумы" (не выполнен

Также в верхней части элемента устройства, при наличии доступа, могут быть доступен элемент для редактирования имени и комментария



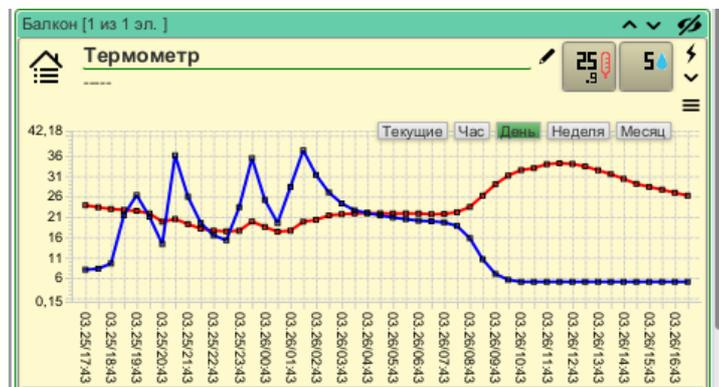
устройства — пиктограмма карандаша в правой части элемента имени. При нажатии поля становятся доступны для редактирования. После редактирования галка выполняет применение изменений, крестик возврат к предыдущим значениям.

Еще о показаниях, стандартно показания выглядят как текст, значение и единица измерения. Но некоторые из них могут не иметь единиц измерения. А некоторые могут иметь двоичное значение, которое отображается в виде пиктограммы (текст значения можно описать в проекте в параметрах устройства). Более того, есть функционал отображения дополнительного значения показаний на базе формулы (также настраивается в проекте, в параметрах устройства). Используется к примеру для датчиков газа, которые отображают значения в условных единицах (значение АЦП от 0 до 1023). На основе данного функционала можно вывести показание в виде миллионных долей (ppm).



Графики, пиктограммы с показаниями в заголовке устройства кликабельны. По нажатию отображается/скрывается график значений выбранного показания. График основан на базовом функционале JavaFX и имеет ряд недостатков, в будущем планируется переход на более функциональную библиотеку. Однако, этого функционала чаще всего достаточно.

В верхней части графиков расположены кнопки выбора периода, аналогично истории действий.



При наведении курсора мыши на поле графиков в верхнем левом углу отображаются данные текущей точки.

Кнопки периодов, аналогично истории действий, анимированы, при ожидании данных мигают. Цвет графика задается в проекте, в свойствах устройства.

Также стоит выделить кнопку периода 'Текущие', в данном режиме не используются архивные данные, отображение строится на получаемых данных в реальном времени. При этом график не статический, т. е. он обновляется при каждом получении данных.

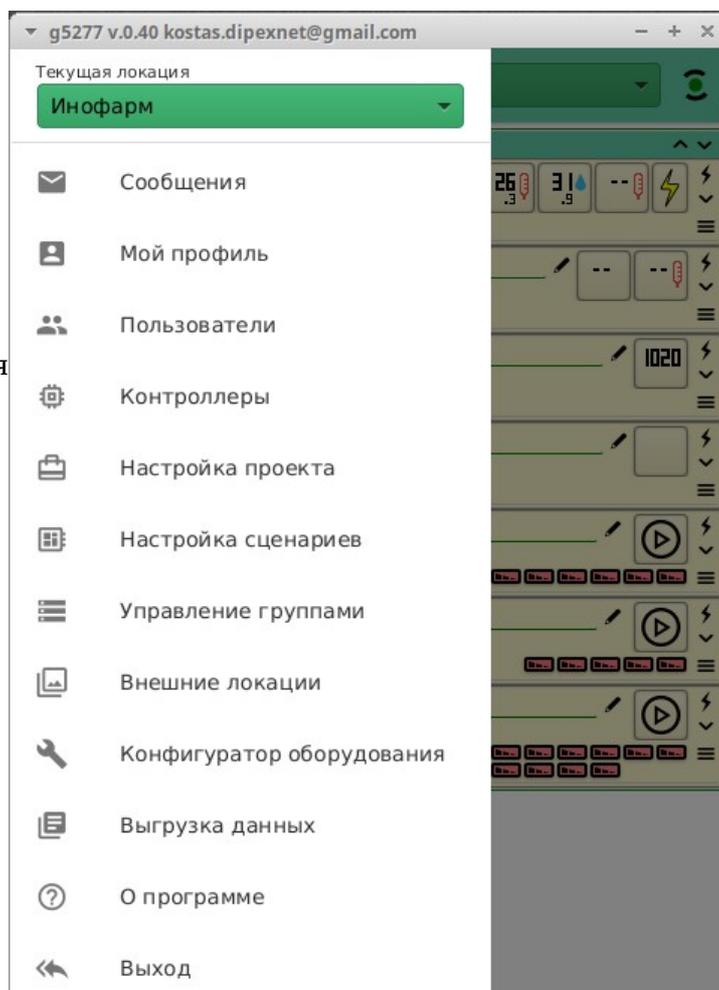
Главный недостаток текущего решения — на графиках не отображаются периоды отсутствия данных, нужно обращать внимание на значения времени в нижней части блока.

## Меню основной формы

Верхним элементом меню является блок выбора локации. Клиент умеет запоминать основные выбранные элементы, поэтому здесь будет локация, которая была выбрана в последний раз (это же касается помещения и других незначительных элементов) либо локация созданная первой.

Элементы меню:

- сообщения, пока еще не доработанная форма, которая позволяет просматривать сообщения пришедшие клиенту от сервера во время работы клиента,
- мой профиль, профиль текущего пользователя (или администратора), аналогичен профилю пользователя.
- пользователи, список пользователей позволяет создавать пользователей и изменять данные профилей.
- контроллеры, список контроллеров проекта, здесь располагаются все инструменты работы с контроллерами.
- настройка проекта(конструктор), в нем выполняется полная настройка проекта.
- настройка сценариев, аналогично, выполняется полная настройка сценариев.
- управление группами, предоставляет функционал для работы с группировкой устройств для визуализации на основной форме.
- внешние локации, позволяет подключать к проекту отображение локаций чужих проектов, сам проект при этом не доступен.
- configurator оборудования, дополнительный инструмент интегратора, предоставляет возможность выполнять конфигурацию устройств и шлюзов.
- выгрузка данных, импорт показаний в документы типа xls и csv.
- о программе, форма описанная ранее, выводит общую информацию о программе.
- выход, выполняет отключение от сервера и переходит на начальную форму.

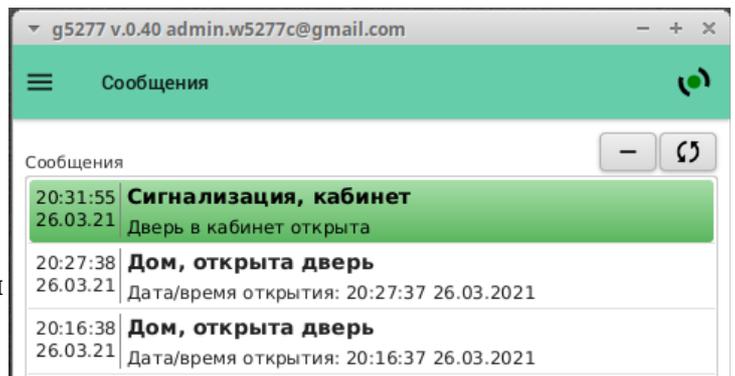


\* пункты меню могут отличаться, пользователю, без прав администратора, часть элементов не доступна.

## Форма сообщений

Форма достаточно проста, отображает сообщения полученные во время работы клиента. Сверху списка доступна кнопка удаления записи и кнопка обновления списка. Сам список автоматически не обновляется.

\*это часть стандартной логики интерфейса, в основном элементы формы сами не обновляются, пользователь сам решает обновлять ему данные или нет, система всего лишь его информирует о поступлении новых данных.



## Мой профиль (профиль пользователя)

Здесь можно посмотреть, изменить данные профиля.

Галка 'Заблокировать' позволяет заблокировать или разблокировать доступ данному пользователю.

Галка 'Администратор' наделяет пользователя максимальными правами, правами администратора.

Поля 'логин' и 'имя' не нуждаются в пояснениях.

Точки управления и информирования (более подробно рассмотрены в отдельной главе). Этот блок позволяет просматривать, создавать и блокировать различные точки управления устройствами в проекте, например 'Telegram' и 'Алиса'.

Здесь стоит обратить внимание на новую кнопку с пиктограммой ключа, она генерирует временный код, который в дальнейшем используется для привязки стороннего сервиса к точке управления.

Язык — здесь Вы можете выбрать язык интерфейса.

Генерация токена — некоторые точки управления требуют токен (уникальный идентификатор пользователя). По сути он является ключом обеспечивающим как идентификацию, так и аутентификацию (за счет своей длины), никому его не сообщайте! Здесь Вы можете его либо создать, либо сгенерировать новый.

Смена пароля — здесь Вы можете сменить свой пароль, для этого необходимо знать текущий. Кнопка 'Сгенерировать' сгенерирует для Вас новый пароль.

The screenshot shows a web browser window with the address bar displaying 'g5277 v.0.40 admin.w5277c@gmail.com'. The page title is 'Профиль'. The interface includes a header with a menu icon and a user profile icon. Below the header, there are two checkboxes: 'Заблокировать' (unchecked) and 'Администратор' (checked). The 'Логин' field contains 'lelik'. The 'Имя' field contains 'Алёна Удовиченко'. There are three icons for editing, deleting, and refreshing. Below this is a section titled 'Точки управления и информирования' with a list of entries:

Имя	Время	Дата	Статус
<b>Alena</b> 464850093	15:41:42	27.05.2019	<b>Telegram</b>
<b>ru.yandex.mobile/440 (Apple iPhone...)</b> 8b247ae8d902b97fa6c3dfff1165f754a12516683c680e651522ec461 440ca4b	15:54:14	27.05.2019	<b>Алиса</b>
<b>HTTP by token</b> b7749c08b1b9a779e0bb0505e3ca171ea9269ce0a63bcae4	13:41:08	10.04.2020	<b>HTTP</b>

Below the list is a 'Язык' dropdown menu set to 'Русский'. There are two sections for token and password management:

- Генерация токена**: A field for 'Ваш текущий пароль' and a 'Пересоздать' button.
- Смена пароля**: Fields for 'Ваш текущий пароль' and 'Новый пароль', and a 'Сгенерировать' button.

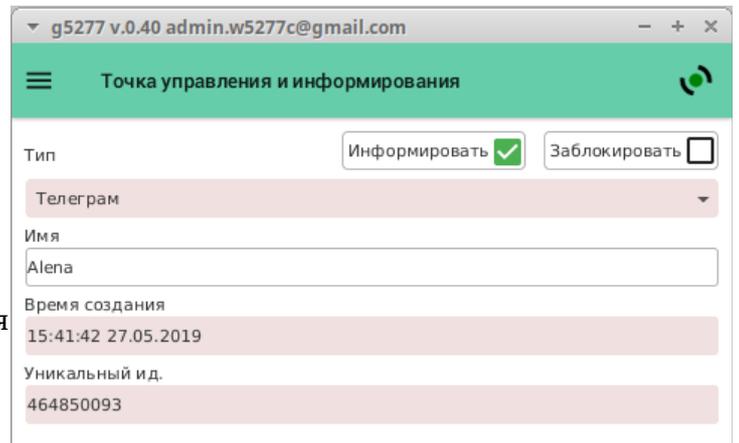
## Точки управления и информирования

Точки управления используются для управления устройствами и информирования пользователя, т. е. создав точку, к примеру 'Телеграм', вы можете управлять устройствами через бота в Telegram, также Вы можете получать сообщения от системы.

Здесь приведена форма одной из точек управления.

Некоторые точки управления используют для идентификации и аутентификации токен, который Вы можете создать на предыдущей форме. Например для точки управления на базе NTTP.

Более подробно читайте в сценариях о информировании и в описании каждой точки управления отдельно.



The screenshot shows a web browser window with the address bar displaying 'g5277 v.0.40 admin.w5277c@gmail.com'. The page title is 'Точка управления и информирования'. The interface includes a green header with a menu icon and a status icon. Below the header, there are two buttons: 'Информировать' (checked) and 'Заблокировать' (unchecked). The main content area displays the following details for a point:

Тип	Телеграм
Имя	Alena
Время создания	15:41:42 27.05.2019
Уникальный ид.	464850093

## Точка управления 'Telegram'

Для активации точки управления 'Telegram' необходимо зайти в профиль пользователя и в блоке 'Точки управления и информирования' нажать на пиктограмму ключа. Получим сообщение с кодом.

Далее работаем в Телеграме. Добавляем бота '5277' введя в поле поиска '@5277', затем пишем в поле сообщения данному боту /start и 'доступ xxxxxx' (вместо x подставляем полученный код).

Если все сделали правильно, то получим сообщение с приветствием в Телеграме и при обновлении блока 'Точки управления и информирования' формы профиля увидим новую запись. При необходимости можно в нее зайти и выставить галку 'Информировать'.

Функционал работы с фразами един для большинства точек управления: 'Telegram', 'Алиса', электронная почта.

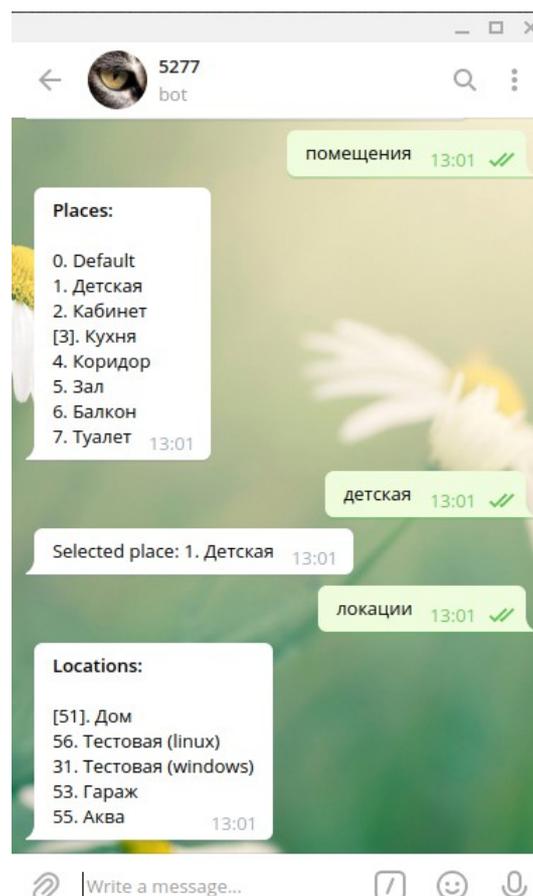
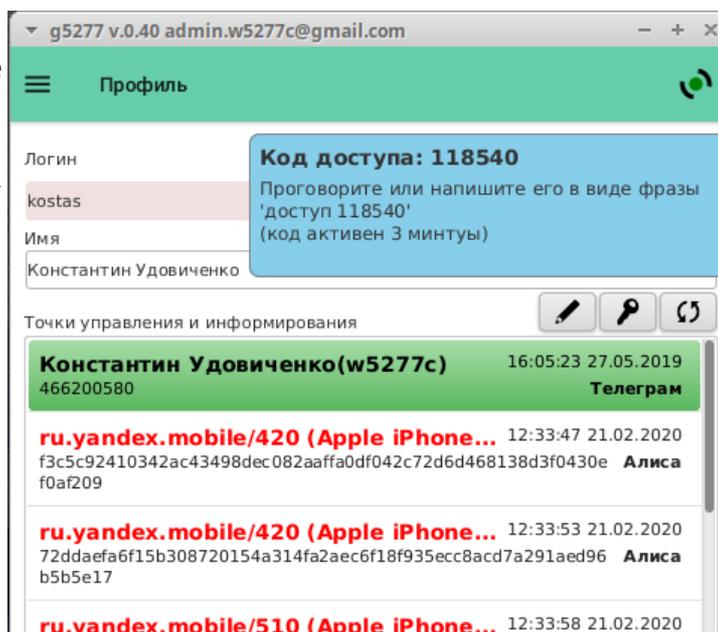
Первое, с чего стоит начать — команда 'помощь'. В ответ получим текст, в котором говорится о основных командах и о дополнительных разделах помощи: 'помощь объекты', 'помощь переход' и 'помощь управление', необходимо внимательно их изучить.

Здесь реализованы функции позволяющие получать информацию о показаниях и состояниях устройств, также можно управлять устройствами (в том числе и по заранее заданным фразам).

Более расширенные функции, такие как управление и изменение сценариев или функционал интегратора здесь не поддерживаются.

Язык интерфейса берется из профиля пользователя. Ввод можно выполнять на любом поддерживаемом языке (на данный момент поддерживается только русский и английский). Выбор локации или помещения также запоминается, текущий элемент в списках выделен квадратными скобками.

Пользуйтесь командой помощь, в ней приведена вся ключевая информация.



## Точка управления 'Алиса'

Эта точка доступа предназначена для голосового помощника 'Алиса' от Яндекса, может быть использована только для управления, информировать Алиса не может.

Название навыка аналогичное — 5277, произносить надо одним числом, иначе Алиса не поймет: 'навык 5277'. Также можно обратиться к навыку в виде 'Попроси 5277 включить свет'.

Есть также дополнительное, шуточное, но легче произносимое имя навыка — 'Тетя Клава', можно ее просить выключить свет.

Работа с навыком ничем не отличается от работы в Телеграм'е, код доступа можно читать блоками по две цифры.

Механизм работы с фразами имеет два набора текстов: для текстового вида, и для произношения.

Для выхода необходимо произнести 'стоп'.

