**Управление по образованию**

**Солигорского райисполкома**

**Государственное Учреждение Образования**

**«Гимназия № 2 Г. Солигорска»**

**«EVOSOL. МОДУЛЬНОСТЬ В ИЗМЕРИТЕЛЬНЫХ ПРИБОРАХ»**

**Автор:** Щуровский Ростислав Юрьевич, XI класс,

Государственное учреждение образования

«Гимназия № 2 г. Солигорска»

**Руководитель:**

Шубин Анатолий Николаевич,

учитель физики

Адрес учреждения образования:

223706Минская обл., г. Солигорск, ул. Богомолова, 22 А,

контактный телефон:

8(0174)263879

e-mail: sol\_gimn2@mail.ru

**Солигорск,2021**

**ОГЛАВЛЕНИЕ**

ВВЕДЕНИЕ……………………………………………………………………...3 - 4

[ТЕОРЕТИЧЕСКАЯ ЧАСТЬ……………………………………………………..5](#_Toc61199927) -8

[ПРАКТИЧЕСКАЯ ЧАСТЬ 9](#_Toc61199928)

[1.Создание люксметра с использованием дисплея. 9](#_Toc61199929)

[2.Создание люксметра с использованием веб-страницы. 9](#_Toc61199930)

[3.Создание люксметра с использованием мобильного приложения. 9](#_Toc61199931)

[4.Создание модульного измерительного прибора. 10](#_Toc61199932)

[5.Мобильное приложение. 10](#_Toc61199933)

[6.Серверная часть: 14](#_Toc61199934)

[7.Сайт проекта: 14](#_Toc61199935)

[ЗАКЛЮЧЕНИЕ 15](#_Toc61199936)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 16](#_Toc61199937)

[ПРИЛОЖЕНИЕ 17](#_Toc61199938)

[1. Компоненты нашего прибора и их характеристики 17](#_Toc61199939)

[2. Внешний вид приборов 18](#_Toc61199940)

[3. Веб-страница прототипа 19](#_Toc61199941)

[4. Arduino IDE – описание 19](#_Toc61199942)

[5. Режимы измерения 20](#_Toc61199943)

[6. SketchUp – описание 20](#_Toc61199944)

[7. ОС для мобильных платформ 20](#_Toc61199945)

[8. IntelliJ IDEA – описание 21](#_Toc61199946)

[9. Языки программирования для написания программ на Android 21](#_Toc61199947)

[10. SQLite – описание 23](#_Toc61199948)

[11. Мобильное приложение 25](#_Toc61199949)

[12. Мобильное приложение – карта 26](#_Toc61199950)

[13. Мобильное приложение – проведения измерения 27](#_Toc61199951)

[14. Мобильное приложение – сохранение данных 27](#_Toc61199952)

[15. Мобильное приложение – чтение данных 28](#_Toc61199953)

[16. Сайт проекта 29](#_Toc61199954)

## ВВЕДЕНИЕ

1. **Актуальность темы**

В данное время наша планета погрязла в электронных отходах. Для наглядности вес электронных устройств, от которых ежегодно отказывается человечество, составляет около 42 млн тонн (на 2014 год). Несмотря на различные попытки правительств стран, а также частных компаний, сократить количество выбросов данного вида отходов, уже в 2019 году (по данным ООН) объём электронных отходов достиг 53,6 млн тонн. Человечество ежегодно производит огромное количество электронных приборов для измерения освещённости, температуры и т.д. Для каждого из которых необходимо создать многочисленные микроконтроллеры, экраны, корпуса и прочие компоненты. Это же влияет и на конечную стоимость продуктов. Поэтому возникла необходимость создать модульный измерительный прибор, который позволил бы измерять различные параметры в зависимости от выбранного модуля.

1. **Цели:**

Создание относительно недорогого прибора, который должен быть модульным и работать в связке со смартфоном пользователя (выполняет функцию, заданную модулем), а также иметь минимальное количество деталей (а значит и небольшие габариты), чтобы отвечать нашим требованиям по экологии.

1. **Для достижения цели были решены следующие задачи:**

* Исследовать возможности одного прибора для измерения многих параметров;
* Испытать прибор в рабочих условиях;
* Собрать данные о качестве реализации продукта;
* Рассчитать экономической эффективности прибора и внедрение в производство;

**4.Гипотеза:**

прибор, созданный нами, будет экономически и экологически выгодным, а также объединять в себе множество функций за счёт модульности и взаимодействия со смартфоном пользователя.

**5.Объект исследования**

Многофункциональный измерительный прибор.

**6.Предмет исследования**

Модульность в измерительных приборах.

**7.Научная новизна**

Создание модульного прибора, который будет обладать более широким спектром возможностей и работать в связке со смартфоном пользователя.

**8.Практическая значимость**

Увеличение доступности приборов с профессиональной точностью, а также уменьшение количества электронных отходов.

**9.Методы научного исследования:**

* + теоретический;
  + экспериментальный;
  + сравнительный.

## 

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

В настоящее время созданы и используются огромное количество различных измерительных приборов: одношкальных, многошкальных, цифровых и др. Многие из них являются устаревшими на сегодняшний день, которые не используют весь потенциал современных технологий.

Идея создания нового прибора – люксметра, появилась при выполнении исследовательской работы по фотометрии. Хотя уже существует прибор-люксметр Ю-116, которым пользуются работники СЭС, возникла необходимость увеличить функциональность и уменьшить стоимость данного прибора. Созданный прототип прибора на основе платы d1 mini с Wi-Fi чипом ESP8266 и датчика BH1750 может успешно взаимодействовать со смартфоном пользователя, что улучшает удобство его использования по сравнению с существующими аналогами. Сама плата в работе использовалась в качестве компьютера (для сбора данных с датчика) и сервера (для передачи данных). Датчиком для нашего люксметра послужил модуль BH1750. Также использовался MicroSD Card Adapter для сохранения и чтения данных. Узнать характеристики данных компонентов можно в Приложении в разделе 1 «Компоненты нашего прибора и их характеристика». А увидеть сам прибор можно в разделе 2 «Внешний вид приборов».

Для работы данного прототипа была написана веб-страница (которая была загружена в память платы), которую можно было увидеть, подключившись к точке доступа, создаваемой люксметром, с любого устройства (имеющего возможность приема данных с помощью сети Wi-Fi) введя в строке браузера IP-адрес (<http://192.168.4.1>/) прибора, а также создана прошивка в Arduino IDE. Скриншоты веб-страницы и описание Arduino IDE вы найдете в Приложении в разделах 3 «Веб-страница прототипа» и 4 «Arduino IDE – описание». Основные идеи уже были реализованы в данном варианте исполнения прибора: управление со смартфона; компактность; сохранение и чтение данных, полученных с люксметра; различные режимы измерения (с режимами измерения можно ознакомиться в Приложении в разделе 5 «Режимы измерения»). Положительной стороной данной реализации прибора является то, что не нужна была предварительная установка приложения. Но отрицательной стороной являлась очень малая скорость передачи данных от прибора к устройству пользователя. Оставлять прибор в таком виде было нельзя. Поэтому решено было отказаться от данного способа передачи данных, оставив его лишь на макетной плате, и попробовать создать полноценное приложение на мобильные платформы. Это дало бы прирост в скорости взаимодействия прибора и пользователя, более красивый графический интерфейс и новые возможности (фактически безграничные). А увидеть сам прибор можно в разделе 2 «Внешний вид приборов».

Но перед этим необходимо было внести изменения в принцип работы самого прибора. Модуль для MicroSD карты было решено убрать, т.к. планировалось, что все операции сохранения и удаления будут проводиться на смартфоне пользователя. Также прибор должен был стать автономным, поэтому в нашем прототипе мы использовали три батарейки типа ААА, которые суммарно нам дали около 4,5 В. По примерным подсчетам это позволяет использовать прибор около 30 часов (при использовании литий-ионного аккумулятора можно добиться более высоких результатов). Для включения/выключения прибора было решено использовать обычный механический ключ (переключатель). Оставлять данный прибор на макетной плате не было смысла (состав элементов данного устройства не будет изменён), поэтому необходимо было создать корпус. Для его создания был использован 3D-редактор SketchUp, подробнее с которым можно ознакомиться в Приложении в разделе 6 «SketchUp – описание». На данный момент имеется только временная версия корпуса (распечатана на 3D-принтере). А увидеть сам прибор можно в разделе 2 «Внешний вид приборов».

Затем необходимо было написать новую прошивку для работы прототипа, т.к. старая выводила веб-страницу и не имела необходимых функций. Поэтому была выбрана новая библиотека, которая все так же позволяла прибору являться сервером, вот только устройство могло уже полноценно «общаться» со смартфоном.

Однако позже идея была расширена. Теперь отсутствует ограничение, в качестве модуля освещённости BH1750. Было решено сконструировать модульную версию прибора. Теперь пользователь сам может решать, какой модуль подключить. А чтобы ему было удобнее, приложение само определит подключаемый модуль.

Теперь же необходимо было создать приложение для смартфона пользователя. На сегодняшний день самыми популярными операционными системами для мобильных устройств являются iOS и Android (более подробную информацию можно найти в Приложении в разделе 7 «ОС для мобильных платформ»). Из-за отсутствия условий для разработки приложения на iOS, было принято решение выбрать Android.

В качестве платформы для разработки нашего приложения был выбран продукт IntelliJ IDEA от JetBrains, который хорошо зарекомендовал себя среди опытных разработчиков. Подробнее с данной IDE можно ознакомиться в Приложении в разделе 8 «IntelliJ IDEA – описание».

В качестве языка программирования IDE предлагает выбрать Java, Kotlin, C++. Кратко ознамиться с каждым из них можно в Приложении в разделе 9 «Языки программирования для написания программ на Android». Выбор был сделан в пользу Java, т.к. именно с него рекомендуют новичкам начинать своё путешествие в мир Android разработки из-за большого количества обучающих материалов.

Затем необходимо было выбрать минимальную версию Android для работы приложения. Решено было остановиться на Android 4.1, т.к. некоторые библиотеки, которые необходимы для функционирования приложения, работают начиная с данной версии «зелёного робота». Стоит также учесть тот факт, что тестирование проводилось на смартфоне с данной версией ОС. Скорость работы была приемлемой, сильных зависаний обнаружено не было. Поэтому, если бы были найдены другие более универсальные библиотеки, скорость работы приложения могла бы быть слишком низкой.

Теперь необходимо было изучить создание программ на Android на языке Java. Для этого использовались уроки, которые создал Google, на англоязычном ресурсе Udacity. Также были использованы уроки YouTube канала Start Android.

Во время разработки самого приложения приходилось сталкиваться с различными задачами и ошибками. На помощь приходили различные форумы, а также developer.android.com. Среди них: overcoder.net, ohandroid.com, cyberforum.ru, stackoverflow.com и многие другие.

Т.к. данный прибор выводит значение люкс в веб-страницу, необходимо её спарсить. Для реализации данной задачи была использована библиотека Jsoup.

Следующей задачей являлась необходимость реализации сохранения и чтения данных. Для её решения была выбрана база данных (БД) SQLite, подробнее с которой можно ознакомиться в Приложении в разделе 10 «SQLite – описание».

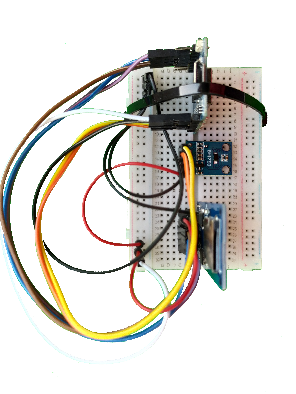
## ПРАКТИЧЕСКАЯ ЧАСТЬ

### C:\Users\User\Desktop\Ростислав\Люксметр\первая_версия_прибора-removebg-preview (1).png**1.Создание люксметра с использованием дисплея.**

**Характеристики:**

Является очень простым в исполнении. Как со стороны кода, так и со стороны компонентов. Не является уникальным, использует лишь более новые компоненты. Подробнее ознакомиться с данным прибором можно в Приложении в разделе 2 «Внешний вид приборов».

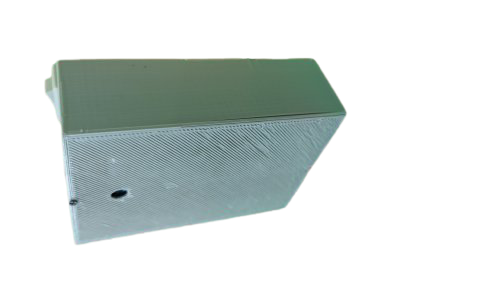
### 2.Создание люксметра с использованием веб-страницы.

****Характеристики:**

Является более сложным по сравнению с люксметром, использовавшим дисплей. Появляется сохранение и чтение данных, а также выбор режимов, которые задают ограничения для получаемых данных и при их превышении нормы сигнализирует об этом, помещая полученные значения в зелёный или красный прямоугольник. Обладает небольшой скоростью работы, а также устаревшим графическим интерфейсом. Подробнее ознакомиться с данным прибором можно в Приложении в разделе 2 «Внешний вид приборов».

### 3.Создание люксметра с использованием мобильного приложения.

**Характеристики:**

****Третье поколение нашего прибора. Создан и распечатан корпус. Добавлены элементы питания, а также механический переключатель. Изменён принцип работы: теперь прибор взаимодействует с приложением на смартфоне пользователя. Для передачи данных используется JSON. Улучшены UX/UI. Добавлена возможность экспорта значений в .csv файл. Улучшена скорость работы. Подробнее ознакомиться с данным прибором можно в Приложении в разделе 2 «Внешний вид приборов».

### 4.Создание модульного измерительного прибора.

**Характеристики:**

Четвёртое поколение нашего прибора. Полностью изменена концепция. Теперь нет ограничений в виде установленного датчика освещённости. Пользователь сам решает, какой датчик ему использовать. Прибор сам понимает, какой датчик к нему подключён. Подключение (крепление) модуля к корпусу осуществляется с помощью магнитов. Подробнее ознакомиться с данным прибором можно в Приложении в разделе 2 «Внешний вид приборов».

### 5.Мобильное приложение.

Перед прочтением необходимо ознакомиться со скриншотами приложения, которые можно найти в Приложении в разделе 11 «Мобильное приложение». Все примеры даны при использовании модуля освещённости на примере режима «Компьютерные классы».

Наше приложение состоит из трёх главных окон:

1. Домашнее окно: главное окно приложения, в котором можно провести измерения с использованием подключенного модуля.
2. Окно настроек: окно приложения, в котором расположены параметры некоторых функций приложения, а также возможно проведение экспорта или удаления всех значений.
3. Окно чтения: окно приложения, в котором можно ознакомиться с сохранёнными значениями, а также изменить или удалить их.

Также возможно использование нашего приложения без прибора для просмотра карты, которая позволяет пользователю увидеть количество человек, пытавшихся посетить определённое место массового скопления людей или торговые (а также иные по назначению) объекты, с температурой, превышающей норму (при условии использования данного приложения и прибора с модулем «Тепловизор» для фиксирования подобных случаев). Это призвано помочь пользователю выбрать наиболее безопасное для посещения место.

Запуск приложения:

Во время запуска приложения пользователь автоматически попадает на экран «Оффлайн», если прибор не обнаруживает подключённый модуль. Поэтому у пользователя появляется возможность повторить проверку на подключение модуля или же продолжить работу в режиме просмотра карты. Также при наличии доступа к сети Интернет все произведенные сохранения, которые не были синхронизированы, будут автоматически отправлены на сервер. А уже с сервера будет получена информация о местах с количеством человек с зафиксированной температурой, превышающей норму.

Просмотр карты:

При выборе пользователем режима работы «Просмотр карты» он перейдёт в окно приложения с картой на которой будут отмечены его текущее местоположение и места с количеством человек с зафиксированной температурой, превышающей норму, которые были получены с сервера и формируются на основании сохранений всех пользователей.

Более подробное описание данного режима работы Вы сможете найти в Приложении в разделе 12 «Мобильное приложение – карта».

Если же прибор обнаружил подключенный модуль, пользователь даже не увидит окно «Оффлайн». Он сразу перейдёт на Домашнее окно.

Проведение измерения:

Перед прочтением необходимо ознакомиться со скриншотами приложения, которые можно найти в Приложении в разделе 13 «Мобильное приложение – проведение измерения».

1. Чтобы провести измерение необходимо нажать кнопку "Провести измерение".
2. Можно наблюдать результат измерения: если показания не соответствуют норме, то количество люкс будет помещено в красный эллипс, как показано на примере слева (также над самим значением появятся нормы). Если же показания будут соответствовать нормам, то они будут помещены в зелёный эллипс, как показано на примере справа (нормы над значением будут сохранены). По умолчанию выбран режим «Стандартный», при котором ограничения отсутствуют. Чтобы его изменить, необходимо перейти в окно настроек.

Сохранение данных:

Перед прочтением необходимо ознакомиться со скриншотами приложения, которые можно найти в Приложении в разделе 14 «Мобильное приложение – сохранение данных».

1. Чтобы провести сохранение, необходимо нажать на кнопку "Сохранить значение".
2. Далее появится диалоговое окно, которое предложит Вам ввести место сохранения (название) и комментарий. Также автоматически приложением будет получена геолокация пользователя (при его согласии).
3. Введя данные, необходимо нажать кнопку "Сохранить" для успешного сохранения либо кнопку "Отмена" для успешной отмены сохранения.

Чтение и изменение данных:

Перед прочтением необходимо ознакомиться со скриншотами приложения, которые можно найти в Приложении в разделе 15 «Мобильное приложение – чтение данных».

1. Необходимо перейти в окно чтения приложения, нажав на кнопку в панели навигации.
2. Чтобы отфильтровать нужные данные (сортировка по режиму работы, при котором они были сохранены) необходимо выбрать в выпадающем списке интересующий режим работы (по умолчанию "Показать всё").

Таким образом можно увидеть нужные данные. Для последующего же их редактирования необходимо выполнить следующие действия:

1. Нажать на нужные данные.
2. Появятся кнопки "Редактировать" и "Удалить", которые отвечают за одноимённые функции.
3. При нажатии на кнопку "Редактировать", появится диалоговое окно, которое предложит изменить выбранные данные. При нажатии на кнопку "Удалить", запись исчезнет, что будет означать её удаление.

### 6.Серверная часть:

Для нормальной работы режима «Карта», а также для идентификации пользователя и распознавания его записей, необходимо было создать серверную часть нашего проекта, под которой подразумевается создание баз данных SQL и PHP кода, который будет обрабатывать POST и GET запросы, поступающие от приложения, на создание/удаление/чтение/изменение БД и обращаться к ней для выполнения данных операций.

### 7.Сайт проекта:

Для Вашего удобства нами также был разработан сайт нашего проекта, расположенный по адресу <https://evosolby.wixsite.com/website>. Больше информации о нём и ссылку в виде QR-кода Вы сможете найти в Приложении в разделе 16 «Сайт проекта».

## 

## ЗАКЛЮЧЕНИЕ

1. Нами был успешно разработан и создан многофункциональный модульный измерительный прибор, который поддерживает работу с различными модулями (освещённости, температуры и т.д.), и мобильное приложение для работы с ним, которое имеет различные режимы работы, основывающиеся на постановлениях Министерства Здравоохранения Республики Беларусь, и поддерживает работу с картой, с помощью которой можно ознакомиться с количеством человек с температурой тела, превышающей норму, пытавшихся посетить выбранны объект, а в будущем увидеть и соблюдение учреждениями предписанных к исполнению норм. Также запущен сайт проекта для более удобного ознакомления конечного потребителя с нашей работой.
2. Наш прибор может заменить неограниченное количество измерительных приборов, необходимых для выполнения различных задач.
3. Для производства прибора требуется меньшее количество компонентов, чем на создание множества измерительных приборов.
4. Наш прибор является ремонтопригодным, поэтому при истечении его жизненного цикла можно заменить те компоненты, которые вышли из строя, тем самым уменьшая количество электронных и пластиковых отходов, что приведёт к улучшению экологической среды.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Build anything on Android :Для разработчиков Андроид [Электронный ресурс]. – 2020. Режим доступа: <https://developer.android.com>.– Дата доступа: 23.09.2020

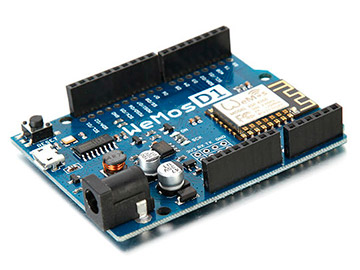
2. We <3 people who code [Электронный ресурс]. – 2020. Режим доступа <https://stackoverflow.com>. – Дата доступа: 03.10.2020

3. Новинки серии накладных регулируемых светодиодных светильников [Электронный ресурс]. – 2020. Режим доступа: <http://sunenergy.by>. – Дата доступа: 25.08.2020

## Приложение

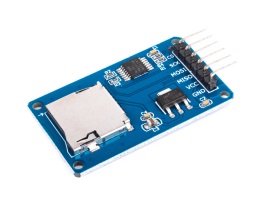
### Компоненты нашего прибора и их характеристики

Плата WeMos D1 mini, которая производится в Китае, выполнена на основе WiFi модуля ESP8266. На модуле имеется разъем под внешнюю WiFi антенну – благодаря этому можно расширить площадь покрытия сетью. Программирование платы осуществляется с помощью стандартной среды разработки Arduino IDE. Контроллер включает в себя процессор, периферию, оперативную память и устройства ввода/вывода.

Технические характеристики WeMos:

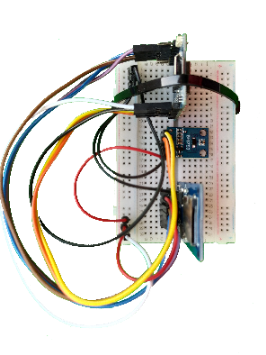
* Входное напряжение 3,3В;
* 11 цифровых выходов;
* Микро USB выход;
* 4 Мб флэш-памяти;
* Наличие WiFi модуля;
* Частота контроллера 80МГц/160МГц;
* Рабочие температуры от -40С до 125С.

Цифровой датчик освещенности GY-302 на чипе BH1750 предназначен для измерения фонового освещения. BH1750 16-битный датчик освещённости (люксметр) с интерфейсом I2C. Эта микросхема хорошо подходит для получения данных об окружающем освещении. Фотодиод на BH1750 определяет интенсивность света, которая преобразуется в выходное напряжение с помощью операционного усилителя. Встроенный АЦП выдает 16-битные цифровые данные. Внутренняя логика BH1750 избавляет от необходимости каких-либо сложных вычислений, поскольку он напрямую выводит значимые цифровые данные в люксах (лк). Согласно документации, датчик BH1750 чувствителен к видимому свету и практически не подвержен влиянию инфракрасного излучения, т.е. реагирует примерно на тот же спектральный диапазон, что и человеческий глаз.

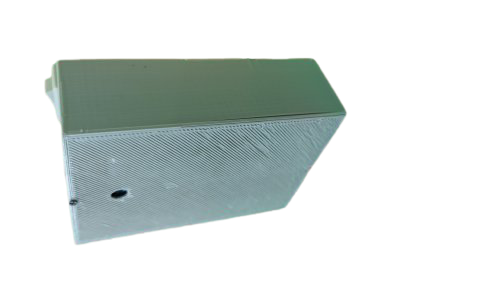
MicroSD Card Reader предназначен для записи и чтения данных с SD карты. Интеграция модуля осуществляется через интерфейс SPI, что позволяет значительно увеличить память прибора на основе микроконтроллера, имеющего SPI интерфейс. Предназначен для использования в устройствах, цифровая часть которых имеет питание 5В. Поддерживает карточки MicroSD объёмом до 32 ГБ.

### C:\Users\User\Desktop\Ростислав\Люксметр\первая_версия_прибора-removebg-preview (1).pngВнешний вид приборов

Первый прототип прибора. Он использует плату d1 mini, BH1750 (датчик) и дисплей для вывода информации. Является достаточно простым в исполнении.

****

Второй прототип прибора. Использует веб-страницу для взаимодействия с пользователем. Состоит из платы, датчика и модуля MicroSd Card Reader.

****

Третий прототип прибора. Состоит из платы и датчика. Также был создан корпус и добавлены элементы питания, а также переключатель. Передаёт полученные данные на смартфон пользователя в приложение, которое было создано.

****Четвёртое поколение нашего прибора. Полностью изменена концепция. Теперь нет ограничений в виде установленного датчика освещённости. Пользователь сам решает, какой датчик ему использовать. Прибор сам понимает, какой датчик к нему подключён. Подключение (крепление) модуля к корпусу осуществляется с помощью магнитов.

### Веб-страница прототипа



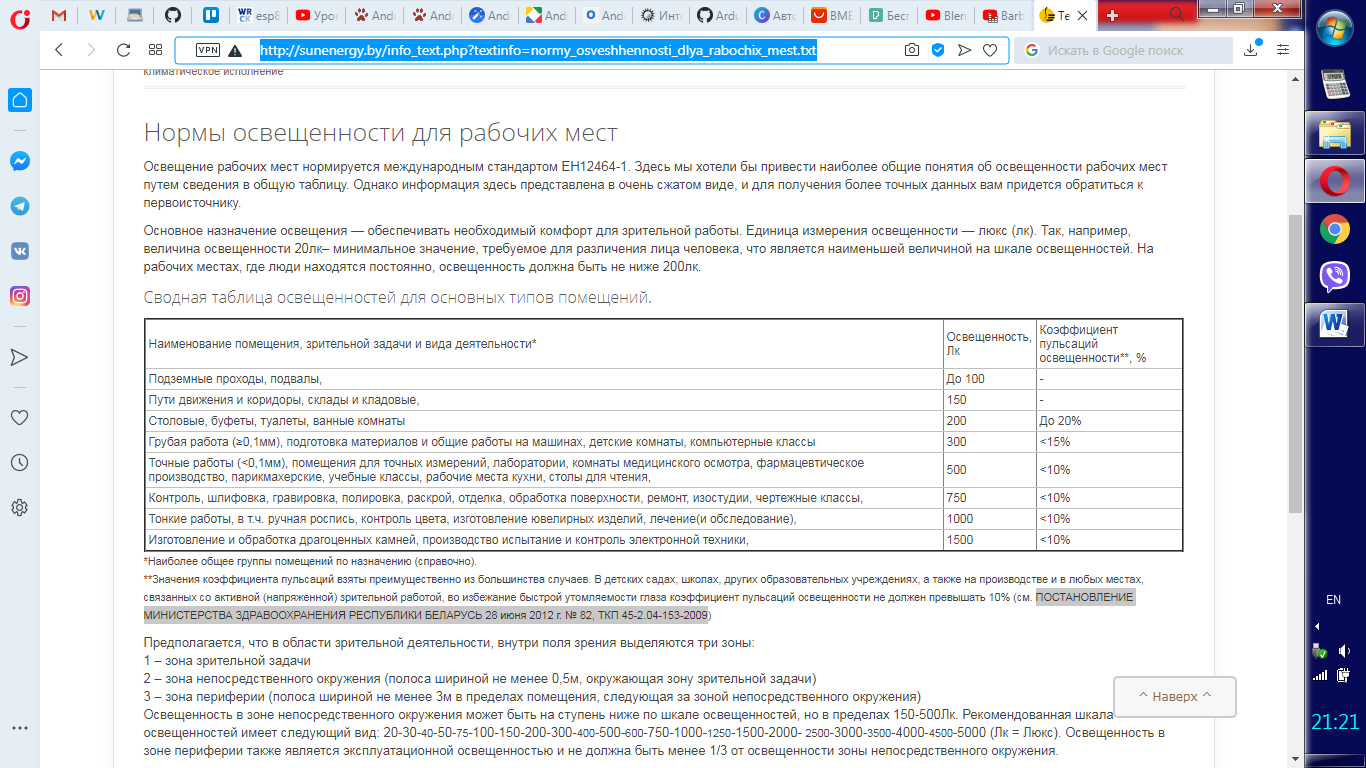
Пример созданной веб-страницы. Всего их было создано несколько. Страница для выбора режимы, считывания и сохранения данных, а также для чтения сохранённых значений. HTML и CSS код помещается в код в среде Arduino IDE в команды вывода клиенту. Основным минусом являлась не только малая скорость работы, но также и неудобная система сохранения.

### Arduino IDE – описание

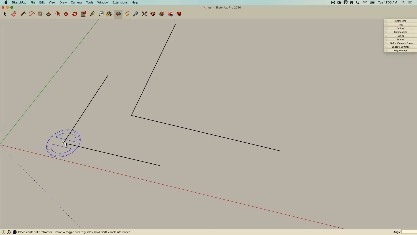
Arduino IDE — интегрированная среда разработки для Windows, MacOS и Linux, разработанная на С и C++, предназначенная для создания и загрузки программ на Arduino-совместимые платы, а также на платы других производителей. На скриншоте представлена веб-версия данной IDE.

### Режимы измерения

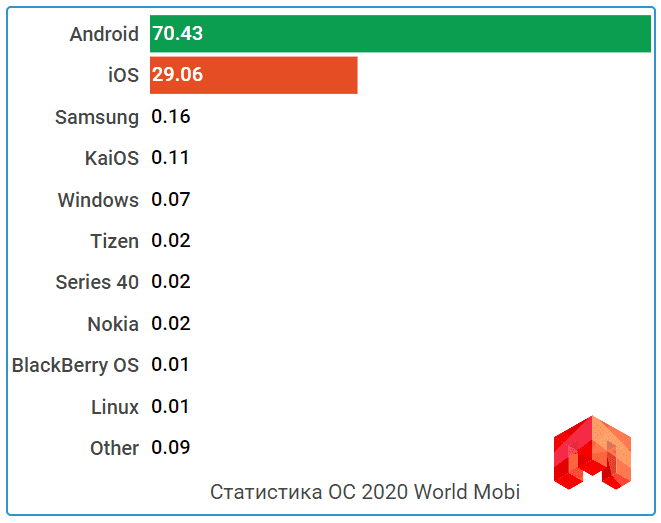
Нормы освещённости были взяты из ПОСТАНОВЛЕНИЯ МИНИСТЕРСТВА ЗДРАВООХРАНЕНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ 28 июня 2012 г. № 82, ТКП 45-2.04-153-2009, размещённого на сайте <http://sunenergy.by/info_text.php?textinfo=normy_osveshhennosti_dlya_rabochix_mest.txt>



### SketchUp – описание

SketchUp (Google SketchUp) - программа для быстрого создания и редактирования трехмерной графики, разрабатываемая компанией @Last Software с 1999 г. и поглощенной в 2006 г. Основной идеей SketchUp является простота интерфейса, что позволяет освоить работу с программой даже непрофессиональному пользователю. Программа реализует концепцию прямого моделирования геометрии, в рамках которой пользователь сначала стоит плоский контур из имеющихся примитивов, затем вытягивает его с целью создания или вычитания объема, после чего придает модели нужную форму посредством перетаскивания ее элементов (вершин, ребер и граней) с помощью указателя мыши.

### ОС для мобильных платформ

Среди ОС для мобильных телефонов лидером остается Android — 70,43%, яблочная мобильная операционная система iOS — 29,06% — практически каждый третий смартфон. Причины популярности Android очевидна: это открытая ОС, а также устройства с данной ОС являются более доступными. Помимо Android и iOS существует ещё несколько ОС. Однако их версии являются довольно старыми, т.к. официально они уже не поддерживаются.

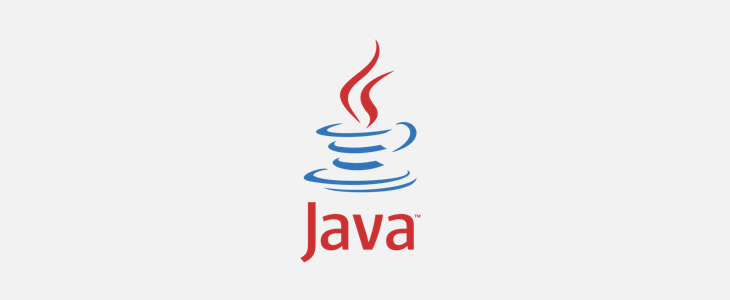
### IntelliJ IDEA: IDE для профессиональной разработки на Java от JetBrainsIntelliJ IDEA – описание

IntelliJ IDEA — интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JavaScript, Python, разработанная компанией JetBrains.

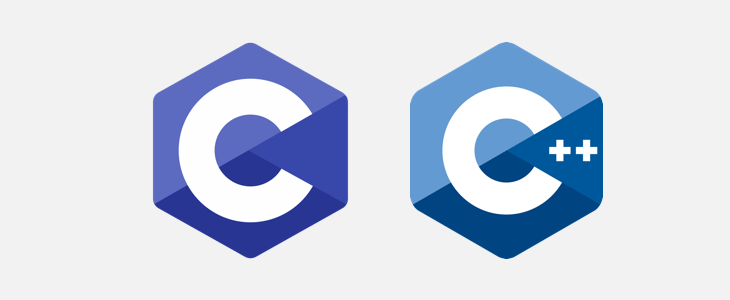
Первая версия появилась в январе 2001 года и быстро приобрела популярность как первая среда для Java с широким набором интегрированных инструментов для рефакторинга, которые позволяли программистам быстро реорганизовывать исходные тексты программ. Дизайн среды ориентирован на продуктивность работы программистов, позволяя сконцентрироваться на функциональных задачах, в то время как IntelliJ IDEA берёт на себя выполнение рутинных операций.

### Языки программирования для написания программ на Android

Официально поддерживающимися Android Studio языками программирования являются Java, Kotlin, C/C++.

**Java** – по данным ежегодного опроса ресурса Stackoverflow, в 2019 году Java вошёл в пятёрку самых популярных языков программирования. При разработке на Java под Android используются не только Java-классы, содержащие код, но также файлы манифеста на языке XML, предоставляющие системе основную информацию о программе, и системы автоматической сборки Gradle, Maven или Ant, команды в которых пишутся на языках Groovy, POM и XML соответственно; по умолчанию в проектах используется Gradle, а на начальных этапах обучения разработке на Java править файлы, написанные на Groovy, практически не придётся. Для вёрстки UI-части обычно также используется язык XML. Тем, кто готов к полному погружению в Android-разработку приложений, Java рекомендуется в первую очередь.

**Kotlin** – язык был официально представлен Android-сообществу в мае 2017 года на Google I/O. Уже в 2019 году Google признали его предпочтительным для разработки приложений на Android, отодвинув Java на второй план. Это в том числе значит, что создание новых стандартных инструментов (библиотек, функций Android Studio) будет в первую очередь нацелено на Kotlin. Kotlin включил в себя лучшее из современных языков программирования, таких как Java, Scala, TypeScript. Среди основных преимуществ — автоматическое выявление типов данных, поддержка функциональной парадигмы, функции-расширения. В то же время язык получил защиту от ряда масштабных проблем, например, «ошибки на миллиард», или NullPointerException. Всё это объясняет его популярность — по данным опроса StackOverflow 2019 года Kotlin вошёл в пятёрку самых любимых сообществом языков. Основой для Kotlin является Java, поэтому переход для бывалых разработчиков облегчён.

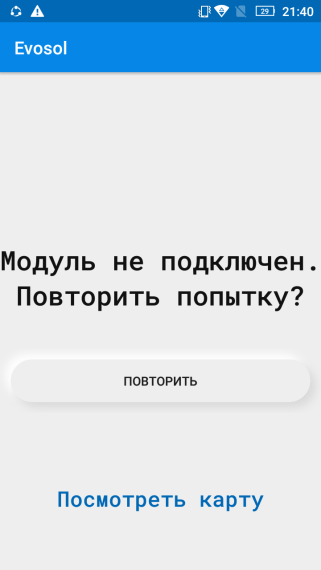
**C/C++** – более низкоуровневые языки, которые также поддерживаются Android Studio с использованием Java NDK. Это позволяет писать нативные приложения, что может пригодиться для создания игр или других ресурсоёмких программ. Android Studio предлагает поддержку C/C++ через Android NDK (Native Development Kit). Это значит, что код будет запускаться не через Java Virtual Machine, а непосредственно через девайс, что даст вам больше контроля над такими элементами системы, как память, сенсоры, жесты и т. д., а также возможность выжать из Android-устройств максимум ресурсов. Это также означает, что пользоваться вам придётся библиотеками, написанными на C или C++. В свою очередь, он сложен в настройке и не слишком удобен, поэтому рекомендуется использовать его для написания только тех модулей программы, где необходимо быстро производить сложные операции: обработку и рендеринг графики, видео и сложных 3D-моделей.

### SQLite – описание

SQLite — компактная встраиваемая СУБД. Исходный код библиотеки передан в общественное достояние. В 2005 году проект получил награду Google-O’Reilly Open Source Awards. Слово «встраиваемый» (embedded) означает, что SQLite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а представляет собой библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором исполняется программа. Простота реализации достигается за счёт того, что перед началом исполнения транзакции записи весь файл, хранящий базу данных, блокируется; ACID-функции достигаются в том числе за счёт создания файла журнала.

### Мобильное приложение

Все примеры даны при использовании модуля освещённости (Люксметр).

**

*Окно «Оффлайн»*

Его отображение озночает отсутствие подключенного модуля к прибору. Пользователь может повторить процедуру проверки или перейти в режим работы с картой.



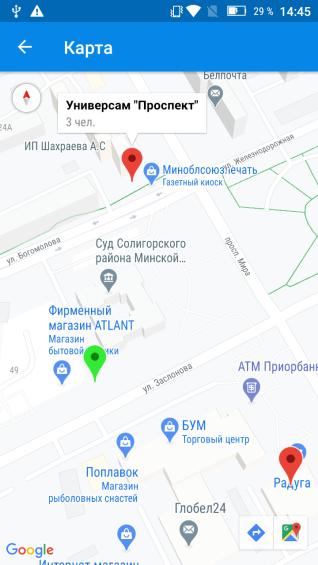
*Окно чтения*

в данном окне приложения можно ознакомиться с сохранёнными значениями, а также изменить или удалить их.

*Домашнее окно* здесь можно провести измерения с использованием подключенного модуля.

*Окно настроек*

в настройках можно изменить параметры некоторых функций приложения, а также произвести экспорт всех значений или удалить их.



*Карта*

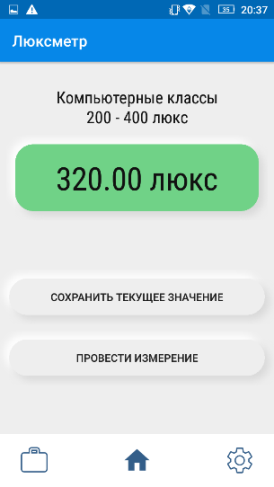
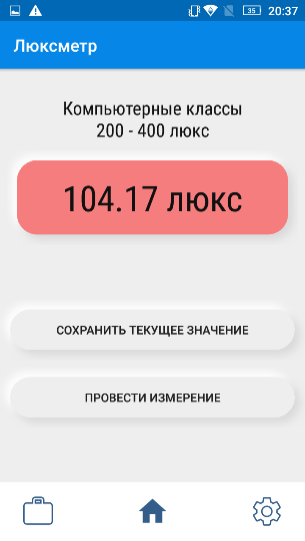
здесь пользователь может ознакомиться с количеством человек, пытавшихся посетить определённое место массового скопления людей или торговые (а также иные по назначению) объекты, с температурой, превышающей норму (при условии использования данного приложения и прибора с модулем «Тепловизор» для фиксирования подобных случаев).

### Мобильное приложение – карта

Для просмотра карты пользователю необходимо отключить смартфон от прибора и подключить его к сети Интернет, чтобы он мог синхронизировать информацию. Таким образом, приложение автоматически отправит все сохранённые значения на сервер, а от него получит информацию о местах с количеством человек с зафиксированной температурой, превышающей норму.

Далее необходимо перейти из окна «Оффлайн» в окно с просмотром карты, нажав на кнопку «Посмотреть карту». Далее приложение автоматически получит текущую геолокацию пользователя (при его согласии) и добавит маркер с текстом «Я здесь», показывающий текущее местоположение пользователя. Далее появятся все места с информацией о количесте человек с температурой, превышающей температурой. Название данных мест формируется на основе данных, вносимых пользователями.

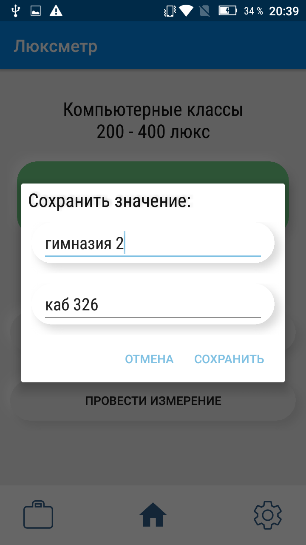
### Мобильное приложение – проведения измерения

 Все примеры даны при использовании модуля освещённости в режиме «Компьютерные классы».

Чтобы провести измерение пользователю необходимо нажать кнопку "Провести измерение". Он сможет наблюдать результат измерения: если показания не соответствуют норме, то количество люкс будет помещено в красный эллипс, как показано на примере слева (также над самим значением появятся нормы). Если же показания будут соответствовать нормам, то они будут помещены в зелёный эллипс, как показано на примере справа (нормы над значением будут сохранены). По умолчанию выбран режим «Стандартный», при котором ограничения отсутствуют. Чтобы его изменить, необходимо перейти в окно настроек.

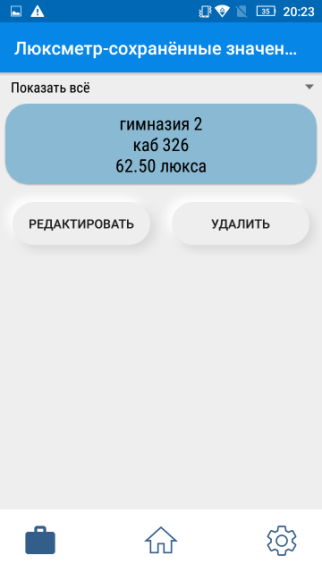
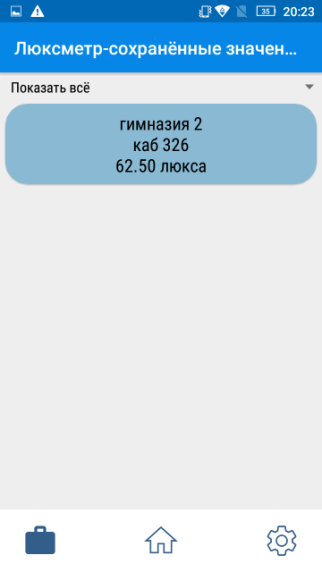
### Мобильное приложение – сохранение данных

Все примеры даны при использовании модуля освещённости в режиме «Компьютерные классы».

Чтобы провести сохранение, необходимо нажать на кнопку "Сохранить значение". Далее перед пользователем появится диалоговое окно, которое предложит ему ввести место сохранения (название) и комментарий к нему. Затем автоматически будет получена приложением текущая геолокация пользователя (при его согласии). Введя данные, пользователю необходимо нажать кнопку "Сохранить" для успешного сохранения либо кнопку "Отмена" для успешной отмены сохранения.

### Мобильное приложение – чтение данных

Все примеры даны при использовании модуля освещённости в режиме «Компьютерные классы».

Пользователю необходимо перейти в окно чтения приложения, нажав на кнопку в панели навигации. Чтобы отфильтровать нужные ему данные (сортировка по режиму работы, при котором они были сохранены) необходимо выбрать в выпадающем списке интересующий его режим работы (по умолчанию "Показать всё"). Таким образом получится увидеть нужные данные. Для последующего же их редактирования необходимо выполнить следующие действия: Нажать на нужные данные. Перед пользователем появятся кнопки "Редактировать" и "Удалить", которые отвечают за одноимённые функции.

При нажатии на кнопку "Редактировать", появится диалоговое окно, которое предложит изменить выбранные данные. При нажатии на кнопку "Удалить", запись исчезнет, что будет означать её удаление.

### http://qrcoder.ru/code/?https%3A%2F%2Fevosolby.wixsite.com%2Fwebsite&4&0Сайт проекта

Нами также был разработан сайт по адресу <https://evosolby.wixsite.com/website> , где Вы сможете в кратком виде ознакомиться с проектом и следить за ним в дальнейшем. Для Вашего удобства мы создали QR-код с ссылкой на него.