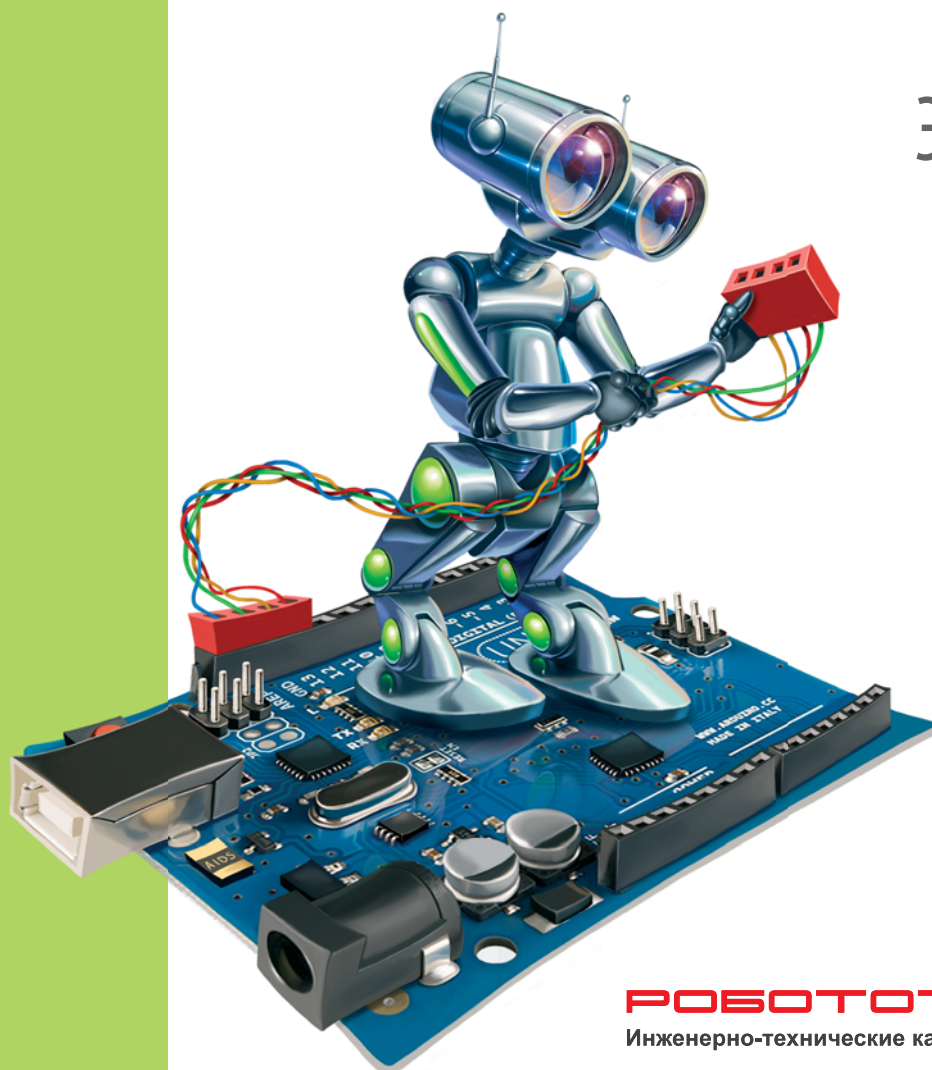


Р • О • Б • О • Ф • И • Ш • К • И

КОНСТРУИРУЕМ РОБОТОВ

на **Arduino®**

Экостанция



РОБОТОТЕХНИКА

Инженерно-технические кадры инновационной России



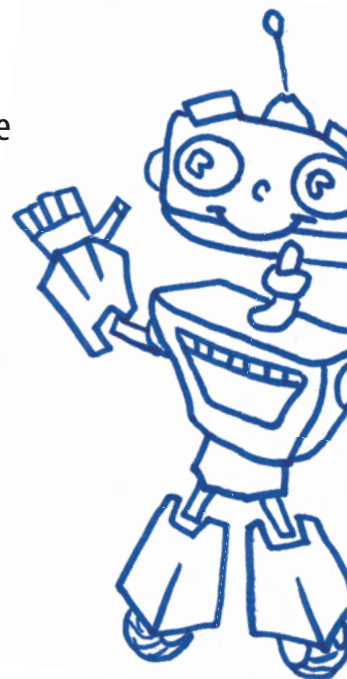
А. А. Салахова

КОНСТРУИРУЕМ РОБОТОВ

на **Arduino**[®]

Экостанция

Электронное
издание



Лаборатория знаний
Москва
2018



УДК 373.167
ББК 32.97
С16

Серия основана в 2016 г.

Ведущие редакторы серии *Т. Г. Хохлова, Ю. А. Серова*

Салахова А. А.

С16 Конструируем роботов на Arduino®. Экостанция [Электронный ресурс] / А. А. Салахова. — Эл. изд. — Электрон. текстовые дан. (1 файл pdf : 67 с.). — М. : Лаборатория знаний, 2018. — (РОБОФИШКИ). — Систем. требования: Adobe Reader XI ; экран 10".

ISBN 978-5-00101-584-0

Стать гениальным изобретателем легко! Серия книг «РОБО-ФИШКИ» поможет вам создавать роботов, учиться и играть вместе с ними.

Вы соберете на платформе Arduino настоящую компактную переносную экостанцию, позволяющую контролировать в помещении качество воздуха, температуру, освещенность и другие параметры.

Для технического творчества в школе и дома, а также на занятиях в робототехнических кружках.

**УДК 373.167
ББК 32.97**

Деривативное электронное издание на основе печатного аналога: Конструируем роботов на Arduino®. Экостанция / А. А. Салахова. — М. : Лаборатория знаний, 2018. — 64 с. : ил. — (РОБОФИШКИ). — ISBN 978-5-00101-111-8.

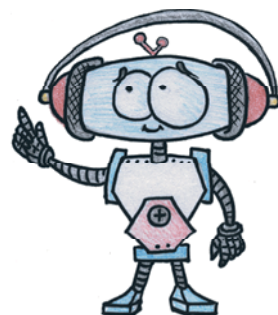
6+

В соответствии со ст. 1299 и 1301 ГК РФ при устранении ограничений, установленных техническими средствами защиты авторских прав, правообладатель вправе требовать от нарушителя возмещения убытков или выплаты компенсации

ISBN 978-5-00101-584-0

© Лаборатория знаний, 2018

Здравствуйте!



Издание, которое вы держите сейчас в руках, — это не просто описание и практическое руководство по выполнению конкретного увлекательного проекта по робототехнике. И то, что в результате вы самостоятельно сумеете собрать своими руками настоящее работающее устройство, — конечно, победа и успех!

Но главное — вы поймёте, что такие ценные качества характера, как терпение, аккуратность, настойчивость и творческая мысль, проявленные при работе над проектом, останутся с вами навсегда, помогут уверенно создавать своё будущее, стать реально успешным человеком, независимо от того, с какой профессией свяжете жизнь.

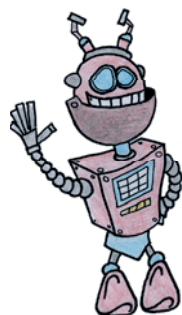
Создавать будущее — сложная и ответственная задача. Каждый день становится открытием, если он приносит новые знания, которые затем могут быть превращены в проекты. Особенно это важно для тех, кто выбрал дорогу инженера и технического специалиста. Знания — это база, которая становится основой для свершений.

Однако технический прогресс зависит не только от знаний, но и от смелости создавать новое. Всё, что нас окружает сегодня, придумано инженерами. Их любопытство, желание узнавать неизведанное и конструировать то, чего никто до них не делал, и создают окружающий мир. Именно от таких людей зависит, каким будет наш завтрашний день. Только идеи, основанные на творческом подходе, прочных знаниях и постоянном стремлении к новаторству, заставляют мир двигаться вперёд.

И сегодня, выполнив этот проект и перейдя к следующим, вы делаете очередной шаг по этой дороге.

Успехов вам!

*Команда Программы «Робототехника:
инженерно-технические кадры инновационной России»
Фонда Олега Дерипаска «Вольное Дело»*



Дорогой друг!

Если ты добрался до платформы Arduino, значит, ты настоящий будущий инженер! Ты прошёл большой путь в робототехнике и решил перейти на новый уровень — роботов на Arduino! Теперь всё будет совершенно серьёзно! Тайны настоящего роботоконструирования ждут именно тебя!

Надоело, что проекты по робототехнике напоминают игрушки? Сделай по-настоящему полезное устройство! А что принесёт больше пользы, чем забота о здоровье? Большую часть времени ты проводишь внутри различных помещений: дома, в школе, центре творчества, технопарке, в гостях у друзей. Их стены и крыши спасают тебя от холода или жары, дождя или снега. Но знаешь ли ты, что внутри них тоже существуют собственная погода и микроклимат? Они оказывают сильное влияние на твоё здоровье. Что ж, пришло время изучить, что под этим подразумевается и как факторы «домашней погоды» связаны с твоим самочувствием. С помощью Arduino ты соберёшь и запрограммируешь настоящую компактную переносную экостанцию, позволяющую взять под контроль качество воздуха, температуру, освещённость и другие не менее важные показатели. Результаты измерений будут передаваться в реальном времени на экран твоего смартфона или планшета, где ты сможешь их сохранить в файл и составить подробный отчёт настоящего эколога!

Экология жилища



В эфире телеканалов, на различных мероприятиях и на уроках в школе ты часто можешь слышать о важности экологии и благоприятной экологической обстановки. Прежде чем думать об обстановке и решать, где же она должна быть благоприятной (где-то снаружи или в доме), необходимо разобраться, что же это за наука — экология. Экология — наука о взаимодействиях живых организмов и их сообществ между собой и с окружающей средой. Под живыми организмами чаще понимают сообщества людей, однако не стоит забывать и о домашних животных, а также микробах и прочих существах, которые, хотим мы этого или нет, проживают вместе с нами в домах и на всей планете. Слово «экология» произошло от древнегреческого οἶκος — «жилище» или «дом» и λόγος — «учение», «наука». Дословно название науки можно перевести как «учение о доме». Конечно же, под домом понимается вся Земля. Но всё большое начинается с малого!

Большую часть времени мы проводим внутри помещений, которые защищают нас от пагубного влияния окружающей среды. Существует высказывание, что одежда — наша вторая кожа, а дом — третья. Но, как и наш собственный кожный покров, наружные стены и крыши жилищ подвергаются влиянию ветра, дождя, солнца. Все неблагоприятные воздействия, которые оказываются на помещение извне, при отсутствии должного ухода и несоблюдении определённых правил влияют на человека, многократно усиливаясь. Тогда говорят о неблагоприятных факторах жилища, или факторах риска.

На качество среды внутри дома влияют:



Современная домашняя метеостанция

Кстати! Несмотря на использование древнего языка, термину «экология» всего полтора века. Он был предложен немецким естествоиспытателем и философом Эрнстом Геккелем в труде «Общая морфология организмов» (1866 г.).

- состояние воздуха снаружи;
- вещества, возникающие при приготовлении пищи (жир, испарения, гарь и т. д.);
- продукты неполного сгорания газа (например, сажа);
- электромагнитное загрязнение (в том числе от работающих бытовых приборов);
- различная пыль;
- наличие комнатных растений;
- проведение регулярной влажной и сухой уборки;
- испарения от бытовой химии и средств гигиены;
- продукты табакокурения (курение вредно и для окружающих!);
- наличие домашних животных и уход за ними;
- интенсивное отопление


и другие факторы.

Среди важных следует также отметить те, что обычно применимы в повседневной речи к состоянию окружающей среды. Ты помнишь слова из известной песни: «Важней всего — погода в доме»? Погода в доме — это комплекс метеорологических условий внутри здания. К ним относятся температура воздуха и окружающих поверхностей, относительная влажность, воздухообмен, скорость движения воздуха, содержание в воздухе пыли (твёрдых частиц) и другие. Вместе они называются **микроклиматом** помещений. Он регламентируется в стандартах и утверждённых санитарных нормах, специальных для производственных (например, российский ГОСТ 12.1.005 «Общие санитарно-гигиенические требования к воздуху рабочей зоны»), общественных и жилых помещений (межгосударственный ГОСТ 30494-2011 «Здания жилые и общественные. Параметры микроклимата в помещениях»). Оптимальные (рекомендованные) показатели зависят не только от назначения помещения, но и от времени года и географического расположения здания. Факторы, которые влияют на микроклимат, образуют две большие категории: нерегулируемые и регулируемые. К первой категории относятся факторы, обусловленные особенностями климата данной местно-

сти, например длительные холодные полярные ночи. Регулируемые факторы — это особенности строительства, интенсивность теплового излучения от приборов отопления, воздухообмен, повышенная влажность из-за просушки белья в доме и прочие. Нарушение норм приводит к ухудшению самочувствия, напряжению, усталости, раздражительности, вялости и даже серьёзным заболеваниям.

Кстати! В начале 1970-х годов немецкие учёные в ходе исследований обнаружили, что в 90% обследованных домов микроклимат выходил за пределы нормы! Одной из причин оказалось применение дешёвых строительных материалов. Они не сохраняли тепло и способствовали запылению, накоплению излишней влаги.

Сохранить здоровье поможет отслеживание показателей составляющих микроклимата и влияние на регулируемые факторы. Так, например, в холодное время температуру воздуха можно повысить с помощью систем отопления (централизованное водяное, печи, электрообогреватели), а влажность — уменьшить с помощью проветривания или увеличить с помощью специальных приборов.


 Итак, давай разберёмся, какие показатели должны быть, чтобы чувствовать себя дома здоровым. Как уже было сказано, ГОСТ регламентирует отдельные нормы для холодного и тёплого времени года. Холодным называется период года, характеризующийся среднесуточной температурой окружающей среды на отметке ниже 8° по Цельсию. Тёплым же временем — с температурой выше или равной 8° по Цельсию.

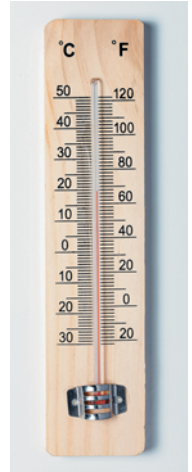
В следующей таблице представлены оптимальные значения температуры в жилых и общественных помещениях согласно ГОСТ 30494-2011 «Здания жилые и общественные. Параметры микроклимата в помещениях». Для измерения используют специальный прибор — термометр, который может быть как электрическим, так и спиртовым.

Помещение	Холодное время года, °С	Тёплое время года, °С
Жилая комната	20–22	22–25
Детская комната	23–24	
Комната для учёбы, умственного труда	20–22	
Ванная, совмещённый санузел	24–26	
Кладовые	16–18	

Температура в комнате младенца должна быть в пределах 18–21 °С. Это связано с тем, что малыши тратят много энергии, когда двигаются, у них теплоотдача намного выше, чем у взрослых. Для пожилых людей ситуация обратная: им комфортно при 24–27 °С.

Температура — самый понятный и легко регулируемый фактор. Кроме перечисленных показаний, следует соблюдать лишь одно правило: разница температур по горизонтали от окон до противоположной стены не должна превышать 2 °С, а от потолка до пола 1 °С на 1 метр высоты.

 Намного сложнее дело обстоит с **влажностью** воздуха. Её измеряют с помощью **гигрометра** (от греч. слов υγρός — влажный



Домашний термометр

и метрѐв — измеряю). Оптимальный показатель влажности для холодного времени года — 30–45%, для тёплого времени года — 40–60%. Однако процент влажности зависит и от температуры в доме: чем жарче в помещении, тем он ниже. Изменение уровня влажности сильно влияет на наше здоровье. Если дома воздух сухой, то иссушаются слизистые оболочки носоглотки, глаз, что приводит к неприятным ощущениям, покраснению глаз, снижению иммунитета, утомляемости. Казалось бы, тогда следует максимально увлажнить воздух! К сожалению, это тоже вредно. Слишком высокая влажность способствует развитию аллергических заболеваний, включая астму.

У нормальной влажности воздуха есть свои коварные враги — наши удобства. Звучит странно? Рассмотрим несколько примеров. Включѐнный летом кондиционер охлаждает воздух, нам становится легче дышать, но при этом кондиционер воздух сушит: испаряемая влага собирается в виде конденсата снаружи системы, за окном. Включѐнное отопление, согревающее нас зимой, также сушит воздух: именно поэтому зимой быстро высыхает одежда, если развесить её, допустим, на кухне. Но есть враг ещё коварнее — это твой компьютер! Он и его периферийное обо-

рудование также сушат воздух при работе. Повышению влажности способствует пар, образующийся во время принятия горячей ванны или душа, а также при приготовлении (несомненно, вкусной) пищи.

Если в квартире слишком сухо, стоит задуматься о покупке или разведении комнатных растений. Поскольку они требуют полива, часто в поддоне и горшках стоит вода, которая испаряется в воздух, увлажняя его. Кроме того, если цветы приходится слишком часто поливать, то и без специальных приборов видно, что воздух слишком сухой. Если же показатель влажности выше допустимой нормы, следует задуматься о более частом проветривании помещения.



Ещё один регулируемый фактор, влияющий на микроклимат помещения, — это **скорость движения воздуха**. Её трудно измерить привычными средствами. Если в каждой квартире есть термометр, не в каж-

Кстати! Зимой влажность в квартире может составлять менее 20% — этот показатель ниже, чем в пустыне Сахара.



«Зимний враг» нормальной влажности

дой — гигрометр, то прибор для измерения скорости движения воздуха — **анемометр** (от греч. άνεμος — ветер и μετρέω — измеряю) почти не встречается. Обычно он состоит из чашечной вертушки, закреплённой на оси, соединённой с измерительным механизмом. Согласно ГОСТ скорость движения воздуха в общественном или жилом помещении не должна превышать 0,25 м/с в тёплое время года и 0,1–0,15 м/с в холодное. Никаких сквозняков!



Оставшимся регулируемым фактором микроклимата помещения, на который стоит обратить внимание, является качество воздуха. Мы рассмотрим только один его компонент. Тебе приходилось слышать фразу: «Здесь мало кислорода»? Обычно её произносят, когда в помещении начинает болеть голова, трудно дышать. На самом деле виноват не недостаток, а избыток, но совершенно другого газа — **углекислого** (CO₂, диоксид углерода), который мы выдыхаем. Оптимальным для здоровья человека уровнем CO₂, согласно ГОСТ, считается 800–1000 мд. Врачи отмечают, что лучше сдвинуть границы до 600–800 мд. Под сокращением «мд» скрываются миллионные доли — количество миллионных долей конкретного вещества от всего множества (в данном случае воздуха как смеси газов). Если проводить аналогию с процентным содержанием, то 1% = 10 000 мд. Углекислый газ не только участвует в процессах глобального потепления и парникового эффекта, но и оказывает негативное влияние на человека. Повышенная концентрация CO₂ в помещении приводит к учащению пульса, головной боли, невозможности сосредоточиться на чём-либо, потливости и затруднённому дыханию.



Есть ещё один немаловажный нерегулируемый фактор, оказывающий сильное влияние на здоровье человека, но не входящий в понятие микроклимата помещений. Этот показатель один и тот же на улице и в доме, а также присутствует в любом прогнозе погоды. Уже догадался? Правильно, речь идёт об **атмосферном давлении**. Для его измерения используют **барометры** (от греч. βάρος — тяжесть и μετρέω — измеряю).

В СИ давление измеряется в паскалях (Па) в честь великого физика и математика Блеза Паскаля (1623–1662 гг.),

$$1 \text{ Па} = \frac{1 \text{ кг}}{\text{м} \times \text{с}^2}.$$

Кстати! Хотя в ГОСТ приводится верхняя допустимая граница 1000 мд, при подобном показателе практически у половины людей наблюдаются головные боли и повышенная утомляемость.



Важно проветривать помещение!

Однако традиционно в России для метеорологических наблюдений используют другую величину — миллиметры ртутного столба (мм рт. ст.).

Нормальным считается атмосферное давление около 760 мм рт. ст. — именно такое давление фиксируется на уровне моря на широте 45° при температуре 0°C . Большинство городов находятся на другой высоте и тем более широте, поэтому для них нормальным является другое давление. Например, Москва расположена в среднем на высоте 120 м над уровнем моря, поэтому для неё характерно атмосферное давление 748 мм рт. ст., а для Санкт-Петербурга, расположенного на высоте 30 м над уровнем моря, нормальным считается давление 753–755 мм рт. ст. Также на изменение давления сильное влияние оказывает изменение температуры. При нагревании воздух увеличивается в объёме, и из-за этого уменьшается атмосферное давление.

Почему же так важно вовремя отслеживать изменение атмосферного давления? Отклонение всего на 10 мм рт. ст. в любую сторону может вызвать сильное ухудшение самочувствия: мигрень, тошноту, сонливость и др. Особенно опасно понижение или повышение давления на 1 мм рт. ст. в течение 3 часов. К сожалению, доказано, что 50% женщин и 30% мужчин страдают от перепадов давления. В группе риска находятся практически все подростки и пожилые люди.

Кстати! Блез Паскаль изобрёл прототип калькулятора — суммирующую машинку, получившую название «Паскалина». Великий математик начал её разработку в 1642 году в возрасте 19 лет.

Итак, ты выяснил, что для хорошего самочувствия и крепкого здоровья очень важно следить не только за микроклиматом помещений, но и за атмосферным давлением. Готов создать домашнюю компактную эко-станцию?

Вперёд, эколог!

Обозначения

1. Скетч — программа, которую обрабатывает Arduino.
2. 5 V (5 вольт) — обозначение напряжения питания платы.
3. 3,3 V (3,3 вольт) — обозначение альтернативного напряжения платы (по конкретным выходам).
4. GND (от англ. *Ground* — земля) — заземление электрических элементов.
5. // — обозначение в программе однострочных комментариев, в которых приводится пояснительная информация.
6. /*текст*/ — обозначение в программе комментариев из нескольких строк.

Оборудование:

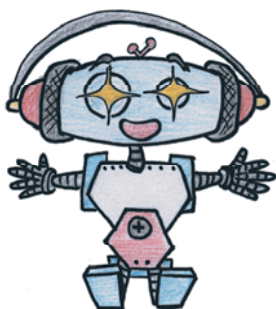
- Компьютер (минимальные требования): Windows XP, Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10 (32/64 bit)/Linux Mint, Ubuntu, Fedora/Mac OS X, оперативная память не менее 512 Мб, процессор — 1,1 ГГц (или быстрее), свободное место на диске — 200 Мб.
- Среда программирования Arduino IDE.
- Плата Arduino Uno.
- Плата расширения Wireless Shield (для беспроводного взаимодействия).
- Модуль Bluetooth Bee или аналогичный (для беспроводного взаимодействия).
- Макетная плата BreadBoard Mini (170 точек).
- Датчик освещённости (Troyka Module).
- Датчик атмосферного давления и температуры BMP180.
- Датчик газа MQ-135.
- Датчик влажности и температуры DHT11.
- Звуковой модуль ISD1820 с динамиком.



- Соединительные провода с двумя концами типа штекер, 13 шт. (4 чёрных, 4 красных, 1 синий, 1 белый, 1 оранжевый, 1 зелёный, 1 жёлтый).
- Соединительные провода с концами типа штекер и гнездо, 3 шт. (2 красных и 1 зелёный).
- Соединительный тройной шлейф с двумя концами типа гнездо (входит в комплект датчика освещённости Troyka Module).
- Кабель USB (A-B) для подключения Arduino к компьютеру.
- Импульсный блок питания для мобильных устройств (2A) или внешний аккумулятор типа PowerBank (необязательно).
- Мобильное устройство с поддержкой технологии Bluetooth (смартфон или планшет под управлением Android 2.0+, iOS 6+, Windows Phone).
- Карандаш.
- Линейка.
- Ножницы или канцелярский нож.
- Скотч или клей для бумаги.
- Лист картона или картонная коробка.

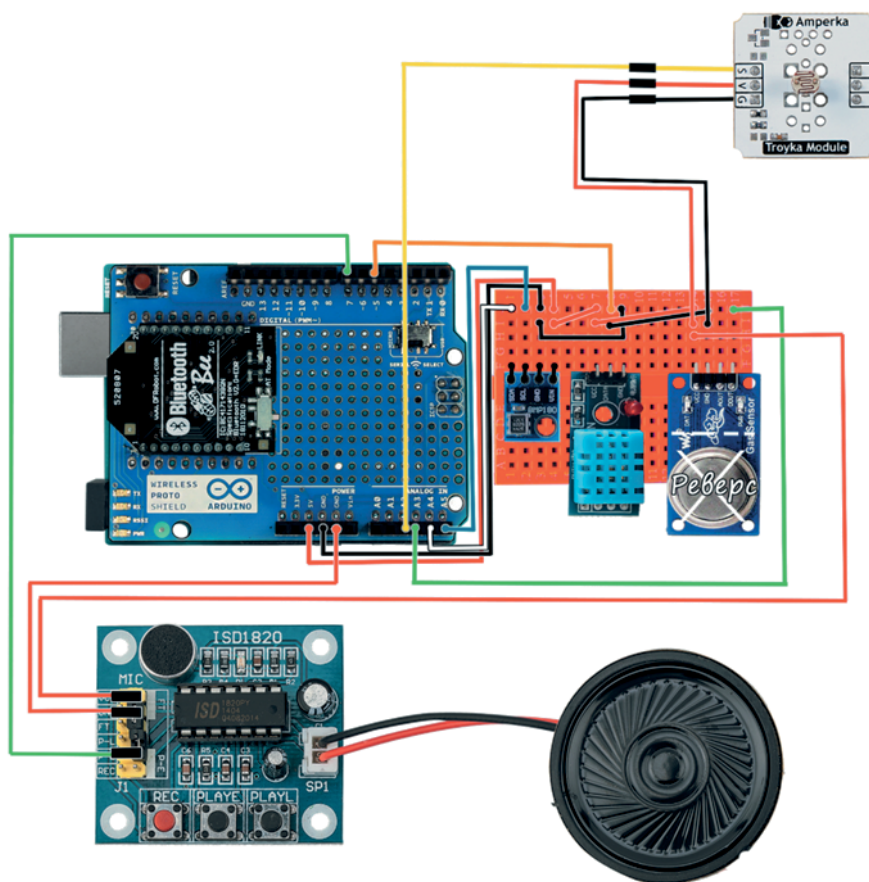
Arduino — платформа с открытой аппаратной архитектурой. Это значит, что в свободном доступе опубликовано подробное описание самой платы, её компонентов, а также все электрические схемы, т. е. *спецификация*. Она позволяет любому производителю создать копию продуктов для платформы, тем самым делая их доступными большому количеству людей, а также создавать улучшенные, более эффективные версии плат и модулей или новые совместимые устройства.

Для своего проекта ты можешь использовать модули, аналогичные указанным в списке, но других производителей.



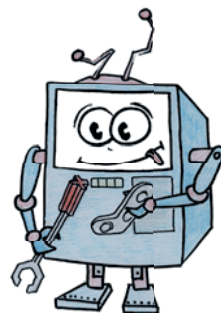
Этап 1. Устройство экостанции

Рассмотри общую схему будущего устройства. Для чего используется каждый компонент?



Как ты думаешь, какую плату не видно на схеме? Попробуй мысленно представить поэтапное подключение устройства. Теперь начинай его собирать. Для этого внимательно рассмотри рисунки следующих этапов и прочитай подписи к ним. В случае затруднений обращай за помощью к взрослым.

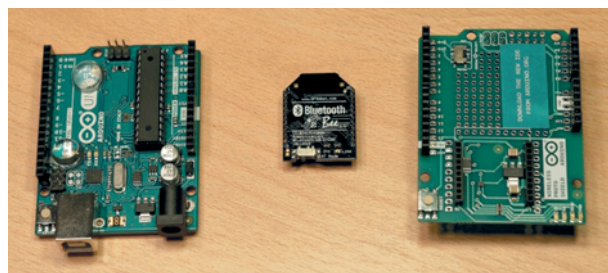
Этап 2. Сборка экостанции



ШАГ 1. ПОДКЛЮЧЕНИЕ МОДУЛЯ БЕСПРОВОДНОЙ СВЯЗИ

Детали для сборки:

- плата Arduino Uno, 1x;
- модуль Bluetooth Bee, 1x;
- плата расширения Wireless Shield, 1x.

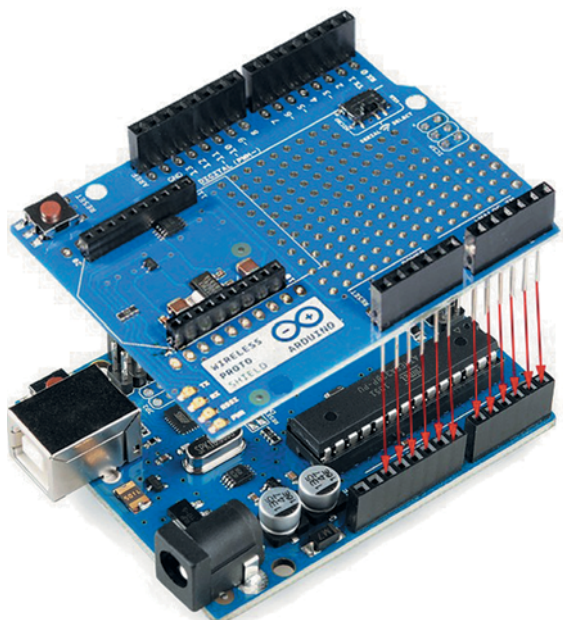


Для удобства доступа к показаниям экостанции из любого места в квартире или кабинете лучше отказаться от проводного соединения. Тем более современные телефоны — это полноценные компьютеры, позволяющие не только просмотреть и сохранить данные, но и обработать их для представления другим.

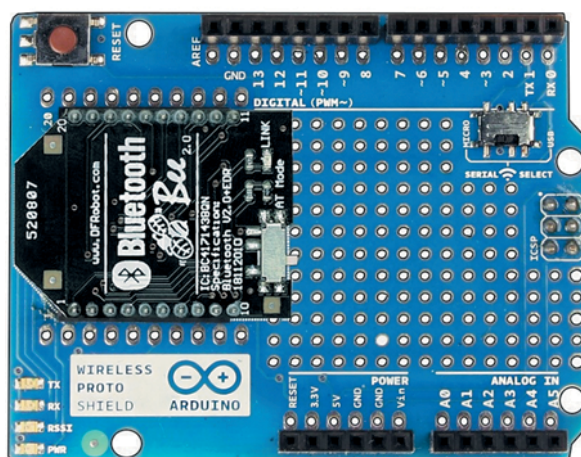
В твоём смартфоне наверняка есть модуль Bluetooth. Он потребляет меньше энергии, чем другие типы беспроводной связи, а если модель твоего телефона выпущена после 2010 года, то затраты будут минимальны! Именно поэтому носимые гаджеты и беспроводные колонки подключаются по Bluetooth.

Если твой телефон старше — не отчаивайся: все версии этой технологии совместимы между собой. Кстати, в настоящее время 95% мобильных телефонов оснащены модулями Bluetooth.

Кстати! Усилить сигнал Bluetooth или Wi-Fi можно с помощью обыкновенной пустой банки, оставшейся от выпитой газировки. Для этого аккуратно разрежь и разверни основную поверхность банки, оставив дно в качестве устойчивого основания. Модем или устройство со встроенным модемом помещается по направлению к получившейся антенне, которая и отражает сигнал.



1. Соедини плату Arduino Uno (на рисунке — снизу) и плату расширения Wireless Shield. Обрати внимание на нумерации цифровых выходов на обеих платах. Они должны совпадать.



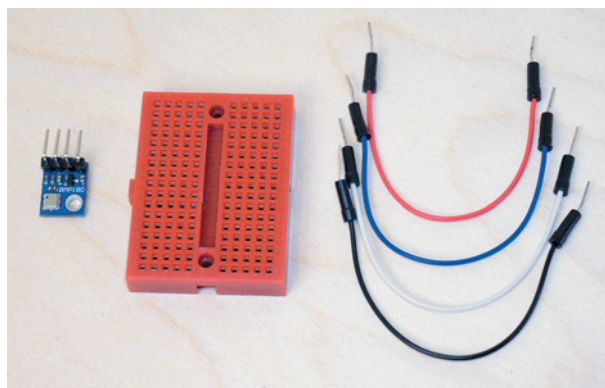
2. Подключи модуль Bluetooth Bee к специальному разъёму на плате Wireless Shield. Сделать это можно единственным способом — по пунктиру на плате расширения, как показано на рисунке.

Молодец! Ты расширил возможности платы Arduino Uno новым способом передачи данных! Пора перейти к датчикам, показания которых и будут отправляться.

ШАГ 2. ПОДКЛЮЧЕНИЕ ДАТЧИКА АТМОСФЕРНОГО ДАВЛЕНИЯ

Детали для сборки:

- датчик атмосферного давления и температуры BMP180, 1х;
- макетная плата BreadBoard Mini, 1х;
- провода с двумя концами типа штекер: красный, 1х; чёрный, 1х; синий, 1х; белый, 1х.



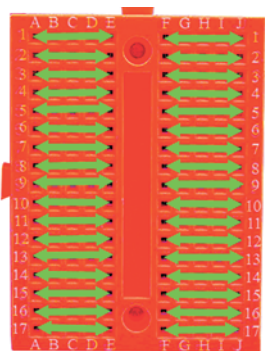
Датчик BMP180 на самом деле также способен измерять высоту над уровнем моря — это может быть полезно для расчёта нормального давления для твоей местности. Компонентами этого многосоставного датчика являются:

- пьезо-резистивный датчик для определения атмосферного давления;
- термодатчик для определения температуры окружающей среды;
- аналогово-цифровой преобразователь;
- собственная энергонезависимая электрически стираемая программируемая память (EEPROM);
- оперативное запоминающее устройство (ОЗУ);
- собственный микроконтроллер;
- преобразователь напряжения.

Именно эту модель датчиков давления помещают в некоторые смартфоны из-за её компактного размера. Преобразователь напряжения позволяет использовать питание как от контакта 3,3 V, так и от контакта 5 V. Передача данных от микроконтроллера BMP180 к микроконтроллеру ATmega328, являющемуся «мозгом» Arduino Uno, осуществляется по шине¹ данных I²C или SPI.

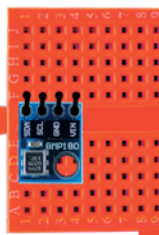
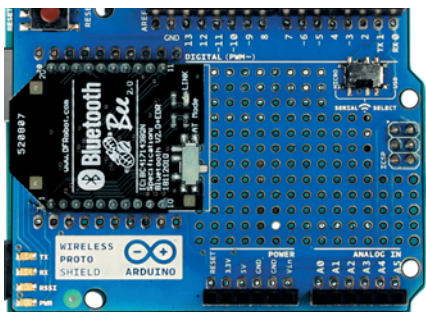
Последовательная шина данных I²C использует две двунаправленные линии связи: последовательную линию данных (serial data, SDA) и последовательную линию тактирования (serial clock, SCL).

¹ Шина — это электронный канал, связывающий несколько входов и выходов.



Теперь рассмотрим **макетную плату BreadBoard Mini**. На ней нанесены буквы, числа — все они являются обозначениями различных шин. Макетная плата позволяет соединять одновременно несколько компонентов без пайки.

Настало время подключить датчик к плате!



1. Установи датчик BMP180 на макетной плате так, чтобы **штырьковый контакт SDA** соответствовал гнезду **F1** платы, контакт **SCL** — гнезду **F2**, контакт **GND** — гнезду **F3**, а контакт **VIN** — гнезду **F4**.

В электронике, включая компьютерную технику, традиционно используются следующие цвета проводов и обозначения соответствующих контактов:

Назначение	Обозначение контактов	Цвет провода
Питание	5 V (voltage)	Красный
Земля	GND (ground)	Чёрный
Сигнал (считывание)	S (signal)	Жёлтый
Данные (передача)	DATA	Зелёный

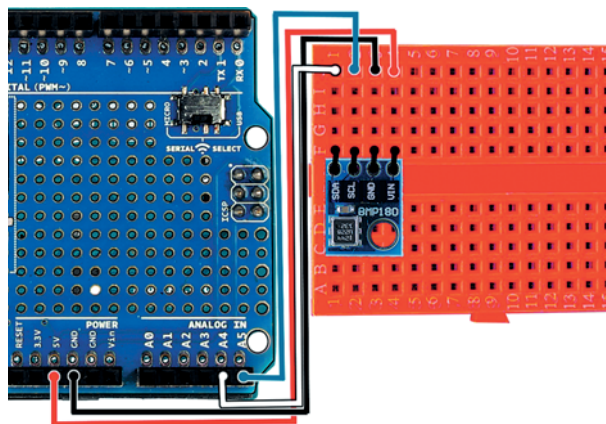
Кстати! Для лёгкого запоминания обозначений контактов можно воспользоваться ассоциацией «Сигнал-Вольтаж-Грунт».

Последние два пункта зачастую не различают, используя на практике только жёлтый или только зелёный провод.

2. Красным проводом соедини **контакт 5 V** платы Wireless Shield и **гнездо J4** макетной платы.

3. Чёрным проводом соедини **контакт GND** платы Wireless Shield и **гнездо J3** макетной платы.
4. Синим проводом соедини аналоговый вход **A5** платы Wireless Shield и **гнездо J2** макетной платы.
5. Белым проводом соедини аналоговый вход **A4** платы Wireless Shield и **гнездо J1** макетной платы.

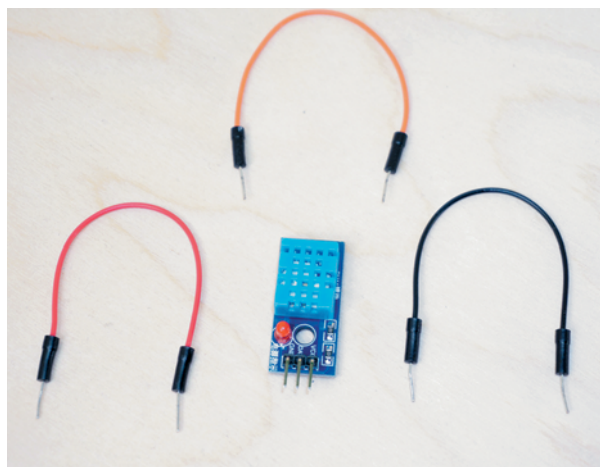
Молодец! Твоя экостанция получила орган, измеряющий давление и температуру. Но у тебя есть ещё один датчик, измеряющий температуру и влажность. Давай подсоединим и его?



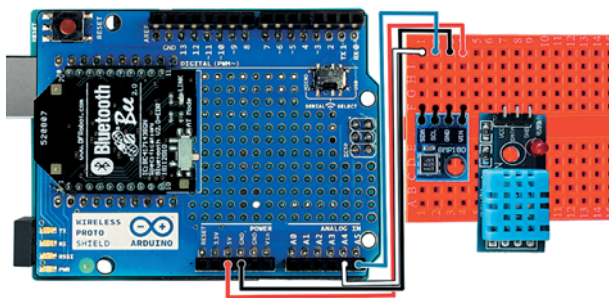
ШАГ 3. ПОДКЛЮЧЕНИЕ ДАТЧИКА ВЛАЖНОСТИ DHT11

Детали для сборки:

- датчик влажности и температуры DHT11, 1х;
- красный провод с двумя концами типа штекер, 1х;
- чёрный провод с двумя концами типа штекер, 1х;
- оранжевый провод с двумя концами типа штекер, 1х.



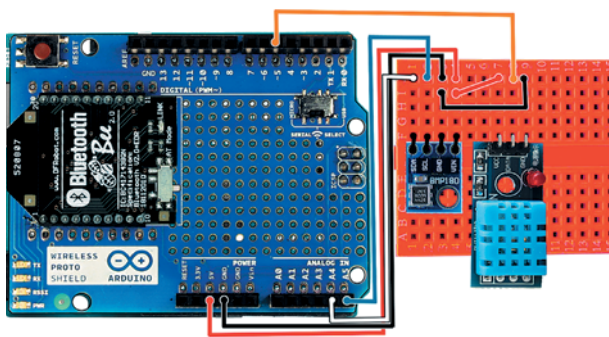
Цифровой датчик влажности DHT11 является оптимальным решением для мониторинга микроклимата помещений благодаря своей низкой стоимости. Он работает при температуре от 0 до 50 °С и влажности от 20 до 90% и имеет погрешность в 2% для температуры и 5% для влажности. Этого достаточно для проверки микроклимата на соответствие нормам. Если тебе захочется сделать более точные измерения, придётся покупать датчик DHT22 — «старшего брата» DHT11.



1. Установи датчик DHT11 на макетной плате так, чтобы **штырьковый контакт VCC** соответствовал гнезду **F7** платы, контакт **DATA** — гнезду **F8**, а контакт **GND** — гнезду **F9**.

2. Красным проводом соедини гнезда **I4** и **J7** макетной платы.

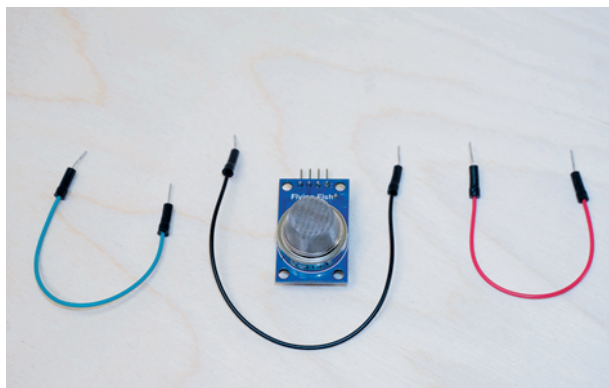
3. Чёрным проводом соедини гнезда **I3** и **J9** макетной платы.



4. Оранжевым проводом соедини **цифровой порт D5** (Digital № 5) платы Wireless Shield и гнездо **J8** макетной платы.

Ура! Ты подключил датчик влажности воздуха, и теперь смело можешь переходить к не менее важному датчику — анализатору газа!

ШАГ 4. ПОДКЛЮЧЕНИЕ ДАТЧИКА ГАЗА



Детали для сборки:

- датчик газа MQ-135, 1x;
- красный провод с двумя концами типа штекер, 1x;
- чёрный провод с двумя концами типа штекер, 1x;
- зелёный провод с двумя концами типа штекер, 1x.

Полное название MQ-135 — «датчик качества воздуха». Поскольку он измеряет плотность частиц (концентрацию) различных газов, то кратко его называют «датчик газа». Кстати, он чувствителен не только к углекислому газу (CO_2). Датчик поможет определить концентрацию:

- дыма;
- бензола (C_6H_6);
- оксидов азота (NO_x);
- аммиака (NH_3);
- паров различных спиртов.

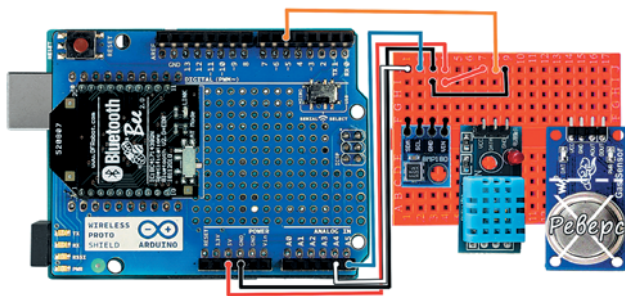
Работа датчика обеспечивается измерением выходного напряжения, зависящего от температуры встроенного элемента, материал которого вступает в химическую реакцию с газами. Чем больше концентрация газа, тем сильнее протекает реакция и выше температура (выделяется больше энергии). Поскольку содержание CO_2 достаточно велико, измерить его данным датчиком можно относительно точно.

Чем ещё интересен MQ-135?

- Датчик может работать в двух режимах: аналоговом и цифровом. В аналоговом режиме передаются значения, преобразовываемые АТМega328 с помощью специальной библиотеки в показания концентрации в миллионных долях. В цифровом режиме датчик передаёт 0, когда концентрация газов меньше заданного уровня, и 1, когда он превышен.
- Допустимый уровень физически регулируется вместе с чувствительностью датчика. Для этого на плате модуля размещён потенциометр — переменный резистор, который позволяет регулировать подаваемое на основной элемент напряжение. Для наших измерений подойдёт его стандартное положение.
- На плате размещён индикатор нормы — светодиод, который выключается, когда допустимое значение превышено.

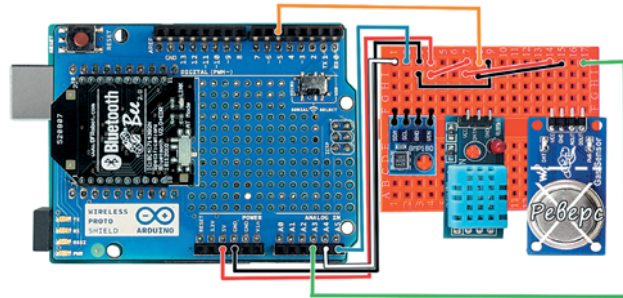
Замечательный датчик, не правда ли? Давай его подключим!

1. Установи датчик MQ-135 на макетной плате так, чтобы **штырьковый контакт VCC** соответствовал **гнезду F14** платы, контакт **GND** — гнезду **F15**, контакт **D0** — гнезду **F16**, а контакт **A0** — гнезду **F17**.



Для удобства мы будем отображать на схеме вид датчика с лицевой стороны. *На плате он должен быть закреплён наоборот!* Подписи и подключение контактов на схеме соответствуют реверсной (плоской) стороне датчика.

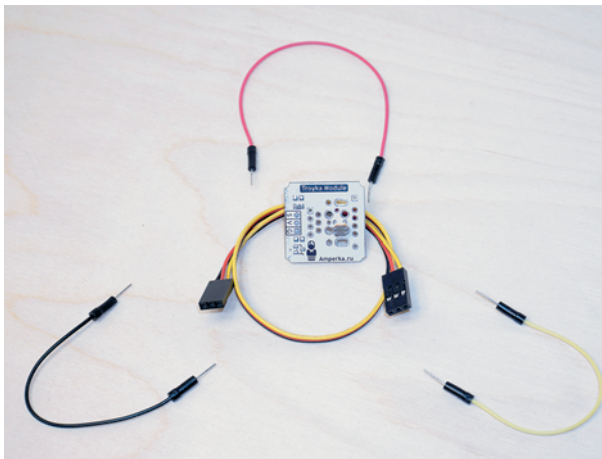
2. Красным проводом соедини **гнезда I7** и **J14** макетной платы.
3. Чёрным проводом соедини **гнезда I8** и **J15** макетной платы.
4. Поскольку нас интересует конкретное показание в миллионных долях, воспользуемся датчиком как аналоговым. Зелёным проводом соедини **аналоговый вход А3** (Analog № 3) платы Wireless Shield и **гнездо J17** макетной платы.



Датчик газа работает благодаря химической реакции. Для получения стабильных и правильных данных лучше прокалить датчик. Для этого, когда соберёшь устройство полностью, оставь его включённым на 12 часов под присмотром. Конечно, производитель рекомендует оставлять на сутки, но без присмотра датчик газа лучше не оставлять — во время прокаливании он сильно нагревается.

Так держать, инженер! Осталось совсем немного. Следующим будет датчик, сферу применения которого мы не затрагивали. Интригует? Давай разбираться!

ШАГ 5. ПОДКЛЮЧЕНИЕ ДАТЧИКА ОСВЕЩЁННОСТИ

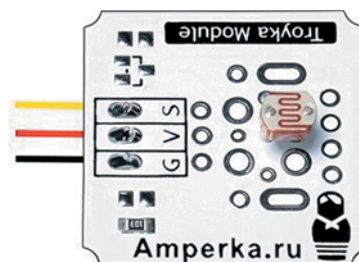


Детали для сборки:

- датчик освещённости (Troyka Module), 1x;
- соединительный тройной шлейф с двумя концами типа гнездо, 1x;
- красный провод с двумя концами типа штекер, 1x;
- чёрный провод с двумя концами типа штекер, 1x;
- жёлтый провод с двумя концами типа штекер, 1x.

На датчике используются следующие стандартные обозначения: S — сигнал (по этому контакту идёт информация), V — питание, G — земля.

1. Возьми идущий в комплекте с датчиком тройной шлейф с концами типа гнездо. Снова руководствуясь правилом, что земля — это чёрный провод, соедини контакты так, чтобы **к S шёл жёлтый провод**. Готово!



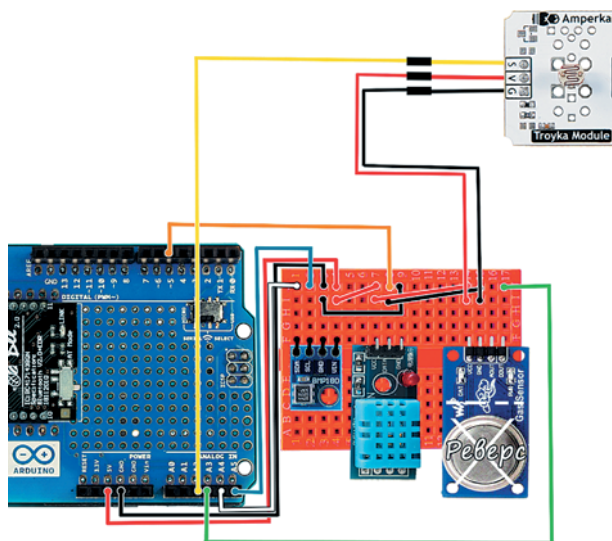
Датчик позволит проверить показания **освещённости** — точной физической величины, измеряемой в люксах (лк, 1 люмен на квадратный метр). Она показывает степень освещения поверхности. Чем сильнее источник света, тем больше освещённость. Низкая освещённость в помещении приводит к головной боли, повышенной утомляемости, проблемам со зрением. Кстати, слишком высокая освещённость тоже вредна. Оптимальные значения представлены в виде требований в СанПиН 2.2.1./2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещённому освещению жилых и общественных зданий» и СНиП 23-05-95 «Естественное и искусственное освещение». Мы приведём для тебя готовую таблицу оптимальных значений:

Место	Освещённость, лк
Аварийное освещение	50–70
Комната	150–250
Класс, рабочее место	250–350
Учебная аудитория (технология)	270
Учебная аудитория (русский язык, математика)	300
Компьютерный класс	350

Поняв, для чего требуется измерять освещённость, вернёмся к подключению датчика.

Но как подключить шлейф с гнездами на конце к таким же гнездам на макетной плате и Wireless Shield? Для этого тебе понадобятся оставшиеся соединительные провода.

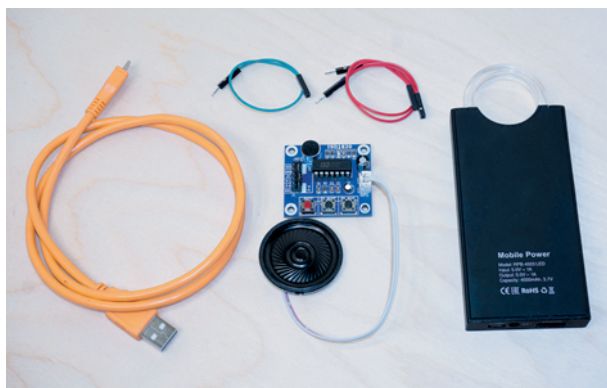
1. Красным проводом соедини гнездо **красного провода** шлейфа и **гнездо I14** макетной платы.
2. Чёрным проводом соедини гнездо **чёрного провода** шлейфа и **гнездо I15** макетной платы.



3. Жёлтым проводом соедини гнездо **жёлтого провода** шлейфа и **аналоговый вход A2** (Analog № 2) платы расширения Wireless Shield.

Отлично! Ты подключил все датчики и платы! Не хочешь добавить экостанции голос? Желательно, твой собственный. Как это сделать, ты сейчас узнаешь. Вперёд!

ШАГ 6. ПОДКЛЮЧЕНИЕ ЗВУКОВОГО МОДУЛЯ



Детали для сборки:

- звуковой модуль ISD1820 с динамиком, 1x;
- провода с концами типа штекер и гнездо: красный, 2x; зелёный, 1x;
- импульсный блок питания/Power Bank и USB-кабель или батарейка типа «Крона» и соответствующий ей провод, 1x.

Официальное полное название нашего звукового модуля — микросхема голосовых сообщений ISD1820. Это сложный многокомпонентный модуль с собственной энергонезависимой памятью и микроконтроллером. Кстати, он может работать и без подключения к Arduino — потребует-

ся только обеспечить его питанием и заземлением. В этом случае для управления используются три кнопки:

Название кнопки	Действие	Пояснение
REC	Запись	Для записи необходимо удерживать кнопку. Во время записи горит светодиод, к концу доступного времени светодиод предупреждающе мигает и гаснет при полном заполнении памяти
PLAYE	Единичное воспроизведение	После нажатия запись воспроизводится полностью, в конце включается светодиод
PLAYL	Повторяющееся воспроизведение	Воспроизведение происходит только при нажатой кнопке. Используется для воспроизведения фрагментов

Встроенная память позволяет перезаписывать звуковые сообщения до 100 000 раз. Правда, хранить на ней возможно лишь одну запись, которая может длиться 8–20 с. Таким образом, длительность записи и воспроизведения зависит от изменения номинала резистора R4 на плате от 80 до 200 кОм. При увеличении длительности качество записи падает.

Память для записи распределяется по стандартным принципам кодирования звуковой информации:

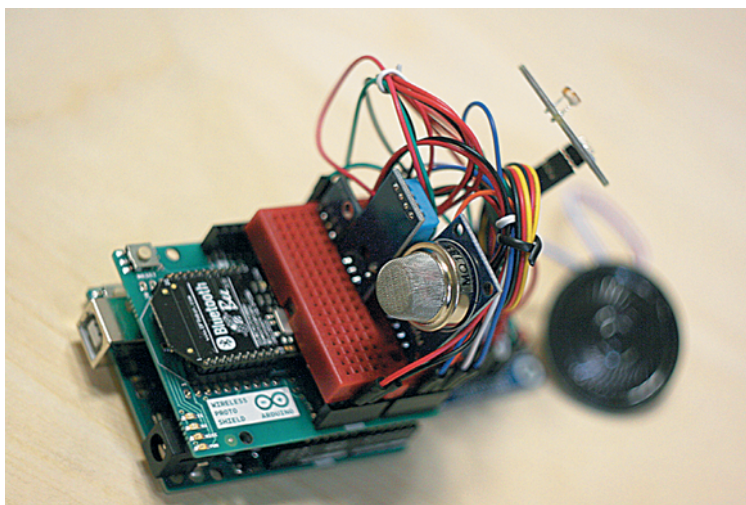
Сопротивление резистора, кОм	Длительность записи, с	Частота дискретизации, КГц	Пропускная способность, КГц
80	8	8	3,4
100	10	6,4	2,6
120	12	5,3	2,3
160	16	4	1,7
200	20	3,2	1,3

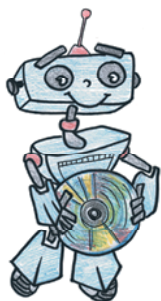
- Зелёным проводом соедини **контакт Р-Е** звукового модуля и **цифровой порт № 7** (Digital № 7) платы Wireless Shield. В будущем ты будешь подавать сигнал на однократное воспроизведение записи.

Отлично! Ты подключил звуковой модуль. Поскольку он имеет собственную энергонезависимую память, давай сразу наделим его твоим голосом.

- Подключи к USB-порту или разъёму питания платы Arduino Uno выбранный тобой элемент питания. Если ты планируешь использовать батарейки типа «Крона», то сначала подключи провод, затем вставь батарейку, соблюдая полярность. То же самое касается блока питания или Power Bank: кабель всегда подключается первым, затем подаётся питание.
- Зажми кнопку REC на звуковом модуле и произнеси официальным тоном текст оповещения. Он не должен быть слишком длинным, помни о значениях в таблице.
- Проверь запись. Для этого нажми один раз кнопку PLAYE. Если результат тебе не нравится, повтори шаг 5.

Молодец! Ты закончил сборку своего устройства, и оно готово к следующему этапу — программированию. А ты готов?





Этап 3. Установка программного обеспечения

Чтобы программировать плату Arduino, требуется установить на компьютер специальную среду разработки программ, которая называется Arduino IDE. Это программное обеспечение распространяется бесплатно, поэтому ты без проблем можешь скачать себе установщик, перейдя на официальный сайт Arduino:

<https://www.arduino.cc/en/Main/Software>

Download the Arduino Software

ARDUINO 1.6.9
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board. Refer to the Getting Started page for installation instructions.

Windows Installer
Windows ZIP file for non admin install

Mac OS X 10.7.11 or newer

Linux 32 bits
Linux 64 bits
Linux ARM (experimental)

Release Notes
Source Code
Checksums

В перечне справа выбери операционную систему, установленную на твоём компьютере.

Если у тебя установлена ОС Windows, но нет прав администратора (например, родители ограничили учётную запись), кликни по второй строке.

На странице загрузки тебе предложат сделать пожертвование разработчикам среды. Если ты хочешь скачать установщик без вноса, кликни на **Just Download**.

Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.

SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED 8,733,645 TIMES. IMPRESSIVE! THIS IDE IS NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS. HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING IT TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEIT. YOU CAN HELP ACCELERATE THE DEVELOPMENT OF THE ARDUINO IDE BY CONTRIBUTING TOWARDS THE EFFORT OF MAKING IT BETTER.

\$3 \$5 \$10 \$25 \$50 OTHER

JUST DOWNLOAD **CONTRIBUTE & DOWNLOAD**

Если у тебя нет компьютера или ты хочешь, чтобы среда программирования для роботов была всегда с собой, то ты можешь найти бесплатные приложения в Google Play, например ArduinoDroid. Кстати, если ты программируешь на школьном компью-

тере, то ты можешь воспользоваться веб-версией среды программирования, доступной по адресу: **<http://editor.arduino.cc/>**

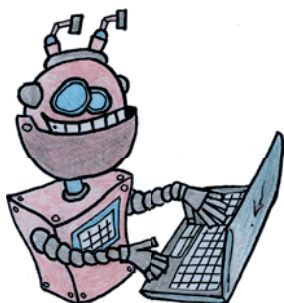
Программы, написанные в редакторе кода среды Arduino IDE, называются **скетчами** (от англ. *sketch* — набросок). Они сохраняются в собственном формате **-.ino**. Программы тебе предстоит писать непосредственно на языке программирования Wiring, который является облегчённой версией языка C++.

Установи на свой смартфон приложение, реализующее терминал последовательного порта по технологии Bluetooth. Бесплатные приложения доступны в большом количестве как в Play Market, так и в AppStore. Найти приложение легко по запросу Serial Bluetooth.

Мы используем систему Android 4.4 и установленное через Play Market бесплатное приложение **Serial Bluetooth Terminal**, потому что оно имеет удобный и неперегруженный интерфейс, сохранение логов (полученных и отправленных команд и данных) в txt-файл на карту памяти телефона и копирование логов в буфер обмена для дальнейшей передачи по Сети.

Инструкция для других приложений отличается незначительно.

Кстати! C++ — один из самых популярных языков программирования, на нём написаны практически все компьютерные игры, например Need for Speed Rivals и EA Sports FIFA 15.



Этап 4. Первый запуск и проверка оборудования

Основные кнопки:



Компилировать (собирать для данного микропроцессора) скетч для проверки на наличие ошибок.



Компилировать программу и загрузить её на подключённое устройство (Arduino сразу же запустит полученную программу).



Создать новый скетч (среда откроет новую вкладку, в качестве названия будет использована дата).

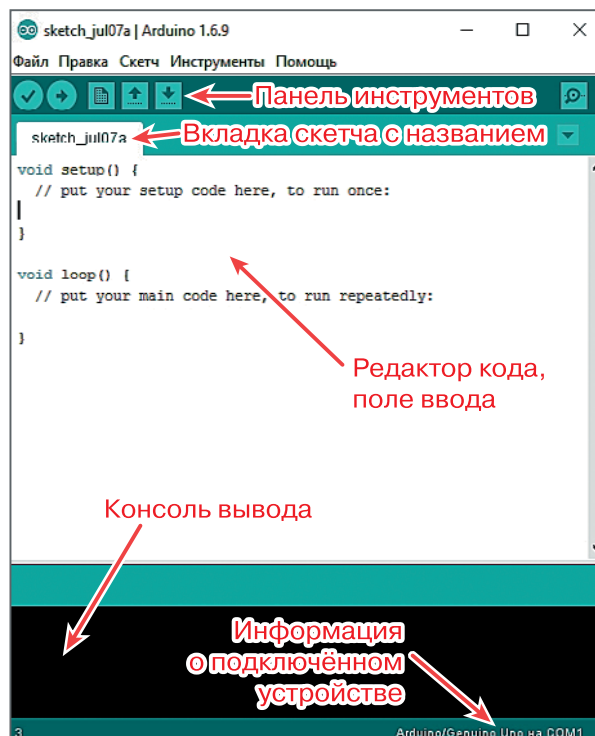


Открыть сохранённый ранее скетч.

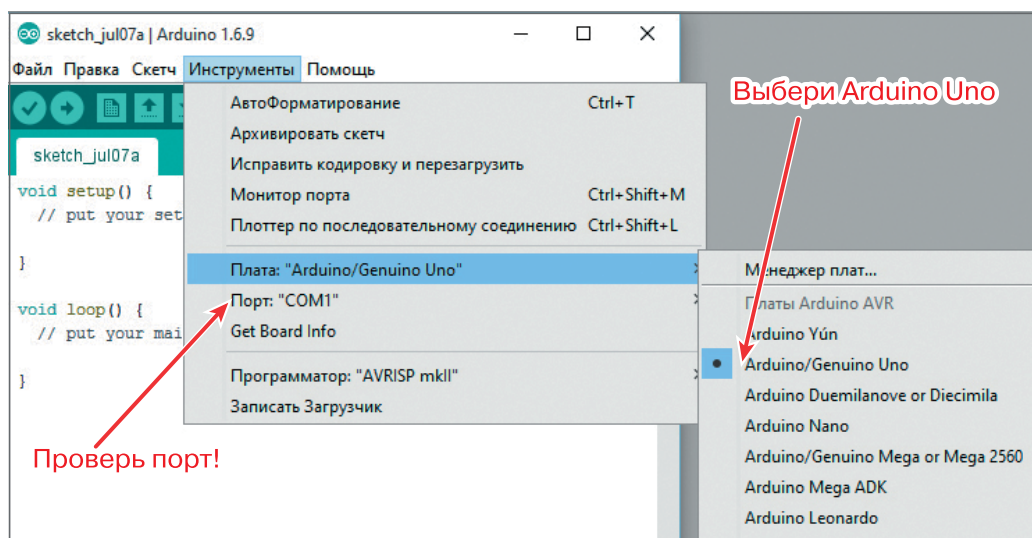


Сохранить открытый скетч. Не забывай использовать эту функцию перед закрытием среды, чтобы не потерять достигнутый результат!

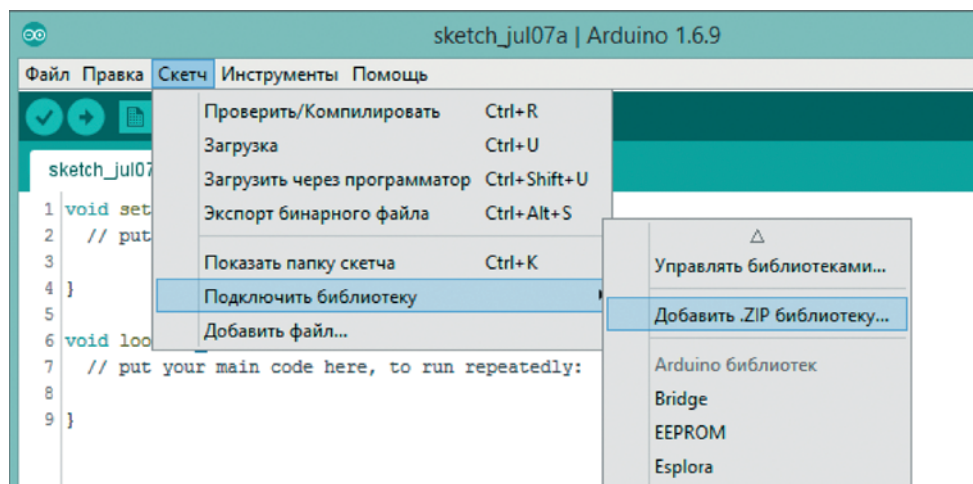
1. Подсоедини к компьютеру Arduino Uno с помощью USB-кабеля.
2. Запусти среду Arduino IDE. Должно появиться следующее окно программы:



3. Если название подключённого устройства не отображается или отображается неверно, выбери плату и порт вручную:



4. Прежде чем подготовить программу, следует убедиться, что всё оборудование подсоединено и работает правильно. Надо будет проверить, есть ли сигнал от датчиков, работает ли звуковой модуль.
5. Для будущего скетча понадобится подключить специальные библиотеки. Для этого следует их сначала загрузить в среду программирования. Скачай их со страницы издания по адресу <http://pilotlz.ru/books/287/10390/>.
6. Добавь скачанные файлы в среду, используя инструмент импорта .ZIP-библиотек:



7. Для проверки компонентов перепиши приведённый ниже скетч в поле ввода. Что значат конкретные команды и функции, ты узнаешь на этапе программирования, а пока требуется проверить физические компоненты.

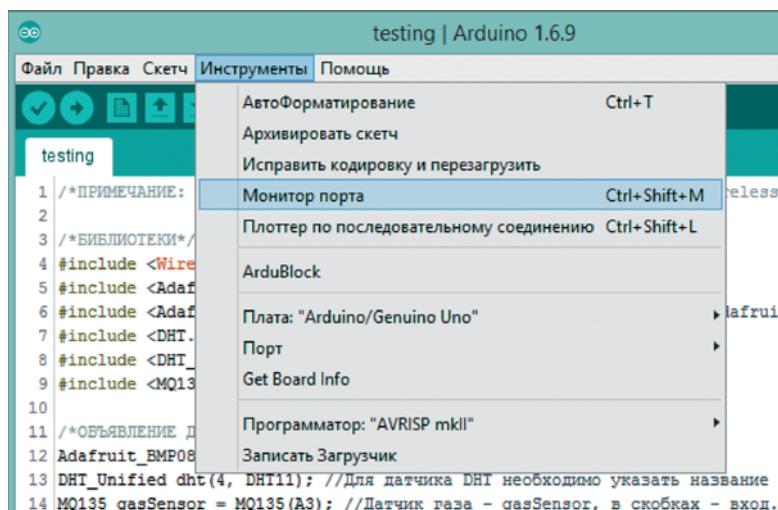
```
1. /*ПРИМЕЧАНИЕ: Если ошибка загрузки, переключить рычажок
   на плате Wireless Shield рядом с первыми цифровыми портами.*/
2.
3. /*БИБЛИОТЕКИ*/
4. #include <Wire.h> //Для работы с шиной SPI
5. #include <Adafruit_BMP085.h> //Для датчика bmp185 (bmp085)
6. #include <Adafruit_Sensor.h> //Для использования функций датчиков Adafruit
7. #include <DHT.h> // DHT11
8. #include <DHT_U.h>
9. #include <MQ135.h> //MQ-135.
10.
11. /*ОБЪЯВЛЕНИЕ ДАТЧИКОВ*/
12. Adafruit_BMP085 bmp; //Объявили конкретный BMP085.
13. DHT_Unified dht(4, DHT11); //Для датчика DHT необходимо указать название
   (dht), вход и тип датчика.
14. MQ135 gasSensor = MQ135(A3); //Датчик газа - gasSensor, в скобках - вход.
15.
16. /*Функция setup() переводится как «установка». Эта функция отрабатывает
   один раз после запуска Arduino, устанавливая конкретные роли цифровых
   контактов и запуская обмен данными по последовательному порту
   и шине SPI.*/
17. void setup() {
18.     Serial.begin (9600); //UART. В данном случае выводится на Bluetooth
19.     pinMode(7, OUTPUT); //Динамик
20.     pinMode(4, INPUT); // Датчик DHT
21.     dht.begin(); //Инициализация датчика влажности.
22.     bmp.begin(); //Инициализация датчика давления.
23.     sensor_t sensor; //Задание названия состояния и датчика bmp, из-за того,
        что библиотека на «чистом» C.
24. }
25.
26. /*Функция loop() отвечает за повторяющиеся действия. Именно здесь
   расположена основная часть программы.*/
27. void loop() {
28.     if (Serial.read() == '0') { //При получении команды 0
        по последовательному порту.
29.         //Влажность
30.         sensors_event_t event; //Объявление переменной
            для текущего состояния датчика
31.         dht.humidity().getEvent(&event); //Получение текущего
            показания влажности
32.         Serial.println(event.relative_humidity); //Выведение текущего
            показания влажности на экран
33.         //Давление в мм рт. ст.
34.         Serial.println(bmp.readPressure() / 133.3);
35.         //Содержание CO2.
```

```

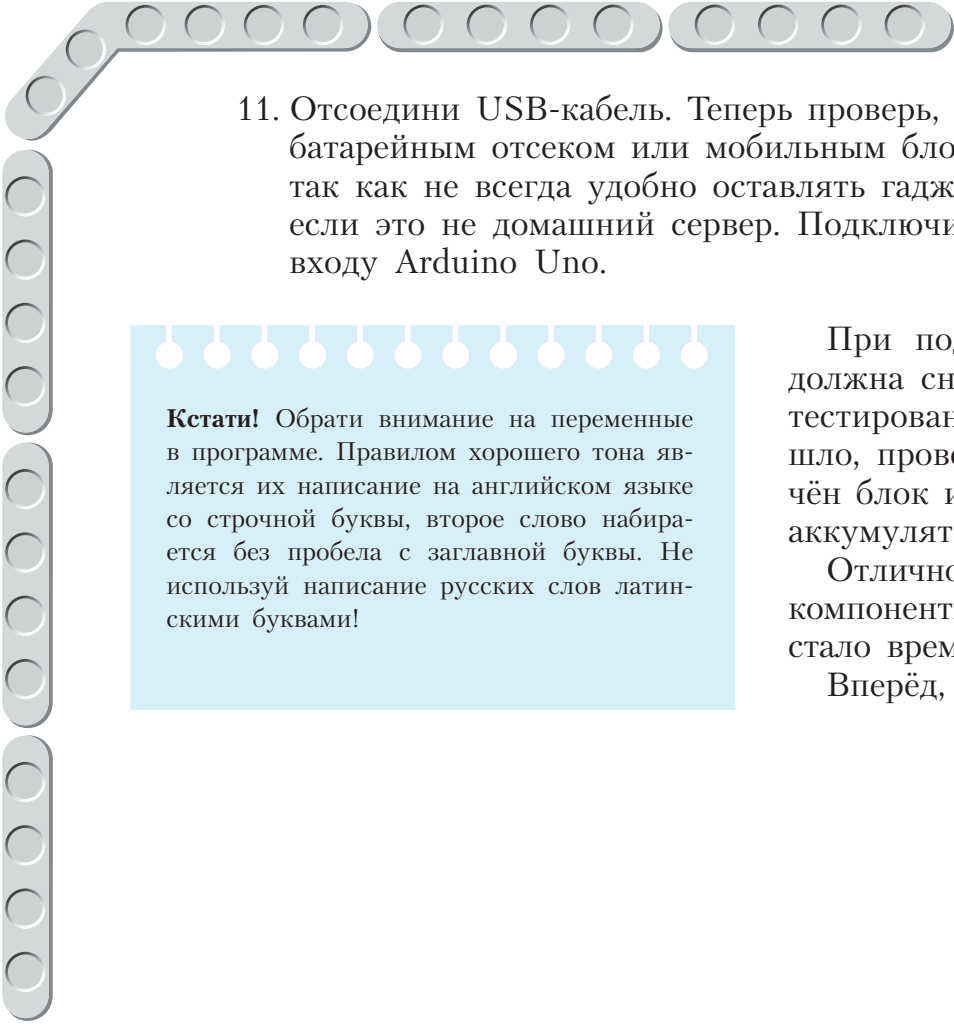
36.         Serial.println(gasSensor.getPPM());
37.         //Освещённость (без преобразования в люксы)
38.         Serial.println(analogRead(0));
39.         //Температура (DHT)
40.         dht.temperature().getEvent(&event);
41.         Serial.println(event.temperature);
42.         //Температура (BMP)
43.         Serial.println(bmp.readTemperature());
44.     }
45. }

```

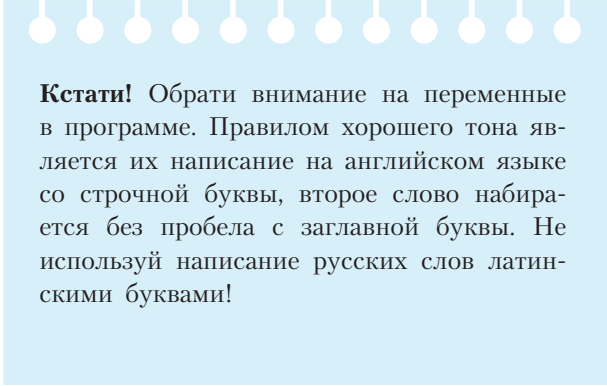
8. Проверь правый нижний угол окна программы. Если название устройства отображается, нажми на кнопку загрузки. Если названия нет, проверь подключение кабеля.
9. Далее появится диалог сохранения скетча. Дай ему имя testing и сохрани в удобной для тебя папке в памяти компьютера. Скетч будет скомпилирован¹ и загружен на подключённое устройство. Если компиляция не удалась, проверь правильность написания переменных и синтаксис. Если не удалась загрузка, переключи небольшой рычажок на плате Wireless Shield рядом с первыми цифровыми выходами. Не забудь после загрузки вернуть его в положение беспроводного порта!
10. Открой монитор последовательного порта. В нём должны отображаться данные. Сравни их с показателями домашнего термометра, барометра (если есть) и данными с сайтов погодных информеров об атмосферном давлении.



¹ Компиляция — это процесс перевода программы на языке программирования, известном человеку, на язык, понятный компьютеру, и «упаковка» его в обрабатываемый микроконтроллером файл.



11. Отсоедини USB-кабель. Теперь проверь, как твой гаджет работает с батарейным отсеком или мобильным блоком питания. Это разумно, так как не всегда удобно оставлять гаджет включённым в сеть ПК, если это не домашний сервер. Подключи штекер блока питания ко входу Arduino Uno.



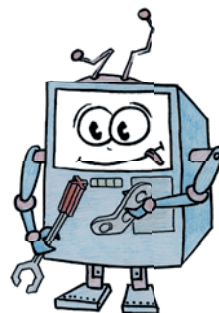
Кстати! Обрати внимание на переменные в программе. Правилom хорошего тона является их написание на английском языке со строчной буквы, второе слово набирается без пробела с заглавной буквы. Не используй написание русских слов латинскими буквами!

При подключении блока питания должна снова включиться программа тестирования. Если этого не произошло, проверь, правильно ли подключён блок или заряжен ли мобильный аккумулятор.

Отлично! Устройство готово, его компоненты работают нормально. Настало время создать ему корпус.

Вперёд, инженер!

Этап 5. Создание декоративного корпуса экостанции



Ты когда-нибудь видел, чтобы готовые электронные устройства лежали на открытом пространстве, поигрывая металлическим блеском проводов на солнце? Если видел, то, пожалуйста, больше туда не ходи. Если Arduino — это слаботочная плата, прикосновение к замыкающим контактам которой не нанесёт сильного вреда человеку, то оголённые контакты бытового прибора в условиях, допустим, повышенной влажности смертельны!

Твоей экостанции не хватает корпуса. Конечно, его можно собрать из пластика, но тогда придётся делать специальные отверстия, чтобы воздух не был неподвижным.

Дело в том, что принцип работы датчика газа основан на химической реакции, при которой происходит выброс энергии, сопровождающийся нагреванием. В замкнутом пространстве это может сильно повлиять на показания температуры и влажности, ведь чем выше температура, тем суше воздух.

Кроме того, картон лучше пропускает воздух, чем пластик, а также при изготовлении корпуса из картона появятся удобные зазоры при закреплении частей. Дополнительное пассивное охлаждение (так по-научному называются дырки) в картоне делать тоже удобнее, чем в пластике.

Тебе понадобятся:

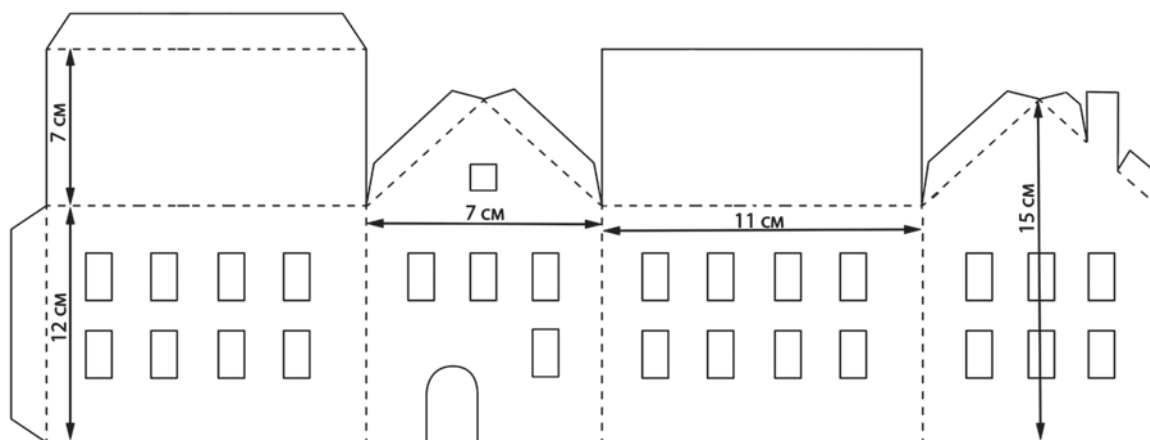
- картон;
- ножницы или канцелярский нож;
- скотч или клей;
- линейка.



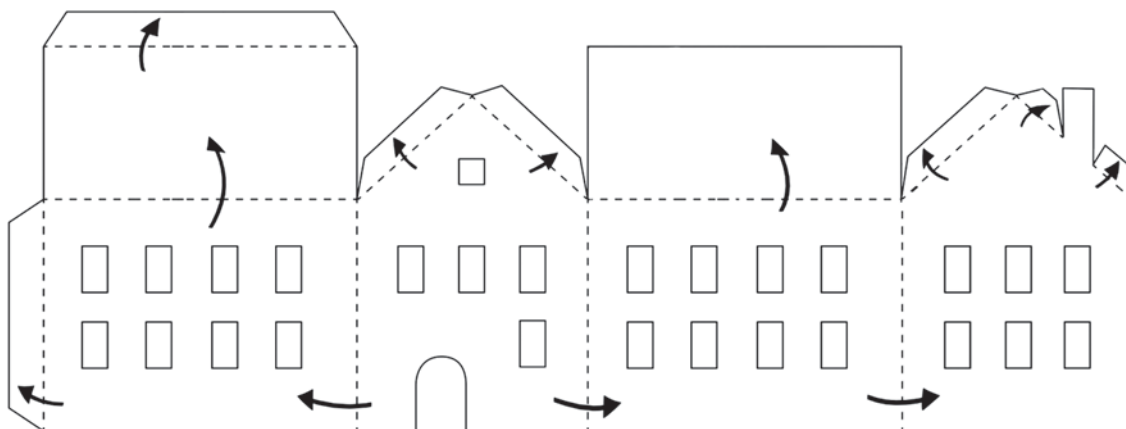
1. Нанеси развертку будущего корпуса на лист картона. Минимальные размеры корпуса должны быть такими:

Высота	9 см
Короткая сторона	5,5 см
Длинная сторона	9 см

Ты можешь изменять размеры в сторону увеличения, мы указали те, которые точно подходят для безопасной эксплуатации экостанции. На рисунке линии сгиба обозначены пунктиром, линии разреза — сплошной чертой.



2. Вырежи развёртку по сплошным линиям. Кстати, ты можешь предварительно её дополнить так, чтобы у экостанции было основание.
3. Согни детали будущего домика по пунктирным линиям, как показано на рисунке.

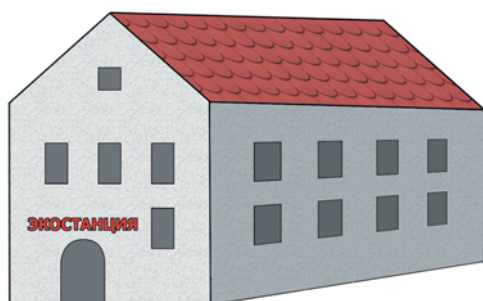


4. Склей выступы, чтобы домик держал форму.

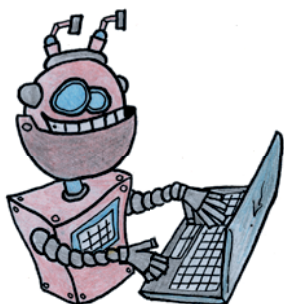
5. Оформи домик по своему вкусу: нарисуй черепицу, добавь кладку на стены. Для декорирования ты также можешь использовать подручные средства. Главное, не забывай, что вентиляционные отверстия закрывать нельзя!

Внимание!

Нежелательно, чтобы датчики упирались в картон. Если ты не прокалил датчик газа, он будет сильно нагреваться, что может привести к крайне неприятным последствиям и даже возгоранию картона!



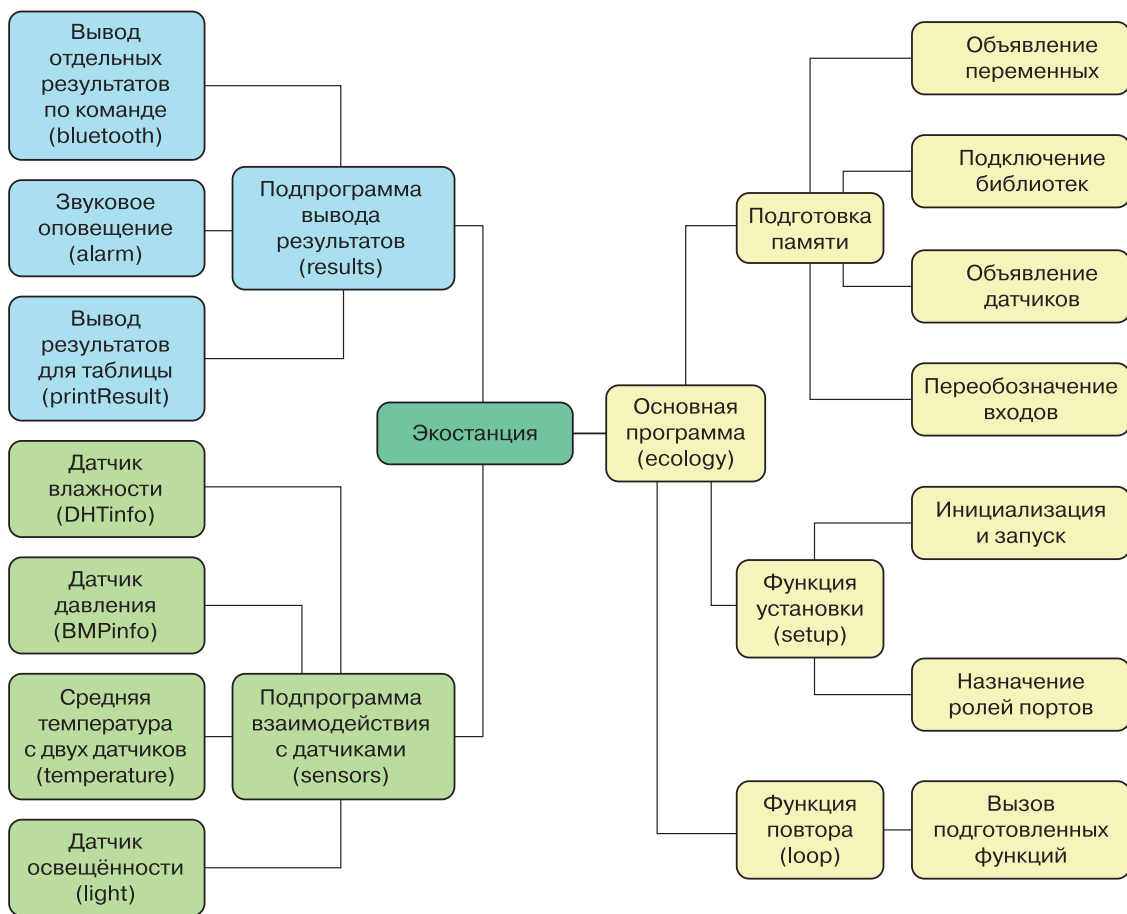
Здорово! Теперь это настоящая экологическая станция! Кстати, не захотелось ли тебе посадить рядом человечка из конструктора — эколога? В любом случае пора добавить немного полезного кода.



Этап 6. Создание программы для устройства


Логика программы

Что требуется от устройства? Оно должно снимать показания датчиков атмосферного давления и влажности, определять уровень освещённости и температуру, а затем выводить соответствующие значения в последовательный порт по протоколу Bluetooth в виде отформатированных результатов для возможной обработки в таблице. Для удобства следует добавить несколько команд, благодаря которым будет выводиться показание конкретного датчика с сопроводительным текстом. Если показания превышают допустимые, должен включиться звуковой сигнал. Схематично представим логику программы:



Многие основные этапы на схеме — самостоятельные функции. Приступим к их написанию.

ШАГ 1. ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ARDUINO IDE

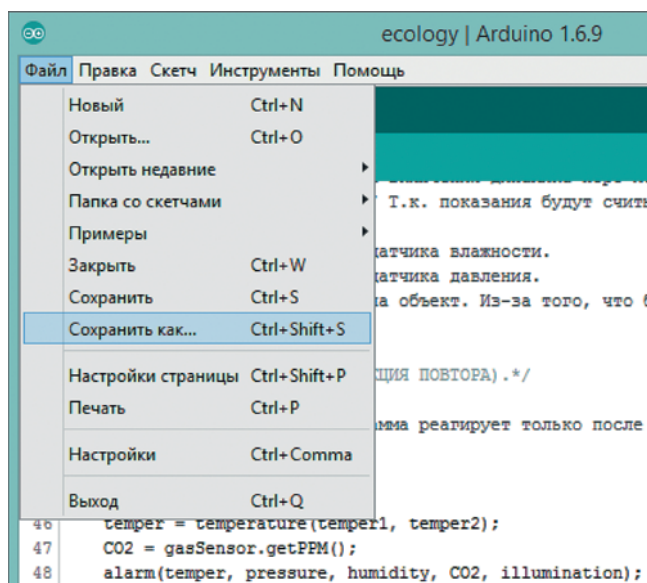
1. Запусти программную среду Arduino IDE.
2. В открывшемся окне выбери кнопку  «Новый».

ШАГ 2. СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ ЭКОСТАНЦИИ

В поле ввода начинай составлять код программы для Arduino.

Начнём с основной программы, которую необходимо сохранить как файл **ecology.ino**. Для этого в контекстном меню во вкладке «Файл» выбери пункт «Сохранить как...».

Одну программу или подпрограмму в Wiring называют скетч («набросок»), а несколько скетчей одного проекта составляют скетчбук («книгу набросков», т.е. альбом). Приступай к заполнению первого листа своего альбома!



Подготовка памяти

- Подключение библиотек.
- Объявление переменных.
- Переобозначение аналоговых входов.
- Объявление датчиков.

1. Подключи библиотеки к данному скетчбуку:

1. `/*БИБЛИОТЕКИ*/`
2. `#include <Wire.h> //Для работы с шиной SPI.`
3. `#include <Adafruit_BMP085.h> //Для улучшенной версии — датчика bmp185.`
4. `#include <Adafruit_Sensor.h> //Для использования функций датчиков Adafruit.`
5. `#include <DHT.h> //Две библиотеки для датчика влажности и температуры DHT11.`
6. `#include <DHT_U.h>`
7. `#include <MQ135.h> //Библиотека для измерения уровня CO2 в воздухе (и не только).`

SPI — это двунаправленная шина передачи данных, которая используется датчиками BMP180 и DHT11 для обмена информацией с Arduino. Именно SPI обеспечивает высокую скорость реагирования и измерений.

Ты подключил библиотеку для датчика BMP085, выпускаемого компанией Adafruit. Он является устаревшей моделью нашего датчика. BMP085 имеет более низкие физические характеристики, включая точность, однако не отличается по логике работы.

2. Теперь объяви и инициализируй некоторые переменные. Все переменные, находящиеся вне функций, являются глобальными, т. е. доступны из любого места программы и их значение постоянно сохраняется в памяти устройства.

8. `/*ПЕРЕМЕННЫЕ*/`
9. `int audio = 7; //Цифровой контакт, к которому подключён аудиомодуль.`
10. `int DHTsensor = 4; //Контакт, к которому подключён DHT11 — датчик влажности.`
11. `float pressure = 0; //Переменная для хранения данных о давлении.`
12. `float temper1 = 0; //Так как имеется два датчика, способных измерять температуру,`
13. `float temper2 = 0; //будет брать среднее значение их показаний.`
14. `float temper = 0; //Для итогового значения температуры.`
15. `float humidity = 0; //Влажность в процентах.`
16. `float CO2 = 0; //Для сохранения данных о концентрации углекислого газа.`
17. `double illumination = 0; //Освещённость в люксах.`

Для обозначения контактов используются целочисленные значения (`int`), для хранения показаний — вещественные (`float`), так как для исследований бывает важно знать точное измерение. Освещённость ты будешь находить по сложной формуле, работающей со степенями и делением вещественных чисел, поэтому лучше выделить под неё место как под большое (двойное) вещественное число — **double**.

3. Переобозначь аналоговые входы. Это упростит задачу оперирования с ними и чтение скетча в будущем.

```
18. /*Для удобства следует переобозначить аналоговые входы, назвав их по датчикам.*/  
19. #define photo A2 //Для фоторезистора (датчик освещённости Troyka Module).  
20. #define gas A3 //Для MQ135 — анализатора газа.
```

Встроенная функция **define** имеет следующую конструкцию: `#define новоеИмя староеИмя`

Она позволяет, например, избежать нудных и длительных поисков и изменений номеров контактов по всему коду в случае перестановки датчика — достаточно заменить номер в строке переопределения.

4. Остался последний шаг в этой части программы — дать имена датчикам. Внимательно ознакомься с комментариями.

```
21. /*В качестве новых типов объектов могут быть представители классов,  
22. описанных в подключённых библиотеках.*/  
23. Adafruit_BMP085 bmp; //Так как датчик BMP185 — улучшенный вариант датчика BMP085.  
24. DHT_Unified dht(DHTsensor, DHT11); //Для датчика DHT необходимо указать вход и тип датчика.  
25. MQ135 gasSensor = MQ135(gas); //Датчик газа — gasSensor, в скобках — вход.
```

Отлично! Ты закончил с первой частью основной программы. Переходи к функциям, которые есть в любой программе для Arduino, — функции установки (**setup**) и повтора (**loop**).

Запуск устройства (функция установки)

- Запуск последовательного порта.
- Определение ролей цифровых контактов.
- Запуск датчиков.

Приступим!

5. Введи заветное название функции и открой фигурные скобки. Для работы с последовательным портом через Bluetooth тебе не потребуются изменять стандартную строчку, поскольку на программном

уровне протокола UART¹ передача данных осуществляется точно так же, как по USB-кабелю.

Последовательный порт в Arduino обозначают словом **Serial**. Для соединения по USB или Bluetooth оптимальным является значение скорости 9600.

```
26. void setup() {
27.     Serial.begin (9600); //UART. В данном случае выводится на Bluetooth.

6.  Определи роли цифровых контактов. В отличие от аналоговых пор-
    тов, которые могут быть только входами, цифровые могут выпол-
    нять роль как входов, так и выходов.

28.     pinMode(audio, OUTPUT); //Для включения динамика порт используется
    как выход.
29.     pinMode(DHTsensor, INPUT); // Так как показания будут считываться, порт
    используется как вход.

7.  В функции установки осталось только запустить работу датчиков.
    Для этого используй функцию begin():

30.     dht.begin(); //Инициализация датчика влажности.
31.     bmp.begin(); //Инициализация датчика давления.
32.     sensor_t sensor; //Указатель на объект. Из-за того, что библиотека
    написана на «чистом» C.
33. } //Не забудь закрыть фигурную скобку!
```

Итак, при запуске программы порты будут работать как положено и одновременно начнётся работа датчиков. Можно приступить к написанию основной функции!

Основная функция (повтора)

- Ограничение измерений.
- Получение информации от датчиков.
- Включение тревоги и вывод результатов.

Для чего нужно ограничение измерений, если мы о нём нигде раньше не упоминали? Это маленькая хитрость. Дело в том, что Arduino по умолчанию передаёт показания только после подключения другого устройства по последовательному порту и наличии от него команды. Мы же хотим считывать показания через определённый промежуток времени, а не только по запросу.

¹ UART (УАПП — универсальный асинхронный приёмопередатчик) — старейший и самый распространённый протокол передачи данных. Для компьютера и Uno протокол UART является «международным языком общения», как для тебя и иностранца — английский.

8. Добавь необычное ограничение. В чём его необычность? Оно до-
словно звучит так: «Работай всегда».

```
34. void loop() {  
35.   while (true) { //Иначе программа реагирует только после включения порта.
```

9. Далее тебе предлагается немного схитрить. В основной функции
напиши вызов функций по названию, а сами функции ты задашь
потом в соответствующей подпрограмме:

```
36.   DHTinfo(); //Для получения информации от датчика влажности.  
37.   BMPinfo(); //Для получения информации от датчика давления.
```

10. Сохрани значения других датчиков в подготовленные переменные.

```
38.   illumination = light(); //Освещённость — результат работы функции light()  
    из подпрограммы.  
39.   temper = temperature(temper1, temper2); //Температура будет  
    рассчитываться из двух показателей специальной функцией.  
40.   CO2 = gasSensor.getPPM(); //Это единственная строка, где используется  
    уже готовая функция из библиотеки.
```

Кстати, внимательно рассмотри последнюю функцию. Wiring — это
объектно-ориентированный язык программирования. В строке 40 (как
и в строках 27, 30, 31) сначала указывается объект, затем после разде-
лителя (точки) — функция, которая к нему применяется.

11. Теперь выведи результаты. Функция тревоги (**alarm**) должна сраба-
тывать при превышенных значениях, значит, на вход к ней нужно
подавать именно эти переменные:

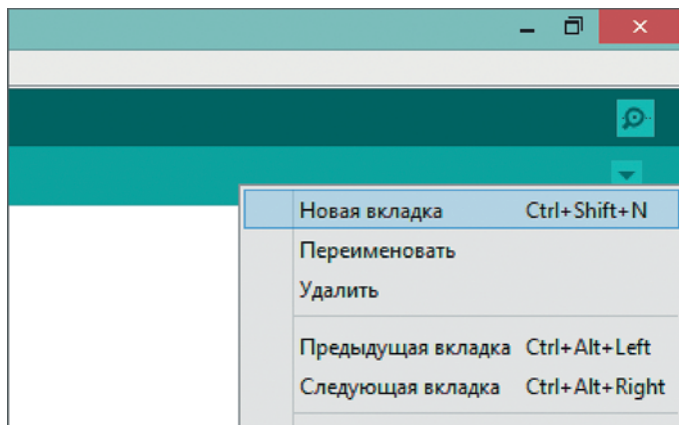
```
41.   alarm(temper, pressure, humidity, CO2, illumination); //Функция тревоги.  
    Она проверяет значения температуры, давления, влажности,  
    загазованности и освещённости.  
42.   printResult(); //Вывод результатов. Это именно те значения, которые  
    печатаются на экране постоянно.  
43.   bluetooth(); //Для вывода конкретных значений по переданной  
    по Bluetooth команде.
```

12. Ура! В основной программе осталось написать всего три строчки!
Быстро дописывай их и начинай творить функции:

```
44.       delay(10000); //Пауза 10 секунд.  
45.   } //Закрытие while()  
46. } //Закрытие loop()
```


Функция паузы в качестве аргумента использует значение в миллисекундах. Перейдём к подпрограмме, работающей непосредственно с датчиками.

13. Для начала создай новый скетч внутри скетчбука. Для этого требуется новая вкладка, создаваемая через меню в правой верхней части экрана:



Назови новую вкладку **sensors**. Автоматически в папке проекта появится файл **sensors.ino**. Подпрограмма может работать с глобальными переменными основной программы или других подпрограмм, поэтому тебе не придётся заново подключать библиотеки и объявлять переменные.

Подпрограмма работы с датчиками (sensors)

- Проверка освещённости.
- Датчик DHT.
- Датчик BMP.
- Подсчёт средней температуры.

14. Начни с проверки освещённости. Для этого потребуется работать с датчиком освещённости Тройка Module, который является аналоговым. В чём состоит трудность? Аналоговые датчики имеют большую зависимость от помех, поэтому измерения надо считывать как средние показатели.

Все предыдущие функции начинались с **void** («пустое») — это значит, что после выполнения функции не будет ничего возвращаться в качестве значения. Для функции **light()** удобнее будет возвращать

полученное точное значение в люксах, поэтому будем использовать тип **double**.

```
1. /*ПРОВЕРКА ОСВЕЩЁННОСТИ*/
2. double light() {
3.     int lightSensor = analogRead(photo); //Объявление переменной
        для показаний датчика освещённости.
4.     long valSensor = 0; //Создаём большую выборку и проверяем среднее
        значение, чтобы «отсечь» показания с помехами. Эта переменная — место
        для среднего значения.
5.     for (int i = 0; i < 9; ++i) { //Пусть выборка будет состоять из девяти значений.
6. //Эта константа даст оптимальный результат. Можно попробовать подобрать
        и другую.
7.         valSensor = valSensor + analogRead(lightSensor); //Суммируем показания
            датчика (он показывает отношение сопротивлений в измерительном
            делителе).
8.     }
9.     valSensor = (valSensor / 9); //Находим среднее арифметическое.
```

Ты получил среднее значение нескольких показаний, однако эти числа достаточно абстрактны — они показывают уровень. Аналоговый сигнал Arduino может принимать значения от 0 до 1023 в зависимости от поступающего от датчика напряжения: от 0 V до 5 V.

15. Переведи полученные данные в люксы, чтобы сравнивать показания в помещении с принятыми нормами.

Значение, показываемое датчиком, рассчитывается по формуле

$$x = \frac{R_{\max} \times R}{R + R_2},$$

где x — показания датчика освещённости, R_{\max} — максимальное сопротивление на плате Arduino Uno, R — сопротивление фоторезистора, R_2 — сопротивление после фоторезистора в электрической цепи датчика.

Известно, что резистор датчика Tpyuka Module имеет постоянное сопротивление 10 кОм, а максимальное по счёту показание датчика — 1024 (так как нумерация идёт от 0 до 1023). Вычисли значение на встроенном в датчик фоторезисторе как $\text{valSensor} = (1024 * R) / (R + 10000)$.

Легче преобразовать формулу вручную:

```
10. double R = 10000 * valSensor / (1024 - valSensor);
```

16. На самом деле зависимость освещённости от полученного показателя определяется нелинейно (через логарифмы, которые ты изучишь позднее) с долгим выводом и подсчётом константы для данного датчика. Воспользуйся готовой формулой, в которую будет подставляться найденное значение R:

```
11. double lux = 255.84 * 1 / (pow(R, (10 / 9)) * 0.0003);
```

17. Отлично! Осталось это значение вернуть как результат работы функции **double light()**. Следовательно, нужно вернуть (по-английски return) значение типа **double**:

```
12.     return lux;  
13. } //Не забываем закрывать скобку!  
14.
```

Ура! Ты закончил с датчиком освещённости. Пора переходить к проверке влажности.

18. Для взаимодействия с датчиком DHT используется библиотека, написанная не на Wiring, а на «чистом» C. Пусть функция **DHTinfo()** ничего не возвращает в качестве значения, сразу выводя в последовательный порт нужные сведения.

```
15. /*ДАТЧИК DHT*/  
16. void DHTinfo () { //Функция возвращает «пустоту».  
17.     sensors_event_t event; //Специфический тип «событие» (event), который  
        нужен для инкапсуляции определённого показания датчика. Получение  
        данных от датчика в определённый момент времени.  
18.     dht.temperature().getEvent(&event); //Получение от объекта (датчика)  
        температуры по адресу текущего момента (адрес ячейки памяти, где  
        хранится состояние).  
19.     temper1 = event.temperature; //Сохранение текущей температуры  
        в качестве первого значения.  
20.     dht.humidity().getEvent(&event); //Сохранение показаний влажности (как  
        числа) в текущий момент.  
21.     humidity = event.relative_humidity; //Пересохранение показания в виде  
        процентов в глобальную переменную.  
22. } //Не забываем закрывать скобку!
```

Инкапсуляция — это один из важнейших принципов объектно-ориентированного программирования. Он заключается в «запаковке в одну капсулу» всех функций, другого кода и данных, касающихся одной области, что позволяет легко оперировать с состоянием объекта (совершить с ним действие или узнать характеристики в данный или любой другой момент времени).

19. Напиши функцию для датчика измерения давления. Здесь всё намного проще:

```
23. /*ДАТЧИК BMP*/  
24. void BMPinfo () { //Снова «пустой» возврат.  
25.     temper2 = bmp.readTemperature(); //В качестве второго значения  
        температуры сохраняется считанное с датчика BMP показание.  
26.     pressure = (bmp.readPressure() / 133.3); //Функция из библиотеки  
        предоставляет значение в Па, вместо используемых в России мм рт. ст.  
27. } //Закрываем скобку!  
28.
```

20. Составь функцию подсчёта итогового значения температуры. На её вход будут подаваться два значения, полученные с датчиков. Результат работы этой функции будет заноситься в вещественную глобальную переменную **temper**, следовательно, функция должна «возвращать» значение типа **float**:

```
29. /*СРЕДНЯЯ ТЕМПЕРАТУРА*/  
30. float temperature (float a, float b) { //На вход подаются два любых  
        вещественных числа. Эту функцию можно будет применить не только  
        к имеющимся temper1 и temper2. Она описывает действия с некоторыми  
        абстрактными объектами определённого типа.  
31.     float c = (a + b) / 2; //Среднее арифметическое двух показаний.  
32.     return c; // Возвращение результата как вещественного числа.  
33. } // Не забываем закрывать скобки!
```

Ты закончил писать подпрограмму для работы с датчиками. Молодец! Самая важная часть позади, осталось научить твою экостанцию выводить результаты.

Подпрограмма вывода результатов (results)

- Включение оповещения.
- Вывод отдельных результатов по требованию (команде).
- Вывод результатов для обработки в таблице.

21. Создай новую вкладку и назови её **results**. Теперь в твоём скетчбуке должно быть три вкладки: **ecology**, **sensors**, **results**.

22. Начни с функции включения оповещения. Она будет подавать сигнал на звуковой модуль для проигрывания звукового сообщения, хранящегося в его памяти. Значит, функция не должна ничего возвращать, поэтому её тип **void**.

1. /*ФУНКЦИЯ ВКЛЮЧЕНИЯ ОПОВЕЩЕНИЯ*/
2. void alarm (float a, float b, float c, float d, float e) {

Что подаётся на её вход? Снова абстрактные переменные, вместо которых в основной программе ты отправлял в функцию показания температуры (вместо a), давления (вместо b), влажности (вместо c), содержания CO₂ (вместо d) и освещённости (вместо e).

23. С помощью условного оператора задай параметры, при которых будет срабатывать оповещение:

3. if ((a < 20.0) || (a > 28.0) || //Если вне нормы показания температуры,
4. (b < 328.0) || (b > 370.0) || //давления,
5. (c < 30.0) || (c > 70.0) || //влажности,
6. (d > 800.0) || //CO2,
7. (e < 50) //недостаточно освещённости, то
8.) {

Числа приведены для примера. Измени их в соответствии с нормами для помещения, где будет стоять твоя экостанция.

24. Осталось заполнить тело условного оператора. Для этого придётся воспользоваться специальной функцией из стандартной библиотеки Arduino **digitalWrite()** — записью цифрового сигнала на конкретный цифровой порт. В качестве аргументов для неё указывается номер порта и передаваемое значение. Цифровой сигнал может принимать значение **логический ноль** (LOW, 0 V) или **логическая единица** (HIGH, 5 V).

9. digitalWrite(audio, HIGH); // Вместо номера порта — его название, благодаря define. Подача тока на порт для включения аудиозаписи.
10. delay(100); // Пауза в 100 миллисекунд.
11. digitalWrite(audio, LOW); // Передача команды закончена.
12. } //Закрываем скобки условного оператора.
13. } //Закрываем скобки тела всей функции.

Кстати! На самом деле ATmega328 поддерживает передачу цифровых сигналов не двух, а четырёх видов, где два дополнительных вида — специальные состояния регистров микроконтроллера. Чтобы ими воспользоваться, необходимо прописать функцию самостоятельно, а не использовать готовую.

Так как будет использоваться контакт PLAYE звукового модуля, во время всего воспроизведения подавать ток (аналог нажатия кнопки) не требуется.

25. Напиши функцию вывода результатов в зависимости от полученной от смартфона по Bluetooth команды. Для удобства пусть команды будут обозначаться цифрами:

Команда	Информация
1	Температура
2	Давление
3	Влажность
4	Уровень углекислого газа
5	Освещённость

Назови функцию **bluetooth()**. Тебе снова придётся столкнуться со специфическими функциями для объекта **Serial** (последовательный порт). Чаще всего используются следующие из них:

Функция	Значение
Serial.begin()	Открытие последовательного порта. В качестве аргумента задаётся скорость последовательного интерфейса в бодах (кстати, 9600 бод означает, что 1 бит информации передаётся за 104 микросекунды)
Serial.print()	Печать. Вывод данных из последовательного порта
Serial.println()	Вывод данных из последовательного порта с последующим автоматическим переводом текстового курсора (каретки) на новую строку
Serial.read()	Считывание одного байта из последовательного буфера
Serial.write()	Побайтовый вывод данных из последовательного буфера

Таким образом, твоя функция будет выглядеть следующим образом:

```
14. /*ВЫВОД ОТДЕЛЬНЫХ РЕЗУЛЬТАТОВ*/
15. void bluetooth() {
16.     int bluetoothData = Serial.read(); //Переменная для хранения считанной
        с последовательного порта команды.
17.
18.     if (bluetoothData == '1') { //Если было отправлено «1», то выводится
        информация о температуре.
19.         Serial.print("Temperature is ");
20.         Serial.print(temper);
21.         Serial.println(" C.");
22.     };
```

```

23.   if (bluetoothData == '2') { //Если было отправлено «2», то выводится
      информация о давлении.
24.       Serial.print("Pressure is ");
25.       Serial.print(pressure);
26.       Serial.println(" mm.");
27.   };
28.   if (bluetoothData == '3') { //Если было отправлено «3», то выводится
      информация о влажности.
29.       Serial.print("Humidity is ");
30.       Serial.print(humidity);
31.       Serial.println(" %.");
32.   };
33.   if (bluetoothData == '4') { //Если было отправлено «4», то выводится
      информация об уровне углекислого газа.
34.       Serial.print("CO2-level is ");
35.       Serial.print(CO2);
36.       Serial.println(" ppm.");
37.   };
38.   if (bluetoothData == '5') { //Если было отправлено «5», то выводится
      информация об освещённости.
39.       Serial.print("Illumination level is ");
40.       Serial.print(illumination);
41.       Serial.println(" lx.");
42.   };
43. }

```

26. Ура! Осталась последняя функция, необходимая для серьёзных исследований микроклимата помещений, которые ты задумал.


Мы уже упоминали, что стандартная функция **Serial.print()** может принять только один аргумент. Чтобы вывести строку, содержащую несколько элементов, приходится вызывать функцию несколько раз.

Выводить с помощью этой функции всю строку можно! Но для этого мы применим маленькую хитрость. Создай переменную типа **String** (строка), в которую объединишь все остальные переменные и знаки. Складывать можно переменные только одного типа, поэтому придётся применить *явное преобразование типов данных*. На языке программирования Wiring оно производится одноимённой с желаемым типом данных функцией, например **String(temper)** преобразует значение вещественной переменной **temper** в строку. Таким образом, в одну строку ты соберёшь все необходимые показания, разделяемые запятой. Кстати, строку необходимо начинать тоже с запятой. Для чего? Это ты узнаешь в одном из следующих этапов!

```
44. /*ВЫВОД РЕЗУЛЬТАТОВ ДЛЯ ОБРАБОТКИ В ТАБЛИЦЕ*/
45. void printResult() {
46.     //В качестве разделителя табличным процессором при импорте
        используется запятая.
47.     String answer = ", " + String(temper) + ", "
48.                   + String(pressure) + ", " + String(humidity) + ", "
49.                   + String(illumination) + ", " + String(CO2);
50.     Serial.println(answer);
51. }
```

Молодец! Ты закончил писать программный код. Теперь дело за малым!

Нажми на кнопку , чтобы проверить, не допустил ли ты ошибок.

Исправь их, если они есть. Если ошибок нет, нажми на кнопку . Назови скетч **ecology.ino**, если сохранение по какой-либо причине не было сделано ранее.


Если появилась ошибка загрузки, но ты уверен, что не забыл подключить кабель, то проверь переключатель на плате Wireless Shield.

Умница! Ты закончил программировать! Давай переходить к загрузке и тестированию.



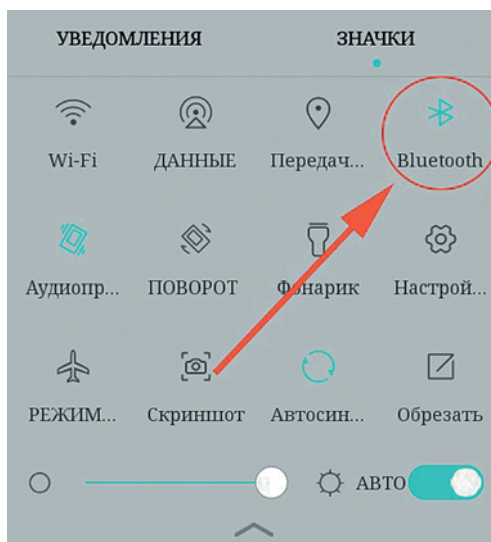
Этап 7. Загрузка программы и её тестирование

ШАГ 1. ЗАГРУЗКА ПРОГРАММЫ В МОДУЛЬ ARDUINO UNO

1. Подключи Arduino Uno с помощью USB-кабеля к компьютеру. Убедись, что программная среда обнаружила устройство.
2. Нажми на кнопку , чтобы произвести проверку, скомпилировать и отправить программу на устройство.

ШАГ 2. ТЕСТИРОВАНИЕ

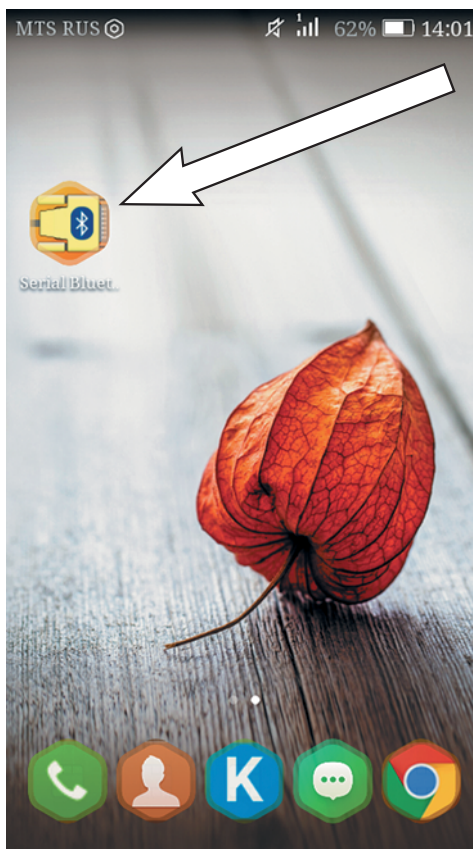
1. Программа запустится сразу же после загрузки. В качестве питания Arduino будет использовать подключение к компьютеру. После включения устройство переходит в режим ожидания открытия порта, т.е. подключения телефона или планшета. Включи на своём гаджете Bluetooth. Обычно для этого необходимо нажать на иконку с символом Bluetooth в верхней панели или виджете на рабочем столе.

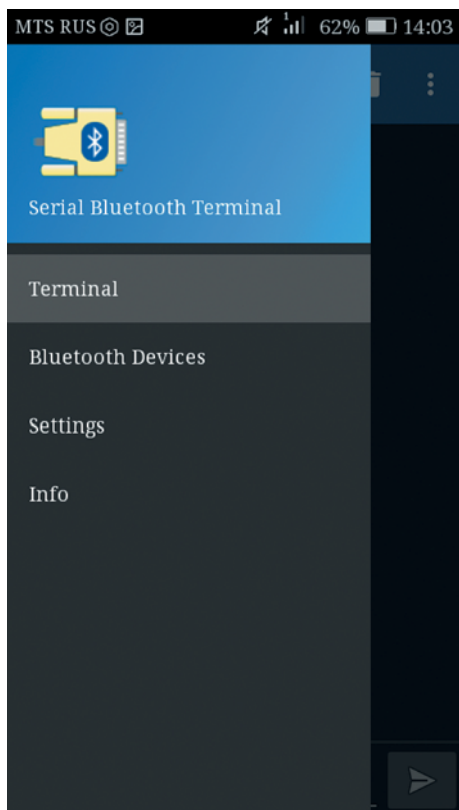


2. Настрой сопряжение с модулем Bluetooth Bee. Для этого в открывшемся меню выбери **Bluetooth_Bee_V2**. В открывшемся окне введи стандартный код 1234. Отлично!

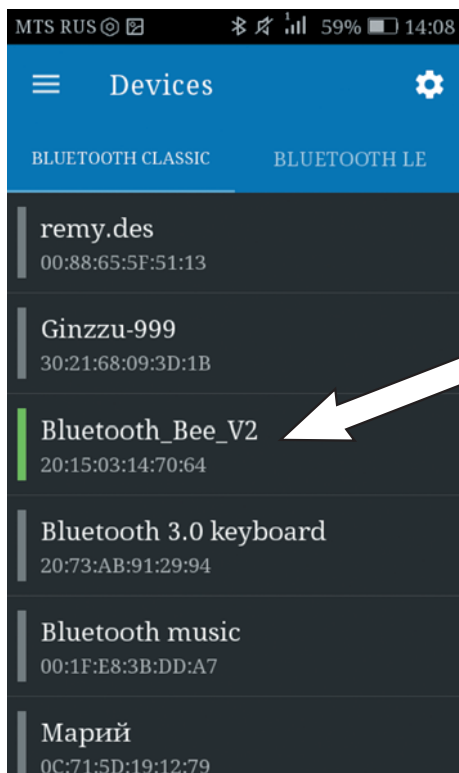


3. Запусти приложение **Serial Bluetooth Terminal** или любое аналогичное, которое ты установил, с рабочего стола своего смартфона.





4. Открой боковое меню. Для этого ты можешь нажать на иконку меню или же сдвинуть шторку слева.

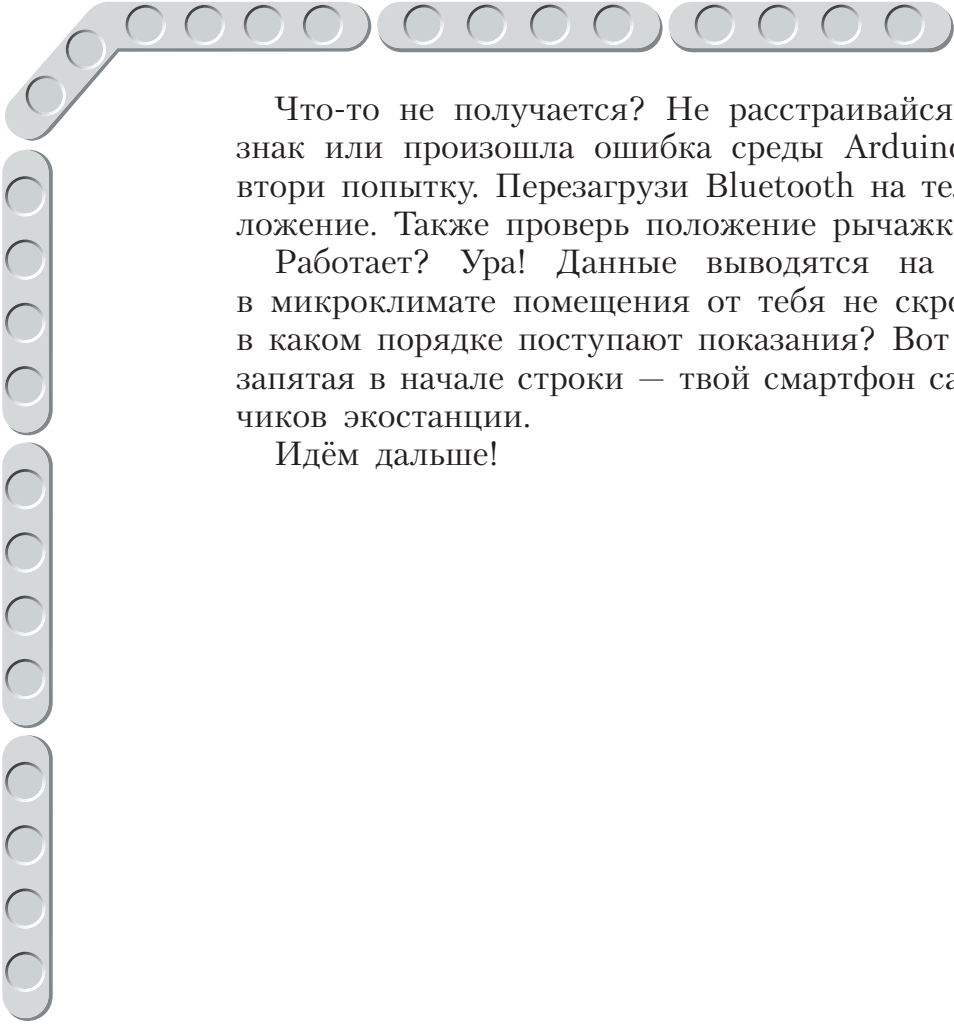


5. Выбери пункт **Bluetooth Devices**. Наш модуль будет отображаться во вкладке Bluetooth Classic, так как он использует старую версию технологии (неэкономичную). Выбери модуль. Он должен быть отмечен зелёным индикатором.

6. Перейди во вкладку Terminal в основном меню и нажми на иконку соединения. На экране появятся показания.

```
704.55
17:30:07.378 , 27.50, 736.89, 49.00, 297.77
, 685.02
17:30:17.850 , 27.50, 736.99, 49.00, 22.00,
685.02
17:30:28.324 , 27.50, 736.99, 49.00, 788.17
, 685.02
17:30:38.798 , 27.50, 736.90, 49.00, 313.53
, 685.02
17:30:49.272 , 27.50, 736.89, 49.00, 32.73,
685.02
17:30:59.629 , 27.45, 737.10, 49.00, 22.93,
685.02
17:31:10.220 , 27.50, 736.98, 50.00, 32.25,
665.85
17:31:20.504 , 27.45, 736.86, 50.00, 797.01
, 665.85
17:31:31.168 , 27.50, 736.86, 49.00, 32.73,
647.04
17:31:41.642 , 27.45, 736.98, 48.00, 25.12,
628.57
17:31:52.116 , 27.45, 736.90, 48.00, 815.30
, 628.57
17:32:02.590 , 27.45, 736.92, 48.00, 834.44
, 592.68
17:32:12.565 , 27.45, 737.07, 48.00, 25.54,
628.57
17:32:23.006 , 27.45, 736.96, 48.00, 824.76
, 628.57
17:32:33.513 , 27.95, 736.98, 47.00, 32.89,
610.45
```

7. Поочерёдно отправь через поле ввода все указанные в коде цифровые команды и посмотри результат их выполнения на экране. Реакция может быть с задержкой, поскольку данные отправляются постоянно.
8. Сравни показания датчика давления с данными погодных информеров, а выведенную температуру — с показаниями домашнего термометра.
9. К сожалению, возможность сравнить показания по содержанию углекислого газа с показаниями, полученными с помощью лабораторного оборудования, представляется редко, однако ты можешь сделать несколько замеров до и после проветривания и сравнить их с нормой.

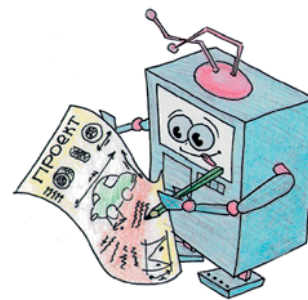


Что-то не получается? Не расстраивайся. Возможно, ты пропустил знак или произошла ошибка среды Arduino IDE. Проверь код и повтори попытку. Перезагрузи Bluetooth на телефоне и перезапусти приложение. Также проверь положение рычажка на плате Wireless Shield.

Работает? Ура! Данные выводятся на экран! Теперь изменения в микроклимате помещения от тебя не скроются! Кстати, ты заметил, в каком порядке поступают показания? Вот и ответ, зачем требовалась запятая в начале строки — твой смартфон сам выступает одним из датчиков экостанции.

Идём дальше!

Этап 8. Обработка полученных результатов




У тебя есть устройство, способное выполнять важные измерения. Ты можешь подключиться к нему из любой точки здания или передать показания через Сеть друзьям или родственникам. Это замечательно, но давай превратим эти, пока что абстрактные, данные в полезную информацию.

Тебе понадобятся:

- компьютер;
- табличный процессор (например, Open Office Calc или Excel).

1. Подготовь экостанцию к передаче. Включи устройство и выполни подключение по Bluetooth. Запусти программу **Bluetooth Serial Terminal** (или аналогичную) и после соединения нажми на «Очистить

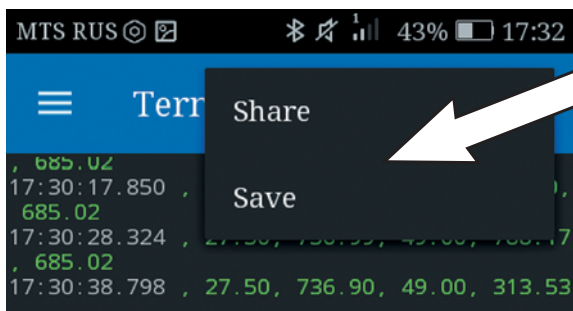
экран» . Это позволит сохранить только нужную информацию, без описания самого подключения.

2. Оставь экостанцию выполнять измерения в течение желаемого времени. Все результаты будут непрерывно транслироваться на экран телефона.

3. Для того чтобы сохранить данные, воспользуйся специальной функцией в программе на смартфоне. Нажми на три точки на панели инструментов. Тебе будет предложено два варианта: **Share** («Поделиться») и **Save** («Сохранить»).

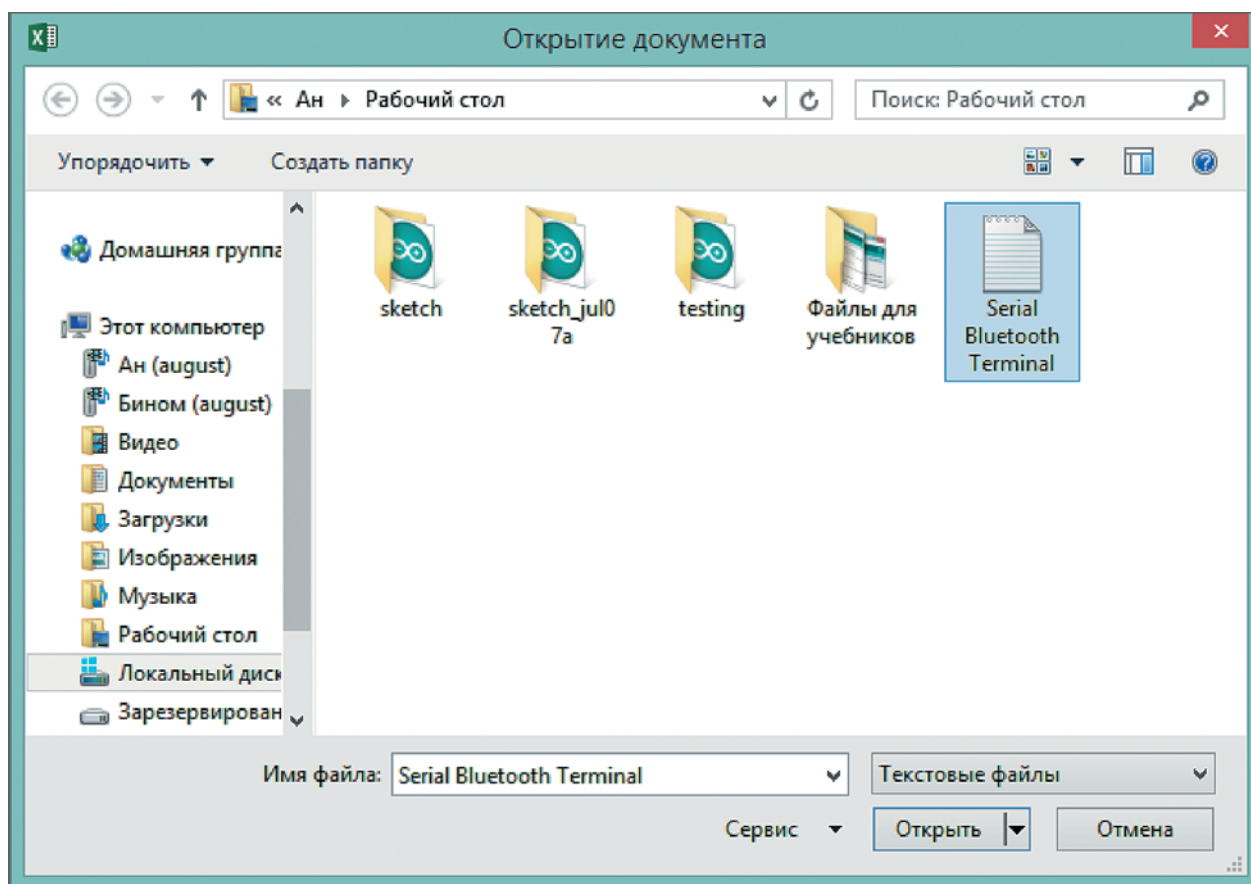
Первая функция копирует весь лог, доступный на экране телефона, в буфер обмена, откуда предоставляет возможность отправить показания по почте или другими видами сообщений. Вторая функция сохраняет показания (вместе со временем) в txt-файл в корневой каталог карты памяти телефона.

Также можно выделить вручную необходимые строчки и скопировать их.

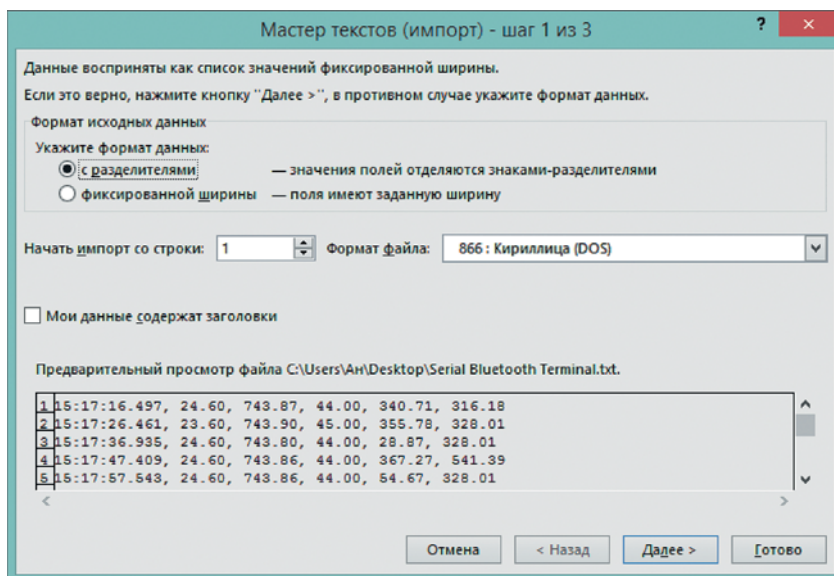


Сохрани текстовый документ на карту памяти и перенеси его на компьютер.

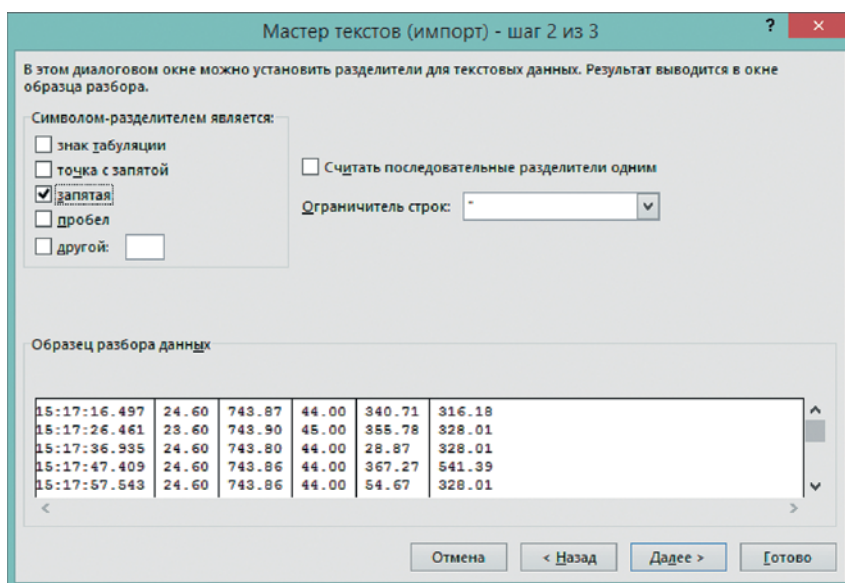
4. Запусти на своём ПК любой табличный процессор. Для демонстрации мы используем Microsoft Excel 2016, однако ты можешь выполнить аналогичные действия в OpenOffice Calc или любом другом свободно распространяемом ПО этого типа.
5. Открой текстовый файл. Для этого выбери меню «Открыть» и в графе «Тип файлов» — «Текстовые файлы».



6. Автоматически запустится «Мастер импорта текстовых файлов». Он автоматически определит, каким образом были отформатированы данные. Убедись, что выбран формат «с разделителями», и импорт начнётся с первой строки, так как ты уже убрал лишние данные.



7. Определи конкретный тип разделителей. Ты использовал разделитель-символ. Какой? Конечно же, запятую. Обрати внимание на ограничитель строк. По умолчанию им является непечатный символ перевода каретки. Убедись, что в образце данные отображаются в правильном формате.



	Calibri	11	A ⁺	A ⁺	%	000		7:16.497
	Ж	К						
1	✂	Вырезать		44.00	340.71	316.18		
2	📄	Копировать		45.00	355.78	328.01		
3	📄	Параметры вставки:		44.00	28.87	328.01		
4	📄			44.00	367.27	541.39		
5	📄			44.00	54.67	328.01		
6	📄	Специальная вставка...		44.00	28.87	340.13		
7	📄			44.00	993.94	378.22		
8	📄	Вставить		45.00	447.43	352.54		

8. Отлично! Данные были импортированы и разбиты по столбикам. Наведём красоту? С помощью команды «**Вставить**» добавь новую строку перед первой и занеси в неё заголовки столбцов.

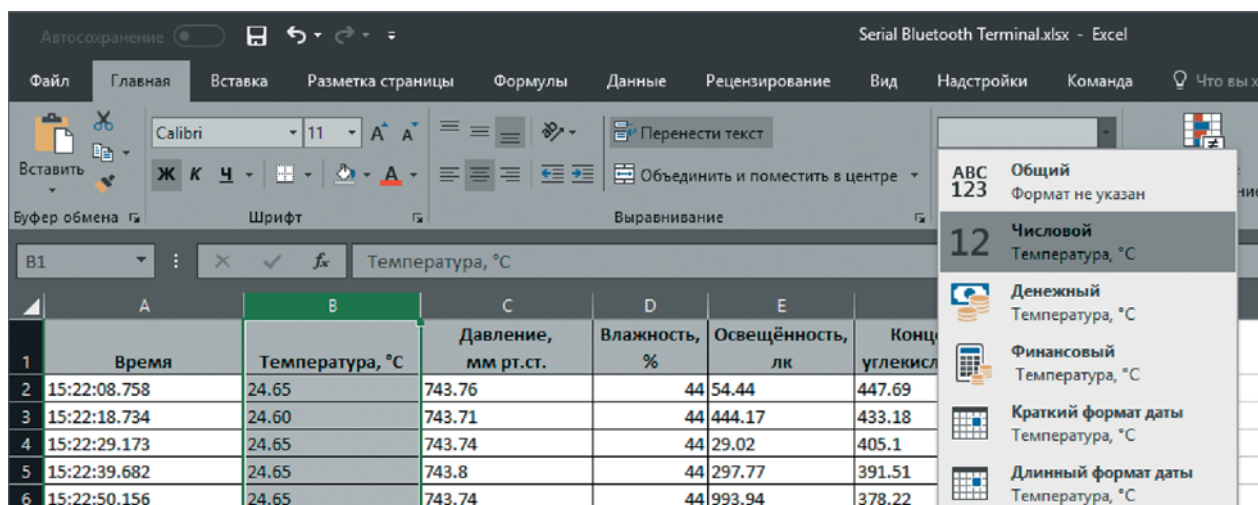
9. Оформи таблицу, выделив заголовки и добавив границы. Основная работа сделана!

A1							Время
	A	B	C	D	E	F	G
1	Время	Температура, °C	Давление, мм рт.ст.	Влажность, %	Освещённость, лк	Концентрация углекислого газа, мд	
2	15:22:08.758	24.65	743.76	44	54.44	447.69	
3	15:22:18.734	24.60	743.71	44	444.17	433.18	
4	15:22:29.173	24.65	743.74	44	29.02	405.1	
5	15:22:39.682	24.65	743.8	44	297.77	391.51	
6	15:22:50.156	24.65	743.74	44	993.94	378.22	
7	15:23:00.548	24.65	743.72	44	355.78	378.22	

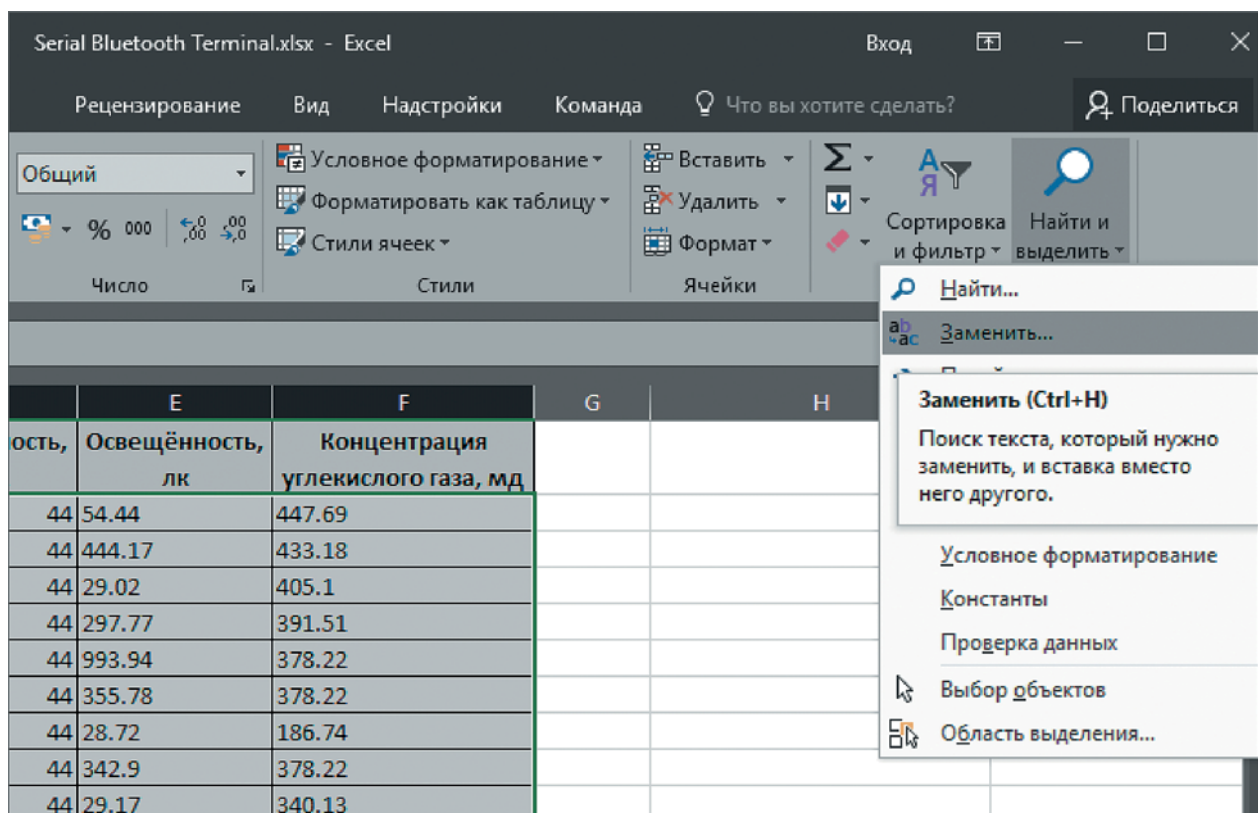
10. Тебе не кажется, что некоторые данные отображаются странно? Выдели столбики и назначь им следующие форматы данных:

Столбец	Формат
Дата и время	Время
Температура	Числовой
Давление	
Влажность	
Освещённость	
Концентрация углекислого газа	

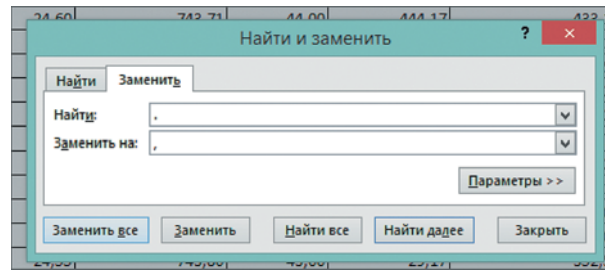
Для этого кликни по названию столбца (например, В) и выбери на главной вкладке панели инструментов формат данных.



11. Упс! Осталась одна небольшая проблема: Excel использует в качестве разделителя для вещественных чисел запятую, а Arduino — точку. Не отчаивайся, ты её быстро решишь! Выдели числовые данные всех столбцов, кроме времени, и открой меню «**Найти и выделить**» на вкладке «Главная». Найди пункт «**Заменить**».



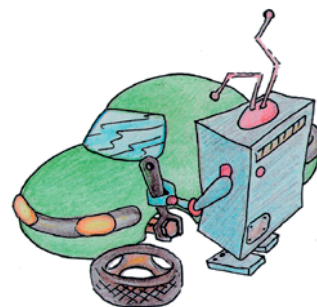
Тебе потребуется найти все точки и заменить их на запятые:



12. Последний штрих! Датчик освещённости в силу своих невысоких технических характеристик иногда выдаёт очень странные значения из-за бликов, случайно набежавших туч или просто ошибок, свойственных аналоговым датчикам. Замени сильно выделяющиеся показания (например, 12 000 лк при 200–300 лк в остальных ячейках) на показание из предыдущей ячейки.

Вот и всё! Твои данные подготовлены к анализу и обработке средствами табличного процессора.

Этап 9. Использование для реальных задач

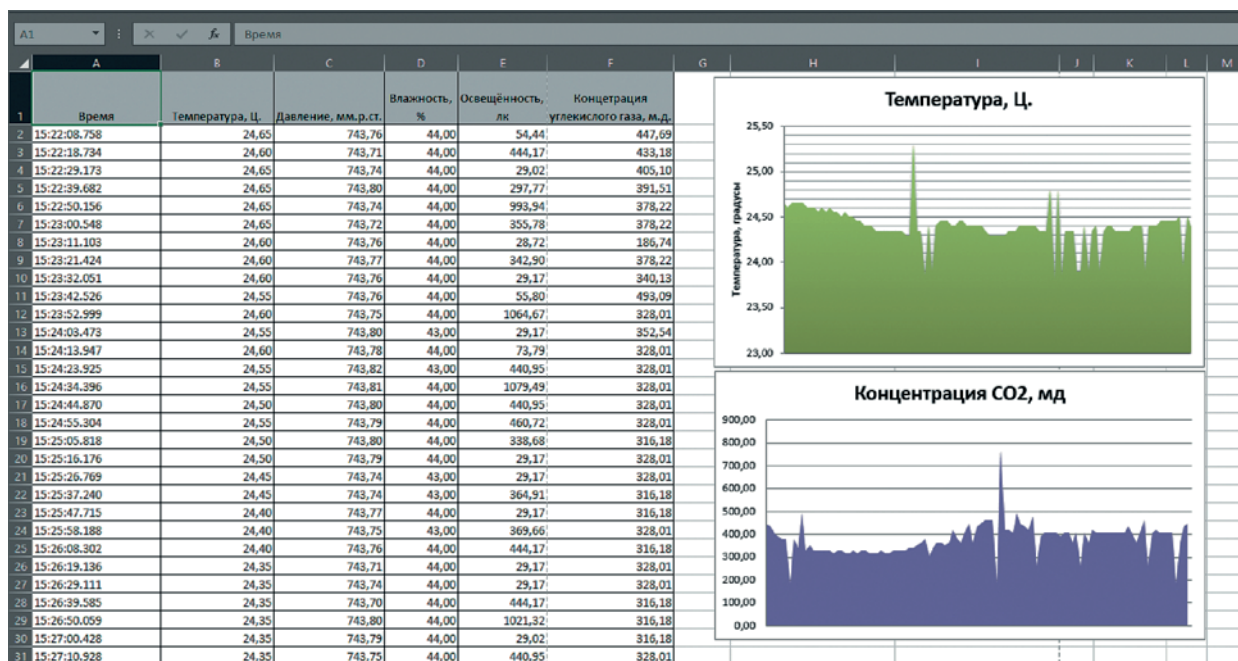


Ты можешь:

- вычислить средние показания за этот промежуток времени;
- найти пик активности;
- исследовать изменение качества воздуха и температуры с помощью построения линии тренда и выявить примерную формулу, описывающую изменения;
- построить графики и диаграммы для представления графического отчёта;
- импортировать данные в базы данных школы

и другое.

Например, результатом нашей работы стал небольшой файл с наглядными графиками, по которым мы смогли лично убедиться, что концентрация углекислого газа снижается при проветривании помещения:



У тебя осталось столько неисследованных помещений. Вперёд, эколог!



А теперь...

С помощью собранной экостанции ты можешь проверить любое помещение. Чтобы результаты были по-настоящему полезными, изучи текст указанного в начале книги стандарта и сравни его с утверждёнными нормами в других странах.

Ты можешь использовать это устройство как инструмент для своих практических или исследовательских проектов. Не бойся экспериментировать и узнавать. Ты также можешь расширить возможности станции, добавив дополнительные датчики, например датчик влажности почвы и аэрометр. А если немного изменить код, то даже с имеющимися датчиками ты сможешь измерить высоту над уровнем моря и содержание различных газов в воздухе.

Как ты применишь полученные знания и возможности, эколог?

До новых встреч!

Ты собрал своими руками настоящую миниатюрную экологическую станцию, научился измерять с её помощью влажность, температуру, атмосферное давление, познакомился с понятием микроклимата помещений. Но впереди ещё так много интересного! Книги серии «Робофишки» познакомят тебя с другими замечательными проектами и сделают из тебя настоящего изобретателя!



Содержание

Здравствуйте!	3
Дорогой друг!	4
Экология жилища	5
 Этап 1. Устройство экостанции	 12
 Этап 2. Сборка экостанции	 13
Шаг 1. Подключение модуля беспроводной связи	13
Шаг 2. Подключение датчика атмосферного давления	15
Шаг 3. Подключение датчика влажности DHT11	17
Шаг 4. Подключение датчика газа	18
Шаг 5. Подключение датчика освещённости	20
Шаг 6. Подключение звукового модуля	22
 Этап 3. Установка программного обеспечения	 26
 Этап 4. Первый запуск и проверка оборудования	 28
 Этап 5. Создание декоративного корпуса экостанции.	 33
 Этап 6. Создание программы для устройства	 36
Шаг 1. Запуск программного обеспечения Arduino IDE	37
Шаг 2. Составление программы для экостанции.	37
 Этап 7. Загрузка программы и её тестирование	 50
Шаг 1. Загрузка программы в модуль Arduino Uno	50
Шаг 2. Тестирование	50
 Этап 8. Обработка полученных результатов	 55
 Этап 9. Использование для реальных задач	 61
 А теперь....	 62
До новых встреч!	63

Минимальные системные требования определяются соответствующими требованиями программ Adobe Reader версии не ниже 11-й либо Adobe Digital Editions версии не ниже 4.5 для платформ Windows, Mac OS, Android и iOS; экран 10"

Учебное электронное издание

Серия: «РОБОФИШКИ»

Салахова Алёна Антоновна

**КОНСТРУИРУЕМ РОБОТОВ НА ARDUINO®.
ЭКОСТАНЦИЯ**

Для детей старшего школьного возраста

Ведущий редактор *М. С. Стригунова*

Руководители проекта от издательства *А. А. Елизаров, С. В. Гончаренко*

Научный консультант канд. пед. наук *Н. Н. Самылкина*

Ведущий методист *В. В. Тарапата*

Художники *В. А. Прокудин, Я. В. Соловцова, И. Е. Марев, Ю. Н. Елисеев*

Фотосъемка: *И. А. Федянин*

Технический редактор *Т. Ю. Федорова*

Корректор *И. Н. Панкова*

Компьютерная верстка: *Е. Г. Ивлева*

Подписано к использованию 20.11.17.

Формат 210×260 мм

Издательство «Лаборатория знаний»

125167, Москва, проезд Аэропорта, д. 3

Телефон: (499) 157-5272

e-mail: info@pilotLZ.ru, <http://www.pilotLZ.ru>

ЛОВИ НОВЫЕ «РОБОФИШКИ»

на **LEGO® MINDSTORMS®**
Education EV3,
Arduino®
и ScratchDuino®:

- ◆ «Крутое пике»
- ◆ «Волшебная палочка»
- ◆ «Секрет ткацкого станка»
- ◆ «Тайный код Сэмюэла Морзе»
- ◆ «Посторонним вход воспрещён!»
- ◆ «В поисках сокровищ»
- ◆ «Умный свет»
- ◆ «Да будет свет!» и другие.

С серией **«РОБОФИШКИ»**
самые удивительные
и неожиданные идеи
станут реальностью.

Создай своего робота,
учись и играй вместе с ним!

Стань настоящим изобретателем!

info@pilotLZ.ru
www.pilotLZ.ru



EAC