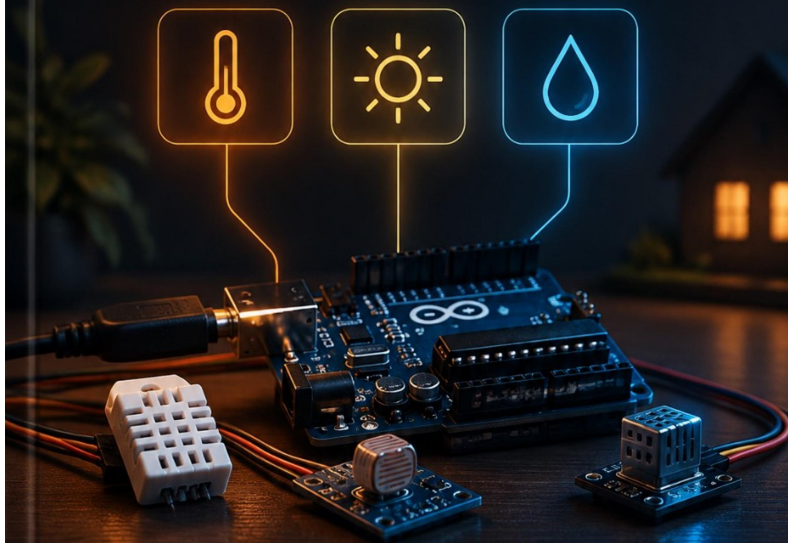


# Домашние датчики на Arduino

Как измерять температуру,  
свет и влажность.



и  
Дмитрий Соколов

# **Дмитрий Соколов**

# **Домашние датчики на**

# **Arduino: Как измерять**

# **температуру, свет и влажность**

*<https://litres.ru/74594614>*

*SelfPub; 2026*

## **Аннотация**

В комнате душно — или просто нагрелась лампа? Слишком темно — или глаза уже устали? Ощущения подсказывают, что что-то не так, но не всегда объясняют причину. Arduino и домашние датчики помогут заменить догадки измерениями — если знать, как правильно подключить датчик, проверить его показания и понять, что именно говорят цифры.

Эта книга проведёт вас от первых схем и программ до собственной системы наблюдения за микроклиматом. Вы научитесь измерять температуру, освещённость и влажность, разберётесь в точности датчиков и сможете отслеживать изменения показателей во времени. Для старта достаточно интереса к домашним задачам и желания попробовать.

В результате вы соберёте домашнюю станцию, а при желании настроите реакцию на показания: например, включение света, вентилятора или увлажнителя. А если что-то пойдёт не так,

книга поможет найти ошибку в подключении, коде или условиях измерения.

# Содержание

|                                                                                   |    |
|-----------------------------------------------------------------------------------|----|
| От бытовой задачи к измерению: что можно<br>узнать с помощью домашних датчиков    | 6  |
| Arduino без страха: платы, компоненты и<br>инструменты для первых проектов        | 16 |
| Основы электрических схем: питание, контакты<br>и безопасное подключение датчиков | 29 |
| Первая программа: как читать показания<br>датчика и выводить их на экран          | 40 |
| Температура под контролем: подключаем и<br>проверяем цифровой датчик              | 51 |
| Конец ознакомительного фрагмента.                                                 | 52 |

# **Дмитрий Соколов Домашние датчики на Arduino: Как измерять температуру, свет и влажность**

**Домашние датчики на Arduino: Как измерять температуру, свет и влажность**

*Дмитрий Соколов*

# От бытовой задачи к измерению: что можно узнать с помощью домашних датчиков

В девять вечера Ирина снова подошла к окну.

— Душно, — сказала она и приоткрыла форточку.

— Да не душно, просто лампа греет, — отозвался муж из-за ноутбука. — И вообще, ты только что проветривала.

В комнате горел торшер, на батарее сушилась футболка, а за окном медленно темнело. Через несколько минут стало прохладнее. Ирина закрыла окно, включила верхний свет и пересела на диван. Теперь ей казалось, что в комнате слишком ярко. Муж пожаловался, что тянет холодом. Никто не спорил из вредности: каждый описывал собственное ощущение. Но окно открывали и закрывали, лампы включали и выключали, а ответ на главный вопрос — *что именно не так с комнатой?* — оставался неизвестным.

Такие домашние загадки не требуют особой любви к технике. Они требуют сначала остановиться и понять, что именно мы хотим выяснить. Датчик полезен не потому, что умеет показывать цифры. Он полезен, когда эти цифры помогают ответить на конкретный вопрос.

**Ощущение — это начало, а не приговор**

Тело замечает изменения раньше, чем мы успеваем их

объяснить. Нам жарко, зябко, темно, влажно, душно. Эти сигналы важны: именно из них рождается бытовая задача. Но ощущение не подсказывает единственную причину.

Человеку может быть некомфортно из-за температуры воздуха, сквозняка, нагретой солнцем стены, яркого экрана или сочетания этих факторов. На ощущение тепла влияют одежда, активность и даже то, сколько времени мы провели в комнате. А слово «душно» может означать, что жарко, влажно, мало движения воздуха, накопились запахи или просто давно не открывали окно.

Представим три ступени.

**Наблюдение:** «К вечеру в комнате становится тяжело находиться».

**Измеряемая величина:** температура воздуха, относительная влажность или освещённость.

**Вывод:** «Комната нагревается к вечеру» или «уровень света на рабочем столе снижается».

Первые две ступени помогают сделать третью обоснованнее. Но сам вывод не всегда раскрывает причину. Например, температура 23 °C показывает, насколько прогрет воздух в точке измерения, но не объясняет, почему одному человеку при этом комфортно, а другому жарко. Датчик — не судья в семейном споре и не волшебный переводчик ощущений. Это инструмент, который добавляет к ощущениям наблюдаемые данные.

**Три величины, с которых удобно начать**

Для первых домашних наблюдений часто достаточно температуры, освещённости и влажности воздуха. Это не все возможные параметры комнаты, зато каждый отвечает на свой круг вопросов.

**Температура** показывает, насколько тёплый или холодный воздух в конкретном месте. Она помогает выяснить, меняется ли температура в комнате в течение дня, отличается ли угол у окна от места возле батареи, становится ли прохладнее после проветривания. Важно помнить: один датчик показывает условия именно там, где он находится. Под потолком, на подоконнике и у пола значения могут различаться. Прямые солнечные лучи, батарея, открытая форточка и тепло от работающего компьютера тоже способны повлиять на показания.

**Освещённость** описывает, сколько света падает на выбранную поверхность. Её измеряют в люксах. Это не то же самое, что «насколько светлой кажется комната»: стены, цвет мебели, направление взгляда и контраст между экраном и окружением меняют зрительное впечатление. Если вопрос звучит «достаточно ли света для чтения?», датчик имеет смысл расположить у книги или на рабочей поверхности, а не где-то на полке в противоположном углу. Важны также время суток, положение штор и включённые лампы.

**Относительная влажность** показывает, насколько воздух насыщен водяным паром по сравнению с тем количеством, которое он мог бы удержать при данной температу-

ре. Это уточнение важно: относительная влажность зависит не только от количества влаги, но и от температуры воздуха. Поэтому её изменение само по себе ещё не означает, что в комнате появилось или исчезло столько-то воды. Влажность может помочь заметить закономерности после душа, приготовления еды или сушки белья, но датчик не измеряет плесень и не устанавливает её причину.

Есть и принципиальный предел: если человеку кажется, что в комнате «нечем дышать», температура и влажность не дают полного ответа о качестве вентиляции. Для оценки некоторых аспектов воздухообмена нужны другие измерения, например концентрации углекислого газа. Если температура и влажность почти не меняются, а ощущение духоты остаётся, это не значит, что показания ошибочны. Возможно, выбранные величины просто не отвечают на весь вопрос.

### **Как превратить жалобу в проверяемый вопрос**

Фраза «у нас дома постоянно что-то не так» звучит весомо, но для измерения она слишком велика. Её нужно уменьшить до вопроса, который можно проверить в одной комнате и за понятное время.

Начните с самого простого: **что именно я наблюдаю?** Например: «К вечеру за рабочим столом хочется включить настольную лампу» или «после ужина в спальне кажется душнее».

Затем спросите себя: **какую величину стоит измерить?** Если дело в том, что стало теплее, подойдёт темпе-

ратура. Если трудно читать или работать, возможно, важна освещённость на столе. Если беспокоит сухость воздуха или влажные поверхности, можно наблюдать относительную влажность — с пониманием, что она не отвечает на все вопросы о влаге.

После этого решите, **где** измерять. Место должно соответствовать задаче. Чтобы понять условия у рабочего стола, измеряйте у рабочего стола. Чтобы сравнить комнату возле окна и в центре, понадобятся наблюдения в обеих точках, но не обязательно одновременно: можно по очереди переставлять один прибор, сохраняя прочие условия как можно более похожими.

Дальше определите **когда** и как часто. Одно значение — снимок. Несколько значений, снятых по плану, — короткая история. Например, показания утром, вечером и после проветривания позволят увидеть больше, чем случайное измерение в момент, когда вы уже открыли окно.

Наконец, запишите, что могло повлиять на результат. В комнате кто-то готовил? Солнце светило прямо в окно? Работал обогреватель? Окно оставалось приоткрытым? Эти обстоятельства не всегда нужно исключать: иногда именно они и интересуют. Но их стоит замечать, чтобы не принять совпадение за причину.

**Карточка наблюдения: пример с «душной» комнатой**

Вернёмся к Ирине. Вместо того чтобы решать, кто луч-

ше чувствует микроклимат, она может проверить более узкий вопрос: **«Меняются ли температура и относительная влажность в гостиной за час, когда дверь и окно закрыты?»**

Карточка измерения будет выглядеть так:

**Пункт**  
**Запись**

**Вопрос**

Меняются ли температура и относительная влажность в гостиной за час при закрытых окне и двери?

**Величина**

Температура воздуха и относительная влажность

**Место**

На открытой полке примерно на уровне сидящего человека, не у батареи и не на подоконнике

**Время**

В 20:00, 20:30 и 21:00

**Возможное влияние**

Число людей в комнате, работающая техника, нагрев от солнца, готовка, отопление

Эта карточка не обещает мгновенно раскрыть тайну духоты. Она делает другое: задаёт понятный план. Ирина сможет увидеть, меняются ли выбранные величины в течение часа. Если меняются, это повод задать следующий вопрос: что происходило в комнате в то же время? Если не меняются, а ощущение духоты остаётся, станет ясно, что нужно искать другой показатель или учитывать другие условия.

Чтобы сравнение было полезным, важно не менять одновременно всё подряд. Если в 20:00 закрыть окно, включить обогреватель, зажечь лампы и позвать гостей, а в 21:00 сравнить результат с пустой комнатой, будет трудно понять, какой фактор сыграл роль. Не нужно устраивать лабораторию посреди гостиной. Достаточно записывать важные обстоятельства и, когда это возможно, менять за один раз что-то одно.

В карточке можно добавить колонку «что происходило». Например: «20:30 — вошли двое», «21:00 — включили торшер». Так показания перестают быть россыпью цифр и складываются в историю комнаты.

### **Маленькое упражнение: начните с одной комнаты**

Выберите место, где вы чаще всего замечаете повторяющееся неудобство: кухню, спальню, рабочий угол или детскую. Не беритесь сразу за всю квартиру. Одна комната и один вопрос — хороший масштаб для первого самостоятельного наблюдения.

Заполните пять строк:

**1. Вопрос.** Что именно хочется проверить? Сформулируйте без слов «всегда» и «почему» — например: «Становится ли у рабочего стола темнее после 17:00?»

**2. Величина.** Что поможет ответить: температура, освещённость, относительная влажность или, возможно, другая величина?

**3. Место.** Где измерение будет связано с вашей задачей? Если важен свет для чтения — на поверхности стола. Если интересует вечерняя температура в спальне — не вплотную к батарее или окну.

**4. Время.** Когда вы снимете показания? Один раз или несколько? Повторите измерение в сопоставимые дни или часы, если это нужно для сравнения.

**5. Возможное влияние.** Что способно изменить результат: солнце, открытое окно, работающая техника, приготовление еды, количество людей, включённый свет?

Вот заготовка:

**Вопрос**

**Величина**

**Место**

**Время**

**Возможное влияние**

Для первого опыта не гонитесь за большим количеством

измерений. Лучше три понятных наблюдения в заранее выбранное время, чем десятки случайных цифр, после которых непонятно, что сравнивать. Если вопрос о свете, старайтесь измерять одну и ту же поверхность, в сходных условиях и с одинаковым положением датчика. Если вопрос о температуре, отмечайте, открывали ли окно и работало ли отопление. Если вопрос о влажности — записывайте, например, сушилось ли в комнате бельё или принимали ли рядом душ.

И ещё одно правило: сначала сформулируйте вопрос, а потом выбирайте, что измерять. Иначе легко получить интересные числа, которые не помогают принять решение. Цифра сама по себе не отвечает, нужно ли переставить стол, чаще проветривать или поменять лампу. Она становится полезной, когда мы знаем, с чем её сравнить и какое действие собираемся обдумать.

### **Какие вопросы можно задать дому**

Чтобы потренироваться, попробуйте продолжить несколько фраз:

- «Я хочу понять, становится ли в спальне теплее, когда...»
- «Мне нужно проверить, хватает ли света на...»
- «После готовки влажность в кухне меняется...»
- «В разных частях комнаты температура отличается...»
- «После того как я переставил стол, освещённость на рабочей поверхности...»

Хороший вопрос не обязан быть сложным. «Где на мо-

ём столе светлее?» — уже начало измерения. «Становится ли в комнате прохладнее после открытия окна?» — тоже. А вот «почему мне здесь некомфортно?» потребует нескольких шагов: сначала наблюдений, затем подходящих величин и, возможно, новых измерений.

В этом есть навык, который выходит далеко за пределы электроники. Мы учимся не обесценивать собственные ощущения, но и не путать их с объяснением. Учимся замечать условия, задавать уточняющие вопросы, проверять предположения и менять план, если данных недостаточно. Дом становится не загадкой, которую нужно победить, а средой, которую можно постепенно понять.

Когда Ирина заполняет карточку, спор о том, душно ли в комнате, не исчезает по щелчку. Зато появляется более полезный разговор: что именно мы наблюдаем, где это происходит и какой следующий шаг поможет разобраться? Домашний датчик начинается именно здесь — не с платы и не с экрана, а с хорошего вопроса. Сначала определите, что хотите узнать. Потом решите, какие данные смогут вам помочь. И только после этого числа займут своё место в истории комнаты.

# **Arduino без страха: платы, компоненты и инструменты для первых проектов**

Когда Ирина открыла коробку с набором для Arduino, на стол высыпались платы, разноцветные провода, пакетики с резисторами и несколько датчиков с названиями, похожими на пароли от Wi-Fi. В описании набора обещали «десятки проектов». Через полчаса Ирина уже знала, что у неё есть модуль ВН1750, но не понимала, куда его подключать, чем он отличается от фоторезистора и почему кабель из коробки заряжает плату, но не позволяет загрузить программу.

Электроника не оказалась сложной. Сложным оказался беспорядок, который продают под видом готового старта.

Чтобы измерить температуру, освещённость или влажность дома, не нужна мастерская и чемодан приборов. Нужна небольшая система, в которой каждая деталь отвечает на простой вопрос: **что она делает и зачем она здесь?** Давайте разберём такую систему — не торопясь, по одному предмету за раз.

**Не всякая Arduino — одна и та же плата**

Словом «Arduino» часто называют сразу несколько вещей: семейство плат, программную среду и сам способ собирать проекты. Для начала полезно разделить три понятия.

**Плата** — это готовая площадка с разъёмами, куда можно подключить питание и компоненты. На ней находится микроконтроллер, а также другие элементы, необходимые для работы и связи с компьютером.

**Микроконтроллер** — небольшой компьютер внутри платы. Он выполняет программу: например, раз в несколько секунд считывает показания датчика и отправляет их на экран.

**Датчик** — источник сведений о мире. Он реагирует на температуру, свет или влажность и передаёт плате электрический сигнал. Датчик можно сравнить с органом чувств, микроконтроллер — с исполнителем инструкции, а плату — с удобным корпусом, который помогает им взаимодействовать.

Это не просто слова. Когда проект не работает, такое разделение подсказывает, где искать причину: датчик не получает подходящего питания, плата выбрана не та или программа обращается не к тому контакту.

Кроме того, Arduino — не одна-единственная модель. Есть, например, популярная **Arduino Uno Rev3** с логикой 5 В и компактные платы на базе **ESP32**, обычно работающие с логическими уровнями 3,3 В. У них могут отличаться разъёмы, доступные контакты, объём памяти и возможности связи. Похожие названия или одинаковые ряды штырьков ещё не гарантируют полной совместимости.

Поэтому держите в уме четыре вопроса: совпадают ли

**разъёмы**, подходит ли **напряжение**, поддерживает ли плата нужный **протокол связи** и есть ли для неё подходящие **программа или библиотека**. Точный ответ ищут в документации конкретной модели, а не в смутном воспоминании «вроде у всех Arduino одинаково».

### **Плата: место, где сходятся команды**

Для первых опытов часто выбирают Arduino Uno Rev3 или плату-совместимый аналог с хорошо документированной моделью. У неё достаточно контактов для типичных учебных задач, а примеры для этой платформы легко найти. Если вы предпочитаете другую плату, это тоже нормально: важно не угадать «правильную навсегда», а понять параметры именно своего устройства.

На плате обычно есть контакты питания, цифровые входы и выходы, аналоговые входы и разъём для связи с компьютером. На первых этапах не нужно запоминать назначение каждого контакта. Достаточно найти официальную схему расположения контактов — **pinout** — и знать, что контакты не взаимозаменяемы автоматически: один принимает аналоговый сигнал, другой может поддерживать определённый протокол связи.

**Начните с документации, например со страницы Arduino Uno Rev3:**

<https://docs.arduino.cc/hardware/uno-rev3/>

Если у вас плата другой модели или изготовителя, найдите страницу именно для неё. У совместимых плат бывают

отличия в микроконтроллере, USB-микросхеме, драйверах и назначении контактов. Слово «совместимая» не означает «во всём идентичная».

**Задача платы:** выполнять вашу программу и связывать компьютер с подключёнными компонентами. **Пока не требуется:** покупать самую мощную, беспроводную или дорогую модель «на будущее». Будущее часто наступает не сразу, а плата всё это время лежит в коробке рядом с ещё одним ненужным датчиком.

### **Кабель и программа: разговор с компьютером**

Для загрузки программы понадобится USB-кабель, подходящий к разъёму вашей платы. Но снаружи одинаковые кабели иногда ведут себя по-разному: одни передают и питание, и данные, другие предназначены только для зарядки. Если компьютер подаёт питание, но не видит плату, кабель без передачи данных — одна из первых вещей, которые стоит проверить.

Для программирования можно установить **Arduino IDE** — среду, где пишут и загружают код. Скачивать её лучше с официального сайта: <https://docs.arduino.cc/software/ide-v2>

В среде нужно выбрать модель платы и соответствующий порт. Для некоторых плат также понадобятся дополнительные программные пакеты или драйверы. Если компьютер не предлагает нужную модель или подключение не появляется в списке, это не обязательно означает, что плата неисправна: возможно, не установлен пакет платы, драйвер или выбран

неподходящий кабель.

Хорошая первая проверка — загрузить в плату простой пример мигания встроенным светодиодом, если такой пример и светодиод предусмотрены моделью. Так вы проверите цепочку «компьютер — кабель — среда — плата», ещё не подключая датчики. Это как сперва убедиться, что телефон включается, а потом уже выяснять, почему в приложении не открывается карта.

### **Макетная плата: стол для экспериментов**

**Макетная плата** позволяет соединять компоненты без пайки. В отверстия вставляют выводы датчиков и перемычки, а внутри некоторые отверстия соединены между собой металлическими контактами. Поэтому внешне пустая пластиковая пластина на самом деле устроена как набор скрытых дорожек.

У типовой макетной платы есть группы отверстий в центральной области и продольные линии по краям, часто используемые для питания. Но устройство конкретной модели лучше проверить: у длинных боковых шин иногда есть разрыв посередине. Значит, один край не обязательно электрически соединён с другим.

**Задача макетной платы:** дать возможность собрать, изменить и разобрать низковольтную схему без пайки. **Полезная привычка:** перед подключением посмотрите схему соединений именно вашей макетной платы. Не считайте, что все отверстия соединены одинаково только потому, что они

выглядят одинаково.

Макетная плата хороша для первых опытов с небольшими датчиками. Она не предназначена для любого напряжения и тока, а соединения в ней могут ослабнуть, если слишком толстый вывод многократно вставлять в одно отверстие. Если деталь сидит неплотно, не пытайтесь удержать её силой: сначала проверьте, подходит ли её вывод для макетной платы.

### **Перемычки: дороги между островками**

Провода для макетной платы часто называют **перемычками**. Они бывают с разными наконечниками: «папа — папа», «папа — мама» и «мама — мама». Названия описывают форму разъёмов, а не качество провода.

Для соединения платы с макетной платой обычно удобны провода «папа — папа». Если у датчика уже есть штыревые контакты, иногда нужны перемычки «мама — папа». Здесь важно заранее посмотреть на разъёмы: красивый набор из ста проводов не поможет, если все они заканчиваются не тем типом наконечника.

**Задача перемычек:** соединять контакты платы, датчика и макетной платы. **Проверка перед сборкой:** сколько проводов нужно, какие у них наконечники и удобно ли будет прочитать маркировку на самой плате после подключения.

Цвета проводов могут помочь, если договориться с собой о правилах: например, один цвет всегда использовать для питания, другой — для общего провода, остальные — для сиг-

налов. Но цвет сам по себе не сообщает, куда провод подключён. Не полагайтесь на память: маленькая схема или запись в блокноте надёжнее, чем «кажется, красный шёл туда».

### **Датчики: сперва вопрос, потом покупка**

Датчик выбирают не потому, что его положили в большой набор, а потому, что он помогает ответить на конкретный вопрос. Для проекта о температуре и влажности может подойти комбинированный датчик. Для оценки освещённости — цифровой датчик света или фоторезистор.

У них разный принцип работы. Модуль цифрового датчика может передавать результат по определённом протоколу. Фоторезистор меняет сопротивление в зависимости от освещения, и для получения измеримого напряжения ему обычно нужен дополнительный элемент схемы. Внешне оба варианта могут называться «датчиком света», но подключаются и используются не одинаково.

Сами датчики тоже не всегда можно вставить в плату как есть. Иногда перед вами готовый **модуль** — маленькая платка с датчиком и дополнительными компонентами. Иногда это отдельная электронная деталь, требующая обвязки. На фотографии в магазине различие бывает незаметно; на столе оно быстро становится очень заметным.

Особенно внимательно проверяйте питание и логические уровни. Модуль, рассчитанный на 3,3 В, нельзя без проверки подключать к сигналу 5 В. У некоторых платок есть стабилизатор напряжения или преобразователь уровней, у других

— нет. Не делайте вывод по одному только названию датчика: ищите описание именно той версии модуля, которую купили.

Для документации полезны страницы и руководства, где указаны питание, интерфейс, распиновка и примеры кода. Например, руководство по датчикам семейства DHT: <https://learn.adafruit.com/dht>

**Для датчика освещённости BH1750 есть практическое руководство SparkFun:**

<https://learn.sparkfun.com/tutorials/bh1750-ambient-light-sensor-hookup-guide/all>

Даже если вы нашли руководство с таким же названием, сверьте внешний вид и маркировку своей платы. Производители модулей могут по-разному размещать контакты и добавлять или не добавлять вспомогательные детали.

## **Нужны ли сразу все инструменты?**

Для первого проекта обычно достаточно платы, USB-кабеля для передачи данных, макетной платы, нескольких подходящих перемычек и одного датчика. Резисторы пригодятся для многих схем — разумно иметь небольшой набор распространённых номиналов, но не обязательно покупать целую профессиональную лабораторию.

**Мультиметр** полезен, когда нужно проверить напряжение, целостность провода или наличие контакта. Для первых шагов его можно не покупать, если проект очень простой, а схема и детали проверены. Но если он уже есть — научи-

тесь хотя бы измерять постоянное напряжение и проверять проводимость, следуя инструкции к прибору. Не пытайтесь измерять то, чего не понимаете, особенно в розетке: домашние проекты Arduino относятся к низковольтным цепям и не требуют работы с сетевым напряжением.

Паяльник на старте тоже не обязателен. Макетная плата создана как раз для того, чтобы можно было экспериментировать без пайки. Если в наборе лежит паяльник, это не приглашение немедленно его включить.

**Инвентарная карточка: чтобы коробка перестала быть загадкой**

Перед сборкой заполните карточку для каждой детали. Она превращает набор из россыпи предметов в понятную систему.

**Деталь**

**Задача**

**Нужные соединения и условия**

**Документация**

Arduino Uno Rev3

Выполняет программу и связывает компьютер с датчиками

USB для загрузки программы; контакты выбирать по схеме платы; логика 5 В

<https://docs.arduino.cc/hardware/uno-rev3/>

## USB-кабель

Питание и передача данных между платой и компьютером

Разъём должен подходить к плате; кабель должен передавать данные

Инструкция к плате и описание кабеля

## Макетная плата

Соединяет детали без пайки

Проверить, какие группы отверстий соединены внутри

Схема конкретной модели; раздел Arduino по электронике: <https://docs.arduino.cc/learn/electronics/>

## Перемычки

Связывают контакты между собой

Подобрать наконечники под плату, макетку и модуль

Маркировка набора и схема проекта

## DHT22-модуль

Измеряет температуру и влажность

Проверить распиновку, питание и наличие необходимых компонентов на модуле

<https://learn.adafruit.com/dht>

## ВН1750-модуль

Оценивает освещённость

Проверить распиновку, питание, логические уровни и интерфейс I<sup>2</sup>C

<https://learn.sparkfun.com/tutorials/bh1750-ambient-light-sensor-hookup-guide/all>

Карточка не заменяет руководство и не обещает, что перечисленные модули совместимы в любой комбинации. Её задача — заставить задать нужные вопросы **до** подключения. Если в графе «условия» вы не знаете, что написать, значит, сначала стоит уточнить документацию.

### **Проверка коробки до первой сборки**

Высыпать всё на стол — эффектно, но не всегда полезно. Разложите детали группами: плата и кабель; макетная плата и провода; датчики; мелкие компоненты. На пакетиках сохраните надписи, а если их нет — приклейте собственные ярлыки.

Проведите короткую проверку:

1. Найдите точное название платы и страницу с её документацией.
2. Проверьте, подходит ли USB-кабель и передаёт ли он данные.
3. Осмотрите датчик и запишите его маркировку и распиновку.
4. Найдите в руководстве питание, логические уровни и тип соединения.
5. Сверьте разъёмы и нужные контакты платы.

**6.** Загрузите простой тестовый пример до подключения дополнительных модулей.

**7.** Перед любым изменением соединений отключайте питание.

Храните компоненты в подписанных пакетах или небольших коробочках. Датчики и плату лучше оберегать от влаги, а мелкие резисторы держать отдельно от проводов: иначе при следующем поиске нужного номинала можно устроить археологическую экспедицию по дну коробки. Полезно хранить вместе с деталями их карточки и распечатанную распиновку.

## **Соберите свой первый набор**

Для простого старта выберите одну конкретную цель, например измерять температуру и влажность в комнате. Подберите плату, USB-кабель, небольшую макетную плату, набор перемычек и один подходящий датчик. Когда этот опыт станет понятным, добавьте измерение света — отдельным модулем или другим датчиком, в зависимости от выбранного проекта.

Именно так набор растёт осмысленно: не «потому что было со скидкой», а потому что следующий вопрос требует новой детали. Можно, конечно, купить сорок модулей «на всякий случай». Это достойный способ развить навык сортировки пакетиков, но не самый короткий путь к работающему датчику.

Главная проверка простая: для каждой вещи вы можете

объяснить её задачу, назвать нужные соединения и показать документацию. Если можете — перед вами уже не загадочная коробка, а комплект для эксперимента. Если не можете — не беда: это не повод бояться электроники, а сигнал остановиться и выяснить один конкретный факт.

Для первых опытов достаточно малого, понятного и совместимого набора. Начните с одной платы и одного датчика, проверяйте каждый компонент по его документации и не подключайте наугад то, что выглядит похожим. Следующая задача — научиться читать путь питания и сигнала на схеме, чтобы соединять детали уверенно и безопасно.

# **Основы электрических схем: питание, контакты и безопасное подключение датчиков**

Вы собираете датчик температуры на макетной плате. Провода выглядят аккуратно: красный тянется к красной шине, чёрный — к синей, третий соединяет датчик с Arduino. Но экран молчит. Вы проверяете программу, меняете датчик местами, открываете поисковик — и только потом замечаете: шина питания на этой половине платы не соединена с той, где стоит провод от Arduino. Внешне всё было похоже на рабочую схему. Электрически это были два отдельных участка.

Такая ошибка случается не потому, что человек «не создан для техники». Просто макетная плата устроена не так, как кажется с первого взгляда. Провода не передают информацию по волшебству, а значки VCC и GND — не декоративные подписи. У любой схемы есть маршруты: откуда приходит питание, куда оно возвращается и каким путём датчик сообщает результат измерения. Когда вы научитесь видеть эти маршруты, схема перестанет быть клубком проводов и станет понятной картой.

**Схема — это путь, а не картинка**

Представьте комнату с водопроводом. Вода должна прий-

ти по одной трубе, пройти через прибор и вернуться по другой. Если один участок перекрыт или трубы соединены не там, прибор не заработает. Электрическая схема устроена иначе физически, но для первого понимания полезна та же мысль: цепь должна быть замкнутой, а каждый провод — вести туда, куда нужно.

В типичном подключении датчика есть три важных участника:

- **VCC** — контакт питания датчика. На нём должно быть напряжение, подходящее именно этому датчику.
- **GND** — общий обратный путь электрической цепи, часто называемый «землёй».
- **Сигнальный вывод** — провод, по которому датчик передаёт данные микроконтроллеру.

Когда Arduino питает датчик, путь можно описать словами: напряжение выходит с подходящего вывода питания платы, приходит на VCC датчика, датчик выполняет свою работу, а цепь возвращается через GND к общей земле платы. Отдельный сигнальный провод идёт от выхода датчика к входу Arduino.

Здесь есть важное уточнение: GND в учебной низковольтной схеме — это общий опорный потенциал и обратный путь тока. Это **не обязательно** защитное заземление электропроводки дома. Не соединяйте выводы Arduino с контактами розетки и не считайте надпись GND приглашением работать с сетевым напряжением.

В схемах принято показывать условное направление тока от плюса к минусу. Реальное движение электронов в металлическом проводнике описывают иначе, но для сборки достаточно запомнить главное: питание должно иметь полный путь, а сигнальные уровни плата измеряет относительно общей земли. Если Arduino и датчик не имеют общей точки GND, сигнал может оказаться для платы неопределённым: у неё не будет понятной «нулевой отметки», с которой сравнивать напряжение.

### **Сначала прочитайте подписи**

Не начинайте с соединения проводов. Начните с самих деталей и документов к ним. Одна и та же внешне похожая плата датчика может иметь другую распиновку, другое рабочее напряжение или иной способ передачи данных. Надпись на плате помогает сориентироваться, но, если выводы не подписаны или подписи неясны, нужен документ именно к вашей модели.

Перед сборкой найдите для платы Arduino и датчика:

1. Назначение каждого контакта.
2. Допустимое напряжение питания датчика.
3. Допустимый уровень сигнала на его выходе.
4. Способ обмена данными: аналоговый, цифровой, I<sup>2</sup>C, SPI или другой.
5. Особенности подключения, например необходимость подтягивающего резистора или преобразователя уровней.

Питание и сигнальный уровень — не одно и то же. Допу-

стим, датчик можно питать от 5 В, но его выход выдаёт сигнал, который не рассчитан на вход конкретной 3,3-вольтовой платы. Или наоборот: датчик работает только от 3,3 В, хотя в наборе есть Arduino с 5-вольтовыми логическими уровнями. В обоих случаях надписи VCC и GND сами по себе не гарантируют совместимость.

Ориентируйтесь не на общую фразу «Arduino обычно работает с...», а на документацию используемых устройств. Платы отличаются друг от друга, а даже внутри одной линейки могут меняться допустимые напряжения и расположение выводов. Проверьте модель и распиновку именно своей платы. Если сигнал может превышать допустимое напряжение входа, нужен подходящий преобразователь уровня или другая схема согласования. Не подключайте вывод «на пробу»: вход микроконтроллера рассчитан на определённый диапазон, и превышение этого диапазона может повредить плату.

### **Макетная плата: дорожки, которых не видно**

Макетная плата позволяет собирать схемы без пайки, но её внутренние контакты скрыты под пластиком. Поэтому полезно знать типичную логику соединений.

В центральной части платы отверстия обычно объединены короткими группами: несколько отверстий в одном ряду связаны между собой. Через центральную канавку группа обычно прерывается. Эта канавка нужна, чтобы установить микросхему так, чтобы её выводы с двух сторон не оказались соединены в одну общую линию.

По краям часто идут длинные шины питания, иногда обозначенные красной и синей линиями или знаками «+» и «-». Однако эти метки — подсказка, а не гарантия. Шина может быть разделена на две части посередине; линии на некоторых платах вовсе не соединены по всей длине. Бывает и так, что внешний вид похож на привычную макетную плату, а внутренняя разводка отличается.

Поэтому запомните не универсальную картинку, а конкретную плату перед вами. Если есть сомнение, найдите её описание или проверьте соединения мультиметром в режиме прозвонки. Делайте это только при отключённом питании. В режиме прозвонки прибор подаёт небольшой тестовый сигнал; включённая схема может дать неверный результат или повредить измеритель.

Шина питания, разорванная посередине, — классическая причина «неработающего» датчика. Красная линия на пластике может выглядеть непрерывной, но электрического контакта между её половинами нет. Если Arduino подаёт питание на левую половину шины, а VCC датчика подключён к правой, датчик останется без питания. Исправить ситуацию можно отдельным перемычным проводом между участками — но только после того, как вы убедились, что обе части действительно должны быть соединены и подключены к одному источнику.

## **Собираем маршрут по шагам**

Ниже — последовательность, которая помогает не угады-

вать, а принимать решения.

**Шаг 1. Определите источник питания.** Решите, откуда схема будет получать питание: от вывода платы, USB или подходящего внешнего источника. Для первого опыта выберите низковольтную учебную схему и питание, предусмотренное документацией. Не подключайте к макетной плате сетевое напряжение. Оно опасно для жизни и совершенно не подходит для экспериментов с Arduino.

Не подавайте питание сразу из нескольких источников, если не знаете, как они связаны и допускает ли это ваша плата. Правила подключения USB и внешнего питания могут различаться — уточните их в документации к конкретной модели.

**Шаг 2. Найдите выводы датчика.** Сверьте маркировку на корпусе с документацией. У одних модулей контакты подписаны VCC, GND и OUT; у других — просто «+», «-» и «S»; у некоторых обозначения нанесены на обратную сторону. Не полагайтесь на расположение выводов у похожего датчика из интернета: считайте его только подсказкой, пока не подтвердите модель.

**Шаг 3. Проверьте совместимость.** Убедитесь, что выбранное питание входит в допустимый диапазон датчика, а выход датчика безопасен для входа платы. Если информация противоречива или её не удалось найти, остановитесь и уточните её. Пауза на чтение документации дешевле, чем замена повреждённого контроллера.

**Шаг 4. Отметьте землю.** Найдите GND на Arduino и GND датчика. Если они питаются от одного источника, соедините эти точки общей землёй. Если используются два отдельных источника, обычно требуется общая земля между ними, чтобы сигналы измерялись относительно одной опоры, — но это правило не применяют вслепую к системам с гальванической развязкой. Если не уверены, как именно соединены источники, сначала разберитесь в схеме.

**Шаг 5. Проложите питание.** Соедините подходящий вывод питания платы с нужной шиной макетной платы, а от неё — с VCC датчика. Затем отдельно проверьте, что участок шины, на котором стоит VCC, действительно соединён с источником. Красная линия сама по себе ничего не доказывает.

**Шаг 6. Проложите сигнал.** Подключите выход датчика к тому входу Arduino, который подходит для выбранного способа обмена. Аналоговый выход должен идти на подходящий аналоговый вход, цифровой — на предусмотренный цифровой. Для интерфейсов I<sup>2</sup>C или SPI нужны соответствующие линии данных и тактовые выводы. Их расположение зависит от платы; сверяйтесь с её документацией.

Не путайте питание с сигналом: если соединить выход датчика с VCC или GND вместо входа, плата не получит измерение, а отдельные компоненты могут оказаться под нежелательным напряжением.

**Шаг 7. Оставьте питание отключённым и осмотри-**

**те сборку.** Пока вы прокладываете или переставляете провода, отсоедините USB и внешний источник. Так вы уменьшите вероятность случайного короткого замыкания инструментом или проводом. Перед включением пройдите цепь от начала до конца — не глазами по цветным линиям, а по реальным точкам контакта.

## **Контрольный осмотр перед запуском**

Используйте короткий чек-лист:

- Питание:** источник и напряжение подходят плате и датчику.

- Земля:** GND платы и GND датчика соединены там, где это требуется.

- Сигнал:** выход датчика приходит на правильный вход, а не на шину питания.

- Совместимость:** напряжение сигнала допустимо для микроконтроллера.

- Проверка без питания:** проводка осмотрена, шины не перепутаны, перемычки стоят в нужных группах; при необходимости соединения проверены мультиметром на обесточенной схеме.

Добавьте ещё одну проверку: убедитесь, что перемычка не соединяет напрямую шину питания с GND. Такое соединение называют коротким замыканием. Источник питания попытается пропустить через него ток, который не ограничен обычной работой датчика. Последствия зависят от источника и защиты платы: от перезапуска и нагрева до поврежде-

ния компонентов.

Если после включения что-то ведёт себя необычно — плата перезагружается, компонент заметно нагревается или появляется запах — сразу отключите питание. Не трогайте горячие детали и не продолжайте эксперимент, пока не найдёте причину. Обычный датчик при корректном подключении не должен требовать от вас терпеть нагрев или запах.

### **Небольшая тренировка: объясните схему словами**

Возьмите трёхконтактный датчик с понятной документацией. Пока ничего не включайте. Нарисуйте три прямоугольника: источник питания, датчик и Arduino. Проведите между ними линии и подпишите каждую: VCC, GND, сигнал. Затем ответьте письменно:

1. Откуда приходит питание и какой путь оно проходит до датчика?
2. Через какой провод схема возвращается к общей земле?
3. Какой контакт несёт измерение и к какому входу платы он подключён?
4. Какое напряжение может появиться на сигнальном выводе?
5. Какие отверстия макетной платы соединены между собой?

Теперь проследите каждый маршрут пальцем по реальной сборке, всё ещё не подключая питание. Если вы не можете объяснить, куда ведёт конкретный провод, не включайте схему — сначала уточните его назначение.

Для проверки соединений можно использовать мультиметр в режиме прозвонки, но только на полностью обесточенной макетной плате. Отдельно проверьте, соединены ли нужные отверстия одной шины и нет ли случайного прямого контакта между линиями питания и земли. Не проверяйте режимом прозвонки участки, на которых уже есть напряжение.

### **А если датчик управляет не только измерением?**

Пока мы говорили о датчиках, которые сообщают показания. Позже вы можете захотеть включить лампу, вентилятор, насос или другой исполнительный прибор. Здесь важно не переносить правила датчика на устройство с мотором или заметной нагрузкой. Вывод микроконтроллера рассчитан на ограниченный ток и не предназначен для прямого питания таких устройств.

Для исполнительных устройств нужны подходящий драйвер, транзисторный ключ или другое согласующее устройство; для некоторых задач также необходима гальваническая изоляция. Конкретное решение зависит от напряжения, тока и характера нагрузки. Особенно важно не подключать бытовую нагрузку с сетевым напряжением непосредственно к Arduino или макетной плате. Такие проекты требуют подходящих компонентов, изоляции, корпуса и специальных мер безопасности; для первых опытов выбирайте низковольтные схемы.

### **Понимание до включения**

Хорошая электрическая сборка начинается не с нажатия кнопки, а с нескольких спокойных вопросов: откуда приходит питание, где проходит общий путь, каким контактом передаётся сигнал и подходят ли друг другу все уровни напряжения. Макетная плата становится понятнее, когда вы учитываете её скрытые соединения, а документация перестаёт быть скучным приложением — она превращается в карту местности.

Перед следующим включением попробуйте сказать о своей схеме одну фразу: «Питание идёт отсюда к VCC датчика, возвращается через GND, а сигнал выходит с этого контакта и приходит на этот вход Arduino». Если фраза верна и подтверждается осмотром, вы уже не просто соединяете детали. Вы понимаете, что именно собрали. А это и есть основа надёжного эксперимента.

# Первая программа: как читать показания датчика и выводить их на экран

На столе лежит собранная схема. Arduino подключена к компьютеру, индикатор на плате моргнул — значит, питание есть. Но в окне программы пусто. Вы ждёте, словно датчик вот-вот заговорит, и начинаете подозревать всё сразу: провод, плату, себя и, возможно, саму идею домашних измерений.

Это обычный момент первого запуска. Схема уже собрана, но между ощущением «я всё подключил» и фразой «в комнате сейчас вот столько-то» ещё нужно построить мост. Этот мост — короткая программа, которая читает сигнал датчика и отправляет результат туда, где его можно увидеть. Не нужно понимать каждую строчку кода с первого раза. Важно научиться проверять их по порядку.

## Начинаем с одного вопроса

Для мини-кейса возьмём аналоговый датчик освещённости с выводами питания и контактом **АО** — аналоговым выходом. Подключим его к Arduino Uno или совместимой плате: питание датчика и земля — согласно его документации, а **АО** — к входу **А0**. У некоторых модулей есть ещё **ДО**, цифровой выход. Для нашей программы нужен именно **АО**: он

передаёт изменяющееся напряжение, а не просто сообщает «светло» или «темно».

Перед включением ещё раз сверьте маркировку и допустимое напряжение модуля с характеристиками платы. У разных Arduino-плат различаются рабочие напряжения и аналоговые входы. Нельзя автоматически считать, что любая плата безопасно принимает до 5 В.

Пока не пытайтесь получить освещённость в люксах. Первый вопрос проще: **меняется ли сигнал, когда меняется свет?** Это важное различие. Аналоговый вход платы не знает, что такое «уютный полумрак» или «яркий солнечный день». Он измеряет напряжение и превращает его в число. Чтобы перевести это число в физическую величину, понадобятся характеристики датчика и калибровка.

### **Сначала готовый пример**

Хорошая привычка — не начинать с переписывания чужого кода. Если производитель датчика или библиотека предлагают проверенный пример именно для вашей модели, сначала запустите его без изменений. У цифрового датчика для этого часто нужна библиотека: она содержит готовые команды, с помощью которых Arduino обменивается с конкретным устройством. Для аналогового модуля с выходом АО библиотека обычно не нужна: достаточно прочитать значение на входе.

Откройте пример, если он есть, и проверьте документацию: подходит ли он к вашей модели, какой вывод исполь-

зуется и какой тип платы предполагается. Один и тот же пример может не подойти двум внешне похожим датчикам. Разъём с надписью **DATA** или **SDA** — ещё не гарантия, что устройство следует читать через `analogRead()`.

В нашей мини-схеме программа короткая, поэтому её можно набрать самостоятельно:

```
```cpp
const int sensorPin = A0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  int rawValue = analogRead(sensorPin);
  Serial.print("AO = ");
  Serial.println(rawValue);
  delay(1000);
}
```

В этой программе удобно увидеть три шага: **прочитать** — **обработать** — **вывести**. Пока обработка минимальна: мы ничего не пересчитываем, а просто даём показанию понятную подпись. Для первого опыта этого достаточно.

### **Что делает программа**

`setup()` запускается один раз, когда плата стартует или перезапускается. В нём команда `Serial.begin(9600)` открывает последовательный канал обмена данными с компьютером. Число 9600 — скорость передачи, или `baud rate`. Позже она

должна совпасть со скоростью в окне монитора порта.

`loop()` — повторяющаяся часть программы. После завершения её команд Arduino возвращается к началу цикла и выполняет их снова. Команда `analogRead(sensorPin)` читает сигнал на выбранном аналоговом входе. В нашем примере выбран **A0**.

`Serial.print()` отправляет текст без перехода на новую строку, а `Serial.println()` выводит значение и завершает строку. Поэтому на экране получается надпись вроде **AO = 417**, а следующее измерение появляется ниже. Наконец, `delay(1000)` делает паузу примерно в одну секунду. Без неё плата отправляла бы значения намного быстрее, чем удобно читать человеку.

На классической Arduino Uno аналоговый вход обычно возвращает числа от **0 до 1023**: так десятиразрядный аналого-цифровой преобразователь представляет диапазон напряжения. При условном опорном напряжении 5 В один шаг соответствует примерно 5/1024 В, то есть около 4,9 мВ. Это не означает, что измерение всегда точное до 4,9 мВ: на результат влияют точность опорного напряжения, свойства платы, датчика и самой схемы. А на других платах диапазон и разрешение могут отличаться — сверяйтесь с документацией именно к своей модели.

Кроме того, число **не является универсальной шкалой яркости**. У одного модуля показание может увеличиваться на свету, у другого — уменьшаться: это зависит от

устройства схемы. И даже если датчик реагирует правильно, значение 700 не означает 700 люкс. Фоторезистор и делитель напряжения обычно не дают прямого линейного пересчёта в освещённость. Пока ваша цель — увидеть устойчивое изменение и понять направление реакции.

## **Выбираем плату, загружаем скетч, открываем монитор**

Сначала подключите Arduino к компьютеру USB-кабелем. Некоторые кабели передают только питание, поэтому плата может загореться, но не появиться в списке портов. В Arduino IDE выберите модель платы — обычно в меню **Tools / Инструменты → Board / Плата**. Затем выберите порт. Если непонятно, какой именно, отключите плату, посмотрите список, подключите снова и отметьте новый появившийся порт.

Нажмите кнопку проверки или **Verify**. На этом шаге IDE пытается собрать программу и обнаружить ошибки, но ещё не загружает скетч в плату. Если проверка прошла, нажмите **Upload / Загрузить**. Дождитесь сообщения об успешной загрузке. Если редактор сообщает об ошибке, не спешите менять провода: сначала посмотрите текст сообщения и исправьте именно программную проблему.

Теперь откройте **Serial Monitor / Монитор порта**. В его нижней части выберите скорость **9600 бод** — ровно такую же, как в `Serial.begin(9600)`. Если скорости не совпадают, вместо нормального текста могут появиться нечитаемые

символы или пустое, на первый взгляд непонятное окно.

Дайте плате секунду после открытия монитора: некоторые модели перезапускаются при подключении к нему. Затем вы должны увидеть строки с новыми значениями примерно раз в секунду.

Положите модуль на стол и запишите несколько показаний. Затем затените его ладонью, уберите руку, направьте на него фонарик и снова понаблюдайте. Не угадывайте заранее, должно ли число расти или падать: сначала установите, как ведёт себя именно ваш модуль. Несколько повторов полезнее одного случайного скачка.

### **Если окно пустое**

Когда данных нет, легко заключить: «сломался датчик». Но пустой монитор может быть не связан с датчиком вовсе. Наша программа выводит результат даже в том случае, если на A0 не подключено ничего, — это будут какие-то значения, возможно нестабильные. Поэтому полезно разделять две задачи: сначала убедиться, что программа отправляет текст, и только потом проверять, правильно ли датчик влияет на показания.

Проверьте по порядку:

**1. Выбран ли тот же порт, на который загружали программу?** Если плату отключали и подключали снова, порт мог измениться. Убедитесь также, что другое приложение не заняло этот порт.

**2. Совпадает ли скорость?** Для приведённого скетча и

монитора нужна скорость 9600.

**3. Загружен ли именно этот скетч?** Проверка кода и его загрузка — разные операции. Убедитесь, что IDE сообщила об успешной загрузке.

**4. Есть ли в программе вывод?** Ищите `Serial.begin(...)` и команды `Serial.print()` или `Serial.println()` внутри выполняющейся части кода.

**5. Используется ли USB-кабель с передачей данных?** Если плата получает питание, но компьютер не видит её, причина может быть в кабеле или настройке драйвера.

Если вместо текста идут бессмысленные знаки, начните со скорости монитора. Если текст появился, но значение не меняется при затенении или освещении датчика, проверьте подключение **АО**, общий контакт земли и правильность номера входа в программе. Убедитесь, что провод подключён к **А0**, а не к цифровому выходу **DO** или другому гнезду. Порядок **VCC**, **GND** и сигнала сверяйте по маркировке и документации модуля — не полагайтесь только на цвет проводов.

Значения, постоянно близкие к нулю или верхней границе диапазона, тоже повод проверить схему: возможно, вход замкнут на землю или питание, перепутан контакт, либо выход датчика не соответствует пределам платы. Но не объявляйте число «неправдоподобным» только потому, что оно не похоже на ожидаемую яркость. Пока это сырое показание, а не физическая единица измерения.

**Что делать с ошибкой компиляции**

Если IDE выделила строку и сообщила об ошибке, прочитайте сообщение сверху: нижняя фраза вроде `exit status 1` часто лишь сообщает, что сборка не удалась, но не объясняет причину. Ошибка может быть в пропущенной скобке, лишнем символе или названии переменной. Если программа использует библиотеку, IDE может сообщить, что файл не найден. В таком случае проверьте, та ли библиотека нужна для примера, и установите её через менеджер библиотек или способом, описанным производителем.

Особенно важно не пытаться «лечить» ошибку компиляции перестановкой проводов. На этом этапе программа ещё не дошла до платы. И наоборот, если загрузка прошла успешно, но датчик не реагирует, бессмысленно бесконечно менять библиотеки, пока не проверено соединение. Диагностика становится проще, когда вы отделяете программную часть от электрической.

### **Сделайте первое изменение**

Когда исходный скетч заработал, измените только одну вещь. Например, замените `delay(1000)` на `delay(2000)`. Показания теперь будут появляться примерно раз в две секунды. Или сделайте строку понятнее:

```
```cpp
Serial.print("Показание АО: ");
Serial.println(rawValue);
```

Снова загрузите программу и проверьте, что данные про-

должают обновляться. Именно так появляется уверенность: вы не просто запустили случайный пример, а понимаете, за что отвечает небольшое изменение.

Для удобства записи можно добавить отметку времени, прошедшего с момента старта платы:

```
```cpp
```

```
Serial.print(millis());
```

```
Serial.print('\t');
```

```
Serial.println(rawValue);
```

Теперь в строках будет сначала время в миллисекундах, затем показание. Такую запись легче сравнивать и переносить в таблицу. Но помните: после перезапуска платы отсчёт времени начинается заново.

Заведите журнал первых измерений. Достаточно записать дату, плату и модель датчика, место установки, условия наблюдения и несколько значений. Например:

**Условие**

**Показание**

Модуль лежит на столе

418

Затенили ладонью

671

Это лишь пример, а не ожидаемые нормы: у вашего модуля числа и направление изменения могут быть другими. Ценность журнала в том, что он показывает связь между действием и результатом. Если изменения повторяются, датчик реагирует. Если данные скачут сами по себе, стоит проверить проводку, контакт и то, не оставлен ли аналоговый вход фактически неподключённым.

Монитор порта — не полноценный приборный экран и не обязательно архив показаний. Это первое окно наблюдения, простой способ увидеть, что программа и плата общаются. Позже появятся графики, пересчёт в привычные единицы и сохранение данных. Но надёжные измерения начинаются не с красивой визуализации, а с понятного ответа на вопрос: что именно сейчас считывается и почему число меняется?

### **Первый результат — не случайность**

Если экран ожил, перед вами не просто набор цифр. Это результат нескольких согласованных шагов: выбрана подходящая плата, программа собрана и загружена, монитор настроен на нужную скорость, а датчик подключён к тому входу, который читает код. Если экран молчит, такая же последовательность проверок помогает найти причину без паники и бессистемной смены компонентов.

Запомните простой цикл: **прочитать сигнал, при необходимости обработать его, вывести результат** . Сначала добейтесь стабильного вывода, затем меняйте формат или частоту и наблюдайте, что изменилось. А когда число начнёт повторяемо откликаться на условия в комнате, вы уже сделали главное: превратили собранную схему в работающий измерительный инструмент. В следующей главе мы перенесём этот навык на конкретный цифровой датчик температуры, для которого важны не только провода, но и правильный способ обмена данными.

# **Температура под контролем: подключаем и проверяем цифровой датчик**

Однажды датчик на столе показал 27 градусов, хотя комнатный термометр у двери упрямо держался возле 22. Arduino ошиблась? Программа сломалась? Или возле компьютера действительно жарче?

Вполне возможно, что ни датчик, ни программа не ошиблись. Температура у вентиляционной решётки ноутбука и температура воздуха в центре комнаты — не одно и то же. К тому же датчик измеряет прежде всего собственную температуру, а она зависит от того, где и как он установлен. Цифра на экране — не готовый ответ о комнате, а свидетельство, которое нужно правильно прочитать.

В этой главе мы подключим цифровой датчик на примере DS18B20, получим первые показания и проверим их не догадками, а последовательным экспериментом. Если у вас другая модель, ход работы останется полезным, но распиновка, библиотека и способ передачи данных могут отличаться. Здесь особенно важно одно правило: сначала документация производителя — потом провода и код.

**Сначала выясним, что именно у нас в руках**

# Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.