

# Arduino

## для умного сада:

Как автоматизировать полив  
и следить за растениями



и  
Александр Громов

**Александр Громов**  
**Arduino для умного сада:**  
**Как автоматизировать полив**  
**и следить за растениями**

*<https://litres.ru/74594609>  
SelfPub; 2026*

**Аннотация**

Одним растениям нужна влажная почва, другим — редкий полив. Поэтому датчик и насос сами по себе ещё не создают умный сад: сначала нужно понять, кому, когда и сколько воды требуется. Эта книга поможет избежать лишних покупок и собрать систему под ваши условия — от одного забываемого горшка до теплицы с несколькими зонами полива.

Шаг за шагом вы освоите Arduino, подключите датчики и исполнительные устройства, напишете код для автополива и проверите систему на практике. Вы научитесь калибровать измерения, защищать электронику от влаги и находить неисправности — чтобы уезжать из дома спокойнее, а за растениями ухаживать точнее.

Из этой книги вы узнаете:

- как выбрать датчики влажности, насос или электроклапан и безопасно соединить их с Arduino;

- как настроить полив по показаниям почвы, расписанию и потребностям разных растений;
- как сохранять данные, следить за садом удалённо и расширять систему новыми датчиками и зонами.

# Содержание

|                                                                       |    |
|-----------------------------------------------------------------------|----|
| С чего начать: цели умного сада, выбор растений<br>и план участка     | 6  |
| Как растения расходуют воду: основы<br>грамотного полива              | 18 |
| Архитектура системы: от датчика в горшке до<br>автоматического полива | 28 |
| Arduino с нуля: плата, среда разработки и первая<br>программа         | 40 |
| Конец ознакомительного фрагмента.                                     | 50 |

**Александр Громов**  
**Arduino для умного сада:**  
**Как автоматизировать**  
**полив и следить**  
**за растениями**

**Arduino для умного сада: Как автоматизировать по-  
лив и следить за растениями**

*Александр Громов*

# С чего начать: цели умного сада, выбор растений и план участка

В воскресенье вечером Марина поставила на подоконник коробку с датчиками, насосом и проводами. Она представляла, как уезжает на выходные, а сад сам понимает, кому нужна вода, и заботливо поливает всех без исключения. Но когда коробка открылась, выяснилось, что сначала нужно ответить на простой вопрос: кому именно — и сколько?

На подоконнике стояли базилик, алоэ и маленький горшок с розмарином. На балконе росли помидоры. Одни растения получали солнце почти весь день, другие прятались за цветочной стойкой. До ближайшей розетки было далеко, а воду Марина носила из кухни. Купленный комплект не был бесполезным. Просто он отвечал на вопрос «чем автоматизировать?», тогда как Марине сначала следовало выяснить, «что именно мешает моему саду?».

Это частая ловушка начинающего садовода: хочется сразу купить датчики, насос и плату Arduino, чтобы однажды проснуться в мире, где растения всё делают сами. Но умный сад начинается не с платы. Он начинается с наблюдения. «Умный» — не значит сложный, дорогой или полностью автономный. Умным можно назвать даже один датчик, который вовремя напоминает проверить землю в забывае-

мом горшке. Полезность системы определяется не числом проводов, а тем, насколько точно она решает вашу задачу.

### **Сначала — не техника, а желание**

Представьте, что через месяц ваш сад стал удобнее. Что изменилось? Вы реже забываете полить рассаду? Хотите замечать, когда почва в горшках пересыхает слишком быстро? Планируете поливать теплицу во время отъезда? Или мечтаете сравнить условия на солнечной и теневой грядках?

Сформулируйте ответ одним предложением. Например:

- «Хочу вовремя замечать, что пересыхают три горшка с пряными травами».

- «Хочу подавать воду к томатам на грядке, пока меня нет дома».

- «Хочу выяснить, как часто нужно поливать рассаду на застеклённом балконе».

Это разные задачи, и решения у них тоже будут разными. В первом случае может хватить измерения влажности и сигнала. Во втором понадобятся подача воды, продуманный маршрут трубки и защита от протечек. В третьем разумнее сначала собрать наблюдения, а не включать насос.

Хорошая цель описывает наблюдаемый результат. «Сделать сад умным» — вдохновляет, но не подсказывает, с чего начать. «За две недели проверить, насколько показания датчика помогают вовремя замечать пересыхание двух одинаковых горшков» — уже план. Его можно выполнить и оценить.

**Инвентаризация: нарисуйте место, где живут расте-**

## ния

Перед покупкой компонентов составьте карту сада. Не обязательно чертить её красиво: для первого наброска подойдут лист бумаги, карандаш и приблизительные размеры. Важно увидеть участок целиком и отметить обстоятельства, которые легко упустить, пока рассматриваешь только отдельный горшок.

Разбейте пространство на зоны: подоконник, балкон, теплица, грядка, клумба. Если сад находится в нескольких местах, отметьте каждое отдельно. Пронумеруйте растения или группы растений, чтобы не путаться в заметках.

На схеме обозначьте:

- примерные границы и размеры каждой зоны;
- положение растений, грядок, стеллажей и крупных ёмкостей;
- двери, проходы и места, где нельзя мешать движению;
- источник воды и путь от него до растений;
- розетки и возможное расположение оборудования;
- участки солнца и тени;
- уклон, сток воды и места, где после полива остаются лужи.

Пока не требуется точность до сантиметра. Но расстояния имеют значение. Насос, который должен поднимать воду из большой ёмкости и проталкивать её через длинную трубку, работает не так, как насос, поливающий один горшок рядом. Трубка, которая пересекает проход, может оказаться более

заметной проблемой, чем отсутствие автоматизации.

Посмотрите на участок в разное время суток. Где бывает прямое солнце утром? Где — в полдень? Какие растения затеняет стена или соседний куст? Повторите наблюдения в несколько дней, потому что облака и сезон меняют картину. Можно записывать не только место, но и продолжительность прямого солнца: «солнце примерно с девяти до полудня», «яркий рассеянный свет весь день». Приложение на телефоне иногда помогает оценить освещённость, но такие измерения не заменяют наблюдений и не всегда точны. Для начального плана важнее понять устойчивые различия между зонами: солнечная, перемененно освещённая, затенённая.

Добавьте к карте бытовые условия. Где вы берёте воду — из крана, бочки, лейки? Можно ли оставить ёмкость с водой на участке? Есть ли рядом розетка, или придётся использовать автономное питание? Как часто вы бываете дома? Кто сможет проверить систему, если вас не будет? Это не мелочи, а часть проекта.

Электричество и вода требуют осторожности. Электронные компоненты и соединения должны быть защищены от влаги, а сетевое питание нельзя бездумно размещать рядом с мокрой почвой и трубками. На этапе планирования достаточно отметить безопасные места для оборудования и не прокладывать провода там, где их можно задеть лейкой, ногой или садовым инструментом.

**Растения не одинаковы — и это хорошая новость**

Мечта о едином расписании полива понятна: включить систему по таймеру и больше ни о чём не думать. Но универсального графика для всех растений не существует. На потребность во влаге влияют вид растения, размер горшка, состав почвы, температура, ветер, освещённость и стадия роста. Один и тот же горшок может высыхать заметно быстрее в жаркий ветреный день, чем в прохладную пасмурную погоду.

Поэтому сгруппируйте растения не по принципу «всё зелёное сюда», а по сходству условий и потребностей. Отдельно отметьте культуры, которым требуется более равномерно влажная почва, и те, для которых постоянная сырость нежелательна. Если не уверены, не угадывайте: запишите название растения и уточните его потребности по надёжному справочнику или инструкции питомника. Важно учитывать и конкретную ёмкость: маленький горшок на солнце способен высохнуть быстрее, чем большой контейнер с таким же растением в полутени.

Начните с простых вопросов:

1. Какие растения стоят или растут рядом?
2. Одинаковые ли у них горшки и почва?
3. Получают ли они похожее количество солнца и ветра?
4. Вы замечаете, что они пересыхают примерно с одинаковой скоростью?
5. Можно ли поливать их из одной линии, не создавая очевидного риска для кого-то из группы?

Если ответы сильно различаются, не объединяйте растения только ради аккуратной схемы. Розмарин и базилик в соседних горшках могут выглядеть как удобная пара для автоматизации, но это не означает, что им подойдёт один режим. Умная система должна учитывать различия, а не скрывать их за общим таймером.

Для первого проекта лучше выбрать растения с похожими условиями. Два-три одинаковых контейнера с одной культурой дадут понятный результат. Когда группа состоит из растений с близкими потребностями, легче сравнивать состояние почвы и разбираться в показаниях датчика. Это особенно полезно на старте: чем меньше переменных, тем проще понять, что именно сработало.

### **Пример: набор датчиков, который не знал, где стоит солнце**

Вернёмся к Марине. Она купила несколько датчиков влажности и насос, рассчитывая подключить сразу все растения. Но инвентаризация открыла три разные ситуации. Базилик стоял у окна и быстро тянулся к свету. Розмарин занимал небольшой горшок на тёплом балконе. Алоэ жило в тени и поливалось гораздо реже. Помидоры росли в крупных контейнерах, до которых нужно было тянуть трубку через балконную дверь.

Марина начала с покупки техники и поэтому почти пропустила главное: её растения отличались по потребностям, освещению и расстоянию до воды. Если бы она подключила

всё к одному таймеру, система исправно повторяла бы расписание — но это не гарантировало бы, что оно подходит каждому растению.

После составления карты выбор стал очевиднее. Марина решила пока не трогать алоэ и помидоры. Для пробного проекта она выбрала два одинаковых горшка с базиликом, стоявших рядом. Её первой задачей стало не автоматическое включение насоса, а наблюдение: проверить, помогают ли показания датчика вовремя замечать, что земля начинает пересыхать. Насос она оставила на следующий этап — когда станет понятнее, где именно он нужен и как надёжно доставить воду.

Это не шаг назад. Это способ не усложнять вопрос раньше времени.

**Выберите цель, которую можно проверить**

В автоматизации полезно разделять три уровня:

**Наблюдать.** Узнать, что происходит: например, замечать изменения влажности почвы.

**Предупреждать.** Получить сигнал, что пора проверить растение.

**Действовать.** Включить насос или открыть клапан, чтобы подать воду.

Эти уровни можно проходить по очереди. Датчик не становится бесполезным оттого, что сам не поливает. Иногда сначала важно понять, насколько его показания соответствуют реальности. Датчики влажности могут вести себя по-раз-

ному в разных почвах и ёмкостях, поэтому для них важна проверка на конкретном месте. Не стоит воспринимать одно число как универсальную команду: «ниже этого значения поливать все растения».

Сформулируйте цель так, чтобы через некоторое время можно было ответить: получилось или нет? Например:

«В течение двух недель наблюдать за влажностью почвы в двух одинаковых горшках с базиликом, записывать показания датчика и сравнивать их с состоянием почвы при ручной проверке».

Или:

«Проверить на трёх контейнерах с одинаковыми томатами, можно ли обеспечить подачу воды без луж и подтеканий в течение недели наблюдений».

Уточните, что будет считаться успехом. В первом примере — не просто наличие цифр на экране, а понимание, в какие моменты датчик помогает заметить необходимость проверки. Во втором — вода доходит до всех выбранных контейнеров, соединения не протекают, а после запуска системы можно вручную проверить результат.

У автоматизации должен быть и план на случай неудачи: что случится, если насос не включится, трубка отсоединится или датчик даст странные показания? В пробном проекте хорошо оставить человека рядом с системой и возможность полить растения вручную. Автоматизация не отменяет наблюдение; на первых этапах она делает наблюдение более

последовательным.

## **Паспорт сада: рабочая тетрадь вместо догадок**

Заполните паспорт сада до покупки компонентов. Он не обязан быть красивым или окончательным. Это документ, который можно дополнять по мере наблюдений.

### **ПАСПОРТ САДА**

**1. Где находится сад** Тип пространства: подоконник / балкон / теплица / грядка / другое: **Размеры или примерная площадь:** Зоны участка:

**2. Растения** Название растения или группы: **Количество:** Место на схеме: **Тип ёмкости и примерный размер:** Почва или субстрат, если известно: **Особенности ухода, которые уже замечены:**

**3. Условия размещения** Прямое солнце и примерное время: **Тень или затеняющие предметы:** Ветер, жара, сквозняк: **Есть ли сток воды? Куда она уходит?**

**4. Доступ к ресурсам** Источник воды: **Расстояние от воды до выбранных растений:** Возможное место для ёмкости или оборудования: **Розетка или другой источник питания:** Есть ли безопасный маршрут для трубки и проводов?

**5. Режим ухода** Как часто вы обычно бываете рядом с растениями? **Когда чаще всего забываете о поливе?** Кто сможет проверить растения в ваше отсутствие? **Что сейчас доставляет больше всего неудобств?**

**6. Цель автоматизации** Какую одну проблему решаю

первой: **Как пойму, что решение работает:** Сколько времени или наблюдений выделю на проверку: **Что буду делать, если система даст сбой:**

**7. Пилотная зона** Какие растения выбраны: **Почему они подходят для первого опыта:** Какую часть сада пока не автоматизирую:

Паспорт помогает заметить противоречия до того, как они превратятся в лишние траты. Например, в цели написано «поливать всю грядку автоматически», а на схеме видно, что вода доступна только у входа, розетки нет, а культуры в грядке требуют разного подхода. Это не повод отказаться от идеи. Это повод уменьшить первый проект и сделать его выполнимым.

### **Начните с маленькой зоны**

Пилотная зона — это небольшой, понятный участок сада, где безопасно проверить идею. Для начала подойдут два одинаковых горшка, один стеллаж с рассадой или несколько контейнеров с одной культурой. Выбирайте место, которое можно легко осмотреть и полить вручную, если что-то пойдёт не так.

Хорошая пилотная зона отвечает нескольким условиям: растения в ней близки по потребностям; до них несложно добраться; воду можно подать без опасных и неудобных пересечений; результат легко заметить. Не выбирайте для первого опыта весь сад, самое дальнее место и набор растений с совершенно разными условиями. Пусть первый проект бу-

дет небольшим настолько, чтобы вы могли понять его поведение.

Набросайте план выбранной зоны. Например:

``text

Вид сверху. Балкон

**Окно и солнечный участок**

[А] Базилик [Б] Базилик |

одинаковые горшки |

|

проход дверь |

источник воды → |

На рабочей схеме добавьте расстояния, место для оборудования и предполагаемый маршрут трубки. Отметьте, что пока остаётся без автоматизации. Это важная часть плана, а не признак незавершённости. Если вы выбрали два горшка для испытания, остальные растения продолжают жить по привычному режиму.

Перед началом сформулируйте короткий план на листе:

1. Выбираю одну группу похожих растений.
2. Записываю, где они стоят и какие условия получают.
3. Определяю задачу и признак успеха.
4. Проверяю доступ к воде, питанию и безопасному месту для оборудования.

5. Наблюдаю за пилотной зоной и фиксирую результаты.

6. Только затем решаю, какие компоненты действительно нужны.

Так техника становится ответом на конкретный вопрос, а не дорогим способом придумать себе задачу.

### **Первый результат — ясность**

В начале проекта может казаться, что прогресс измеряется количеством купленных деталей. На самом деле ценный первый результат — это ясный план: вы знаете, какие растения входят в проект, что им нужно, какие условия предлагает участок и какой вопрос должна решить автоматика.

Заполните паспорт сада, отметьте на схеме свет, воду, розетки и расстояния. Затем выберите одну небольшую зону и сформулируйте для неё измеримую цель. Не стремитесь избавиться себя от всякого ухода: растения всё равно нужно осматривать, а систему — проверять. Задача умного сада не в том, чтобы убрать человека из сада, а в том, чтобы помочь ему замечать важное и тратить силы разумнее.

Когда место, растения и цель становятся понятны, выбор оборудования перестаёт быть лотереей. Следующий вопрос уже не «какой датчик купить?», а «как понять, когда растению действительно нужна вода?».

# Как растения расходуют воду: основы грамотного полива

Однажды Марина поставила два одинаковых горшка с одинаковыми растениями на разные подоконники. Оба поливала по одному правилу: каждое воскресенье — стакан воды. Через месяц растение у солнечного окна повесило листья, а в горшке, стоявшем в глубине комнаты, земля оставалась сырой, и нижние листья начали желтеть.

Марина решила, что одному растению не хватает воды, а другое, вероятно, заболело. Но причина была проще: календарь видел два воскресенья, а растения жили в совершенно разных условиях. Одно стояло на свету и тепле, другое — в прохладной тени. Для первого недельный промежуток оказывался слишком долгим, для второго — слишком коротким.

Полив по расписанию удобен человеку: он превращает заботу в привычный ритуал. Но потребность растения в воде подчиняется не дням недели, а тому, сколько влаги оно теряет и сколько остаётся в грунте. Чтобы поливать грамотно, полезно сменить вопрос «Когда пора поливать?» на другой: «Что сейчас происходит в этом горшке?»

**Вода: не запас на неделю, а движение**

Растению вода нужна для поддержания упругости тканей,

переноса минеральных веществ и многих процессов, связанных с ростом и фотосинтезом. Корни поглощают воду из почвы, она движется по растению к листьям, а часть её испаряется через крошечные поры на поверхности листа — устьица. Так растение не только теряет влагу, но и помогает себе охлаждаться.

Вода уходит и другим путём: испаряется с поверхности грунта. Поэтому земля в горшке может высыхать быстрее, когда листья активно испаряют влагу, а солнце нагревает почву и стенки ёмкости. Чем интенсивнее водообмен с окружающей средой, тем чаще растению может потребоваться полив.

Но «чаще» не значит «каждый день понемногу». Постоянно мокрый грунт — не знак постоянной заботы. Корням, как и другим живым тканям, нужен кислород. Когда почва долго заполнена водой и в неё плохо поступает воздух, корни могут задыхаться и повреждаться. Растение при этом порой выглядит так, словно ему не хватает влаги: листья теряют упругость, рост замедляется. Вот почему один и тот же признак — например, увядание — не всегда означает одно и то же.

Полив по календарю предполагает, что потребность растения повторяется с одинаковым интервалом. Полив по состоянию учитывает, что скорость высыхания меняется. В жаркую солнечную неделю горшок может высохнуть быстро; в прохладную пасмурную — медленно. Дата в календаре оста-

ётся прежней, а условия уже другие.

## **Почему одинаковые горшки высыхают по-разному**

Представим два горшка одинакового размера с похожей землёй. Даже они не обязательно будут расходовать воду одинаково. На это влияет целая группа обстоятельств.

**Свет и температура.** На ярком свете растение обычно активнее растёт и теряет больше воды через листья. Если солнце нагревает горшок, быстрее испаряется и влага из грунта. Но важно не только количество света: прохладная затенённая комната и тёплый солнечный подоконник — разные среды, даже если на обоих окнах стоят растения одного вида.

**Движение воздуха и влажность.** Поток воздуха от открытого окна, вентилятора или отопления способен ускорить испарение. Сухой воздух тоже увеличивает потерю влаги листьями. Влажность воздуха, напротив, может замедлить её. Впрочем, движение воздуха не всегда плохо: оно помогает листьям не оставаться надолго во влажной неподвижной среде. Важно лишь понимать, что рядом с обогревателем и в спокойном углу горшок высыхает по-разному.

**Состав грунта.** Один субстрат быстро пропускает воду, другой удерживает её дольше. Смеси с рыхлыми крупными частицами и хорошим дренажом обычно быстрее отдают избыток влаги, чем плотная мелкая почва. Торф, кокосовое волокно, кора, перлит и другие компоненты по-разному удерживают воду и воздух. Поэтому одинаковое количество воды

в разных субстратах не обязательно приведёт к одинаковому состоянию корней.

**Размер и материал горшка.** В большом объёме грунта влаги может быть больше, но и небольшое растение не всегда способно быстро её использовать. Если корневая система занимает лишь часть крупного горшка, нижняя земля может долго оставаться сырой. Неглазурованная глиняная ёмкость способна отдавать часть влаги через стенки, а пластиковая — почти не испаряет её таким способом. Отверстия для стока тоже важны: без них лишней воде некуда уйти.

**Само растение и его корни.** Растения отличаются устройством листьев, привычками роста и тем, как приспособлены переживать засуху. У разных видов неодинаковы и предпочтения к влажности грунта. У молодого растения, активно наращивающего листья, потребности могут отличаться от потребностей взрослого растения в период замедленного роста. Имеет значение и корневая система: хорошо освоенный корнями горшок может просыхать быстрее, чем ёмкость с таким же объёмом земли, но малым количеством корней.

Из этого следует важное правило: нельзя переносить наблюдение за одним горшком на все остальные. Даже одинаковые растения могут иметь разный режим, если стоят в разных местах или растут в разном грунте.

**Как понять, что происходит в горшке**

Листья — полезная подсказка, но не точный датчик поли-

ва. Пониженные листья могут указывать на сухую почву, однако иногда так растение реагирует на переувлажнение и повреждение корней. Пожелтение бывает связано не только с водой, но и со старением листа, изменением освещения или другими условиями. Поэтому сначала стоит проверить грунт и горшок, а уже затем делать вывод по внешнему виду.

Для такой проверки не нужны приборы. Начните с трёх простых наблюдений.

**Проверьте влажность не только на поверхности.** Верхний слой может высохнуть, пока глубже почва остаётся влажной. У небольшого горшка осторожно коснитесь грунта пальцем. Для более глубокого горшка можно аккуратно ввести деревянную палочку в землю, стараясь не повредить корни: если она выходит с влажными частицами, внутри ещё может быть достаточно влаги. Такой способ не даёт лабораторной точности, но помогает заметить разницу между сухой поверхностью и более влажным корневым слоем.

**Возьмите горшок в руки.** После полива, когда лишняя вода стекла, запомните его вес. По мере высыхания ёмкость становится легче. Несколько таких сравнений помогут научиться распознавать изменения без всяких измерительных приборов. Особенно удобно это для небольших горшков, которые легко поднять.

**Посмотрите на сток и поддон.** Если вода стоит в поддоне долго, корни могут оставаться в постоянно сырой среде. У многих комнатных растений после полива лишнюю во-

ду лучше слить, но конкретные требования зависят от вида и способа выращивания. Есть растения, которым требуется особый режим, например постоянно влажный грунт; им нельзя автоматически назначать правила для обычных комнатных цветов.

Для многих распространённых растений полезно поливать достаточно равномерно, чтобы увлажнить корневой ком, а затем дать воде стечь. Но это не универсальная команда. Суккулент, орхидея в коре и влаголюбивое растение живут в разных условиях и требуют разных подходов. Грамотный полив — не один приём на все случаи, а соответствие между потребностью конкретного растения и состоянием его грунта.

### **Небольшой опыт вместо догадок**

Самый простой способ узнать свой сад — завести дневник высыхания. Не нужно специально доводить растение до увядания или, наоборот, поливать его ради эксперимента. Наблюдайте обычный цикл: полили, дали лишней воде стечь, затем время от времени проверяете состояние грунта и отмечаете условия.

В течение одной-двух недель выберите несколько растений, которые заметно отличаются местом или размером горшка. Проверяйте их примерно в одно и то же время суток. Записывайте не только факт полива, но и то, что происходило с грунтом. Для сравнения пригодятся такие вопросы:

- Где стоит растение: на солнце, в тени, рядом с отоплени-

ем или открытым окном?

- Каким кажется верхний слой и что происходит глубже?
- Изменился ли вес горшка?
- Сколько времени прошло после предыдущего полива?
- Решили ли вы полить — и на каком основании?

Вот пример короткой записи. Она не задаёт норму для всех растений, а показывает, как наблюдение постепенно становится полезным знанием.

**Дата**

**Растение и условия**

**Состояние грунта и горшка**

**Решение**

3 мая

Растение у солнечного окна; тёплый день

После полива грунт влажный, горшок тяжёлый

Не поливать

6 мая

То же место, несколько часов солнца

Поверхность сухая, глубже грунт ещё влажный; горшок легче

Подождать, проверить завтра

7 мая

Солнечно и тепло

Грунт заметно суше на глубине, горшок лёгкий

Полить, дать воде стечь

7 мая

Похожее растение в глубине комнаты

Грунт остаётся влажным, горшок тяжёлый

Не поливать

Через несколько таких циклов вы увидите не «правильное число дней», а собственный рисунок высыхания. Возможно, растение у окна просыхает заметно быстрее. Возможно, большой пластиковый горшок в прохладной комнате остаётся влажным дольше. Это не готовый график на все будущие месяцы: с изменением сезона, освещения и роста условия снова поменяются. Но дневник даст исходную точку, с которой проще заметить перемену.

Чтобы наблюдения были сравнимыми, отмечайте и обстоятельства. Если в один день стояла жара, а в другой неделю шли пасмурные дни, два интервала между поливами нельзя считать одинаковыми условиями. Полезнее записать: «грунт просох на глубине за пять дней после тёплой солнечной недели», чем сделать вывод: «этому растению всегда нужна вода раз в пять дней».

**Упражнение.** Выберите два-три разных горшка и понаблюдайте за ними один обычный цикл между поливами. Не

меняйте резко привычный уход, не переставляйте растения специально и не ждите появления тревожных симптомов. Ваша задача — заметить, как отличаются место, влажность грунта и вес ёмкости. После полива запишите исходное состояние, а затем отмечайте изменения до момента, когда растение по его потребностям действительно пора полить.

Если сомневаетесь, не превращайте неопределённость в эксперимент на выносливость. Сначала изучите требования конкретного вида и проверяйте влажность особенно осторожно, если растение плохо переносит пересыхание. Дневник нужен не для того, чтобы удержаться от воды любой ценой, а чтобы принимать решение внимательнее.

### **Наблюдение превращает календарь в подсказку**

У календаря всё же есть своё место: он напоминает проверить растения. Но проверка и полив — не одно и то же. Можно назначить себе, например, день обхода сада, а затем для каждого горшка решить, нуждается ли он в воде. Так регулярность остаётся, но одинаковая команда не выдаётся растениям с разными условиями.

В будущем к наблюдениям можно добавить датчик влажности. Однако и он измеряет только состояние среды в определённой точке, а не «жажду» растения напрямую. Показание зависит от грунта, расположения датчика и места, где проходят корни. Поэтому электронный сигнал полезнее понимать как ещё одну подсказку, которую нужно сопоставить с поведением конкретного горшка.

Вспомним Марину. Её ошибка была не в том, что она завела расписание. Ошибка — в предположении, будто календарь знает о двух подоконниках больше, чем сама почва. Когда она начала проверять грунт и записывать, как быстро сохнут горшки, забота перестала быть одинаковой для всех и стала точной для каждого.

Умение поливать начинается с малого: заметить, что один горшок легче другого, что солнце изменило привычный ритм, что поверхность сухая, а глубина ещё влажная. Растение не обязано сообщать о потребности строго по воскресеньям. Зато оно оставляет подсказки — в грунте, в весе ёмкости, в свете и температуре вокруг. Научившись читать их, вы перестаёте угадывать и начинаете ухаживать осмысленно.

# Архитектура системы: от датчика в горшке до автоматического полива

Однажды утром вы открываете окно и видите: на экране Arduino — аккуратное число, насос тихо жужжит, а земля в горшке всё такая же сухая. Техника будто бы работает. Но растение этого не замечает.

Вы проверяете трубку — она пережата краем подоконника. Или насос тянет воздух, потому что уровень воды в бутылке упал ниже входного отверстия. Или датчик установлен у стенки горшка и сообщает о влажной земле там, где корни почти не растут. По отдельности детали исправны, а полива нет.

Так часто выглядит первый умный сад: датчик куплен, плата подключена, насос найден, код загружен. Однако между «система что-то измерила» и «растение получило воду» лежит целая цепочка. Если представить её как набор деталей на столе, легко потерять самое важное — связи между ними.

Чтобы увидеть эти связи, начнём с одного горшка. Не с целой теплицы и не с десятка датчиков, а с растения, которое вы хорошо знаете. Например, с базилика на кухонном подоконнике. У него есть грунт, корни, потребность во влаге и привычка не сообщать о проблеме понятными словами. Наша задача — построить систему, которая наблюдает за ча-

стью условий, принимает ограниченное и безопасное решение, подаёт воду и затем проверяет, что изменилось.

## **Два маршрута: сигнал и вода**

В автоматическом поливе одновременно движутся два потока.

Первый — **путь данных:**

**грунт → датчик → Arduino → правило принятия решения.**

Второй — **путь воды:**

**резервуар → насос или клапан → трубка → грунт.**

Между ними есть связующее звено: решение контроллера. Arduino получает показания, сравнивает их с заданными условиями и включает исполнительное устройство — например, насос. Но включение насоса ещё не доказывает, что вода дошла до горшка. Поэтому система должна не только действовать, но и проверять результат.

Полная цепочка выглядит так:

**измерение → обработка → решение → действие → проверка.**

Эта последовательность важнее конкретной модели датчика или платы. Если её нет, автоматизация похожа на человека, который умеет нажимать кнопку, но не знает, зачем и что произошло после нажатия.

## **Что именно сообщает датчик**

Почвенный датчик не измеряет «жажду базилика». Он сообщает о локальном состоянии среды рядом со своей

чувствительной частью. В зависимости от модели это может быть электрический сигнал, меняющийся с влажностью грунта. Arduino преобразует сигнал в число, с которым можно работать.

Но одно число не описывает весь горшок. У поверхности земля может быть сухой, а глубже — влажной; возле капельницы — мокрой, с противоположной стороны — почти сухой. На показания также влияют состав субстрата, плотность укладки, соли и положение самого датчика. Поэтому корректнее думать не «датчик сказал, что растению нужна вода», а «датчик оценил состояние грунта в конкретной точке».

Это различие меняет подход. Вы выбираете место установки, наблюдаете показания именно в своём горшке и сопоставляете их с тем, как грунт выглядит и ощущается. Если используется аналоговый датчик, полезно записать несколько значений для сухого и недавно политого субстрата. Универсального числа, которое подходит всем горшкам, обычно нет.

Есть ещё один важный момент: датчик нужно правильно подключить к контроллеру. В самой простой системе он получает питание и передаёт измерительный сигнал на вход Arduino. Такой элемент называют **входом**: он приносит системе сведения о внешнем мире. К входам также относятся кнопка ручного полива, датчик уровня воды в резервуаре или датчик протечки в поддоне.

## **Контроллер: не волшебник, а диспетчер**

Arduino — это **контроллер**. Представьте небольшую диспетчерскую: на один экран приходят сообщения от датчиков, а затем по заданным правилам отправляются команды насосам, клапанам, светодиодам или дисплею.

Контроллер сам по себе не знает, какой полив полезен базилику. Он выполняет инструкцию, которую вы ему дали. Если правило звучит как «при низком показании включать насос навсегда», система исполнит именно его — даже если резервуар уже пуст, трубка выскочила, а датчик сломался.

Поэтому между измерением и действием нужна **логика**. В простейшем случае она отвечает на вопросы:

- Показание похоже на правдоподобное?
- Сухой ли грунт по выбранному для этого горшка порогу?
- Не поливали ли растение совсем недавно?
- Есть ли вода в резервуаре?
- Не достигнуто ли ограничение по времени работы насоса?

Условия должны быть сформулированы до подключения насоса. Чем яснее правило на бумаге, тем меньше вероятность, что вы будете искать ошибку одновременно в коде, проводке и трубках.

## **Что происходит, когда Arduino включает насос**

Arduino обычно не должна питать насос напрямую со своего вывода. Вывод платы предназначен для управляющего сигнала, а мотору требуется отдельное питание с подходя-

щими параметрами. Между контроллером и насосом ставят управляющий элемент — например, MOSFET-модуль или подходящий транзистор, а в некоторых конструкциях используют реле.

Упрощённо цепь управления выглядит так:

**вывод Arduino → драйвер → насос.**

Сам насос получает энергию от подходящего источника питания. Для низковольтного насоса важно соблюдать его рабочее напряжение и допустимый ток; для электродвигателя постоянного тока в схеме обычно предусматривают защиту от выбросов напряжения при выключении. Конкретное подключение зависит от драйвера и насоса, поэтому его нужно сверять с документацией компонентов.

Не подключайте насос к выводу Arduino в надежде, что плата «как-нибудь потянет». Перегрузка может повредить плату или вызвать нестабильную работу. В простом садовом проекте разумно держаться низковольтного питания, а электрическую часть располагать так, чтобы вода не могла попасть на соединения. Резервуар и трубки находятся рядом с растением; электроника должна быть защищена от капель и случайных протечек. Необходимость в отдельном питании — не досадная сложность, а часть архитектуры системы.

Насос выбирают не только по принципу «он качает». Нужно проверить, подходит ли его производительность, хватает ли напора, может ли он работать с выбранной трубкой и не перегреется ли при предполагаемом режиме. Для небольшо-

го горшка часто достаточно короткого включения, но точное время устанавливают испытанием: объём подачи зависит от насоса, высоты подъёма воды, длины и диаметра трубки.

### **Почему недостаточно команды «поливать»**

Допустим, датчик сообщил, что грунт достаточно сухой. Arduino проверила показание и включила насос на несколько секунд. Вода прошла по трубке и попала в горшок. На этом простая система могла бы остановиться.

Но правильно спроектированная система задаёт следующий вопрос: **что теперь?**

Вода не всегда мгновенно распределяется по субстрату. Если сразу снова считать датчик, он может ещё не заметить изменение, и насос получит повторную команду. Если датчик стоит близко к капельнице, он, наоборот, быстро покажет влажность, хотя другая часть корневой зоны остаётся сухой. Поэтому после полива полезна пауза на впитывание, а затем — повторное измерение.

Этот цикл и называется **обратной связью**: система совершает действие, получает новые данные и оценивает результат. Насос включился — это команда. Показание датчика изменилось после паузы — уже свидетельство того, что среда отреагировала. А если добавлен датчик потока, можно получить более прямое подтверждение, что вода действительно движется по трубке.

Для первого проекта достаточно простой обратной связи: полить короткой порцией, подождать, снова измерить

грунт. Важно не превращать проверку в бесконечный опрос с немедленными повторными включениями. Заранее задайте интервал ожидания и условия для нового полива.

### **Ограничения, которые защищают систему**

Автоматике нужны не только правила включения, но и правила остановки. Даже простой проект должен предусматривать ограничения по времени и безопасное поведение при непонятных показаниях.

Например, система может работать так:

1. Считать несколько показаний датчика и получить устойчивую оценку.
2. Проверить, что значения находятся в правдоподобном диапазоне.
3. Если грунт достаточно влажный, оставить насос выключенным.
4. Если грунт сухой и с прошлого полива прошло достаточно времени, включить насос на короткий заданный интервал.
5. Отключить насос независимо от остальных условий по достижении максимального времени работы.
6. Подождать, пока вода впитается, затем выполнить повторное измерение.
7. Если датчик сообщает невозможные значения или вода в резервуаре закончилась, не продолжать полив и сообщить об ошибке, если система умеет это делать.

Ограничение по времени — простой предохранитель.

Оно не гарантирует, что полив идеален, но не позволяет ошибке в коде включить насос на неопределённый срок. Полезно также ограничить частоту полива: если сенсор зашумел или установлен неудачно, система не должна запускать насос каждую минуту.

Для датчика можно предусмотреть проверку диапазона и поведения. Если показания внезапно стали постоянно максимальными или минимальными, это повод заподозрить обрыв провода, плохой контакт или неисправность. Надёжное правило для сомнительного сигнала — не поливать вслепую. Лучше остановиться и попросить человека проверить установку, чем бесконтрольно подавать воду.

Порог включения и порог возврата не обязательно делать одинаковыми. Если система включается при одном значении и выключается при почти таком же, небольшие колебания сигнала могут многократно переключать насос. Разница между порогами помогает избежать этого «дребезга» решения. В системах с поливом порциями дополнительно помогает пауза после каждого цикла.

## **Базовая архитектура одного горшка**

Для первого испытания достаточно минимального набора:

- датчик влажности, установленный в корневой зоне;
- плата Arduino;
- драйвер, совместимый с выбранным насосом;
- низковольтный насос и подходящий источник питания;

- резервуар с водой;
- трубка, направленная в горшок;
- программное правило с ограничением времени и паузой перед повторной проверкой.

Схему можно нарисовать так:

```text

## **ПУТЬ ДАННЫХ**

**Грунт → датчик → вход Arduino → проверка показаний → правило решения**

**↓ ПУТЬ ДЕЙСТВИЯ Резервуар → насос → трубка → вода в горшок ← драйвер ← выход Arduino ↓ пауза на впитывание и повторное измерение**

Схема нарочно показывает сразу два маршрута. Вода не идёт «из Arduino»: плата только отдаёт управляющий сигнал драйверу. А данные не заканчиваются на показании датчика: они должны пройти проверку и стать частью правила.

На бумаге отметьте не только компоненты, но и связи: какой провод передаёт сигнал, откуда получает питание насос, где физически находится резервуар, куда направлена трубка и где в горшке установлен датчик. Так проще заметить, например, что при отключении питания Arduino насос тоже должен перейти в безопасное состояние или что трубка может вытянуться из горшка.

## **Расширение — только после работающей основы**

Когда система на одном горшке надёжно измеряет грунт, дозирует воду и проходит повторную проверку, архитектуру

можно расширить. В резервуар добавляют поплавковый датчик уровня, чтобы не запускать насос без воды. Датчик потока помогает проверить, движется ли вода по трубке. Датчик протечки в поддоне может обнаружить перелив. Экран или светодиод сообщит, что полив выполнен, а беспроводной модуль позволит передавать показания на телефон.

Каждое расширение решает конкретную задачу и добавляет новые связи. Поплавок становится дополнительным входом; индикатор — выходом; программа получает ещё одно условие. Если подключить всё сразу, не проверив базовую систему, при неполадке будет трудно понять, виноват ли датчик, код, питание, насос или новая функция.

Масштабирование — не просто повторение одного горшка несколько раз. Несколько растений могут отличаться по субстрату, расположению датчиков и допустимым режимам полива. Для каждого канала нужно решить, будет ли у него свой датчик и отдельный управляющий выход, как распределяются вода и питание, и что произойдёт при отказе одной части системы. Начать с одного растения — не компромисс, а способ получить понятный опыт до того, как проект станет сложнее.

### **Практика: нарисуйте свой маршрут**

Возьмите лист бумаги и изобразите один горшок. Слева нарисуйте путь данных, справа — путь воды. Между ними разместите Arduino и блок «правило решения». Затем подпишите:

1. Что именно измеряет ваш датчик и где он будет установлен?
2. Как система поймёт, что показание правдоподобно?
3. Какое условие разрешает полив?
4. Что физически подаёт воду и откуда получает питание?
5. Как система ограничит длительность полива?
6. Сколько она подождёт перед повторной проверкой?
7. Что произойдёт, если датчик отключится, резервуар опустеет или вода не пойдёт?

Если на какой-то вопрос пока нет ответа, это не провал проекта, а полезное открытие. Вы нашли неясность до того, как она превратилась в мокрый подоконник или сгоревший компонент.

Теперь представьте первое испытание без растения: датчик лежит в субстрате, трубка направлена в мерную ёмкость, а насос работает короткими импульсами. Сравните поданную воду с ожидаемым объёмом, убедитесь, что насос выключается вовремя, проверьте, что показания меняются после увлажнения и паузы. Затем испытайте систему на одном горшке под наблюдением. Только после этого оставляйте её работать без постоянного контроля.

В итоге умный полив — это не насос, подключённый к числу на экране. Это согласованная система: датчик сообщает о локальном состоянии грунта, контроллер проверяет данные, правило определяет, допустимо ли действие, исполнительное устройство подаёт воду, а обратная связь помогает

оценить результат. Когда вы видите всю цепочку, легче выбрать детали, найти слабое место и не приписывать датчику знание о потребностях растения.

Нарисуйте маршрут одного горшка — от измерения до повторной проверки. Если каждый переход на этой схеме понятен, у проекта уже есть прочный каркас. Следующий шаг — познакомиться с платой, научиться загружать в неё программу и проверить первую команду на практике.

# Arduino с нуля: плата, среда разработки и первая программа

В субботу утром Ирина подключила новую плату Arduino к ноутбуку, открыла пример программы и нажала кнопку загрузки. Почти сразу на экране появилось: «Загрузка не выполнена». В саду в это время терпеливо ждали рассада томатов и новый датчик влажности — но Ирина ещё ничего к ним не подключала. И это было удачей: первая ошибка случилась на безопасном этапе, когда единственным пострадавшим оказался её настрой.

Причина оказалась простой: в программе была выбрана не та модель платы. Компьютер и Arduino как будто разговаривали на разных языках. Ирина исправила выбор — и через минуту увидела в окне программы первые строки от платы.

Так начинается работа с микроконтроллером: не с магии и не с обязательного умения быть инженером, а с небольших проверяемых шагов. Ошибка при первом запуске — не знак, что электроника «не для вас». Это обычная часть знакомства с новым инструментом. Главное — научиться понимать, на каком именно шаге он споткнулся.

## Небольшая плата — целая цепочка решений

Представьте, что умный сад — это внимательный помощник. Сначала он получает сведения о мире: например, дат-

чик сообщает, насколько влажная почва. Затем Arduino выполняет программу: сравнивает показание с заданным условием и решает, что делать дальше. Наконец, плата может подать сигнал или включить устройство.

Получается знакомый нам цикл: измерить, оценить, действовать, проверить результат. Но сама плата не угадывает, чего хочет растение. Ей нужно дать понятные инструкции — программу — и правильно соединить с датчиками и исполнительными устройствами.

На плате вы обычно найдёте:

- **микроконтроллер** — главный вычислительный элемент, который выполняет программу;

- **USB-разъём** — для связи с компьютером и, на многих платах, подачи питания;

- **цифровые выходы** — контакты, которые могут считывать или выдавать цифровые сигналы;

- **аналоговые входы** — контакты для измерения изменяющегося напряжения, например от некоторых датчиков;

- **контакты питания и GND** — для подключения питания и общего провода;

- **кнопку Reset** — она перезапускает программу на плате.

В зависимости от модели есть и другие возможности: беспроводная связь, дополнительные входы и выходы, разъёмы для подключения модулей. Начинать необязательно с поиска «самой мощной Arduino». Подходящая плата зависит от проекта: сколько датчиков вы подключите, нужны ли Wi-Fi

или Bluetooth, сколько требуется входов и выходов, какое напряжение используют устройства.

Запишите название модели, напечатанное на плате или указанное продавцом. Оно пригодится при настройке среды разработки. Внешне похожие платы могут иметь разные микроконтроллеры, особенности загрузки и рабочее напряжение. Поэтому не стоит считать, что инструкция для одной модели автоматически подходит всем остальным.

**Контрольная точка:** пока к плате не подключены датчики, насосы или другие нагрузки, вы можете спокойно разобратся с компьютером и программой. Для первого запуска понадобится только плата и исправный USB-кабель.

### **Сначала проверьте кабель**

USB-кабель может подходить к разъёму и при этом не передавать данные. Некоторые кабели предназначены только для зарядки. С ними плата иногда загорается, но компьютер не видит её как устройство.

Проверьте кабель, которым точно передавали файлы между компьютером и телефоном, или возьмите заведомо рабочий кабель для передачи данных. Подключите Arduino напрямую к USB-порту компьютера, без переходника или USB-концентратора, если это возможно. Посмотрите, загорелся ли на плате индикатор питания.

Если индикатор не горит, попробуйте другой кабель и другой USB-порт. Если горит — это хороший знак, но ещё не доказательство, что компьютер установил связь: питание

и передача данных не одно и то же.

**Контрольная точка:** плата подключена, индикатор питания горит, кабель заведомо передаёт данные. К насосу её по-прежнему ничего не подключено.

### **Установите среду разработки**

Arduino IDE — это программа, в которой вы пишете код, проверяете его и отправляете на плату. Скачивайте Arduino IDE с официального сайта Arduino. Это помогает избежать старых, изменённых или сомнительных установщиков. Для первого знакомства удобнее настольная версия: в ней проще увидеть настройки платы, порт и сообщения об ошибках.

В IDE находятся три инструмента, которые новичку полезно различать:

- 1. Редактор**, куда вводят программу.
- 2. Проверка (Verify)**, которая пытается собрать код и находит ошибки в нём.
- 3. Загрузка (Upload)**, которая передаёт собранную программу на плату.

Проверка и загрузка — не одно и то же. Если код не собирается, до передачи на Arduino дело ещё не дошло. А если код собрался, но не загрузился, искать проблему нужно уже в связи с платой, её настройках или порте.

У разных моделей Arduino и совместимых плат могут отличаться программные компоненты, необходимые для работы IDE. В среде они часто называются пакетами плат или ядрами (core). Для популярной модели нужные компоненты

могут быть уже доступны; для другой их потребуется установить через менеджер плат. Если IDE предлагает установить поддержку именно вашей модели, следуйте этому предложению и сверяйте название с документацией производителя.

**Контрольная точка:** IDE установлена и запускается. Вы пока не обязаны понимать каждое меню — достаточно найти редактор программы и настройки платы.

## **Выберите модель и порт**

Подключённая Arduino должна определиться как устройство, через которое компьютер сможет обмениваться с платой. Такой канал обычно отображается в IDE как **порт**. Его название зависит от операционной системы и платы.

В Arduino IDE 2.x выбрать плату и порт можно через меню выбора платы или через меню **Tools / Инструменты**. Найдите точное название вашей модели, затем выберите порт, появившийся после подключения. Если IDE показывает несколько устройств, отсоедините плату, посмотрите, какой порт исчез, затем подключите её снова и проверьте, какой появился.

Не уверены, какой пункт соответствует вашей плате? Откройте список устройств до подключения, запомните его, а затем сравните со списком после подключения. На некоторых компьютерах нужный порт отображается с названием производителя или адаптера — это нормально.

Если порт не появился:

- проверьте, что кабель передаёт данные;

- попробуйте другой USB-разъём компьютера;
- переподключите плату и перезапустите IDE;
- проверьте, установлены ли компоненты или драйверы, рекомендованные для вашей модели;
- закройте другие программы, которые могли занять порт.

Некоторые платы с USB-интерфейсом, встроенным непосредственно в микроконтроллер, могут менять порт при переходе в режим загрузки. Если обычные проверки не помогли, сверьтесь с инструкцией именно к этой модели — не устанавливайте наугад драйверы из случайных источников.

**Контрольная точка:** в IDE выбрана именно ваша модель, а не просто похожая, и выбран порт, который появляется при подключении платы.

### **Первая программа: «Я на связи»**

Для первого теста не нужны датчик, светодиод или насос. Попросим плату отправлять сообщение компьютеру. Так вы проверите, что код загружается и обмен данными работает.

Создайте новую программу — в Arduino IDE такие программы называются **скетчами** — и вставьте этот код:

```
```cpp
void setup() {
  Serial.begin(9600); // запускаем обмен данными
}
void loop() {
  Serial.println("Arduino на связи: тест пройден.");
  delay(1000); // ждём одну секунду
}
```

}

Здесь есть три важных элемента.

``setup()`` выполняется один раз после запуска или перезагрузки платы. Внутри него мы начинаем последовательную связь с компьютером командой `Serial.begin(9600);`.

``loop()`` повторяется снова и снова, пока плата включена. В нашем примере при каждом повторении Arduino отправляет строку, а затем ждёт одну секунду.

**Комментарии** начинаются с `//`. Это пояснения для человека: микроконтроллер их не выполняет. Комментарии — не лишние украшения, а способ оставить себе понятные подсказки, особенно когда через несколько недель вы вернётесь к программе.

Число 9600 задаёт скорость передачи данных — условный темп разговора между компьютером и платой. Чтобы стороны понимали друг друга, такую же скорость нужно будет выбрать в мониторе последовательного порта. Это окно в IDE, где можно видеть данные, которые Arduino отправляет компьютеру.

Сначала нажмите **Verify / Проверить**. Если программа собралась без ошибок, нажмите **Upload / Загрузить**. Во время загрузки индикаторы на плате могут мигать — это обычное поведение. Дождитесь сообщения об успешной загрузке. Затем откройте **Serial Monitor / Монитор последовательного порта** и выберите скорость **9600 baud** (9600 бод), если она не установилась автоматически.

Через некоторое время в окне появится:

```
```text
```

Arduino на связи: тест пройден. Arduino на связи: тест пройден. Arduino на связи: тест пройден.

Плата не просто получила питание: она выполнила вашу программу и передала результат компьютеру. Для человека это несколько строк текста; для начинающего проекта — первая замкнутая цепочка «написал → загрузил → проверил».

**Контрольная точка:** сообщение появляется примерно раз в секунду. Если монитор пуст, проверьте, что он открыт для выбранного порта и что установлена скорость 9600.

### **Первая ошибка — и способ её разобрать**

Предположим, на экране появилось сообщение о неудачной загрузке. Не пытайтесь сразу менять всё: плату, код, кабель и настройки одновременно. Тогда невозможно понять, что именно помогло. Вместо этого определите, на каком этапе возникла проблема.

### **IDE сообщает об ошибке при проверке**

Значит, компьютер пока разбирает программу, а не пытается связаться с платой. Прочитайте текст выше итоговой строки: сообщение вроде `exit status 1` само по себе лишь говорит, что сборка не удалась, но не всегда объясняет почему. Ищите первую конкретную подсказку выше: например, опечатку или пропущенную скобку. Сравните код с примером и проверьте, не удалили ли случайно знак `}`.

## **Программа собирается, но не загружается**

Начните с самых частых причин: выбрана другая модель платы, неверный порт, кабель только для зарядки или порт занят другой программой. Проверьте выбор платы и порта заново. Закройте монитор последовательного порта и другие программы, работающие с устройством, затем попробуйте загрузить код ещё раз.

## **Загрузка прошла, а в мониторе ничего нет**

Убедитесь, что монитор открыт для того же порта и скорость установлена на **9600**. Нажмите Reset на плате или закройте и снова откройте монитор: некоторые платы успевают отправить первые данные, пока окно ещё не открыто. В нашем примере сообщения повторяются, поэтому через секунду они должны появиться снова.

## **Вместо текста — бессмысленные символы**

Чаще всего компьютер и плата настроены на разную скорость обмена. Проверьте число в мониторе: оно должно совпадать с 9600 в строке `Serial.begin(9600);`.

Названия ошибок различаются между моделями и версиями IDE. Относитесь к сообщению как к следу на дорожке: оно не обвиняет вас, а помогает сузить круг поиска. Если подсказка непонятна, сохраните её целиком и найдите объяснение для точной модели платы и конкретной версии среды. Не спешите случайным образом менять код — возможно, он исправен, а проблема только в соединении.

## **Маленькое изменение — настоящее понимание**

Когда исходный тест заработал, измените текст сообщения, например на:

```
```cpp
```

```
Serial.println("Сад готовится к автоматизации.");
```

Снова выполните проверку и загрузку. Увидели новую фразу? Значит, вы не просто открыли чужой пример: вы изменили программу и убедились, что именно эта версия теперь выполняется на плате.

Затем можете изменить `delay(1000)` на `delay(250)`. Сообщения станут появляться примерно четыре раза в секунду. Верните значение обратно, если текст в мониторе мелькает слишком быстро. Так вы на практике почувствуете связь между числом в программе и поведением устройства.

`delay()` удобен для первого опыта, но он заставляет плату ждать и не делать других задач. В будущем, когда Arduino нужно будет одновременно считывать датчик и следить за поливом, мы рассмотрим более гибкий способ организовать повторяющиеся действия. Сейчас цель скромнее и важнее: понять, что плата исполняет программу буквально и предсказуемо.

Запишите в блокнот три вещи: модель платы, название выбранного порта и параметры теста, который сработал. Эти заметки сэкономят время, когда вы подключите датчик или вернётесь к проекту после перерыва.

# 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.