



НТО ЮНИОРЫ 2026/27 · СФЕРА

Агротех

Задачник-справочник для подготовки к
отборочному этапу

5–7 класс



Автор-составитель

Артём Осинowski, методист и разработчик детского технопарка «Кванториум»

Детский технопарк «Кванториум», Новосибирск · 2026

Агротех

Задачник-справочник для подготовки к отборочному этапу НТО Юниоры. 5–7 класс

Автор-составитель – **Артём Осиновский**, методист и разработчик детского технопарка «Кванториум».

Нашли ошибку или неточность? Напишите на osinovsky.aa@kvantorium54.ru (рабочая почта): сфера, версия книги (1.0) и номер страницы – исправим в следующей версии. Актуальная версия и список изменений – nto.kvantorium54.ru/knigi. Личная почта для других вопросов – artem081033@gmail.com.

Важно. Это не официальное пособие олимпиады, а книга методиста «Кванториума». Каких заданий ждать на отборе 2026/27, мы не знаем: организаторы их заранее не показывают, а задания разных лет не повторяются. Книга готовит к темам из программы сферы и приёмам решения, а не к конкретным задачам.

Книга помогает школьнику самостоятельно подготовиться к отборочному этапу сферы «Агротех» Национальной технологической олимпиады «НТО Юниоры»: теория с нуля, подробные разборы задач и задачи для самопроверки с ответами.

Большинство задач составлены автором по образовательной программе сферы, часть – по мотивам открытых заданий прошлых лет; источник указан у каждой задачи. Уровни сложности – шкала этой книги, а не баллы олимпиады. Правила, сроки и формат отбора – на сайте junior.ntcontest.ru.

Версия 1.0 от 02.10.2026, сборка 02.10.2026. © Артём Осиновский, 2026.

Содержание

Как устроена эта книга	4
Для кого эта книга	4
Как устроен отбор	4
Что пригодится на отборе и где это в книге	5
Как пользоваться книгой	6
Как вводить ответы на отборе	6
Что понадобится	7
План на месяц до отбора	27
Родителям и наставникам	27
Честно об этой книге	28
Глава 1 Растение как живая система	29
1.1. Что нужно растению	29
1.2. Фотосинтез: еда из воздуха и света	31
1.3. Дыхание: растение тоже тратит	33
1.4. Корни: вода и минеральное питание	34
1.5. Суточные ритмы	36
1.6. Микроклимат: что мерить и зачем	37
1.7. Влажность воздуха и роса	38
1.8. Разбираем задачи	39
Глава 2 Микроклимат теплицы: считаем	51
2.1. Что меряют в теплице и в чём	52
2.2. Переводим единицы: ppm, люксы, время	53
2.3. Влажность воздуха и почвы: проценты	54
2.4. Свет: люксы и часы света	57
2.5. Температура за сутки	59
2.6. Пороги: когда включать и выключать	60
2.7. Разбираем задачи	62
Глава 3 Графики и таблицы агронома	71
3.1. Как читать график	72
3.2. Максимум, минимум, среднее и размах	73
3.3. Строим график по таблице	75
3.4. Электронная таблица считает за тебя	76
3.5. Порог: когда пора действовать	78
3.6. Графики-обманщики	79
3.7. Разбираем задачи	80
Глава 4 Вода и полив	92
4.1. Сколько воды нужно грядке	92
4.2. Литры, баки и слой воды	94
4.3. Насос: литры в минуту	95
4.4. Капельный полив: считаем экономию	96
4.5. Давление воды: глубже – сильнее давит	98

4.6. Испарение и конденсация: почему потеют стёкла	100
4.7. Разбираем задачи	101
Глава 5 Электричество, энергия и датчики	108
5.1. Цепь: ток, напряжение, сопротивление	109
5.2. Закон Ома: считаем резистор	110
5.3. Датчик переводит мир в вольты	113
5.4. Мощность и энергия: солнечная панель	117
5.5. Тепло: обогрев и теплоизоляция	118
5.6. Разбираем задачи	119
Глава 6 Программирование умной теплицы	128
6.1. Алгоритм и исполнитель	128
6.2. Переменные, условия, циклы	130
6.3. Скетч: датчик, порог, действие	133
6.4. Два порога в программе	136
6.5. Машина состояний теплицы	138
6.6. Разбираем задачи	140
Глава 7 Сигналы и обработка данных	155
7.1. Сигнал и шум	156
7.2. Дискретизация: датчик меряет не всё время	159
7.3. Детектирование: по уровню и по частоте	161
7.4. Сглаживание: скользящее среднее	162
7.5. Выбросы: когда датчик соврал	165
7.6. Спектр и фильтр – на пальцах	167
7.7. Кодирование: сигнал несёт число	169
7.8. Разбираем задачи	170
Глава 8 Метеостанция и автоматика	181
8.1. Метеостанция: теплица смотрит на улицу	182
8.2. Анемометр: из оборотов в скорость ветра	184
8.3. Сервомотор и DC-мотор: форточка и вентилятор	186
8.4. Солнце, панель и трекер	189
8.5. Метеостанция управляет теплицей	193
8.6. Разбираем задачи	194
Ответы и решения	203
Справочник	211
Числа, которые используются в книге	211
Перевод единиц	212
Формулы и правила	212
Мини-шпаргалки по инструментам	217
Словарик	222
Шпаргалка: всё «Запомни» из глав	224
Где что найти	229

Как устроена эта книга

Эта книга – тренажёр для отборочного этапа сферы «Агротех» олимпиады «НТО Юниоры». В ней теория с нуля, подробные разборы задач и сразу после каждого разбора – похожие задачи «Реши сам» с ответами в конце книги. Своя теплица и набор с датчиками не нужны: половина задач решается в тетради с калькулятором, остальное – в браузере, в том же симуляторе, что и на курсе сферы.

Для кого эта книга

- Для школьников **5–7 классов**, которые выбрали (или думают выбрать) сферу «Агротех».
- Для тех, кто **ни разу не держал в руках Arduino** и ещё не проходил физику. Ток, напряжение, датчик, программа – всё начинается с самого начала, а первый код разобран построчно.
- Для тех, кто живёт в городе и помидоры видел только на тарелке. Про растения тоже рассказываем с нуля.
- Для **наставников и родителей**, которые хотят помочь. Для них есть отдельный раздел в конце введения.

Как устроен отбор

Таблица В.1

Когда	Что происходит
до 9 ноября	регистрация на олимпиаду (наш совет – успеть до 20 октября, тогда будут обе попытки)
20–30 октября	отбор, попытка 1 (3 часа)
31 октября – 10 ноября	отбор, попытка 2 (засчитывают лучшую из двух)
7–17 декабря	финалы в регионах

- Отбор проходит **дома, онлайн, с компьютера**. Задания открываются в личном кабинете на my.ntcontest.ru и решаются на платформе «Орбита» (orbita.education). Каждый решает сам за себя.
- Задания – это задачи с ответом: ввести число, выбрать верные варианты, сопоставить, расставить шаги по порядку, прочитать график или таблицу, написать короткую программу.
- Разработчик сферы на установочном вебинаре в сентябре 2026 года сказал: вся основная информация для отбора и финала есть в онлайн-курсе сферы. Поэтому книга построена по программе сферы «Агротех. Основы инженерных агросистем». Он же предупредил, что это не тест на зубрёжку: даже на знакомом материале задачи потребуют подумать как инженер.

- На вебинаре для наставников разработчик сказал ещё одну важную вещь: часть задач отбора связана с практикой в симуляторе. Если не умеешь собрать в нём схему и написать простую программу, решать будет очень трудно. Поэтому в главах 5–8 много схем и программ, которые стоит повторить руками.
- За тест бесплатного курса своей сферы дают **+5 баллов**. Курс – на academy.junior.online, вход через «Талант». Бонус нужно включить самому в личном кабинете («Мои привилегии») до конца отбора. Учти: после этого сферу уже не поменять.
- Финал – **командный** и очный: участники приезжают на площадку в своём регионе, собираются в команды и решают на компьютерах теоретические задачи и задачи в симуляторе – собирают схемы, пишут программы, обрабатывают данные. Общая тема сезона – космос, поэтому на финале, по словам разработчика, нужно будет построить теплицу для космической базы, где растениям совсем непривычно.

Совет. Сроки и правила могут уточняться. Главный источник – сайт олимпиады junior.ntcontest.ru и памятка для Новосибирской области nto.kvantorium54.ru. Свежие новости сферы публикуют в канале «Сфера Агротех» в мессенджере MAX.

Что пригодится на отборе и где это в книге

На странице сферы перечислено, что пригодится на отборе. Вот где это в книге:

Таблица В.2

Что пригодится на отборе	Глава
Биология: факторы роста, фотосинтез, дыхание, биоритмы, корневое питание	1. Растение как живая система
Математика: пропорции, проценты, дроби, единицы измерения	2. Микроклимат теплицы: считаем
Математика: графики, временные интервалы, среднее; работа с таблицами	3. Графики и таблицы агронома
Физика: давление жидкостей, испарение и конденсация	4. Вода и полив
Физика: электрические цепи, тепловые явления, преобразование энергии	5. Электричество, энергия и датчики
Информатика: алгоритмы, исполнители, язык и среда программирования	6. Программирование умной теплицы
Анализ данных и обработка сигналов с датчиков (из программы сферы)	7. Сигналы и обработка данных
Метеостанция и альтернативная энергетика (из программы сферы)	8. Метеостанция и автоматика

Главы 1–4 можно читать в любом порядке: хватит тетради, карандаша и калькулятора. Главу 5 прочитай раньше главы 6 – без понятия «напряжение» программа для датчика будет казаться

заклинанием. Главы 7 и 8 – после шестой: в них программа, датчики и расчёты работают вместе, как в настоящей теплице.

Как пользоваться книгой

В каждой главе теория и разборы идут вперемешку: за разбором сразу стоят две похожие задачи «Реши сам», чтобы закрепить приём, пока он свежий в голове.

1. **Теория.** Коротко и по делу, с рисунками. Каждая новая тема начинается с большой картинкой «на пальцах», а самое важное собрано в рамках «Запомни». Все «Запомни» из всех глав повторяются в конце книги, в шпаргалке.
2. **Разбираем задачи.** Задачи похожи на задачи отбора, и к каждой – решение по шагам: что дано, что делаем и почему, где обычно ошибаются, какой ответ получился и похож ли он на правду.
3. **Реши сам.** Сразу после разбора – две задачи на тот же приём, но с другими числами и сюжетом.
4. **Ответы** – в конце книги, в разделе «Ответы и решения». Там только ответ, иногда – короткая подсказка, если задача с подвохом. Полного решения нет специально: если подсказки не хватило, вернись к разбору перед задачей.

Не готов читать всю главу целиком? Есть отдельная короткая книжка «Быстрый старт»: из каждой главы главное, один разбор и несколько задач «Реши сам». Её удобно пройти за неделю, а эту книгу держать под рукой.

У каждой задачи есть уровень – как в обычном задачнике: у трудных задач звёздочка рядом с номером («задача 3*»), лёгкие подписаны словом «разминка». Это уровни нашей книги, а не баллы олимпиады. Сколько задач будет на отборе и сколько стоит каждая, заранее не объявляют.

Совет. Разбор читай с листком бумаги. Закрой решение, попробуй решить сам, а потом сверяйся по шагам. Если в разборе есть схема или программа – собери её в симуляторе. Прочитанная программа забывается к вечеру, запущенная своими руками – помнится до отбора.

У каждой задачи внизу написано, откуда она взялась. «По мотивам задачи отборочного этапа» – значит, похожая задача была на отборе прошлых лет: мы пересказали её своими словами и поменяли числа. «Своя задача по программе сферы» – значит, мы придумали её по теме из программы 2026/27 года. Таких в этой книге большинство, и честно об этом сказано в конце введения.

Как вводить ответы на отборе

Так было на отборочных этапах прошлых лет. На «Орбите» перед каждой задачей читай инструкцию: там сказано, в каком виде нужен ответ.

- **Только число**, без единиц измерения и без знака процента, если не сказано иначе. Смотри, в чём просят ответ: в литрах или миллилитрах, в минутах или секундах, в вольтах или милливольтмах.

- **Дробную часть** на сайте обычно отделяют **точкой**: 24.8 , а не 24,8 . В книге мы пишем по-русски, через запятую.
- **Округляй так, как сказано в условии.** Промежуточные результаты не округляй – только ответ.
- **Показание `analogRead` – всегда целое число** от 0 до 1023. Если у тебя вышло 511,5, значит, в задаче спрашивали что-то другое, или ошибка где-то раньше.
- **Запасы округляй вверх.** Если воды в баке должно хватить, а поливов получилось 9,2, бак нужен на 10.
- **Выбор нескольких вариантов** часто вводят строкой из их номеров – например, 135 . Порядок цифр и нужны ли пробелы, смотри в инструкции к задаче.
- **У каждого могут быть свои числа.** В некоторых задачах условие генерируется для каждого участника отдельно, так что списать ответ не выйдет.
- **Попытки ограничены.** На задачу даётся несколько попыток ввода, и каждая неверная снижает балл за неё. Сначала реши на бумаге или проверь программу в симуляторе, и только потом вводи ответ.

Что понадобится

- Тетрадь в клетку, карандаш, линейка – для глав 1–4 и для графиков.
- Калькулятор. Хватит обычного на телефоне: проценты, дроби, деление.
- Компьютер с интернетом и мышью – для глав 5–8. Схему на планшете собирать мучительно: провода тянутся мышкой.
- **Tinkercad** – онлайн-симулятор, в котором работают курс, отбор и финал сферы. Раздел «Цепи» (по-английски Circuits): собираешь схему из платы Arduino Uno, датчиков и проводов, пишешь программу и смотришь, как всё работает. В курсе сферы есть ссылка на класс «Агротех» и подробная инструкция, как в него войти. Отдельно регистрироваться не нужно.
- **Wokwi** (wokwi.com) – запасной симулятор Arduino в браузере. Если на твоём компьютере он не открывается, хватит Tinkercad. На бесплатной учётной записи запуск иногда ждёт в очереди несколько секунд.
- **Кибериада IDE** (по-английски Cyberiada IDE; ищи по обоим названиям) – редактор, где программу не пишут строчками, а рисуют схемой из состояний и переходов (язык ПРИМС). С ней работают в модулях курса про машины состояний, ссылка – в курсе.
- **Arduino IDE** – по желанию, если дома есть настоящая плата. Писать в ней код удобно и без платы, а потом копировать в симулятор. Для подготовки к отбору хватит симулятора; программу для настоящей платы поможет поставить педагог кружка.
- **Таблицы** – Excel, LibreOffice Calc или любые онлайн-таблицы: для главы 3 и для задач, где данные с датчиков нужно посчитать и построить по ним график.

Эти же четыре инструмента – онлайн-симулятор, Кибериада IDE, Arduino IDE и таблицы – названы в официальных требованиях к площадкам финала сферы «Агротех».

Tinkercad: вход в класс и первая схема

Tinkercad (tinkercad.com) – бесплатный симулятор электронных схем и Arduino прямо в браузере: ставить ничего не нужно. В нём идёт практика курса сферы. Регистрироваться тоже не

надо – курс даёт ссылку на класс и общее имя ученика. Ниже весь путь по шагам, рамка с номером на картинке – куда нажимать.

Шаг 1. Войти в класс. Открой ссылку на класс курса: tinkercad.com/joinclass/XWZDYPNAX (класс курса «Агротех»; та же ссылка – в курсе сферы, модуль 5 «Теория и практика по созданию инженерных агросистем. Часть 1»). Нажми «Присоединиться, используя имя учащегося» (1 на левой половине рисунка). Впиши имя учащегося test (1 на правой половине) и нажми «Перейти в класс» (2).

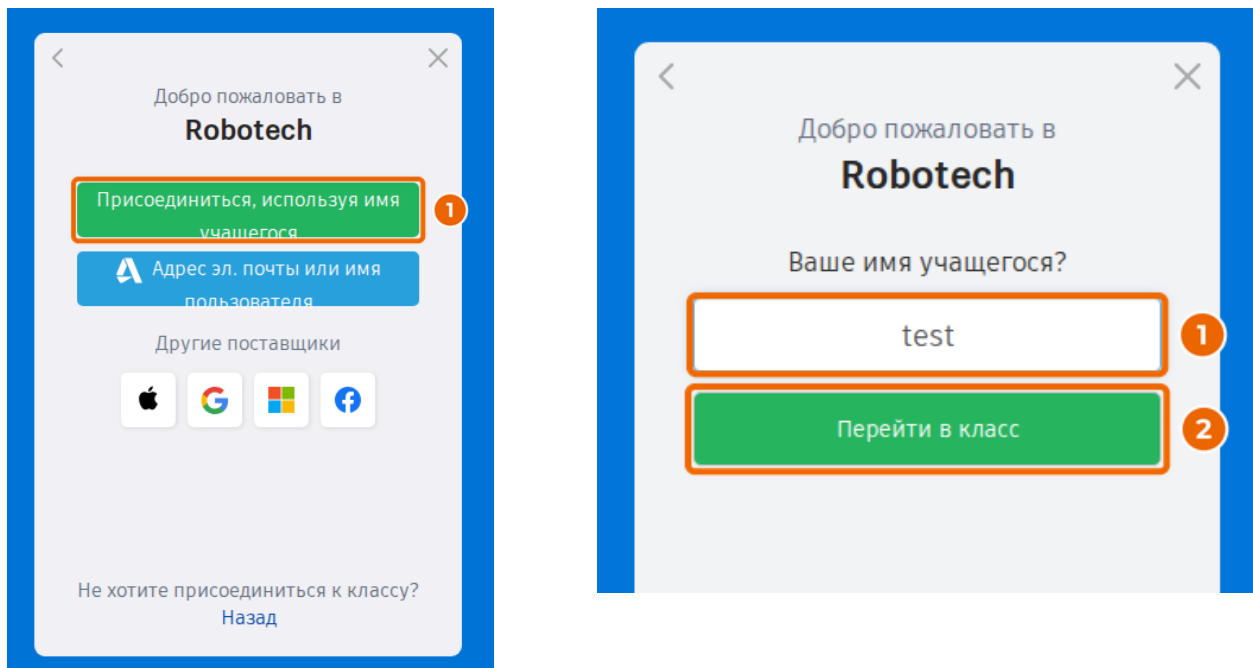


Рисунок В.1 – Вход в класс Tinkercad: слева – выбор способа входа, справа – имя учащегося

Шаг 2. Новая схема. Вверху справа нажми «Создать» (1), в меню – «Цепи» (2). Откроется редактор схем.

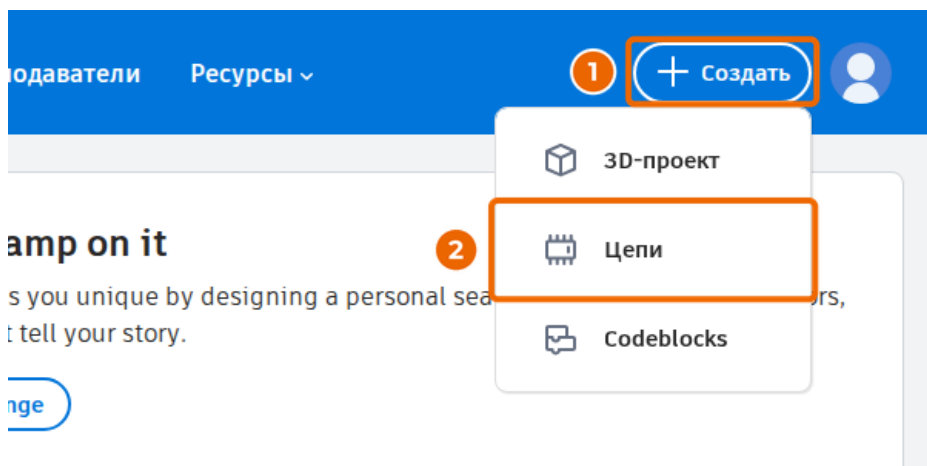


Рисунок В.2 – Новая схема: «Создать» → «Цепи»

Шаг 3. Окно редактора. Вот что в нём где.

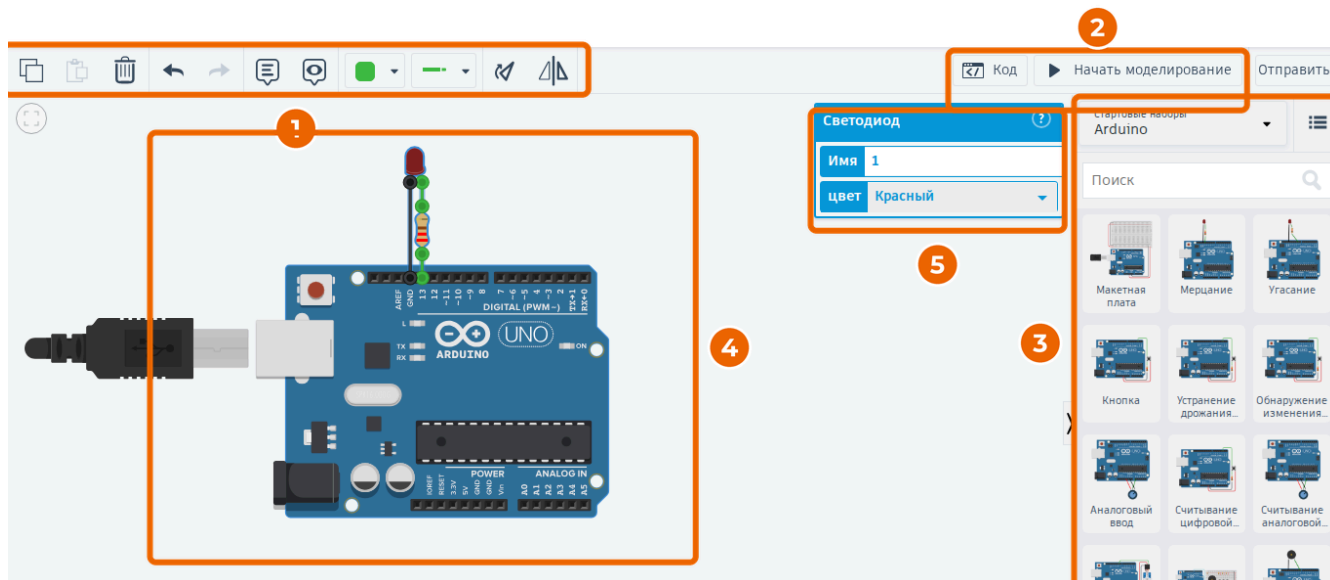


Рисунок В.3 – Редактор схем Tinkercad

1. **Инструменты:** копировать, вставить, удалить, отменить и вернуть действие, повернуть деталь.
2. **«Код»** – программа для Arduino; **«Начать моделирование»** – запуск схемы.
3. **Компоненты:** сверху выбор набора, ниже поиск и сами детали. Деталь перетаскивают мышью на поле.
4. **Рабочее поле** – здесь собирают схему. Провод тянется так: щёлкни по выводу одной детали, потом по выводу другой.
5. **Свойства выбранной детали** – например, цвет светодиода или сопротивление резистора.

Шаг 4. Готовая схема для старта. В списке компонентов выбери «Стартовые наборы» → «Arduino» (1 на левой половине) и перетащи на поле «Мерцание» (1 на правой половине): это Arduino со светодиодом, резистором и уже написанной программой. Рядом лежат и другие готовые схемы – кнопка, сервопривод, ультразвуковой датчик.

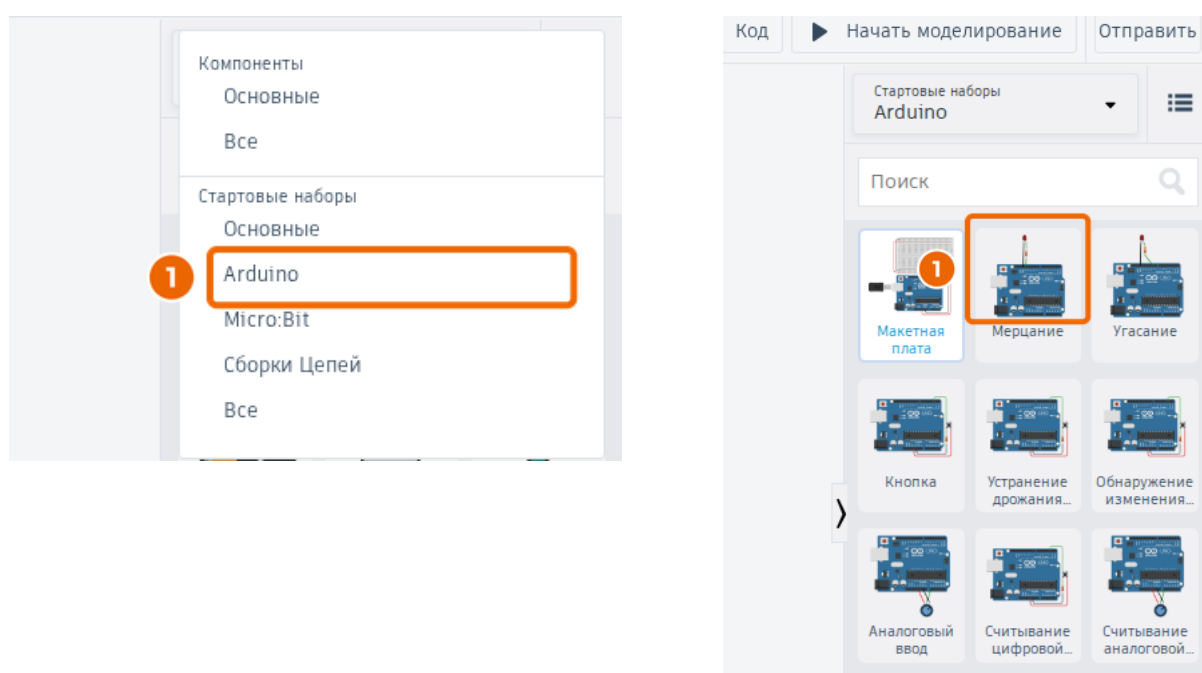


Рисунок В.4 – Стартовые наборы: слева – выбор набора «Arduino», справа – схема «Мерцание»

Шаг 5. Код и запуск. Кнопка «Код» (1) открывает программу. Её можно смотреть блоками или текстом на языке Arduino: переключатель «Блоки» вверху окна кода. Кнопка «Начать моделирование» (2) запускает схему.

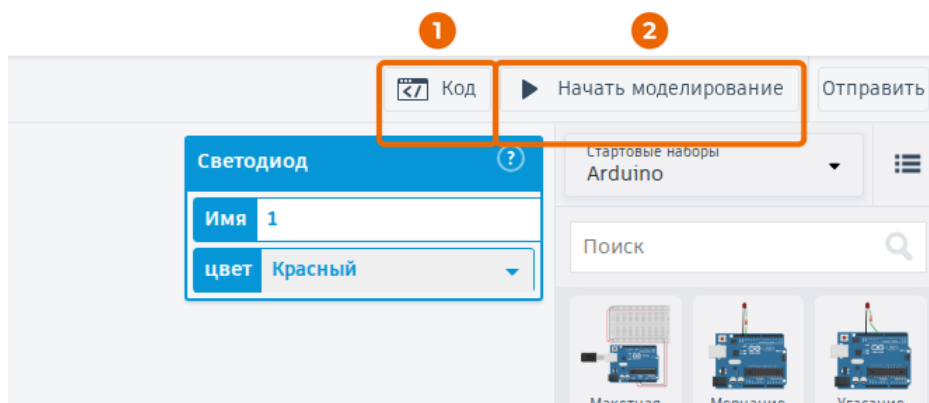


Рисунок В.5 – Кнопки «Код» и «Начать моделирование»

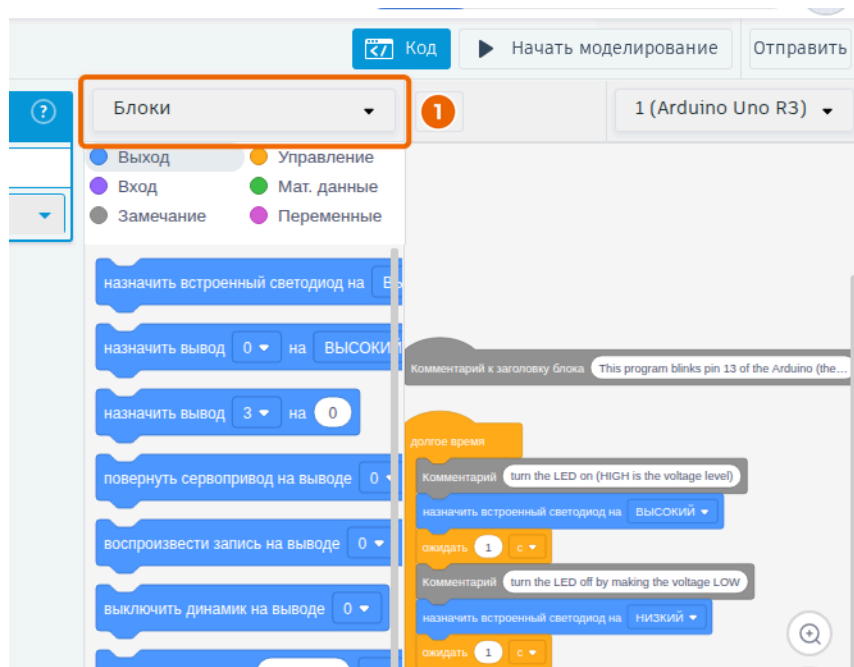


Рисунок В.6 – Программа «Мерцания» блоками

Во время моделирования кнопка называется «Остановить моделирование» (1), а светодиод на плате мигает (2): секунду горит, секунду нет. Остановить – той же кнопкой.

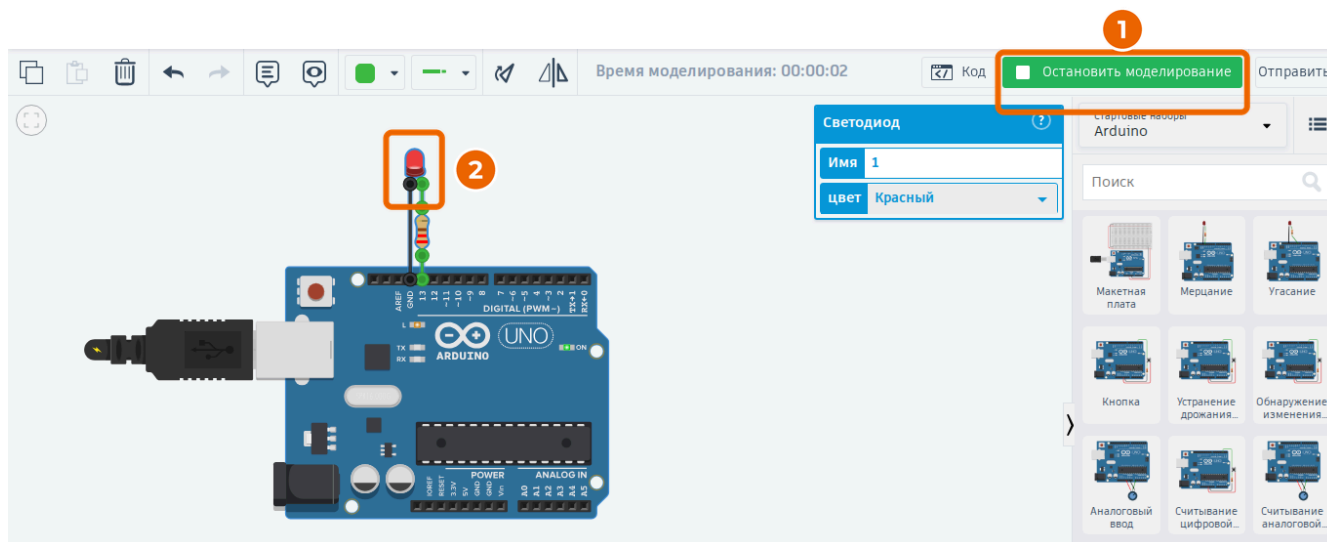


Рисунок В.7 – Схема работает

Важно про имя test . Под ним одновременно сидят все ученики курса: твою схему видят и могут поменять другие, а сохранение не гарантируется. Не пиши туда ничего личного. Если хочешь хранить свои работы, заведите вместе со взрослым свой аккаунт Autodesk – вход «Адрес эл. почты или имя пользователя» на том же экране, где выбирают способ входа.

Кибериада IDE: установка и первая программа

Кибериада IDE – бесплатная отечественная среда, в которой программу не пишут строчками, а рисуют диаграммой на языке ПРИМС: режимы работы – прямоугольники-состояния, смена режима – стрелки-переходы. Ниже весь путь от установки до первой программы «мигалка» – задача из курса сферы. Рамка с номером на картинке – куда нажимать.

Шаг 1. Скачать и поставить. Открой method.insitulab.pro/cyberiaда и нажми «Скачать программу». Выбери файл под свою систему: .exe – Windows, .dmg – macOS, .deb или .AppImage – Linux. Запусти его. Установщик попросит права администратора – он ставит драйверы для плат, так и должно быть; дома нажимай «Да» вместе со взрослым. После установки программа откроется сама.

Шаг 2. Новый документ. Слева – панель «Документ», в ней «Создать...» (1).



Рисунок В.8 – Стартовый экран Кибериады: «Документ» → «Создать...»

Шаг 3. Платформа. Кибериада умеет работать с разными платами, поэтому сначала выбирают, для какой пишем. Дважды щёлкни «Arduino Uno» в списке платформ (1) – она появится в «Выбрано» (2). Нажми «Создать» (3).

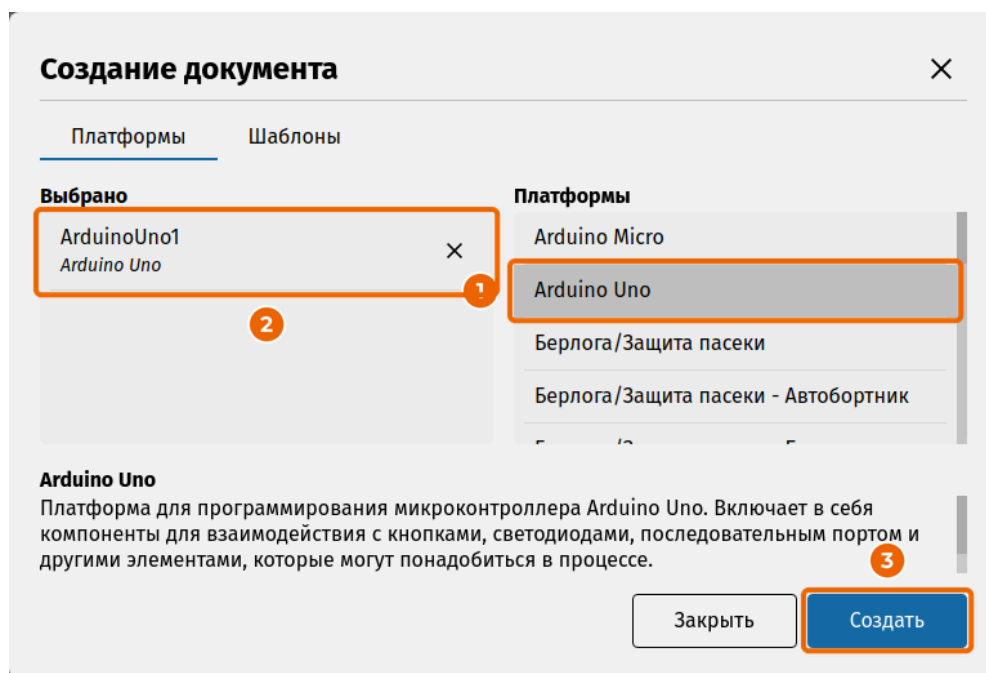


Рисунок В.9 – Выбор платформы для документа

Шаг 4. Компоненты. Компонент – это то, чем программа управляет или от чего ждёт сигнала: светодиод, кнопка, таймер. Нажми «+» у «Компонентов» (1).

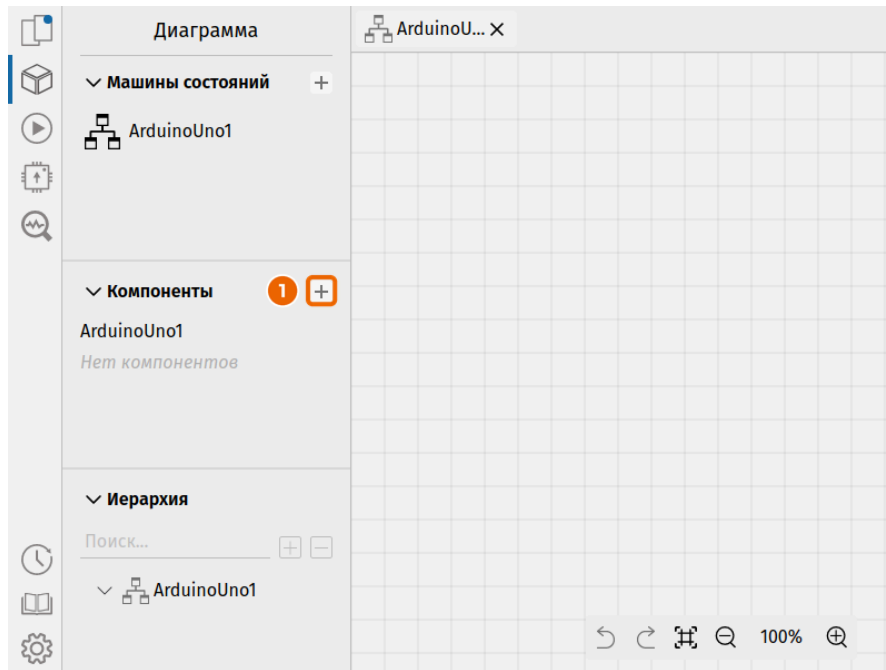


Рисунок В.10 – Окно диаграммы: плюс у «Компонентов»

Выбери «Светодиод» (1 на левой половине) и нажми «Добавить» (2). В форме впиши название «Светодиод» (1 на правой половине) и выбери пин (2) – номер вывода Arduino, к которому подключён светодиод; на занятии курса это 2. Нажми «Применить» (3). Так же добавь «Таймер» – у него пина нет, только название «Таймер».

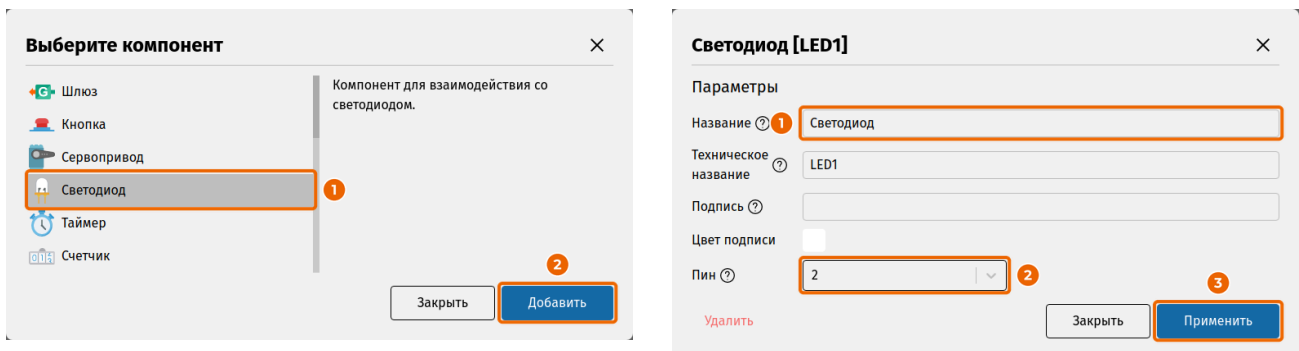


Рисунок В.11 – Новый компонент: слева – выбор «Светодиода», справа – его название и пин

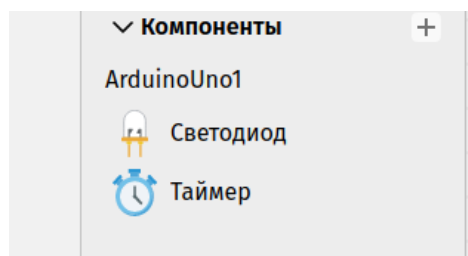


Рисунок В.12 – Оба компонента в списке

Шаг 5. Состояние «Не горит». Дважды щёлкни по пустому месту поля – появится прямоугольник «Состояние». К нему от синего кружка ведёт стрелка: отсюда программа начинает работу. Нажми карандаш (1) и впиши «Не горит», затем Enter.

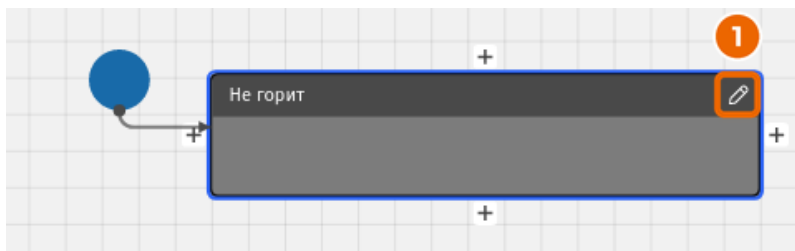


Рисунок В.13 – Новое состояние: карандаш – переименовать

Теперь скажем, что делать в этом состоянии. Дважды щёлкни по нему – откроется «Редактор состояния». Нажми «+» (1).

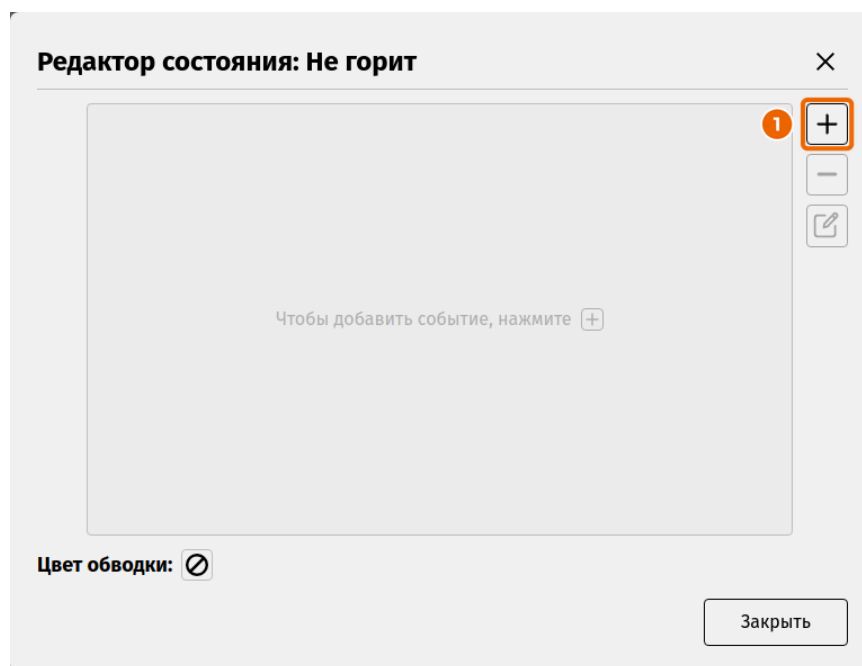


Рисунок В.14 – Редактор состояния

В «Редакторе события» в строке «Когда» уже стоит «Общие → Вход» (1): действия выполняются, как только программа войдёт в это состояние. Так и оставь. Нажми «+» у «Делай» (2).

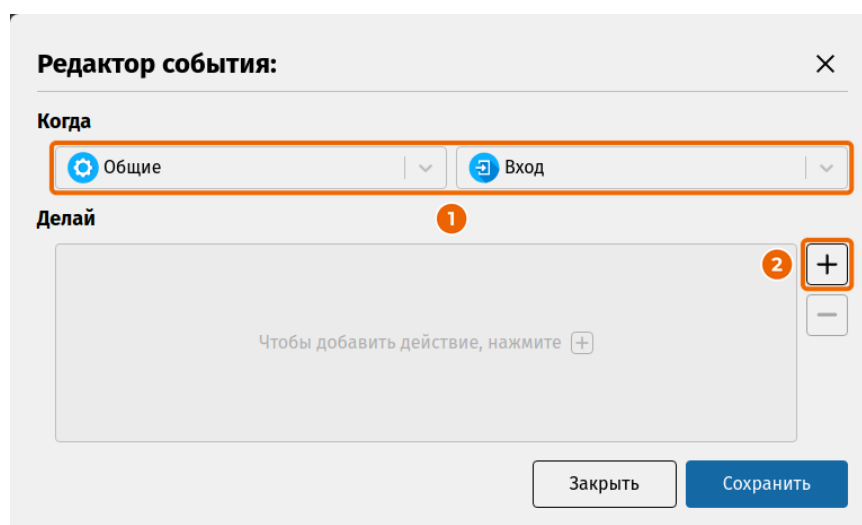


Рисунок В.15 – Редактор события: «Вход» и плюс у «Делай»

Выбери компонент «Светодиод» (1) и действие «Погасить» (2), нажми «Сохранить».

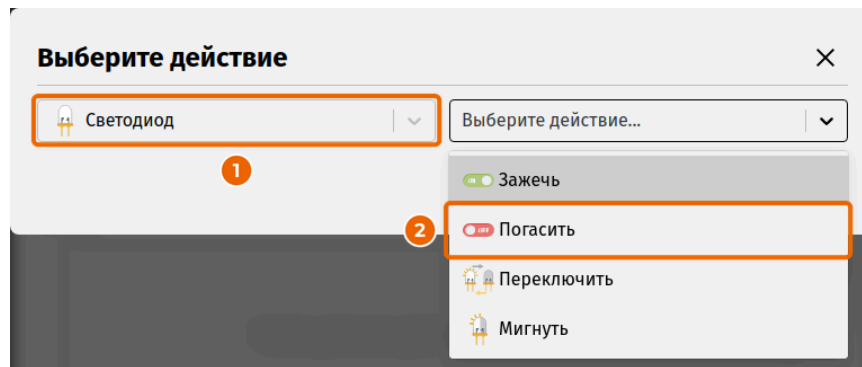


Рисунок В.16 – Действие «Светодиод – Погасить»

Ещё раз «+» у «Делай»: компонент «Таймер» (1), действие «Старт» (2), в поле интервала – 1000 (3), это миллисекунды, то есть одна секунда. Нажми «Сохранить» (4). Переключатель «Интервал» не трогай: он нужен, чтобы брать число из другого компонента, а нам хватит простого числа.



Рисунок В.17 – Действие «Таймер – Старт» на 1000 мс

В «Делай» теперь два действия (1). Нажми «Сохранить» (2), потом «Закреть» в редакторе состояния.

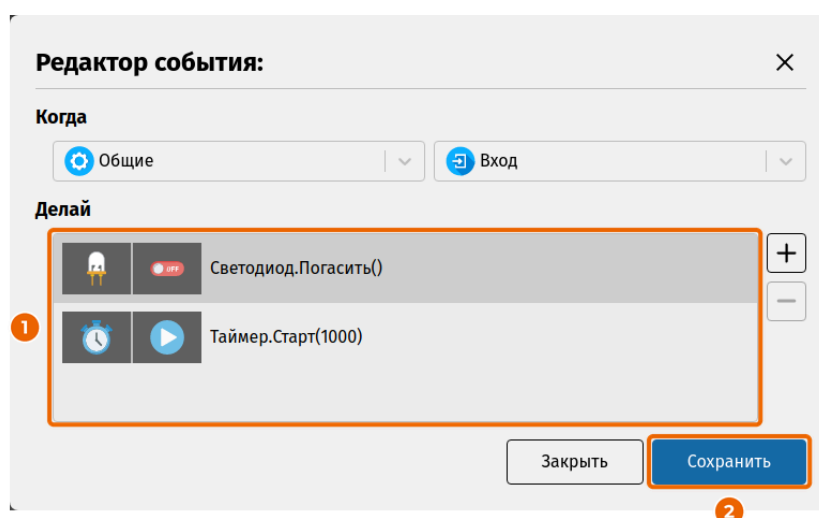


Рисунок В.18 – Событие «Вход» с двумя действиями

Шаг 6. Состояние «Горит». Так же: двойной щелчок по пустому месту, имя «Горит», одно действие – «Светодиод → Зажечь». Таймер второй раз запускать не нужно: на платформе Arduino Uno таймер повторяющийся – запустили один раз, и он срабатывает каждую секунду. На других платформах бывает иначе – смотри описание компонента.

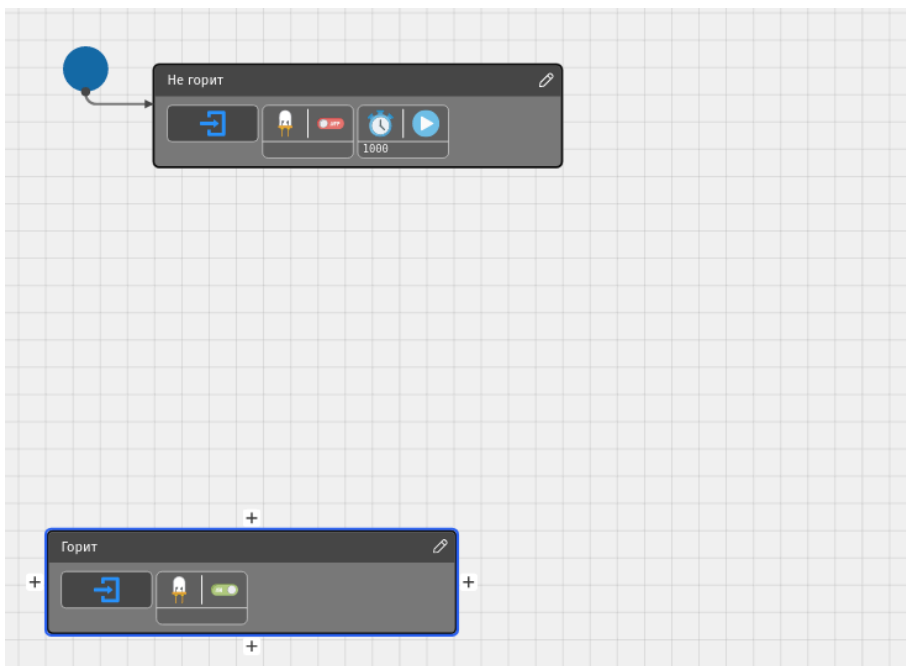


Рисунок В.19 – Два состояния готовы: на каждом видно, что оно делает при входе

Шаг 7. Переходы. Щёлкни по «Не горит» – вокруг появятся плюсики. Потяни мышью нижний плюстик (1) до состояния «Горит» (2) и отпусти.

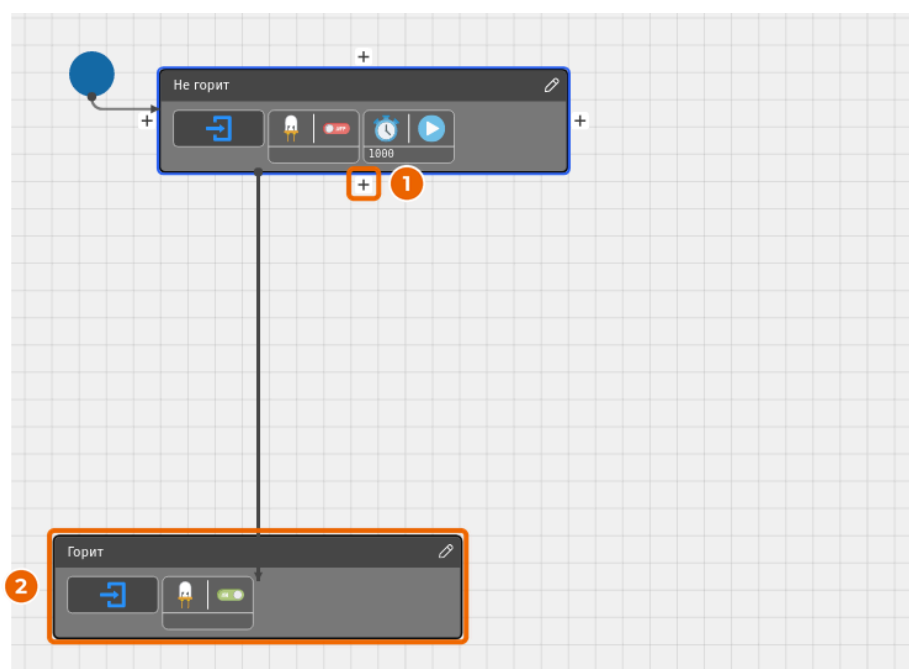


Рисунок В.20 – Тянем стрелку перехода

Откроется «Редактор соединения». В «Когда» выбери «Таймер» (1) и «Тайм-аут» (2) – переход случится, когда таймер отсчитает секунду. Условия и действия не нужны. Нажми «Сохранить» (3).

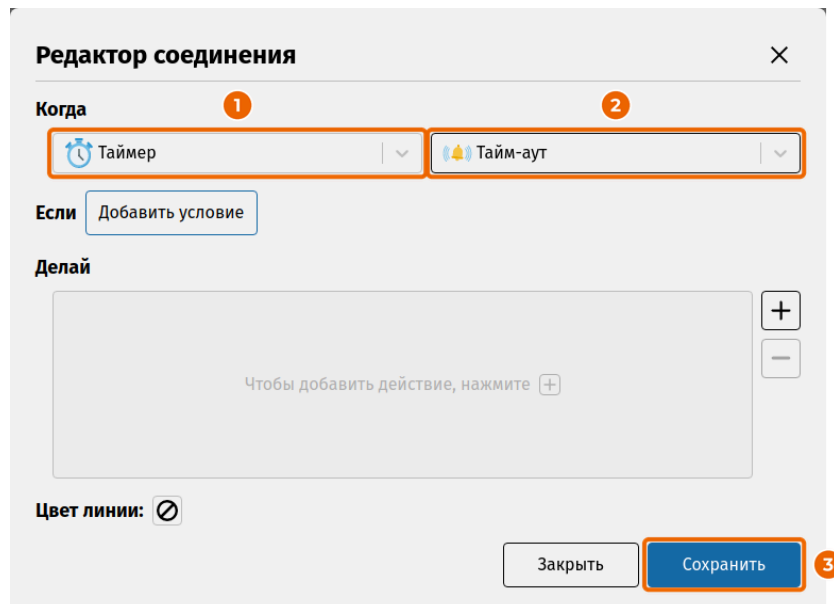


Рисунок В.21 – Переход по тайм-ауту таймера

Так же протяни стрелку обратно – из «Горит» в «Не горит», с тем же событием. Если две стрелки легли одна на другую, перетащи значок перехода мышью в сторону.

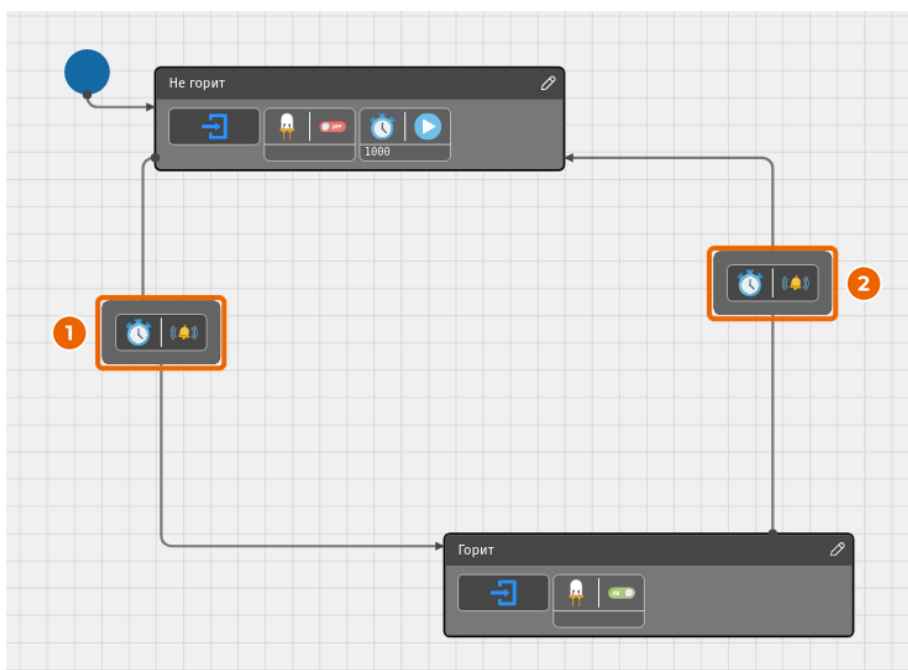


Рисунок В.22 – Готовая «мигалка»: переход «Не горит → Горит» (1) и обратный (2)

Прочитай программу по стрелкам: старт → «Не горит» (погасить, запустить таймер) → через секунду тайм-аут → «Горит» (зажечь) → тайм-аут → «Не горит» – и так по кругу. Светодиод мигает раз в секунду. Сохрани работу: «Документ → Сохранить как...», программа сохранится файлом `.graphml`.

Карта значков слева. По ним переключают панели.

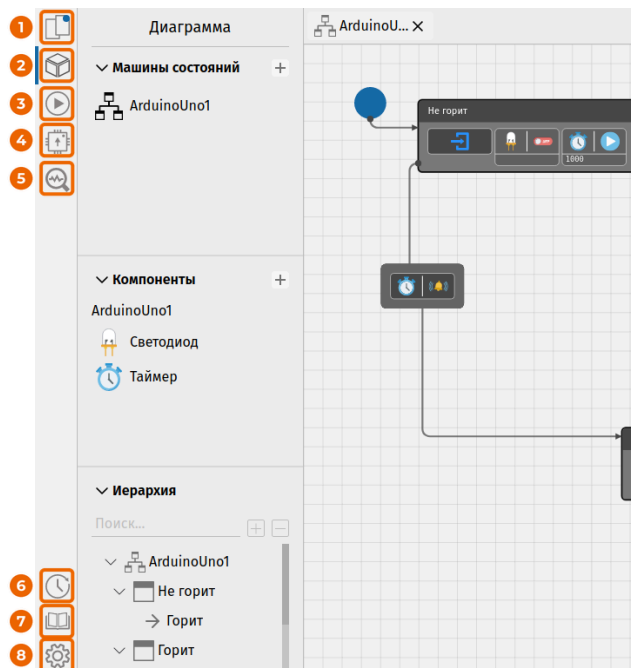


Рисунок В.23 – Значки левой панели

1. **Документ** – создать, открыть, сохранить.
2. **Диаграмма** – машины состояний, компоненты, иерархия состояний.
3. **Компилятор** – превращает диаграмму в прошивку для платы.
4. **Загрузчик** – записывает прошивку в плату через USB.
5. **Монитор порта** – показывает, что плата присылает по USB.
6. **История изменений** – что и когда менялось в документе.
7. **Документация** – справка (её же открывает F1).
8. **Настройки**.

Шаг 8. Компиляция и прошивка – когда есть плата. Открой «Компилятор» (3 на карте) и нажми большую синюю кнопку (1) – «Скомпилировать». Кружок рядом с заголовком (2) показывает связь с компилятором: если он красный, кнопка называется «Переподключиться» – нажми её и дождись, пока кружок станет зелёным.

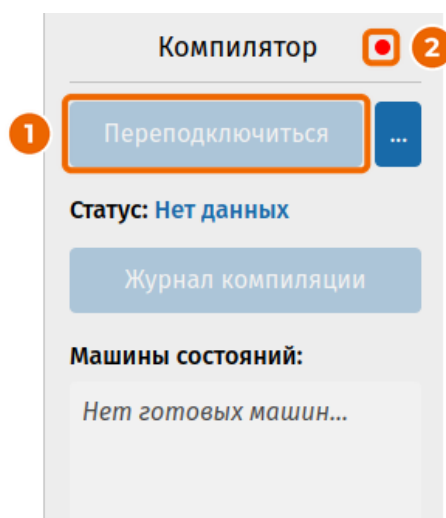


Рисунок В.24 – Панель «Компилятор»

Подключи Arduino к компьютеру кабелем и открой «Загрузчик» (4 на карте). Нажми «Подключить плату» (1), выбери свою плату в списке и нажми синюю кнопку с изображением платы – «Прошить». Прошивка готова, когда в «Журнале действий» появится подтверждение.

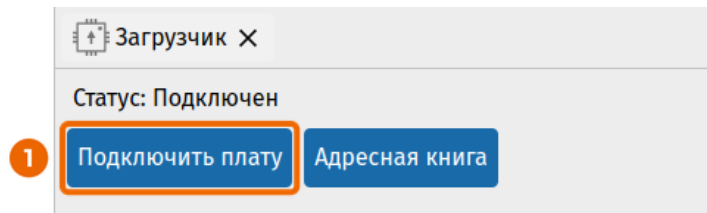


Рисунок В.25 – Панель «Загрузчик»

Платы под рукой нет – диаграмму всё равно можно проверить: пройди пальцем по стрелкам, как в разборах этой книги, и посмотри, в каком состоянии окажется программа после каждого события.

Wokwi: симулятор Arduino в браузере

Wokwi (wokwi.com) – бесплатный симулятор Arduino прямо в браузере. Ставить ничего не нужно, регистрироваться тоже: плату, светодиоды, кнопки и датчики «собирают» на экране проводами, а программу пишут тут же и запускают кнопкой. Ничего не сгорит, даже если ошибёшься. Если на твоём компьютере сайт не открывается, все схемы этой книги собираются и в Tinkercad. Интерфейс английский: названия кнопок ниже даны в кавычках по-английски, с переводом. Рамка с номером на картинке – куда нажимать. Зачем в схеме резистор – в главе об электричестве и электронике, что делают `setup()`, `loop()` и `Serial` – в главе про программирование Arduino. А здесь – как работать в самой программе и три задачи, чтобы руки привыкли.

Шаг 1. Открыть. Самый короткий путь – ссылка wokwi.com/projects/new/arduino-uno: сразу откроется чистый проект с платой Arduino Uno. То же самое кнопками: на главной странице wokwi.com выбери «Arduino (Uno, Mega, Nano)» (1 на левой половине рисунка), а на открывшейся странице в блоке «Start from Scratch» («Начать с нуля») – «Arduino Uno» (1 на правой половине).

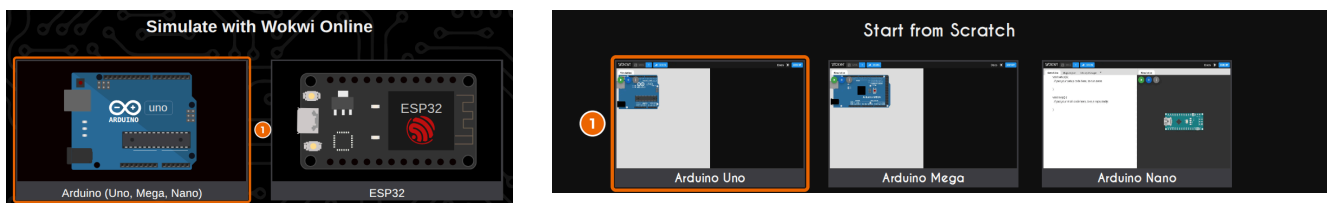


Рисунок В.26 – Как открыть чистый проект: слева – главная страница, справа – выбор платы

Входить в аккаунт не нужно: кнопки «Sign in» и «Sign up» (вход и регистрация) вверху справа не нажимай. Зато у работы без входа есть два неудобства. Первое: проект нигде не хранится. Обновишь страницу или закроешь вкладку – всё пропадёт, так что перед закрытием скачай проект (шаг 7). Второе: перед запуском Wokwi собирает программу на своём сервере. Обычно это доли секунды, иногда до минуты. Подожди, это нормально.

Шаг 2. Окно Wokwi. Слева – программа, справа – плата и схема.

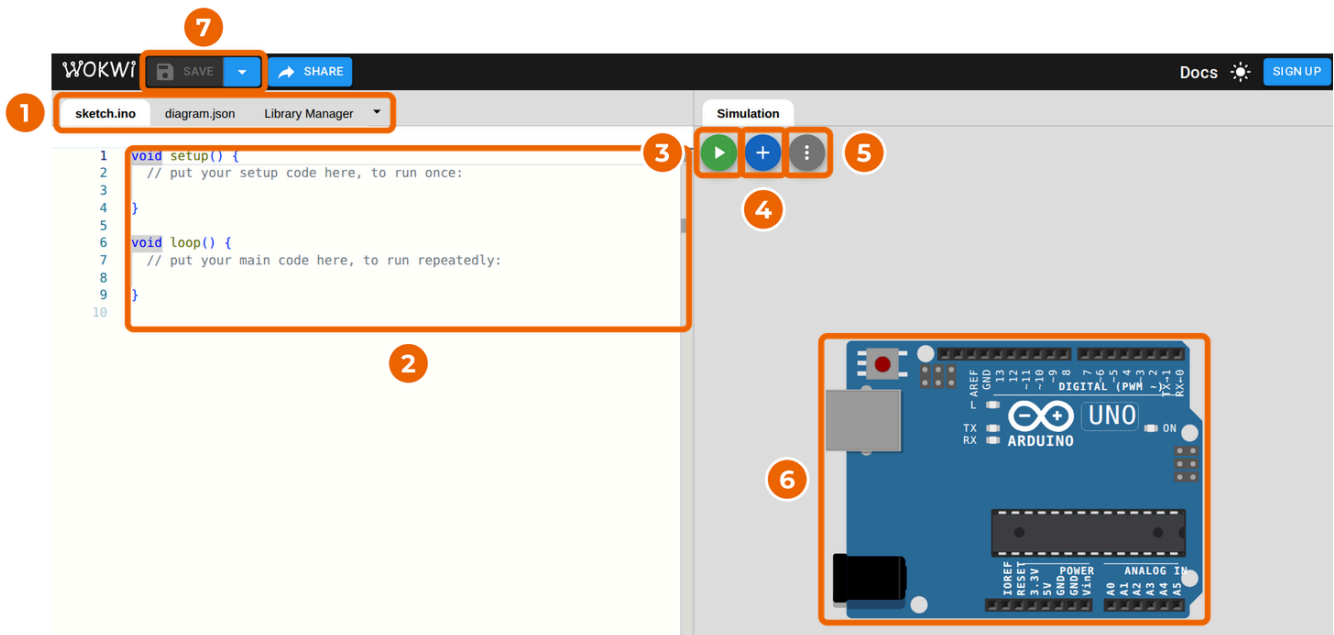


Рисунок В.27 – Окно Wokwi

1. **Вкладки.** «sketch.ino» – твоя программа. «diagram.json» – та же схема, записанная текстом: Wokwi пишет её сам, и в эту вкладку можно вставить готовую схему. «Library Manager» – подключаемые библиотеки, нам не понадобится.
2. **Окно программы.** Сразу лежат две обязательные функции, `setup()` и `loop()`.
3. **«Start the simulation»** («запустить симуляцию») – зелёная кнопка.
4. **«Add a new part»** («добавить деталь») – синяя кнопка «+».
5. **Меню «Zoom, grid, and more»** (три точки). «Undo» – отменить (Ctrl + Z), «Zoom in» и «Zoom out» – приблизить и отдалить, «Fit» – вернуть схеме размер по окну, «Toggle Grid» – показать сетку. Меню само не закрывается: закрой его клавишей Esc.
6. **Плата Arduino Uno.** Ножки подписаны: цифровые 0–13 – сверху, а питание (5V, GND – «земля») и аналоговые A0–A5 – снизу.
7. **«Save»** («сохранить») работает только со входом, поэтому нам нужна стрелка рядом с ним: в её меню лежит «Download project ZIP» (шаг 7).

Шаг 3. Добавить деталь. Нажми «+» (4 на карте окна). Откроется список деталей. В разделе «Basic» («основные») лежат LED – светодиод, Pushbutton и Pushbutton 6mm – кнопки (по-больше и поменьше), Resistor – резистор (2). Нужной детали на виду нет? Впиши её английское название в строку поиска (1), например `potentiometer`. Щёлкни по детали – она появится в левом верхнем углу поля. Нажми на неё и перетащи мышью на свободное место рядом с платой.

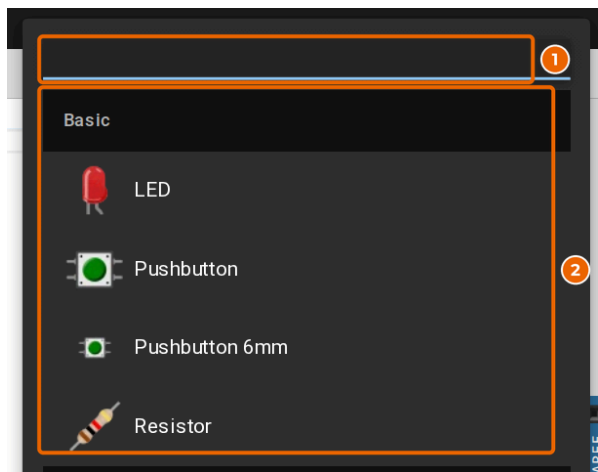


Рисунок В.28 – Список деталей: строка поиска (1) и раздел «Basic» (2)

Шаг 4. Настроить и повернуть. Щёлкни по детали – над полем появится её панель. У резистора в ней поле «Resistance» («сопротивление», 1), кнопка «Rotate» («повернуть», 2) и корзина «Delete» («удалить», 3). Впиши в поле 220 и нажми Enter: полоски на резисторе перецветятся. В поле число – это омы, а буква к – «кило»: 1k – это 1000 Ом.

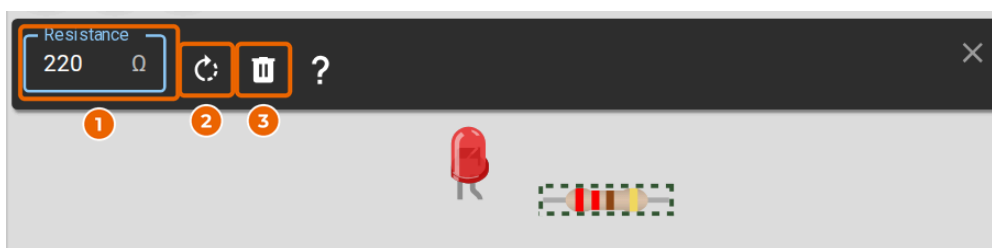


Рисунок В.29 – Панель резистора: «Resistance» (1), «Rotate» (2), «Delete» (3)

Деталь поворачивают кнопкой «Rotate» или клавишей R: одно нажатие – четверть оборота. Удалить деталь можно корзиной или клавишей Delete.

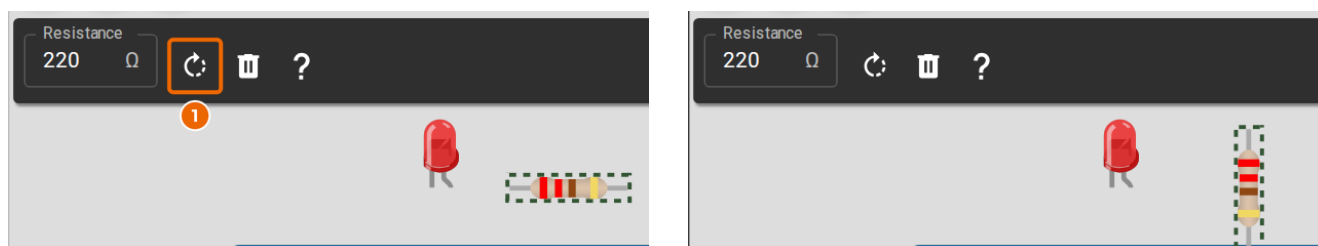


Рисунок В.30 – Поворот: слева резистор лежит, справа – после нажатия «Rotate» (1) он встал вертикально

Шаг 5. Провода. Щёлкни по ножке одной детали – от неё потянется провод. Щёлкни по ножке другой детали – провод закрепится. Передумал, пока тянешь, – нажми Esc, и провода не будет. Наведи мышь на ножку, и Wokwi её подпишет: led1:C – это деталь led1 и её ножка C, катод (у светодиода есть ещё led1:A – анод). Пока тянешь провод, цвет можно сменить цифрой: 2 – красный, 0 – чёрный, 5 – зелёный (как цвета полосок на резисторах). Лишний провод убирают так: щёлкни по нему и нажми Delete.

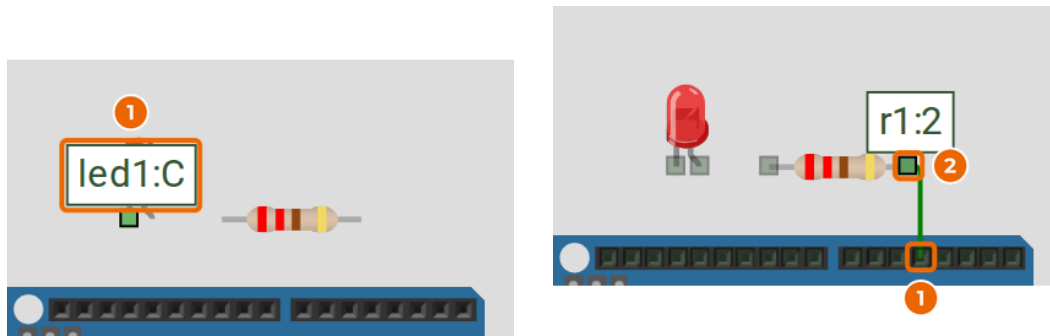


Рисунок В.31 – Слева – подпись ножки при наведении (1), справа – провод тянется от ножки 4 (1) к ножке r1:2 резистора (2)

Шаг 6. Запуск и монитор порта. Нажми «Start the simulation». Пока Wokwi собирает программу, подожди, потом схема оживёт: светодиоды горят, кнопки нажимаются мышью. На месте зелёной кнопки появятся три: «Restart the simulation» (начать заново), «Stop the simulation» (остановить, 1) и «Pause» (пауза). Кнопки «+» и меню с тремя точками при этом спрятаны: чтобы поправить схему, сначала нажми «Stop». Вверху справа бежит время симуляции (2). Если в программе есть `Serial.println(...)`, внизу откроется **монитор порта** (3). Это окно, куда плата «печатает» числа и слова. В нижнем ряду у него кнопка графика («Serial plotter»), корзина (очистить окно) и стрелка (свернуть монитор).

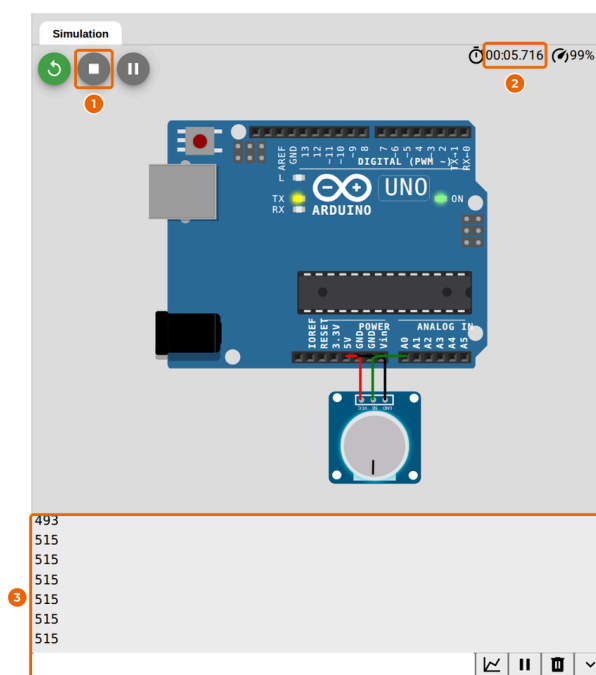


Рисунок В.32 – Идёт симуляция: «Stop the simulation» (1), время (2), монитор порта (3)

Шаг 7. Сохранить. Без входа работу сохраняют архивом: нажми стрелку рядом с «Save» (1) и выбери «Download project ZIP» (2). Скачается `project.zip`: в нём `sketch.ino` – программа, `diagram.json` – схема и ещё `wokwi-project.txt` – ссылка на сайт. Чтобы вернуться к работе, открой новый проект, во вкладку `sketch.ino` вставь текст из `sketch.ino`, а во вкладку `diagram.json` – текст из `diagram.json`.

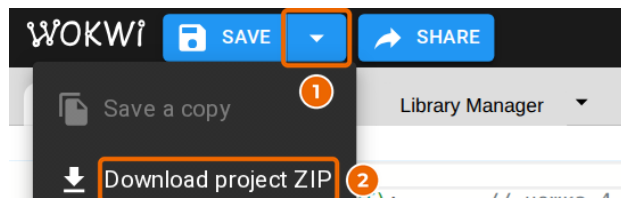


Рисунок В.33 – Меню рядом с «Save»

Задача П.1. Мигалка. Собери схему: ножка 4 → резистор 220 Ом → анод светодиода, катод светодиода → GND. На рисунке провод от ножки 4 (1) идёт к одному концу резистора (2), от другого его конца – к аноду светодиода (3), а от катода (левая ножка) – на GND (4) рядом с ножкой 13. Резистор ограничивает ток: в настоящей схеме без него светодиод бы сгорел.

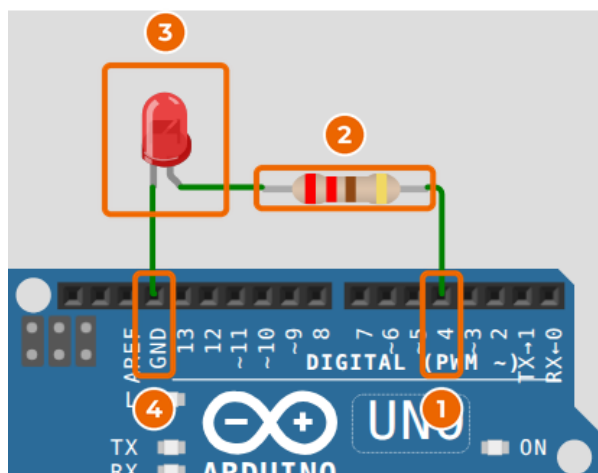


Рисунок В.34 – Схема мигалки

Вставь в окно программы текст и нажми «Start the simulation».

```
void setup() {
  pinMode(4, OUTPUT);      // ножка 4 - выход
}

void loop() {
  digitalWrite(4, HIGH);   // светодиод включить
  delay(1000);             // ждать 1000 мс = 1 с
  digitalWrite(4, LOW);    // выключить
  delay(1000);            // ждать ещё 1 с
}
```

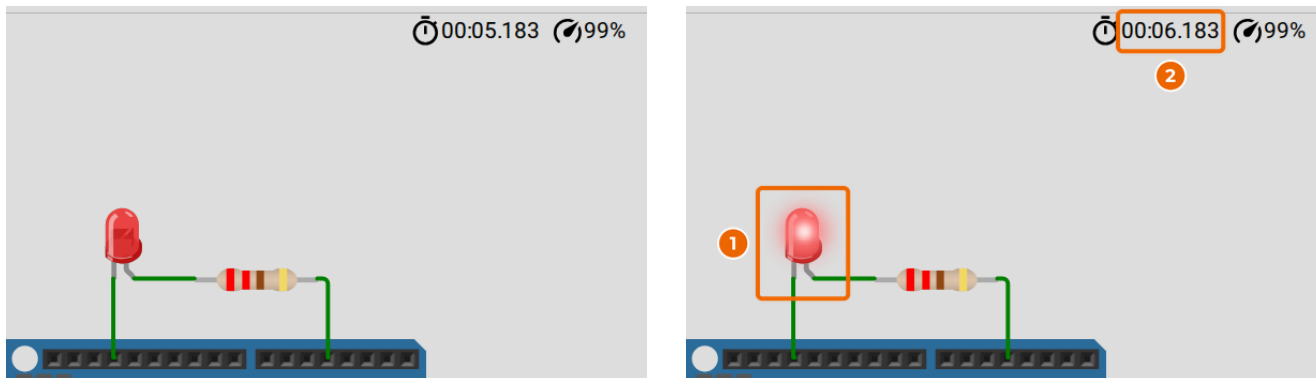


Рисунок В.35 – Слева светодиод не горит, справа горит (1); время симуляции – (2)

Что должно получиться: секунду светодиод горит, секунду нет. Теперь нажми «Stop», поменяй оба `delay(1000)` на `delay(250)` и запусти снова. Сколько раз вспыхнет светодиод за 10 секунд по времени симуляции? Ответ: 20. Одна вспышка – это 250 мс горит и 250 мс не горит, то есть 0,5 с, а $10 : 0,5 = 20$. Мы прогнали это в Wokwi: за 10 секунд ровно 20 вспышек. Светодиод не горит совсем? Проверь полярность: анод (правая ножка) должен смотреть в сторону резистора, а катод – на GND. Включённый наоборот светодиод не загорится, но и не сгорит.

Задача П.2. Кнопка. Открой новый проект (шаг 1): схема из П.1 больше не нужна. Лампочку возьмём ту, что уже стоит на плате: это светодиод L, он подключён к ножке 13. Добавь Pushbutton. У кнопки четыре вывода – два сверху и два снизу. Внутри кнопки два верхних вывода соединены всегда, два нижних тоже, а нажатие соединяет верхнюю пару с нижней. Поэтому провода берут так: один вывод верхней пары – к ножке 2, один вывод нижней пары – к GND. На рисунке это верхний правый вывод (1) и ножка 2 (3), нижний левый вывод (2) и GND (4).

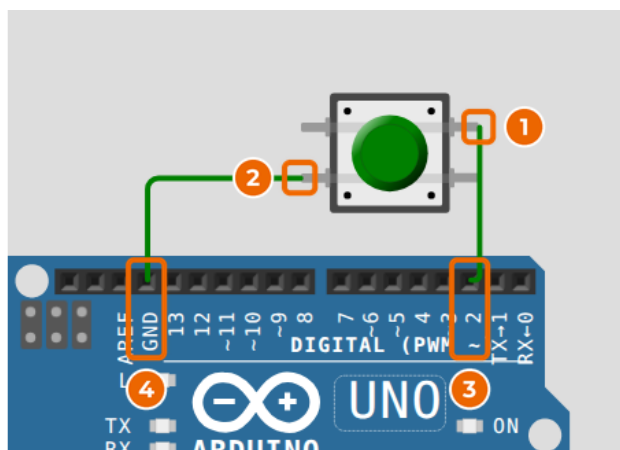


Рисунок В.36 – Схема с кнопкой

Программа. Здесь кнопка – вход с подтяжкой (`INPUT_PULLUP`): пока кнопка отпущена, на ножке 1.

```

void setup() {
  pinMode(2, INPUT_PULLUP); // кнопка: вход, подтянутый к 5 В
  pinMode(13, OUTPUT);      // светодиод L на плате
  Serial.begin(9600);
}

void loop() {
  int knopka = digitalRead(2);
  Serial.println(knopka);
  if (knopka == LOW) {      // нажата = LOW
    digitalWrite(13, HIGH);
  } else {
    digitalWrite(13, LOW);
  }
  delay(200);
}

```

Запусти, потом нажми на кнопку мышью и подержи, а затем отпусти.

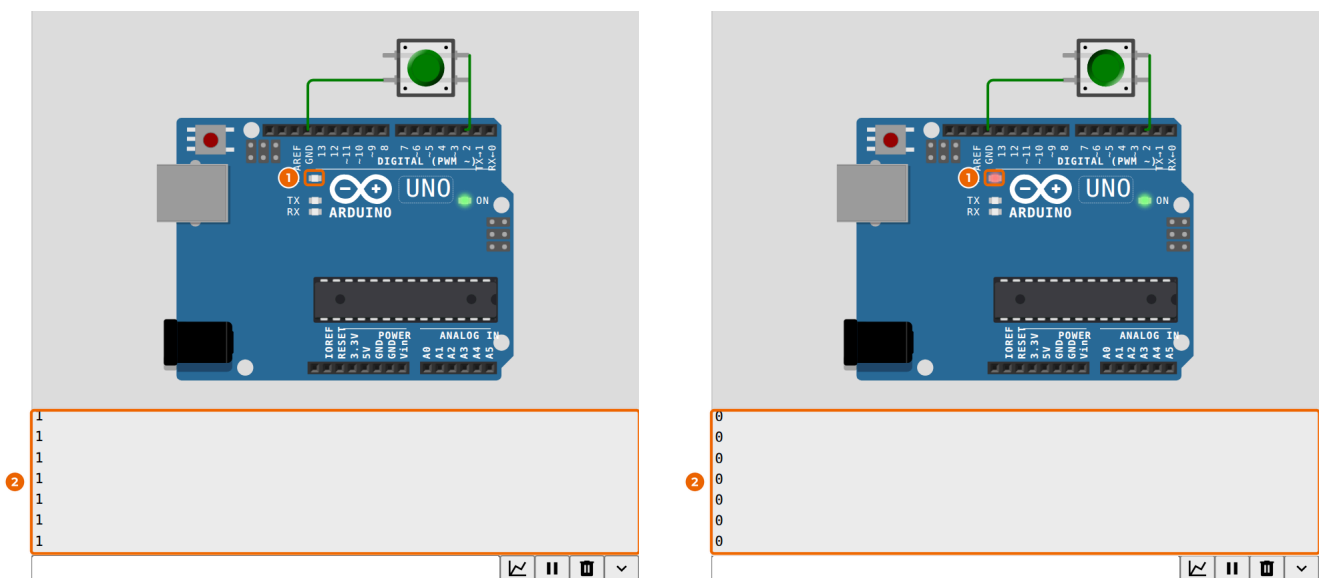


Рисунок В.37 – Слева кнопка отпущена, справа нажата: лампочка L (1) и монитор порта (2)

Что должно получиться: пока кнопка отпущена, в мониторе порта единицы, а L не горит. Нажал и держишь – пошли нули, и L загорается. Отпустил – снова единицы. Ответ: нажатой кнопке соответствует 0 (LOW), отпущенной – 1. Сплошные нули без нажатия? Значит, оба провода сели на выводы одной пары, например на два верхних: такая кнопка «нажата» всегда. Перенеси один провод на вывод другой пары.

Задача П.3. Потенциометр. Открой новый проект. Перетащи плату повыше: под ней нужно место. Добавь «Potentiometer» (строка поиска, potentiometer), поставь его под платой, щёлкни по нему и дважды нажми R: выводы окажутся сверху, ближе к плате. Они подписаны мелко: VCC, SIG, GND. Провода: красный – от VCC на 5V, чёрный – от GND на GND рядом с 5V, зелёный – от SIG на A0. На рисунке это выводы потенциометра (1), питание платы (2) и аналоговая ножка A0 (3).

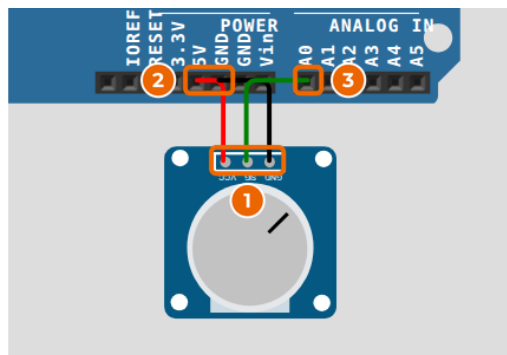


Рисунок В.38 – Схема с потенциометром

Программа печатает в монитор порта число с ножки A0.

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  int n = analogRead(A0);    // число от 0 до 1023
  Serial.println(n);
  delay(300);
}
```

Запусти, нажми мышью на круглую ручку и води по кругу: ручка идёт за курсором, а числа в мониторе меняются сразу. Между двумя упорами у ручки есть пустой промежуток. Если провести через него, число перескочит с 1023 на 0 или обратно.

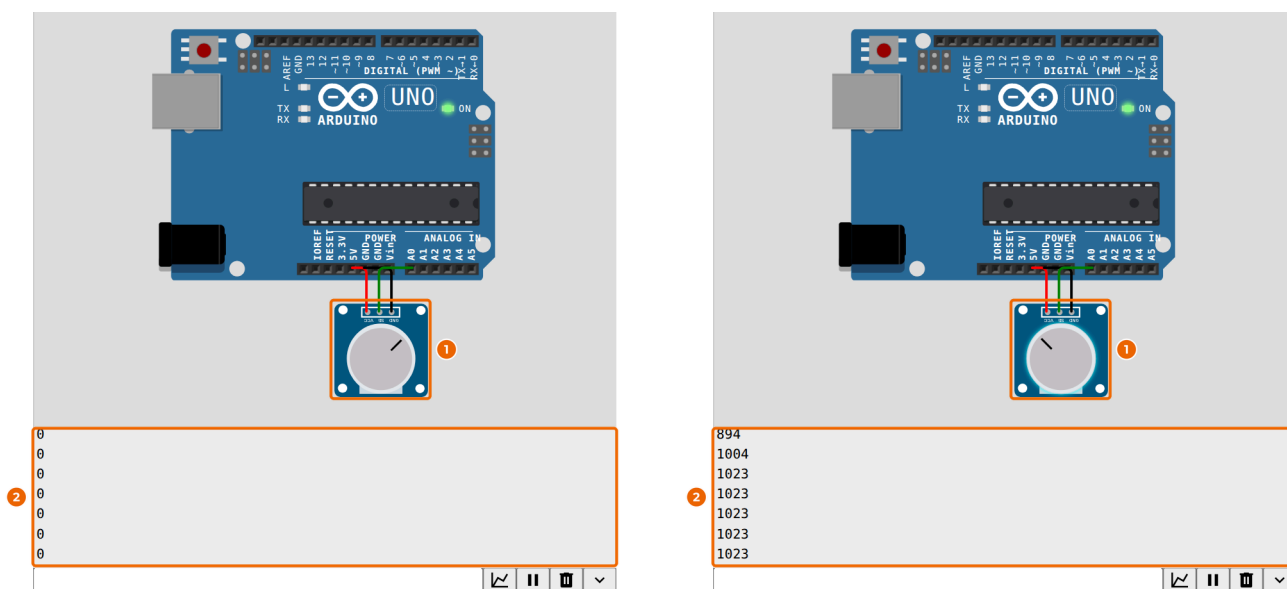


Рисунок В.39 – Слева ручка в одном крайнем положении, справа – в другом: ручка (1) и монитор порта (2)

Что должно получиться: в одном крайнем положении ручки в мониторе 0, в другом – 1023, а посередине – около 500. Сразу после добавления ручка стоит на нуле, поэтому первые числа в мониторе – нули. Ответ: 0 и 1023. Это границы `analogRead`: 0 соответствует 0 В, а 1023 – 5 В. Вопрос на подумать: какое напряжение на A0, когда в мониторе 512? Ответ: около 2,5 В, потому что $512 \cdot 5 / 1023 \approx 2,5$.

План на месяц до отбора

Если начинаешь в конце сентября, вот спокойный план: примерно по часу в день.

Таблица В.3

Неделя: что читаем	Главное дело недели
1: главы 1–3	записаться на курс сферы
2: главы 4–5	собрать схемы в Tinkercad
3: главы 6–7	запустить программы из разборов
4: глава 8 и шпаргалка	пройти тест курса сферы, если ещё не прошёл (+5 баллов)

Подробнее по неделям:

- **Неделя 1.** Запишись на курс сферы и войди в класс «Агротех» в Tinkercad – просто чтобы убедиться, что всё открывается. Каждый день решай «Реши сам» к свежим разборам.
- **Неделя 2.** Собери в Tinkercad схемы из главы 5: светодиод с резистором, фоторезистор, термометр.
- **Неделя 3.** Повтори программы из разборов глав 6 и 7, посмотри показания датчиков в мониторе порта и на графике.
- **Неделя 4.** Заново реши задачи, где были ошибки, пролистай шпаргалку и пройди тест курса сферы.

С 1 октября у олимпиады по плану открывается демо-этап с пробными заданиями. Попробуй их, как только появятся: так заранее увидишь, как выглядят задачи на «Орбите» и куда вводить ответы, и не будешь тратить на это время отбора.

Отбор длится до 10 ноября, и попыток две. Если первая прошла неудачно, вернись к главам, где потерял баллы, и иди на вторую.

Родителям и наставникам

- **Помогайте разбираться, но не решайте за ребёнка.** На командном финале сразу видно, кто разобрался сам, а за кого решили. Лучше пусть ребёнок ошибётся на отборе и научится, чем попадёт в финал и растеряется там.
- **Садитесь рядом на первых задачах в симуляторе.** Детей останавливает скорее «не могу войти в класс», «кнопка моделирования не нажимается» или «куда делась моя схема», чем трудная задача. Две подсказки из вебинара сферы: общая учётная запись класса «Агротех» не сохраняет проекты (нужное стоит переписать в тетрадь или скопировать код в файл), а если после ошибки в программе моделирование не запускается, помогает обновить страницу.
- **Если дома есть настоящая плата Arduino,** напомните правила, которые разработчики сферы просят повторять даже при работе в симуляторе: схему меняют только при отключённом питании, мокрыми руками к ней не прикасаются, свои детали без спроса не

подключают. В симуляторе ничего не сгорит, а настоящий светодиод без резистора сгорит за секунду.

- **Всё про регистрацию, бонусы и финал** для Новосибирской области – на nto.kvantorium54.ru (вкладки «Я наставник» и «Я участник или родитель»).
- **Региональный координатор НТО Юниоры в Новосибирской области** – детский технопарк «Кванториум», info@kvantorium54.ru.

Честно об этой книге

Главное, что нужно знать про эту книгу: заданий отбора 2026/27 мы не видели – организаторы их заранее не показывают. Угадывать их бесполезно: наставники, которые следят за олимпиадой не первый год, говорят, что задания из года в год не повторяются. Поэтому книга учит темам из программы сферы и приёмам решения, а не готовит к конкретным задачам. Когда организаторы выложат пробные задания, мы сверим с ними книгу и выпустим новую версию.

Это не официальное издание НТО. Книгу составил методист «Кванториума» по открытым материалам олимпиады: программе сферы «Агротех. Основы инженерных агросистем» 2026/27, записям вебинаров сферы для наставников и участников, официальным требованиям к площадкам финала.

Сфера «Агротех» появилась только в сезоне 2026/27, поэтому своих задач прошлых лет у неё нет. Ближе всего к ней по темам сфера «Технологии и среда обитания» прошлых сезонов: там тоже были метеостанция, умная теплица и Arduino. Но тексты её задач в открытом доступе закрыты – видны только названия вроде «Графики спасают урожай» или «Как полить сити-ферму и сэкономить?». Эти темы мы использовали как подсказку, о чём спрашивают, а сами задачи придумали заново по программе сферы. Поэтому большинство задач в книге – свои.

Книга написана до демо-этапа с пробными заданиями. Когда он откроется, мы сверим книгу с настоящим форматом задач и обновим её. Кнопки Tinkercad и Кибериады описаны такими, какими их показывали на вебинарах в сентябре 2026 года: если интерфейс изменится, смысл останется тем же.

Каждое число в ответах мы пересчитали на компьютере, а каждую программу запустили. Если ты всё-таки нашёл ошибку или что-то осталось непонятным, напиши автору: osinovsky.aa@kvantorium54.ru. Следующие версии книги станут лучше благодаря тебе.

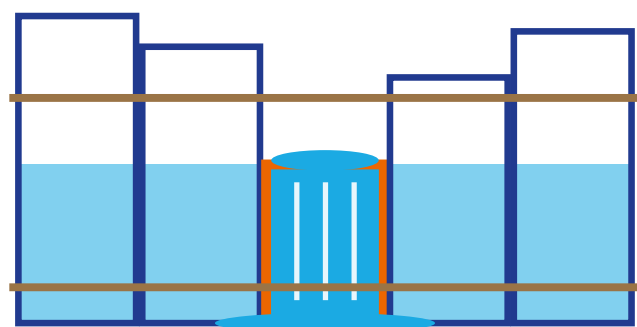
Растение как живая система



Умная теплица начинается с растения. Датчик покажет «+31 °C», но хорошо это или плохо, скажет только тот, кто знает, чего хочет помидор. Здесь – из чего растение строит себя, когда оно ест и когда дышит, зачем ему корни и почему к утру в закрытой теплице копится углекислый газ. В задачах по этой теме придётся выбирать верные утверждения, подбирать причину к признаку и делать выводы из описания опыта. Счёт тоже будет: проценты, часы и минуты, пропорции.

1.1. Что нужно растению

НА ПАЛЬЦАХ



тепло вода **свет** азот CO₂

урожай – как вода:
выше самой короткой
доски не поднимется
короткая доска – свет:
через неё и уходит вода

Рисунок 1.1 – Бочка из досок разной высоты. Вода вытекает через самую короткую

Каждая доска – одно из условий жизни растения. Сколько ни наращивай длинные доски, воды больше не станет, пока не надставишь короткую.

Семечко огурца весит меньше грамма, а за лето из него вырастает плеть длиной в пару метров с десятком огурцов. Строительный материал для этого растение берёт из окружающего мира. Условия, от которых зависит, как оно растёт, называют **факторами роста**. Главных пять:

- **свет** – источник энергии, без него растение не может сделать себе еду;
- **тепло** – при холоде жизнь внутри растения идёт медленно, при жаре оно страдает;
- **вода** – из неё травянистое растение на 80–95 % и состоит, она же несёт по стеблю питательные вещества;
- **воздух** – листьям нужен углекислый газ, а корням и всем живым клеткам – кислород;
- **минеральное питание** – соли азота, фосфора, калия и других веществ, которые корни берут из почвы.

Ни один фактор не заменяет другой. Нельзя «добрать» свет лишним поливом, а азот – теплом. В XIX веке немецкий химик Юстус Либих прославил простое правило: урожай зависит от того условия, которого растению не хватает сильнее всего. Его называют **законом минимума**, а рисуют его бочкой из досок разной высоты, как на картинке выше. Вода в ней стоит вровень с самой короткой доской.

Зимой в теплице под Новосибирском самая короткая доска почти всегда – свет: день длится около семи часов, а солнце висит низко. Летом в той же теплице в солнечный полдень может быть за +40 °С, и короткой доской становится тепло, вернее, его избыток. Поэтому агроном в разные месяцы борется с разными бедами.

На лунной базе растению не достанется ничего бесплатно: свет дадут лампы, тепло – обогреватель, воду и углекислый газ – система жизнеобеспечения. Каждую доску бочки там строит инженер. Кстати, теплица для космической базы – сквозная тема финала Агротеха в этом сезоне (так говорили разработчики сферы на вебинаре).

Сколько нужно. Точные нормы зависят от сорта и от возраста растения, поэтому ниже – ориентиры, примерные для многих сортов. По ним агрономы настраивают тепличную автоматику.

Таблица 1.1

Культура	Днём, °С	Ночью, °С	Влажность воздуха, %
Томат	20–25	16–18	60–70
Огурец	24–28	18–20	80–90
Листовой салат	16–20	10–15	около 70

Огурец любит тепло и сырость: его родина – влажные леса у подножия Гималаев в Индии. Томату сырость вредна, во влажном воздухе его листья быстро поражает грибок. Салат – культура прохладная: в жару он пускает цветочную стрелку, и листья горчат. А выше +30–32 °С у томата хуже развивается пыльца: завязей становится меньше, а при +35 °С и выше цветы опадают, так и не дав плодов. Освещённость для томата в тесте курса – 20 000–50 000 лк.

Запомни. Факторов роста пять, и ни один не заменяет другой. Урожай держит на привязи тот, которого не хватает сильнее всего, – это закон минимума. У каждой культуры свои нормы: огурцу подавай тепло и сырость, томату – посуше, салату – прохладу.

1.2. Фотосинтез: еда из воздуха и света

НА ПАЛЬЦАХ

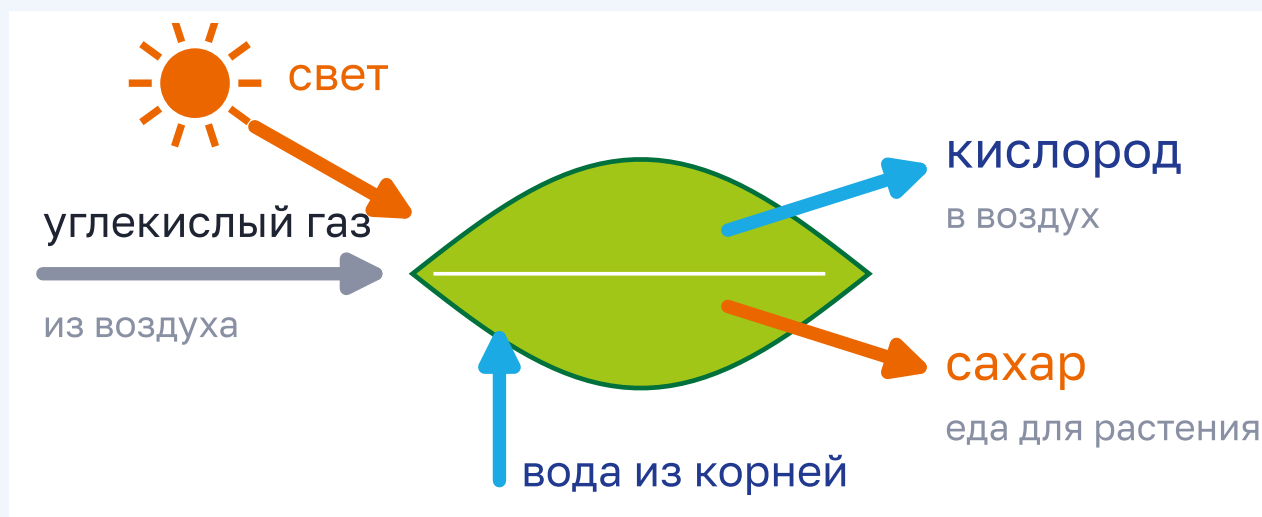


Рисунок 1.2 – Лист как кухня: на входе свет, углекислый газ и вода, на выходе сахар и кислород

Лист – кухня, которая работает на солнечной энергии. Продукты – газ из воздуха и вода из корней, блюдо – сахар.

Растение не ест готовую еду, как мы. Оно делает её само, и делает в листьях. Этот процесс называют **фотосинтезом**: «фото» – свет, «синтез» – соединение, сборка. Словами: **на свету из углекислого газа и воды лист делает сахар и выделяет кислород**. Коротко это записывают так:



Работает эта кухня благодаря **хлорофиллу** – зелёному веществу в клетках листа. Хлорофилл ловит свет и тратит его энергию на сборку сахара. Из сахара растение потом делает всё остальное: крахмал про запас, клетчатку для стенок клеток, а с помощью солей из почвы – белки.

Откуда тогда берётся вес растения? Не из земли, хотя так думали сотни лет. В XVII веке учёный из Фландрии (сейчас это Бельгия) Ян ван Гельмонт насыпал в горшок около 90 кг сухой земли и посадил ветку ивы весом чуть больше 2 кг. Пять лет он поливал её только дождевой водой. Ива потяжелела примерно до 77 кг, а земля за это время полегчала граммов на шестьдесят. Сам ван Гельмонт решил, что ива выросла из воды. Сейчас мы знаем: почти вся сухая масса растения собрана из углекислого газа, который лист вытянул из воздуха.

Почему листья зелёные. Солнечный свет только кажется белым – в нём смешаны все цвета радуги. Хлорофилл сильнее всего поглощает **красный и синий** свет, а зелёный почти не берёт и отражает. Этот отражённый зелёный свет и попадает нам в глаза. Отсюда розово-фиолето-

вые фитолампы в теплицах: в их свете много красного и синего, того, что растению нужное всего.

Сколько света. Освещённость измеряют в **люксах** (лк). Для сравнения (по справочной таблице Википедии):

Таблица 1.2

Где	Освещённость, лк
Прямое солнце	32 000–100 000
Ясный день в тени	10 000–25 000
Пасмурный день	около 1000
Рабочее место в офисе	320–500
Восход или закат в ясную погоду	около 400

Глаз этой разницы почти не замечает: в комнате нам светло, а растению – сумерки. Солнечный полдень ярче офисной лампы в сотни раз.

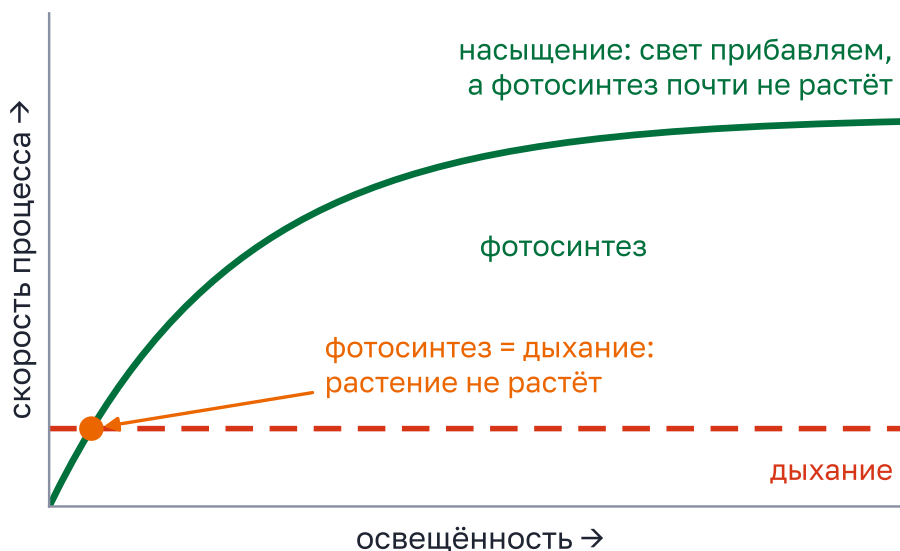


Рисунок 1.3 – Как скорость фотосинтеза зависит от освещённости. Красная линия – дыхание, о нём в следующем разделе

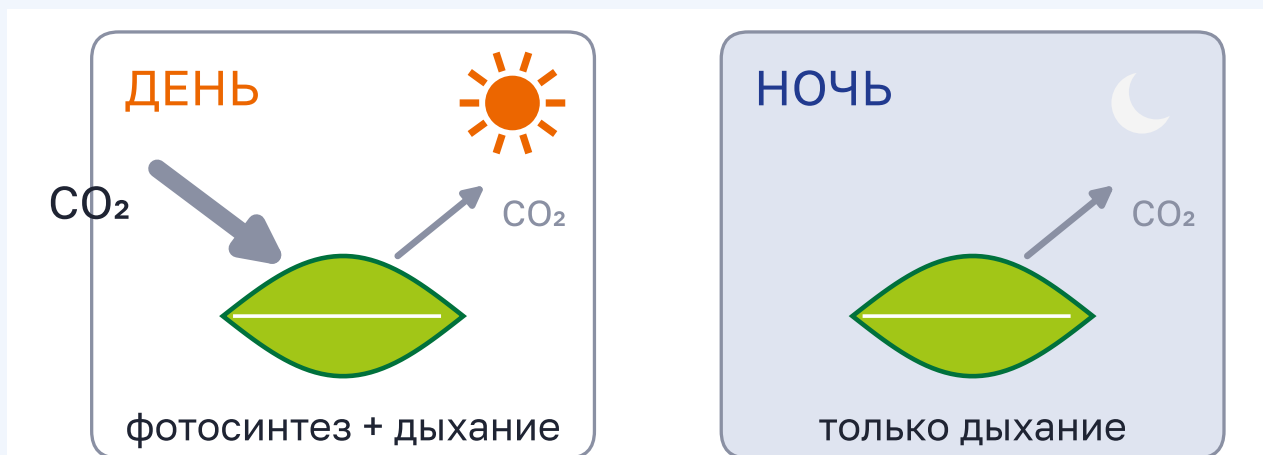
Чем больше света, тем быстрее фотосинтез, – но только до поры. На графике видно: сначала линия круто идёт вверх, а потом становится почти горизонтальной. Это **насыщение**: листу уже хватает света, и тормозит его что-то другое, например нехватка углекислого газа или холод. Опять закон минимума. А при очень слабом свете, левее оранжевой точки, растение тратит на дыхание больше сахара, чем успевает сделать. Такое растение не растёт, а худеет.

Запомни. Фотосинтез идёт в листьях на свету: углекислый газ + вода → сахар + кислород. Хлорофилл берёт в основном красный и синий свет, зелёный отражает. После насыщения лишний свет почти ничего не прибавляет.

Совет. В задачах про фотосинтез держи в голове «кухню» с картинки: что входит (свет, CO_2 , вода) и что выходит (сахар, кислород). Утверждение, где газы перепутаны местами, сразу отбрасывай.

1.3. Дыхание: растение тоже тратит

НА ПАЛЬЦАХ



дыхание не выключается никогда: CO_2 выходит и днём, и ночью

Рисунок 1.4 – Днём и ночью. Днём лист забирает много углекислого газа и немного отдаёт; ночью только отдаёт

Фотосинтез похож на зарплату, дыхание – на расходы. Зарплата приходит только днём, а тратить приходится круглые сутки.

Сахар, сделанный листом, – это запас энергии. Чтобы им воспользоваться, растение его «сжигает» с помощью кислорода, как мы сжигаем еду. Этот процесс называют **дыханием**. Словами: **сахар и кислород превращаются в углекислый газ и воду, а освобождённая энергия идёт на жизнь и рост**. Выходит фотосинтез наоборот:

$\text{сахар} + \text{кислород} \rightarrow \text{углекислый газ} + \text{вода} + \text{энергия}$.

Дышат все живые клетки растения – листья, стебель, корни, цветы, плоды – и дышат **всегда**, днём и ночью. Частая ошибка школьников – думать, что растения днём делают кислород, а ночью дышат. Днём они делают и то и другое, просто фотосинтез в светлое время идёт намного быстрее дыхания. Поэтому днём лист в итоге поглощает углекислый газ, а ночью, когда фотосинтеза нет, только выделяет его.

Проверить, что растения дышат, можно без приборов. Насыпь в термос горсть проросших семян фасоли, а в другой – такую же горсть семян, заранее прокипячённых (живых клеток в них уже нет). Через сутки в первом термосе будет заметно теплее: при дыхании выделяется тепло. А если в воздух из первого термоса опустить горящую лучинку, она погаснет – кислород там израсходован, вместо него углекислый газ.

Зачем ночью прохладнее. Чем теплее, тем быстрее дыхание. Тёплой ночью растение сжигает много сахара впустую – днём заработанное уходит в трубу. Поэтому в теплице ночью держат на несколько градусов ниже, чем днём (сравни столбцы таблицы в разделе 1.1).

Углекислый газ в теплице. В уличном воздухе углекислого газа мало – около 0,04 %. Такие маленькие доли удобнее считать в **ppm** – «частях на миллион»: 1 ppm – одна частица из миллиона. Сейчас на улице около 420–430 ppm. Если теплицу закрыть на ночь, растения надышат углекислого газа, и к утру его станет больше, чем снаружи. Утром включается фотосинтез – и к обеду газа в закрытой теплице может остаться меньше, чем на улице. Растениям его уже не хватает, поэтому теплицу проветривают, а в больших тепличных комбинатах углекислый газ даже подают специально – до 800–1000 ppm.

Запомни. Дыхание: сахар + кислород → углекислый газ + вода + энергия. Растение дышит всегда. Днём фотосинтез сильнее дыхания, ночью фотосинтеза нет. В закрытой теплице CO₂ растёт ночью и падает днём.

1.4. Корни: вода и минеральное питание

НА ПАЛЬЦАХ

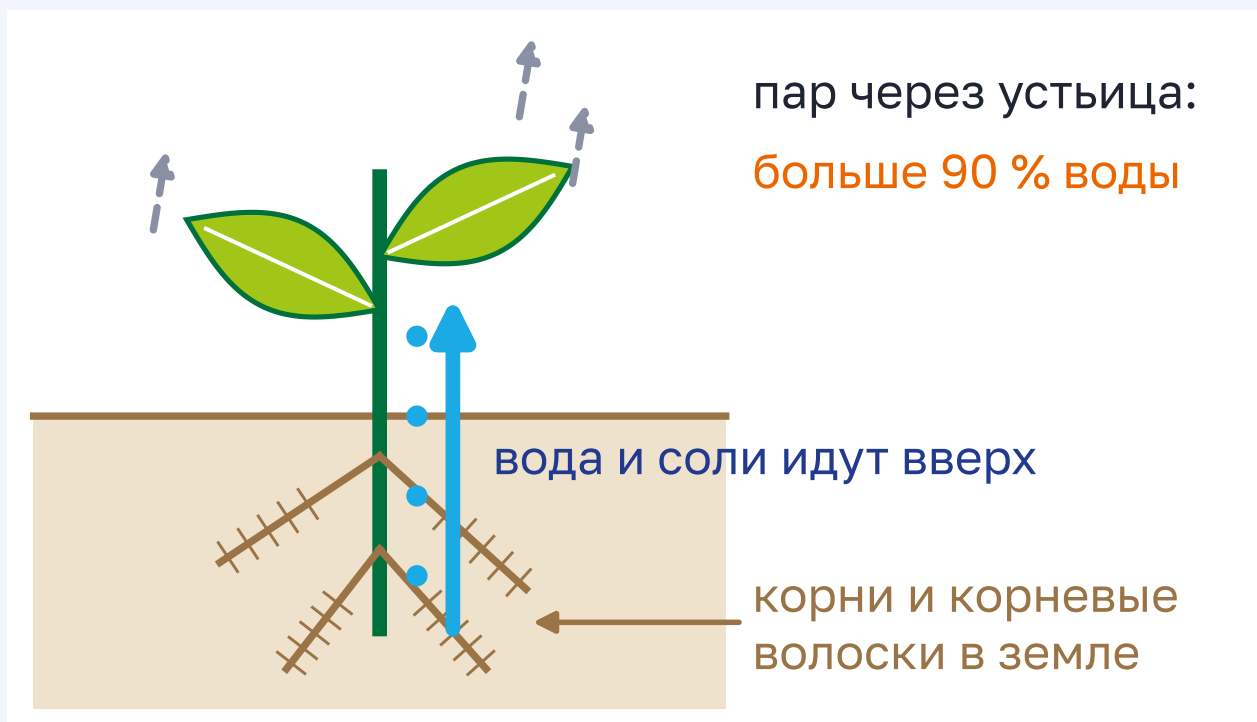


Рисунок 1.5 – Путь воды: из земли в корень, по стеблю вверх и паром из листьев

Растение работает как водопровод. Корни забирают воду из земли, стебель поднимает её вверх, а листья выпускают почти всю наружу паром.

У корня две работы: держать растение в земле и всасывать воду. Всасывает он кончиками, на которых растут тысячи **корневых волосков** – тоненьких выростов тоньше волоса. Они сильно

увеличивают поверхность корня, как ворс на махровом полотенце: такое полотенце впитывает больше гладкого.

Вместе с водой в корень входят растворённые в ней **минеральные соли**. Главных «элементов питания» три, их пишут на каждом пакете удобрения буквами N, P, K:

- **азот** (N) – для роста листьев и стеблей; нехватку выдают бледные, желтеющие старые листья;
- **фосфор** (P) – для корней, цветения, созревания семян;
- **калий** (K) – для плодов и устойчивости к болезням и холоду.

Почти всю воду, которую корни подняли, растение отдаёт в воздух – больше 90 %. Испаряется она через **устьица** – крошечные «форточки», которых больше всего на нижней стороне листа. Испарение воды растением называют **транспирацией**. Зачем такая расточительность? Испаряясь, вода охлаждает лист, как пот охлаждает нас в жару. А ещё она тянет за собой снизу новую воду с солями.

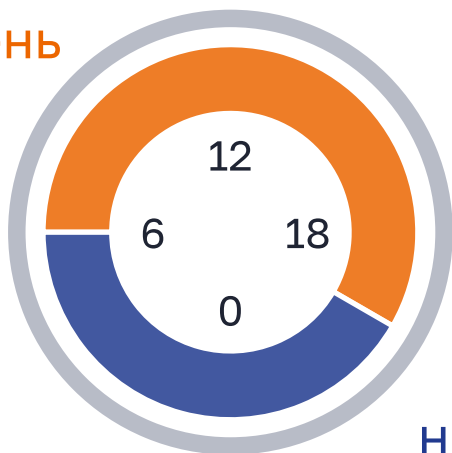
Корни тоже дышат. Им нужен кислород из воздуха, который есть между комочками почвы. Если землю в горшке постоянно заливать, все промежутки занимает вода, корни задыхаются и начинают гнить. Растение при этом вянет, хотя воды вокруг полно. Поэтому датчик влажности почвы в умной теплице следит за обеими границами: и «слишком сухо», и «слишком мокро».

Запомни. Корни всасывают воду с минеральными солями через корневые волоски. N – листья, P – корни и цветы, K – плоды. Больше 90 % воды испаряется через устьица. Корням нужен кислород, поэтому перелив опасен так же, как засуха.

1.5. Суточные ритмы

НА ПАЛЬЦАХ

день



днём: фотосинтез,
устьица открыты

ночью: устьица закрыты

серое кольцо – дыхание,
оно идёт круглые сутки

ночь

Рисунок 1.6 – Сутки по кругу: днём фотосинтез и открытые устьица, ночью устьица закрыты, дыхание – всё время

У растения нет будильника, но режим дня у него есть. Его задают восход и закат.

Всё, что в растении повторяется раз в сутки, называют **суточными (биологическими) ритмами**. Примеры ты видел сам: одуванчик к вечеру закрывает соцветие, а утром раскрывает. У большинства растений на свету открываются устьица – лист «вдыхает» углекислый газ для фотосинтеза. В темноте они закрываются, чтобы зря не терять воду. В сильную жару и засуху устьица тоже закрываются, даже днём, – и фотосинтез тормозит.

Растения замечают и **длину дня** – сколько часов в сутках светло. По ней они «узнают» время года. Редис, например, при длинном летнем дне уходит в стрелку и цветёт вместо того, чтобы наливать корнеплод. Поэтому его сеют весной или в конце лета.

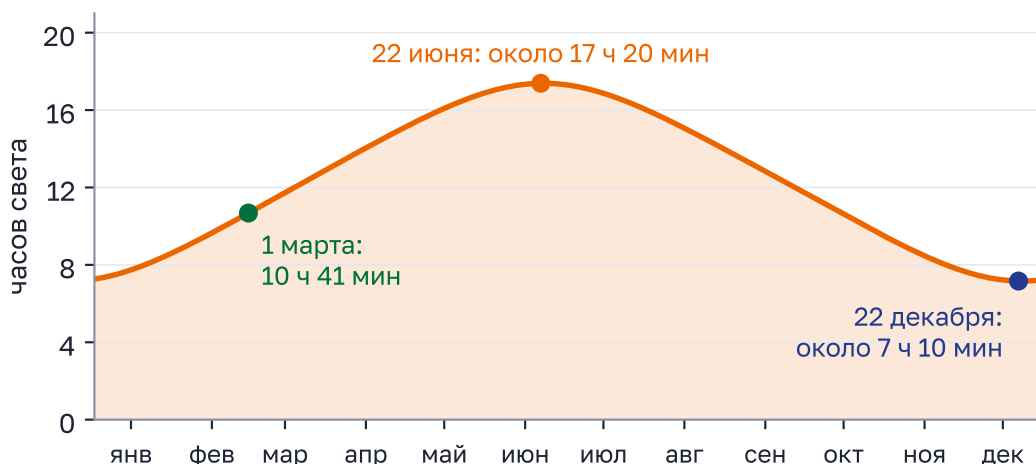


Рисунок 1.7 – Сколько часов длится светлое время суток в Новосибирске в течение года (расчёт по формулам положения Солнца)

В Новосибирске летом день длиннее семнадцати часов, а в декабре – чуть больше семи. Рассаду сеют в конце зимы, когда света мало, поэтому её **досвечивают** – включают лампы утром

до рассвета и вечером после заката. Так делают на подоконниках и в больших теплицах. Сколько часов должны гореть лампы – задача на временные интервалы, мы её решим в задаче 1.5.

Как считать длину дня. Из времени заката вычитаем время восхода. Часы и минуты вычитаем отдельно, а если минут не хватает – занимаем один час, это 60 минут. Пример: восход в 8:21, закат в 19:02. Минуты: 2 – 21 не получается, занимаем час: 62 – 21 = 41 мин, часов стало 18 – 8 = 10. Длина дня – 10 ч 41 мин.

Запомни. Устьица у большинства растений открыты на свету и закрыты в темноте. Длина дня подсказывает растению время года. Длина дня = закат – восход, в часе 60 минут, а не 100.

Совет. В задачах на время переводим всё в минуты: 8:21 – это $8 \cdot 60 + 21 = 501$ мин. Дальше обычное вычитание, а в конце переведем обратно в часы и минуты.

1.6. Микроклимат: что мерить и зачем

НА ПАЛЬЦАХ

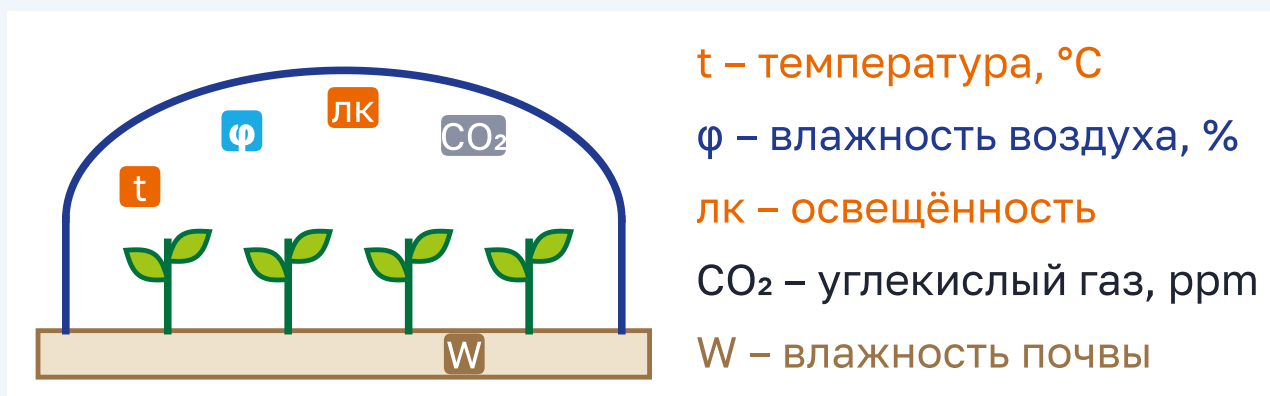


Рисунок 1.8 – Теплица в разрезе и её датчики

У теплицы своя погода, и хозяин не может сидеть в ней круглые сутки. Вместо него дежурят пять датчиков, у каждого своя работа.

Микроклимат – погода внутри небольшого замкнутого места: теплицы, комнаты, аквариума. Для растения главное в нём – температура воздуха, влажность воздуха, освещённость, количество углекислого газа и влажность почвы. Каждый из этих параметров меряет свой датчик, а все вместе – сырьё для автоматики, о которой пойдёт речь в следующих главах.

Таблица 1.3

Что мерим	Датчик	Единица
Температура воздуха	термометр (датчик температуры)	°C
Влажность воздуха	датчик влажности воздуха (например, DHT11)	%

Что мерим	Датчик	Единица
Освещённость	фоторезистор или датчик освещённости	лк
Углекислый газ	датчик CO ₂	ppm
Влажность почвы	датчик влажности почвы	условные единицы или %

Зачем мерить, если можно зайти и посмотреть? Затем, что человек ошибается и спит. Теплица в солнечный майский день нагревается за час-два: стекло и плёнка пропускают солнечный свет внутрь, а тёплый воздух наружу не выпускают. Форточку надо открыть вовремя, а не когда хозяин вернётся с работы. Ночью, наоборот, может ударить заморозок.

Запомни. Микроклимат – погода внутри теплицы. У каждой его величины свой датчик и своя единица: градусы, проценты, люксы, ppm. Автоматике нужны числа, «на глаз» она не умеет.

1.7. Влажность воздуха и роса

НА ПАЛЬЦАХ

тёплый воздух, +22 °C



налит на 70 %

ОСТЫЛ

за ночь

холодный воздух, +12 °C



лишнее – роса

Рисунок 1.9 – Воздух – стакан для пара. Тёплый воздух – большой стакан, холодный – маленький

Пара в воздухе столько же, а стакан за ночь стал меньше. Что не влезло, выливается на стёкла и листья каплями росы.

Влажность воздуха – самый хитрый из пяти параметров. В воздухе всегда есть невидимый водяной пар. Но тёплый воздух может удержать пара намного больше, чем холодный: в кубометре воздуха при +22 °C помещается до 19 г пара, а при +12 °C – только около 11 г. **Относительная влажность** показывает, на сколько процентов «стакан» заполнен паром от того предела, который возможен при этой температуре:

$$\text{влажность} = \frac{\text{сколько пара есть}}{\text{сколько пара может быть при этой температуре}} \cdot 100 \%$$

Посчитаем на числах. В комнате +20 °С, в каждом кубометре воздуха 8,7 г пара, а поместиться может 17,3 г. Влажность: $8,7 : 17,3 \cdot 100 \approx 50 \%$, стакан налит наполовину. Ночью комната остыла до +10 °С, и предел упал до 9,4 г. Пара всё те же 8,7 г, а влажность уже $8,7 : 9,4 \cdot 100 \approx 93 \%$. Воды в воздухе не прибавилось ни капли – просто стакан стал меньше.

Если тёплый влажный воздух остывает, предел опускается. Когда пара становится больше предела, лишний пар превращается в капли – выпадает **роса**. Ты видел это на холодной бутылке из холодильника и на траве летним утром. В теплице ночью роса садится на стёкла и листья, а мокрые листья – лучшее место для грибковых болезней. Температуру, при которой начинает выпадать роса, называют **точкой росы**.

Запомни. Относительная влажность – сколько пара есть, в процентах от того, сколько может поместиться. Остывая, воздух «сжимает стакан»: проценты растут, хотя пара не прибавилось. Остыл ниже точки росы – выпадает роса.

1.8. Разбираем задачи

Задача 1.1 Кухня и расходы

разминка · фотосинтез, дыхание, выбор верных утверждений

Выбери все верные утверждения. Ответ – номера верных утверждений по возрастанию, без пробелов и запятых.

1. Для фотосинтеза растению нужен свет.
2. Растения дышат только ночью.
3. При фотосинтезе растение поглощает углекислый газ и выделяет кислород.
4. Лист зелёный, потому что хлорофилл лучше всего поглощает зелёный свет.
5. При дыхании растение расходует сахар и выделяет углекислый газ.
6. Большую часть своей массы растение берёт из почвы.
7. Ночью в закрытой теплице углекислого газа становится больше.

Дано: 7 утверждений.

Найти: номера верных.

Решение

Шаг 1. Утверждения про фотосинтез (1, 3, 4, 6). Вспоминаем «кухню»: на входе свет, углекислый газ и вода, на выходе сахар и кислород. Утверждение 1 – верно. Утверждение 3 – верно, газы стоят на своих местах. Утверждение 4 – неверно: хлорофилл зелёный свет как раз почти не берёт и отражает, поэтому мы видим лист зелёным. Утверждение 6 – неверно, это старая ошибка, которую опроверг опыт ван Гельмонта с ивой: земля почти не полегчала, масса пришла из воздуха и воды.

Шаг 2. Утверждения про дыхание (2, 5, 7). Растение дышит всегда, значит, 2 – неверно. Утверждение 5 – верно: при дыхании сахар «сжигается», выходит углекислый газ. Утвержде-

ние 7 – верно: ночью фотосинтеза нет, дыхание идёт, и углекислый газ в закрытой теплице копится.

Шаг 3. Собираем ответ. Верные: 1, 3, 5, 7. Записываем по возрастанию без пробелов.

⚠ Типичная ошибка. Отметить утверждение 2, потому что в памяти засело «днём растения делают кислород, ночью дышат». Слово «только» в тестах – повод насторожиться: проверь, нет ли исключений. Вторая ловушка – утверждение 4: оно звучит логично («зелёный – значит, любит зелёный»), но всё наоборот.

Ответ: 1357.

Своя задача по программе сферы, модуль 2

Реши сам

1.1. Выбери верные утверждения о корнях. Ответ – номера по возрастанию, без пробелов.

1. Корень всасывает воду вместе с растворёнными в ней минеральными солями.
2. Корням кислород не нужен: они же под землёй.
3. Корневые волоски сильно увеличивают поверхность, через которую всасывается вода.
4. Сахар растение получает из почвы через корни.
5. Если землю в горшке постоянно заливать, корни могут загнить.
6. Азот, фосфор и калий растение берёт из почвы.

1.2. Выбери верные утверждения о свете. Ответ – номера по возрастанию, без пробелов.

1. Хлорофилл сильнее всего поглощает красный и синий свет.
2. Фитолампы часто светят розово-фиолетовым, потому что в их свете много красного и синего.
3. Зелёный лист лучше всего использует зелёный свет.
4. В солнечный летний полдень освещённость может быть в сотни раз больше, чем на рабочем месте в офисе.
5. Если освещённость вырастет вдвое, фотосинтез обязательно тоже вырастет вдвое.
6. При очень слабом свете растение может тратить на дыхание больше сахара, чем успевает сделать.

Задача 1.2 Что с растением?

факторы роста, признаки нехватки, сопоставление

В кружке ребята ведут дневник наблюдений за растениями. Сопоставь признаки (А–Д) и самую вероятную причину (1–6). Одна причина лишняя. Ответ – пять цифр подряд: номер причины для А, для Б, для В, для Г и для Д.

- **А.** Рассада вытянулась, стебли тонкие и бледные, все тянутся к окну.
- **Б.** В жаркий полдень листья поникли, а земля в горшке сухая и отстала от стенок.

- **В.** Земля в горшке мокрая уже неделю, из него пахнет гнилью, растение вянет.
- **Г.** Старые нижние листья равномерно желтеют, растение растёт медленно, хотя света и воды хватает.
- **Д.** Томаты в теплице цвели, а завязей нет; днём термометр там показывал +38 °С.

Причины: **1** – перегрев; **2** – не хватает азота; **3** – мало света; **4** – корням не хватает воздуха; **5** – не хватает воды; **6** – слишком холодные ночи.

Дано: 5 признаков, 6 причин, одна лишняя.

Найти: строку из пяти цифр.

Решение

Шаг 1. Начинаем с самых ясных. Д: при жаре выше 30–32 °С пыльца томата развивается плохо, а при +35 °С и выше цветы опадают без завязей – это перегрев, **1**. Б: сухая земля и полуденная жара – растению не хватает воды, **5**.

Шаг 2. Два случая с вянущим растением. Вянет и в Б, и в В, но причины противоположные. В Б земля сухая, а в В – мокрая целую неделю и пахнет гнилью. Воды там с избытком, а вот воздуха корням не осталось – они задохнулись и загнили. В – **4**.

Шаг 3. Бледная вытянутая рассада. Растение «ищет» свет: вытягивает стебель к окну и не успевает окрепнуть. А – **3**.

Шаг 4. Желтеющие нижние листья. Свет и вода в условии специально исключены. Равномерно желтеют старые листья, когда растению не хватает азота: он уходит из старых листьев в молодые. Г – **2**.

Шаг 5. Проверка. Использованы 3, 5, 4, 2, 1. Лишней осталась причина 6 – «холодные ночи», ни один признак на неё не указывает. Каждая цифра встречается один раз.

 **Типичная ошибка.** Увидеть в В слово «вянет» и ответить «не хватает воды». Вянет растение и от засухи, и от перелива: в обоих случаях вода не доходит до листьев. Решает то, что сказано про землю.

Ответ: 35421.

Своя задача по программе сферы, модуль 2

Реши сам

1.3. Сопоставь задачу умной теплицы (А–Д) и датчик, который для неё нужен (1–6). Один датчик лишний. Ответ – пять цифр подряд для А, Б, В, Г, Д.

- **А.** Узнать, не пора ли поливать грядку.
- **Б.** Вовремя открыть форточку, пока теплица не перегрелась.
- **В.** Включить фитолампы, когда на улице стемнело.
- **Г.** Понять, хватает ли растениям углекислого газа днём.
- **Д.** Заметить, что воздух слишком сырой и на стёклах вот-вот выступит роса.

Датчики: **1** – термометр; **2** – датчик влажности почвы; **3** – фоторезистор; **4** – датчик CO₂; **5** – датчик влажности воздуха; **6** – анемометр (меряет скорость ветра).

1.4. В трёх теплицах записали средние показания за неделю. Пользуясь таблицей норм из раздела 1.1, определи, какая культура лучше всего подойдёт для каждой: **А** – томат, **Б** – огурец, **В** – листовой салат. Ответ – три буквы подряд для теплиц 1, 2, 3.

- **Теплица 1:** днём 26 °С, ночью 19 °С, влажность воздуха 85 %.
- **Теплица 2:** днём 18 °С, ночью 12 °С, влажность воздуха 70 %.
- **Теплица 3:** днём 23 °С, ночью 17 °С, влажность воздуха 65 %.

Задача 1.3 Пузырьки элодеи

фотосинтез и свет, вывод по опыту

Элодея – водное растение из аквариума. Ребята поставили веточку элодеи в стакан с водой и светили на неё настольной лампой с разного расстояния. Каждый раз ждали 5 минут, а потом считали пузырьки газа, которые поднимались от веточки.

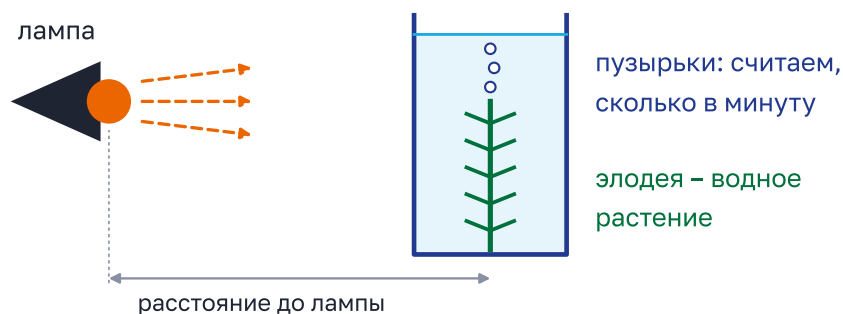


Рисунок 1.10 – Опыт с элодеей

Таблица 1.4

Расстояние до лампы, см	Пузырьков в минуту
10	42
20	40
30	23
40	12

Расстояние до лампы, см	Пузырьков в минуту
50	6

Какие выводы **следуют из этого опыта**? Ответ – номера по возрастанию, без пробелов.

1. Чем ближе лампа, тем больше пузырьков в минуту.
2. Пузырьки – это кислород.
3. Если отодвинуть лампу на 100 см, пузырьков не будет совсем.
4. Между 10 и 20 см число пузырьков почти не меняется: вблизи лампы света растению уже почти хватает.
5. Когда лампу отодвинули с 20 до 40 см, число пузырьков уменьшилось больше чем в 3 раза.
6. Элодея дышит только в темноте.

Дано: таблица «расстояние – число пузырьков».

Найти: выводы, которые можно сделать из этих чисел.

Решение

Шаг 1. Что значит «следует из опыта». Вывод должен опираться на то, что ребята увидели и посчитали. Утверждение может быть верным в жизни, но если опыт его не проверял, из опыта оно не следует. Проверяем каждое по таблице.

Шаг 2. Утверждение 1. Идём по таблице сверху вниз: 42, 40, 23, 12, 6 – числа только уменьшаются, пока лампа отодвигается. Первый вывод опыт подтверждает.

Шаг 3. Утверждение 2. Мы знаем, что при фотосинтезе выделяется кислород. Но ребята газ не проверяли – например, тлеющей лучинкой, которая в кислороде вспыхивает. Они только считали пузырьки, так что из опыта это не следует.

Шаг 4. Утверждение 3. На 100 см лампу не ставили. Сколько будет пузырьков, можно только гадать: в комнате есть и другой свет. Опыт об этом молчит.

Шаг 5. Утверждение 4. От 10 до 20 см: 42 и 40 – разница 2 пузырька. От 20 до 30 см: 40 и 23 – разница 17. Вблизи лампы прибавка света почти ничего не даёт – это насыщение, как на графике раздела 1.2. Следует.

Шаг 6. Утверждение 5. На 20 см – 40 пузырьков, на 40 см – 12. Делим: $40:12 \approx 3,3$. Уменьшилось в 3,3 раза – больше чем в 3, утверждение подходит.

Шаг 7. Утверждение 6. Про темноту в опыте ничего нет, а в жизни это вообще неверно: растение дышит всегда.

Настоящий исследователь заметил бы ещё одну вещь: если лампа старая, накаливания, она греет воду, и вблизи вода теплее. Так что часть разницы могла дать температура. Чтобы это исключить, между лампой и стаканом ставят банку с водой – она забирает тепло, а свет пропускает.

 **Типичная ошибка.** Отметить утверждение 2: «это же правда, я читал». В задачах «что следует из опыта» проверяют не эрудицию, а умение отделить увиденное от того, что знаешь заранее. Спроси себя про каждое утверждение: какие числа или наблюдения его доказывают?

Ответ: 145.

Своя задача по программе сферы, модуль 2

Реши сам

1.5. Комнатную пеларгонию (герань) двое суток держали в тёмном шкафу. Потом один лист с обеих сторон закрыли полоской фольги и поставили растение на солнечное окно на целый день. Вечером лист сорвали, прокипятили, обесцветили в спирте и капнули на него раствор йода. Йод синее там, где есть крахмал. Открытая часть листа посинела, закрытая фольгой – нет. Какие выводы следуют из этого опыта? Ответ – номера по возрастанию, без пробелов.

1. На свету в листе образовался крахмал.
2. Без света крахмал в листе не образуется.
3. Для образования крахмала нужен углекислый газ.
4. При фотосинтезе выделяется кислород.
5. Крахмал появился только в той части листа, куда попадал свет.
6. Крахмал из листа уходит в корни.

1.6. Семена фасоли разложили в четыре банки и неделю наблюдали за ними. Банки 1–3 стояли в тёмном шкафу при +22 °С, банка 4 – в холодильнике при +4 °С (там тоже темно).

- **Банка 1:** семена на сухой вате – не проросли.
- **Банка 2:** семена на влажной вате – проросли.
- **Банка 3:** семена целиком залиты прокипячённой и остуженной водой, в которой почти не осталось воздуха, – не проросли.
- **Банка 4:** семена на влажной вате – не проросли.

Какие две банки нужно сравнить, чтобы доказать, что семенам для прорастания нужен воздух? А какие – что нужно тепло?

Задача 1.4 Самая короткая доска

закон минимума, проценты

На гидропонной ферме в Академгородке растят салат. Когда всего хватает, с 1 м² снимают 3,6 кг за один урожай. Датчики показали, сколько каждого фактора растения получают сейчас (в процентах от того, что им нужно): свет – 55 %, тепло – 90 %, вода – 100 %, азот – 70 %, углекислый газ – 80 %.

Считай по упрощённому закону минимума: урожай равен наибольшему возможному, умноженному на долю того фактора, которого не хватает сильнее всего.

- а) Какой урожай с 1 м² ферма получает сейчас?
- б) Поставили фитолампы, и свет стал 100 %. Какой урожай будет теперь?
- в) На сколько процентов вырастет урожай после установки ламп? Ответ округли до целых.

Дано: максимум 3,6 кг/м²; свет 55 %, тепло 90 %, вода 100 %, азот 70 %, CO₂ 80 %.

Найти: урожай сейчас, урожай после ламп, рост в процентах.

Решение

Шаг 1. Ищем самую короткую доску. Сравниваем: 55 %, 90 %, 100 %, 70 %, 80 %. Меньше всех – свет, 55 %.

Шаг 2. Урожай сейчас. 55 % – это 0,55. Урожай: $3,6 \cdot 0,55 = 1,98$ кг с 1 м².

Шаг 3. После ламп. Свет стал 100 %, и он больше не короткая доска. Смотрим заново: 100 %, 90 %, 100 %, 70 %, 80 %. Теперь меньше всех азот – 70 %. Урожай: $3,6 \cdot 0,7 = 2,52$ кг с 1 м².

Шаг 4. Рост в процентах. Процент изменения считаем от того, что было до: урожай прибавил $2,52 - 1,98 = 0,54$ кг, а было 1,98 кг. $0,54 : 1,98 \cdot 100 = 27,3$ %, до целых – 27 %.

Лампы дали прибавку, но не ту, на которую надеялся хозяин: «ведь света стало почти вдвое больше». Бочку теперь держит азот. Следующий шаг хорошего агронома – подкормка, а не новые лампы.

Модель, конечно, упрощённая: в жизни факторы влияют друг на друга, и урожай не всегда строго пропорционален самому слабому. Но для прикидки, куда тратить деньги, её хватает.

 **Типичная ошибка.** Перемножить все проценты: $3,6 \cdot 0,55 \cdot 0,9 \cdot 1 \cdot 0,7 \cdot 0,8 = 1,0$ кг. Или взять среднее всех процентов. По закону минимума работает только одна доска – самая короткая. И вторая ловушка в пункте б: оставить в ответе 3,6 кг, забыв, что после света короткой доской стал азот.

Ответ: а) 1,98 кг; б) 2,52 кг; в) на 27 %.

Своя задача по программе сферы, модуль 0

Реши сам

1.7. В идеальных условиях с 1 м² теплицы снимают 12 кг огурцов за сезон. Сейчас растения получают: свет – 85 %, тепло – 60 %, вода – 95 %, питание – 90 %. Какой фактор ограничивает урожай и сколько килограммов снимут с 1 м²? Считай по упрощённому закону минимума, как в задаче 1.4.

1.8. На ферме салата максимальный урожай – 4 кг с 1 м². Сейчас растения получают: свет – 70 %, тепло – 95 %, вода – 100 %, азот – 75 %, CO₂ – 85 %. Хозяин хочет снимать 3,4 кг с 1 м². а) Какие факторы обязательно нужно подтянуть? б) Сколько он получит, если улучшит только свет (до 100 %)? Считай по упрощённому закону минимума.

Задача 1.5 Досветка рассады

длина дня, временные интервалы, досветка

1 марта в Новосибирске солнце встаёт примерно в 8:21, а садится в 19:02. Рассаде томатов на подоконнике хотят дать 16 часов света в сутки. Утром лампа включается в 6:00 и горит до

восхода, вечером включается на закате.

- а) Сколько длится светлое время суток 1 марта?
- б) Сколько минут в сутки должна гореть лампа?
- в) Во сколько её выключить вечером?

Дано: восход 8:21; закат 19:02; нужно 16 ч света; утром лампа с 6:00.

Найти: длину дня; время работы лампы, мин; время выключения.

Решение

Шаг 1. Переводим время в минуты от полуночи. Восход: $8 \cdot 60 + 21 = 501$ мин. Закат: $19 \cdot 60 + 2 = 1142$ мин.

Шаг 2. Длина дня. $1142 - 501 = 641$ мин. Переводим обратно: $641:60 = 10$ и 41 в остатке, то есть 10 ч 41 мин.

Шаг 3. Сколько не хватает. 16 ч – это $16 \cdot 60 = 960$ мин. Не хватает $960 - 641 = 319$ мин – столько и должна гореть лампа за сутки.

Шаг 4. Утренняя часть. С 6:00 (360 мин) до восхода (501 мин): $501 - 360 = 141$ мин.

Шаг 5. Вечерняя часть. $319 - 141 = 178$ мин = 2 ч 58 мин. Выключаем в $19:02 + 2:58 = 22:00$.

Для пункта в есть путь короче. Свет – солнечный или от лампы – идёт без перерыва с 6:00 до момента выключения. Значит, выключить надо через 16 часов после 6:00, в 22:00, и восход с закатом для этого не нужны. Без них не обойтись только в пункте б.

Пять с лишним часов лампы в сутки для начала марта в Сибири – вполне жизненно: на картинке с длиной дня видно, что солнца тогда меньше одиннадцати часов.

 **Типичная ошибка.** Вычитать время как обычные десятичные числа: $19,02 - 8,21 = 10,81$ и написать «10 ч 81 мин» или «10,81 ч». В часе 60 минут, поэтому минуты при вычитании занимают у часов 60, а не 100.

Ответ: а) 10 ч 41 мин; б) 319 мин; в) в 22:00.

Своя задача по программе сферы, модуль 2

Реши сам

1.9. 15 февраля в Новосибирске солнце встаёт примерно в 8:53, а садится в 18:32. Сколько длится светлое время суток? Ответ дай в часах и минутах.

1.10. Салату в школьной теплице нужно 14 часов света в сутки. Лампы включают только вечером, на закате. 15 марта солнце встаёт примерно в 7:46, а садится в 19:30. Во сколько выключить лампы? Сколько минут они будут гореть?

Задача 1.6 Сутки теплицы

дыхание, фотосинтез, углекислый газ, чтение графика

Датчик CO₂ в закрытой теплице с томатами записывал показания каждые 2 часа. Солнце встаёт около 6:00, садится около 20:00. На улице углекислого газа около 420 ppm.

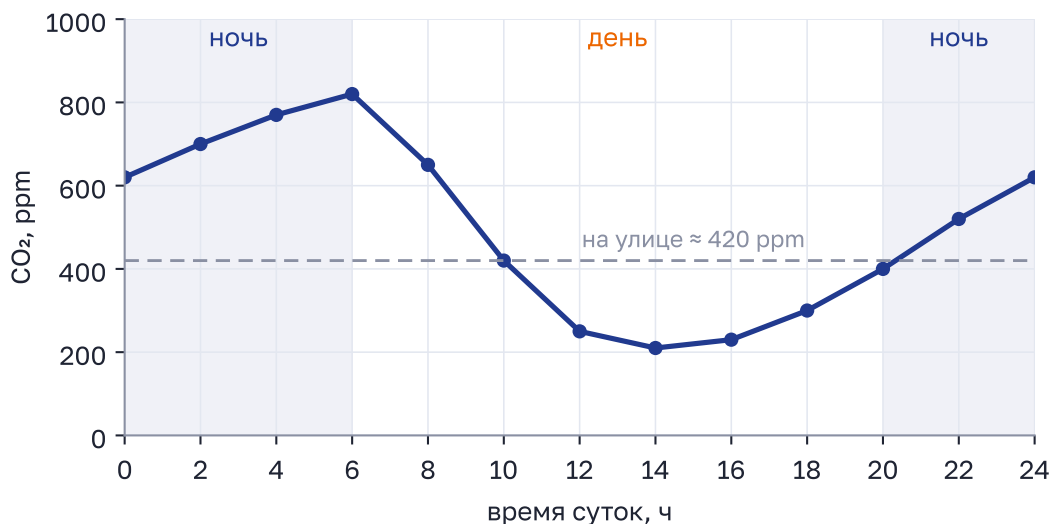


Рисунок 1.11 – Углекислый газ в закрытой теплице за сутки

Выбери все верные утверждения. Ответ – номера по возрастанию, без пробелов.

1. Больше всего углекислого газа в теплице около 6 утра, перед рассветом.
2. Ночью углекислый газ прибавляется, потому что растения дышат, а фотосинтеза нет.
3. Меньше всего углекислого газа в 12:00, когда солнце выше всего.
4. Днём растения перестают дышать.
5. В 14:00 углекислого газа в теплице примерно вдвое меньше, чем в уличном воздухе.
6. Если днём открыть форточку, растениям достанется больше углекислого газа.
7. С 20:00 до 6:00 углекислый газ прибавлялся в среднем примерно на 40 ppm в час.

Дано: график CO₂ за сутки; на улице 420 ppm.

Найти: номера верных утверждений.

Решение

Шаг 1. Максимум и минимум по графику. Самая высокая точка – 820 ppm в 6:00. Утверждение 1 верно. Самая низкая – 210 ppm в 14:00, а в 12:00 было 250 ppm. Утверждение 3 неверно: минимум позже полудня. Растения всё утро и полдень вытягивают газ из закрытой теплицы, и запас кончается к середине дня.

Шаг 2. Почему так. Ночью фотосинтеза нет, дыхание идёт – газ копится. Утверждение 2 верно. Утверждение 4 неверно: дыхание не выключается никогда, днём его просто «перекрывает» фотосинтез.

Шаг 3. Сравнение с улицей. В 14:00 – 210 ppm, на улице 420 ppm: $420:210 = 2$. Вдвое меньше, утверждение 5 верно. Значит, через открытую форточку в теплицу пойдёт воздух, где углекислого газа больше. Утверждение 6 верно.

Шаг 4. Скорость прибавки ночью. В 20:00 – 400 ppm, в 6:00 – 820 ppm. Прибавилось $820 - 400 = 420$ ppm. Прошло времени: от 20:00 до полуночи 4 ч, от полуночи до 6:00 ещё 6 ч, всего 10 ч. В среднем $420 : 10 = 42$ ppm в час – примерно 40. Утверждение 7 верно.

Шаг 5. Собираем ответ. Верные: 1, 2, 5, 6, 7.

 **Типичная ошибка.** Посчитать время от 20:00 до 6:00 как $20 - 6 = 14$ ч: через полночь так вычитать нельзя, дели промежуток на две части. И второе – ответить на утверждение 3 «по логике», не глядя в график: «солнце в полдень выше всего – значит, газа меньше всего». Задача проверяет, умеешь ли ты читать график, а не угадывать.

Ответ: 12567.

Своя задача по программе сферы, модуль 2

Реши сам

1.11. В закрытом боксе с рассадой датчик CO_2 в 21:00 показал 430 ppm, а в 6:00 – 610 ppm. На сколько ppm в час в среднем прибавлялся углекислый газ? Сколько показывал датчик в 3:00, если газ прибавлялся равномерно?

1.12*.* Растение закрыли прозрачным колпаком с датчиком CO_2 . Лампа горела 30 минут, и за это время углекислого газа под колпаком стало меньше: было 450 ppm, стало 330 ppm. Потом лампу выключили, и за следующие 30 минут газ поднялся с 330 до 390 ppm. Считай, что дыхание идёт с одной и той же скоростью и при свете, и в темноте. Сколько ppm углекислого газа в минуту поглощал фотосинтез, пока горела лампа?

Задача 1.7*.*. Роса на стёклах

влажность воздуха, конденсация, точка росы, проценты

Вечером в 20:00 в теплице с томатами $+22^\circ\text{C}$, влажность воздуха 70 %. Объём воздуха в теплице – 36 м^3 . Метеостанция обещает холодную ночь, и к утру воздух в теплице остынет до $+12^\circ\text{C}$. Теплица закрыта, воздух не меняется. В таблице – сколько граммов пара может быть в 1 м^3 воздуха самое большее при разной температуре.

Таблица 1.5

Температура, $^\circ\text{C}$	Пара в 1 м^3 самое большее, г
10	9,4
12	10,7
14	12,1
16	13,6
18	15,4
20	17,3

Температура, °C	Пара в 1 м³ самое большее, г
22	19,4
25	23,0

- а) Примерно при какой температуре на стёклах начнёт выпадать роса?
б) Сколько граммов воды к утру выпадет росой? Ответ округли до целых.

Дано: $t_1 = 22\text{ °C}$; влажность 70 %; $V = 36\text{ м}^3$; $t_2 = 12\text{ °C}$; таблица.

Найти: точку росы; массу росы.

Решение

Шаг 1. Сколько пара в воздухе сейчас. При $+22\text{ °C}$ в 1 м^3 может быть самое большее 19,4 г пара. Влажность 70 % значит, что пара 70 % от этого предела: $19,4 \cdot 0,7 = 13,58\text{ г}$ в каждом кубометре.

Шаг 2. Точка росы. Пока теплица остывает, пара в воздухе остаётся те же 13,58 г/м³, а предел опускается. Ищем в таблице, когда предел станет меньше 13,58: при 16 °C он 13,6 г – почти столько же, при 14 °C уже 12,1 г. Значит, роса начнёт выпадать около $+16\text{ °C}$.

Шаг 3. Сколько пара «не поместится» утром. При $+12\text{ °C}$ в 1 м^3 может остаться самое большее 10,7 г. Лишнее выпадет каплями: $13,58 - 10,7 = 2,88\text{ г}$ с каждого кубометра.

Шаг 4. На всю теплицу. $2,88 \cdot 36 = 103,68\text{ г}$, до целых – 104 г.

Сто граммов воды – это полстакана, размазанные тонкой плёнкой по стёклам и листьям. На листьях томата такая плёнка – приглашение для грибка. Поэтому вечером перед холодной ночью теплицу проветривают, пока воздух снаружи суше, или подогревают, чтобы не опуститься до точки росы.

⚠ Типичная ошибка. Вычесть проценты вместо граммов: «было 70 %, при остывании станет 100 %, выпадет 30 %». Проценты влажности считаются каждый раз от своего предела, а предел зависит от температуры. Складывать и вычитать можно только граммы. Ещё одна ошибка – забыть умножить на объём и ответить «2,88 г»: это роса с одного кубометра, а не со всей теплицы.

Ответ: а) около $+16\text{ °C}$; б) 104 г.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «У природы есть плохая погода»

Реши сам

1.13. В теплице $+20\text{ °C}$, в каждом кубометре воздуха 12,1 г водяного пара. Пользуясь таблицей из задачи 1.7, найди относительную влажность (в процентах, до целых). При какой температуре из таблицы начнёт выпадать роса?

1.14*.* В теплице с огурцами +25 °С, влажность 80 %, объём воздуха 50 м³. Пользуйся таблицей из задачи 1.7. а) Какая температура из таблицы – самая низкая, при которой роса ещё не выпадет? б) Ночью теплица остыла до +18 °С. Сколько граммов воды выпало росой? в) Ночная норма для огурца – 18–20 °С (раздел 1.1). Выпадет ли роса, если ночью держать в теплице +20 °С, а воздух вечером не осушать?

Микроклимат теплицы: считаем



Умная теплица живёт по числам. Датчики присылают «26 °C», «61 %», «3200 лк», автоматика сравнивает их с порогами и решает, открыть ли форточку и пора ли поливать. Здесь ты научишься читать эти числа, переводить ppm в проценты, считать влажность и часы света – и сам решать за автоматику. Задачи здесь устроены как на отборе: короткая история с показаниями датчиков, а ответ – число или строка из номеров устройств.

2.1. Что меряют в теплице и в чём

НА ПАЛЬЦАХ

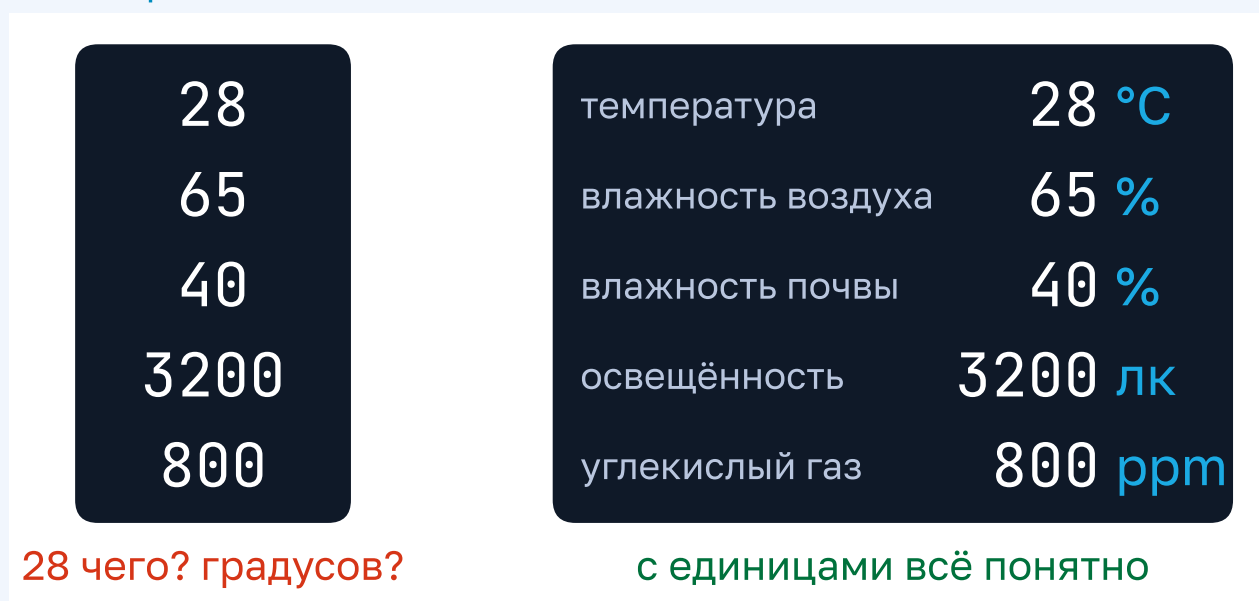


Рисунок 2.1 – Одно и то же табло теплицы без единиц и с единицами

Слева пять чисел, и по ним ничего не понять. Справа те же числа с единицами и подписями – и сразу видно, что в теплице тепло, свет слабоват, а углекислого газа хватает.

Какие датчики стоят в теплице и зачем, мы разобрали в главе 1 (раздел 1.6). Теперь нас интересуют их числа. У каждого числа своя единица, и одно и то же «28» может значить совсем разное. 28 °C – тепло, 28 % влажности воздуха – сушь, как в квартире зимой у батареи, а 28 ppm углекислого газа в воздухе у земли не бывает вовсе.

- **Температура воздуха t** – в градусах Цельсия, °C. При 0 °C замерзает вода, при 100 °C она кипит. Разница температур тоже измеряется в градусах: «ночью на 8 °C холоднее, чем днём».
- **Влажность воздуха φ** (греческая буква «фи») – в процентах. Что это за проценты и от чего они считаются, разберём в разделе 2.3.
- **Влажность почвы W** – тоже в процентах: 0 % – земля сухая, как песок на пляже в жару, 100 % – мокрее уже не бывает. Как датчик получает эти проценты, рассказано в главе 5.
- **Освещённость** – в люксах, лк (раздел 2.4).
- **Углекислый газ, CO₂**, – в ppm (раздел 2.2).

Нормы для томата, огурца и салата – в таблице раздела 1.1, общие ориентиры для тепличных овощей – в справочнике в конце книги. Но у каждой культуры нормы свои, у рассады – одни, у взрослого куста – другие. Поэтому в задачах отбора нормы и пороги дают прямо в условии, и считать надо по ним, даже если ты помнишь из таблицы другие числа.

И ещё. Датчик в симуляторе (Wokwi, Tinkercad) или на настоящей плате Arduino выдаёт голое число от 0 до 1023 – без градусов и процентов. Как перевести его в понятные единицы, рассказано в главе 5. В этой главе перевод уже сделан, и на табло – нормальные единицы.

Запомни. Температура – в °С, влажность воздуха и почвы – в %, свет – в люксах, углекислый газ – в ppm. Число без единицы – не ответ. Нормы и пороги бери из условия задачи.

2.2. Переводим единицы: ppm, люксы, время

НА ПАЛЬЦАХ

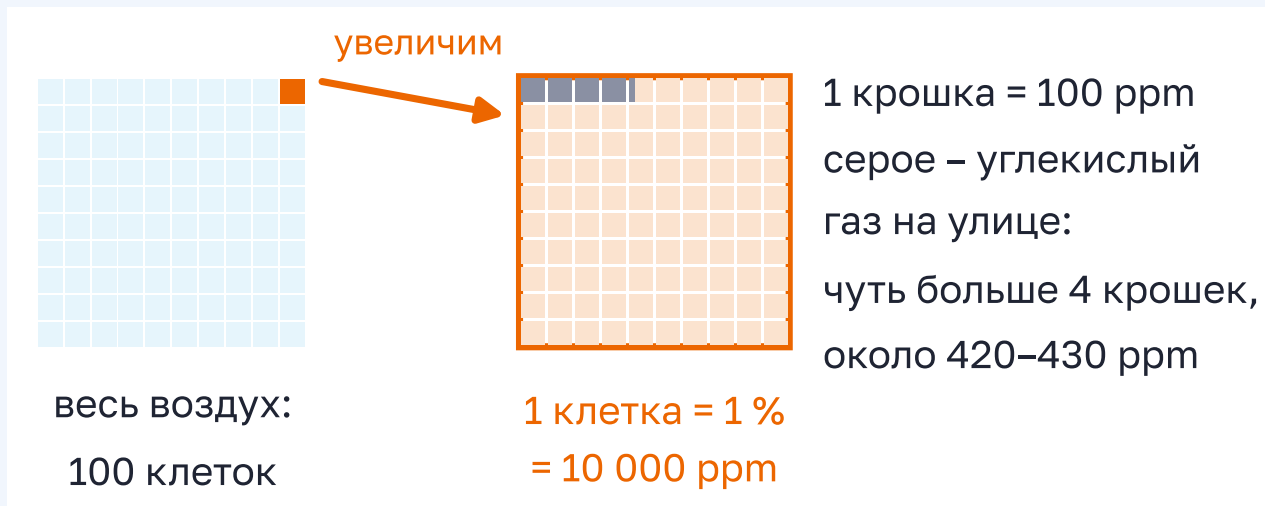


Рисунок 2.2 – Весь воздух – сто клеток. Одна клетка – это 1 %. Разрежем её ещё на сто крошек

Углекислого газа в воздухе так мало, что в процентах получается чуть больше 0,04 – неудобное число. Поэтому его считают крошками по одной миллионной – в ppm.

Процент – одна сотая часть. **ppm** (читается «пи-пи-эм», от английского parts per million) – одна миллионная часть, «штука на миллион». Если в воздухе 500 ppm углекислого газа, то из каждого миллиона частиц воздуха 500 – это CO₂.

Как связаны проценты и ppm? В миллионе – десять тысяч сотен, поэтому

$$1 \% = 10\,000 \text{ ppm}.$$

Отсюда правило: **проценты в ppm – умножь на 10 000, ppm в проценты – раздели на 10 000.**

- $0,04 \% = 0,04 \cdot 10\,000 = 400 \text{ ppm}$. На улице углекислого газа сейчас чуть больше – около 420–430 ppm (глава 1).
- $1200 \text{ ppm} = 1200 : 10\,000 = 0,12 \%$.

Зачем растению эти крошки и почему днём в закрытой теплице их становится меньше, чем на улице, – в главе 1. Для задач важнее другое: датчик может показывать CO₂ и в процентах, и в ppm, а порог в условии записан в чём-то одном. Для сравнения: в классе с закрытыми окнами к концу урока легко набирается больше 1000 ppm – и тянет в сон.

Люксы и килолюксы. Приставка «кило-» означает тысячу: 1 клк = 1000 лк. 3,2 клк = 3200 лк, а 800 лк = 0,8 клк.

Время. Датчики опрашивают быстро, и время в программах Arduino пишут в миллисекундах: $1 \text{ с} = 1000 \text{ мс}$. Если датчик опрашивают раз в 2000 мс, то есть раз в 2 с, за час он сделает $3600 : 2 = 1800$ замеров.

Часы и минуты считаются по 60, десятичные привычки тут подводят. Поэтому 7 ч 09 мин в часах – это $7 + 9 : 60 = 7,15 \text{ ч}$, а вовсе не 7,09 ч. И наоборот: $0,75 \text{ ч} = 0,75 \cdot 60 = 45 \text{ мин}$.

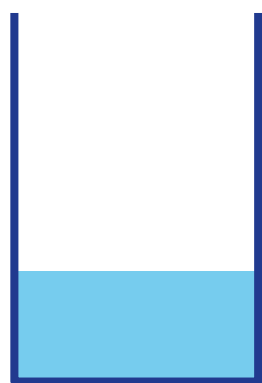
Запомни. $1 \% = 10\,000 \text{ ppm}$, 1 клк = 1000 лк, $1 \text{ с} = 1000 \text{ мс}$. В часе 60 минут, поэтому 1,5 ч – это 1 ч 30 мин.

Совет. Перед сравнением с порогом приведи оба числа к одной единице. Датчик показывает 0,12 %, а порог – 1000 ppm? Переведи показание в ppm и только потом сравнивай.

2.3. Влажность воздуха и почвы: проценты

НА ПАЛЬЦАХ

тёплый воздух, 25°C



вмещает 23 г

пара 7 г

7 г из 23 г – 30 %

холодный, 10°C



вмещает 9,4 г

пара 7 г

7 г из 9,4 г – 74 %

Рисунок 2.3 – Тёплый воздух – большая кружка, холодный – маленькая. Пара в них поровну

Пара одинаково, 7 г, но большая кружка заполнена меньше чем на треть, а маленькая – почти доверху. Остынет воздух – кружка станет ещё меньше, и пар начнёт переливаться через край каплями.

Про водяной пар и росу ты уже читал в главе 1: тёплый воздух удерживает пара больше, холодный – меньше. Теперь посчитаем это в граммах и процентах. Вот сколько пара самое большее вмещает один кубометр воздуха при разной температуре (числа те же, что в таблице задачи 1.7, просто шаг другой):

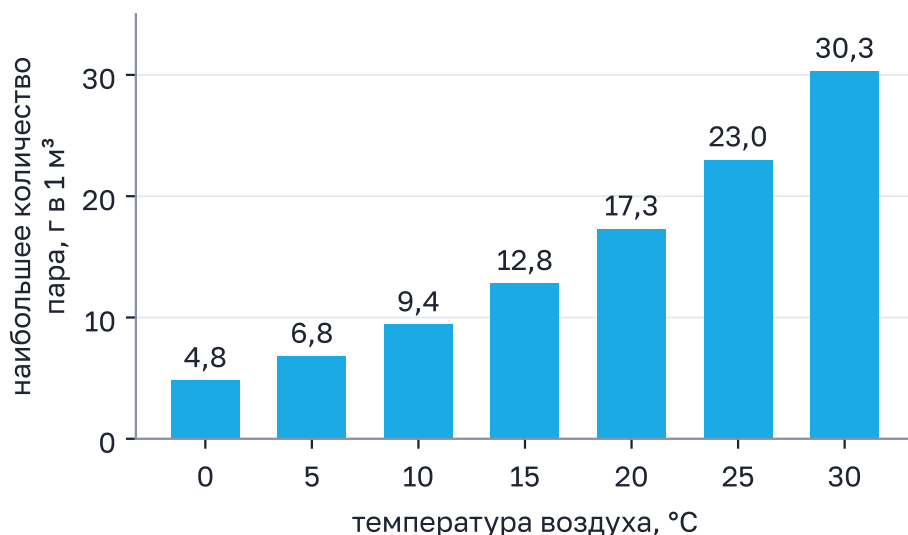


Рисунок 2.4 – Наибольшее количество водяного пара в 1 м³ воздуха при разной температуре

Таблица 2.1

Температура воздуха, °C	Наибольшее количество пара в 1 м³, г
0	4,8
5	6,8
10	9,4
15	12,8
20	17,3
25	23,0
30	30,3

Относительная влажность показывает, насколько воздух заполнен паром – сколько процентов пара в нём есть от того количества, которое он может вместить при этой температуре. Словами: влажность равна пару, который есть, делённому на пар, который поместился бы, и умноженному на 100 %. Буквами:

$$\varphi = \frac{a}{a_{\max}} \cdot 100 \%,$$

где a – сколько граммов пара в 1 м³ воздуха сейчас, $a_{\max}$ – сколько поместилось бы при этой температуре (из таблицы).

Пример. В теплице 15 °C, в каждом кубометре 9,6 г пара. Из таблицы $a_{\max} = 12,8$ г, значит, $\varphi = 9,6 : 12,8 \cdot 100 = 75 \%$.

Формулу можно повернуть и в другую сторону: если известна влажность, пар найдём так: $a = a_{\max} \cdot \varphi : 100$. При 25 °C и влажности 40 % в кубометре $23,0 \cdot 40 : 100 = 9,2$ г пара.

Отчего влажность меняется? Прежде всего от температуры. Пар никуда не делся, а воздух остыл – «кружка» стала меньше, и процент вырос. Поэтому к ночи влажность в теплице поднимается сама, без всякого полива, а на 100 % лишний пар выпадает росой: на стёклах теплицы, на траве, на очках, когда с мороза заходишь в тёплый автобус. Как считать росу в граммах, разобрано в задаче 1.7.

А ещё пар можно добавить или убрать, не трогая температуру. Этим занимается **увлажнитель**: он превращает воду в мелкий туман или пар. Один миллилитр воды весит 1 г, поэтому 100 мл испарившейся воды – это 100 г пара. Пока температура та же, «кружка» не меняется, и проценты влажности можно спокойно вычитать: поднять влажность с 40 % до 70 % – значит добавить 30 % от $a_{\max}$.

Растениям плохо и в слишком сыром воздухе, и в слишком сухом. В сырой теплице быстрее заводятся плесень и грибковые болезни. В сухой листья сильно испаряют воду и вянут, даже если земля влажная.

Влажность почвы – тоже проценты, но попроще: датчик настроен так, что 0 % – это сухая земля, а 100 % – только что политая водой. Если влажность ниже порога, пора поливать.

НА ПАЛЬЦАХ

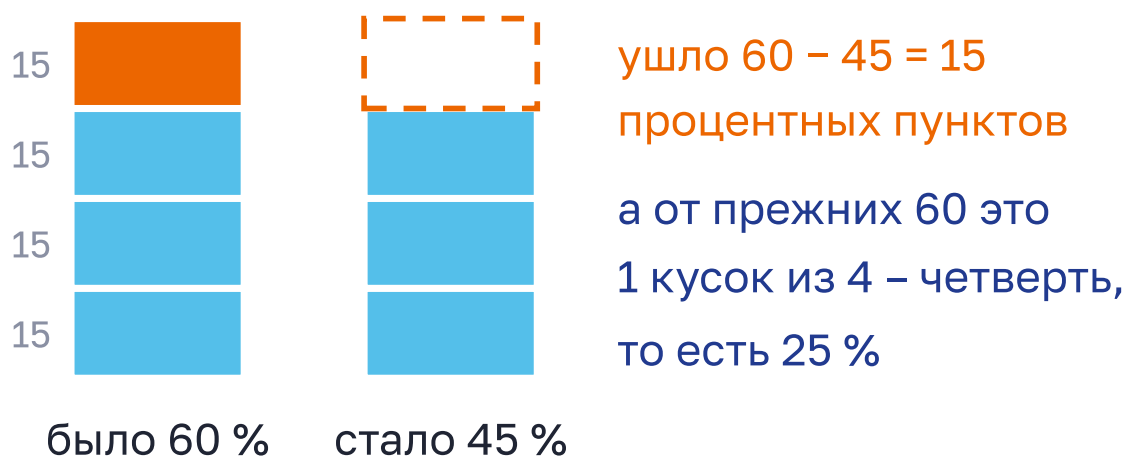


Рисунок 2.5 – Одна и та же потеря, измеренная двумя линейками

Влажность почвы упала с 60 % до 45 %. Ушёл один оранжевый кусок – 15. Сколько это: 15 или 25? Зависит от того, о чём спросили.

Процентные пункты. Влажность почвы была 60 %, стала 45 %. Разность $60 - 45 = 15$ – это **15 процентных пунктов**. А «на сколько процентов уменьшилась» считают от прежнего значения: $15 : 60 \cdot 100 = 25 \%$. Вопросы похожи как близнецы, а ответы разные, и на отборе легко ввести не тот.

Запомни. $\varphi = a : a_{\max} \cdot 100 \%$, и обратно $a = a_{\max} \cdot \varphi : 100$. Воздух остывает – влажность растёт. Разность процентов – это процентные пункты; «на сколько процентов» – всегда от прежнего значения.

2.4. Свет: люксы и часы света

НА ПАЛЬЦАХ

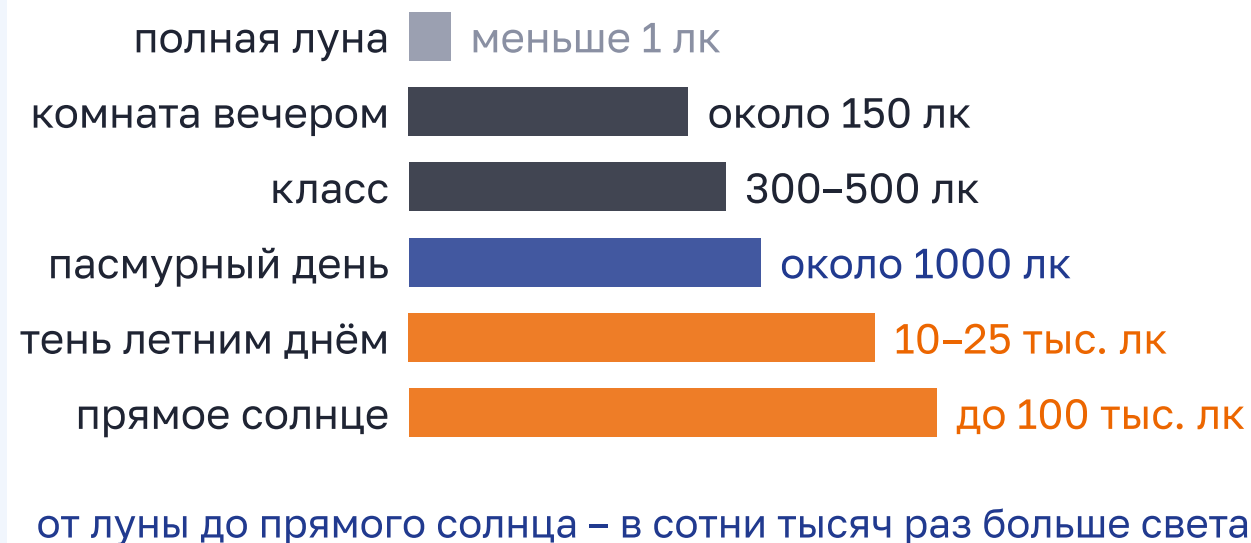


Рисунок 2.6 – Сколько люксов бывает вокруг нас

Глазу кажется, что в классе и на улице в пасмурный день светло примерно одинаково. Люксметр говорит другое: на улице света в несколько раз больше, а под летним солнцем – в сотни раз.

Освещённость – сколько света падает на каждый квадратный метр поверхности. Измеряют её в **люксах** (лк) прибором **люксметром**. Глаз – плохой люксметр: он подстраивается и под полумрак, и под яркое солнце, поэтому на глаз освещённость не оценить. А растению разница очень важна: свет для него – это еда (глава 1).

Таблица с люксами есть в разделе 1.2, а числа с картинки полезно держать в голове как «линейку» для проверки. Если в задаче у тебя вышло, что лампа досветки даёт 50 лк, это меньше, чем в комнате вечером, – подозрительно. А 500 000 лк – больше, чем солнце в полдень.

Фоторезистор в симуляторе выдаёт голое число от 0 до 1023, люксов он не знает. Разработчики сферы на вебинаре честно говорили: для теплицы переводить его в люксы не всегда нужно, порог «темно – включить лампу» можно подобрать прямо в этих числах. А в расчётных задачах освещённость дают уже в люксах или килолюксах.

Часы света. Растению важна не только яркость, но и то, сколько часов в сутки оно на свету. Как считать длину дня от восхода до заката и сколько досвечивать рассаду, разобрано в главе 1 (раздел 1.5 и задача 1.5): из позднего времени вычитаем раннее, а если минут не хватает, занимаем час – 60 минут. Но бывает, что свет горит ночью.

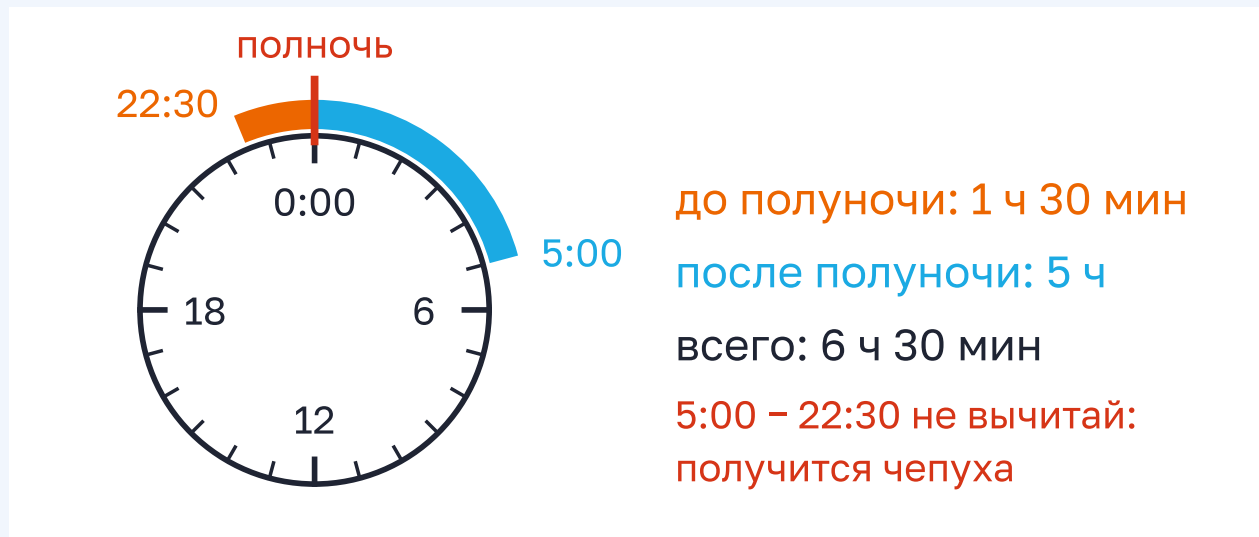


Рисунок 2.7 – Сутки – это круг. Отрезок через полночь режем на два куска

Лампа горит с 22:30 до 5:00. Полночь разрезает этот отрезок: кусок до полуночи и кусок после. Каждый считаем отдельно, потом складываем.

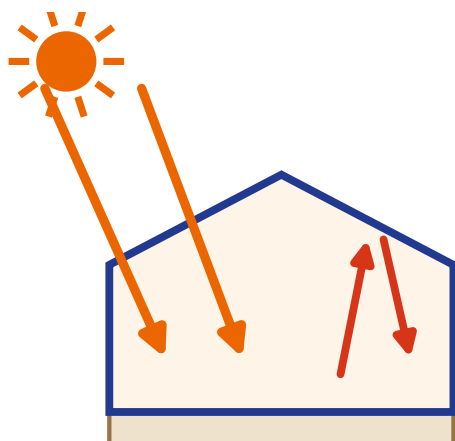
Через полночь. На сити-ферме в подвале солнца нет, и лампы работают по таймеру, часто ночью – по ночному тарифу электричество дешевле. Если отрезок переходит через полночь, вычитание «5:00 минус 22:30» даёт чепуху. Считай две части: до полуночи и после. Лампа горела с 22:30 до 5:00: до полуночи $24:00 - 22:30 = 1 \text{ ч } 30 \text{ мин}$, после – 5 ч, всего 6 ч 30 мин.

Доля суток. Переведи время в часы и раздели на 24. Лампа горит 10 ч 45 мин: это $10 + 45:60 = 10,75 \text{ ч}$, а доля суток $10,75:24 \cdot 100 = 44,8 \approx 45 \%$.

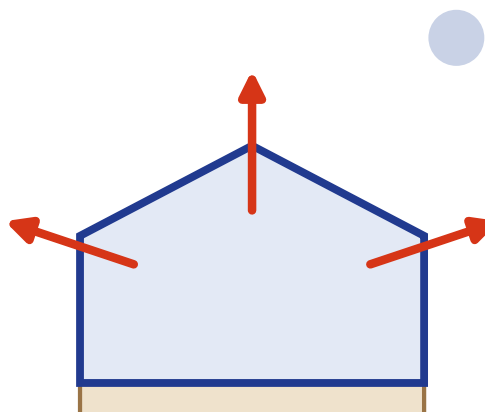
Запомни. Освещённость – в люксах; от лунной ночи до солнечного полдня она растёт в сотни тысяч раз. Отрезок через полночь – это две части: до 24:00 и после 0:00. Доля суток = часы : 24 · 100 %.

2.5. Температура за сутки

НА ПАЛЬЦАХ



**ДЕНЬ: свет входит,
тепло остаётся внутри**



**НОЧЬ: тепло уходит
через стекло и плёнку**

Рисунок 2.8 – Днём теплица копит тепло, ночью теряет его

Солнечный свет проходит сквозь стекло и греет землю, листья и воздух, а тёплому воздуху наружу не выйти. Ночью греть некому, и тепло уходит через стенки.

Температура в теплице за сутки ходит вверх и вниз, как качели. Ночью теплица остывает, и самое холодное время – под утро, перед восходом. Взошло солнце – начинает греть. Самое жаркое время наступает после полудня, часам к двум-трём дня: земля и воздух ещё какое-то время продолжают нагреваться. Потом солнце опускается, и к вечеру становится прохладнее.

В солнечный день в закрытой теплице бывает намного жарче, чем на улице, а жара опасна для урожая (раздел 1.1). Поэтому днём теплицу проветривают. А ночью растениям нужно прохладнее, чем днём, но не холодно, и в морозную ночь теплицу подогревают.

Датчик температуры не пишет плавную кривую – он делает **замеры** через равные промежутки: раз в 10 минут, раз в час, раз в 2 часа. По замерам считают:

- **сколько замеров вне нормы** – пересчитать те, что выше верхней границы и ниже нижней;
- **какую долю** они составляют – поделить на число всех замеров и умножить на 100 %;
- **сколько часов** было вне нормы – если в условии сказано, что показание держится до следующего замера, умножить число таких замеров на промежуток между ними;
- **размах** – самую высокую температуру минус самую низкую.

Внимательно с границами. «Норма 18–28 °C» означает, что и 18, и 28 – ещё норма. «Выше 28 °C» – 28 уже не выше.

Запомни. Холоднее всего под утро, жарче всего после полудня. Доля замеров = число нужных замеров : число всех · 100 %. Время вне нормы = число таких замеров · промежуток между замерами.

2.6. Пороги: когда включать и выключать

НА ПАЛЬЦАХ

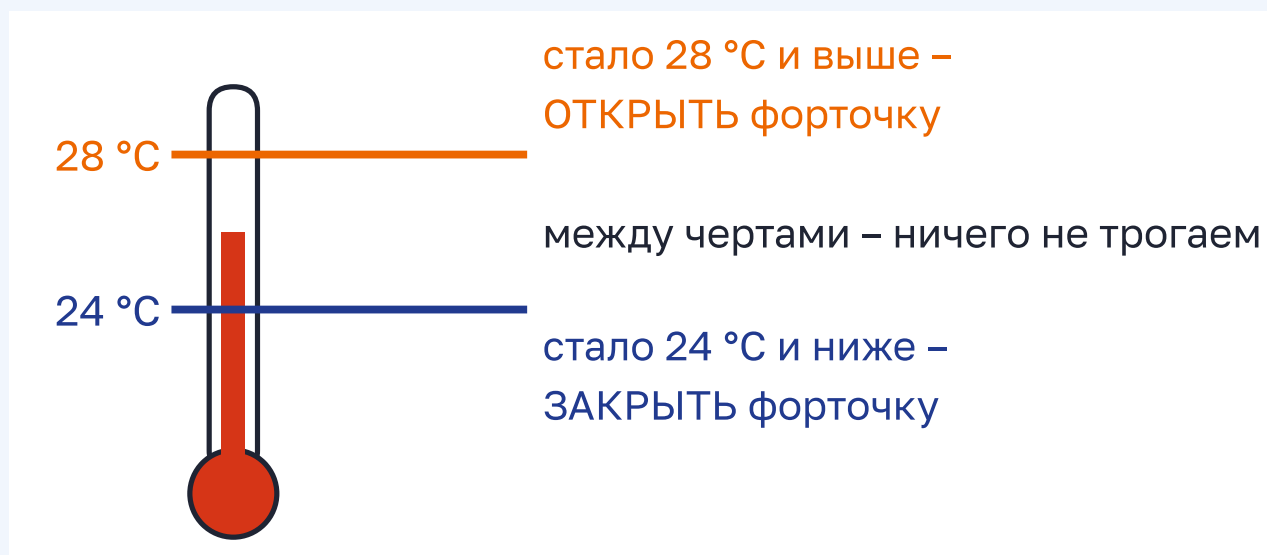


Рисунок 2.9 – Два порога у форточки

Верхняя черта открывает форточку, нижняя закрывает. Пока температура между ними, форточка остаётся такой, какой была.

Автоматика теплицы устроена просто: взять показание датчика, сравнить с числом-**порогом** и решить, включать устройство или нет. Например:

- температура 28 °C и выше – открыть форточку;
- температура ниже 16 °C – включить обогреватель;
- влажность почвы ниже 40 % – включить насос полива;
- освещённость ниже 5000 лк **и** на часах день – включить лампы.

Правила простые, а спотыкаются на них постоянно.

Граница. «Выше 28» и «28 и выше» – разные правила. Если показание равно порогу, первое правило ничего не включит, а второе включит. В программе это разница между знаками `>` и `≥`.

Единицы. Порог 5000 лк, показание 3,2 клк. Сравнить 3,2 с 5000 нельзя, сначала 3,2 клк = 3200 лк – и тогда видно, что света меньше порога.

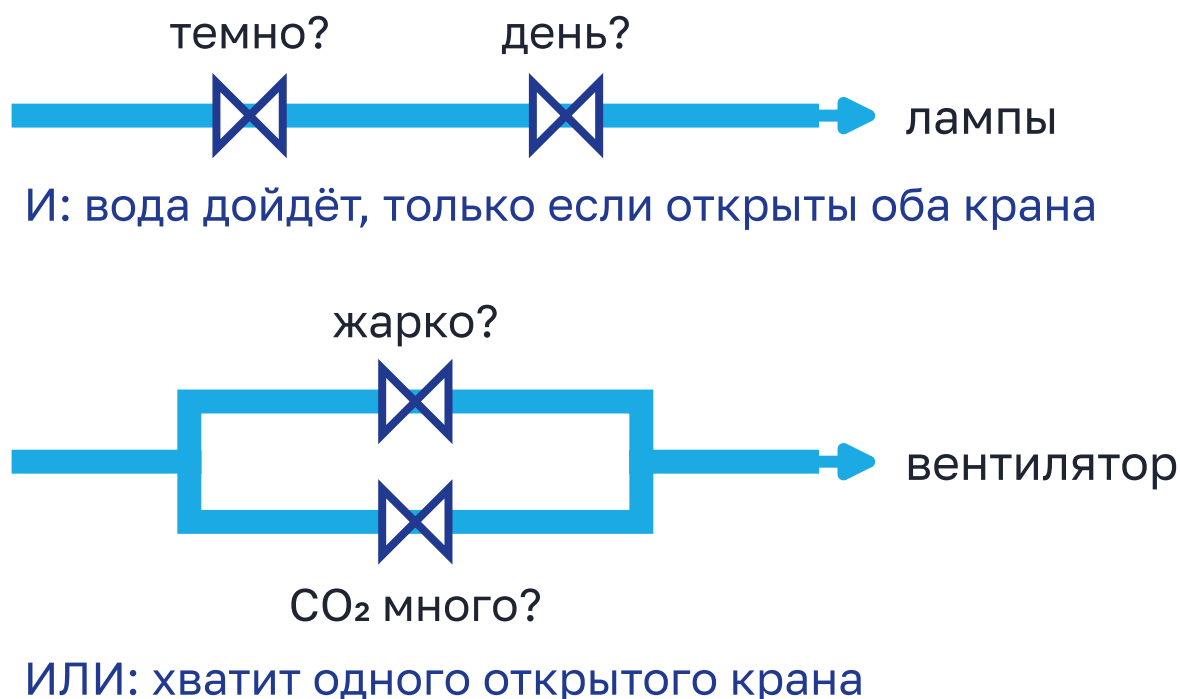


Рисунок 2.10 – «И» – краны подряд на одной трубе. «ИЛИ» – краны на двух трубах рядом

Каждый кран – один вопрос: ответ «да» открывает кран. По трубе с кранами подряд вода пройдёт, только если открыты оба. Если труб две, воде хватит и одной.

«И» и «или». Условие с «и» выполняется, только если верны обе части: лампы досветки не включатся в полночь, хотя темно, – сейчас не день. Условие с «или» выполняется, если верна хотя бы одна часть: вентилятор включается, когда жарко, когда много углекислого газа или когда и то и другое сразу.

Два порога. Возьмём форточку с одним порогом: открыть при 25 °С и выше, закрыть ниже 25 °С. Температура около 25 скачет на полградуса туда-сюда – и форточка хлопает каждую минуту, мотор изнашивается. Поэтому ставят два порога, как на картинке: открыть при 28 °С, а закрыть только при 24 °С. Между ними ничего не меняем. Такой приём называется **гистерезис**. Слово трудное, а суть бытовая: не дёргаться из-за каждой мелочи.

С двумя порогами одного показания мало. При 26 °С форточка может быть и открыта, и закрыта – смотря что было раньше. Поэтому задачи на два порога решают по порядку, замер за замером, и каждый раз записывают, в каком состоянии устройство.

Запомни. Проверь знак на границе ($>$ или $\geq$) и единицы. «И» – нужны обе части, «или» – хватит одной. С двумя порогами между ними устройство остаётся как было, поэтому иди по замерам по порядку.

2.7. Разбираем задачи

Задача 2.1 Хватит подкормки

разминка · перевод единиц, ppm и проценты

В теплицу подают углекислый газ, чтобы растения быстрее росли. Подачу выключают, когда углекислого газа больше 1000 ppm. Датчик показывает долю углекислого газа в процентах: 0,12 %. Сколько это ppm? На сколько ppm показание выше порога? Нужно ли выключать подачу?

Дано: показание 0,12 %; порог выключения – больше 1000 ppm; 1 % = 10 000 ppm.

Найти: показание в ppm; на сколько выше порога.

Решение

Шаг 1. Переводим проценты в ppm. В одном проценте 10 000 ppm, значит, умножаем:
 $0,12 \cdot 10\,000 = 1200$ ppm.

Шаг 2. Сравниваем с порогом. $1200 > 1000$ – показание выше порога на
 $1200 - 1000 = 200$ ppm. Подачу нужно выключить.

Прикинем: 0,1 % – это 1000 ppm, а 0,12 % чуть больше, значит, и ppm чуть больше тысячи. 1200 подходит.

 **Типичная ошибка.** Сравнить 0,12 с 1000 напрямую и решить, что углекислого газа совсем мало. Числа в разных единицах не сравнивают. Вторая ошибка – умножить на 1000 или на 100 и получить 120 или 12 ppm. Миллионных в одном проценте десять тысяч.

Ответ: 1200 ppm; на 200 ppm выше порога; подачу выключить.

Своя задача по программе сферы, модуль 2

Реши сам

2.1. В конце урока датчик в классе показал 1850 ppm углекислого газа. Сколько это процентов?

2.2. Лампы досветки включаются, если освещённость меньше 4,5 клк. Датчик показывает 3800 лк. Включатся ли лампы? Сколько люксов не хватает до порога?

Задача 2.2 Почему к ночи сыро

относительная влажность, проценты, процентные пункты

Днём в теплице 25 °С, и в каждом кубометре воздуха 16,1 г водяного пара. К ночи воздух остыл до 20 °С, окна закрыты, пара осталось столько же. Найди относительную влажность днём и ночью (до целых процентов). На сколько процентных пунктов она выросла? Наибольшее количество пара возьми из таблицы в разделе 2.3.

Дано: днём $t = 25\text{ }^{\circ}\text{C}$; ночью $t = 20\text{ }^{\circ}\text{C}$; $a = 16,1\text{ г/м}^3$ (не меняется); $a_{\text{max}}(25\text{ }^{\circ}\text{C}) = 23,0\text{ г/м}^3$; $a_{\text{max}}(20\text{ }^{\circ}\text{C}) = 17,3\text{ г/м}^3$.

Найти: $\varphi_{\text{день}}$, $\varphi_{\text{ночь}}$ и их разность.

Решение

Шаг 1. Что происходит. Пара в воздухе столько же, а вмещать воздух может всё меньше – он остывает. Значит, влажность должна вырасти. Посчитаем, насколько.

Шаг 2. Днём. При $25\text{ }^{\circ}\text{C}$ кубометр может вместить $23,0\text{ г}$ пара. В нём $16,1\text{ г}$:
 $\varphi = 16,1 : 23,0 \cdot 100 = 70\text{ \%}$.

Шаг 3. Ночью. При $20\text{ }^{\circ}\text{C}$ вмещается $17,3\text{ г}$. Пара те же $16,1\text{ г}$:
 $\varphi = 16,1 : 17,3 \cdot 100 = 93,06 \approx 93\text{ \%}$.

Шаг 4. Разность. $93 - 70 = 23$ процентных пункта.

Ночью $93\text{ \%}$ – воздух почти полон, ещё чуть-чуть остынет, и стёкла станут мокрыми. Кто заходил утром в теплицу на даче, видел это своими глазами.

⚠ Типичная ошибка. Посчитать ночную влажность по дневной таблице – снова разделить на $23,0$ – и получить те же $70\text{ \%}$: «пара же столько же». Пар тот же, а знаменатель другой, потому что другая температура. Ещё путают «на сколько процентных пунктов» и «на сколько процентов»: $23 : 70 \cdot 100 = 33\text{ \%}$ – это ответ на другой вопрос.

Ответ: днём $70\text{ \%}$, ночью $93\text{ \%}$; выросла на 23 процентных пункта.

Своя задача по программе сферы, модуль 2

Реши сам

2.3. Днём в теплице $30\text{ }^{\circ}\text{C}$, в каждом кубометре воздуха $18,2\text{ г}$ пара. Какова относительная влажность? Какой она станет, если воздух остынет до $25\text{ }^{\circ}\text{C}$, а пара останется столько же? Оба ответа – до целых процентов.

2.4. Пока ты неделю был в лагере, влажность земли в горшке с рассадой упала с $55\text{ \%}$ до $44\text{ \%}$. На сколько процентных пунктов она упала? На сколько процентов уменьшилась по сравнению с прежней?

Задача 2.3 Сити-ферма в подвале

временные интервалы через полночь, доля суток в процентах

На сити-ферме в подвале окон нет, весь свет салату дают фитолампы. Таймер включает их в $17:40$, а выключает в $7:25$ следующего утра. а) Сколько часов и минут в сутки горят лампы? б) Какую долю суток это составляет? Ответ – в процентах, округли до целых. в) Агроном решил, что салату нужно 16 часов света в сутки. Время включения оставили прежним. На какое время перевести выключение?

Дано: включение 17:40; выключение 7:25 следующего утра; нужно 16 ч света.

Найти: сколько горят лампы; доля суток, %; новое время выключения.

Решение

Шаг 1. Делим отрезок полночью. Лампы горят с вечера до утра, и посередине – полночь. Считаем две части отдельно.

Шаг 2. До полуночи. С 17:40 до 24:00: $24 \text{ ч } 00 \text{ мин} - 17 \text{ ч } 40 \text{ мин}$. Минут не хватает, занимаем час: $23 \text{ ч } 60 \text{ мин} - 17 \text{ ч } 40 \text{ мин} = 6 \text{ ч } 20 \text{ мин}$.

Шаг 3. После полуночи. С 0:00 до 7:25 – это 7 ч 25 мин.

Шаг 4. Всего. $6 \text{ ч } 20 \text{ мин} + 7 \text{ ч } 25 \text{ мин} = 13 \text{ ч } 45 \text{ мин}$.

Шаг 5. Доля суток. Переводим в часы: 45 мин – это $45:60 = 0,75 \text{ ч}$, всего 13,75 ч. Делим на 24: $13,75:24 \cdot 100 = 57,3 \approx 57 \%$.

Шаг 6. Новое выключение. Свет идёт без перерыва, поэтому к моменту включения просто прибавляем 16 ч: $17:40 + 16 \text{ ч} = 33:40$. Таких часов на циферблате нет – вычитаем сутки: $33:40 - 24:00 = 9:40$ утра. Проверим другим путём: не хватало $16 \text{ ч} - 13 \text{ ч } 45 \text{ мин} = 2 \text{ ч } 15 \text{ мин}$, и $7:25 + 2:15 = 9:40$.

полночь делит отрезок света на две части: $6 \text{ ч } 20 \text{ мин} + 7 \text{ ч } 25 \text{ мин} = 13 \text{ ч } 45 \text{ мин}$



Рисунок 2.11 – Сутки на сити-ферме: лампы горят через полночь

Больше половины суток под лампами – в подвале без окон салат по-другому не вырастить.

⚠ **Типичная ошибка.** Вычесть «из большего меньшее»: $17:40 - 7:25 = 10 \text{ ч } 15 \text{ мин}$. Это время, когда лампы **не горят**, – тёмная часть суток. Если сомневаешься, сложи свет и темноту: вместе должно выйти 24 ч.

Ответ: а) 13 ч 45 мин; б) 57 %; в) в 9:40.

Своя задача по программе сферы, модуль 2

Реши сам

2.5. Лампы для рассады включаются в 21:15 и выключаются в 6:45 утра. Сколько часов и минут они горят? Какая это доля суток в процентах? Округли до целых.

2.6. В конце июня солнце в Новосибирске встаёт около 4:48 и садится около 22:11. Растению по условию нужно 12 часов солнечного света в сутки, поэтому часть светлого времени теплицу закрывают плотной шторкой. Сколько часов и минут в светлое время шторка должна быть закрыта?

Задача 2.4 Сутки в теплице

суточный ход температуры, доля в процентах, интервалы

Датчик записывал температуру в теплице каждые 2 часа, начиная с полуночи: 0:00 – 16 °С, 2:00 – 15 °С, 4:00 – 14 °С, 6:00 – 15 °С, 8:00 – 19 °С, 10:00 – 24 °С, 12:00 – 29 °С, 14:00 – 31 °С, 16:00 – 28 °С, 18:00 – 23 °С, 20:00 – 19 °С, 22:00 – 17 °С. Норма для этих растений – от 18 до 28 °С включительно. а) Какой процент замеров вне нормы? Округли до целых. б) Сколько часов за сутки теплица была вне нормы, если каждое показание держится до следующего замера? в) Каков размах температуры?

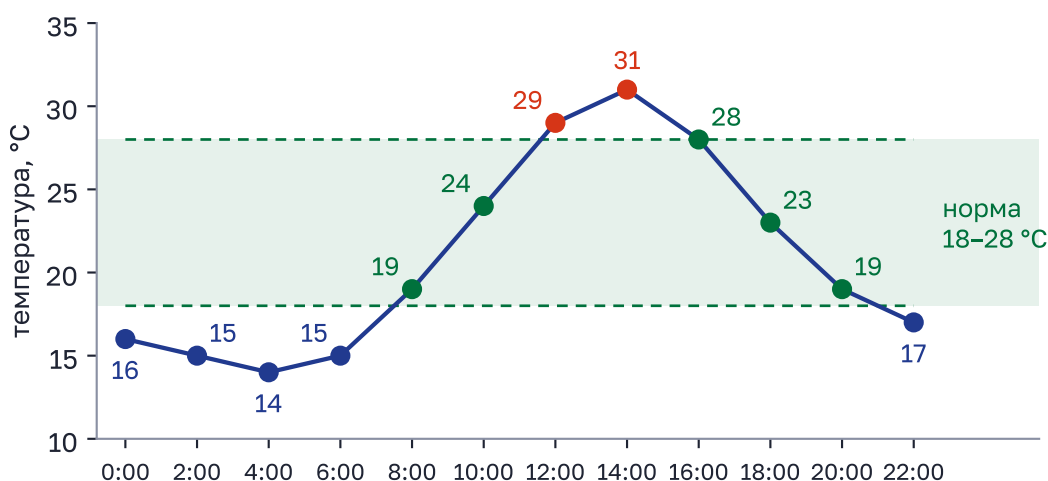


Рисунок 2.12 – Температура в теплице за сутки. Зелёная полоса – норма

Дано: 12 замеров через 2 ч; норма 18–28 °С, границы входят в норму.

Найти: процент замеров вне нормы; часы вне нормы; размах.

Решение

Шаг 1. Выше нормы. Выше 28 °С – только 29 и 31, это 2 замера. Замер в 16:00 показывает 28 °С – это ещё норма.

Шаг 2. Ниже нормы. Ниже 18 °С – 16, 15, 14, 15 и 17 °С, это 5 замеров, все ночью и ранним утром.

Шаг 3. Доля. Вне нормы $2 + 5 = 7$ замеров из 12: $7 : 12 \cdot 100 = 58,3 \approx 58 \%$.

Шаг 4. Часы. Каждое показание держится 2 часа: $7 \cdot 2 = 14$ ч.

Шаг 5. Размах. Самая высокая температура – 31 °С, самая низкая – 14 °С: $31 - 14 = 17$ °С.

По графику видно, откуда столько брака: ночью теплице не хватает обогрева, а в обед – проветривания. Самый холодный замер – в 4:00, перед рассветом, самый жаркий – в 14:00: те самые качели из раздела 2.5.

❗ **Типичная ошибка.** Засчитать 28 °С как выход за норму: «28 – это же край». В условии сказано «включительно», так что 28 – норма. Кто засчитал его в брак, получит 8 замеров из 12 и ответ 67 % вместо 58 %. Другая ошибка – ответить на вопрос б) числом 7: это замеры, а спрашивали часы.

Ответ: а) 58 %; б) 14 ч; в) 17 °С.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «Анализ температурных данных»

Реши сам

2.7. Замеры каждые 2 часа с полуночи: 17, 16, 16, 17, 20, 23, 26, 27, 27, 25, 21, 19 °С. Норма – от 18 до 27 °С включительно. Какой процент замеров в норме? Округли до целых.

2.8. Замеры каждые 3 часа с полуночи: 12, 11, 13, 20, 27, 30, 24, 16 °С. «Жарко» – это выше 26 °С. Каждое показание держится до следующего замера. Сколько часов в сутки было жарко? Какая это доля суток в процентах?

Задача 2.5 Какие светодиоды горят

пороги, условия «и»/«или», перевод единиц

На пульте умной теплицы у каждого устройства свой светодиод: горит – устройство включено. Автоматика работает по правилам:

Таблица 2.2

№	Устройство	Что проверяем	Включено, когда
1	форточка	температура	28 °С и выше
2	обогреватель	температура	ниже 16 °С
3	насос полива	влажность почвы	ниже 40 %
4	лампы досветки	свет и часы	ниже 5000 лк и время с 6:00 до 22:00
5	увлажнитель	влажность воздуха	ниже 55 %

В 14:30 датчики показывают: температура 28 °С, влажность воздуха 48 %, влажность почвы 40 %, освещённость 3,2 клк, углекислый газ 380 ppm. Какие светодиоды горят? Ответ – номера устройств по возрастанию без пробелов, например 235.

Дано: $t = 28\text{ °С}$; $\varphi = 48\text{ %}$; $W = 40\text{ %}$; освещённость 3,2 клк; CO_2 380 ppm; время 14:30.

Найти: номера включённых устройств.

Решение

Шаг 1. Приводим единицы. Порог света – в люксах, показание – в килолюксах: 3,2 клк = 3200 лк.

Шаг 2. Проверяем правила по одному.

- Форточка: $28 \geq 28$ – да, «28 и выше» включает саму границу. Горит.
- Обогреватель: 28 ниже 16? Нет.
- Насос: 40 ниже 40? Нет, 40 равно порогу, а нужно строго ниже. Не горит.
- Лампы: $3200 < 5000$ – да, и 14:30 лежит между 6:00 и 22:00 – да. Обе части «и» верны. Горит.
- Увлажнитель: $48 < 55$ – да. Горит.

Шаг 3. Записываем ответ. Горят светодиоды 1, 4 и 5 – строка 145.

Углекислый газ ни в одном правиле не участвует – это лишнее данное. В задачах такое встречается, пугаться не надо. Картина жизненная: жаркий пасмурный день, воздух сухой – проветриваем, досвечиваем, увлажняем.

 **Типичная ошибка.** Все три ловушки – на границах и единицах. Не включить форточку при 28°C («28 – это ещё не жарко»), включить насос при 40 % («40 – это уже мало») или сравнить 3,2 с 5000 и решить, что света полно. Каждая ошибка меняет строку ответа, а вводить её на отборе приходится за попытки.

Ответ: 145.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «Индикаторы в виде светодиодов»

Реши сам

2.9. Правила те же, что в задаче 2.5. В 23:10 датчики показывают: температура $15,5^\circ\text{C}$, влажность воздуха 82 %, влажность почвы 37 %, освещённость 20 лк. Какие светодиоды горят? Ответ – номера по возрастанию без пробелов.

2.10. К правилам задачи 2.5 добавили шестое: вентилятор (№ 6) включается, если влажность воздуха выше 85 % **или** углекислого газа больше 1500 ppm. В 7:40 датчики показывают: температура 17°C , влажность воздуха 79 %, влажность почвы 55 %, освещённость 0,9 клк, углекислый газ 0,16 %. Какие светодиоды горят? Ответ – номера по возрастанию без пробелов.

Задача 2.6^{*}. Увлажнитель для огурцов

относительная влажность, проценты, объём, время работы

Теплица – прямоугольный ящик длиной 5 м, шириной 4 м и высотой 2,5 м. В ней 25°C , влажность воздуха 40 %, а огурцам нужно 70 %. Увлажнитель превращает в пар 400 мл воды в час. Сколько граммов пара нужно добавить в воздух? Сколько минут должен проработать увлажнитель? Считай, что температура не меняется, весь пар остаётся в воздухе, 1 мл воды весит 1 г. Время округли до целых минут вверх. Наибольшее количество пара возьми из таблицы в разделе 2.3.

Дано: $5 \times 4 \times 2,5$ м; $t = 25$ °С; $\varphi_1 = 40$ %; $\varphi_2 = 70$ %; $a_{\max}(25$ °С) = 23,0 г/м³; увлажнитель 400 мл/ч.

Найти: масса пара; время работы, мин.

Решение

Шаг 1. План. Узнаем, сколько пара в кубометре сейчас и сколько должно стать. Разница – добавка на один кубометр. Умножим на объём – получим добавку на всю теплицу. Потом поделим на скорость увлажнителя.

Шаг 2. Пар сейчас. $a_1 = 23,0 \cdot 40 : 100 = 9,2$ г в кубометре.

Шаг 3. Пар нужен. $a_2 = 23,0 \cdot 70 : 100 = 16,1$ г в кубометре.

Шаг 4. Добавка на кубометр. $16,1 - 9,2 = 6,9$ г. Температура не меняется, поэтому можно короче: $70 - 40 = 30$ процентных пунктов от 23,0 г, $23,0 \cdot 0,3 = 6,9$ г.

Шаг 5. Объём теплицы. $V = 5 \cdot 4 \cdot 2,5 = 50$ м³.

Шаг 6. Добавка на всю теплицу. $6,9 \cdot 50 = 345$ г. Это 345 мл воды.

Шаг 7. Время. Увлажнитель испаряет 400 мл за час. $345 : 400 = 0,8625$ ч. В минутах: $0,8625 \cdot 60 = 51,75$ мин. Округляем вверх, чтобы влажность точно дошла до нормы: 52 мин.

Треть литра воды на всю теплицу – примерно полторы кружки. Выглядит мало, но пар лёгкий: целый кубометр воздуха при 25 °С вмещает всего 23 г. В настоящей теплице увлажнитель работал бы дольше: пар уходит в щели, а растения сами добавляют влагу, так что это грубая оценка.

 **Типичная ошибка.** Ответить на первый вопрос числом 6,9 г – забыть умножить на объём. Или взять 30 % от объёма теплицы, $50 \cdot 0,3 = 15$, – проценты влажности считаются от граммов пара, а не от кубометров. В конце часто пишут «0,86 мин»: это часы, их ещё надо умножить на 60.

Ответ: 345 г пара; 52 мин.

Своя задача по программе сферы, модуль 2

Реши сам

2.11. Комната-питомник объёмом 30 м³, в ней 20 °С и влажность 40 %. Сколько граммов пара надо добавить, чтобы влажность стала 60 %? Температура не меняется. Округли до целых.

2.12*.* Теплица 4 м × 2,5 м × 2 м, в ней 30 °С и влажность 50 %. Увлажнитель расходует 0,25 л воды в час, и вся она превращается в пар. Сколько минут ему нужно работать, чтобы влажность поднялась до 75 %? Температура не меняется, пар не уходит. Округли вверх до целых минут.

Задача 2.7* *. Форточка, которая не хлопает

два порога (гистерезис), интервалы времени

Форточка теплицы открывается, когда температура 27°C и выше, и закрывается, когда 23°C и ниже. При температуре между ними форточка остаётся такой, какой была. В 10:00 форточка закрыта. Датчик делает замер каждые 10 минут, с 10:00 до 12:10: 22, 24, 26, 24, 26, 28, 27, 25, 24, 26, 23, 25, 26, 22 $^{\circ}\text{C}$. Каждое показание держится до следующего замера, последнее – до 12:20.

а) Сколько раз открылась форточка? б) Сколько минут она была открыта? в) Сколько раз она открылась бы, если бы порог был один: открыть при 25°C и выше, закрыть ниже 25°C ?

Дано: открыть при $t \geq 27^{\circ}\text{C}$; закрыть при $t \leq 23^{\circ}\text{C}$; вначале закрыта; 14 замеров через 10 мин.

Найти: число открываний; минуты открытой форточки; число открываний с одним порогом 25°C .

Решение

Шаг 1. Идём по замерам с двумя порогами. Состояние записываем после каждого замера.

- 10:00 – 22, 10:10 – 24, 10:20 – 26, 10:30 – 24, 10:40 – 26: ни разу не 27 и выше, форточка закрыта.
- 10:50 – 28: $28 \geq 27$, форточка **открывается** (первый раз).
- 11:00 – 27, 11:10 – 25, 11:20 – 24, 11:30 – 26: всё выше 23 , закрывать не за что, форточка открыта.
- 11:40 – 23: $23 \leq 23$, форточка **закрывается**.
- 11:50 – 25, 12:00 – 26, 12:10 – 22: до 27 ни разу не дошло, форточка закрыта.

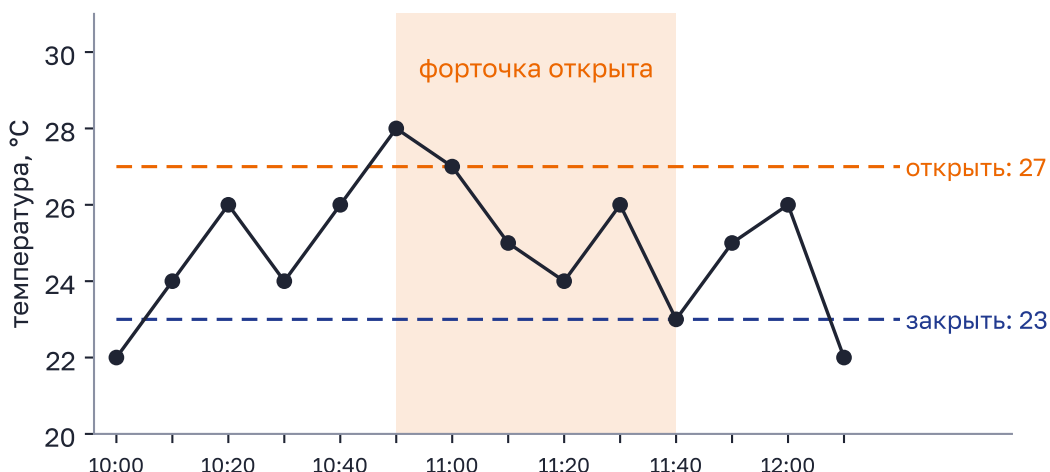


Рисунок 2.13 – Температура и форточка с двумя порогами. Оранжевая полоса – форточка открыта

Шаг 2. Считаем. Форточка открылась один раз. Открыта она с 10:50 до 11:40:

11 ч 40 мин – 10 ч 50 мин = 50 мин.

Шаг 3. Один порог 25°C . Теперь каждый замер решает сам за себя: 25 и выше – открыта, ниже – закрыта. Ряд состояний: 22 – закр., 24 – закр., 26 – **откр.**, 24 – закр., 26 – **откр.**, 28 – откр., 27 – откр., 25 – откр., 24 – закр., 26 – **откр.**, 23 – закр., 25 – **откр.**, 26 – откр., 22 – закр. Открываний – переходов из «закрыта» в «открыта» – четыре.

Четыре раза за два часа против одного: теперь видно, зачем нужны два порога. В настоящей теплице датчик опрашивают хоть каждую секунду, и с одним порогом мотор форточки щёлкал бы без остановки.

 **Типичная ошибка.** Решать каждый замер отдельно, как будто порог один: «26 – это между порогами, значит, закрыта». С двумя порогами между ними ничего не решается – форточка остаётся как была, а была она открыта. Ещё считают открываниями все замеры, когда форточка открыта (при одном пороге таких восемь), хотя открываний – только переходы из «закрыта» в «открыта».

Ответ: а) 1 раз; б) 50 мин; в) 4 раза.

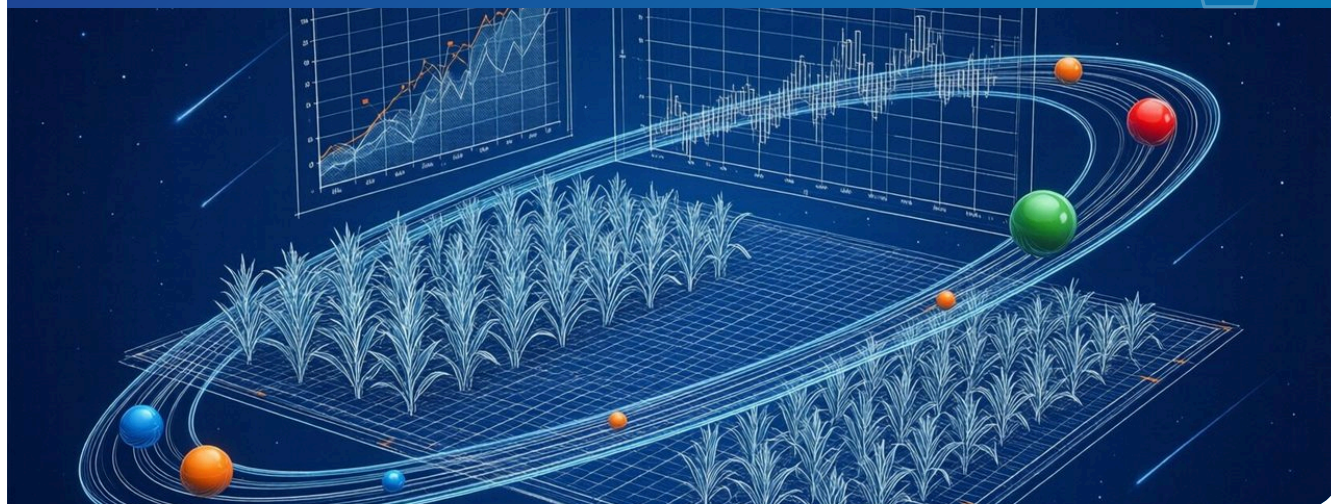
Своя задача по программе сферы, модуль 2

Реши сам

2.13. Обогреватель включается, когда температура 15°C и ниже, и выключается, когда 18°C и выше; между ними – как был. В 18:00 он выключен. Замеры каждые 15 минут с 18:00 до 21:15: 19, 17, 16, 15, 16, 17, 18, 17, 16, 15, 14, 16, 18, 19 $^{\circ}\text{C}$. Каждое показание держится до следующего замера. Сколько раз включался обогреватель и сколько минут он работал всего?

2.14*.* Вентилятор включается, когда влажность воздуха 85 % и выше, и выключается, когда 75 % и ниже; между ними – как был. В 13:00 он выключен. Замеры каждые 5 минут с 13:00 до 13:55: 70, 78, 84, 86, 88, 83, 79, 76, 75, 80, 85, 82 %. Последнее показание держится до 14:00. Сколько минут за этот час работал вентилятор? Какой это процент часа? Округли до целых.

Графики и таблицы агронома



Датчик в теплице меряет температуру круглые сутки, и к утру у него набираются сотни чисел. Столбиком их никто не читает: агроном смотрит на график и за пять секунд видит, когда было жарко, когда холодно и пора ли открывать форточку. В этой главе научимся читать графики, считать среднее и размах, строить график по таблице и заставлять электронную таблицу считать за нас. На отборе такие задачи выглядят так: график или таблица, короткий вопрос, в ответ – число или номера верных утверждений.

3.1. Как читать график

НА ПАЛЬЦАХ

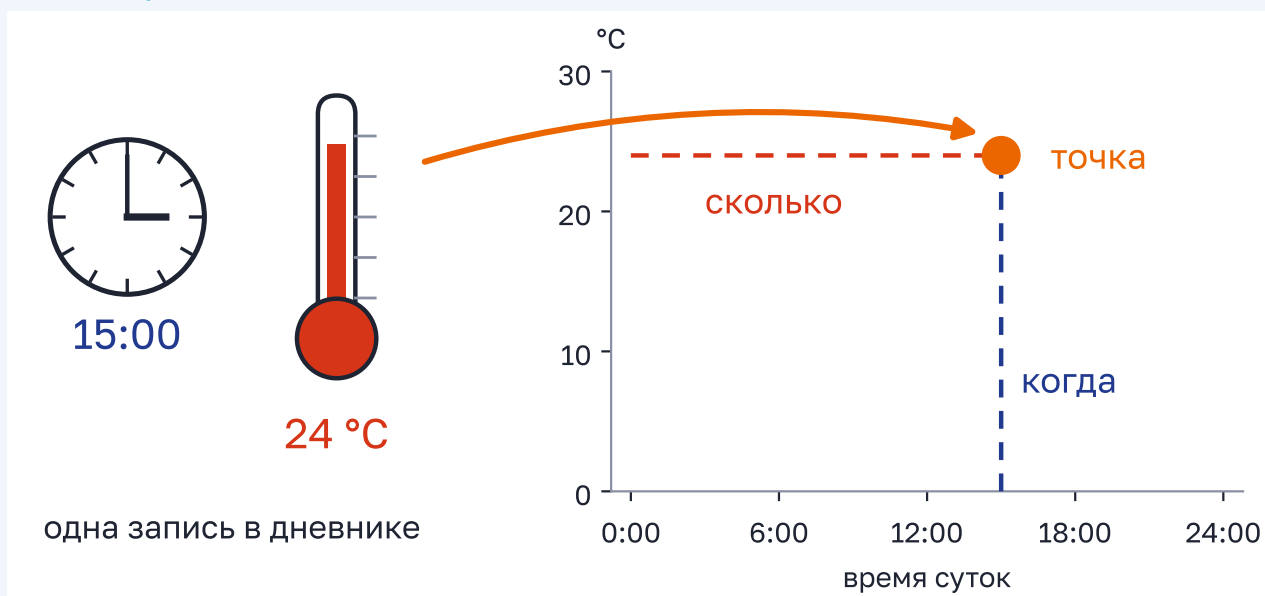


Рисунок 3.1 – Запись «в 15:00 было 24 °C» превращается в точку на графике

Каждая точка графика – одна запись в дневнике погоды. Вправо – когда измерили, вверх – сколько намерили.

График – рисунок, на котором числа превращены в точки. У графика две оси. По горизонтальной откладывают, **когда** был сделан замер, по вертикальной – **сколько** показал прибор. Соседние точки соединяют линией, и по её форме сразу видно, как менялась величина.

Читать график начинают не с линии, а с подписей. Сначала посмотри, что отложено по осям и в каких единицах: градусы, проценты, часы или минуты. Потом узнай, сколько стоит одна клетка сетки – её называют **ценой деления**. Подписи стоят не у каждой линии сетки, поэтому цену считают так: берут две соседние подписи, вычитают меньшую из большей и делят на число клеток между ними.

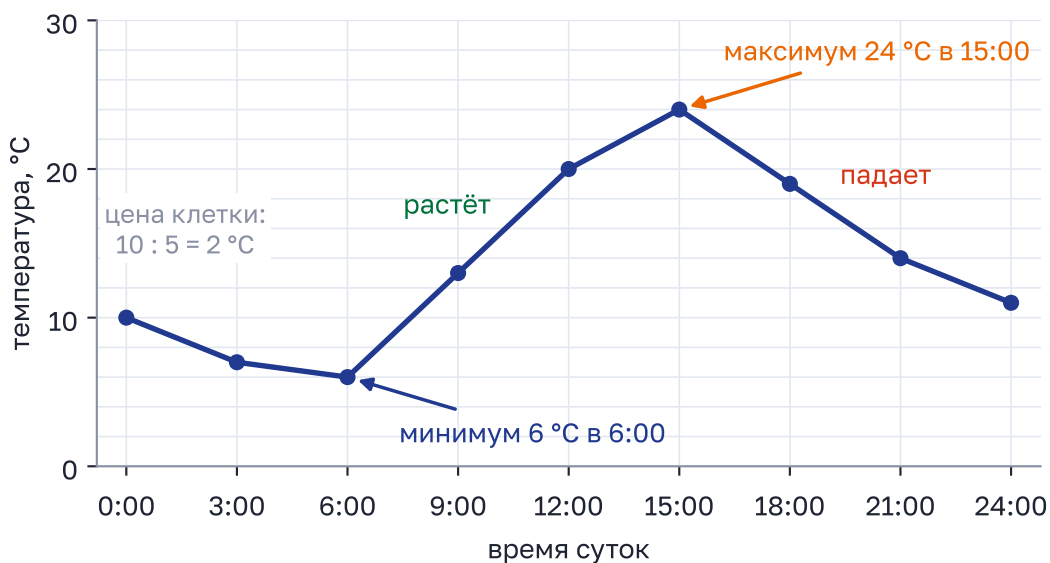


Рисунок 3.2 – Температура воздуха за весенние сутки на школьной метеостанции, замер раз в 3 часа

На этом графике по вертикали подписаны 0, 10, 20 и 30 °С. Между 0 и 10 – пять клеток, значит, одна клетка стоит $10 : 5 = 2$ °С. Точка на второй линии над подписью 10 – это $10 + 2 \cdot 2 = 14$ °С. Точка посередине клетки между 6 и 8 – это 7 °С.

Дальше следим за линией. Слева направо она поднимается – величина растёт, опускается – падает, идёт горизонтально – не меняется. Чем круче участок, тем быстрее идёт изменение. С 6:00 до 9:00 потеплело на $13 - 6 = 7$ °С, а с 0:00 до 3:00 похолодало всего на $10 - 7 = 3$ °С: первый участок крутой, второй пологий. Скорость изменения считают как скорость машины: изменение делят на время. С 6:00 до 12:00 температура выросла на $20 - 6 = 14$ °С за 6 часов, то есть в среднем на $14 : 6 \approx 2,3$ °С в час.

Не забывай, откуда взялась линия. Метеостанция мерила раз в три часа, а всё, что между точками, дорисовано. Какой была температура в 7:30, прибор не знает. Линия подсказывает разумную оценку: если в 6:00 было 6 °С, а в 9:00 – 13 °С, то в 7:30 было где-то около 9–10 °С. Если данные идут с датчика каждую минуту, точки сливаются, и линия почти не врёт.

Строить такие картинки руками нужно не всегда. В Tinkercad у окна монитора порта есть кнопка, которая рисует показания датчика линией прямо во время работы схемы. На вебинаре сферы так смотрели, как меняется сигнал фоторезистора, когда двигаешь ползунок освещённости. В Arduino IDE тот же инструмент есть у настоящей платы и называется плоттером (Serial Plotter).

Запомни. Порядок чтения: оси и единицы → цена деления → точки → форма линии.

Точки – настоящие замеры, линия между ними – оценка. Круче участок – быстрее меняется величина.

3.2. Максимум, минимум, среднее и размах

НА ПАЛЬЦАХ



Рисунок 3.3 – Столбики из кубиков до и после того, как лишние кубики переложили в низкие столбики

Переложи лишние кубики из высоких столбиков в низкие, пока все не станут одной высоты. Эта общая высота и называется средним.

Четыре числа описывают любой набор замеров коротко.

- **Максимум** – самое большое значение. На графике это самая высокая точка.
- **Минимум** – самое маленькое значение, самая низкая точка.
- **Размах** – максимум минус минимум. Он говорит, насколько сильно величина «гуляла». На уроках географии этот же промежуток между самой высокой и самой низкой температурой называют амплитудой. В главе 7 про сигналы слово «амплитуда» будет значить другое – половину размаха, так что не перепутай.
- **Среднее арифметическое** – сумма всех значений, делённая на их количество. Буквами: если замеров n , а их сумма S , то среднее равно $S : n$.

Посчитаем для графика из раздела 3.1. За сутки там восемь замеров: в 0:00, 3:00, 6:00 и дальше до 21:00. Точка в 24:00 – это уже полночь следующих суток, её не берём. Многие метеостанции так и считают среднесуточную температуру – по восьми замерам раз в три часа.

$$\text{среднее} = \frac{10 + 7 + 6 + 13 + 20 + 24 + 19 + 14}{8} = \frac{113}{8} = 14,125 \approx 14,1 \text{ }^{\circ}\text{C}.$$

Максимум – 24 °С, минимум – 6 °С, размах $24 - 6 = 18 \text{ }^{\circ}\text{C}$.

Среднее 14,1 °С не совпало ни с одним замером, и в этом нет ничего странного: средний рост в классе тоже может не совпасть ни с чьим ростом. А вот выйти за пределы от минимума до максимума среднее не может. Получилось больше максимума – ищи ошибку в сложении.

Числа с минусом. В мае под Новосибирском ночью запросто бывает –3 °С, и такие числа попадают в сумму. Складывая их как долги: было 5 рублей, отдал долг 3 рубля – осталось 2. Так же $5 + (-3) = 2$. Пример: замеры –4, –1, +2, +3 °С. Сумма $-4 - 1 + 2 + 3 = 0$, среднее $0 : 4 = 0 \text{ }^{\circ}\text{C}$. Размах $3 - (-4) = 3 + 4 = 7 \text{ }^{\circ}\text{C}$. Проверь по термометру: от –4 до 0 – четыре деления, от 0 до +3 – ещё три, всего семь.

Один сломанный замер. Иногда датчик выдаёт чепуху: отошёл провод или на корпус упал прямой солнечный луч. Пусть вместо последнего замера 14 °С в таблицу попало 85 °С. Среднее станет $(113 - 14 + 85) : 8 = 184 : 8 = 23 \text{ }^{\circ}\text{C}$ – на девять градусов выше правды из-за одного числа. Поэтому, прежде чем считать, пробеги числа глазами. Как отсеивать такие скачки автоматически, разберём в главе 7.

Запомни. Среднее = сумма : количество. Размах = максимум – минимум. Среднее всегда лежит между минимумом и максимумом. С минусами считай внимательно: от –3 до +13 размах 16 градусов, а вовсе не 10.

3.3. Строим график по таблице

НА ПАЛЬЦАХ

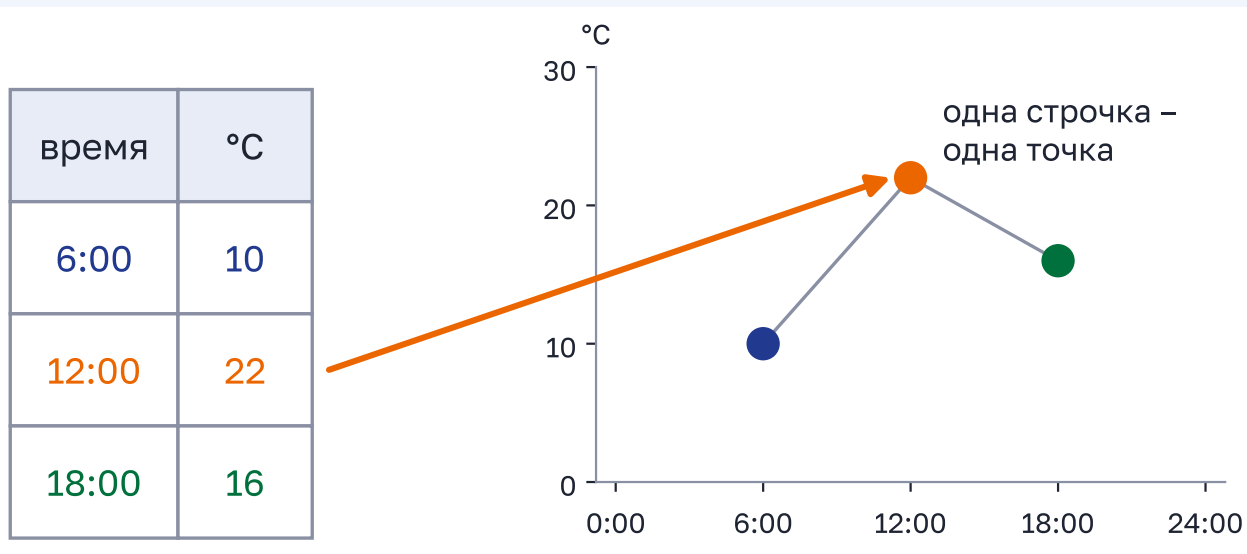


Рисунок 3.4 – Три строчки таблицы и три точки графика того же цвета

Одна строчка таблицы – одна точка. Время отмеряем вправо, температуру – вверх.

Построить график по таблице можно на бумаге в клетку, и порядок действий всегда один.

Шаг 1. Что по какой оси. Время – по горизонтали, измеренную величину – по вертикали. Так принято, и так быстрее читается.

Шаг 2. Масштаб. Посмотри на самое большое число в таблице и прикинь, сколько клеток у тебя есть по высоте. Пусть самое большое показание 820, а места – 10 клеток. Клетка по 50 не подойдет: $10 \cdot 50 = 500$, и половина точек улетит за край листа. Клетка по 100 даёт $10 \cdot 100 = 1000$ – всё влезает. Цену клетки бери круглую: 1, 2, 5, 10, 20, 50, 100. Считать по клеткам «по 37» мучительно.

Шаг 3. Подписи. Подпиши оси, единицы и несколько круглых делений. График без подписей – картинка, по которой ничего нельзя проверить.

Шаг 4. Точки. Для каждой строчки находишь время на горизонтальной оси, поднимаешься вверх до нужного значения и ставишь точку.

Шаг 5. Линия. Соединяешь точки по порядку времени – слева направо.

На горизонтальной оси время идёт равномерно. Если замеры делали в 8:00, 10:00 и 16:00, отрезок от 10 до 16 должен быть втрое длиннее отрезка от 8 до 10. Поставишь три точки на равных расстояниях – график сойдёт: покажет, что к 16:00 всё поменялось быстро, хотя прошло шесть часов.

Линия или столбики. Линией рисуют то, что меняется непрерывно: температуру, влажность, освещённость. У такой величины есть значение в любой момент, даже между замерами. Столбиками рисуют отдельные порции: сколько литров вылили в понедельник, во вторник, в среду; сколько миллиметров осадков выпало за каждые сутки; какой урожай дала каждая

рядка. Верхушки столбиков линией не соединяют. Между понедельником и вторником не было «половины полива».

Запомни. Строчка таблицы – точка графика. Масштаб выбирай по самому большому числу. Равное время – равные отрезки по оси. Непрерывно меняется – рисуй линию, отдельными порциями – столбики.

3.4. Электронная таблица считает за тебя

НА ПАЛЬЦАХ

B6 =СРЗНАЧ(B2:B5)			
	A	B	C
1	время	°C	
2	6:00	18	
3	9:00	22	
4	12:00	26	
5	15:00	24	
6		22,5	

адрес – как в «Морском бое»:
буква столбца + номер строки

B2:B5 – клетки
от B2 до B5

в B6 – их среднее,
таблица посчитала сама

Рисунок 3.5 – Клетки таблицы с адресами. В клетке B6 формула, которая считает среднее по клеткам B2:B5

У каждой клетки есть адрес: буква столбца и номер строки. Формула берёт числа по адресам и считает сама.

Электронная таблица – программа, где числа лежат в клетках, а клетки умеют считать. Это Excel, LibreOffice Calc или онлайн-таблицы в браузере. Таблицы есть в официальном списке программ для финала сферы «Агротех» – рядом с симулятором и Arduino IDE. Столбцы в них обозначены буквами A, B, C..., строки – числами 1, 2, 3... Адрес клетки складывается из буквы и числа: B3 – столбец B, строка 3, как клетка в «Морском бое».

Формула начинается со знака `=`. Напишешь в клетку `=B2+B3` и нажмёшь Enter – в клетке появится сумма, а сама формула останется видна в строке формул над таблицей. Поменяешь число в B2 – сумма пересчитается сама.

Диапазон – несколько клеток подряд, его записывают через двоеточие: B2:B9 – клетки от B2 до B9. Сколько в нём клеток? Не $9 - 2 = 7$, а $9 - 2 + 1 = 8$: считаются обе крайние. На этом спотыкаются чаще, чем на самих формулах.

Таблица 3.1

Формула	Что считает	В английской версии
=СУММ(B2:B9)	сумму	SUM
=СРЗНАЧ(B2:B9)	среднее	AVERAGE
=МАКС(B2:B9)	максимум	MAX
=МИН(B2:B9)	минимум	MIN
=СЧЁТЕСЛИ(B2:B9;">25")	сколько чисел больше 25	COUNTIF
=ОКРУГЛ(E2;1)	округляет до десятых	ROUND

Условие в СЧЁТЕСЛИ пишут в кавычках. ">25" – строго больше, саму 25 не считает. " ≥ 25 " – 25 и больше. В русской версии числа внутри формулы разделяют точкой с запятой, в английской – запятой. Функции можно вкладывать друг в друга: =ОКРУГЛ(СРЗНАЧ(B2:B9);1) сначала найдёт среднее, потом округлит его до десятых. Размах – одной формулой: =МАКС(B2:B9)-МИН(B2:B9) .

Протягивание. Пусть в столбце В – температура, и ты хочешь знать, на сколько она менялась от замера к замеру. Пишешь в D3 формулу =B3-B2 и тянешь клетку за правый нижний уголок вниз. В D4 окажется =B4-B3, в D5 – =B5-B4 : адреса съезжают вниз вместе с клеткой. Одна формула – и весь столбец разностей готов.

График из таблицы. Выдели столбец времени и столбец значений, открой вкладку «Вставка», выбери диаграмму типа «График». Программа сама подберёт масштаб, но подписи осей всё равно проверь.

То же самое можно посчитать короткой программой на Python – этот язык есть в программе сферы:

```
t = [10, 7, 6, 13, 20, 24, 19, 14] # замеры раз в 3 часа, °C

print(sum(t) / len(t)) # среднее
print(max(t), min(t)) # максимум и минимум
print(max(t) - min(t)) # размах

count = 0 # сколько замеров теплее 15 °C
for x in t:
    if x > 15:
        count = count + 1
print(count)
```

Разберём построчно. В первой строке список `t` – восемь замеров из раздела 3.1 в квадратных скобках. После знака `#` идёт комментарий: его пишут для людей, программа его пропускает. `sum(t)` складывает все числа списка, `len(t)` считает, сколько их, – делим одно на другое и получаем среднее 14.125 (Python пишет дроби через точку). `max` и `min` находят 24 и 6, их разность – размах 18. Последний кусок заводит счётчик `count`, равный нулю, перебирает замеры по одному и прибавляет единицу за каждый теплее 15 °C. Таких три: 20, 24 и 19.

Запомни. Формула начинается с $=$. В диапазоне B2:B9 восемь клеток – обе крайние считаются. При протягивании вниз адреса в формуле сдвигаются сами. ">25" саму 25 не считает, " ≥ 25 " – считает.

3.5. Порог: когда пора действовать

НА ПАЛЬЦАХ



Рисунок 3.6 – График температуры и красная линия порога 27 °C. Кусок графика над линией закрашен

Порог – горизонтальная линия. Пока график над ней, форточка открыта. Где линия графика её пересекла, там автоматика и сработала.

График нужен агроному, чтобы действовать. В умной теплице решения записаны правилами с порогами: «теплее 27 °C – открыть форточку», «почва суше 35 % – включить полив». **Порог** – то самое число из правила, при котором пора что-то делать. На графике его рисуют горизонтальной линией. Участки графика температуры выше неё – время, когда форточка должна быть открыта. Участки графика влажности ниже своей линии – время, когда растениям не хватает воды.

Обычно порог попадает куда-то между двумя замерами. Тогда считают, что между точками величина менялась равномерно, по одинаковому числу градусов в час. Пример с картинки: в 9:00 в теплице 22 °C, в 11:00 – 30 °C, форточку открывают при 27 °C. За 2 часа потеплело на $30 - 22 = 8$ °C, то есть на $8 : 2 = 4$ °C в час. От 22 до 27 не хватало $27 - 22 = 5$ °C, на это ушло $5 : 4 = 1,25$ ч. Четверть часа – 15 минут, значит, 1,25 ч – это 1 ч 15 мин, и форточку пора было открывать около 10:15.

Остывание считают так же, только температура теперь убывает. В 15:00 было 29 °C, в 17:00 – 24 °C: за 2 часа минус 5 градусов, по 2,5 °C в час. До порога 27 °C спускаться $29 - 27 = 2$ °C,

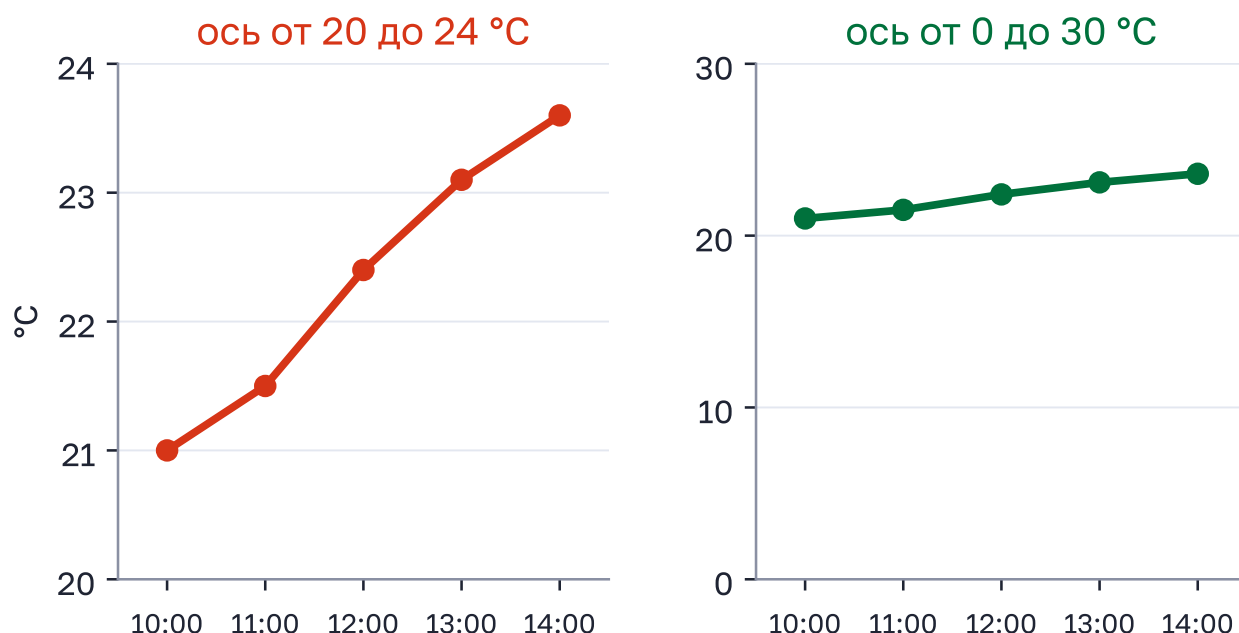
это $2:2,5 = 0,8$ ч. Вот тут не спеши: в часе 60 минут, поэтому $0,8$ часа – это $0,8 \cdot 60 = 48$ мин (рука так и тянется написать «80»). Форточку закрыли около 15:48.

Запомни. Порог на графике – горизонтальная линия. Момент, когда её пересекли, ищут по двум соседним замерам: сначала «сколько градусов в час», потом «сколько часов до порога». Доли часа переводят в минуты умножением на 60.

Совет. Проведи карандашом линию порога прямо по графику и выпиши замеры в маленькую таблицу «время – значение». Дальше решаешь уже по числам, и глаз не обманет.

3.6. Графики-обманщики

НА ПАЛЬЦАХ



числа одни и те же – сначала смотри на подписи оси

Рисунок 3.7 – Одни и те же пять замеров на двух графиках с разной вертикальной осью

Слева кажется, что температура взлетела, справа – что почти не менялась. А числа одни и те же: с 21 до 23,6 °C, на 2,6 градуса.

График может обмануть – бывает, что нечаянно, а бывает, что нарочно, в рекламе. Приёмы у обманщиков одни и те же, и их полезно знать в лицо.

- **Ось начинается не с нуля.** На картинке выше левая ось идёт от 20 до 24 °C, и подъём на 2,6 градуса выглядит взлётом. Сравнивай числа на осях, а не крутизну на глаз.
- **У двух графиков разный масштаб.** Когда сравниваешь две теплицы, сначала сравни их оси. Линия, прошедшая «от пола до потолка» на графике с осью 19–29 °C, может озна-

чать меньшие перемены, чем скромный горбик на оси 0–40 °С.

- **Неравные промежутки времени.** Если подписи 8:00, 10:00, 16:00 стоят на равных расстояниях, наклон врёт.
- **Нет подписей или единиц.** Число 40 – это градусы, проценты влажности почвы или минуты? Без подписи вывод делать нельзя.
- **Среднее прячет крайности.** Средняя температура за сутки 24 °С – вроде бы идеально для томатов. Но если в полдень было 36 °С, пыльца у томатов могла пострадать: выше 30–32 °С завязей становится меньше, а при 35 °С и выше жара совсем опасна, даже когда листья выглядят бодро. Для решения «открыть – закрыть» смотри на максимум, минимум и порог; среднее годится для общей картины.
- **Редкие замеры.** Если мерить раз в шесть часов – в 0:00, 6:00, 12:00 и 18:00, – можно пропустить пик жары в 14:00.

Запомни. Два графика рядом – сначала сравни оси, потом линии. Среднее за сутки не скажет, была ли жара в полдень: для решений в теплице смотри на крайние значения.

3.7. Разбираем задачи

Задача 3.1 Сутки на метеостанции

разминка · чтение графика, цена деления, максимум, минимум, размах

Школьная метеостанция записала температуру воздуха за июльские сутки (рисунок). Ответь: а) какой была самая высокая температура и во сколько; б) какой была самая низкая температура и во сколько; в) каков размах температуры за сутки; г) сколько часов подряд температура держалась на отметке 20 °С и выше. Ответы – целые числа.

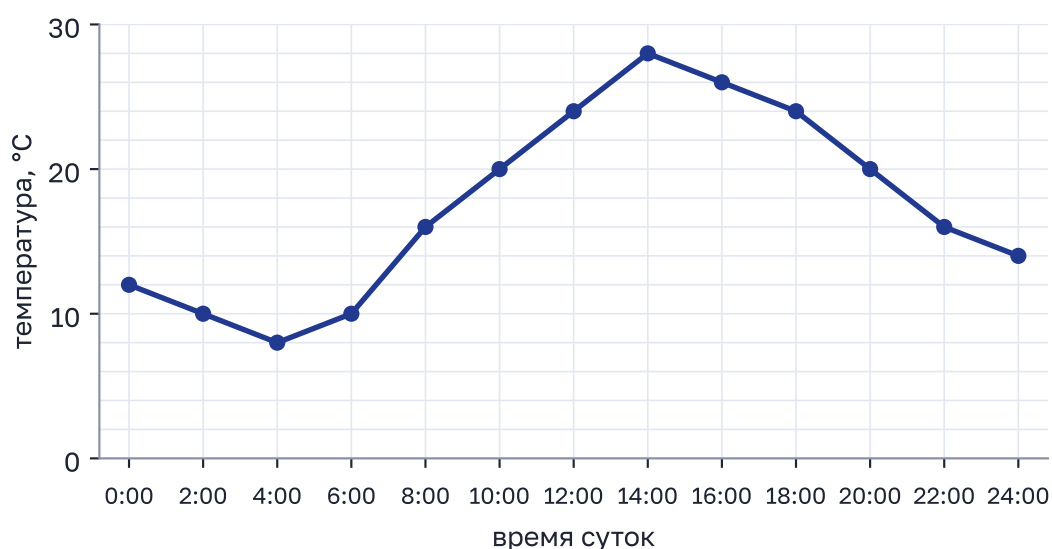


Рисунок 3.8 – Температура воздуха за июльские сутки, замер раз в 2 часа

Дано: график температуры, замеры каждые 2 часа.

Найти: максимум, минимум, размах, число часов при 20 °С и выше.

Решение

Шаг 1. Оси и цена деления. По горизонтали – время суток, подписи через каждые 2 часа. По вертикали – температура, подписаны 0, 10, 20 и 30 °С. Между 0 и 10 – пять клеток, одна клетка стоит $10:5 = 2$ °С.

Шаг 2. Максимум. Самая высокая точка – над 14:00. Она на четвёртой линии выше подписи 20: $20 + 4 \cdot 2 = 28$ °С.

Шаг 3. Минимум. Самая низкая точка – над 4:00, на четвёртой линии выше нуля: $4 \cdot 2 = 8$ °С. Точки в 0:00 и 24:00 (12 и 14 °С) выше, это не минимум.

Шаг 4. Размах. $28 - 8 = 20$ °С.

Шаг 5. Сколько часов при 20 °С и выше. Ищем точки на линии 20 и над ней. Первая такая – в 10:00, она лежит на линии 20. Дальше линия идёт выше: 24, 28, 26, 24 °С. В 20:00 температура снова опустилась до 20 °С, а в 22:00 уже 16 °С. Значит, 20 °С и выше было с 10:00 до 20:00: $20 - 10 = 10$ ч.

Для сибирского июля картина знакомая: ночью у костра без свитера зябко, днём можно купаться в Обском море.

 **Типичная ошибка.** Принять за минимум первую точку слева (12 °С в 0:00) и получить размах $28 - 12 = 16$ °С. Самая левая точка – не обязательно самая низкая, смотри на всю линию. В пункте г) часто считают точки вместо часов: на линии 20 и выше лежат шесть точек, и рука пишет «6». Но точки стоят через 2 часа, а считать нужно время от первой до последней: 10 часов.

Ответ: а) 28 °С в 14:00; б) 8 °С в 4:00; в) 20 °С; г) 10 ч.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «У природы есть плохая погода»

Реши сам

3.1. По графику задачи 3.1: на сколько градусов потеплело с 6:00 до 14:00? На сколько градусов в час в среднем росла температура в это время?

3.2. По тому же графику найди, между какими соседними замерами температура росла быстрее всего. Сколько градусов в час она прибавляла на этом участке?

Задача 3.2 Майский заморозок

среднее арифметическое, размах, отрицательные числа

Метеостанция в саду под Новосибирском записывает температуру воздуха раз в 3 часа. Вот её записи за 24 мая.

Таблица 3.2

Время	Температура, °С
-------	-----------------

Время	Температура, °C
0:00	+2
3:00	-1
6:00	-3
9:00	+4
12:00	+11
15:00	+13
18:00	+9
21:00	+6

Найди: а) среднесуточную температуру (среднее всех восьми замеров), округли до десятых; б) размах температуры; в) сколько замеров показали мороз, то есть температуру ниже 0 °C.

Дано: 8 замеров: +2, -1, -3, +4, +11, +13, +9, +6 °C.

Найти: среднее, размах, число замеров ниже 0 °C.

Решение

Шаг 1. Сумма. Складываем по одному числу, минусы не теряем: $2 + (-1) = 1$; $1 + (-3) = -2$; $-2 + 4 = 2$; $2 + 11 = 13$; $13 + 13 = 26$; $26 + 9 = 35$; $35 + 6 = 41$. Сумма - 41.

Шаг 2. Среднее. Замеров восемь: $41 : 8 = 5,125$ °C. До десятых: после запятой оставляем 1, следующая цифра 2 - меньше 5, поэтому округляем вниз: 5,1 °C.

Шаг 3. Размах. Максимум +13 °C в 15:00, минимум -3 °C в 6:00. $13 - (-3) = 13 + 3 = 16$ °C. По термометру: от -3 до 0 - три градуса, от 0 до +13 - ещё тринадцать, всего 16.

Шаг 4. Мороз. Ниже нуля были -1 °C в 3:00 и -3 °C в 6:00 - два замера.

Ночь была морозной, а среднее всё равно вышло плюсовым: дневного тепла набралось больше, чем ночного холода. Для сада важнее другое - рассаду, которую уже высадили в грунт, в эту ночь надо было укрыть. В Сибири заморозки в конце мая - дело привычное, поэтому огородники и ждут начала июня.

 **Типичная ошибка.** Потерять минусы и сложить $2 + 1 + 3 + 4 + \dots = 49$. Тогда среднее выйдет $49 : 8 \approx 6,1$ °C, а размах $13 - 3 = 10$ °C. Ещё бывает, что 5,125 округляют до 5,2 «потому что там пятёрка» - но решает только цифра сразу за десятичными, а это 2.

Ответ: а) 5,1 °C; б) 16 °C; в) 2 замера.

Своя задача по программе сферы, модуль 8

Реши сам

3.3. Датчик влажности почвы в теплице каждое утро в 8:00 показывал: 42, 38, 35, 47, 44, 40 %. Найди среднюю влажность почвы за эти шесть дней и размах.

3.4. Средняя дневная температура за пять дней по данным метеостанции – 14 °С. За первые четыре дня записано 12, 17, 9 и 15 °С, а запись за пятый день стёрлась. Какой была температура в пятый день?

Задача 3.3 Формулы метеостанции

электронная таблица, СРЗНАЧ, МАКС, МИН, СЧЁТЕСЛИ, протягивание формулы

Данные метеостанции за день перенесли в электронную таблицу.

Таблица 3.3

	А	В	С
1	время	температура, °С	влажность, %
2	8:00	17	78
3	10:00	21	70
4	12:00	26	58
5	14:00	29	51
6	16:00	27	55
7	18:00	25	64
8	20:00	19	73

В клетки справа записали формулы:

- E2: =ОКРУГЛ(СРЗНАЧ(В2:В8);1)
- E3: =МАКС(С2:С8)-МИН(С2:С8)
- E4: =СЧЁТЕСЛИ(В2:В8;">25")
- D3: =В3-В2, а потом эту клетку протянули вниз до D8.

Какие числа появятся в клетках E2, E3, E4 и D6?

Дано: таблица, строки 2–8; формулы в E2, E3, E4, D3.

Найти: значения E2, E3, E4, D6.

Решение

Шаг 1. Сколько клеток в диапазоне. В строке 1 – заголовки, числа идут со строки 2 по строку 8. Клеток в В2:В8: $8 - 2 + 1 = 7$.

Шаг 2. E2 – среднее температуры. Сумма: $17 + 21 + 26 + 29 + 27 + 25 + 19 = 164$. Среднее: $164 : 7 = 23,428...$ Функция ОКРУГЛ с числом 1 оставляет одну цифру после запятой; следующая цифра 2, поэтому получается 23,4.

Шаг 3. E3 – размах влажности. Смотрим столбец С. Максимум – 78 % в 8:00, минимум – 51 % в 14:00. $78 - 51 = 27$.

Шаг 4. E4 – сколько замеров теплее 25 °С. Условие ">25" – строго больше. Подходят 26, 29 и 27. Замер 25 °С в 18:00 не считается: он равен 25, а не больше. Итого 3.

Шаг 5. D6 – протянутая формула. В D3 стоит `=B3-B2`. Протянули на три строки вниз – адреса тоже съехали на три строки: в D6 окажется `=B6-B5`. Это $27 - 29 = -2$: с 14:00 до 16:00 похолодало на 2 градуса.

Заодно видно полезное: самый жаркий замер (29 °С в 14:00) совпал с самым сухим воздухом (51 %). Тёплый воздух может удержать больше пара, поэтому, даже если пара в нём не убавилось, влажность в процентах падает – подробнее об этом в главе 2.

❗ Типичная ошибка. Поделить сумму на 8, потому что в таблице 8 строк: $164:8 = 20,5$. Но строка 1 – заголовок, чисел только 7. Вторая ловушка – посчитать 25 в `СЧЁТЕСЛИ` и получить 4. Третья – решить, что протянутая формула осталась прежней, и написать в D6 разность $21 - 17 = 4$.

Ответ: E2 = 23,4; E3 = 27; E4 = 3; D6 = -2.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «Анализ температурных данных»

Реши сам

3.5. В клетках B2:B6 записано, сколько литров воды ушло на полив за пять дней: 12, 15, 9, 14, 10. Что покажут формулы `=СУММ(B2:B6)`, `=СРЗНАЧ(B2:B6)` и `=СЧЁТЕСЛИ(B2:B6;">=12")`?

3.6. Датчик сутки записывал температуру раз в час, 24 замера лежат в клетках B2:B25. Какие формулы правильно посчитают среднюю температуру? Ответ – номера по возрастанию, без пробелов.

1. `=СРЗНАЧ(B2:B25)`
2. `=СУММ(B2:B25)/24`
3. `=СУММ(B2:B25)/25`
4. `=СРЗНАЧ(B2;B25)`
5. `=(МАКС(B2:B25)+МИН(B2:B25))/2`

Задача 3.4 Какой график построен по таблице

график по таблице, масштаб, проверка по точкам

На крыше теплицы стоит фоторезистор – датчик света. Arduino раз в 2 часа записывала его показание `analogRead`. В этой схеме (фоторезистор здесь стоит между +5 В и A0, поэтому, в отличие от схемы курса из раздела 5.3, чем светлее, тем больше число) Arduino записала такие числа:

Таблица 3.4

Время	Показание
-------	-----------

Время	Показание
8:00	310
10:00	560
12:00	820
14:00	450
16:00	790
18:00	520
20:00	140

Какой из четырёх графиков построен по этой таблице? Ответ – номер графика.

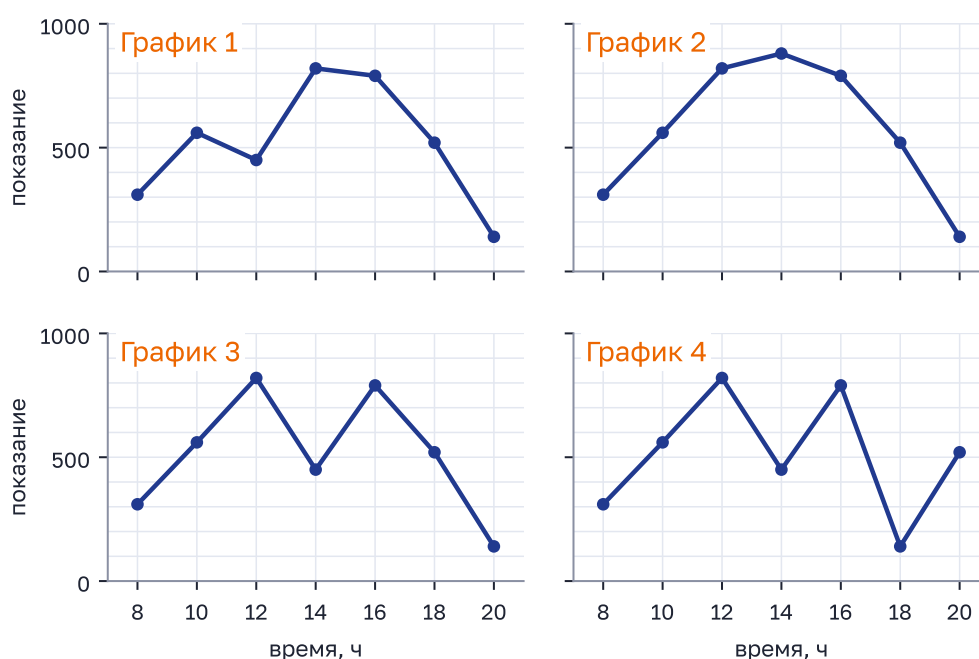


Рисунок 3.9 – Четыре графика показаний фоторезистора

Дано: таблица из 7 замеров, 4 графика.

Найти: номер верного графика.

Решение

Шаг 1. Оси. У всех четырёх графиков оси одинаковые: время от 8 до 20 часов, показания от 0 до 1000. Между 0 и 500 – пять клеток, одна клетка стоит $500:5 = 100$.

Шаг 2. Особые места таблицы. Сравнить все 28 точек подряд долго. Сначала ищем в таблице то, что бросится в глаза на графике. Первое – провал в 14:00: до него 820, после 790, а в 14:00 только 450. Похоже, солнце закрыло облако. Второе – вечером свет падает до 140 в 20:00.

Шаг 3. Отсеиваем. На графике 1 провал стоит в 12:00, а в 14:00 точка поднялась к 820 – значения 12:00 и 14:00 перепутаны. На графике 2 провала нет вовсе: в 14:00 точка около 880. На графике 4 самая низкая точка стоит в 18:00, а в 20:00 линия снова поднимается к 520 – вечером стало светлее, чего по таблице не было. Остаётся график 3.

Шаг 4. Проверяем оставшийся по всем точкам. В 8:00 – чуть выше 300, в 10:00 – между 500 и 600, в 12:00 – чуть выше 800, в 14:00 – между 400 и 500, в 16:00 – чуть ниже 800, в 18:00 – чуть выше 500, в 20:00 – между 100 и 200. Все семь точек совпали с таблицей.

Если бы ни один из графиков не прошёл полную проверку, пришлось бы искать, где ошибся ты сам. Отсев «по особым местам» экономит время, но последнее слово – за проверкой каждой точки.

 **Типичная ошибка.** Выбрать график 2, потому что он «красивый»: солнце утром встаёт, днём светит, вечером садится. Но задача спрашивает, что записал датчик, а не как должно быть. Ещё одна ловушка – график 4: первые пять точек у него совпадают с таблицей, и тот, кто проверил только начало, на нём и остановится.

Ответ: 3.

Своя задача по программе сферы, модуль 6

Реши сам

3.7. Показания датчика света за день – от 140 до 980. Ты рисуешь график на листе в клетку, по высоте есть 20 клеток, отсчёт начинается с нуля. Какую цену клетки выбрать – 20, 50, 100 или 200, – чтобы все точки поместились, а график получился как можно крупнее?

3.8. По таблице задачи 3.4: между какими соседними замерами линия графика идёт круче всего? На сколько изменилось показание на этом участке и в какую сторону?

Задача 3.5 Когда открыть форточку и когда полить

выводы по графику, порог, два графика с общим временем

В теплице с томатами датчики каждые 2 часа записывали температуру воздуха и влажность почвы (рисунок). Правила автоматики: форточку открывают, если воздух теплее 27 °С; полив нужен, если влажность почвы ниже 35 %.

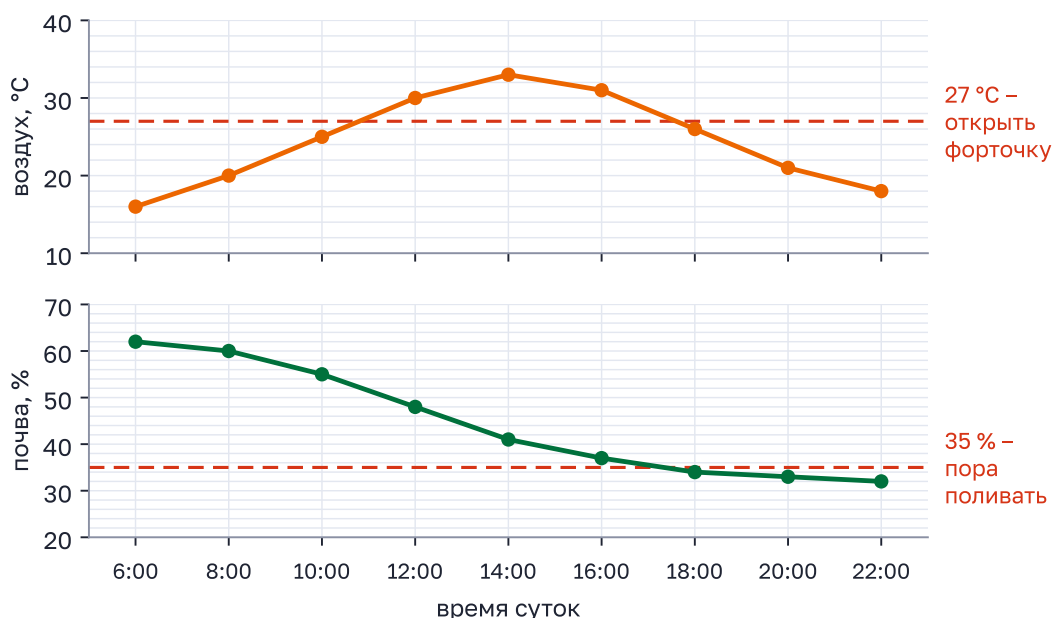


Рисунок 3.10 – Температура воздуха и влажность почвы в теплице за день

Выбери все верные утверждения. Ответ – номера по возрастанию, без пробелов.

1. По замерам форточку нужно было держать открытой в 12:00, 14:00 и 16:00.
2. Самый жаркий замер совпал по времени с самой сухой почвой.
3. Впервые полив понадобился между 16:00 и 18:00.
4. Быстрее всего почва сохла с 10:00 до 14:00, пока воздух разогревался до самой жары.
5. Средняя температура воздуха по всем замерам выше 27 °C.
6. Если не поливать, к 22:00 почва станет суше 30 %.

Дано: два графика с общей осью времени; пороги 27 °C и 35 %.

Найти: номера верных утверждений.

Решение

Шаг 1. Цена деления. Верхний график: подписи 10, 20, 30, 40 °C, между соседними пять клеток – по 2 °C. Нижний: подписи 20, 30 ... 70 %, тоже пять клеток – по 2 %.

Шаг 2. Выписываем замеры. Совет из раздела 3.5: перевести картинку в числа.

Таблица 3.5

Время	Воздух, °C	Почва, %
6:00	16	62
8:00	20	60
10:00	25	55
12:00	30	48
14:00	33	41
16:00	31	37
18:00	26	34

Время	Воздух, °C	Почва, %
20:00	21	33
22:00	18	32

Шаг 3. Утверждение 1. Теплее 27 °C: 30 в 12:00, 33 в 14:00, 31 в 16:00. В 10:00 было 25, в 18:00 – 26, оба ниже порога. Верно.

Шаг 4. Утверждение 2. Самый жаркий замер – 33 °C в 14:00, почва тогда 41 %. Самая сухая почва – 32 % в 22:00. Неверно.

Шаг 5. Утверждение 3. В 16:00 почва 37 % – ещё не ниже 35. В 18:00 – 34 %, уже ниже. Порог пройден между 16:00 и 18:00. Верно.

Шаг 6. Утверждение 4. Сколько процентов почва теряла за каждые 2 часа: 2, 5, 7, 7, 4, 3, 1, 1. Больше всего – по 7 % с 10:00 до 12:00 и с 12:00 до 14:00. Верно.

Шаг 7. Утверждение 5. Сумма температур:
 $16 + 20 + 25 + 30 + 33 + 31 + 26 + 21 + 18 = 220$. Замеров 9: $220 : 9 \approx 24,4$ °C. Это ниже 27. Неверно, хотя три замера и были выше порога.

Шаг 8. Утверждение 6. В 22:00 почва 32 % – это суше 35, но не суше 30. Неверно.

Верные: 1, 3, 4. Интересно, что почва отстаёт от воздуха. Воздух раскалился к 14:00, а почва дошла до порога только к вечеру: вода испаряется и уходит в растения постепенно. Поэтому поливать по часам «в самую жару» бессмысленно – агроном ждёт, что скажет датчик в земле.

⚠ Типичная ошибка. Отметить утверждение 5, потому что «днём почти весь график выше линии 27». Выше порога только три точки из девяти, а утренние и вечерние замеры тянут среднее вниз. Вторая ловушка – в утверждении 3 выбрать 16:00: точка 37 % лежит близко к линии, и кажется, что она уже под ней. Одна клетка здесь – 2 %, посчитай.

Ответ: 134.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «Графики спасают урожай»

Реши сам

3.9. Почва в горшке с рассадой сохнет равномерно: в 7:00 датчик показал 58 %, в 11:00 – 46 %. Полив включается, когда влажность падает до 34 %. Во сколько включится полив?

3.10. Вечером теплица остывает равномерно: в 17:00 было 26 °C, в 19:00 – 21 °C. Форточку закрывают, когда температура опускается до 22 °C, – иначе к утру теплица выстынет. Во сколько её нужно закрыть? Ответ – время в часах и минутах.

Задача 3.6* *. Две теплицы – два графика

сравнение графиков с разными осями, размах, среднее, нормы для растений

Агроном сравнивает две теплицы с томатами по графикам температуры за сутки, замер раз в 3 часа (рисунок). Норма для томатов в этой задаче – от 20 до 28 °С включительно; выше 32 °С пыльца развивается плохо.

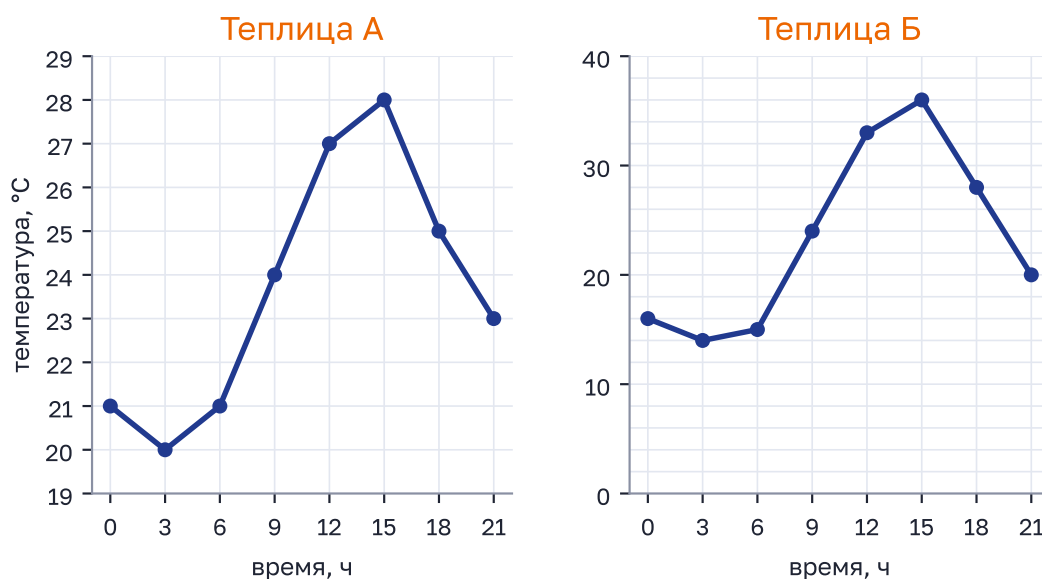


Рисунок 3.11 – Температура в теплицах А и Б за сутки. Обрати внимание на вертикальные оси

Выбери все верные утверждения. Ответ – номера по возрастанию, без пробелов.

1. В теплице А температура менялась сильнее, чем в Б: там линия прошла почти от низа графика до верха.
2. Размах температуры в теплице Б на 14 °С больше, чем в А.
3. Средние температуры за сутки в двух теплицах отличаются меньше чем на 1 °С.
4. Раз средние почти равны, для томатов обе теплицы одинаково хороши.
5. В теплице Б днём пыльца томатов могла пострадать от жары.
6. В теплице А все замеры попали в норму.

Дано: два графика, по 8 замеров; норма 20–28 °С; пыльца страдает выше 32 °С.

Найти: номера верных утверждений.

Решение

Шаг 1. Оси – первым делом. Левый график начинается не с нуля: ось идёт от 19 до 29 °С, каждая линия – 1 °С. У правого ось от 0 до 40 °С, подписи через 10, между ними пять клеток – по 2 °С. Масштабы разные, значит, на глаз сравнивать нельзя.

Шаг 2. Выписываем замеры.

Таблица 3.6

Время	А, °С	Б, °С
0:00	21	16
3:00	20	14
6:00	21	15
9:00	24	24

Время	А, °С	Б, °С
12:00	27	33
15:00	28	36
18:00	25	28
21:00	23	20

Шаг 3. Размахи (утверждения 1 и 2). А: $28 - 20 = 8$ °С. Б: $36 - 14 = 22$ °С. В Б температура менялась почти втрое сильнее, утверждение 1 неверно. Разница размахов $22 - 8 = 14$ °С – утверждение 2 верно.

Шаг 4. Средние (утверждения 3 и 4). А: $21 + 20 + 21 + 24 + 27 + 28 + 25 + 23 = 189$, $189 : 8 = 23,625$ °С. Б: $16 + 14 + 15 + 24 + 33 + 36 + 28 + 20 = 186$, $186 : 8 = 23,25$ °С. Разница $23,625 - 23,25 = 0,375$ °С, меньше градуса – утверждение 3 верно. Но в Б ночью было 14–16 °С, ниже нормы, а днём 33 и 36 °С – выше. Средние равны, а условия разные, утверждение 4 неверно.

Шаг 5. Пыльца (утверждение 5). В Б в 12:00 было 33 °С, а в 15:00 – 36 °С, оба значения выше 32. Верно.

Шаг 6. Норма в А (утверждение 6). Все замеры А – от 20 до 28 °С. Самые крайние, 20 °С в 3:00 и 28 °С в 15:00, лежат на границах нормы, а границы в неё входят («включительно»). Верно.

Верные: 2, 3, 5, 6. Задача показывает сразу два обмана из раздела 3.6. Растянутая ось сделала спокойную теплицу А похожей на американские горки. А одинаковое среднее спрятало, что в Б томаты ночью мёрзнут, а днём перегреваются.

 **Типичная ошибка.** Отметить утверждение 1, сравнив графики на глаз, – ради этой ловушки задача и придумана. Вторая ошибка – выкинуть утверждение 6 из-за замеров 20 и 28 °С: «на границе – значит, уже не норма». Читай условие: норма дана включительно.

Ответ: 2356.

Своя задача по программе сферы, модуль 2

Реши сам

3.11. На графике А вертикальная ось идёт от 18 до 26 °С и разбита на 8 клеток. На графике Б ось от 0 до 40 °С, тоже 8 клеток. За утро линия на графике А поднялась на 6 клеток, а на графике Б – на 3 клетки. В какой теплице потеплело сильнее? На сколько градусов потеплело в каждой?

3.12*.* Датчик записывал температуру в теплице неравномерно: ночью редко, днём часто. Записи: 0:00 – 14 °С, 6:00 – 12 °С, 9:00 – 20 °С, 10:00 – 24 °С, 11:00 – 27 °С, 12:00 – 29 °С, 13:00 – 30 °С, 14:00 – 29 °С, 18:00 – 22 °С. Аня посчитала среднее всех девяти записей. Боря взял только записи с равным шагом – в 0:00, 6:00, 12:00 и 18:00 – и посчитал среднее по ним. Найди оба средних. У кого из ребят среднее завышено и почему?

Вода и полив



Вода – самое простое, что есть в теплице, и на ней же постоянно путаются в арифметике. В этой главе посчитаем, сколько литров выпивает грядка, сколько минут насос будет наполнять бак и сколько воды сбережёт капельный полив. Потом немного физики: зачем бочку ставят повыше и отчего по утрам стёкла теплицы мокрые изнутри. На отборе такие задачи выглядят как короткая история про сити-ферму или дачу с парой чисел, а ответ – одно число в литрах, минутах или процентах.

4.1. Сколько воды нужно грядке

НА ПАЛЬЦАХ

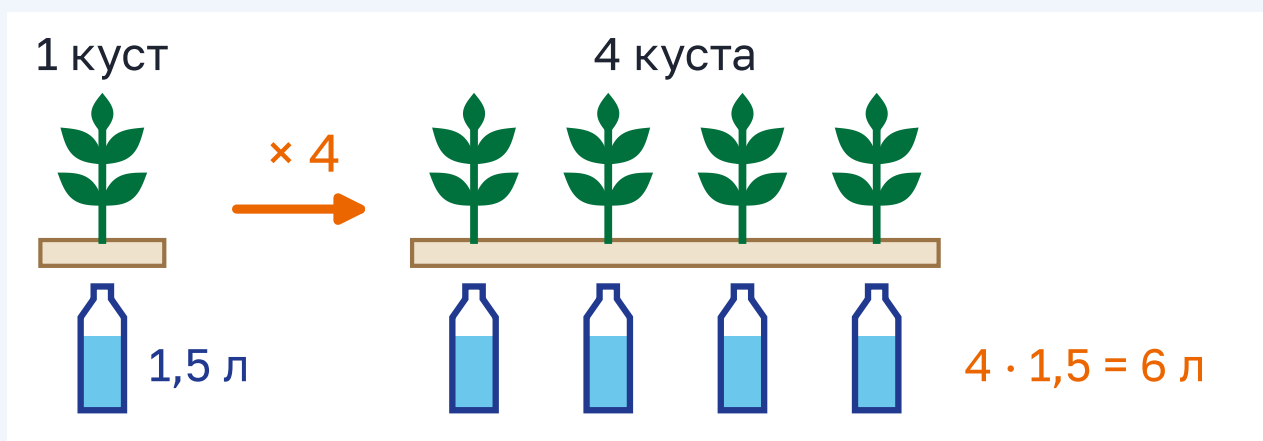


Рисунок 4.1 – Каждому кусту – своя бутылка. Кустов вчетверо больше – и воды вчетверо больше

Если каждому кусту нужно одно и то же, воду на всю грядку считают умножением.

Сколько пьёт растение, зависит от культуры, возраста, погоды и почвы. Рассада томата и взрослый куст с плодами в жаркий июльский день пьют совсем по-разному. Поэтому в задачах норму полива дают прямо в условии, и считать надо по ней. Встречается она в двух видах.

- **На одно растение:** «каждому кусту 1,5 л в сутки». Тогда воду на грядку находим так: норма на куст умножить на число кустов.
- **На площадь:** «4 л на 1 м² за полив». Тогда норму умножаем на площадь грядки в квадратных метрах. Площадь прямоугольной грядки – длина, умноженная на ширину: грядка 1,2 м × 5 м занимает $1,2 \cdot 5 = 6$ м².

В обоих случаях работает **прямая пропорциональность**: во сколько раз больше кустов (метров, дней), во столько раз больше воды. Одному кусту нужно 1,5 л в сутки – четырём нужно $4 \cdot 1,5 = 6$ л, а за неделю им же нужно $6 \cdot 7 = 42$ л. Порядок умножения не важен, важно не потерять ни одного множителя.

Пропорцию можно решать и «через одну штуку». Для 12 горшков нужно 3 л воды в сутки, а сколько нужно для 20? Сначала на один горшок: $3 : 12 = 0,25$ л. Потом на двадцать: $0,25 \cdot 20 = 5$ л. Этот приём спасает, когда числа неудобные и сразу не видно, «во сколько раз».

Теперь про круглые числа. Воду запасают так, чтобы **хватило**, поэтому запасы округляют вверх. Если полив съест 3,15 бочки, наполнять бочку придётся 4 раза: после трёх наполнений грядке не хватит. А вопрос «на сколько целых суток хватит бака» решается наоборот, вниз: 2,78 суток – это 2 полных дня, на третий воды уже не хватит.

Запомни. Вода на грядку = норма × число кустов (или × площадь) × число дней. Запас – округляем вверх, «сколько полных дней хватит» – вниз.

4.2. Литры, баки и слой воды

НА ПАЛЬЦАХ

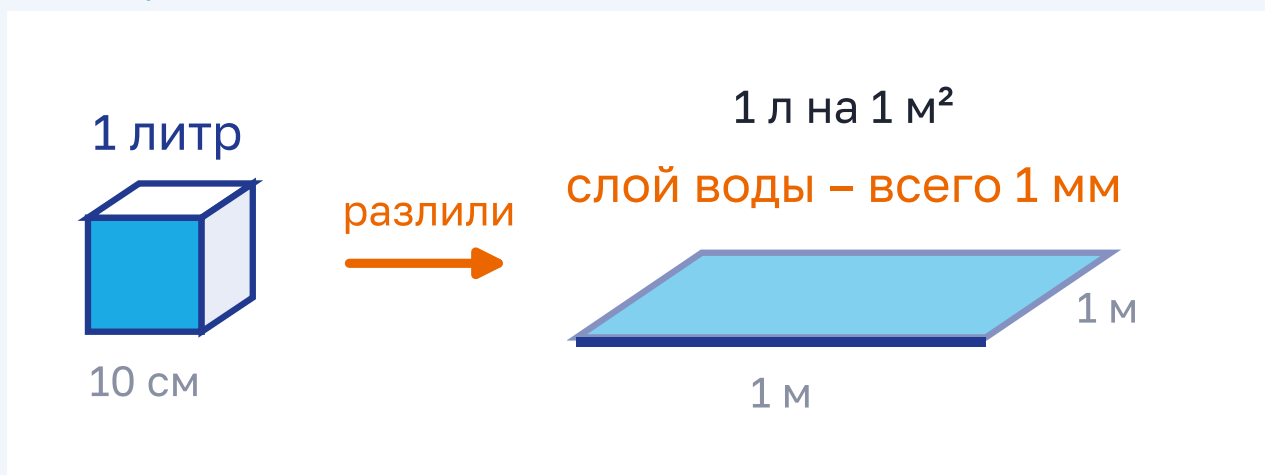


Рисунок 4.2 – Литр – это кубик со стороной 10 см. Разлей его по квадратному метру – выйдет слой толщиной 1 мм

Литр на квадратный метр – это миллиметр воды. Поэтому дождь и полив меряют в миллиметрах.

Литр – объём кубика со стороной 10 см, то есть 1 дм. Пакет молока – почти он. Всё, что нужно для полива, выражается через литр:

- 1 л = 1000 мл, а 1 мл = 1 см³ (кубик со стороной 1 см, как игральная кость);
- 1 л = 1 дм³ = 1000 см³;
- 1 м³ = 1000 л – кубометр, куб со стороной 1 м;
- литр воды весит 1 кг, миллилитр – 1 г.

Когда нужно найти, сколько литров в баке, бак почти всегда оказывается **бруском** или **цилиндром** (бочкой).

Объём бруска – длина, умноженная на ширину и на высоту: $V = a \cdot b \cdot c$. Бак 60 см × 40 см × 50 см вмещает $60 \cdot 40 \cdot 50 = 120\,000$ см³. В литре 1000 см³, поэтому это $120\,000 : 1000 = 120$ л. Можно сразу считать в дециметрах: $6 \cdot 4 \cdot 5 = 120$ дм³ = 120 л – короче, и нули не теряются.

Объём цилиндра – площадь круглого дна, умноженная на высоту: $V = \pi r^2 h$. Здесь $\pi \approx 3,14$ – число «пи» (кнопка на калькуляторе), r – **радиус**, половина диаметра, h – высота. Бочка диаметром 50 см и высотой 80 см: радиус 25 см = 2,5 дм, высота 8 дм, $V = 3,14 \cdot 2,5^2 \cdot 8 = 157$ дм³, то есть около 157 л. В условиях дают диаметр, а в формулу нужен радиус. Кто забыл поделить на 2, получит бочку вчетверо больше настоящей.

Картинка выше даёт полезное правило: **1 л на 1 м² – это слой в 1 мм**. Проверим счётом: квадратный метр – это 10 дм × 10 дм = 100 дм². Слой 1 мм – это 0,01 дм. Объём слоя $100 \cdot 0,01 = 1$ дм³ = 1 л. Отсюда: если метеостанция намерила за ночь 5 мм дождя, каждый квадратный метр грядки получил 5 л воды. И наоборот: полив «4 л на 1 м²» – это слой в 4 мм.

Запомни. $1 \text{ л} = 1 \text{ дм}^3 = 1000 \text{ см}^3 = 1000 \text{ мл}$; $1 \text{ м}^3 = 1000 \text{ л}$; $1 \text{ л воды} = 1 \text{ кг}$. Считай объём в дециметрах – сразу получишь литры. $1 \text{ мм осадков или полива} = 1 \text{ л на каждый квадратный метр}$.

Совет. Перед подсчётом объёма переведи все размеры в одну единицу. Бак « $1,2 \text{ м} \times 50 \text{ см} \times 400 \text{ мм}$ » в дециметрах выглядит как $12 \cdot 5 \cdot 4 = 240 \text{ л}$ – и никаких ошибок в нулях.

4.3. Насос: литры в минуту

НА ПАЛЬЦАХ

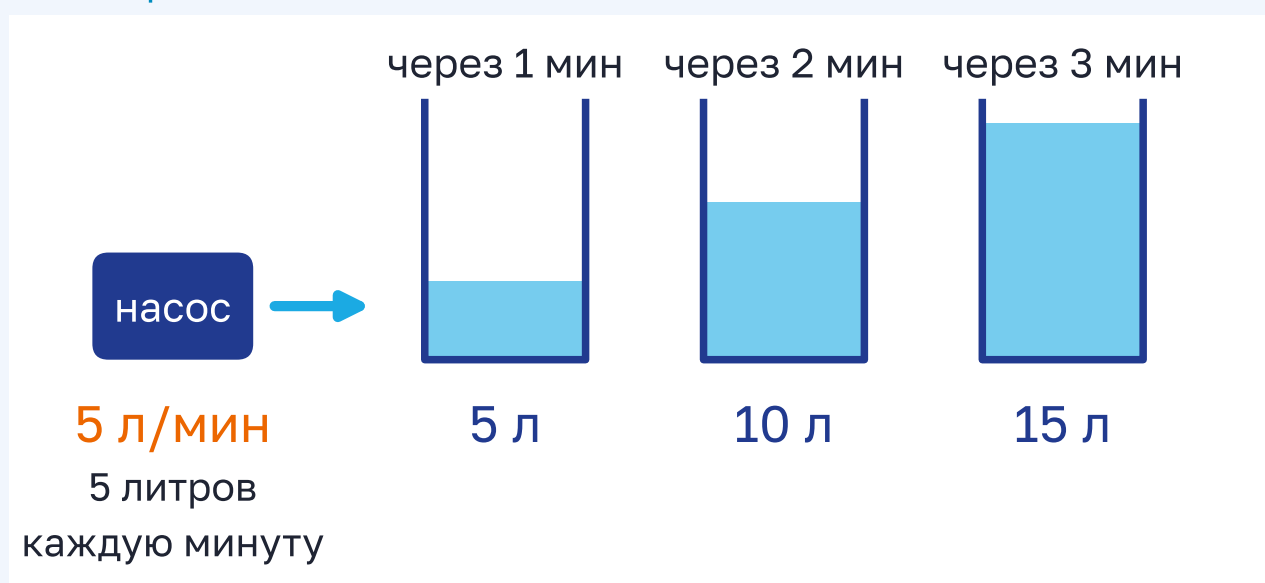


Рисунок 4.3 – Насос на 5 л/мин: через минуту в баке 5 л, через две – 10 л, через три – 15 л

Производительность насоса – сколько литров он перекачивает за одну минуту (или за час).

Насос – моторчик с крыльчаткой, который гонит воду по трубке. В умной теплице он поливает грядки и наполняет баки. Его паспортное число – **производительность**: сколько литров насос перекачивает за единицу времени. На насосе или в его описании пишут, например, «300 л/ч» или «5 л/мин».

Считают с насосом, как с дорогой: литры – это путь, производительность – скорость, минуты – время.

- **Сколько воды накачает:** объём = производительность $\times$ время. 5 л/мин за 12 мин – $5 \cdot 12 = 60 \text{ л}$.
- **Сколько времени качать:** время = объём : производительность. 120 л при 5 л/мин – $120 : 5 = 24 \text{ мин}$.
- **Какой нужен насос:** производительность = объём : время. Бак на 90 л нужно наполнять за 15 мин – насос должен давать $90 : 15 = 6 \text{ л/мин}$.

Единицы должны подходить друг к другу. Если производительность в литрах в час, а время спрашивают в минутах, сначала переведи: $300 \text{ л/ч} = 300 : 60 = 5 \text{ л/мин}$ (в часе 60 минут, и за минуту насос успеет в 60 раз меньше). Маленькие насосы для комнатных теплиц иногда описывают в миллилитрах в секунду: 50 мл/с – это $50 \cdot 60 = 3000 \text{ мл/мин} = 3 \text{ л/мин}$.

В программе умной теплицы насос часто включают **на время**. Надо налить 3 л насосом на 2 л/мин – держим его включённым $3 : 2 = 1,5 \text{ мин}$, то есть 90 с. В программе Arduino время пишут в миллисекундах, значит, насос должен работать 90 000 мс. Как Arduino включает насос и держит его нужное время, будет в главе 6.

Бывает, что вода одновременно и прибывает, и убывает: насос наполняет бак, а капельницы из этого же бака поливают грядку. Тогда бак наполняется со скоростью «насос минус капельницы». Насос даёт 8 л/мин, капельницы берут 2 л/мин – уровень в баке растёт на $8 - 2 = 6 \text{ л}$ каждую минуту. Так же наполняется ванна, если забыть заткнуть пробку: вода прибывает, но медленнее, чем течёт из крана.

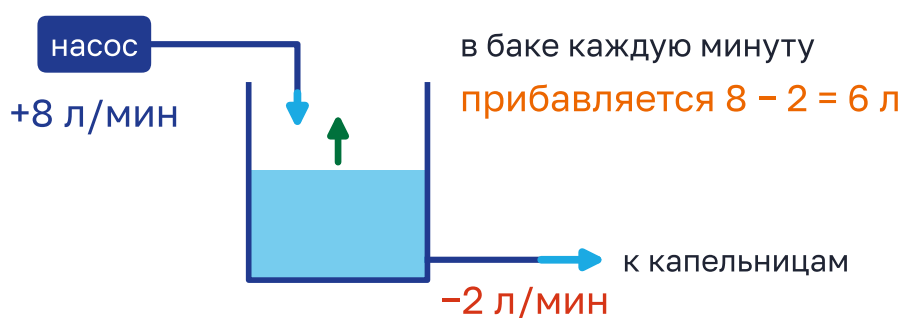


Рисунок 4.4 – Насос подливает 8 л в минуту, капельницы забирают 2 л. В баке прибавляется разница – 6 л в минуту

Запомни. Объем = производительность × время; время = объем : производительность. Перед делением приведи л/ч и минуты (или мл и литры) к одним единицам. Вода и приходит, и уходит – считай разницу.

4.4. Капельный полив: считаем экономию

НА ПАЛЬЦАХ



Рисунок 4.5 – Из каждых 10 л при поливе из шланга до корней доходят 6 л, при капельном – 9 л. Числа для примера

Вода, которая растеклась мимо корней или испарилась с земли, растению не досталась. Капельный полив теряет её меньше.

Когда поливаешь из шланга или лейки, вода разливается по всей грядке. Часть уходит в дорожки, часть впитывается туда, где корней нет, часть испаряется с мокрой земли. **Капельный полив** устроен иначе: вдоль грядки лежит трубка или лента с маленькими отверстиями – **капельницами**, и у каждого растения вода сочится по капле прямо под корень. Земля между кустиками остаётся сухой, испаряться с неё нечему, и сорнякам там тоже пить нечего.

Долю воды, которая дошла до корней, в задачах дают в процентах. Скажем, «при поливе шлангом до корней доходит 60 % вылитой воды». Это значит, что растению достаётся 0,6 от вылитого. Отсюда две формулы:

- **сколько дойдёт:** дошло = вылили · доля. Вылили 20 л, доля 60 % – дошло $20 \cdot 0,6 = 12$ л;
- **сколько вылить:** вылили = нужно : доля. Корням нужно 12 л, доля 60 % – вылить $12 : 0,6 = 20$ л.

Экономия в процентах считается от того, что было **до** перемен:

$$\text{экономия, \%} = \frac{\text{было} - \text{стало}}{\text{было}} \cdot 100.$$

Тратили 250 л в неделю, стали тратить 200 л. Сберегли 50 л, а это $50 : 250 \cdot 100 = 20$ %. Если поделить 50 на 200 (на новый расход), получится 25 % – неверно: сравнивают с прежним расходом.

И обратный ход. «Расход уменьшили на 20 %, стало 200 л» – сколько было? Уменьшить на 20 % – значит оставить 80 %. Значит, 200 л – это 80 % от прежнего, и было $200 : 0,8 = 250$ л. Прибавлять к 200 двадцать процентов нельзя: $200 \cdot 1,2 = 240$ – не то число.

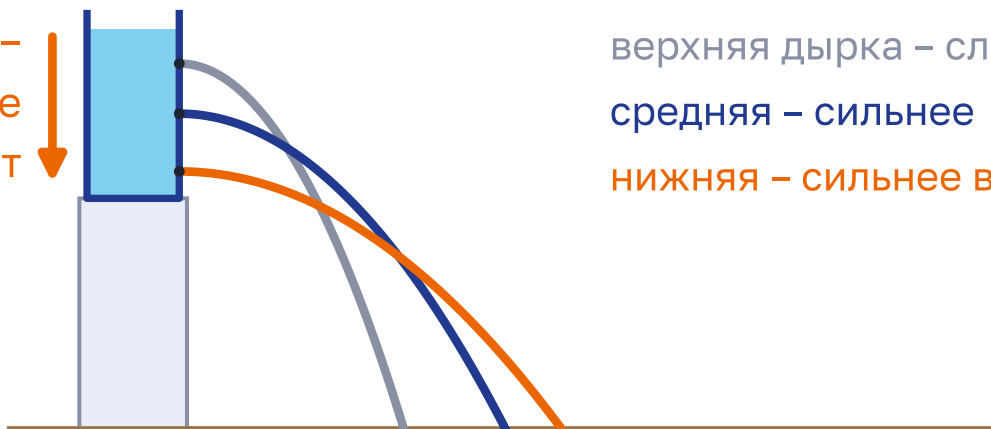
Запомни. Вылить = нужно корням : доля. Процент экономии – от старого расхода. «Стало меньше на 20 %» = «стало 80 % от прежнего».

Совет. Проценты в задаче про две доли не вычитай друг из друга. Из «60 % и 90 %» не следует «экономия 30 %» – посчитай литры для каждого способа и сравни литры.

4.5. Давление воды: глубже – сильнее давит

НА ПАЛЬЦАХ

глубже –
сильнее
давит



верхняя дырка – слабо
средняя – сильнее
нижняя – сильнее всего

Рисунок 4.6 – Бутылка с тремя дырками на разной глубине

Чем глубже дырка, тем больше воды над ней, и тем сильнее вода выталкивает струю.

Нырни в бассейне на дно – заложит уши. Это давит вода: чем глубже, тем сильнее. Чтобы посчитать, насколько сильнее, нужны два слова из физики.

Сила – то, чем одно тело толкает или тянет другое. Её меряют в **ньютонах** (Н). Земля тянет к себе каждый килограмм с силой около 9,8 Н. Поэтому пакет молока (1 кг) давит на ладонь с силой около 10 Н, а ведро воды на 10 л – около 98 Н. Число 9,8 Н/кг обозначают буквой *g*.

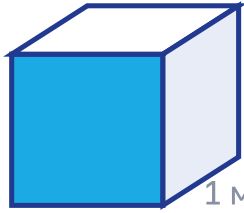
Давление – сила, которая приходится на каждый квадратный метр. Словами: давление равно силе, делённой на площадь. Буквами:

$$p = \frac{F}{S},$$

где *p* – давление в **паскалях** (Па), *F* – сила в ньютонах, *S* – площадь в квадратных метрах. Один паскаль – это сила 1 Н на квадратный метр, очень мало. Поэтому часто пишут **килопаскали**: 1 кПа = 1000 Па. Одна и та же сила на маленькой площади давит сильнее: поэтому гвоздь с острым концом входит в доску, а лыжи не проваливаются в сугроб.

Теперь посчитаем давление воды на дне. Возьмём столб воды над площадкой в 1 м² высотой 1 м.

1 м³ воды



1 м

1 м

объём	1 м ³ = 1000 л
масса	1000 л · 1 кг/л = 1000 кг
сила (вес)	1000 кг · 9,8 Н/кг = 9800 Н
давление	9800 Н : 1 м ² = 9800 Па

Рисунок 4.7 – Столб воды высотой 1 м над квадратным метром дна: считаем давление по шагам

Объём столба – 1 м³, это 1000 л и 1000 кг. Вес такой воды – $1000 \cdot 9,8 = 9800$ Н, и вся эта сила давит на 1 м² дна. Давление – 9800 Па, почти 10 кПа. Столб высотой 2 м весит вдвое больше и давит вдвое сильнее, 19 600 Па. Получается правило: **каждый метр глубины воды добавляет 9,8 кПа**. Формулой:

$$p = \rho \cdot g \cdot h,$$

где ρ («ро») – плотность воды, 1000 кг/м³ (масса одного кубометра), $g = 9,8$ Н/кг, h – глубина или высота столба воды в **метрах**. Для воды $\rho \cdot g = 9800$, поэтому $p = 9800 \cdot h$ Па.

Вот где в задачах спотыкаются:

- **Давление зависит только от глубины.** В узкой трубе и в широком бассейне на одной глубине давление одинаковое. Литры тут ни при чём: большой бак давит на дно сильнее целиком (дно больше), но каждый его квадратный метр получает столько же.
- **Высоту считают от поверхности воды**, а не от дна бака и не от земли. Для капельницы на грядке h – это расстояние по высоте от уровня воды в бочке до самой капельницы.
- **Уровень опускается – давление падает.** Бочка пустеет, столб воды над капельницами становится ниже, и вода течёт всё слабее.

Поэтому на дачах бочку для самотёчного полива ставят на подставку, а в городе строили **водонапорные башни**: бак высоко наверху давит на воду в трубах, и она доходит до кранов. Для сравнения: на глубине 10 м вода давит с силой $9800 \cdot 10 = 98\,000$ Па, около 100 кПа. Примерно так же давит на нас весь воздух над головой.

Запомни. $p = F : S$, давление в паскалях. Вода: каждый метр глубины – плюс 9,8 кПа, $p = 9800 \cdot h$ Па, где h в метрах от поверхности воды. Ширина бака и число литров на давление не влияют.

Совет. Высоту в сантиметрах сначала переведи в метры: 60 см = 0,6 м. А ответ в паскалях перед сравнением с «кПа» из условия подели на 1000.

4.6. Испарение и конденсация: почему потеют стёкла

НА ПАЛЬЦАХ

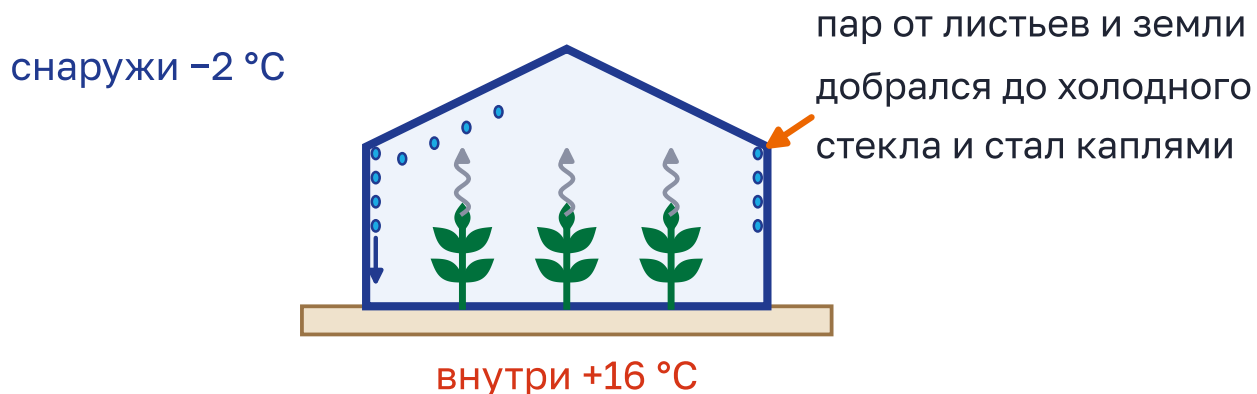


Рисунок 4.8 – Холодная ночь. Пар от листьев и земли долетает до холодного стекла и оседает каплями

Стекло с одной стороны тёплое от воздуха теплицы, а с другой его остужает мороз. Возле такого стекла пар превращается обратно в воду.

Испарение – превращение воды в невидимый водяной пар. Лужа после дождя высыхает, мокрая футболка на верёвке сохнет – это испарение. Идёт оно тем быстрее,

- чем теплее (в жару лужа высыхает за час, в прохладу – за день);
- чем суше воздух вокруг (в сыром воздухе бельё сохнет еле-еле);
- чем сильнее ветер (он уносит пар от поверхности);
- чем больше открытая поверхность воды (разлитая по полу вода высохнет быстрее, чем в стакане).

Испаряясь, вода забирает тепло. Выйди из реки на ветер – сразу станет холодно, хотя на улице жара. Так же и лист: в главе 1 мы видели, что растение испаряет через устьица больше 90 % выпитой воды, и этим охлаждает себя в жару.

Конденсация – обратный путь: пар снова становится каплями. Происходит это на холодном. Достань бутылку из холодильника – через минуту она мокрая снаружи. Зимой в школьном автобусе окна мокрые изнутри: снаружи мороз, а внутри тёплый воздух и дыхание пассажиров.

Помнишь «стакан для пара» из главы 1 (раздел 1.7)? Тёплый воздух – большой стакан, холодный – маленький. Воздух теплицы касается холодного стекла, остывает, и стакан у него сжимается. Пар, который не влез, выпадает каплями на стекло. В граммах это считали в задаче 1.7 и в разделе 2.3.

В теплице вода ходит по кругу. Утром полили грядку. Корни выпили воду, листья испарили её в воздух, заодно испарилось немного с земли. Ночью стёкла остыли – пар сел на них каплями. Капли стекают по стенкам вниз, на землю, а с крыши иногда капают прямо на листья. Мокрые листья на всю ночь – раздолье для грибковых болезней. Поэтому теплицу проветривают, пока воздух на улице сухое, и не поливают поздно вечером.

Для полива испарение – потеря. Вода испаряется из открытой бочки, с мокрой земли вокруг кустов, из дорожек. Чтобы терять меньше, поливают утром или вечером, а не в полуденную

жару, накрывают бочку крышкой и укрывают землю **мульчей** – слоем соломы, скошенной травы или плёнки. Мουλча закрывает землю от солнца и ветра, и влага держится дольше.

Запомни. Испарение ускоряют тепло, сухой воздух, ветер и большая открытая поверхность; испаряясь, вода охлаждает. Конденсация идёт на холодном: пар из тёплого воздуха оседает каплями на холодном стекле. 1 г воды – это 1 мл.

4.7. Разбираем задачи

Задача 4.1 Томаты на неделю

разминка · пропорция, расход воды, округление запаса

В теплице три ряда томатов, в каждом ряду 8 кустов. В жаркую неделю каждому кусту нужно 1,5 л воды в сутки. Воду берут из бочки на 80 л.

- а) Сколько литров воды нужно всем кустам за неделю?
- б) Сколько раз за неделю придётся наполнить бочку, если она каждый раз наполняется доверху?

Дано: 3 ряда по 8 кустов; 1,5 л на куст в сутки; 7 суток; бочка 80 л.

Найти: воду за неделю; сколько раз наполнять бочку.

Решение

Шаг 1. Сколько кустов. $3 \cdot 8 = 24$ куста.

Шаг 2. Вода за сутки. Каждому по 1,5 л: $24 \cdot 1,5 = 36$ л в сутки.

Шаг 3. Вода за неделю. $36 \cdot 7 = 252$ л.

Шаг 4. Сколько бочек. $252 : 80 = 3,15$. Три полные бочки – это 240 л, не хватает ещё 12 л. Значит, бочку наполняют 4 раза: последний раз её выльют не до дна, но без этого наполнения 12 л не взять.

Прикинем на вёдрах: 36 л в день – три с половиной ведра, за неделю – около 25 вёдер. Трёх бочек по 8 вёдер на это и правда не хватает.

 **Типичная ошибка.** Округлить 3,15 «по правилам» до 3. Но три бочки – 240 л, и в последний день части кустов воды не достанется. Запас округляют вверх. Бывает и другое: забывают про 7 дней и отвечают «36 л» – это только одни сутки.

Ответ: а) 252 л; б) 4 раза.

Своя задача по программе сферы, модуль 2

Реши сам

4.1. Грядка с огурцами – $1,2 \text{ м} \times 5 \text{ м}$. За один полив ей нужно 4 л воды на каждый квадратный метр. Сколько литров нужно на один полив? Каким слоем (в миллиметрах) легла бы эта вода, если разлить её равномерно по всей грядке?

4.2. На сити-ферме 4 стеллажа, на каждом 3 полки, на каждой полке 16 горшков с зеленью. Каждому горшку нужно 250 мл воды в сутки. Сколько литров воды ферма тратит в сутки? На сколько полных суток хватит бака на 150 л?

Задача 4.2 Сколько минут наполнять бак

объём бруска, перевод единиц, производительность насоса

Для капельного полива на крыше поставили пластиковый бак-брусок. Его внутренние размеры: длина 80 см, ширина 50 см, высота 45 см. Бак наполняют насосом, на котором написано «300 л/ч». Сколько минут нужно, чтобы наполнить пустой бак доверху?

Дано: $a = 80 \text{ см}$; $b = 50 \text{ см}$; $c = 45 \text{ см}$; производительность 300 л/ч.

Найти: время наполнения t в минутах.

Решение

Шаг 1. Объём бака. Переведём размеры в дециметры, тогда объём сразу выйдет в литрах: $80 \text{ см} = 8 \text{ дм}$, $50 \text{ см} = 5 \text{ дм}$, $45 \text{ см} = 4,5 \text{ дм}$. $V = 8 \cdot 5 \cdot 4,5 = 180 \text{ дм}^3 = 180 \text{ л}$.

Шаг 2. Производительность в минутах. Ответ нужен в минутах, а насос подписан в литрах в час. В часе 60 минут: $300 : 60 = 5 \text{ л/мин}$.

Шаг 3. Время. Время = объём : производительность: $t = 180 : 5 = 36 \text{ мин}$.

Проверим другим путём. Поделим литры на литры в час, получим часы: $180 : 300 = 0,6 \text{ ч}$. А $0,6 \cdot 60 = 36 \text{ мин}$ – то же самое. Меньше часа на бак, в который помещается почти два десятка вёдер, – для небольшого насоса правдоподобно.

⚠ Типичная ошибка. Поделить 180 на 300 и записать в ответ «0,6». Это часы, а спрашивали минуты. Другая – считать объём в сантиметрах и забыть перевести:

$80 \cdot 50 \cdot 45 = 180\,000$ – это кубические сантиметры, а не литры, и время выйдет в тысячу раз больше.

Ответ: 36 мин.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «Насос для зелёного оазиса»

Реши сам

4.3. На насосе написано «150 л/ч». Сколько литров он накачает за 18 минут?

4.4. Бочка-цилиндр имеет внутренний диаметр 60 см и высоту 90 см. Её наполняют насосом на 12 л/мин. За сколько минут наполнится пустая бочка? Число π бери с калькулятора, ответ округли до десятых.

Задача 4.3 Шланг или капельницы

проценты, доля, экономия воды

Фермер поливает грядки сити-фермы из шланга. Растениям на ферме нужно, чтобы до корней доходило 18 л воды в сутки. При поливе из шланга до корней доходит 60 % вылитой воды, при капельном поливе – 90 %.

- а) Сколько литров в сутки нужно выливать при каждом способе полива?
- б) На сколько процентов меньше воды тратит капельный полив? Округли до целых.
- в) Сколько литров сэкономит капельный полив за лето – с 1 июня по 31 августа?

Дано: до корней нужно 18 л в сутки; доля шланга 60 %; доля капельниц 90 %.

Найти: расход каждого способа; экономию в процентах; экономию за лето в литрах.

Решение

Шаг 1. Что значит «доходит 60 %». Из каждого вылитого литра корням достаётся 0,6 л. Нам известно, сколько должно дойти, а неизвестно, сколько вылить. Значит, делим: вылить = нужно : доля.

Шаг 2. Шланг. $18 : 0,6 = 30$ л в сутки. Проверка: $30 \cdot 0,6 = 18$ л доходит до корней.

Шаг 3. Капельницы. $18 : 0,9 = 20$ л в сутки. И обратно: $20 \cdot 0,9 = 18$ л.

Шаг 4. Экономия в литрах. $30 - 20 = 10$ л в сутки.

Шаг 5. Экономия в процентах. Сравниваем с тем, что было до перехода на капельницы, – со шлангом: $10 : 30 \cdot 100 = 33,3$ %, до целых – 33 %.

Шаг 6. За лето. В июне 30 дней, в июле и августе по 31: $30 + 31 + 31 = 92$ дня.
 $10 \cdot 92 = 920$ л.

Почти тонна воды с одной маленькой фермы – примерно пять больших бочек. Ради такого капельный полив и ставят.

⚠ Типичная ошибка. Вычесть проценты: $90 - 60 = 30$ %. Но 60 % и 90 % – это доли от разных количеств вылитой воды, и их разница не говорит, на сколько уменьшился расход. Сравнивать надо литры. Вторая ошибка – умножить вместо деления: $18 \cdot 0,6 = 10,8$ л. Выходит, что вылить нужно меньше, чем должно дойти до корней, а так не бывает.

Ответ: а) шланг – 30 л, капельницы – 20 л; б) на 33 %; в) 920 л.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «Как полить сити-ферму и сэкономить?»

Реши сам

4.5. Раньше дачный участок тратил на полив 250 л воды в неделю, после установки капельного полива – 175 л. На сколько процентов уменьшился расход?

4.6. После перехода на капельный полив расход воды на сити-ферме уменьшился на 35 % и стал 390 л в неделю. Сколько литров в неделю тратили раньше? Сколько литров в неделю теперь сберегают?

Задача 4.4 Бочка на подставке

давление жидкости, перевод единиц

На даче грядку поливают самотёком: бочка стоит на подставке, от неё к грядке идёт трубка с капельной лентой (рисунок ниже). Уровень воды в бочке на 2,1 м выше капельниц. Считай $g = 9,8 \text{ Н/кг}$, плотность воды 1000 кг/м^3 .

- а) Какое давление создаёт вода у капельниц? Ответ дай в килопаскалях, округли до десятых.
б) По описанию капельницы работают как надо, если давление не меньше 14,7 кПа. На какой наименьшей высоте над капельницами должен оставаться уровень воды в бочке?

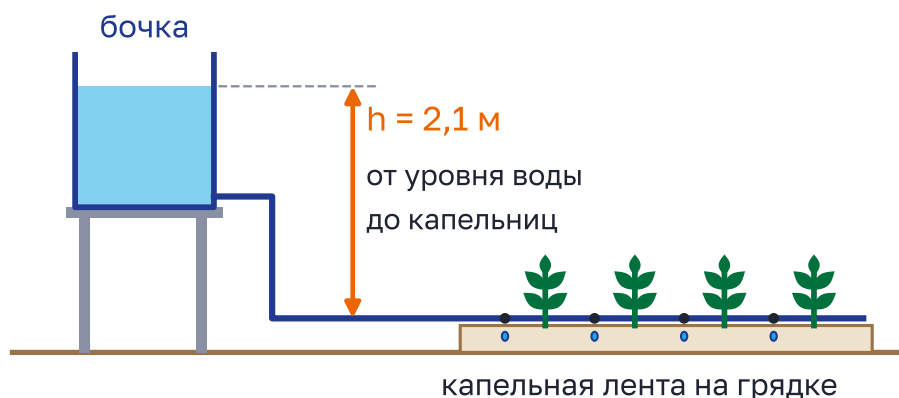


Рисунок 4.9 – Бочка на подставке. Высота h считается от уровня воды в бочке до капельниц

Дано: $h = 2,1 \text{ м}$; $\rho = 1000 \text{ кг/м}^3$; $g = 9,8 \text{ Н/кг}$; нужно не меньше 14,7 кПа.

Найти: p в кПа; наименьшую h .

Решение

Шаг 1. Давление столба воды. $p = \rho \cdot g \cdot h = 1000 \cdot 9,8 \cdot 2,1 = 20\,580 \text{ Па}$.

Шаг 2. В килопаскалях. В 1 кПа 1000 Па: $20\,580 : 1000 = 20,58 \text{ кПа}$, до десятых – 20,6 кПа.

Шаг 3. Наименьшая высота. Давление и высота пропорциональны: каждый метр даёт 9800 Па. Переведём 14,7 кПа в паскалы: $14,7 \cdot 1000 = 14\,700 \text{ Па}$. Сколько метров воды дают такое давление? $h = 14\,700 : 9800 = 1,5 \text{ м}$.

Умножим обратно: $9800 \cdot 1,5 = 14\,700 \text{ Па}$. Пока уровень воды выше капельниц хотя бы на 1,5 м, капельницы работают как надо. Уровень опустится на 0,6 м ($2,1 - 1,5$) – и полив начнёт слабеть.

⚠ Типичная ошибка. Взять высоту подставки или высоту от земли до дна бочки. Давит весь столб воды над капельницей – от поверхности воды в бочке до самой капельницы. Ещё одна – подставить высоту в сантиметрах: $1000 \cdot 9,8 \cdot 210$ даёт 2 058 000 Па, в сто раз больше верного. В формулу высота идёт в метрах.

Ответ: а) 20,6 кПа; б) 1,5 м.

Своя задача по программе сферы, модуль 2

Реши сам

4.7. В баке с водой глубина 60 см. Какое давление создаёт вода на дне бака? Ответ дай в паскалях. $g = 9,8 \text{ Н/кг}$.

4.8. В теплице два бака. В баке А – 200 л воды, высота воды в нём 0,8 м. Бак Б широкий, в нём 1000 л, а высота воды 0,5 м. На дне какого бака давление воды больше и во сколько раз?

Задача 4.5 Мокрые стёкла

конденсация, перевод граммов в литры, проценты

За жаркий день растения и земля в теплице испарили 14 л воды. Ночью ударил заморозок, и к утру на внутренней стороне стёкол осело по 35 г воды на каждый квадратный метр стекла. Площадь всех стёкол теплицы – 60 м².

а) Сколько литров воды оказалось утром на стёклах?

б) Какую часть испарённой за день воды это составляет? Ответ – в процентах.

Дано: испарилось 14 л; на стёклах 35 г/м²; площадь стёкол 60 м².

Найти: воду на стёклах в литрах; её долю от испарённой в процентах.

Решение

Шаг 1. Откуда капли. Пар, который днём ушёл в воздух теплицы, ночью касается холодного стекла. Воздух у стекла остывает и уже не держит весь пар, лишний оседает каплями. Поэтому стёкла мокрые изнутри, со стороны тёплого воздуха.

Шаг 2. Вода на всех стёклах в граммах. $35 \cdot 60 = 2100 \text{ г}$.

Шаг 3. В литры. Грамм воды – это миллилитр, значит, 2100 г = 2100 мл. В литре 1000 мл: $2100:1000 = 2,1 \text{ л}$.

Шаг 4. Доля. Сравниваем с тем, что испарилось: $2,1:14 \cdot 100 = 15 \%$.

Значит, большая часть пара осталась в воздухе или ушла через щели, а на стёкла села примерно седьмая часть. Два с лишним литра – это больше восьми стаканов воды, которые к утру стекут по стенкам.

⚠ Типичная ошибка. Ответить «2100 л» – забыть, что 35 – это граммы. Литр воды весит килограмм, а тут всего 2,1 кг. И ещё: поделить наоборот, $14:2,1$, и получить 6,7. Доля – всегда часть, делённая на целое, и в процентах она здесь меньше 100.

Ответ: а) 2,1 л; б) 15 %.

Своя задача по программе сферы, модуль 2

Реши сам

4.9. Выбери верные утверждения. В ответ запиши их номера по возрастанию, без пробелов.

1. Стёкла теплицы потеют изнутри, потому что вода с улицы просачивается сквозь стекло.
2. Капли появляются на стекле, когда воздух у холодного стекла остывает и не может удержать весь пар.
3. При одинаковой погоде в теплице, полной растений, стёкла к утру мокрее, чем в пустой.
4. Если вечером проветрить теплицу, пока воздух на улице суше, к утру капель на стёклах будет меньше.
5. Лужа высохнет быстрее в прохладный безветренный день, чем в жаркий ветреный.
6. Когда вода испаряется с листа, лист охлаждается.

4.10. В жаркую погоду из открытой бочки диаметром 60 см каждый день испаряется слой воды толщиной 6 мм. Сколько литров воды испарится за 10 таких дней? Округли до целых.

Задача 4.6^{*}. Автополив с баком

производительность насоса, приход и расход, пороги

В умной теплице 48 капельниц, каждая даёт 2,5 л воды в час. Все они берут воду из бака. Бак подкачивает насос на 8 л/мин. Автоматика следит за уровнем: когда в баке остаётся 50 л, насос включается, а когда воды становится 200 л – выключается. Капельницы работают всё время без перерыва.

а) Сколько минут насос работает за одно включение?

б) Полив шёл 10 часов и начался с полным баком (200 л). Сколько раз за это время включался насос?

Дано: 48 капельниц по 2,5 л/ч; насос 8 л/мин; включение при 50 л, выключение при 200 л; 10 ч; в начале 200 л.

Найти: время одной работы насоса; число включений за 10 ч.

Решение

Шаг 1. Расход капельниц в минуту. Все вместе: $48 \cdot 2,5 = 120$ л/ч. Насос подписан в литрах в минуту, поэтому переводим: $120 : 60 = 2$ л/мин.

Шаг 2. Как быстро наполняется бак. Пока насос работает, капельницы продолжают брать воду. Каждую минуту насос добавляет 8 л, капельницы забирают 2 л, в баке прибавляется $8 - 2 = 6$ л.

Шаг 3. Одна работа насоса. Он поднимает уровень от 50 л до 200 л, то есть на $200 - 50 = 150$ л. Время: $150 : 6 = 25$ мин.

Шаг 4. Как быстро бак пустеет. Насос выключен, работают одни капельницы: минус 2 л в минуту. От 200 л до 50 л – те же 150 л: $150 : 2 = 75$ мин.

Шаг 5. Рисуем картину. Полив начался с полного бака, значит, сначала 75 мин бак пустеет. Потом насос включается, 25 мин наполняет, выключается, снова 75 мин спада – и так по кругу. Один круг – $25 + 75 = 100$ мин.

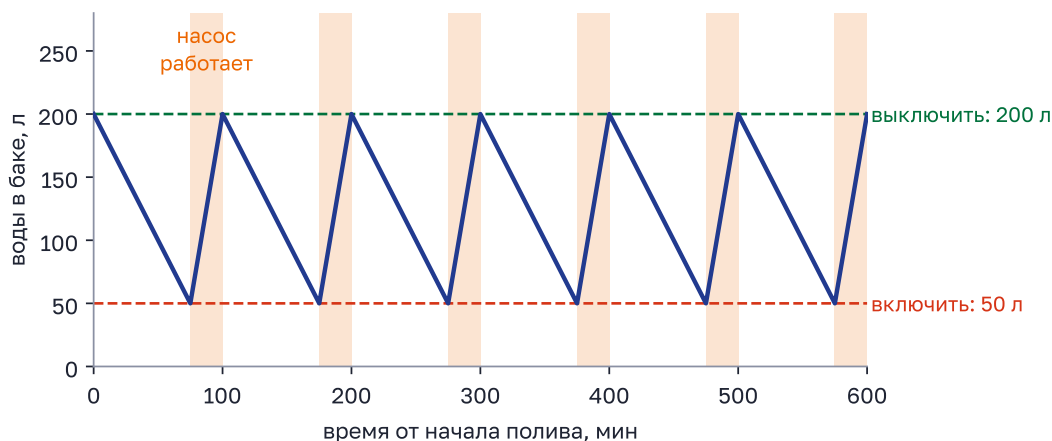


Рисунок 4.10 – Уровень воды в баке за 10 часов полива. Оранжевые полосы – насос работает

Шаг 6. Считаем включения. 10 ч = 600 мин. Насос включается на 75-й минуте, потом каждые 100 мин: на 75, 175, 275, 375, 475 и 575-й минуте. Следующее включение было бы на 675-й минуте – это уже после конца полива. Всего 6 включений. Последнее закончится как раз на 600-й минуте.

На графике это видно сразу: шесть «зубцов» вверх. А насос работает только четверть времени – 25 мин из каждых 100.

⚠ Типичная ошибка. Забыть про капельницы, пока бак наполняется, и посчитать $150 : 8 = 18,75$ мин. Капельницы не выключаются, поэтому бак растёт на 6 л в минуту, а не на 8. Вторая ошибка – поделить 600 на 100 и сразу ответить «6», не глядя на начало. Здесь совпало, но при поливе 9 ч 40 мин деление даёт 5,8, а включений всё равно 6: последнее – на 575-й минуте. Надёжнее выписать минуты включений по порядку.

Ответ: а) 25 мин; б) 6 раз.

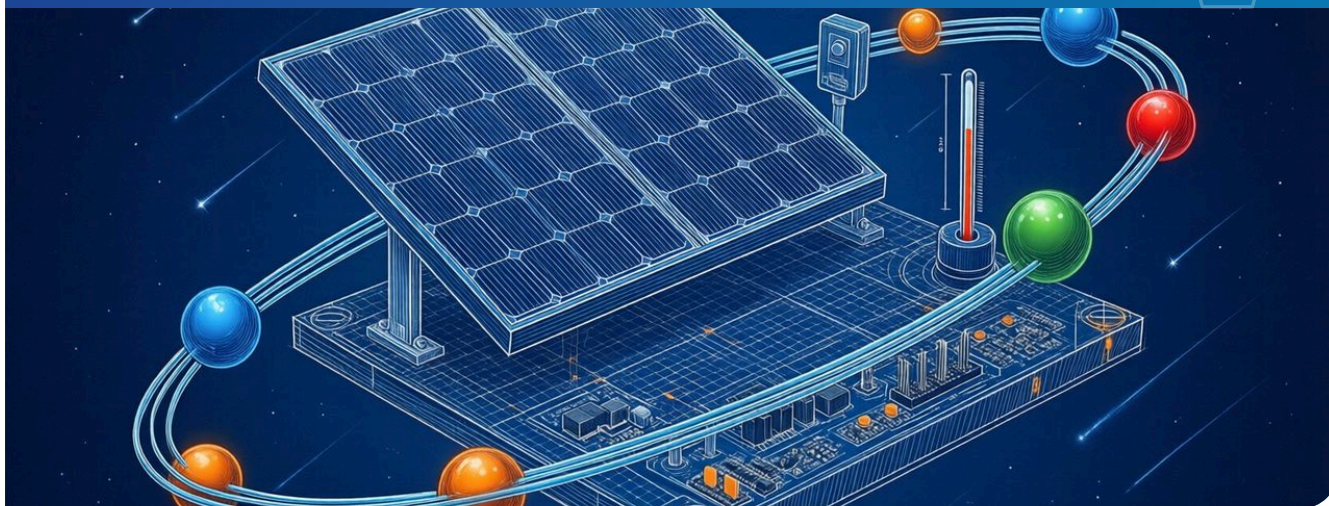
Своя задача по программе сферы, модуль 7

Реши сам

4.11. Бак на 120 л полон. Капельницы берут из него 3 л/мин, а насос одновременно подкачивает 1 л/мин. Через сколько минут бак опустеет?

4.12*.* В теплице 36 капельниц по 4 л/ч, они работают без перерыва. Насос на 10 л/мин включается, когда в баке 30 л, и выключается, когда 180 л. Сколько минут насос работает за одно включение? Округли до десятых. Какую часть времени (в процентах, до целых) насос работает, если считать полный круг «наполнение + спад»?

Электричество, энергия и датчики



Умная теплица – это электричество: оно бежит по проводам к датчикам, крутит насос, греет воздух ночью. В этой главе ты разберёшься, что такое ток, напряжение и сопротивление, посчитаешь резистор для светодиода и узнаешь, как термометр превращает градусы в число от 0 до 1023. А ещё – сколько энергии даёт солнечная панель и сколько её уходит на обогрев зимой. На отборе такие задачи выглядят как короткая история про схему или прибор, а в ответ вводится число с единицей или номер подходящего варианта.

5.1. Цепь: ток, напряжение, сопротивление

НА ПАЛЬЦАХ

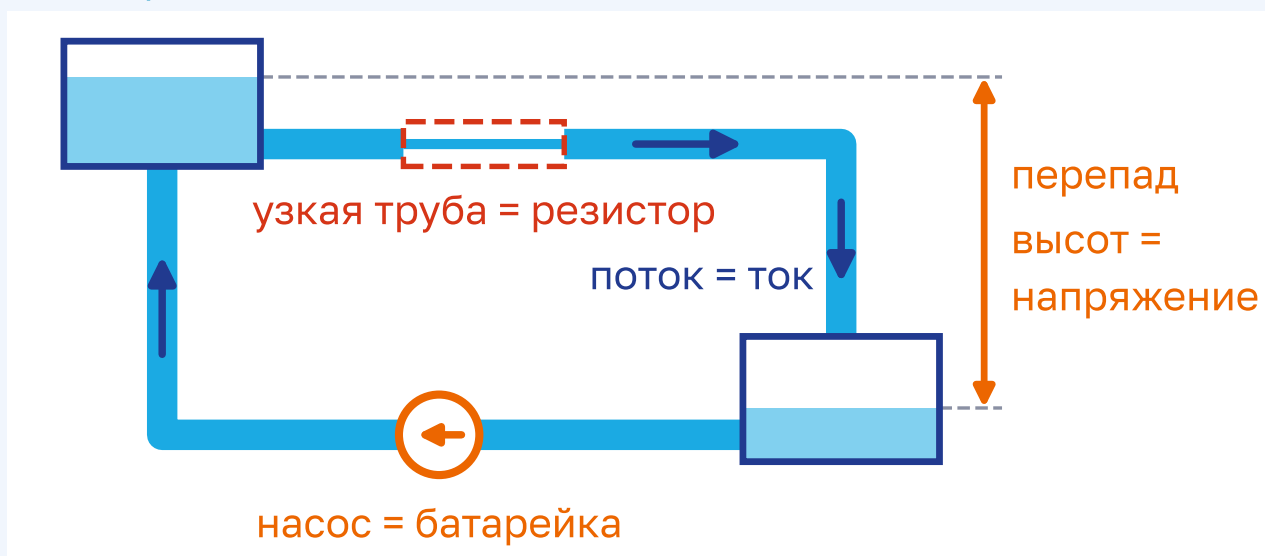


Рисунок 5.1 – Электрическая цепь похожа на водопровод с насосом

Насос поднимает воду наверх, вода стекает по трубе и снова попадает к насосу.
Батарейка делает с зарядами то же, что насос с водой.

Электричество руками не потрогаешь (и не надо), поэтому его удобно сравнивать с водой. По проводам движутся крошечные заряженные частицы. Их упорядоченное движение называют **электрическим током**. Ток идёт только по **замкнутой цепи** – когда от плюса источника до минуса есть сплошной путь из проводов и деталей. Разорвали провод в одном месте – ток пропал везде, как вода в перекрытой трубе.

Три величины описывают любую цепь.

- **Сила тока I** – сколько заряда пробегает через провод за секунду, как литры воды в секунду в трубе. Измеряют в **амперах (А)**. Для светодиода 1 А – очень много, поэтому в электронике в ходу **миллиамперы**: 1 А = 1000 мА. Светодиоду хватает 10–20 мА, телефон заряжается током около 1–2 А.
- **Напряжение U** – насколько сильно источник «толкает» заряды, как перепад высот между баками. Измеряют в **вольтах (В)**. Пальчиковая батарейка – 1,5 В, плата Arduino даёт датчикам 5 В, в розетке около 220 В – это опасно для жизни, с розеткой никаких опытов.
- **Сопротивление R** – насколько деталь мешает току, как узкое место в трубе. Измеряют в **омах (Ом)**; 1 кОм (килоом) = 1000 Ом. Детали, у которых главная работа – сопротивляться, называют **резисторами**.

Источник (батарейка, аккумулятор, плата Arduino, солнечная панель) создаёт напряжение. У него два вывода: плюс и минус. На Arduino минус подписан **GND** (от английского ground – «земля»), его напряжение считают за 0 В. Провода на схемах рисуют линиями, резистор по российским правилам – прямоугольником.

Если соединить плюс и минус просто проводом, без деталей, сопротивление почти нулевое, и ток становится огромным. Это **короткое замыкание**: провода греются, батарейка садится за минуты, а плата может сгореть. Светодиод, включённый в 5 В напрямую, тоже почти не мешает току. Tinkercad в таком случае ругается: ток во много раз больше допустимых 20 мА. Настоящий светодиод просто перестанет светить – навсегда.

И ещё одна деталь про светодиод: ток идёт через него только в одну сторону. Длинная ножка (анод) подключается к плюсу, короткая (катод) – к минусу. Перевернёшь – не загорится, хотя схема та же.

Запомни. Ток – поток зарядов, амперы. Напряжение – «толкание», вольты. Сопротивление – помеха току, омы. Ток идёт только по замкнутой цепи. 1 А = 1000 мА, 1 кОм = 1000 Ом.

5.2. Закон Ома: считаем резистор

НА ПАЛЬЦАХ



Рисунок 5.2 – 5 вольт – как горка высотой в 5 ступенек

Заряд скатывается от +5 В до 0 В. Три ступеньки забирает резистор, две – светодиод. Вместе всегда пять.

Возьмём водопровод с картинки из раздела 5.1. Поднимешь верхний бак выше – вода побежит быстрее. Сделаешь трубу уже – медленнее. С током то же самое: **ток тем больше, чем больше напряжение, и тем меньше, чем больше сопротивление**. Это **закон Ома** (Георг Ом – немецкий физик, открыл его в XIX веке). Буквами:

$$I = \frac{U}{R},$$

где I – ток в амперах, U – напряжение в вольтах, R – сопротивление в омах. Из этой формулы получаются ещё две: $U = I \cdot R$ и $R = U/I$. Пример: резистор 1 кОм подключили к 5 В. Сна-

чала переводим: $1 \text{ кОм} = 1000 \text{ Ом}$. Ток $I = 5/1000 = 0,005 \text{ А} = 5 \text{ мА}$.

Единицы в формуле Ома должны быть основными: вольты, амперы, омы. Если подставить миллиамперы или килоомы как есть, ответ уедет в тысячу раз.

Резистор для светодиода. Вывод Arduino даёт 5 В, а светодиоду нужно около 2 В и ток не больше 20 мА. (Это и рабочий предел вывода Arduino: в паспорте платы написано около 40 мА, но дольше живёт тот, кто держится в 20.) Лишние вольты должны «упасть» на резисторе – он стоит в цепи перед светодиодом (рисунок ниже). Когда детали соединены одна за другой, ток через них один и тот же, а напряжения складываются: $2 + 3 = 5 \text{ В}$. Считаем в три шага:

- напряжение на резисторе: $5 - 2 = 3 \text{ В}$;
- ток: $20 \text{ мА} = 0,02 \text{ А}$;
- сопротивление: $R = 3/0,02 = 150 \text{ Ом}$.

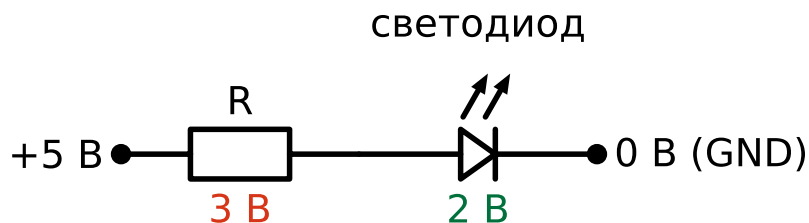


Рисунок 5.3 – Светодиод с резистором: из 5 В резистор берёт 3 В, светодиод – 2 В

Резисторы продают не всех номиналов: в наборах для Arduino почти всегда есть 220 Ом, 1 кОм и 10 кОм. Если нужного нет, берут **ближайший больший**. С 220 Ом ток станет $3/220 = 0,0136 \text{ А} \approx 13,6 \text{ мА}$ – светодиод чуть тусклее, зато цел. Меньший резистор брать нельзя: ток вырастет выше 20 мА.

НА ПАЛЬЦАХ

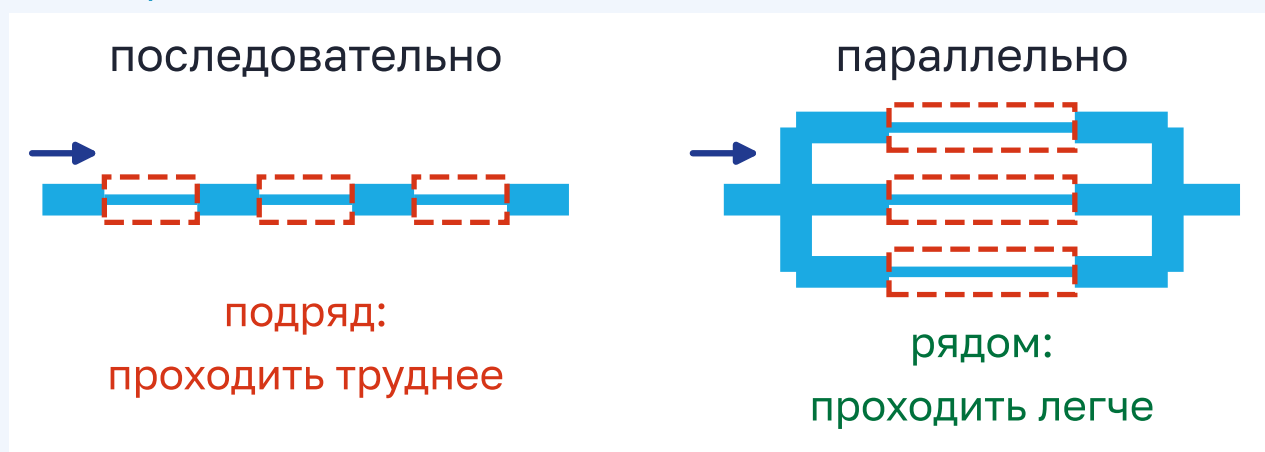


Рисунок 5.4 – Три узких места подряд на одной трубе и три узкие трубы рядом

Воде, которой надо протиснуться через три сужения подряд, труднее, чем через одно. А если узкие трубы стоят рядом, вода течёт сразу по всем, и проходит её больше.

Последовательно и параллельно. Резисторы можно соединить двумя способами (рисунок ниже).

- **Последовательно** – один за другим, как вагоны поезда. Ток на всех один, сопротивления складываются: $R = R_1 + R_2 + R_3$. Три резистора по 1 кОм дают 3 кОм.
- **Параллельно** – рядом, как полосы на шоссе. Напряжение на всех одинаковое, а ток делится между ними. Общее сопротивление **меньше самого маленького**: у тока появилось несколько дорог.

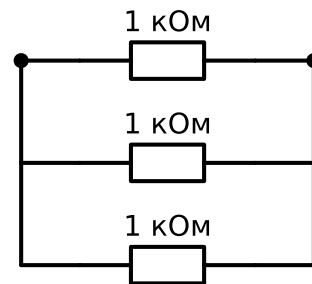
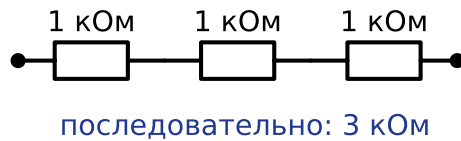


Рисунок 5.5 – Три одинаковых резистора: последовательно и параллельно

Для одинаковых резисторов считать совсем просто: **n одинаковых резисторов параллельно дают в n раз меньше одного**. Три по 1 кОм – $1000/3 \approx 333$ Ом, два по 1 кОм – 500 Ом, четыре по 220 Ом – 55 Ом. В Tinkercad это легко проверить мультиметром – прибором, который измеряет ток, напряжение и сопротивление.

Для любопытных. Когда резисторы разные, общее сопротивление параллельного соединения ищут через дроби: $\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$. Для 300 Ом и 600 Ом:
 $\frac{1}{300} + \frac{1}{600} = \frac{2}{600} + \frac{1}{600} = \frac{3}{600} = \frac{1}{200}$, то есть $R = 200$ Ом. В школе эту формулу проходят в 8 классе. Для задач этой главы хватит правила про одинаковые резисторы.

Запомни. $I = U/R$ – в вольтах, амперах и омах. Резистор для светодиода:
 $R = (5 - U_{\text{св}})/I$, номинал берём ближайший больший. Последовательно – сопротивления складываются, параллельно – общее меньше самого маленького (n одинаковых – в n раз меньше одного).

Совет. Посчитал ток – сравни с жизнью. Светодиоду нужны единицы и десятки миллиампер. Получилось 3 А или 0,00003 мА – где-то потерялся перевод кОм в Ом или мА в А.

5.3. Датчик переводит мир в вольты

НА ПАЛЬЦАХ



Рисунок 5.6 – Цепочка превращений: свет – сопротивление – напряжение – число

Arduino не видит свет и не чувствует тепло. Она умеет только мерить напряжение. Датчик переводит свет или тепло на её язык – в вольты.

Датчик (сенсор) – деталь, которая превращает то, что нам нужно узнать (освещённость, температуру, влажность почвы), в электрический сигнал. В теплице из курса сферы главных датчиков три.

- **Фоторезистор** – резистор, у которого сопротивление зависит от света. Чем светлее, тем оно меньше: в темноте сотни килоом и больше, на ярком свете – единицы килоом и меньше.
- **Аналоговый термометр** – маленькая микросхема с тремя ножками. Она сама выдаёт напряжение, которое растёт с температурой. В симуляторе Tinkercad это **TMP36**, в наборах с Arduino часто встречается **LM35**.
- **Датчик влажности почвы** – вилка из двух пластин, её втыкают в землю. Во влажной земле сигнал один, в сухой – другой.

Делитель напряжения. Сопротивление Arduino мерить не умеет, поэтому фоторезистор соединяют последовательно с обычным резистором (часто 10 кОм) и подключают к 5 В. Точку между ними ведут на аналоговый вход A0 (рисунок ниже). Так собрана схема в курсе: резистор 10 кОм – между +5 В и A0, фоторезистор – между A0 и GND. Ток через обе детали одинаковый, и 5 В делятся между ними **пропорционально сопротивлениям**: у кого больше сопротивление, тому больше вольт. Посчитать напряжение на входе можно в два шага:

$$I = \frac{5}{R_{\phi} + R}, \quad U_{A0} = I \cdot R_{\phi},$$

где R_{ϕ} – сопротивление фоторезистора (нижняя деталь, между A0 и GND), R – верхний резистор (между +5 В и A0). Вход A0 меряет напряжение на нижней детали. Пример: ярко, у фоторезистора 2,5 кОм. $I = 5 / (2500 + 10\,000) = 0,0004$ А, $U_{A0} = 0,0004 \cdot 2500 = 1$ В. В темноте сопротивление вырастет, и на нём окажется большая часть из 5 В. В этой схеме **больше света – меньше напряжение, меньше число; темнота – большое число**. В симуляторе Tinkercad у курса получилось 49 на ярком свете и 969 в темноте. Поменяешь детали местами – станет наоборот.

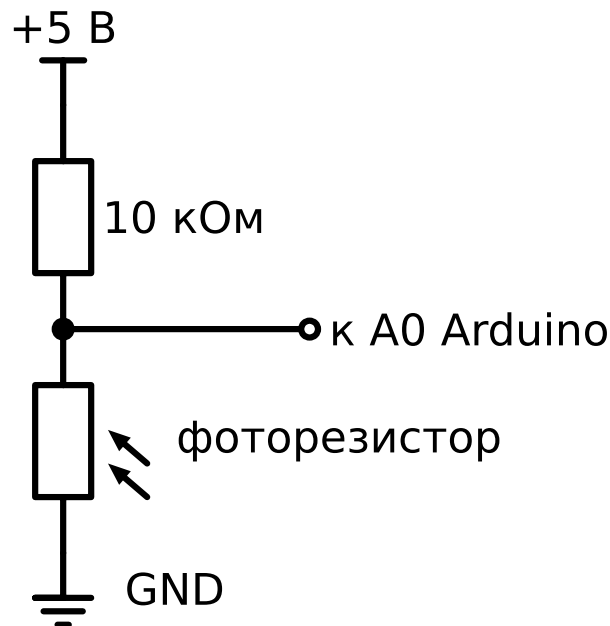


Рисунок 5.7 – Фоторезистор в делителе напряжения. Середина делителя идёт на аналоговый вход A0

НА ПАЛЬЦАХ

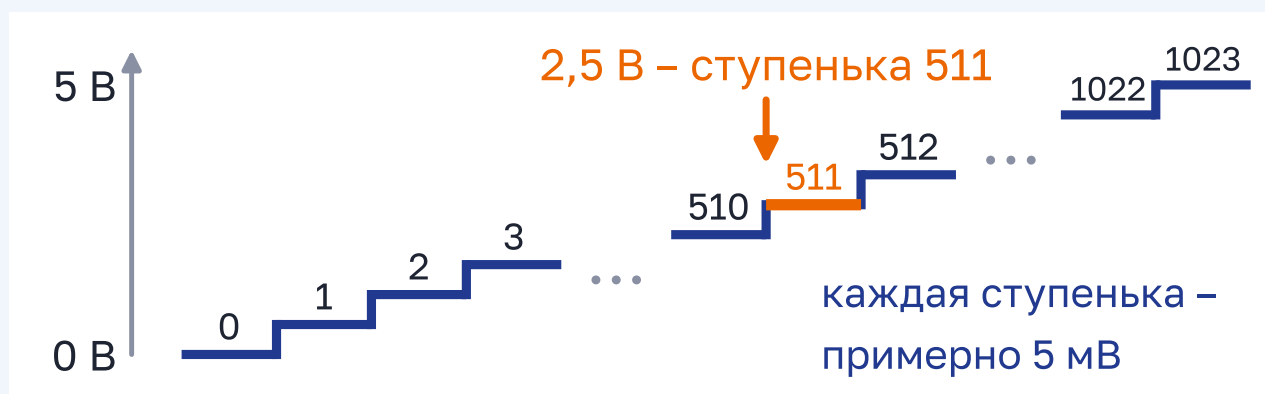


Рисунок 5.8 – Лестница от 0 до 5 В из 1024 ступенек. Напряжение 2,5 В попадает на ступеньку номер 511

Прибор внутри платы работает как счётчик ступенек: докуда дотянулось напряжение, такой номер он и выдаст.

Число `analogRead`. Внутри Arduino есть **АЦП** – аналого-цифровой преобразователь (переводчик из вольт в числа). Он делит промежуток от 0 до 5 В на 1024 ступеньки и сообщает номер ступеньки, на которой оказалось напряжение. Нумерация начинается с нуля, поэтому команда `analogRead(A0)` возвращает целое число от 0 (0 В) до 1023 (5 В). Одна ступенька – примерно $5/1023 \approx 0,0049$ В, около 5 милливольт (милливольт – тысячная доля вольта).

Переводы туда и обратно – обычная пропорция:

$$U = N \cdot \frac{5}{1023}, \quad N = \frac{U}{5} \cdot 1023,$$

где N – показание `analogRead`. Во второй формуле дробную часть **отбрасывают**: АЦП ничего не округляет – он сообщает номер ступеньки, до которой напряжение дотянулось. 2,5 В – это $2,5/5 \cdot 1023 = 511,5$, а `analogRead` покажет 511. В некоторых примерах и учебниках делят не на 1023, а на 1024 – разница крошечная. На отборе делай так, как сказано в условии.

Для любопытных. Откуда взялось число 1024? Номер ступеньки АЦП записывает десятию двоичными цифрами – нулями и единицами, их называют битами. Десятью такими цифрами можно записать $2^{10} = 1024$ разных числа, от 0 до 1023.

Из вольт – в градусы. У TMP36 и у LM35 напряжение растёт на 10 мВ (0,01 В) на каждый градус. Разница в нуле: LM35 при 0 °C выдаёт 0 В, а TMP36 – уже 0,5 В. Так TMP36 может показывать мороз: при –10 °C у него 0,4 В.

$$\text{TMP36: } t = (U - 0,5) \cdot 100, \quad \text{LM35: } t = U \cdot 100,$$

где t – температура в °C, U – напряжение в вольтах. Пример: TMP36 выдаёт 0,75 В, $t = (0,75 - 0,5) \cdot 100 = 25$ °C. Одна ступенька АЦП – это 5 мВ, то есть половина градуса. Точнее Arduino такой термометр не прочтает.

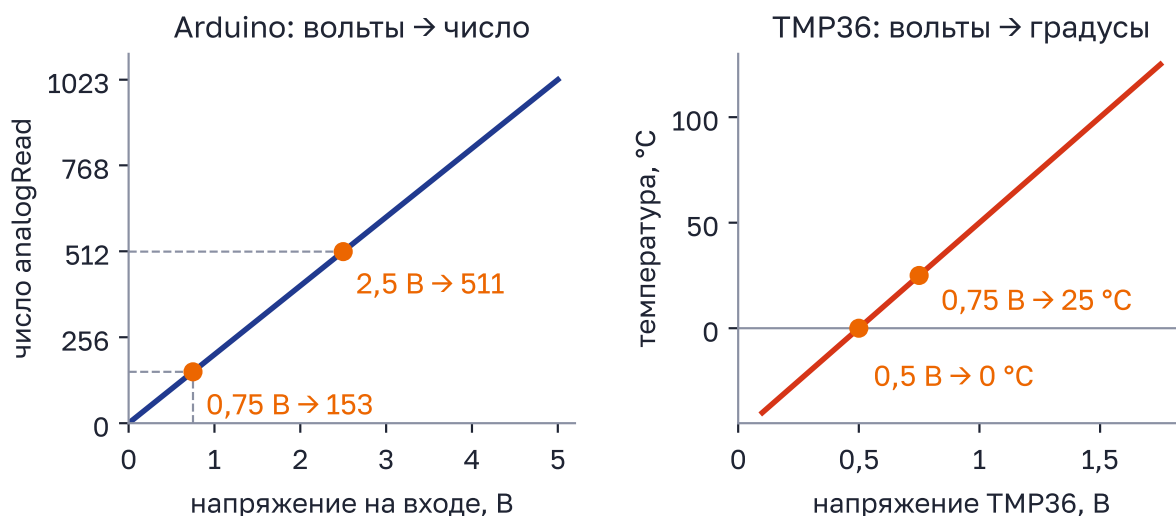


Рисунок 5.9 – Два перевода: напряжение на входе – в число analogRead, напряжение TMP36 – в градусы

Так выглядит программа для Arduino, которая читает TMP36 и печатает температуру. Всё, что после `//`, – пояснения для человека, плата их не читает.

```
void setup() {  
  Serial.begin(9600);           // включаем связь с компьютером  
}  
  
void loop() {  
  int n = analogRead(A0);       // число от 0 до 1023  
  float u = n * 5.0 / 1023;     // переводим в вольты  
  float t = (u - 0.5) * 100;    // вольты TMP36 – в градусы  
  Serial.println(t);            // печатаем в монитор порта  
  delay(1000);                 // ждём секунду  
}
```

`int` – целое число, `float` – число с дробной частью. Обратите внимание на `5.0` с точкой (в программах дробную часть отделяют точкой). Если написать `n * 5 / 1023`, компьютер посчитает всё в целых числах и отбросит дробь: при $n = 153$ получится 0 вольт вместо 0,748.

НА ПАЛЬЦАХ



Рисунок 5.10 – Полоска от «сухо» (180) до «мокро» (820). Показание 500 стоит посередине – это 50 %

Влажность в процентах – это то, какую часть пути от «сухо» до «мокро» уже прошло показание датчика.

Датчик влажности почвы и калибровка. Для почвы готовой формулы нет: показания зависят от датчика и от земли. Поэтому датчик **калибруют** – записывают показания в двух известных точках. Воткнули в совсем сухую землю – $N_{\text{сух}}$, полили досыта – $N_{\text{мокр}}$. Далее влажность в процентах – это доля пути от «сухо» к «мокро»:

$$W = \frac{N - N_{\text{сух}}}{N_{\text{мокр}} - N_{\text{сух}}} \cdot 100.$$

Пример: сухо – 180, мокро – 820, сейчас – 500.

$W = (500 - 180) / (820 - 180) \cdot 100 = 320 / 640 \cdot 100 = 50 \%$. У многих датчиков во влажной земле число, наоборот, падает – формула работает и тогда: и числитель, и знаменатель станут отрицательными, а доля выйдет положительной. В Arduino то же делает команда `map(N, 180, 820, 0, 100)`, только она считает в целых числах и отбрасывает дробную часть.

Запомни. Датчик превращает свет, тепло, влажность в напряжение, Arduino – напряжение в число 0...1023. $U = N \cdot 5 / 1023$; обратно $N = U / 5 \cdot 1023$ с отброшенной дробью. TMP36: $t = (U - 0,5) \cdot 100$. Почву калибруют по двум точкам: «сухо» и «мокро».

Совет. Держи в голове опорные точки: 0 В – 0, 2,5 В – 511, 5 В – 1023; у TMP36 0,5 В – это 0 °С, 0,75 В – 25 °С. Сверх с ними ответ, и грубая ошибка сразу вылезет.

5.4. Мощность и энергия: солнечная панель

НА ПАЛЬЦАХ

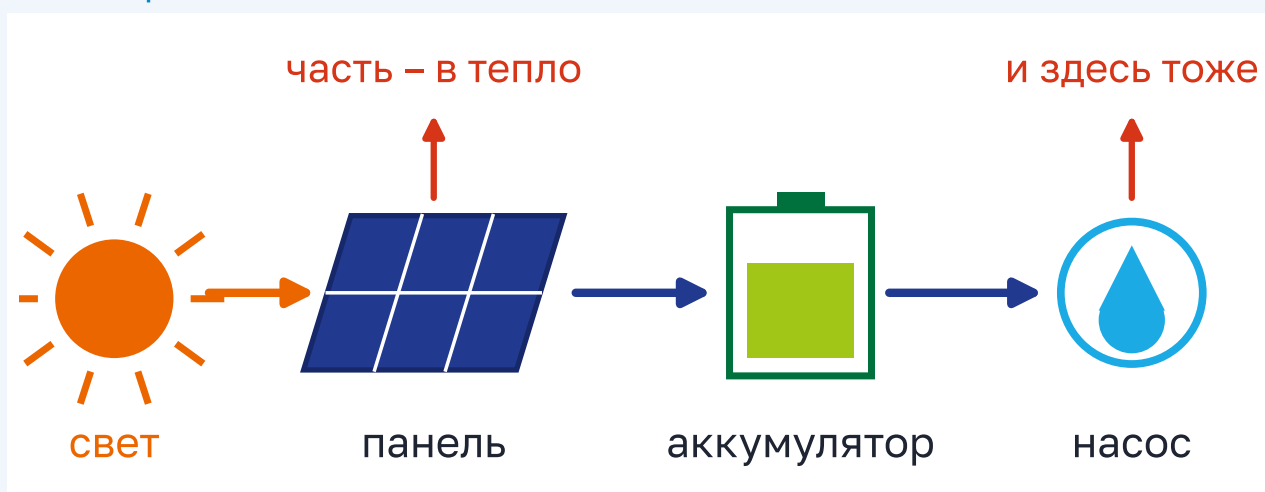


Рисунок 5.11 – Путь энергии: свет – электричество – работа насоса. На каждом шаге часть уходит в тепло

Энергия не появляется из ниоткуда и не исчезает. Она только переходит из одного вида в другой.

Энергия – способность сделать работу: поднять воду, раскрутить вентилятор, нагреть воздух. Она переходит из вида в вид. Солнечная панель превращает свет в электрическую энергию, насос – электрическую в движение воды, лампа – в свет, обогреватель – в тепло. И почти на каждом шаге часть энергии уходит в тепло без пользы: панель на солнце горячая, мотор насоса после работы тёплый.

Мощность P – сколько энергии прибор тратит (или выдаёт) за одну секунду. Измеряют в **ваттах** (Вт); 1 кВт = 1000 Вт. Для электрического прибора мощность равна напряжению, умноженному на ток:

$$P = U \cdot I.$$

Насос на 12 В, который берёт 0,5 А, работает с мощностью $12 \cdot 0,5 = 6$ Вт. Светодиодная лампочка дома – около 10 Вт, электрочайник – около 2000 Вт.

Энергия за время – мощность, умноженная на время: $E = P \cdot t$. Если мощность в ваттах, а время в часах, энергия выйдет в **ватт-часах** (Вт·ч). Домашний счётчик считает **киловатт-часы**: 1 кВт·ч = 1000 Вт·ч. Чайник мощностью 2 кВт за полчаса тратит $2 \cdot 0,5 = 1$ кВт·ч. За эти киловатт-часы семья и платит.

Аккумулятор запасает энергию. На нём пишут **ёмкость** в миллиампер-часах (мА·ч): сколько времени он может отдавать какой-то ток. Время работы = ёмкость : ток. Аккумулятор 2000 мА·ч при токе 250 мА продержится $2000/250 = 8$ ч. Единицы должны совпадать: мА·ч делим на мА. Если ёмкость в А·ч, сначала переведи: 2,4 А·ч = 2400 мА·ч.

Солнечная панель. В ясный полдень на каждый квадратный метр, повернутый к солнцу, приходит около 1000 Вт солнечного света. Панель превращает в электричество только часть – у обычных кремниевых панелей примерно 15–20 %. Эту долю называют **КПД** – коэффициент полезного действия:

$$\text{КПД} = \frac{\text{полезная мощность}}{\text{затраченная мощность}} \cdot 100 \, \%.$$

Панель площадью 1 м² с КПД 18 % в ясный полдень даст 1000 · 0,18 = 180 Вт. Утром, вечером, в облака и зимой – намного меньше. Поэтому в задачах часто говорят: «за день панель даёт столько же, сколько за 4 часа полной мощности», и энергия за сутки – это полная мощность, умноженная на эти часы. Солнечный трекер – механизм, который поворачивает панель вслед за солнцем, – нужен, чтобы полной мощности было побольше: к панели, повернутой боком, света приходит меньше.

Запомни. $P = U \cdot I$ (ватты = вольты · амперы). $E = P \cdot t$ (Вт · ч = Вт·ч). Время работы аккумулятора = ёмкость : ток. КПД – доля полезной энергии, у солнечной панели около 15–20 %.

5.5. Тепло: обогрев и теплоизоляция

НА ПАЛЬЦАХ

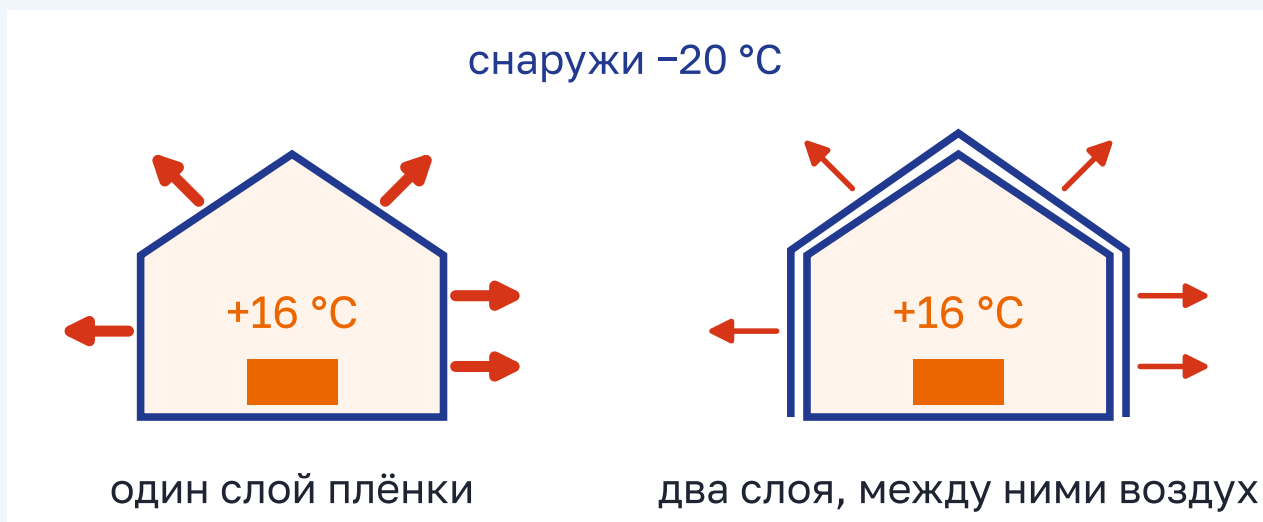


Рисунок 5.12 – Зимней ночью тепло уходит из теплицы сквозь плёнку. Второй слой с воздухом внутри держит его лучше

Тепло всегда течёт от тёплого к холодному, как вода с горки. Горку – разницу температур – утепление не меняет. Оно сужает дорогу, по которой тепло бежит наружу.

Теплица в Новосибирской области весной и осенью борется с холодом. Тепло сквозь стенки уходит на улицу само, пока температуры внутри и снаружи не сравняются. Путь у него три.

- **Теплопроводность** – тепло проходит сквозь вещество. Металлическая ложка в горячем чае через минуту обжигает пальцы. Металл пропускает тепло хорошо, дерево, пенопласт и неподвижный воздух – плохо.
- **Конвекция** – тепло переносит движущийся воздух или вода. Тёплый воздух легче холодного и поднимается вверх, поэтому форточки в теплице делают под крышей: через них уходит самый горячий воздух.

- **Излучение** – тепло передаётся лучами, даже через пустоту. Так Солнце греет Землю. Солнечные лучи проходят сквозь стекло и плёнку, нагревают землю и растения, и теплица прогревается.

Теплоизоляция – всё, что мешает теплу уходить. Лучший дешёвый изолятор – неподвижный воздух. Поэтому теплицы обтягивают двумя слоями плёнки с воздушной прослойкой, а окна в домах делают с двумя-тремя стёклами. По той же причине пушистый снег укрывает озимые: между снежинками много воздуха.

Сколько тепла уходит. Чем больше разница температур внутри и снаружи, тем быстрее уходит тепло. В задачах это считают по простой модели: теплица теряет каждую секунду столько-то ватт **на каждый градус разницы**. Это число (назовём его k) зависит от размеров теплицы и от утепления и даётся в условии:

$$P_{\text{потерь}} = k \cdot (t_{\text{внутри}} - t_{\text{снаружи}}).$$

Пример: маленькая теплица теряет 100 Вт на градус. Внутри нужно +15 °С, снаружи –5 °С. Разница – не 10, а $15 - (-5) = 20$ °С. Потери: $100 \cdot 20 = 2000$ Вт.

Обогрев. Электрический обогреватель превращает в тепло всю энергию, которую берёт из сети. Чтобы температура в теплице не падала, он должен давать столько тепла, сколько уходит наружу: мощность обогревателя не меньше мощности потерь. Энергия за ночь – та же формула $E = P \cdot t$. Утепили теплицу, k стал меньше – во столько же раз меньше и счёт за электричество.

Запомни. Тепло идёт от тёплого к холодному: сквозь стенки, с воздухом, лучами. Потери = $k \cdot$ разность температур. Чтобы держать температуру, обогреватель должен покрывать потери. Мороз вычитай со знаком: между +10 °С и –10 °С двадцать градусов, а не ноль.

5.6. Разбираем задачи

Задача 5.1 Жёлтый огонёк насоса

разминка · закон Ома, резистор для светодиода, номинал

На пульте теплицы жёлтый светодиод показывает, что насос работает. Светодиод подключён к выводу Arduino (5 В) через резистор. По паспорту светодиоду нужно 2,1 В и ток 20 мА. а) Какое сопротивление должно быть у резистора? б) В наборе есть резисторы 100 Ом, 220 Ом и 1 кОм. Какой из них взять?

Дано: $U = 5$ В; $U_{\text{св}} = 2,1$ В; $I = 20$ мА; набор: 100 Ом, 220 Ом, 1 кОм.
Найти: R – ?; номинал из набора.

Решение

Шаг 1. Напряжение на резисторе. Светодиод и резистор стоят один за другим, 5 В делятся между ними: $U_R = 5 - 2,1 = 2,9$ В.

Шаг 2. Ток – в амперы. 20 мА = $20 : 1000 = 0,02$ А.

Шаг 3. Закон Ома. $R = U_R / I = 2,9 / 0,02 = 145 \text{ Ом}$.

Шаг 4. Выбираем из набора. 145 Ом нет. Берём ближайший больший – 220 Ом. Проверим, что будет с каждым:

- 100 Ом: $I = 2,9 / 100 = 0,029 \text{ А} = 29 \text{ мА}$ – больше 20, светодиод перегреется;
- 220 Ом: $I = 2,9 / 220 = 0,0132 \text{ А} \approx 13,2 \text{ мА}$ – горит чуть тусклее, но надёжно;
- 1 кОм: $I = 2,9 / 1000 = 0,0029 \text{ А} = 2,9 \text{ мА}$ – светится еле-еле, на пульте днём не разглядишь.

⚠ Типичная ошибка. Подставить в формулу все 5 В: $5 / 0,02 = 250 \text{ Ом}$. Так считают, будто светодиода в цепи нет. А если оставить ток в миллиамперах, выйдет $2,9 / 20 = 0,145 \text{ Ом}$ – резистор, который почти ничего не ограничивает. Ещё одна ловушка – округлить 145 до 100 «по-честному, к ближайшему». Ближайший здесь не годится, нужен больший.

Ответ: а) 145 Ом; б) 220 Ом.

Своя задача по программе сферы, модуль 1

Реши сам

5.1. К батарейному отсеку на 6 В подключили резистор. Мультиметр показал ток 12 мА. Какое сопротивление у резистора? Ответ в омах.

5.2. Красный светодиод (2 В) включили в 5 В через резистор 220 Ом. Какой ток идёт через светодиод? А какой пойдёт, если по ошибке поставить резистор 22 Ом? Ответы в миллиамперах, округли до десятых. Выдержит ли светодиод, если ему можно не больше 20 мА?

Задача 5.2 Резисторы из одной коробки

последовательное и параллельное соединение, закон Ома

Для опыта нужен резистор 1,5 кОм, а в коробке только резисторы по 1 кОм. Их соединили так, как на рисунке: резистор R1, а за ним – пара R2 и R3, соединённых параллельно. Всё подключили к 5 В. а) Какое сопротивление у всей схемы? Ответ в килоомах. б) Какой ток идёт через R1? Ответ в миллиамперах, округли до десятых.

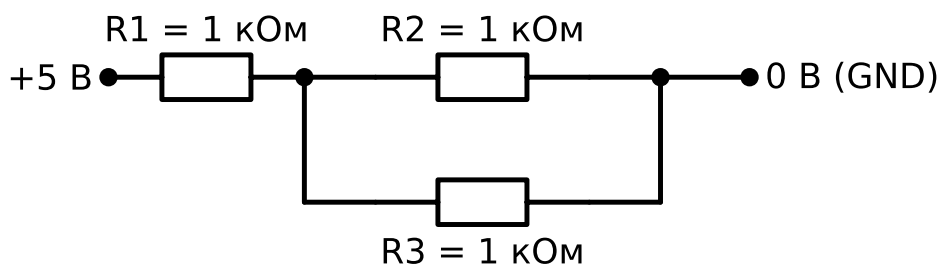


Рисунок 5.13 – Схема к задаче 5.2: R1 соединён последовательно с парой параллельных резисторов R2 и R3

Дано: $R_1 = R_2 = R_3 = 1 \text{ кОм} = 1000 \text{ Ом}$; $U = 5 \text{ В}$.

Найти: $R_{\text{общ}}$ - ?; I - ?

Решение

Шаг 1. Начинаем с параллельной пары. Два одинаковых резистора параллельно дают вдвое меньше одного: $R_{23} = 1000/2 = 500 \text{ Ом}$. У тока две дороги, поэтому пара пропускает его легче, чем один резистор.

Шаг 2. Пара стоит последовательно с R1. Сопротивления складываются:

$R_{\text{общ}} = 1000 + 500 = 1500 \text{ Ом} = 1,5 \text{ кОм}$. Нужный номинал собран.

Шаг 3. Ток. Весь ток проходит через R1, значит, это ток всей схемы: $I = 5/1500 = 0,00333 \text{ А} \approx 3,3 \text{ мА}$.

Дальше он делится: через R2 и через R3 идёт по половине, около 1,7 мА. А напряжение на R1 – $0,00333 \cdot 1000 \approx 3,3 \text{ В}$, на паре – остальные 1,7 В. В сумме снова 5 В.

❗ **Типичная ошибка.** Сложить все три резистора: 3 кОм. Сложение годится только для деталей, стоящих одна за другой, а R2 и R3 стоят рядом. Вторая ловушка – посчитать ток через R2 по полному напряжению: $5/1000 = 5 \text{ мА}$. На паре только часть напряжения, 1,7 В, остальное забрал R1.

Ответ: а) 1,5 кОм; б) 3,3 мА.

Своя задача по программе сферы, модуль 1

Реши сам

5.3. Есть четыре резистора по 220 Ом. Какое сопротивление получится, если соединить их все последовательно? А если все параллельно? Ответы в омах.

5.4. Резисторы 300 Ом и 200 Ом соединили последовательно и подключили к 5 В. Какой ток идёт по цепи (в мА)? Какое напряжение на каждом резисторе?

Задача 5.3 Что показывает термометр

analogRead, перевод числа в вольты и градусы, TMP36

В теплице стоит термометр TMP36, подключённый к входу A0 платы Arduino. Программа прочитала `analogRead(A0)` и получила 164. а) Какая температура в теплице? Округли до десятых. б) Обогреватель должен включаться при 18 °С. Какое число покажет `analogRead`, когда в теплице 18 °С? Для перевода используй $U = N \cdot 5/1023$ и $t = (U - 0,5) \cdot 100$.

Дано: $N_1 = 164$; $t_2 = 18 \text{ °С}$; TMP36: $t = (U - 0,5) \cdot 100$.

Найти: t_1 - ?; N_2 - ?

Решение

Шаг 1. Число – в вольты. $U_1 = 164 \cdot 5/1023 = 820/1023 = 0,8016 \text{ В}$.

Шаг 2. Вольты – в градусы. $t_1 = (0,8016 - 0,5) \cdot 100 = 0,3016 \cdot 100 = 30,16 \approx 30,2 \text{ }^{\circ}\text{C}$.
Жарковато – пора открывать форточку.

Шаг 3. Обратный путь: градусы – в вольты. Из $t = (U - 0,5) \cdot 100$: сначала делим на 100, потом прибавляем 0,5. $U_2 = 18/100 + 0,5 = 0,18 + 0,5 = 0,68 \text{ В}$.

Шаг 4. Вольты – в число. $N_2 = 0,68/5 \cdot 1023 = 0,136 \cdot 1023 = 139,128$. АЦП отбрасывает дробную часть: 139.

Прикидка по опорным точкам. 25 °С – это 0,75 В и число 153. 30 °С теплее – число должно быть больше 153, и 164 подходит. 18 °С холоднее 25 – и 139 меньше 153. Каждый градус – около двух ступенек АЦП: от 18 до 25 °С семь градусов, $153 - 139 = 14$ ступенек.

⚠ Типичная ошибка. Забыть вычесть 0,5: $0,8016 \cdot 100 = 80,2 \text{ }^{\circ}\text{C}$ – в теплице такого не бывает, это формула LM35, а не TMP36. На обратном пути путают порядок:
 $(18 + 0,5)/100 = 0,185 \text{ В}$. Проверь себя подстановкой: $(0,185 - 0,5) \cdot 100$ даёт минус 31,5 °С, а должно быть 18.

Ответ: а) 30,2 °С; б) 139.

Своя задача по программе сферы, модуль 6

Реши сам

5.5. Мартовская ночь, теплица ещё не отапливается. Программа с термометром TMP36 прочитала `analogRead(A0) = 82`. Какая температура в теплице? Округли до десятых.

5.6. Датчик влажности почвы откалибровали: в сухой земле он показал 870, в только что политой – 390. Сейчас он показывает 600. Какая влажность почвы в процентах? Дробную часть отбрось, как делает команда `map`.

Задача 5.4 *. Когда включать досветку

делитель напряжения, фоторезистор, analogRead, порог

Фоторезистор и резистор 10 кОм собраны в делитель, как на рисунке в разделе 5.3: резистор 10 кОм – между +5 В и входом A0, фоторезистор – между A0 и GND (так собрана схема в курсе). По паспорту у фоторезистора такое сопротивление:

Таблица 5.1

Освещённость	100 лк	1000 лк	5000 лк
Сопротивление фоторезистора	40 кОм	8 кОм	2,5 кОм

Лампы досветки должны гореть, когда света меньше 1000 лк. а) Какое число покажет `analogRead(A0)` при 1000 лк? Дробную часть отбрось. б) Какое условие включения ламп верное: 1) `analogRead(A0) < порог`; 2) `analogRead(A0) > порог`? В ответ – номер условия.

Дано: $U = 5$ В; $R = 10$ кОм; при 1000 лк $R_{\phi} = 8$ кОм; таблица сопротивлений.

Найти: N при 1000 лк; номер условия.

Решение

Шаг 1. Ток через делитель при 1000 лк. Фоторезистор и резистор стоят последовательно, сопротивления складываются: $8 + 10 = 18$ кОм = 18 000 Ом. $I = 5/18\,000 = 0,000278$ А.

Шаг 2. Напряжение на входе А0. Вход подключён к верхнему концу нижней детали, значит, на нём напряжение фоторезистора: $U_{A0} = I \cdot R_{\phi} = 5/18\,000 \cdot 8000 = 2,222$ В. Можно и пропорцией: фоторезистору достаётся 8/18 от 5 В – те же 2,222 В.

Шаг 3. Напряжение – в число. $N = 2,222/5 \cdot 1023 = 454,7$. Отбрасываем дробь: 454. Это и есть порог.

Шаг 4. В какую сторону сравнивать. Посчитаем число при 100 лк, когда темнее: $R_{\phi} = 40$ кОм, $U_{A0} = 5 \cdot 40/(40 + 10) = 4$ В, $N = 4/5 \cdot 1023 = 818,4$, то есть 818. При 5000 лк: $U_{A0} = 5 \cdot 2,5/(2,5 + 10) = 1$ В, $N = 204,6$, то есть 204. Темнее – больше число. Лампы нужны, когда темно, то есть когда число **больше** 454. Верно условие 2.

Проверка на здравый смысл: полутень в 1000 лк дала почти середину шкалы (454 из 1023) – датчик работает в удобной части, и порог не прилип к краю.

❗ **Типичная ошибка.** Сосчитать напряжение на верхнем резисторе, а не на фоторезисторе: $5 \cdot 10/18 = 2,78$ В и число 568. Вход А0 меряет напряжение между собой и GND – это напряжение нижней детали, фоторезистора. Вторая ошибка – перепутать направление: в этой схеме число растёт к темноте, а в схеме с деталями наоборот (задача 5.8*) – падает. Направление проверяй расчётом во второй точке, как в шаге 4.

Ответ: а) 454; б) 2.

Своя задача по программе сферы, модуль 6

Реши сам

5.7. Делитель тот же, что в задаче 5.4. В пасмурный день у фоторезистора 25 кОм. Какое число покажет `analogRead(A0)` ? Дробную часть отбрось.

5.8*.* Делитель из задачи 5.4 собрали наоборот: фоторезистор – между +5 В и А0, резистор 10 кОм – между А0 и GND. Какое число покажет `analogRead(A0)` при 1000 лк? Какое условие включения ламп досветки (света меньше 1000 лк) теперь верное: 1) `analogRead(A0) < порог` или 2) `analogRead(A0) > порог` ?

Задача 5.5 Хватит ли солнца метеостанции

КПД, мощность, энергия за сутки

Метеостанцию на краю поля питает солнечная панель размером 20 × 30 см с КПД 16 %. В ясный полдень на 1 м² приходит 1000 Вт солнечного света. В сентябре за день панель выраба-

тывает столько же, сколько за 4 часа работы на полной мощности. Метеостанция (плата, датчики, дисплей) круглые сутки берёт от аккумулятора 0,25 А при 5 В. а) Какую мощность даёт панель в ясный полдень? Ответ в ваттах. б) Хватит ли выработанной за день энергии на сутки работы? Сколько ватт-часов останется в запасе или не хватит? Потерями в аккумуляторе пренебреги.

Дано: панель $0,2 \times 0,3$ м; КПД 16 %; 1000 Вт/м^2 ; 4 ч полной мощности; $U = 5 \text{ В}$; $I = 0,25 \text{ А}$; 24 ч.

Найти: $P_{\text{пан}}$ – ?; запас энергии за сутки – ?

Решение

Шаг 1. Площадь – в квадратные метры. $20 \text{ см} = 0,2 \text{ м}$, $30 \text{ см} = 0,3 \text{ м}$. $S = 0,2 \cdot 0,3 = 0,06 \text{ м}^2$.

Шаг 2. Сколько света падает на панель. $1000 \cdot 0,06 = 60 \text{ Вт}$.

Шаг 3. Сколько из этого станет электричеством. 16 % – это 0,16: $P_{\text{пан}} = 60 \cdot 0,16 = 9,6 \text{ Вт}$. Остальные 50,4 Вт нагревают саму панель.

Шаг 4. Энергия за день. $E_{\text{пан}} = 9,6 \cdot 4 = 38,4 \text{ Вт} \cdot \text{ч}$.

Шаг 5. Сколько нужно метеостанции. Мощность $P = 5 \cdot 0,25 = 1,25 \text{ Вт}$. За сутки $E = 1,25 \cdot 24 = 30 \text{ Вт} \cdot \text{ч}$.

Шаг 6. Сравниваем. $38,4 - 30 = 8,4 \text{ Вт} \cdot \text{ч}$ – хватает, и немного остаётся. Запас около четверти от нужного: в облачный сентябрьский день его может и не хватить.

⚠ **Типичная ошибка.** Подставить площадь в квадратных сантиметрах: $20 \cdot 30 = 600$, и получить панель мощностью 96 000 Вт – больше, чем у целого дома. Вторая ошибка – забыть, что метеостанция работает 24 часа, а панель – только днём, и сравнить $38,4 \text{ Вт} \cdot \text{ч}$ с $1,25 \text{ Вт}$. Ватт-часы с ваттами не сравнивают.

Ответ: а) 9,6 Вт; б) хватит, запас 8,4 Вт·ч.

Своя задача по программе сферы, модуль 8

Реши сам

5.9. Солнечная панель выдаёт 18 В и 0,5 А. Какая у неё мощность? Сколько энергии она вырабатывает за 6 часов такой работы? Ответы в Вт и Вт·ч.

5.10. Ночью метеостанция берёт от аккумулятора 180 мА. На аккумуляторе написано 2,4 А·ч. Сколько часов он продержится? Округли до десятых. Хватит ли его на зимнюю ночь длиной 16 часов?

Задача 5.6 Блок питания для насоса

выбор блока питания, мощность, энергия, кВт·ч

Для полива сити-фермы на крыше купили насос. На корпусе написано: «12 В, 0,8 А». В магазине есть четыре блока питания:

Таблица 5.2

Блок	1	2	3	4
Напряжение	5 В	12 В	12 В	24 В
Наибольший ток	2 А	0,5 А	1,5 А	1 А

а) Какой блок подходит? В ответ – его номер. б) Насос включается 4 раза в сутки по 15 минут. Сколько энергии он потратит за 30 дней? Ответ в кВт·ч.

Дано: насос 12 В, 0,8 А; 4 включения по 15 мин в сутки; 30 дней.

Найти: номер блока; E – ?

Решение

Шаг 1. Напряжение должно совпадать. Насосу нужно 12 В. Блок 1 даст 5 В – насос не закрутится. Блок 4 даст 24 В – вдвое больше, мотор перегреется и сгорит. Остаются блоки 2 и 3.

Шаг 2. Ток – не меньше нужного. На блоке пишут, какой **наибольший** ток он может отдать. Насос возьмёт 0,8 А. Блок 2 даёт не больше 0,5 А – он перегреется или отключится. Блок 3 выдерживает 1,5 А, это больше 0,8 А. Лишний запас по току не страшен: насос возьмёт столько, сколько ему нужно, не больше.

Шаг 3. Мощность насоса. $P = 12 \cdot 0,8 = 9,6$ Вт.

Шаг 4. Время работы. За сутки $4 \cdot 15 = 60$ мин = 1 ч. За 30 дней – 30 ч.

Шаг 5. Энергия. $E = 9,6 \cdot 30 = 288$ Вт·ч. В киловатт-часах: $288/1000 = 0,288$ кВт·ч.

Много это или мало? Чайник на 2 кВт тратит столько же примерно за 9 минут. Автоматический полив на крыше стоит копейки.

⚠ Типичная ошибка. Выбрать блок 4, «потому что он мощнее», или блок 2, «потому что у него 12 В». Подходит только блок, у которого совпадает напряжение **и** хватает тока. В расчёте энергии легко забыть перевести минуты в часы: $9,6 \cdot 60 \cdot 30 = 17\,280$ – это не ватт-часы, а «ватт-минуты».

Ответ: а) 3; б) 0,288 кВт·ч.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «Насос для зелёного оазиса»

Реши сам

5.11. Вентилятор теплицы на 12 В берёт ток 0,25 А и в июле работает по 6 часов в сутки. Сколько киловатт-часов он потратит за июль?

5.12. От одного блока питания решили питать сразу насос (12 В, 0,8 А) и вентилятор (12 В, 0,25 А), включённые параллельно. Блоки: 1) 12 В, 1 А; 2) 5 В, 3 А; 3) 12 В, 1,5 А; 4) 12 В, 2 А; 5) 9 В, 2 А. Какие подходят? Ответ – номера по возрастанию без пробелов.

Задача 5.7* *. Зимняя ночь в теплице

теплопотери, мощность обогрева, энергия, теплоизоляция

Маленькая теплица из одного слоя плёнки теряет 120 Вт на каждый градус разницы температур внутри и снаружи. Ночью на улице $-14\text{ }^{\circ}\text{C}$, а внутри нужно держать $+16\text{ }^{\circ}\text{C}$. Ночь длится 14 часов. Греют теплицу электрическими обогревателями по 1,5 кВт. а) Сколько обогревателей нужно, чтобы удержать $+16\text{ }^{\circ}\text{C}$? б) Теплицу обтянули вторым слоем плёнки, и потери тепла уменьшились на 40 %. Сколько киловатт-часов за ночь это сэкономит? Округли до десятых.

Дано: $k_1 = 120\text{ Вт/}^{\circ}\text{C}$; $t_{\text{вн}} = 16\text{ }^{\circ}\text{C}$; $t_{\text{нар}} = -14\text{ }^{\circ}\text{C}$; 14 ч; обогреватель 1,5 кВт; второй слой: потери меньше на 40 %.

Найти: число обогревателей; экономия за ночь – ?

Решение

Шаг 1. Разница температур. $16 - (-14) = 16 + 14 = 30\text{ }^{\circ}\text{C}$.

Шаг 2. Потери с одним слоем. $P_1 = 120 \cdot 30 = 3600\text{ Вт} = 3,6\text{ кВт}$. Столько тепла каждую секунду уходит на улицу, и столько же должны давать обогреватели.

Шаг 3. Сколько обогревателей. $3,6/1,5 = 2,4$. Два дадут только 3 кВт – мало, теплица остынет. Округляем вверх: 3 обогревателя (термостат будет их иногда выключать).

Шаг 4. Энергия за ночь с одним слоем. $E_1 = 3,6 \cdot 14 = 50,4\text{ кВт}\cdot\text{ч}$.

Шаг 5. Потери с двумя слоями. Минус 40 % – значит, остаётся 60 %: $k_2 = 120 \cdot 0,6 = 72\text{ Вт/}^{\circ}\text{C}$, $P_2 = 72 \cdot 30 = 2160\text{ Вт} = 2,16\text{ кВт}$. Хватит и двух обогревателей.

Шаг 6. Экономия. $E_2 = 2,16 \cdot 14 = 30,24\text{ кВт}\cdot\text{ч}$. Разница $50,4 - 30,24 = 20,16 \approx 20,2\text{ кВт}\cdot\text{ч}$. Быстрее: сэкономили 40 % от 50,4 кВт·ч, а это $50,4 \cdot 0,4 = 20,16$ – то же самое.

Для сравнения: 20 кВт·ч – это почти столько, сколько квартира тратит за три-четыре дня. За одну ночь. Второй слой плёнки окупается очень быстро.

⚠ Типичная ошибка. Посчитать разницу как $16 - 14 = 2\text{ }^{\circ}\text{C}$ и получить потери 240 Вт – «хватит и настольной лампы». Мороз вычитаем со знаком: минус на минус даёт плюс. Вторая ошибка – округлить 2,4 обогревателя до 2 «по правилам округления». Тут нужно, чтобы тепла хватило, поэтому только вверх. Третья – умножить потери на 0,4 вместо 0,6 и найти, сколько тепла уходит после утепления, а не сколько сэкономили.

Ответ: а) 3 обогревателя; б) 20,2 кВт·ч.

Своя задача по программе сферы, модуль 2

Реши сам

5.13. Теплица теряет 80 Вт на каждый градус разницы температур. Внутри нужно $+20\text{ }^{\circ}\text{C}$, на улице $-5\text{ }^{\circ}\text{C}$. Какой мощности нужен обогрев? Сколько киловатт-часов уйдёт за ночь длиной 10 часов?

5.14*.* В теплице стоит один обогреватель на 2 кВт, других нет. Теплица теряет 100 Вт на градус разницы температур. На улице -25°C – морозная январская ночь под Новосибирском. Какую самую высокую температуру обогреватель удержит внутри? А если утеплить теплицу так, что потери уменьшатся на 30 %? Округли до десятых.

Программирование умной теплицы



Умная теплица сама поливает грядку и открывает форточку, но думает за неё человек: он пишет программу, а плата Arduino её выполняет. В этой главе мы научимся читать такие программы и писать свои – от записки «как поливать цветы» до машины состояний, которая командует насосом. Задача с кодом выглядит так: дан короткий скетч и несколько показаний датчика, а спрашивают, что сделает устройство, где в программе ошибка или в каком порядке стоят перепутанные строки. В ответ вводят число, строку из цифр или из букв.

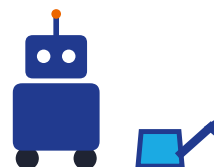
6.1. Алгоритм и исполнитель

НА ПАЛЬЦАХ

Записка соседу

1. Потрогай землю в горшке.
2. Если сухая – вылей одну кружку.
3. Завтра утром – снова с шага 1.

выполняет



алгоритм

исполнитель

Рисунок 6.1 – Записка соседу – это алгоритм, а тот, кто её выполняет, – исполнитель

Сосед сделает всё в точности так, как написано в записке. Забудешь написать «закрой кран» – вода так и будет литься.

Летом ты уезжаешь на две недели в лагерь и просишь соседа поливать твой кактус и помидоры на подоконнике. Если сказать «ну, поливай, когда надо», сосед зальёт кактус или засушит помидоры. Надёжнее оставить записку, где всё расписано по шагам: что проверить, что сделать, когда повторить. Такая записка и есть **алгоритм** – список точных шагов, который приводит к нужному результату.

Тот, кто выполняет алгоритм, называется **исполнителем**. Исполнитель знает только свой набор команд – **систему команд**. Сосед понимает «потрогай землю» и «вылей кружку». Плата Arduino понимает «измерь напряжение на входе А0» и «подай 5 вольт на вывод 7». Разница в том, что сосед может догадаться, чего ты хотел, а плата – никогда. Она выполняет каждое слово буквально и не задаёт вопросов. Поэтому программы для неё пишут без «ну, как-нибудь».

Любой алгоритм собирают из трёх видов «кирпичиков».

- **Следование** – шаги идут один за другим: измерить, потом сравнить, потом включить.
- **Ветвление** – развилка «если... то... иначе...»: если земля сухая – полить, иначе ничего не делать.
- **Цикл** – повторение: «повтори 5 раз» или «повторяй, пока бак не наполнится».

Алгоритм удобно рисовать **блок-схемой**. Прямоугольник на ней – действие, ромб – вопрос с ответами «да» и «нет», стрелки показывают, куда идти дальше. Так выглядит алгоритм автополива, который мы потом превратим в настоящую программу:

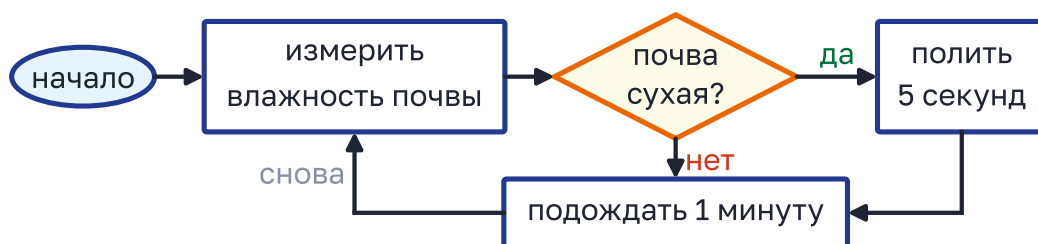


Рисунок 6.2 – Блок-схема автополива. Ромб – вопрос: от ответа зависит, по какой стрелке идти. Стрелка «снова» замыкает круг – это цикл

В задачах на алгоритмы часто придумывают простого исполнителя. Пусть у нас будет **Робот-поливальщик**. Он ездит вдоль ряда горшков и знает три команды:

- «вперёд» – проехать к следующему горшку;
- «полить» – налить в горшок, у которого стоит, одну порцию воды (сколько миллилитров – скажут в условии);
- «если земля сухая – ...» – проверить землю в горшке, у которого стоит, и выполнить команду, только если земля сухая.

А ещё он умеет повторять: «повтори 3 раза: полить, вперёд». Команды внутри повторения выполняются по порядку, потом всё сначала, и так три раза. Прежде чем отвечать, что сделает робот, пройди программу пальцем, команду за командой, и записывай, у какого горшка он стоит. Угадывать тут бесполезно.

Запомни. Алгоритм – точные шаги, исполнитель – тот, кто их выполняет без догадок. Из трёх кирпичиков (следование, ветвление, цикл) собирается любая программа. Что делает исполнитель, узнают только одним способом: проходят алгоритм шаг за шагом.

6.2. Переменные, условия, циклы

НА ПАЛЬЦАХ

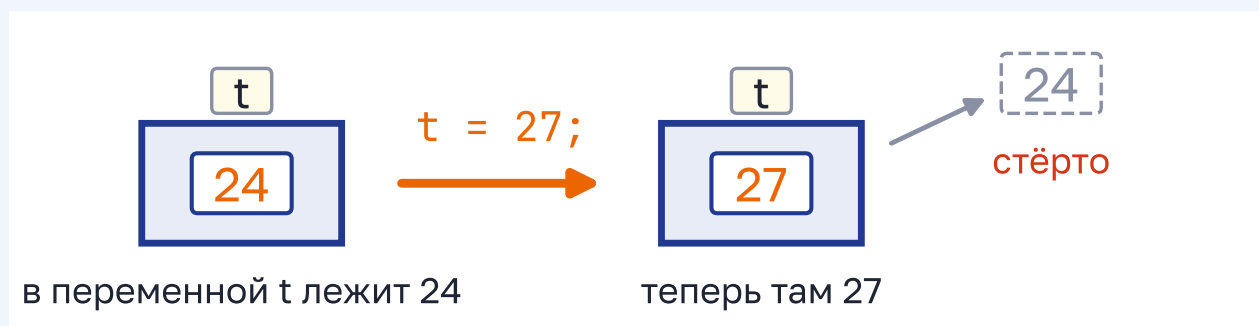


Рисунок 6.3 – Переменная – коробка с наклейкой. Положили новое число – старое пропало

Имя у коробки одно, а содержимое можно менять сколько угодно раз.

Дальше – язык C++, на котором пишут программы для Arduino. Если ты никогда не программировал, читай код медленно, по строке: рядом с каждой стоит пояснение. После двух косых черт `//` идёт **комментарий** – это пометка для людей, плата его пропускает.

Переменные

Переменная – ячейка памяти с именем. В неё кладут число и потом берут его оттуда. Перед первым использованием переменную **объявляют**: пишут её тип и имя.

```
int count = 0;           // сколько раз поливали – целое число
float t = 23.5;          // температура – дробное число, пишется через точку
bool pumpOn = false;    // включён ли насос: да (true) или нет (false)

count = count + 1;       // было 0, стало 1
t = 27;                  // старое значение 23.5 стёрто, теперь там 27
```

Тип говорит, что лежит в коробке: `int` – целое число, `float` – дробное, `bool` – «да» или «нет». Каждая команда заканчивается точкой с запятой.

Знак `=` в программе означает «положи в коробку». Это команда, а вовсе не равенство из математики. Запись `count = count + 1` на уроке математики была бы бессмыслицей, а здесь она понятна: возьми, что лежит в `count`, прибавь 1 и положи обратно. Так программы считают: сколько раз поливали, сколько минут работал насос.

Одна ловушка про деление. Если делятся два целых числа, дробная часть **отбрасывается**:

```
int a = 7 / 2;           // 3 - дробную часть выбросили
float b = 7 / 2;         // тоже 3! делили целые числа, а потом положили в дробную коробку
float c = 7.0 / 2;       // 3.5 - одно из чисел дробное, и деление честное
```

Коварнее всех строка с `b`: коробка дробная, но в неё кладут уже готовую тройку, и половинке взяться неоткуда. Если в формуле нужна дробь, пиши хотя бы одно число с точкой: `5.0`, а не `5`.

Условия

Условие – это ветвление из блок-схемы, записанное на C++:

```
if (t ≥ 28) {
    digitalWrite(FAN, HIGH); // жарко - включить вентилятор
} else {
    digitalWrite(FAN, LOW);   // иначе - выключить
}
```

`if` значит «если», `else` – «иначе». В круглых скобках – вопрос, на который бывает только два ответа: «верно» или «неверно». Если верно, выполняются команды в первых фигурных скобках, если нет – во вторых. Ветки `else` может и не быть: тогда при неверном условии программа просто идёт дальше.

Сравнивают такими знаками: `<` меньше, `>` больше, `≤` меньше или равно, `≥` больше или равно, `=` равно, `≠` не равно. Обрати внимание на `=`: сравнение пишется двумя знаками «равно». Один знак `=` – это «положи в коробку», и если перепутать, программа сломается молча (что тогда случится – в «Реши сам» 6.7).

Когда вариантов больше двух, пишут цепочку `if ... else if ... else`. Проверки идут **сверху вниз**, и срабатывает только первая верная ветка – остальные даже не проверяются. Поэтому порядок строк в цепочке важен: во второй ветке уже известно, что первое условие оказалось неверным.

Два условия склеивают связками: `&&` – «и» (нужно, чтобы были верны оба), `||` – «или» (хватит одного). Например, `t ≥ 28 && door == LOW` – жарко **и** дверь закрыта.

Циклы

НА ПАЛЬЦАХ

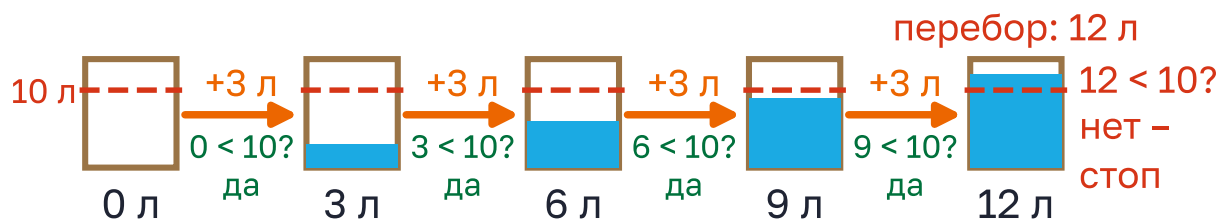


Рисунок 6.4 – Цикл `while` : пока в бочке меньше 10 литров – ещё ведро. Вопрос задают перед ведром, а не после

При 9 литрах ответ «да», и ведро выливают целиком: в бочке 12 литров вместо десяти.

Цикл повторяет команды. Цикл `for` нужен, когда известно, сколько раз повторить:

```
for (int i = 1; i ≤ 3; i++) {    // i = 1, 2, 3 - три круга
    digitalWrite(LED, HIGH);
    delay(200);
    digitalWrite(LED, LOW);
    delay(200);
}
```

В скобках три части через точку с запятой: с чего начать (`i = 1`), до каких пор крутиться (`i ≤ 3`) и что делать после каждого круга (`i++` – прибавить к `i` единицу). Светодиод мигнёт три раза, на это уйдёт $3 \cdot (200 + 200) = 1200$ мс.

Цикл `while` («пока») крутится, пока условие верно. Сколько будет кругов, заранее может быть неизвестно:

```
int litres = 0;
while (litres < 10) {    // пока в бочке меньше 10 литров
    litres = litres + 3;  // принести ещё одно ведро на 3 литра
}
```

Чтобы узнать, чем это кончится, заведём **таблицу трассировки**: столбец на каждую переменную, строка на каждый шаг.

Таблица 6.1

Круг	litres до круга	litres < 10 ?	litres после круга
1	0	да	3
2	3	да	6
3	6	да	9
4	9	да	12

Круг	litres до круга	litres < 10 ?	litres после круга
выход 12		нет	12

Вёдер понадобилось четыре, а в бочке 12 литров – больше десяти. Будь бочка на 10 литров, два литра оказались бы на земле. В задачах про насосы и баки эта «лишняя порция» встречается постоянно, так что считай круги по таблице.

Запомни. = кладёт значение в переменную, == сравнивает. Целое, делённое на целое, теряет дробную часть. В цепочке if ... else if срабатывает только первая верная ветка. Что делает цикл, проверяй таблицей трассировки, а не в уме.

Совет. Таблицу трассировки рисуй на черновике даже для коротких программ. На отборе на задачу даётся несколько попыток ввода, и каждая неверная снимает часть балла. Две минуты на табличку обходятся дешевле.

6.3. Скетч: датчик, порог, действие

НА ПАЛЬЦАХ



Рисунок 6.5 – setup() выполняется один раз, а loop() крутится по кругу, пока плата включена

Так работает сторож в теплице: обошёл, посмотрел на градусник, решил, открыть ли дверь, – и пошёл на следующий обход.

Программа для Arduino называется **скетч**. В ней всегда две обязательные части.

- `setup()` – «настройка». Выполняется один раз, сразу после включения питания. Здесь говорят плате, какой вывод будет выходом, и включают связь с компьютером.
- `loop()` – «петля». Выполняется снова и снова, без конца. Здесь живёт вся работа: измерить, сравнить, включить или выключить.

Вот основные команды. Их немного, и все они встретятся в задачах главы.

Таблица 6.2

Команда	Что делает
<code>pinMode(7, OUTPUT);</code>	сделать вывод 7 выходом

Команда	Что делает
<code>analogRead(A0)</code>	измерить напряжение на входе A0, получить число 0...1023
<code>digitalWrite(7, HIGH);</code>	подать на вывод 7 напряжение 5 В; LOW – снять
<code>analogWrite(9, 100);</code>	включить вывод 9 «вполсилы»: 0 – выключен, 255 – полная сила
<code>delay(5000);</code>	ждать 5000 мс, то есть 5 секунд
<code>millis()</code>	сколько миллисекунд прошло с включения платы
<code>Serial.println(x);</code>	показать значение x в мониторе порта

Откуда берётся число от 0 до 1023 и как перевести его в градусы, рассказано в главе 5. Здесь нам хватит одного: датчик выдаёт число, и с ним можно сравнивать.

Соберём настоящую автоматику – полив по датчику влажности почвы. Датчик подключён ко входу A0, насос – к выводу 7. Насосу нужен ток побольше, чем может дать вывод платы, поэтому между ними стоит **реле** – выключатель, который щёлкает от слабого сигнала Arduino и включает мощную цепь насоса. Для программы это неважно: HIGH – насос качает, LOW – стоит.



Рисунок 6.6 – Что куда подключено: датчик даёт плате число, плата командует реле, реле включает насос

У датчика в нашей теплице чем суше земля, тем меньше число. Бывают датчики, устроенные наоборот, поэтому в каждой задаче смотри, что сказано в условии. Землю считаем сухой, если число меньше 400.

```

const int SOIL = A0;    // датчик влажности почвы – на входе A0
const int PUMP = 7;     // насос через реле – на выводе 7

void setup() {
  pinMode(PUMP, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int soil = analogRead(SOIL);    // число от 0 до 1023
  Serial.println(soil);
  if (soil < 400) {               // сухо?
    digitalWrite(PUMP, HIGH);    // включить насос
    delay(5000);                 // поливать 5 секунд
    digitalWrite(PUMP, LOW);     // выключить насос
  }
  delay(60000);                  // следующая проверка через минуту
}
  
```

Разберём по строкам. `const int S0IL = A0;` – **константа**: коробка, содержимое которой менять нельзя. Имя удобнее номера: читать `analogRead(S0IL)` проще, чем `analogRead(A0)`, а если перепаяешь датчик на другой вход, исправишь одну строку. В `setup()` делаем вывод насоса выходом и включаем монитор порта. В `loop()` измеряем почву, выводим число на экран, и если оно меньше 400 – включаем насос, ждём 5 с и выключаем. Потом минуту ждём, и круг начинается снова.

Посмотри на блок-схему из раздела 6.1 – это она, только на C++. Подставь по очереди показания 452, 418, 400, 391: насос включится один раз, на четвёртой проверке, и проработает 5 с. На 400 он промолчит: знак в условии строгий, а 400 не меньше 400.

Полив на время. В главе 4 мы считали, сколько держать насос включённым: 3 л насосом на 2 л/мин – это 1,5 мин, или 90 с. В программе время пишут в миллисекундах, поэтому в `delay` пойдёт $90 \cdot 1000 = 90\,000$. Цепочка всегда одна: литры делим на производительность, получаем минуты или секунды, переводим в миллисекунды.

Пока идёт `delay`, плата спит. Она не смотрит на датчики, не видит кнопок и не может выключить насос раньше. В нашем скетче это почти не мешает: насос работает 5 с, а потом плата честно ждёт минуту. Но учти: круг `loop()` длится минуту плюс время полива, и в круге с поливом выходит 65 с. В задаче 6.3 это пригодится. А как обойтись без долгого сна, покажем в разделе 6.5.

Запомни. `setup()` – один раз, `loop()` – по кругу. Схема любой автоматики: измерить (`analogRead`), сравнить с порогом (`if`), включить или выключить (`digitalWrite`). Время в `delay` – в миллисекундах: секунды умножь на 1000.

Совет. Любую программу из этой главы можно собрать в онлайн-симуляторе: в Tinkercad (вход по ссылке на класс из курса) или в Wokwi (wokwi.com). Датчик влажности или термометр там крутят ползунком прямо во время работы, а в мониторе порта видно каждое число. Запущенная программа учит больше, чем три прочитанные.

6.4. Два порога в программе

НА ПАЛЬЦАХ

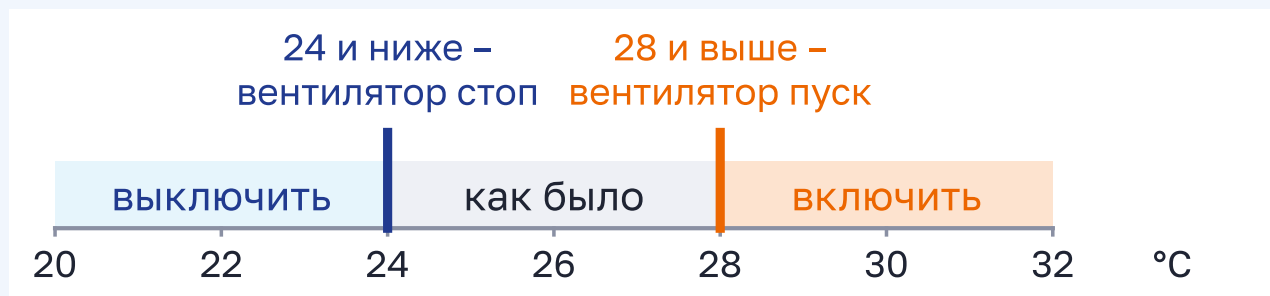


Рисунок 6.7 – Вентилятор с двумя порогами. Между 24 и 28 градусами программа ничего не трогает

Как в классе с окном: открывают, когда стало душно, а закрывают, когда уже прохладно, – а не каждый раз, когда кто-то чихнул.

В разделе 2.6 мы уже выяснили, зачем нужны **два порога** (гистерезис): с одним порогом форточка или вентилятор около границы «хлопают» каждую минуту. Теперь посмотрим, как это записать в программе.

Первая попытка – условие с `else`:

```
if (t ≥ 28) {  
    digitalWrite(FAN, HIGH);  
} else {  
    digitalWrite(FAN, LOW);  
}
```

Это один порог. Температура 28,1 – вентилятор работает, 27,9 – стоит, 28,1 – снова работает. Датчик всегда немного дрожит, и реле будет щёлкать без остановки.

С двумя порогами программе нужна **память**. При 26 градусах вентилятор может работать (если недавно было жарко и ещё не остыло), а может стоять (если жарко пока не было). Одним показанием датчика этого не решить, нужно помнить, что было раньше. Для памяти заводим переменную `bool`:

```

const int FAN = 9;           // вентилятор через реле - на выводе 9
bool fanOn = false;         // память: включён ли вентилятор сейчас

void setup() {
  pinMode(FAN, OUTPUT);
}

void loop() {
  float t = readT();         // температура, °C
  if (t ≥ 28) {               // жарко - включаем
    fanOn = true;
  }
  if (t ≤ 24) {               // остыло - выключаем
    fanOn = false;
  }
  // от 24 до 28 ни одно условие не сработало - fanOn остаётся прежним
  if (fanOn) {
    digitalWrite(FAN, HIGH);
  } else {
    digitalWrite(FAN, LOW);
  }
  delay(1000);
}

```

Здесь `readT()` – функция, которая читает термометр и возвращает градусы (как её написать, рассказано в главе 5 и в справочнике). Посмотри на два отдельных `if` без `else`. Если температура между порогами, не срабатывает ни один, и `fanOn` хранит то, что в нём было. В этом и весь фокус двух порогов.

Прогоним программу по показаниям 25; 27; 28,2; 27,5; 26; 24,5; 24; 25; 27,9. Вентилятор включился на 28,2, работал при 27,5, 26 и 24,5, выключился на 24 и больше не включался – даже при 27,9, потому что до 28 не дотянуло.

На чём спотыкаются? Пишут второй `if` через `else`: `if (t ≥ 28) fanOn = true; else fanOn = false;`. Выглядит почти так же, но это снова один порог – память затирается на каждом круге. А вот `else if (t ≤ 24)` вместо второго `if` – можно: оба условия сразу верными быть не могут, и между порогами опять не сработает ни одно.

Запомни. Два порога = переменная-память + два условия: «выше верхнего – включить», «ниже нижнего – выключить». Если ни одно не сработало, память не трогаем. Голый `else` превращает два порога обратно в один.

6.5. Машина состояний теплицы

НА ПАЛЬЦАХ

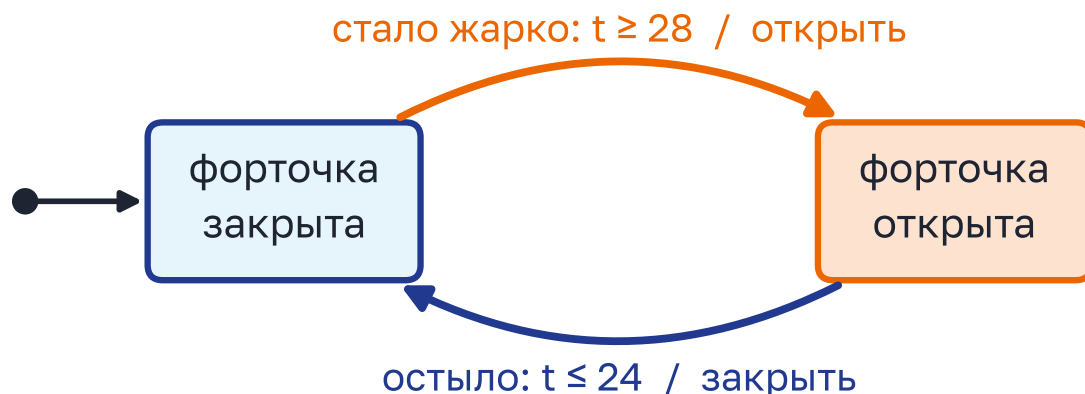


Рисунок 6.8 – Форточка – машина с двумя состояниями. Стрелки – переходы, на стрелках – когда переходить и что сделать

Форточка в каждый момент либо закрыта, либо открыта. Переключает её температура.

Посмотри на программу с двумя порогами из прошлого раздела иначе. У вентилятора два **состояния** – «стоит» и «работает». Из одного в другое он переходит, когда случается что-то определённое. Так можно описать почти любое устройство, и у инженеров для этого есть готовый способ – **машина состояний** (её ещё называют конечным автоматом). Она состоит из пяти деталей.

- **Состояние** – режим работы устройства: «ждём», «поливаем», «форточка открыта». В каждый момент устройство находится только в одном состоянии.
- **Событие** – то, что случилось и требует реакции: пришло новое показание, прошло 10 секунд, нажали кнопку.
- **Условие** – проверка, без которой переход не сработает: «почва суше порога», «температура не ниже 28».
- **Действие** – что устройство делает: включить насос, повернуть форточку.
- **Переход** – смена состояния, на схеме это стрелка.

Подпись на стрелке читают так: **событие [условие] / действие**. «Новое показание [$t \geq 28$] / открыть форточку» значит: пришло показание, температура не ниже 28 – открываем форточку и переходим в состояние «открыта». Если условие неверно, стрелка не срабатывает, и устройство остаётся, где было.

Теперь полезная машина. Воду в сухую землю лучше лить порциями: налил – подождал, пока впитается, – проверил снова. Если лить без перерыва, вода успевает стечь по краям горшка мимо корней, а датчик, воткнутый в одном месте, узнает об этом поздно. Получается три состояния:



Рисунок 6.9 – Машина состояний «полив порциями». Чёрная точка показывает, с какого состояния машина начинает

Программа по этой схеме хранит номер текущего состояния в переменной `state` и время входа в него – в переменной `start`. Функция `loop()` крутится очень быстро и на каждом круге спрашивает одно: «не пора ли переходить?»

```

const int SOIL = A0;           // датчик влажности почвы
const int PUMP = 7;            // насос через реле

int state = 0;                 // 0 - ждём, 1 - поливаем, 2 - вода впитывается
unsigned long start = 0;       // когда вошли в текущее состояние, мс

void go(int newState) {       // переход в новое состояние
  state = newState;
  start = millis();
  if (state == 1) {
    digitalWrite(PUMP, HIGH);  // входим в «поливаем» - включить насос
  } else {
    digitalWrite(PUMP, LOW);   // в остальных состояниях насос стоит
  }
}

void setup() {
  pinMode(PUMP, OUTPUT);
  go(0);                       // начинаем с «ждём»
}

void loop() {
  unsigned long t = millis() - start; // сколько мс длится текущее состояние
  int soil = analogRead(SOIL);
  if (state == 0 && soil < 400) {
    go(1);
  } else if (state == 1 && t ≥ 10000) {
    go(2);
  } else if (state == 2 && t ≥ 60000) {
    go(0);
  }
  delay(10);
}
  
```

Что здесь нового. `void go(int newState)` – **функция**, своя команда из нескольких строк: мы один раз объяснили плате, что значит «перейти в состояние», и дальше пишем просто `go(1)`. Включение насоса внутри `go` – это действие на входе в состояние: оно выполняется один раз, когда в состояние попали. `unsigned long` – тип для очень больших целых чисел: на Arduino Uno в обычный `int` помещается число не больше 32 767. Для `millis()` это всего 33

секунды работы платы, дальше счёт сломается. В `unsigned long` влезает больше четырёх миллиардов – это почти 50 суток. А разность `millis() - start` – это «сколько я уже в этом состоянии». Каждая строка `loop()` читается прямо по схеме: «если ждём и почва суше 400 – переходим в состояние 1».

Главное отличие от скетча из раздела 6.3 – здесь нет долгого `delay`. Плата не спит 60 секунд, пока вода впитывается, а сто раз в секунду проверяет: не пора ли? Значит, в тот же `loop()` можно дописать ещё одну машину – например, форточку, – и обе будут работать одновременно. Со скетчем на `delay(60000)` так не получится: форточка ждала бы, пока досыпает полив. В совсем сухой земле машина работает так: 10 с полива, минута тишины, снова 10 с – и так, пока датчик не покажет 400 или больше. Пусть после порции земля ещё сухая – насос всё равно минуту подождёт, пока вода впитается.

ПРИМС и Кибериада IDE. В курсе сферы машины состояний рисуют на визуальном языке **ПРИМС** (программирование расширенных иерархических машин состояний) в редакторе **Кибериада IDE**. Программу там рисуют: прямоугольники-состояния и стрелки-переходы с подписями, как на наших схемах. По такому рисунку редактор умеет сам сделать код на C++. Логика та же, что в нашей программе: состояния, события, условия, действия. Поэтому, если ты понял схему на рисунке выше, в Кибериаде тебе останется освоить только, какой кнопкой что рисуется – а это покажут на курсе.

Проверь каждую схему на две беды. **Тупик** – состояние, из которого не выходит ни одна стрелка: попав туда, устройство «зависнет» навсегда. Исключение – **конечное состояние**: в курсе его рисуют как кружок в кружке, и оно нарочно завершает работу машины. Тупик, который не отмечен как конечное состояние, – ошибка. **Недостижимое состояние** – в него не ведёт ни одна стрелка, и оно никогда не включится. И ещё: машина делает **один переход за раз**. Если форточка закрыта, а температура сразу подскочила до 35, она не прыгнет через все ступеньки: сначала сработает первая подходящая стрелка, а следующая – на следующем замере.

Как проследить работу машины на бумаге: таблица «замер – состояние до – какое условие верно – состояние после – действие». Заполняй строку за строкой, как трассировку в разделе 6.2.

Запомни. Машина состояний = состояния + переходы «событие [условие] / действие». Номер состояния хранят в одной переменной, переходы проверяет цепочка `if ... else if`. Вместо долгого `delay` – вопрос «сколько я уже в этом состоянии?»: `millis() - start`. За один замер – один переход. Из каждого состояния, кроме конечного, должна выходить хотя бы одна стрелка. Две стрелки с одним событием из одного состояния допустимы, только если условия у них не пересекаются.

6.6. Разбираем задачи

Задача 6.1 Робот-поливальщик

разминка · алгоритм, исполнитель, цикл «повтори N раз»

Вдоль стеллажа стоят в ряд 14 горшков с рассадой, они пронумерованы от 1 до 14. Робот-поливальщик стоит у горшка 1. Команда «вперёд» перевозит его к следующему горшку, команда «полить» наливает 100 мл в горшок, у которого он стоит. Робот выполняет программу:

повтори 4 раза:
 полить
 вперёд
 вперёд
 вперёд

а) Какие горшки будут политы? Ответ – номера по возрастанию. б) Сколько миллилитров воды уйдёт? в) У какого горшка робот окажется в конце?

Дано: 14 горшков, старт у горшка 1; в повторении: 1 раз «полить» (100 мл) и 3 раза «вперёд»; 4 повторения.

Найти: номера политых горшков; расход воды; где остановится робот.

Решение

Шаг 1. Что делает один круг. Робот поливает горшок, у которого стоит, и уезжает на три горшка вперёд. Значит, каждый следующий полив – через три номера от предыдущего.

Шаг 2. Проходим все четыре круга. Записываем, где стоит робот перед каждым «полить»:

Таблица 6.3

Круг	Робот стоит у горшка	Полил	После трёх «вперёд»
1	1	горшок 1	у горшка 4
2	4	горшок 4	у горшка 7
3	7	горшок 7	у горшка 10
4	10	горшок 10	у горшка 13

Шаг 3. Вода. Полито 4 горшка по 100 мл: $4 \cdot 100 = 400$ мл.

Шаг 4. Где робот. После четвёртого круга команд больше нет. Три «вперёд» робот уже сделал, он стоит у горшка 13 и поливать его не будет.

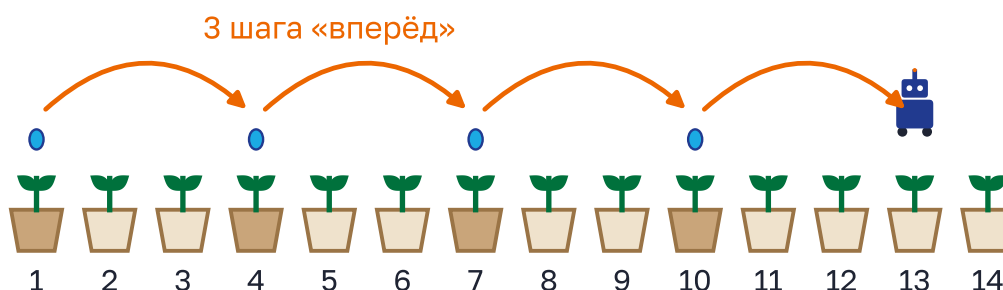


Рисунок 6.10 – Путь робота: полив, три шага вперёд, снова полив. Капли – политые горшки

Проверим по-другому: всего команд «вперёд» $4 \cdot 3 = 12$, и робот начал у горшка 1, значит, закончит у горшка $1 + 12 = 13$. До края стеллажа ещё один горшок в запасе – робот не свалится.

❗ **Типичная ошибка.** Посчитать 12 шагов «вперёд» и ответить «у горшка 12», забыв, что робот стартовал с первого, а не с нулевого горшка. Вторая ошибка – решить, что последний горшок, у которого остановился робот, тоже полит. Полив в круге стоит **до** шагов вперёд, а пятого круга нет.

Ответ: а) 1, 4, 7, 10; б) 400 мл; в) у горшка 13.

Своя задача по программе сферы, модуль 1

Реши сам

6.1. Робот-поливальщик из задачи 6.1 стоит у горшка 1, «полить» наливает 150 мл. Программа: «повтори 5 раз: полить, полить, вперёд, вперёд». Сколько литров воды уйдёт и у какого горшка робот окажется в конце?

6.2. В ряду 10 горшков, сухая земля в горшках 1, 3, 4, 6 и 9. Робот стоит у горшка 1, «полить» наливает 200 мл. Программа А: «повтори 8 раз: если земля сухая – полить; вперёд». Программа Б: «повтори 8 раз: вперёд; если земля сухая – полить». Какие горшки польёт каждая программа и сколько воды уйдёт?

Задача 6.2 Светофор для рассады

разминка · if – else if – else, граница условия, что делает программа

На стеллаже с рассадой висит индикатор из трёх светодиодов: синий – холодно, зелёный – норма, красный – жарко. Функция `readT()` уже написана и возвращает температуру в градусах. Программа:

```

const int BLUE_LED = 2;    // синий: холодно
const int GREEN_LED = 3;  // зелёный: норма
const int RED_LED = 4;    // красный: жарко

void setup() {
  pinMode(BLUE_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
  pinMode(RED_LED, OUTPUT);
}

void loop() {
  float t = readT();        // температура в градусах
  digitalWrite(BLUE_LED, LOW); // сначала гасим все три
  digitalWrite(GREEN_LED, LOW);
  digitalWrite(RED_LED, LOW);
  if (t < 18) {
    digitalWrite(BLUE_LED, HIGH);
  } else if (t ≤ 26) {
    digitalWrite(GREEN_LED, HIGH);
  } else {
    digitalWrite(RED_LED, HIGH);
  }
  delay(1000);
}

```

Термометр по очереди показал 17,5; 18; 26; 26,4 и 31 °С. Какой светодиод горел после каждого замера? Ответ – строка из пяти цифр: 1 – синий, 2 – зелёный, 3 – красный.

Дано: синий при $t < 18$; иначе зелёный при $t \leq 26$; иначе красный. Замеры: 17,5; 18; 26; 26,4; 31.

Найти: цвет после каждого замера.

Решение

Шаг 1. Переводим цепочку на русский. Проверки идут сверху вниз. Меньше 18 – синий. Если не меньше 18, но не больше 26 – зелёный. Во вторую ветку попадают только числа от 18 и выше, поэтому зелёный горит при температуре от 18 до 26 включительно. Всё остальное – больше 26 – красный.

Шаг 2. Каждый замер по очереди.

- 17,5: $17,5 < 18$ – верно, синий, **1**.
- 18: $18 < 18$ – неверно (18 не меньше самого себя). Дальше $18 \leq 26$ – верно, зелёный, **2**.
- 26: $26 < 18$ – неверно; $26 \leq 26$ – верно (знак «меньше или равно»), зелёный, **2**.
- 26,4: $26,4 \leq 26$ – неверно, остаётся ветка `else`, красный, **3**.
- 31: красный, **3**.

Шаг 3. Почему горит только один. В начале каждого круга программа гасит все три светодиода, а потом зажигает один нужный. Без этих трёх строк при переходе с синего на зелёный горели бы оба.

Два замера из пяти стоят точно на границах, и оба попали в «норму»: так задумано знаками `<` и `≤`.

❗ **Типичная ошибка.** Ответить «3» для 26 °C, прочитав $\leq$ как $<$. Или для 18 °C ответить «1», решив, что 18 уже холодно. Граничные числа проверяй отдельно, выписывая сравнение целиком: «18 < 18 – неверно».

Ответ: 12233.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «Индикаторы в виде светодиодов»

Реши сам

6.3. Лампа досветки управляется так (light – показание фоторезистора, включённого как в курсе: чем темнее, тем больше число):

```
if (light > 800) {  
    analogWrite(LAMP, 255);    // полная яркость  
} else if (light > 500) {  
    analogWrite(LAMP, 100);    // вполсилы  
} else {  
    analogWrite(LAMP, 0);      // выключена  
}
```

Какое число получит analogWrite при показаниях 900, 800, 501, 500 и 270? Ответ – пять чисел через запятую.

6.4. Программист переписал условие из задачи 6.2 и поменял ветки местами:

```
if (t < 26) {  
    digitalWrite(GREEN_LED, HIGH);  
} else if (t < 18) {  
    digitalWrite(BLUE_LED, HIGH);  
} else {  
    digitalWrite(RED_LED, HIGH);  
}
```

Какие светодиоды будут гореть при 15, 22 и 30 °C (строка из трёх цифр: 1 – синий, 2 – зелёный, 3 – красный)? Какой светодиод в этой программе не загорится никогда?

Задача 6.3 Сколько выпьет грядка

скетч, analogRead, порог, delay, время круга loop

На сити-ферме на крыше работает скетч полива из раздела 6.3: раз в минуту он проверяет почву и, если число меньше 400, включает насос на 5 с. Насос качает 30 мл/с. Первая проверка – сразу после включения платы. Датчик на первых десяти проверках показал:

452, 418, 400, 391, 447, 425, 404, 388, 436, 409.

а) Сколько раз включится насос? б) Сколько миллилитров воды он нальёт? в) Через сколько секунд после включения платы начнётся десятая проверка?

Дано: полив при `soil < 400`, 5 с; насос 30 мл/с; после проверки `delay(60000)`; 10 показаний.

Найти: число включений; объём воды; момент 10-й проверки.

Решение

Шаг 1. Какие показания включают насос. Условие `soil < 400` – строго меньше. Идём по списку: 452, 418 – нет; 400 – нет, 400 не меньше 400; 391 – **да**; 447, 425, 404 – нет; 388 – **да**; 436, 409 – нет. Насос включится 2 раза: на 4-й и на 8-й проверке.

Шаг 2. Вода. Одно включение: $30 \cdot 5 = 150$ мл. Два включения: $150 \cdot 2 = 300$ мл.

Шаг 3. Сколько длится один круг `loop()`. Круг без полива – это ожидание `delay(60000)`, 60 с. Круг с поливом – 5 с полива и ещё 60 с ожидания, всего 65 с. Остальные строки (измерение, сравнение) плата выполняет за доли миллисекунды, их не считаем.

Шаг 4. Когда начнётся 10-я проверка. До неё пройдут 9 полных кругов – с 1-го по 9-й. Среди них два круга с поливом (4-й и 8-й) и семь без полива:

$$7 \cdot 60 + 2 \cdot 65 = 420 + 130 = 550 \text{ с.}$$

Это 9 мин 10 с. Можно выписать и все моменты проверок подряд: 0, 60, 120, 180, 245, 305, 365, 425, 490, 550 с. После каждого полива шаг между ними вырастает с 60 до 65 с.

Полив скромный – полтора стакана за десять минут, – но и горшки на сити-ферме небольшие. А лишние 10 с набегали только из-за двух поливов.

 **Типичная ошибка.** Посчитать, что десятая проверка идёт через $10 \cdot 60 = 600$ с. Во-первых, между первой и десятой проверкой 9 промежутков, а не 10. Во-вторых, `delay(5000)` внутри условия тоже тратит время, и круг с поливом длиннее. И третья ловушка – показание 400: насос не включится, условие строгое.

Ответ: а) 2 раза; б) 300 мл; в) через 550 с.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «Программирование для умного сити-фермера»

Реши сам

6.5. Скетч полива проверяет почву, при `soil < 350` включает насос на 8 с (`delay(8000)`), а в конце каждого круга ждёт 30 с (`delay(30000)`). Насос качает 25 мл/с. Показания на первых восьми проверках: 372, 351, 350, 344, 398, 362, 349, 390. Сколько миллилитров воды нальёт насос? Через сколько секунд после включения платы начнётся восьмая проверка?

6.6. В горшок нужно налить 0,9 л воды насосом, который качает 45 мл/с. Какое число написать в `delay(...)` между включением и выключением насоса?

Задача 6.4 Обогреватель, который греет летом

найди ошибку, `pinMode`, знак сравнения

Весной ночи в Новосибирской области холодные, и в теплице ставят обогреватель. Он должен **включаться, когда температура ниже 16 °С**, и выключаться, когда 16 °С и выше. Обогреватель подключён через реле к выводу 5, термометр TMP36 – ко входу A0. Школьник написал скетч, но в нём две ошибки. Найди строки с ошибками. Ответ – два номера строк через запятую, по возрастанию.

```
const int TEMP = A0;           // 1
const int HEATER = 5;          // 2
void setup() {                 // 3
    pinMode(HEATER, INPUT);    // 4
}                               // 5
void loop() {                  // 6
    int n = analogRead(TEMP);  // 7
    float t = (n * 5.0 / 1023 - 0.5) * 100; // 8
    if (t > 16) {              // 9
        digitalWrite(HEATER, HIGH); // 10
    } else {                   // 11
        digitalWrite(HEATER, LOW);  // 12
    }                           // 13
    delay(1000);               // 14
}                               // 15
```

Дано: скетч из 15 строк; нужно: $t < 16$ – обогреватель включён, иначе выключен.

Найти: номера строк с ошибками.

Решение

Шаг 1. Идём по строкам и спрашиваем: что делает эта строка и нужно ли это. Строки 1–2 – константы с номерами выводов, всё в порядке.

Шаг 2. Строка 4. `pinMode(HEATER, INPUT)` делает вывод 5 **входом** – как для кнопки или датчика. Но на выводе 5 висит реле обогревателя, им нужно командовать, значит, вывод должен быть **выходом**: `OUTPUT`. С `INPUT` команда `digitalWrite(5, HIGH)` не даст реле нужного тока, и оно не сработает. Первая ошибка – строка 4.

Шаг 3. Строки 7–8. Показание переводим в вольты и в градусы по формуле для TMP36 из главы 5. Число 5.0 записано с точкой, поэтому деление честное, дробное. Здесь ошибки нет. Проверим на примере: при 10 °С датчик даёт 0,6 В, и `analogRead` возвращает 122. По строке 8: $(122 \cdot 5 : 1023 - 0,5) \cdot 100 \approx 9,6$ °С – верно.

Шаг 4. Строка 9. Написано `t > 16`: «если теплее 16 – включить обогреватель». Это наоборот! Греть нужно, когда холодно: `t < 16`. Вторая ошибка – строка 9. Заодно проверим границу: при 16 °С на термометре условие `t < 16` неверно, сработает `else` и выключит обогреватель, как и требует задание.

Шаг 5. Остальное. Строки 10–14 – включить, иначе выключить, пауза на секунду. Скобки все закрыты.

Оставь строку 9 как есть – и холодной ночью при 10 °С обогреватель стоит, а в июльскую жару при 30 °С – печёт рассаду всю.

❗ Типичная ошибка. Искать ошибку в самой сложной строке – в формуле из строки 8 – и не заметить `INPUT` в «скучном» `setup()`. Ошибки прячутся где угодно. Надёжнее пройти **все** строки по порядку и для каждой сказать вслух, что она делает. А знак сравнения проверь, подставив одно число с каждой стороны от порога: 10 °C и 20 °C.

Ответ: 4, 9.

Своя задача по программе сферы, модуль 1

Реши сам

6.7. Пока дверь теплицы открыта, вентилятор нужно выключать – незачем гнать воздух на улицу. Концевик двери на выводе 2 даёт `HIGH`, когда дверь открыта. В какой строке ошибка? Что будет делать вентилятор, когда дверь закрыта?

```
const int DOOR = 2;           // 1
const int FAN = 9;            // 2
void setup() {                // 3
    pinMode(FAN, OUTPUT);     // 4
}                              // 5
void loop() {                 // 6
    int door = digitalRead(DOOR); // 7
    if (door = HIGH) {        // 8
        digitalWrite(FAN, LOW); // 9
    } else {                  // 10
        digitalWrite(FAN, HIGH); // 11
    }                          // 12
    delay(100);               // 13
}                              // 14
```

6.8. Скетч полива (в `setup()` вывод `PUMP` сделан выходом):

```
void loop() {
    int soil = analogRead(A0);
    if (soil < 400) {
        digitalWrite(PUMP, HIGH);
    }
    delay(1000);
}
```

Датчик на первых четырёх проверках показал 450, 380, 420, 500. После каких проверок насос будет работать? Что нужно дописать в программу, чтобы он выключался, когда почва влажная?

Задача 6.5 Перепутанные строки полива

расставь строки, порядок команд, полив на время

Нужен скетч: сразу после включения платы проверить почву; если число меньше 350 (сухо) – включить насос на 45 с и выключить. Затем в любом случае подождать 10 минут и проверить снова. Насос подключён к выводу PUMP, в setup() он уже сделан выходом. Семь строк функции loop() перемешались:

```
void loop() {  
  // А digitalWrite(PUMP, LOW);  
  // Б int soil = analogRead(A0);  
  // В delay(45000);  
  // Г if (soil < 350) {  
  // Д digitalWrite(PUMP, HIGH);  
  // Е }  
  // Ж delay(600000);  
}
```

а) Расставь строки по порядку. Ответ – семь букв без пробелов. б) Насос качает 2 л/мин. Сколько литров он нальёт за одно включение?

Дано: 7 строк; полив 45 с при `soil < 350`; пауза 10 мин после каждой проверки; насос 2 л/мин.

Найти: порядок строк; объём за одно включение.

Решение

Шаг 1. Что должно быть первым. Прежде чем сравнивать `soil` с порогом, эту переменную нужно создать и положить в неё показание. Значит, первая – строка Б. Да и по условию проверка идёт сразу после включения, без ожидания.

Шаг 2. Развилка. Дальше вопрос «сухо ли?» – строка Г. У неё открывается фигурная скобка, а закрывает её строка Е. Всё, что между ними, выполнится только при сухой почве.

Шаг 3. Что внутри развилки. Полив – это три действия в таком порядке: включить (Д), подождать 45 с (В), выключить (А). Если поставить А сразу после Д, насос выключится через миллионные доли секунды, а потом плата будет 45 с ждать с выключенным насосом.

Шаг 4. Пауза 10 минут. По условию ждём **в любом случае** – и когда полили, и когда нет. Значит, строка Ж стоит после закрывающей скобки Е, снаружи развилки. Проверим:

`delay(600000)` – это $600\,000 : 1000 = 600$ с, а $600 : 60 = 10$ мин, число в скобках верное.

Порядок: Б, Г, Д, В, А, Е, Ж. Прочитаем его как рассказ, пусть датчик покажет 300, 400, 320. В момент 0 почва сухая, насос работает 45 с. Потом 10 минут паузы, и вторая проверка приходится на 645 с: 400, полива нет. Третья – в 1245 с: 320, насос снова включается.

Шаг 5. Литры. 45 с – это $45 : 60 = 0,75$ мин. Насос за это время нальёт $2 \cdot 0,75 = 1,5$ л. Одна большая бутылка из-под лимонада на грядку – обычный полив, не потоп.

 **Типичная ошибка.** Поставить Ж внутрь развилки, перед Е. Тогда при влажной почве программа не ждёт вообще и проверяет датчик тысячи раз в секунду, а условие «подождать в любом случае» нарушено. Другая ошибка – перепутать В и А: насос «поливает» 0 секунд. Прежде чем вводить ответ, прочитай получившуюся программу сверху вниз, как рассказ.

Ответ: а) БГДВАЕЖ; б) 1,5 л.

Своя задача по программе сферы, модуль 7

Реши сам

6.9. Если воды в баке мало (датчик уровня на A2 показывает меньше 100), красная лампа ALARM должна вспыхивать: гореть 300 мс, потом 700 мс не гореть. Если бак не пуст, программа сразу идёт на следующий круг. Расставь строки `loop()`, ответ – семь букв.

```
// А digitalWrite(ALARM, LOW);
// Б int level = analogRead(A2);
// В delay(700);
// Г if (level < 100) {
// Д digitalWrite(ALARM, HIGH);
// Е delay(300);
// Ж }
```

6.10. Увлажнитель должен включаться каждые 30 минут и работать 20 секунд:

```
void loop() {
    digitalWrite(HUM, HIGH);
    delay(20000);
    digitalWrite(HUM, LOW);
    delay(X);
}
```

Какое число нужно поставить вместо `X`, чтобы от одного включения до следующего прошло 30 минут?

Задача 6.6 Какая программа держит два порога

два порога, гистерезис, переменная-память, if без else

Вентилятор теплицы должен включаться при 28 °C и выше, а выключаться только при 24 °C и ниже. Три школьника написали основную часть программы по-разному (в начале `fanOn = false`, а после этих строк вентилятор включают или выключают по `fanOn`):

```
// Программа 1
if (t ≥ 28) fanOn = true;
if (t ≤ 24) fanOn = false;

// Программа 2
if (t ≥ 28) fanOn = true;
else fanOn = false;

// Программа 3
if (t ≥ 28) fanOn = true;
else if (t ≤ 24) fanOn = false;
```

Если после `if (...)` стоит одна команда, фигурные скобки можно не писать – так здесь и сделано.

а) Какие программы работают с двумя порогами, как задумано? Ответ – номера по возрастанию. б) Температуру меряют раз в минуту, 16 минут подряд: 26; 27,5; 28; 27,8; 28,3; 27,9; 28,1; 27; 25,5; 24; 25; 26,8; 28; 29; 27,5; 26. Сколько раз включится вентилятор у программы 1 и у программы 2?

Дано: пороги: включить при $t \geq 28$, выключить при $t \leq 24$; три варианта кода; 16 замеров; вначале выключен.

Найти: правильные программы; число включений у программ 1 и 2.

Решение

Шаг 1. Что делает каждая программа между порогами. Возьмём 26 °С – число между 24 и 28 – и посмотрим, тронет ли программа `fanOn`.

- Программа 1: $26 \geq 28$ – неверно, $26 \leq 24$ – неверно. Ни одна строка не сработала, `fanOn` остался прежним. Два порога – как задумано.
- Программа 2: $26 \geq 28$ – неверно, значит, срабатывает `else`, и `fanOn = false`. Вентилятор выключается уже при 26, хотя до 24 не остыло. Это **один** порог, 28 °С.
- Программа 3: $26 \geq 28$ – неверно, дальше $26 \leq 24$ – тоже неверно, `else if` не сработал, `fanOn` не тронут. Два порога, работает так же, как программа 1.

Шаг 2. Трассировка программы 1. Вентилятор включается, когда впервые дойдёт до 28, и держится, пока не упадёт до 24.

- Минуты 1–2 (26; 27,5): до 28 не дошло, стоит.
- Минута 3 (28): $28 \geq 28$ – **включение первое**. Дальше 27,8; 28,3; 27,9; 28,1; 27; 25,5 – выше 24, работает.
- Минута 10 (24): $24 \leq 24$ – выключился.
- Минуты 11–12 (25; 26,8): стоит.
- Минута 13 (28): **включение второе**. До конца (29; 27,5; 26) не остыл до 24 – работает.

Итого 2 включения.

Шаг 3. Трассировка программы 2. Здесь всё решает одно показание: 28 и выше – работает, ниже – стоит. Работает на минутах 3 (28), 5 (28,3), 7 (28,1), 13 (28) и 14 (29). Между 3-й и 5-й, между 5-й и 7-й минутой был замер ниже 28, и вентилятор успевал выключиться. Включения начинаются на минутах 3, 5, 7 и 13 – это 4 включения, и столько же выключений: реле щёлкнуло 8 раз.

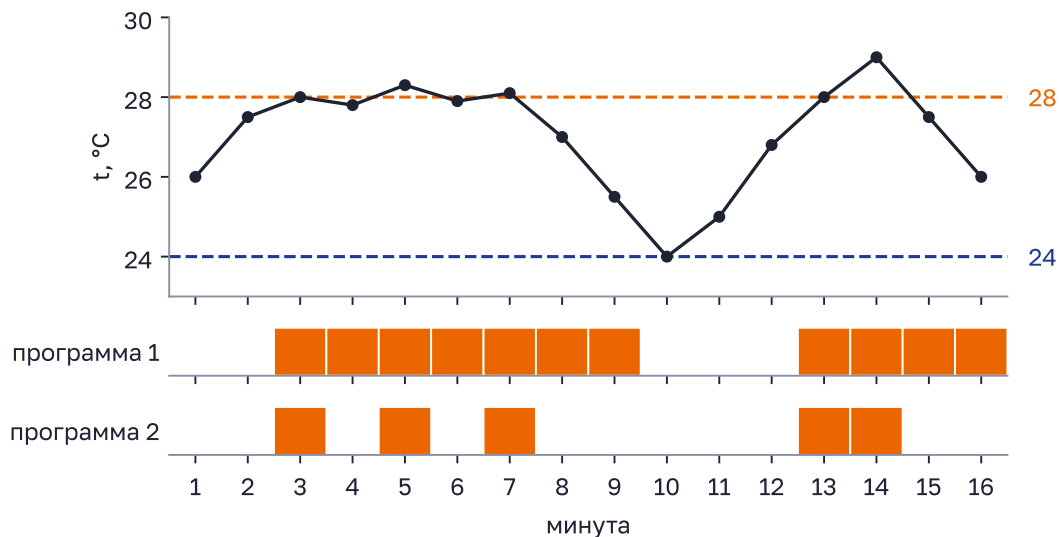


Рисунок 6.11 – Температура и работа вентилятора. Оранжевые клетки – вентилятор работает в эту минуту

На графике видно, чем плохи программы с одним порогом: на минутах 3–7 температура «дрожит» у отметки 28, и вентилятор у программы 2 то включается, то выключается. У программы 1 он спокойно работает все эти минуты. Посчитай оранжевые клетки: у программы 1 их 11, у программы 2 – всего 5. Программа 3 закрасила бы те же 11 клеток, что и первая.

❗ **Типичная ошибка.** Решить, что программа 3 неправильная, раз в ней есть `else`. Опасен голый `else`, который срабатывает на всё подряд. `else if (t ≤ 24)` срабатывает только при 24 и ниже, и между порогами память не затирается. Вторая ошибка – в трассировке засчитать минуту 13 как продолжение первого включения. Вентилятор к тому времени уже выключился на 10-й минуте, так что это новое включение.

Ответ: а) 13 (программы 1 и 3); б) программа 1 – 2 раза, программа 2 – 4 раза.

Своя задача по программе сферы, модуль 7

Реши сам

6.11. Увлажнитель воздуха управляется так (`h` – влажность в процентах, вначале `humOn = false`):

```
if (h ≤ 60) humOn = true;
if (h ≥ 75) humOn = false;
```

Замеры влажности: 68, 63, 60, 64, 70, 74, 75, 71, 66, 61, 59, 65, 76. Сколько раз увлажнитель включится? Будет ли он работать после последнего замера?

6.12*. Обогреватель: `if (t ≤ 16) heatOn = true; if (t ≥ 20) heatOn = false;`, вначале выключен. Замеры каждые 10 минут: 18, 17, 16, 17, 18, 19, 20, 19, 17, 16, 15, 18 °C; каждое показание держится 10 минут, до следующего замера. а) Сколько минут всего проработал обогреватель? б) Сколько минут он проработал бы, если бы второй `if` заменили на `else heatOn = false;` ?

Задача 6.7 *. Форточка в три ступеньки

машина состояний, трассировка, один переход за замер, сервомотор

Форточку теплицы поворачивает сервомотор. У неё три состояния: 0 – закрыта (угол 0°), 1 – щель (30°), 2 – открыта (90°). Переходы показаны на схеме. Температуру меряют раз в 5 минут, и за один замер машина делает не больше одного перехода. В 11:00 форточка закрыта.

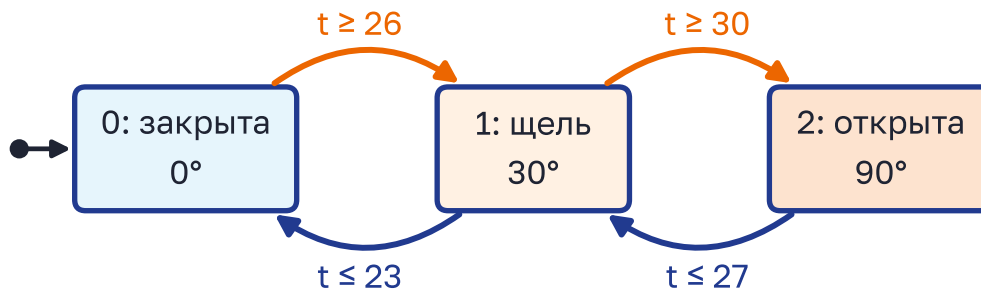


Рисунок 6.12 – Машина состояний форточки. Оранжевые стрелки открывают, синие – закрывают

Та же машина в программе. Сервомотором управляют готовые команды: `Servo window`; заводит мотор с именем `window`, `window.attach(6)` говорит, что он подключён к выводу 6, а `window.write(30)` поворачивает его на 30°.

```
Servo window; // сервомотор форточки
int state = 0; // 0 - закрыта, 1 - щель, 2 - открыта

void go(int s) { // переход: запомнить состояние и повернуть форточку
    state = s;
    if (s == 0) window.write(0);
    if (s == 1) window.write(30);
    if (s == 2) window.write(90);
}

void setup() {
    window.attach(6);
    go(0);
}

void loop() {
    float t = readT(); // температура, °C
    if (state == 0 && t ≥ 26) {
        go(1);
    } else if (state == 1 && t ≥ 30) {
        go(2);
    } else if (state == 1 && t ≤ 23) {
        go(0);
    } else if (state == 2 && t ≤ 27) {
        go(1);
    }
    delay(300000); // следующий замер через 5 минут
}
```

Замеры с 11:00 до 11:55: 24, 31, 32, 29, 27, 31, 26, 24, 22, 27, 30, 25 °C.

а) Запиши номер состояния после каждого замера – строку из 12 цифр. б) Сколько раз сервомотор поворачивал форточку (не считая начальной установки в `setup()`)? в) Сколько минут форточка была открыта полностью, если каждое состояние держится до следующего замера, а после последнего замера – ещё 5 минут?

Дано: $0 \rightarrow 1$ при $t \geq 26$; $1 \rightarrow 2$ при $t \geq 30$; $1 \rightarrow 0$ при $t \leq 23$; $2 \rightarrow 1$ при $t \leq 27$; один переход за замер; 12 замеров через 5 мин; вначале состояние 0.

Найти: строку состояний; число поворотов; минуты в состоянии 2.

Решение

Шаг 1. Из каждого состояния – свои стрелки. В каждом замере смотрим **только** на стрелки, выходящие из текущего состояния. Из 0 – одна стрелка (вверх, при $t \geq 26$). Из 1 – две (вверх при $t \geq 30$, вниз при $t \leq 23$). Из 2 – одна (вниз при $t \leq 27$). Если ни одна не сработала – остаётся. Из состояния 1 выходят две стрелки на одно и то же событие – новое показание, и это не ошибка: условия у них не пересекаются (≥ 30 и ≤ 23), так что машина всегда знает, куда идти. Две стрелки с одним событием и одинаковым условием из одного состояния ставить нельзя.

Шаг 2. Трассировка. Заполняем таблицу строка за строкой.

Таблица 6.4

Время	$t, ^\circ\text{C}$	Было	Сработала стрелка	Стало
11:00	24	0	нет: $24 < 26$	0
11:05	31	0	$0 \rightarrow 1$	1
11:10	32	1	$1 \rightarrow 2$	2
11:15	29	2	нет: $29 > 27$	2
11:20	27	2	$2 \rightarrow 1$: $27 \leq 27$	1
11:25	31	1	$1 \rightarrow 2$	2
11:30	26	2	$2 \rightarrow 1$	1
11:35	24	1	нет: $24 > 23$ и $24 < 30$	1
11:40	22	1	$1 \rightarrow 0$	0
11:45	27	0	$0 \rightarrow 1$	1
11:50	30	1	$1 \rightarrow 2$	2
11:55	25	2	$2 \rightarrow 1$	1

Самое хитрое место – 11:05. Там 31°C , и хочется сразу распахнуть форточку. Но из состояния 0 есть только стрелка в 1, а за один замер делается один переход. Форточка встанет на щель и откроется полностью лишь в 11:10. В программе это видно по цепочке `else if`: после `go(1)` остальные ветки в этом круге уже не проверяются.

Шаг 3. Считаем повороты. Поворот – каждый раз, когда состояние поменялось. Выпишем строку: 0, 1, 2, 2, 1, 2, 1, 1, 0, 1, 2, 1. Считаем соседние пары, где цифры разные: $0 \rightarrow 1$, $1 \rightarrow 2$, $2 \rightarrow 1$, $1 \rightarrow 2$, $2 \rightarrow 1$, $1 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 2$, $2 \rightarrow 1$. Их 9.

Шаг 4. Минуты в состоянии 2. Состояние 2 наступило после замеров 11:10 и 11:15 (с 11:10 до 11:20 – 10 мин), после 11:25 (до 11:30 – 5 мин) и после 11:50 (до 11:55 – 5 мин). Всего $10 + 5 + 5 = 20$ мин.

За час форточка шевельнулась 9 раз. В жаркий день с облаками так и бывает: солнце выглянуло – открылась, спряталось – прикрылась.

 **Типичная ошибка.** В 11:05 перескочить из 0 сразу в 2, «потому что 31 больше 30». Тогда вся строка дальше съедет. Вторая ловушка – 11:35, 24 °C: кажется, что «холодно, закрыть». Но стрелка из 1 в 0 требует 23 и ниже, а 24 – между порогами, форточка остаётся на щели. Третья – 11:20: при 27 °C стрелка из 2 срабатывает, потому что там знак $\leq$.

Ответ: а) 012212110121; б) 9 раз; в) 20 мин.

Своя задача по программе сферы, модуль 1

Реши сам

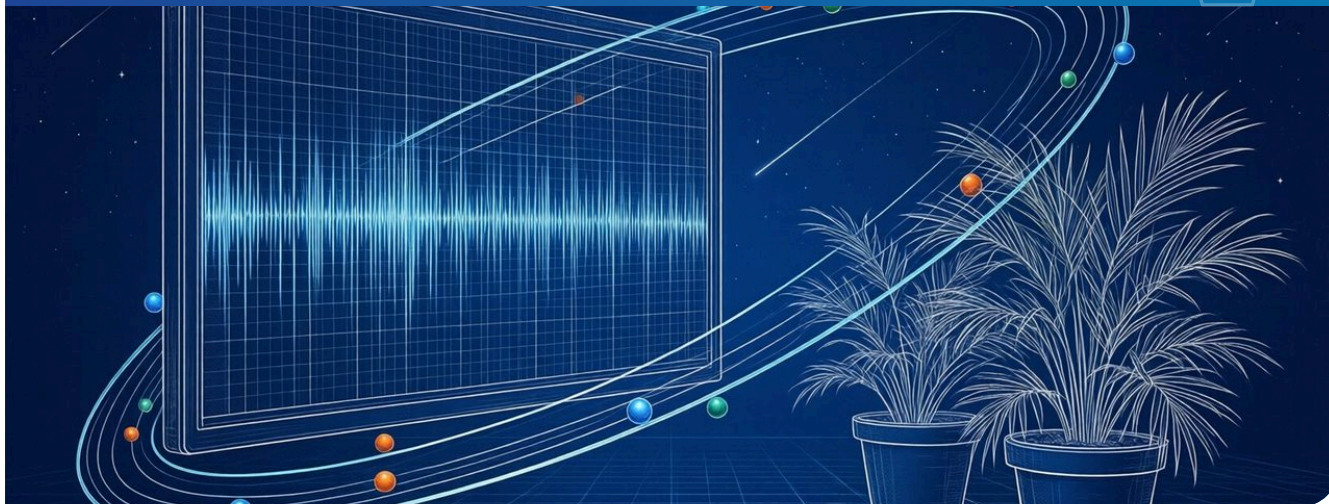
6.13*. Лампа досветки – машина с тремя состояниями: 0 – выключена, 1 – вполсилы, 2 – полная яркость. Фоторезистор включён как в курсе: чем темнее, тем больше число. Переходы: 0 → 1 при показании больше 500; 1 → 2 – больше 750; 2 → 1 – меньше 650; 1 → 0 – меньше 400. За один замер – не больше одного перехода, вначале лампа выключена. Показания: 280, 420, 520, 700, 760, 740, 660, 640, 580, 410, 390, 550. Запиши строку состояний после каждого замера. Сколько раз лампа меняла режим?

6.14. Машина «полив с тревогой» имеет четыре состояния и такие переходы:

- «ждём» → «поливаем»: новое показание [почва < 400] / включить насос;
- «поливаем» → «впитывается»: прошло 10 с [почва ≥ 200] / выключить насос;
- «поливаем» → «тревога»: прошло 10 с [почва < 200] / выключить насос, мигать красным;
- «впитывается» → «ждём»: прошло 60 с.

(Если после 10 с полива датчик всё ещё показывает почти сухо, значит, пуст бак или датчик выпал из земли.) Какое состояние – тупик (конечным оно не отмечено)? Придумай переход, который из него выведет.

Сигналы и обработка данных



Датчик в теплице не знает слова «жарко». Он каждые несколько секунд присылает число, и среди этих чисел попадают дрожание, случайные, а иногда совсем дикие – 85 °С майским утром. В этой главе учимся отделять то, что в теплице меняется по-настоящему, от помех: сглаживать ряд, ловить выбросы, считать события и период, прикидывать, как часто мерить и как число превращается в нули и единицы. На отборе такая задача приходит рядом чисел или графиком с отсчётами, а в ответ вводится одно число.

7.1. Сигнал и шум

НА ПАЛЬЦАХ



Рисунок 7.1 – Плавная линия плюс мелкая дрожь – получается то, что записал датчик

Температура в теплице меняется плавно. Датчик добавляет к ней случайную дрожь. В записи мы видим сумму, и наша работа – вытащить из неё плавную линию.

Сигнал – величина, которая меняется со временем и этим что-то сообщает. Температура в теплице, освещённость над грядкой, напряжение на выходе фоторезистора, голос в телефоне – всё это сигналы. Когда в школьной столовой друг рассказывает тебе новость, его голос – сигнал, а гул сотни других голосов вокруг – помеха. Такие помехи называют **шумом**: мелкие случайные изменения, которые к делу не относятся. Шумят провода (в них наводятся помехи от моторов и соседней проводки), шумит сам датчик, шумит воздух, когда по теплице гуляет сквозняк.

У шума узнаваемый почерк: он мелкий, быстрый и беспорядочный. Воздух в теплице за минуту нагревается на десятые доли градуса, а соседние отсчёты термометра могут прыгать туда-сюда на полградуса. Числа то растут, то падают без всякой системы – скорее всего, это шум. Растут долго и дружно – это сигнал: например, из-за туч вышло солнце.

Аналоговый сигнал меняется плавно и может принять любое значение: напряжение на фоторезисторе бывает и 2,31 В, и 2,312 В. **Цифровой** сигнал принимает только строго определённые значения: «ноль» и «единица», и меняется скачками. Помеха портит такие сигналы по-разному: аналоговый «портится понемногу» (шипит, искажается), а цифровой передаётся целиком или пропадает совсем. Arduino превращает напряжение в целое число от 0 до 1023 (как – разобрано в главе 5), то есть **оцифровывает** сигнал: после этого у нас ряд чисел. Дальше в этой главе мы работаем уже с числами.

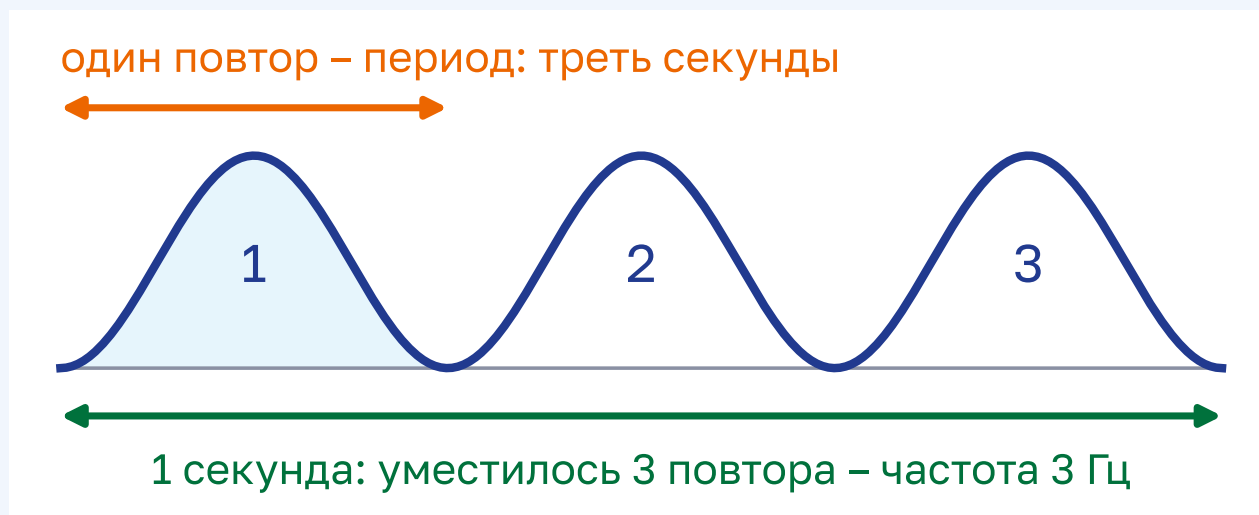


Рисунок 7.2 – Три одинаковых горба уместились в одну секунду: частота 3 Гц, а один горб длится треть секунды

Период – сколько длится один повтор. **Частота** – сколько повторов влезает в секунду. Узнал одно – узнаешь и другое.

Волна. Многие сигналы повторяются. Днём в теплице тепло, ночью прохладно, на завтра всё сначала. Такой сигнал называют **периодическим**, а самый плавный из них, похожий на волну на воде, – **гармоническим**. В курсе волну описывают тремя числами: амплитудой, частотой и фазой. Период – это частота, записанная «наоборот». Начнём с периода, амплитуды и частоты, а про фазу – во врезке ниже.

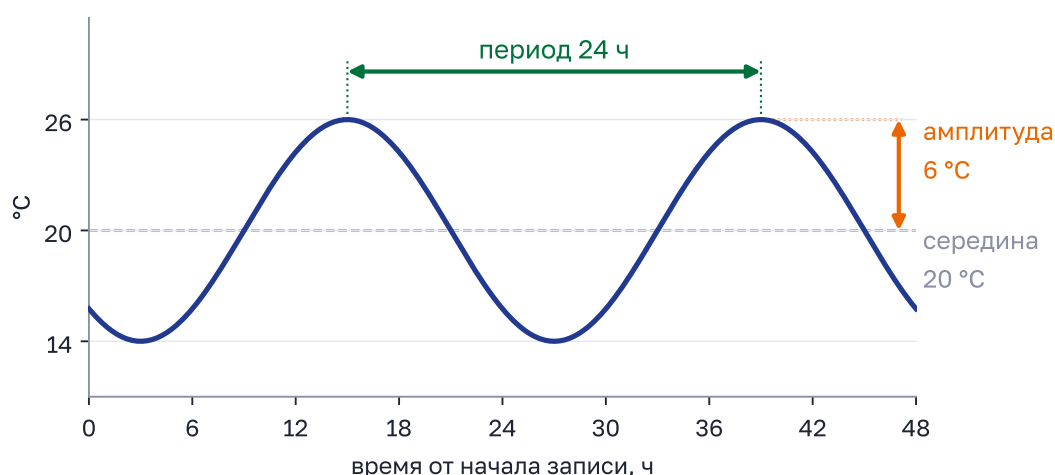


Рисунок 7.3 – Температура в теплице за двое суток: середина 20 °С, амплитуда 6 °С, период 24 ч

- **Период T** – время одного полного повтора, например от вершины до следующей вершины. У суточного хода температуры период 24 ч.
- **Амплитуда A** – на сколько волна поднимается над своей серединой (и на столько же опускается под неё). Это половина размаха: $A = (\text{максимум} - \text{минимум}) : 2$. Днём до 26 °С, ночью до 14 °С – размах 12 °С, амплитуда 6 °С, середина $(26 + 14) : 2 = 20$ °С. Размах ты уже считал в главе 3.
- **Частота f** – сколько повторов укладывается в единицу времени. Частота и период «переворачивают» друг друга: $f = 1 : T$ и $T = 1 : f$.

Частоту в повторах за секунду измеряют в **герцах** (Гц): 1 Гц – один повтор в секунду. Вентилятор, который делает 5 оборотов в секунду, крутится с частотой 5 Гц, и один оборот занимает $1:5 = 0,2$ с. Напряжение в розетке делает 50 полных колебаний в секунду – это 50 Гц, период $1:50 = 0,02$ с, или 20 мс. Для суточного хода температуры герцы неудобны (выходит 0,0000116 Гц), поэтому говорят проще: один раз в сутки.

Фаза и формула волны. В курсе гармоническую волну записывают формулой $x(t) = A \cdot \sin(\omega t + \varphi_0)$. Здесь A – амплитуда, $\omega = 2\pi f$ – скорость вращения (частоту в курсе обозначают буквой ν , мы пишем f), а φ_0 – **фаза**: с какого места волна начинается. Если $\varphi_0 = 0$, в момент $t = 0$ волна идёт через середину вверх; другая фаза просто сдвигает всю картинку по времени. Две волны одной частоты с разницей фаз в половину оборота (180°) идут в **противофазе**: одна вверх, другая вниз, и сложенные вместе они гасят друг друга. На этом держится шумоподавление в наушниках. В задачах главы фаза не нужна, но в курсе встретишь.

Запомни. Сигнал – то, что меняется по делу, шум – мелкая беспорядочная дрожь поверх него. У волны период T – время одного повтора, частота $f = 1:T$ (в герцах – повторов в секунду), амплитуда – половина размаха.

Совет. Период часто дают в миллисекундах. Прежде чем делить единицу на период, переведи его в секунды: 20 мс = 0,02 с, и тогда $1:0,02 = 50$ Гц. Разделишь 1 на 20 – получишь 0,05 «чего-то», и ответ уедет в тысячу раз.

7.2. Дискретизация: датчик меряет не всё время

НА ПАЛЬЦАХ

между точками датчик
ничего не видит

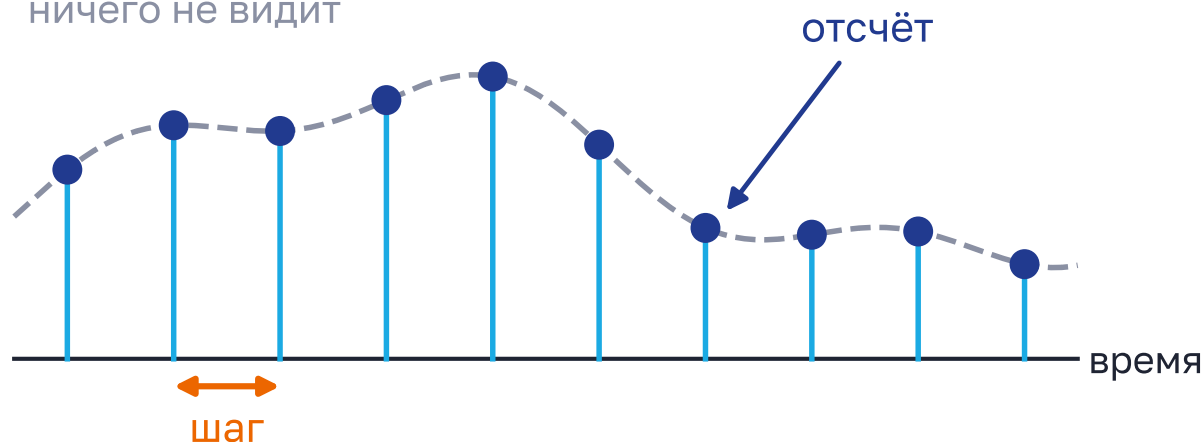


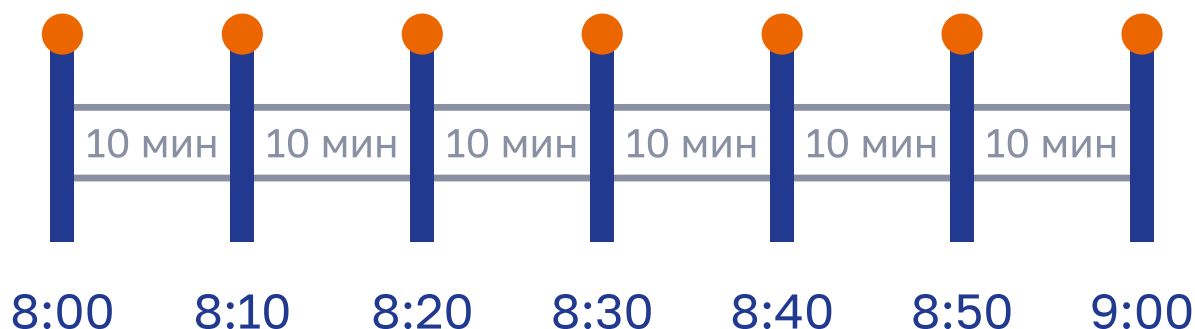
Рисунок 7.4 – Датчик «моргает»: замер – пауза – замер

Линия – как менялась температура. Точки – что успел увидеть датчик. Что было между точками, программа не знает.

Arduino не смотрит на датчик непрерывно. Программа читает его командой `analogRead`, потом ждёт или занимается другими делами, потом читает снова. Одно такое измерение называют **отсчётом**, а превращение плавного сигнала в ряд отсчётов – **дискретизацией** (латинское *discretus* значит «разделённый, прерывистый»). Промежуток между соседними отсчётами называют **шагом**, а число отсчётов за секунду – **частотой дискретизации**.

Если в `loop()` стоит `delay(500)`, отсчёт делается раз в 0,5 с: 2 отсчёта в секунду, 120 в минуту. Сама программа тоже тратит время, но короткий код выполняется за доли миллисекунды, и в задачах этим пренебрегают, если не сказано иначе. Метеостанция, которая пишет раз в 10 мин, делает 6 отсчётов в час и $24 \cdot 6 = 144$ отсчёта в сутки.

7 столбов-замеров



6 пролётов-промежутков

Рисунок 7.5 – Замеры с 8:00 до 9:00 через 10 минут: пролётов шесть, а столбов семь

Замер – столб, промежуток между замерами – пролёт. Когда замеры стоят на обоих концах, столбов на один больше.

Столбы и пролёты. Кто строил забор на даче, знает: на 6 пролётов уходит 7 столбов. С отсчётами то же самое. Час с 8:00 до 9:00 с шагом 10 мин – это 6 промежутков, а замеров 7, считая оба конца. Когда в счёт идут и первый, и последний замер,

$$n = \frac{\text{длительность}}{\text{шаг}} + 1,$$

где n – число отсчётов. А если датчик пишет круглые сутки подряд – 0:00, 0:10, ..., 23:50, а следующий 0:00 уже относится к новым суткам, – отсчётов 144, без прибавки единицы. Так что сначала посмотри, где начало и где конец.

Каждый отсчёт – кусочек времени. Когда отсчёты идут с одинаковым шагом, считают, что каждый отвечает за один шаг времени. Тогда «сколько времени было жарко» находят так:

$$\text{время} = \text{число «жарких» отсчётов} \cdot \text{шаг}.$$

Пример: замеры раз в 15 мин, и в 6 из них температура была 28 °C и выше. Жарко было около $6 \cdot 15 = 90$ мин, то есть полтора часа. «Около» – потому что между замерами датчик ничего не видел.

Слишком редко – значит, вслепую. Мерь температуру в теплице раз в сутки в 13:00 – и ряд выйдет ровный и тёплый, а ночной заморозок программа не заметит ни разу. Чтобы колебание вообще попало в запись, на один его период нужно больше двух отсчётов, а чтобы разглядеть форму – десяток и больше. Для суточного хода замер раз в час (24 отсчёта на период) уже хорош, раз в 12 часов – на самой грани, раз в сутки – бесполезен. Ты видел это и сам: на видео колёса машины иногда будто крутятся назад или стоят на месте. Камера снимает кадры-отсчёты слишком редко для такого быстрого вращения.

Для любопытных. Правило «больше двух отсчётов на период» в России называют теоремой Котельникова: инженер-связист Владимир Котельников доказал её в 1933 году. На ней держится вся цифровая запись звука и видео.

Запомни. Отсчётов на один больше, чем промежутков, если считаешь с обоих концов. Время по отсчётам = число отсчётов · шаг. Два отсчёта на период и меньше – колебание может пропасть из записи.

7.3. Детектирование: по уровню и по частоте

НА ПАЛЬЦАХ

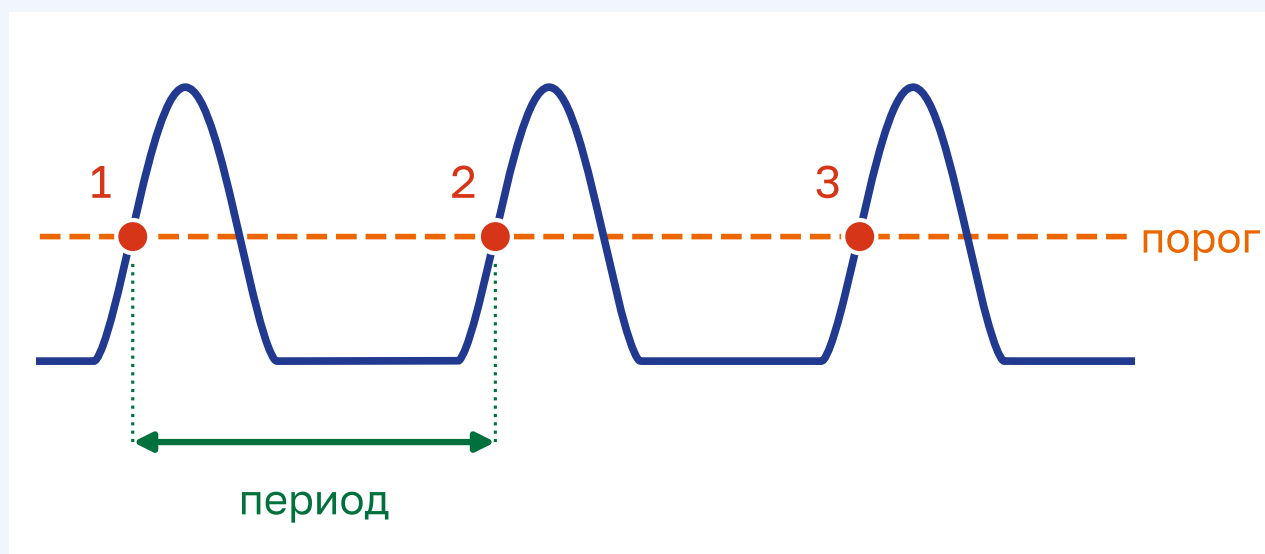


Рисунок 7.6 – Сигнал трижды перешагнул порог снизу вверх – три срабатывания

Считаем не точки выше черты, а моменты, когда линия её перешагнула снизу вверх. Расстояние между такими моментами – период.

Детектировать сигнал – обнаружить в нём событие: «стало светло», «прошла капля», «метка на лопасти пролетела мимо датчика». Способов два, оба простые. Прибор бывает и совсем без детектирования: термометр просто показывает число и ничего не ждёт.

Детектирование по уровню. Выбираем число-порог. Сигнал поднялся до порога или выше – событие произошло. Про пороги для форточки и насоса уже шла речь в главе 2, а здесь научимся **считать события**. Пусть порог 500, а отсчёты такие:

120, 130, 610, 640, 150, 140, 700, 160, 130, 580, 620, 590, 150.

Выше порога 6 отсчётов, а событий три: на 3-м, 7-м и 10-м отсчёте сигнал **перешагнул порог снизу вверх**. (Так мы договорились считать в этой главе. В других заданиях событием может быть переход в любую сторону – и когда стало сильнее, и когда слабее; строгость порога («при значении, равном порогу, событие уже есть или ещё нет») тоже всегда оговаривает условие.) Пока он держится выше порога (отсчёты 3–4 или 10–12), это то же самое событие,

просто долгое. Чтобы найти события, ищи пары соседей, где левый ниже порога, а правый – на пороге или выше. Такие места называют **фронтами**.

Детектирование по частоте (по периоду). Бывает, что нужно знать, как часто события повторяются. Анемометр на метеостанции – вертушка с чашками – на каждом обороте один раз даёт импульс. Датчик капель на капельнице – импульс на каждую каплю. Засаекаем моменты событий и находим период – время между соседними событиями.

Пример. Импульсы пришли в моменты 3,0 с; 5,5 с; 8,0 с; 10,5 с. Между первым и последним прошло $10,5 - 3,0 = 7,5$ с, а промежутков между четырьмя импульсами – три (опять столбы и пролёты). Период $T = 7,5 : 3 = 2,5$ с, частота $f = 1 : 2,5 = 0,4$ Гц, то есть $60 : 2,5 = 24$ импульса в минуту.

$$T = \frac{t_{\text{последнего}} - t_{\text{первого}}}{\text{число событий} - 1}.$$

Почему через первый и последний, а не по двум соседним? Из-за шума соседние промежутки чуть гуляют: 2,4 с, 2,6 с, 2,5 с. Длинный отрезок, делённый на число промежутков, сам усредняет эту неточность.

Запомни. Событие – момент, когда сигнал перешагнул порог снизу вверх, а не каждый отсчёт выше порога. Период = (время последнего события – время первого) : (число событий – 1).

Совет. Если моменты событий даны номерами отсчётов, сначала найди период в отсчётах, а потом умножь на шаг: 8 отсчётов при шаге 0,01 с – это 0,08 с.

7.4. Сглаживание: скользящее среднее

НА ПАЛЬЦАХ



Рисунок 7.7 – Окошко на три клетки едет по ряду и каждый раз записывает среднее

Под каждым положением окошка – среднее трёх чисел внутри. Сдвинули окошко на одну клетку – посчитали снова.

Самый простой способ убрать шум – усреднить. Случайные ошибки то в плюс, то в минус, и в сумме нескольких соседних отсчётов они частично гасят друг друга. А медленное измене-

ние, которое нам и нужно, от усреднения почти не страдает.

Скользящее среднее по k отсчётам: берём первые k чисел ряда и находим их среднее. Потом сдвигаем «окно» на одно число вправо – выбрасываем самое старое, добавляем следующее – и снова считаем среднее. И так до конца ряда. Число k называют **размером окна**.

Возьмём восемь отсчётов температуры и окно 3:

Таблица 7.1

1-й в окне	2-й	3-й	Сумма	Среднее, °C
20	22	21	63	21
22	21	25	68	22,7
21	25	23	69	23
25	23	24	72	24
23	24	28	75	25
24	28	26	78	26

Сырой ряд прыгал вверх-вниз (20, 22, 21, 25, 23 ...), а сглаженный растёт ступенька за ступенькой: 21; 22,7; 23; 24; 25; 26. Стало видно главное: в теплице теплеет примерно на градус за замер.

Обрати внимание, что чисел стало меньше. Из 8 отсчётов при окне 3 получилось 6 средних: первое окно занимает отсчёты 1–3, последнее – 6–8. Правило:

$$\text{число средних} = n - k + 1,$$

где n – сколько было отсчётов, k – размер окна.

Чем больше окно, тем глаже линия – но и тем больше она опаздывает. Среднее за последние 9 минут откликнется на открытую форточку позже, чем среднее за 3 минуты: в окне ещё сидят старые, тёплые значения. А острые вершины широкое окно срезает. Поэтому окно выбирают так, чтобы шум ушёл, а нужные изменения остались: для термометра теплицы хватает 3–5 отсчётов.



Рисунок 7.8 – Шумный ряд, сглаженный окном 3 и окном 9. Окно 9 глаже, но опаздывает и срезает вершину

На Arduino часто делают ещё проще: вместо одного отсчёта читают датчик 10 раз подряд и берут среднее. Разберём код по строчкам.

```
void setup() {
  Serial.begin(9600);           // включаем связь с компьютером
}

void loop() {
  long summa = 0;               // сюда будем складывать отсчёты
  for (int i = 0; i < 10; i++) { // повторить 10 раз
    summa = summa + analogRead(A0); // прочитать датчик и прибавить к сумме
    delay(10);                     // подождать 10 мс
  }
  int srednee = summa / 10;      // среднее из десяти отсчётов
  Serial.println(srednee);       // отправить его в монитор порта
  delay(1000);                  // следующий замер через секунду
}
```

Строка `for (int i = 0; i < 10; i++)` – это цикл: всё, что внутри фигурных скобок после неё, выполнится 10 раз. На Arduino Uno тип `int` вмещает целые числа только до 32 767. Десять отсчётов (сумма не больше 10 230) в него влезают, а сто – уже нет, поэтому для суммы по привычке берут `long`: он вмещает числа больше двух миллиардов. При делении `summa / 10` дробная часть отбрасывается: для показаний 0...1023 это не страшно.

Запомни. Скользящее среднее – среднее из k последних отсчётов, окно сдвигается на один. Из n отсчётов получается $n - k + 1$ средних. Широкое окно – глаже, но с опозданием.

7.5. Выбросы: когда датчик соврал

НА ПАЛЬЦАХ

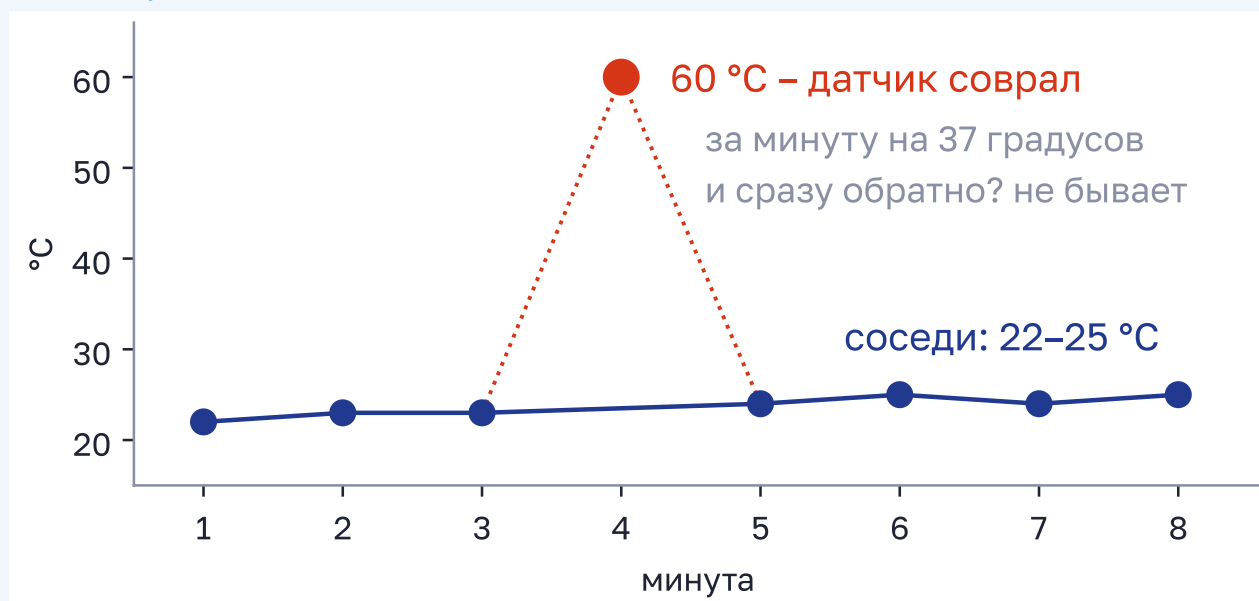


Рисунок 7.9 – Одна точка улетела вверх. За минуту воздух так не нагреется – это выброс

Соседи говорят «22–25 °C», а одна точка кричит «60». Верить ей нельзя: так не бывает.

Выброс – отсчёт, который резко отличается от соседей, причём так, как в жизни быть не может. Отошёл контакт – датчик влажности почвы показал 0 %. Солнечный луч упал прямо на корпус термометра – 60 °C посреди прохладного утра. В проводах проскочила помеха от мотора – одно число оказалось случайным. Шум – это дрожь на полградуса, а выброс – прыжок, после которого значение тут же возвращается обратно.

Как распознать выброс? Задай себе вопрос: могла ли величина так быстро измениться? Воздух в теплице за минуту не нагреется на 30 градусов, почва за секунду не высохнет с 45 % до 0 %. В задачах правило для выброса дают прямо: «отсчёт отличается от обоих соседей больше чем на 10 °C». Ещё помогают границы датчика: если термометр по паспорту меряет от –40 до +125 °C, показание 300 °C – заведомая ошибка.

Среднее боится выбросов. Шесть отсчётов: 22, 23, 23, 60, 24, 25 °C. Среднее $(22 + 23 + 23 + 60 + 24 + 25) : 6 = 177 : 6 = 29,5$ °C. Ни один честный отсчёт не был выше 25 °C, а среднее вышло 29,5 °C – одно враньё испортило всё. Скользящее среднее тоже не спасает: окно «23, 60, 24» даёт $107 : 3 = 35,7$ °C, и выброс размазывается сразу на три соседних результата вместо одного.



Рисунок 7.10 – Три числа выстроили по росту. Выброс уехал на край, посередине осталось честное 24

Медиана – тот, кто стоит посередине, когда всех построили по росту. Самому высокому туда никак не попасть.

Медиана выбросов не боится. Медиана – число, которое оказывается посередине, если выстроить числа по возрастанию. Для трёх чисел: расставь их по порядку и возьми среднее по месту. Для окна «23, 60, 24» порядок такой: 23, 24, 60 – медиана 24. Выброс ушёл на край и в ответ не попал. Если с каждым отсчётом делать так – брать медиану из него и двух соседей, – получится **медианный фильтр**. У первого и последнего отсчёта одного из соседей нет, их оставляют как есть.

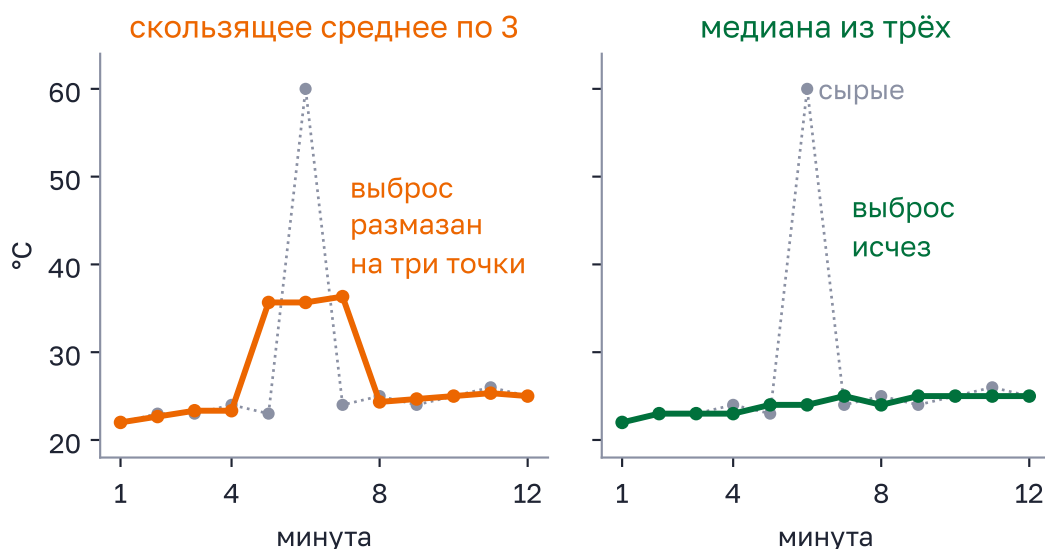


Рисунок 7.11 – Один и тот же ряд с выбросом: скользящее среднее размазывает выброс, медиана из трёх его убирает

У медианы для целого ряда то же правило: 22, 23, 23, 24, 25, 60 – при чётном количестве посередине два числа, 23 и 24, и медиана равна их среднему, 23,5 °C. Сравни со средним 29,5 °C: медиана описывает эти шесть замеров честнее.

Медиана из трёх немного подрезает и настоящие вершины: одиночный честный пик она тоже примет за выброс. Поэтому иногда медианой заменяют только те отсчёты, которые подозрительны по правилу.

Ложные срабатывания. Выброс опасен не только для отчёта. Если программа сравнивает каждый отсчёт с порогом, случайные 60 °С включат вентилятор, и через минуту он выключится, – это **ложное срабатывание**. Защищаются двумя способами. Можно сначала пропустить отсчёты через медианный фильтр, а потом сравнивать с порогом. А можно требовать подтверждения: включать, только если порог превышен в нескольких отсчётах подряд, например в трёх. Выброс длится один отсчёт и подтверждения не получит. Цена защиты – опоздание: настоящий перегрев программа тоже заметит на два отсчёта позже.

```
from statistics import median

ryad = [22, 23, 23, 60, 24, 25]
filtr = [ryad[0]] # первый отсчёт – как есть
for i in range(1, len(ryad) - 1):
    filtr.append(median(ryad[i-1:i+2])) # медиана из отсчёта и двух соседей
filtr.append(ryad[-1]) # последний – как есть
print(filtr) # [22, 23, 23, 24, 25, 25]
```

`ryad[i-1:i+2]` – это три числа ряда: предыдущее, текущее и следующее. Функция `median` сама расставит их по порядку и вернёт среднее по месту.

Запомни. Выброс – одиночный прыжок, невозможный в жизни. Среднее он портит, медиана из трёх его убирает. Против ложных срабатываний – медианный фильтр или «несколько отсчётов подряд».

7.6. Спектр и фильтр – на пальцах

НА ПАЛЬЦАХ

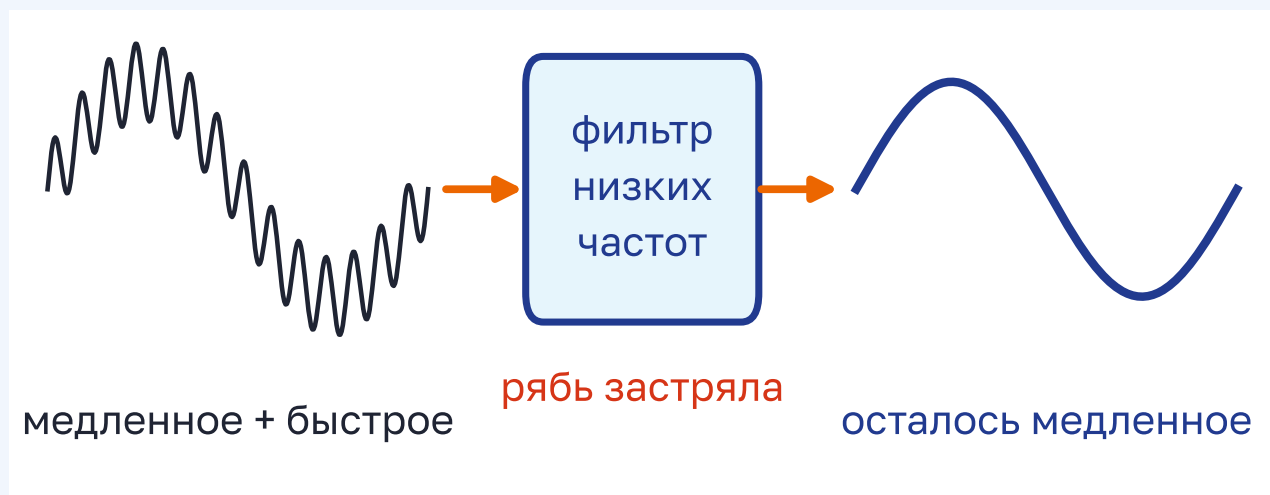


Рисунок 7.12 – Медленная волна плюс быстрая рябь. Фильтр низких частот пропускает медленное и задерживает быстрое

Фильтр – как сито для колебаний. Медленная волна проходит насквозь, мелкая быстрая рябь застревает.

Посмотри ещё раз на запись температуры из раздела 7.1. В ней сразу несколько колебаний. Самое медленное – сутки: днём тепло, ночью холодно. Побystрее – форточка: открылась, воздух остыл, закрылась, нагрелся, и так каждые 20–30 минут. Совсем быстрое – дрожь датчика, от отсчёта к отсчёту. Любой сигнал можно разложить на такие простые волны с разными частотами.

Спектр – картинка, на которой видно, какие частоты есть в сигнале и насколько каждая сильна. Её рисуют столбиками: по горизонтали частота, по вертикали амплитуда. Ты видел спектр на экране музыкального плеера или колонки: прыгающие полоски эквалайзера слева показывают басы (низкие частоты), справа – писк и шелест (высокие).

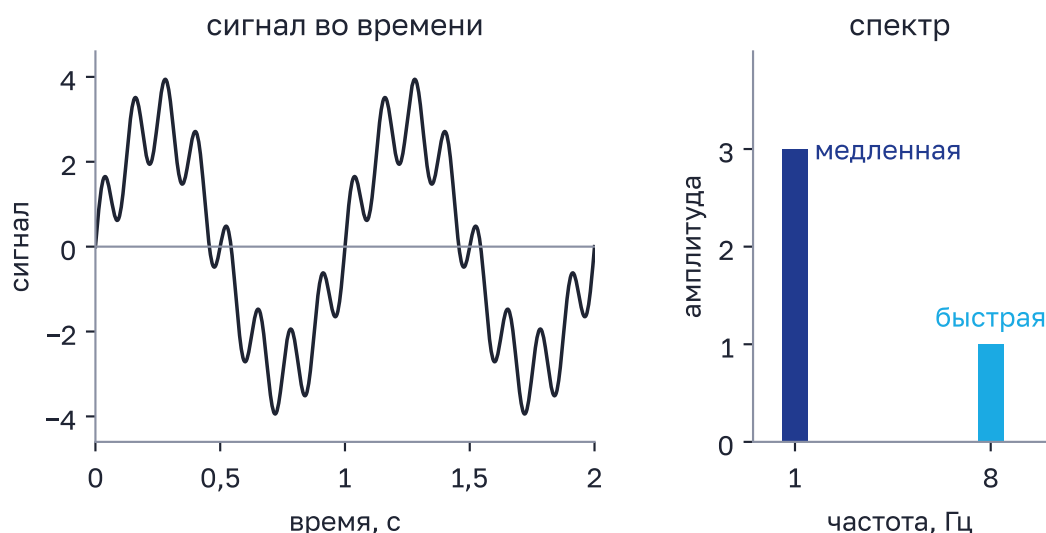


Рисунок 7.13 – Слева – сигнал во времени, справа – его спектр: два столбика, медленная волна с большой амплитудой и быстрая с маленькой

Фильтр убирает из сигнала часть частот. **Фильтр низких частот** (ФНЧ) пропускает медленные колебания и задерживает быстрые. Это то, что нужно для датчика: полезный сигнал в теплице меняется медленно, а шум – быстрый. Скользящее среднее из раздела 7.4 и есть простой ФНЧ. Быстрая дрожь в окне успевает несколько раз сходить вверх и вниз и гасит себя сама, а медленная волна за время окна почти не меняется и проходит насквозь. Бывает и **фильтр высоких частот** – наоборот, он убирает медленное и оставляет быстрое. Им, например, ловят вибрацию: метеостанцию трясёт ветер, а медленный наклон мачты не интересен.

Одна полезная хитрость. Если помеха повторяется с известным периодом, окно скользящего среднего делают **длиной в этот период**. Тогда в каждое окно попадает весь повтор помехи целиком: и её подъём, и её спад. Плюсы и минусы складываются в ноль, и помеха исчезает полностью. Так убирают мерцание ламп досветки из показаний датчика освещённости – это задача 7.7.

Запомни. Спектр показывает, из каких частот сложен сигнал. ФНЧ пропускает медленное и убирает быстрое; скользящее среднее – простой ФНЧ. Окно длиной в период помехи убирает её целиком.

7.7. Кодирование: сигнал несёт число

НА ПАЛЬЦАХ

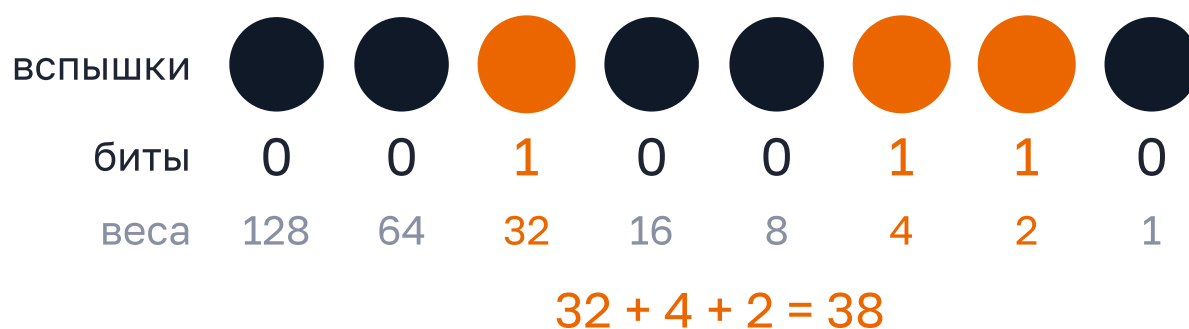


Рисунок 7.14 – Восемь вспышек фонарика – восемь цифр 0 и 1. Горит – 1, темно – 0

У каждой клетки свой вес: 128, 64, 32 ... 1. Складываем веса горящих клеток – получаем число.

Метеостанция стоит в огороде, а данные нужны в доме. Можно передать их по проводу, по радио или даже миганием фонарика – но в любом случае передаётся сигнал, у которого два состояния: есть и нет, высокое напряжение и низкое, вспышка и темнота. Правило, по которому сообщение превращают в такие сигналы, называют **кодом**.

Азбука Морзе – старый код из коротких и длинных сигналов: точек и тире. Сигнал бедствия SOS – три точки, три тире, три точки. Сейчас почти везде используют **двоичный код**: только два знака, 0 и 1. Один такой знак называют **битом**, восемь битов – **байтом**.

Двоичное число. В обычной записи у каждой цифры свой вес: единицы, десятки, сотни – каждый следующий в 10 раз больше. В двоичной записи вес каждого следующего разряда в 2 раза больше: 1, 2, 4, 8, 16, 32, 64, 128 (справа налево). Чтобы перевести двоичное число в обычное, сложи веса тех разрядов, где стоит 1:

$$1011_2 = 8 + 0 + 2 + 1 = 11, \quad 00100110_2 = 32 + 4 + 2 = 38.$$

Маленькая двойка внизу напоминает, что число записано в двоичной системе.

Сколько вариантов уместается в n битов. Один бит – 2 варианта (0 или 1), два бита – 4 (00, 01, 10, 11), и каждый следующий бит удваивает число вариантов:

Таблица 7.2

Битов	Вариантов
1	2
2	4
3	8
4	16
8	256
10	1024

Отсюда знакомые числа: 10 битов – $2^{10} = 1024$ варианта, поэтому `analogRead` выдаёт от 0 до 1023. 8 битов – 256 вариантов, поэтому ШИМ-команда `analogWrite` принимает от 0 до 255.

Обратный вопрос: **сколько битов нужно**, чтобы передать N разных значений? Ищем самую маленькую степень двойки, не меньшую N . Состояний у теплицы пять («норма», «жарко», «холодно», «сухо», «темно») – двух битов (4 варианта) мало, трёх (8 вариантов) хватит. Влажность в целых процентах от 0 до 100 – это 101 значение: 6 битов дают 64, мало, 7 битов – 128, хватит.

Отрицательные числа. Если код – просто номер от 0 до 255, минусов в нём нет. Выход простой: перед передачей ко всем температурам прибавляют одно и то же число, например 50. Тогда $-12\text{ }^{\circ}\text{C}$ передаётся кодом 38, а приёмник вычитает 50 обратно. Такую добавку называют **сдвигом**.

Скорость передачи считают в битах в секунду. Метеостанция раз в сутки отправляет 24 числа по 8 битов, по одному на каждый час, – это 192 бита. На скорости 4 бита в секунду (так быстро можно мигать фонариком руками) передача займёт $192:4 = 48$ с, а обычный радиомодуль справится за доли секунды.

Бит чётности. Помеха может превратить 0 в 1. Чтобы это заметить, к каждому байту добавляют ещё один бит: 1, если единиц в байте нечётное число, и 0, если чётное. Тогда во всех девяти битах вместе единиц всегда чётное количество. Приёмник пересчитывает единицы: нечётно – значит, по дороге что-то испортилось, и байт просят прислать заново. Одну ошибку так видно всегда, а две сразу – уже нет.

Для любопытных. Чтобы сигнал ушёл далеко, его «сажают» на быструю волну – **несущую**: радиоволну, свет лампы, звук. Меняя у несущей силу или частоту в такт сообщению, получают **модуляцию**, а линию, по которой меняется сила несущей, – **огибающей**. Мигание фонариком – самая простая модуляция: несущая – свет, огибающая – «горит – не горит».

Запомни. Двоичное число переводят, складывая веса разрядов с единицами: 128, 64, 32, 16, 8, 4, 2, 1. n битов дают 2^n вариантов. Нужно N значений – бери такое наименьшее n , что $2^n \geq N$.

7.8. Разбираем задачи

Задача 7.1 Сколько записей за день

разминка · дискретизация, число отсчётов, байты

Датчик влажности почвы на даче делает замер каждые 15 мин. Первый замер – в 7:00, последний – в 19:00. а) Сколько отсчётов запишется за день? б) Каждый отсчёт занимает в памяти 4 байта (показание и время). Сколько байтов займёт день записей?

Дано: шаг 15 мин; с 7:00 до 19:00, оба замера считаются; 4 байта на отсчёт.

Найти: n – ?; объём в байтах – ?

Решение

Шаг 1. Длительность. С 7:00 до 19:00 – 12 ч. В минутах: $12 \cdot 60 = 720$ мин.

Шаг 2. Промежутки. $720 : 15 = 48$ промежутков по 15 мин.

Шаг 3. Отсчёты. Замеры стоят на обоих концах каждого промежутка, как столбы забора: отсчётов на один больше, чем промежутков. $n = 48 + 1 = 49$.

Проверь на маленьком кусочке: с 7:00 до 7:30 замеры в 7:00, 7:15 и 7:30 – три отсчёта на два промежутка.

Шаг 4. Память. $49 \cdot 4 = 196$ байт.

 **Типичная ошибка.** Ответить 48: поделить время на шаг и забыть первый замер в 7:00. Другая – считать часы до 19:00 на пальцах «7, 8, ..., 19» и получить 13 часов вместо 12.

Ответ: а) 49 отсчётов; б) 196 байт.

Своя задача по программе сферы, модуль 3

Реши сам

7.1. Термометр в теплице записывает температуру каждые 20 мин, с 6:00 до 18:00 включительно. Сколько отсчётов получится?

7.2. Память регистратора вмещает 2880 записей. Замеры идут без перерыва. На сколько суток хватит памяти, если мерить раз в 5 мин? А если раз в 30 с?

Задача 7.2 Влажность после полива

скользящее среднее, шум

После полива датчик раз в минуту записывал влажность воздуха в теплице, %:

60, 64, 59, 66, 64, 62, 69, 67, 71.

Влажность то росла, то падала – в сыром ряду она уменьшалась 4 раза. Сгладь ряд скользящим средним по 3 отсчётам. а) Запиши сглаженный ряд. б) Сколько раз в нём значение уменьшалось? в) На сколько выросла влажность от первого сглаженного значения до последнего?

Дано: 9 отсчётов; окно $k = 3$.

Найти: сглаженный ряд; число уменьшений; рост.

Решение

Шаг 1. Сколько будет средних. $n - k + 1 = 9 - 3 + 1 = 7$. Если к концу у тебя получится не 7 чисел, где-то пропущено окно.

Шаг 2. Считаем окна. Каждое следующее окно сдвинуто на один отсчёт:

Таблица 7.3

1-й в окне	2-й	3-й	Сумма	Среднее, %
60	64	59	183	61
64	59	66	189	63
59	66	64	189	63
66	64	62	192	64
64	62	69	195	65
62	69	67	198	66
69	67	71	207	69

Сумму следующего окна можно не складывать заново: вычти ушедшее число и прибавь пришедшее. Из первого окна во второе ушло 60, пришло 66: $183 - 60 + 66 = 189$.

Шаг 3. Уменьшения. Сглаженный ряд: 61, 63, 63, 64, 65, 66, 69. Каждое следующее число больше предыдущего или равно ему – уменьшений ноль. Скачки вниз в сыром ряду были шумом.

Шаг 4. Рост. Процент здесь – единица влажности, как градус у температуры, поэтому просто вычитаем: $69 - 61 = 8\%$. Сырые данные дали бы $71 - 60 = 11\%$ – в них первое и последнее число случайно оказались на «дне» и на «ребре» дрожи.

Рост влажности после полива на несколько процентов за 9 минут – обычная картина: вода испаряется с листьев и почвы.

⚠ Типичная ошибка. Двигать окно не на один, а на три отсчёта: (60, 64, 59), (66, 64, 62), (69, 67, 71). Так получается просто среднее по кусочкам – три числа вместо семи. Для скользящего среднего соседние окна перекрываются на $k - 1$ отсчёт.

Ответ: а) 61, 63, 63, 64, 65, 66, 69; б) ни разу; в) на 8 %.

Своя задача по программе сферы, модуль 3

Реши сам

7.3. Датчик освещённости записал раз в минуту, лк: 1400, 1700, 1100, 1600, 1900, 1500. Посчитай скользящее среднее по 3 отсчётам. Сколько чисел получится? Запиши их, округлив до целых.

7.4. Датчик освещённости записал четыре отсчёта, лк: 410, 430, ?, 450 – третий отсчёт в таблице стёрся. Известно, что первое скользящее среднее по 3 отсчётам равно 420 лк. Найди стёртый отсчёт и второе скользящее среднее. Второе округли до десятых.

Задача 7.3 Вентилятор, который включался зря

выбросы, ложные срабатывания, медианный фильтр, порог

Вентилятор в теплице включается, когда температура 28 °С и выше, и выключается, как только она становится ниже 28 °С. За 14 минут термометр раз в минуту записал, °С:

22, 23, 23, 38, 24, 25, 25, 26, 33, 26, 27, 28, 29, 29.

Выбросом считаем отсчёт, который отличается от **обоих** соседей больше чем на 5 °С. а) Сколько раз включится вентилятор, если программа сравнивает с порогом сырые отсчёты? б) Сколько из этих включений ложные? в) Сколько раз и на какой минуте он включится, если сначала пропустить ряд через медианный фильтр из трёх (первый и последний отсчёт – без изменений)?

Дано: 14 отсчётов раз в минуту; порог 28 °С; выброс – отличие от обоих соседей больше 5 °С.

Найти: число включений; число ложных; включения после медианного фильтра.

Решение

Шаг 1. Включения по сырым данным. Включение – момент, когда отсчёт стал 28 или больше, а предыдущий был меньше 28. Идём по ряду:

- 4-я минута: 38 после 23 – включение; на 5-й минуте 24, вентилятор выключается;
- 9-я минута: 33 после 26 – включение; на 10-й минуте 26, выключается;
- 12-я минута: 28 после 27 – включение; дальше 29, 29 – вентилятор так и работает.

Всего **3** включения.

Шаг 2. Кто выброс. Проверяем подозрительные отсчёты по правилу.

- 38 на 4-й минуте: соседи 23 и 24. $38 - 23 = 15$ и $38 - 24 = 14$ – оба больше 5. Выброс.
- 33 на 9-й минуте: соседи 26 и 26. $33 - 26 = 7$ – больше 5 с обеих сторон. Выброс.
- 28 на 12-й минуте: соседи 27 и 29, разница 1. Честный отсчёт, дальше температура держится.

Значит, ложных включений **2**, на 4-й и 9-й минутах.

Шаг 3. Медианный фильтр. Для каждого внутреннего отсчёта берём его и двух соседей, выстраиваем по порядку и берём среднее по месту. Там, где выбросов нет, медиана почти ничего не меняет. На выбросах:

- 4-я минута: 23, 38, 24 → по порядку 23, 24, 38 → медиана 24;
- 9-я минута: 26, 33, 26 → 26, 26, 33 → медиана 26.

Отфильтрованный ряд: 22, 23, 23, 24, 25, 25, 25, 26, 26, 27, 27, 28, 29, 29. Первый раз 28 °С появляется на 12-й минуте – одно включение.

Проверка кодом.

```

from statistics import median

t = [22, 23, 23, 38, 24, 25, 25, 26, 33, 26, 27, 28, 29, 29]

def vkl_yucheniya(ryad, porog):
    minuty = []
    for i in range(len(ryad)):
        byl_nizhe = (i == 0) or (ryad[i-1] < porog)    # до начала вентилятор выключен
        if ryad[i] >= porog and byl_nizhe:
            minuty.append(i + 1)                      # минуты считаем с 1
    return minuty

f = [t[0]] + [median(t[i-1:i+2]) for i in range(1, len(t) - 1)] + [t[-1]]
print(vkl_yucheniya(t, 28))    # [4, 9, 12]
print(vkl_yucheniya(f, 28))    # [12]

```

За 14 минут воздух прогрелся с 22 до 29 °С, на 7 градусов, – так бывает, когда из-за туч выходит солнце. А вот скачок на 15 градусов за минуту и обратно не объяснит никакое солнце.

❗ Типичная ошибка. Считать включениями все отсчёты не ниже 28: их пять (38, 33, 28, 29, 29). Но вентилятор на 13-й и 14-й минутах не включается заново – он уже работает. Считаем только переходы «ниже порога → порог и выше».

Ответ: а) 3 включения; б) 2 ложных (4-я и 9-я минуты); в) одно включение, на 12-й минуте.

Своя задача по программе сферы, модуль 3

Реши сам

7.5. Насос полива включается, когда влажность почвы ниже 35 %. Раз в минуту датчик записал, %: 48, 47, 0, 46, 45, 44, 3, 43, 41, 40, 38, 36, 34, 33. Почти нулевые значения – это отходящий контакт. Сколько раз насос включался ложно? На какой минуте произошло настоящее включение?

7.6. Чтобы не реагировать на выбросы, программу вентилятора переписали: он включается, только если температура 28 °С и выше **три отсчёта подряд**. Раз в минуту записано, °С: 26, 27, 31, 30, 27, 28, 28, 29, 30. На какой минуте включится вентилятор? На сколько минут позже, чем включился бы по старому правилу «один отсчёт 28 °С и выше»?

Задача 7.4 Сколько оборотов делает вентилятор

детектирование по уровню, период, частота

На лопасть вентилятора наклеили белую метку, а напротив поставили датчик освещённости. Каждый раз, когда метка пролетает мимо, показание подскакивает. Arduino делает отсчёт каждые 0,01 с (рисунок ниже). Порог – 500. Показания:

- отсчёты 1–8: 110, 120, 105, 115, 780, 640, 130, 110;
- отсчёты 9–16: 120, 105, 115, 125, 810, 140, 120, 110;

- отсчёты 17–24: 115, 125, 105, 120, 760, 690, 125, 115;
- отсчёты 25–32: 110, 120, 130, 115, 800, 135, 120, 110.

Сколько оборотов в минуту делает вентилятор?

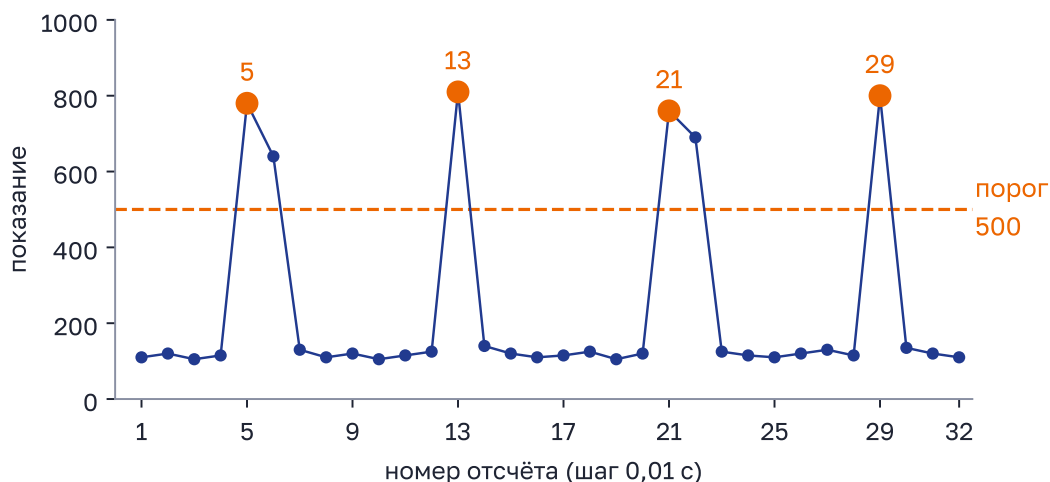


Рисунок 7.15 – Отсчёты датчика у лопасти с меткой. Оранжевые точки – начало каждого прохода метки

Дано: 32 отсчёта; шаг 0,01 с; порог 500.

Найти: обороты в минуту – ?

Решение

Шаг 1. Выше порога. Не ниже 500 отсчёты 5, 6, 13, 21, 22, 29 – шесть штук.

Шаг 2. События. Метка иногда светит в датчик два отсчёта подряд (5–6, 21–22): она пролетает не мгновенно. Считаем только фронты – переходы снизу вверх через порог. Это отсчёты **5, 13, 21, 29**. Проходов метки четыре, а не шесть.

Шаг 3. Период в отсчётах. От первого фронта до последнего $29 - 5 = 24$ отсчёта, промежутков между четырьмя событиями три: $24 : 3 = 8$ отсчётов на один оборот. Проверка по соседям: $13 - 5 = 8$, $21 - 13 = 8$, $29 - 21 = 8$ – вентилятор крутится равномерно.

Шаг 4. Период в секундах. $T = 8 \cdot 0,01 = 0,08$ с.

Шаг 5. Частота. $f = 1 : 0,08 = 12,5$ оборота в секунду (12,5 Гц).

Шаг 6. В минуту. В минуте 60 с: $12,5 \cdot 60 = 750$ оборотов в минуту.

Комнатный вентилятор крутится со скоростью порядка тысячи оборотов в минуту, так что 750 – спокойный, вполне жизненный режим.

⚠ Типичная ошибка. Взять за события все шесть отсчётов выше порога и решить, что за 0,32 с было 6 оборотов. Вторая ловушка – столбы и пролёты: $24 : 4 = 6$ отсчётов на оборот вместо 8, и в ответе выйдет 1000 оборотов в минуту.

Ответ: 750 оборотов в минуту.

Своя задача по программе сферы, модуль 3

Реши сам

7.7. Датчик капель у капельницы отметил первую каплю на 2-й секунде, а одиннадцатую – на 32-й секунде. Капли падают равномерно. Найди период капания в секундах и число капель в минуту.

7.8. Геркон анемометра выдаёт 1, когда магнит на вертушке проходит мимо, и 0 в остальное время. Отсчёт каждые 0,05 с. Записано 30 отсчётов:

0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0.

Сколько оборотов в минуту делает вертушка?

Задача 7.5 Зимняя посылка с метеостанции

двоичный код, сдвиг, число вариантов

Метеостанция в огороде передаёт в дом температуру одним байтом – 8 битами. Чтобы передавать и морозы, к температуре в целых градусах перед отправкой прибавляют 50, и отправляют сумму в двоичном коде. Январским утром приёмник получил байт 00100110. а) Какая температура на улице? б) Какую самую низкую и самую высокую температуру можно передать таким способом?

Дано: 8 битов; код = температура + 50; принято 00100110.

Найти: t – ?; наименьшая и наибольшая температура.

Решение

Шаг 1. Веса разрядов. Подписываем над каждым битом его вес. Начинаем справа: у крайнего правого бита вес 1, дальше 2, 4, 8 и так до 128 у крайнего левого.

вес:	128	64	32	16	8	4	2	1
бит:	0	0	1	0	0	1	1	0

Шаг 2. Складываем веса единиц. Единицы стоят в разрядах 32, 4 и 2: $32 + 4 + 2 = 38$.

Шаг 3. Убираем сдвиг. Код = температура + 50, значит, температура = код – 50:
 $38 - 50 = -12$ °C.

Шаг 4. Границы. В 8 битах $2^8 = 256$ вариантов: коды от 0 (00000000) до 255 (11111111). Самая низкая температура – у кода 0: $0 - 50 = -50$ °C. Самая высокая – у кода 255:
 $255 - 50 = 205$ °C.

Минус 12 для январского утра под Новосибирском – даже мягко. А –50 °C как нижняя граница – с запасом: морозы сильнее – даже в Сибири большая редкость. Верхние 205 °C не нужны вовсе, так что сдвиг можно было взять и побольше.

⚠ Типичная ошибка. Читать веса слева направо: принять самый левый бит за 1, а самый правый за 128. Тогда 00100110 превращается в $4 + 32 + 64 = 100$ – и на улице «+50 °C». Вес 1 всегда у крайнего **правого** бита. Вторая ошибка – забыть про сдвиг и ответить «38 °C».

Ответ: а) –12 °C; б) от –50 °C до +205 °C.

Реши сам

7.9. Переведи в обычные числа двоичные 10101_2 и 1101_2 .

7.10. Метеостанция передаёт влажность воздуха в целых процентах, от 0 до 100. а) Сколько битов нужно на одно число? б) Сколько битов понадобится, если передавать влажность с шагом 0,5 % (0; 0,5; 1; ... 100)? в) За сколько секунд уйдут 24 числа из пункта а) на скорости 12 бит/с?

Задача 7.6^{*}. Отчёт регистратора

выброс, медиана, среднее, время по отсчётам

Регистратор в теплице записывал температуру каждые 30 мин, с 8:00 до 15:30. Каждый отсчёт считаем за полчаса. Один отсчёт испорчен: на датчик упал прямой солнечный луч. Испорченным считаем отсчёт, который отличается от обоих соседей больше чем на 10 °С, и заменяем его медианой из него и двух соседей. Найди: а) среднюю температуру за это время, округли до десятых; б) сколько часов температура была 25 °С и выше.

Таблица 7.4

Время	°С	Время	°С
8:00	18	12:00	28
8:30	19	12:30	27
9:00	85	13:00	26
9:30	22	13:30	26
10:00	24	14:00	25
10:30	25	14:30	24
11:00	26	15:00	23
11:30	27	15:30	21

Дано: 16 отсчётов через 30 мин; выброс – отличие от обоих соседей больше 10 °С; замена медианой из трёх.

Найти: среднее – ?; время при 25 °С и выше – ?

Решение

Шаг 1. Сколько отсчётов. С 8:00 до 15:30 – 7,5 ч, это 15 промежутков по 30 мин, значит, отсчётов $15 + 1 = 16$. Каждый отвечает за полчаса, всего $16 \cdot 0,5 = 8$ ч наблюдений.

Шаг 2. Ищем выброс. 85 °С в 9:00, соседи 19 и 22: $85 - 19 = 66$ и $85 - 22 = 63$, оба больше 10. Остальные отсчёты отличаются от соседей на 1–3 градуса. Выброс один.

Шаг 3. Заменяем. Три числа: 19, 85, 22 → по порядку 19, 22, 85 → медиана 22. В 9:00 ставим 22 °C.

Шаг 4. Сумма. Сумма всех 16 сырых отсчётов – 446. Выброс меняем на 22:

$446 - 85 + 22 = 383$. Вычитать и прибавлять проще, чем складывать все 16 чисел заново.

Но если складываешь заново, сложи по столбцам таблицы: левый

$18 + 19 + 22 + 22 + 24 + 25 + 26 + 27 = 183$, правый

$28 + 27 + 26 + 26 + 25 + 24 + 23 + 21 = 200$, вместе 383.

Шаг 5. Среднее. $383 : 16 = 23,9375 \approx 23,9$ °C.

Шаг 6. Часы при 25 °C и выше. Такие отсчёты: 10:30, 11:00, 11:30, 12:00, 12:30, 13:00, 13:30, 14:00 – восемь штук. В 9:00 теперь 22 °C, туда не входит. Каждый отсчёт – полчаса: $8 \cdot 0,5 = 4$ ч.

Проверка кодом.

```
from statistics import median

t = [18, 19, 85, 22, 24, 25, 26, 27, 28, 27, 26, 26, 25, 24, 23, 21]
for i in range(1, len(t) - 1):
    if abs(t[i] - t[i-1]) > 10 and abs(t[i] - t[i+1]) > 10: # выброс
        t[i] = median(t[i-1:i+2])
print(t[2], round(sum(t) / len(t), 1)) # 22 23.9
print(sum(1 for x in t if x >= 25) * 0.5) # 4.0
```

Среднее вышло между самым холодным (18 °C) и самым тёплым (28 °C) отсчётом, ближе к тёплым: теплица полдня простояла на солнце.

⚠ Типичная ошибка. Посчитать среднее вместе с выбросом: $446 : 16 = 27,9$ °C – на четыре градуса выше правды, и заодно насчитать 9 «жарких» отсчётов и 4,5 ч. Вторая ловушка – посчитать время «от 10:30 до 14:00» и получить 3,5 ч: так теряются последние полчаса, за которые отвечает отсчёт 14:00. По условию каждый отсчёт – полчаса, значит, число отсчётов умножаем на 0,5.

Ответ: а) 23,9 °C; б) 4 ч.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «Анализ температурных данных»

Реши сам

7.11. Лампы досветки горят, когда освещённость ниже 2500 лк. Датчик мерил каждые 15 мин с 16:00 до 19:45, каждый отсчёт считаем за 15 мин. Отсчёты, лк: 3500, 3200, 2600, 2100, 1800, 1500, 1900, 2400, 2800, 3100, 2900, 2200, 1700, 1200, 800, 400. Сколько часов горели лампы?

7.12*.* Средняя температура по 12 отсчётам оказалась 23,5 °C. Потом выяснилось, что один из отсчётов – 83 °C – выброс, а его соседи – 24 °C и 26 °C. Выброс заменили медианой из него и двух соседей. Какой стала средняя температура? Ответ до сотых.

Задача 7.7*. Лампа мерцает

спектр, период помехи, окно фильтра, скользящее среднее

Датчик освещённости стоит под лампой досветки. Яркость лампы пульсирует 100 раз в секунду – так мигают многие лампы, работающие от сети. Глазом этого не видно, а датчик видит. Отсчёт делается каждые 2,5 мс. Заодно в теплице светает, поэтому освещённость понемногу растёт. Первые 12 отсчётов:

600, 644, 608, 572, 616, 660, 624, 588, 632, 676, 640, 604.

а) Сколько отсчётов укладывается в один период мерцания? б) Каким наименьшим окном скользящего среднего мерцание убирается полностью? в) Чему равно первое сглаженное значение при таком окне?

Дано: мерцание 100 Гц; шаг 2,5 мс; 12 отсчётов.

Найти: отсчётов на период; наименьшее окно; первое сглаженное значение.

Решение

Шаг 1. Период мерцания. $T = 1:100 = 0,01$ с. В миллисекундах: $0,01 \cdot 1000 = 10$ мс.

Шаг 2. Отсчётов на период. $10:2,5 = 4$ отсчёта. Проверим по ряду: 600, 644, 608, 572 – подъём, спад, провал; потом 616, 660, 624, 588 – тот же узор, только всё выше на 16. Узор повторяется каждые 4 отсчёта.

Шаг 3. Почему окно должно быть длиной в период. Разложим отсчёты на два слагаемых: медленный рост (600, 604, 608, 612, ... – плюс 4 за отсчёт) и мерцание, которое повторяется по кругу: 0, +40, 0, -40. Проверь: $604 + 40 = 644$, $612 - 40 = 572$. В окно из 4 соседних отсчётов попадает весь круг мерцания, каждая его часть по одному разу: $0 + 40 + 0 - 40 = 0$. Мерцание в сумме пропадает, остаётся только медленный рост.

Шаг 4. Проверим окно поменьше. Окно 3 по первым отсчётам:

$(600 + 644 + 608):3 = 617,3$; дальше 608; 598,7; 616; 633,3. Сглаженный ряд всё ещё качается вверх-вниз – мерцание осталось, только слабее. Окно 2 тоже не помогает: 622, 626, 590, 594, ... Значит, наименьшее окно – 4.

Шаг 5. Первое сглаженное значение. $(600 + 644 + 608 + 572):4 = 2424:4 = 606$. Следующие: 610, 614, 618 – растут на 4 за отсчёт, как и медленная часть.

Проверка кодом.

```
x = [600, 644, 608, 572, 616, 660, 624, 588, 632, 676, 640, 604]
for k in (2, 3, 4):
    sr = [sum(x[i:i+k]) / k for i in range(len(x) - k + 1)]
    print(k, [round(v, 1) for v in sr[:5]])
# 2 [622.0, 626.0, 590.0, 594.0, 638.0]
# 3 [617.3, 608.0, 598.7, 616.0, 633.3]
# 4 [606.0, 610.0, 614.0, 618.0, 622.0]
```

Окно 4 – это всего 10 мс. Для теплицы, где свет меняется за минуты, такое опоздание незаметно.

 **Типичная ошибка.** Решить, что «чем больше окно, тем лучше», и взять, например, 5. Окно 5 тоже сгладит ряд, но неровно: в каждое окно одна часть мерцания попадёт дважды, и ряд продолжит слегка качаться. Мерцание пропадает полностью, когда окно равно периоду или целому числу периодов: 4, 8, 12 отсчётов. Ещё ошибаются в шаге 2 – делят 10 мс на 2,5 с и получают 0,004 отсчёта.

Ответ: а) 4 отсчёта; б) окно 4; в) 606.

Своя задача по программе сферы, модуль 3

Реши сам

7.13. Температура в теплице сложена из трёх колебаний: суточного (период 24 ч, амплитуда 7 °C), от форточки (период 20 мин, амплитуда 0,4 °C) и дрожи датчика (период 10 с, амплитуда 0,2 °C). а) Сколько повторов в час делает каждое колебание? б) Фильтр низких частот пропускает только колебания, которые повторяются реже одного раза в час. Какой размах будет у сигнала после фильтра?

7.14*.* Помеха в показаниях датчика повторяется каждые 6 отсчётов по кругу: 0, +30, +30, 0, -30, -30. Первые 12 отсчётов: 500, 533, 536, 509, 482, 485, 518, 551, 554, 527, 500, 503. Каким наименьшим окном скользящего среднего помеха убирается полностью? Чему равно первое сглаженное значение при этом окне?

Метеостанция и автоматика



Теплица стоит на улице, и погода за стенкой решает за неё очень многое: ветер может сорвать открытую форточку, ночной заморозок – погубить рассаду, солнце – дать бесплатную энергию. В этой главе ставим рядом с теплицей метеостанцию, учимся переводить обороты анемометра в скорость ветра, поворачивать форточку сервомотором, крутить вентилятор DC-мотором и ловить солнце панелью. На отборе такие задачи приходят короткой историей с числами или кусочком программы, а ответ – одно число или несколько чисел по порядку.

8.1. Метеостанция: теплица смотрит на улицу

НА ПАЛЬЦАХ

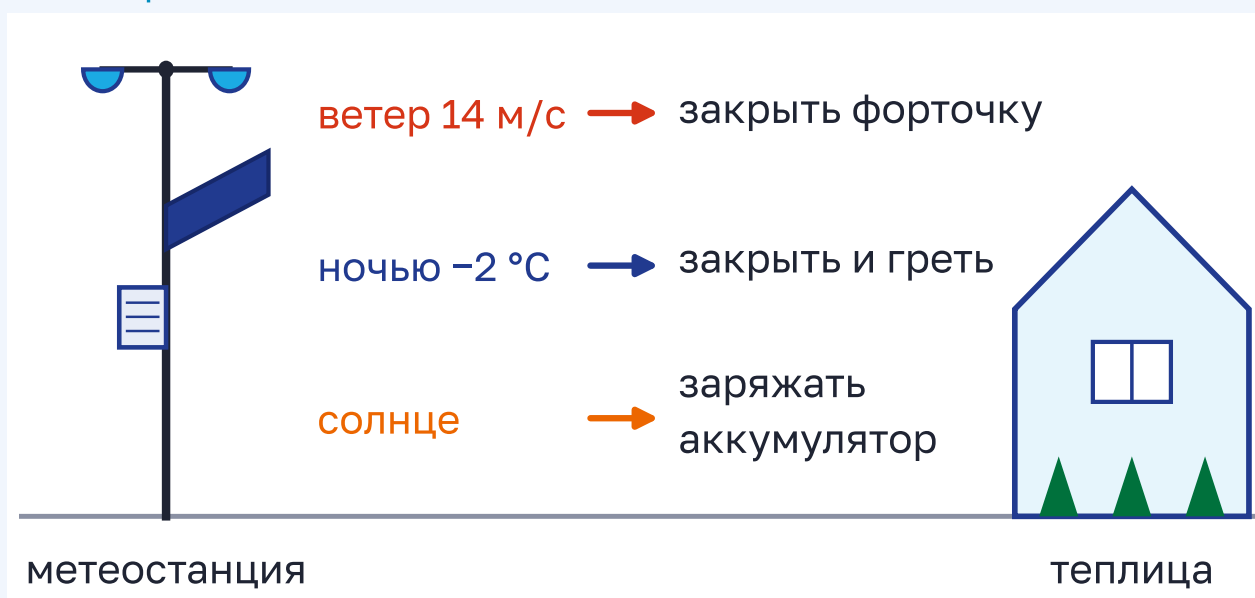


Рисунок 8.1 – Метеостанция на шесте рядом с теплицей и её подсказки автоматике

Внутри теплицы датчики видят только то, что уже случилось. Метеостанция снаружи видит, что сейчас придёт: ветер, холод, солнце.

Метеостанция – набор датчиков погоды на одном шесте или мачте. Школьная метеостанция из набора меряет температуру и влажность воздуха, скорость ветра и освещённость, а питается от солнечной панели. Всё это читает та же плата Arduino, что управляет теплицей, или отдельная плата, которая передаёт числа по проводу или по радио.

Зачем теплице знать погоду снаружи? Датчик внутри заметит холод, только когда холод уже зашёл внутрь. Метеостанция предупреждает заранее. Пример из новосибирской весны: в мае днём $+20^{\circ}\text{C}$, а к ночи на улице падает до -2°C . Если вечером станция видит, что снаружи быстро холодает, автоматика закроет форточки и включит обогрев, пока рассада ещё в тепле. Второй пример – ветер. Открытую форточку сильный порыв может сломать, поэтому при сильном ветре её закрывают, даже если внутри жарко, а проветривают вентилятором.

Датчик DHT11 – маленькая голубая коробочка с решёткой, внутри два датчика сразу: температуры и влажности воздуха. Термометр TMP36 из главы 5 выдаёт напряжение, которое ещё надо пересчитывать в градусы. DHT11 присылает готовые числа – цифровым сигналом по одному проводу. Проводов у него три: питание, земля и сигнал. В симуляторе Tinkercad такого датчика нет, поэтому код с ним мы читаем, а не собираем. Считывать его помогает библиотека – готовый кусок программы, который кто-то уже написал. В Arduino IDE её ставят через «Управление библиотеками», она называется «DHT sensor library».

```

#include <DHT.h>           // библиотека для датчиков DHT
DHT dht(2, DHT11);        // датчик на выводе 2, модель DHT11

void setup() {
  Serial.begin(9600);
  dht.begin();             // запускаем датчик
}

void loop() {
  float t = dht.readTemperature(); // температура, °C
  float h = dht.readHumidity();     // влажность воздуха, %
  Serial.print(t);
  Serial.print(" C ");
  Serial.println(h);
  delay(2000);                  // DHT11 не отвечает чаще раза в 1-2 секунды
}

```

Первая строка подключает библиотеку. Вторая говорит, что датчик висит на выводе 2 и что это модель DHT11. Дальше всё знакомо: `dht.readTemperature()` возвращает температуру, `dht.readHumidity()` – влажность, и мы печатаем их в монитор порта. Пауза в 2 секунды нужна самому датчику: спросишь его чаще – он будет отдавать старое число или ошибку.

У DHT11 есть старший брат DHT22 – он точнее и не боится мороза. В симуляторе Wokwi есть как раз DHT22, а работа с ним в программе та же, меняется только слово в скобках: `DHT dht(2, DHT22);`. Чем они отличаются (по паспортам производителя):

Таблица 8.1

	DHT11	DHT22
Температура	от 0 до 50 °C	от -40 до 80 °C
Точность температуры	±2 °C	±0,5 °C
Влажность	от 20 до 90 %	от 0 до 100 %
Точность влажности	±5 %	±2 %

Точность ±2 °C значит, что настоящая температура может отличаться от показаний на 2 градуса в любую сторону. DHT11 показал 25 °C – а в теплице может быть и 23, и 27. Для теплицы терпимо, для медицинского градусника – нет. А **диапазон** важнее точности: DHT11 рассчитан на температуру от нуля и выше. На улице в новосибирском январе, когда -30 °C, он будет показывать ерунду. Для уличной станции у нас нужен DHT22 или другой датчик, который выдерживает мороз.

Запомни. Метеостанция нужна теплице, чтобы автоматика успела среагировать до того, как погода придёт внутрь. DHT11 меряет температуру (0...50 °C, ±2 °C) и влажность воздуха (20...90 %) и отдаёт готовые числа; спрашивать его – не чаще раза в 1-2 секунды.

Совет. Выбираешь датчик по таблице характеристик – первым делом проверь диапазон: влезает ли в него самая холодная и самая жаркая погода, которая может случиться. Точность смотри уже потом.

8.2. Анемометр: из оборотов в скорость ветра

НА ПАЛЬЦАХ

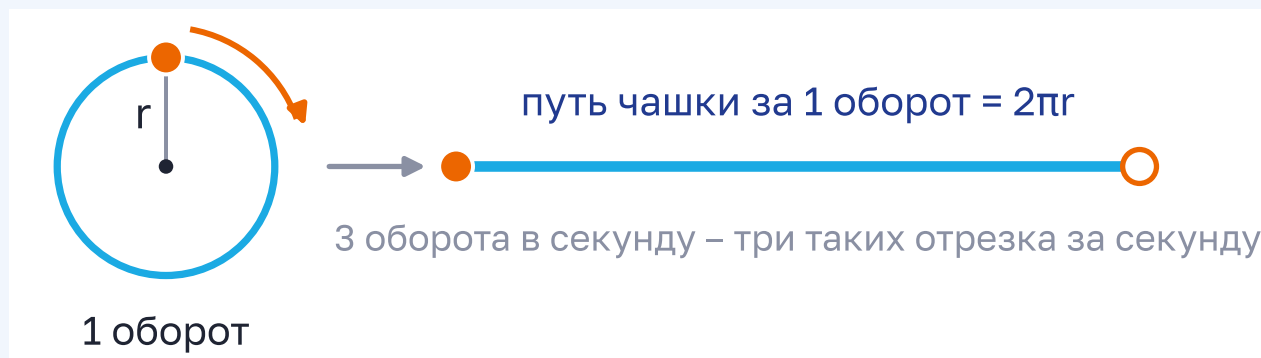


Рисунок 8.2 – За один оборот чашка проходит круг. Разверни круг в прямую – получишь её путь

Посчитай, сколько кругов чашка сделала за секунду, умножь на длину круга – получишь, сколько метров она пролетела за секунду. Это и есть её скорость.

Анемометр – вертушка с чашечками на оси. Ветер давит в вогнутую сторону чашки сильнее, чем в выпуклую, и вертушка крутится: чем сильнее ветер, тем быстрее. Скорость ветра измеряют в **метрах в секунду** (м/с): сколько метров воздух пролетает за одну секунду. Прогноз погоды тоже говорит «ветер 5 м/с».

Скорость – путь, делённый на время. Велосипедист проехал 100 м за 20 с – его скорость $100:20 = 5$ м/с.

За один оборот центр чашки проходит круг. Длина этого круга (**длина окружности**) равна $2\pi r$, где r – **радиус**: расстояние от оси вертушки до центра чашки, а $\pi \approx 3,14$. Если вертушка делает n оборотов за секунду, то за секунду чашка проходит n кругов. Словами: **скорость чашки равна длине круга, умноженной на число оборотов в секунду**. Буквами:

$$v_{\text{ч}} = 2\pi r \cdot n,$$

где $v_{\text{ч}}$ – скорость чашки в м/с, r – радиус в метрах, n – оборотов в секунду (об/с).

Пример. Радиус 10 см = 0,1 м, вертушка делает 2 оборота за секунду. Длина круга $2 \cdot 3,14 \cdot 0,1 = 0,628$ м, скорость чашки $0,628 \cdot 2 = 1,256$ м/с, около 1,3 м/с.

Поправка. Чашки не успевают за ветром: они крутятся медленнее, чем летит воздух. Поэтому у настоящего анемометра есть **поправочный коэффициент** k – число, во сколько раз ветер быстрее чашек. Его находят на заводе, продувая прибор в трубе с известным ветром. У самодельных вертушек он бывает около 2–3. Если k дан в условии, скорость ветра

$$v = k \cdot v_{\text{ч}}.$$

Если про поправку в задаче ни слова – считай $v = v_{\text{ч}}$.

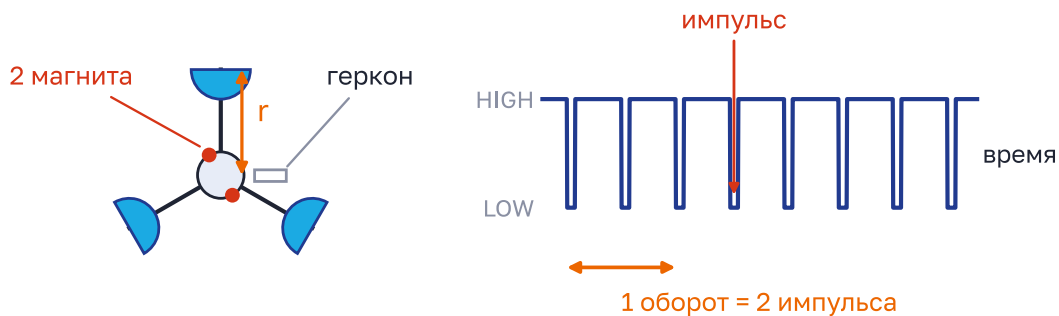


Рисунок 8.3 – Слева – анемометр сверху: на оси два магнита, рядом неподвижный геркон. Справа – что видит плата: каждый раз, когда магнит проходит мимо, геркон замыкается, и на выводе короткий провал – импульс

Как плата считает обороты. На вертушке закрепляют маленький магнит, а рядом – неподвижный **геркон**: стеклянную трубочку с двумя пластинками, которые смыкаются, когда рядом магнит. Магнит пролетел мимо – контакт на миг замкнулся, плата увидела **импульс**. Считать импульсы мы учились в главе 7: ищем моменты, когда сигнал перешёл из одного состояния в другое. Иногда магнитов ставят два или четыре, чтобы на медленном ветру импульсов было больше. Тогда **число оборотов = число импульсов : число магнитов**.

Вот программа, которая 5 секунд считает импульсы и печатает скорость ветра. Геркон подключён к выводу 2: пока магнита нет, на выводе HIGH, когда магнит рядом – LOW.

```
const int GERKON = 2;           // геркон анемометра – на выводе 2
const float R = 0.07;          // радиус вертушки, м
const float K = 3.0;           // поправка: ветер в 3 раза быстрее чашек
const int MAGNITOV = 2;        // сколько импульсов на один оборот

void setup() {
  pinMode(GERKON, INPUT_PULLUP); // без магнита на выводе будет HIGH
  Serial.begin(9600);
}

void loop() {
  int impulsy = 0;
  int bylo = digitalRead(GERKON);
  unsigned long start = millis();
  while (millis() - start < 5000) { // считаем 5 секунд
    int stalo = digitalRead(GERKON);
    if (bylo == HIGH && stalo == LOW) impulsy++; // магнит подошёл - +1
    bylo = stalo;
  }
  float n = impulsy / (float)MAGNITOV / 5.0; // оборотов в секунду
  float v = 2 * 3.14 * R * n * K;             // скорость ветра, м/с
  Serial.println(v);
}
```

Сердце программы – цикл `while`: он крутится, пока от старта прошло меньше 5000 мс. На каждом круге он читает геркон и сравнивает с прошлым чтением: было HIGH, стало LOW – значит, магнит только что подошёл, и счётчик `impulsy` растёт на 1. Если просто считать все LOW, один проход магнита дал бы сотни «импульсов»: пока он рядом, плата успеет прочитать геркон много раз. `(float)` перед `MAGNITOV` заставляет Arduino делить с дробной частью, иначе `15:2` дало бы 7, а не 7,5.

Километры в час. Скорость машины и ветра в новостях часто дают в км/ч. В часе 3600 с, в километре 1000 м, поэтому $1 \text{ м/с} = 3600:1000 = 3,6 \text{ км/ч}$. Из м/с в км/ч – умножаем на 3,6, обратно – делим на 3,6. Ветер 10 м/с – это 36 км/ч, как школьный автобус по городу.

Запомни. Скорость чашки: $v_{\text{ч}} = 2\pi r \cdot n$ (радиус в метрах, n – оборотов в секунду). Скорость ветра – это скорость чашки, умноженная на поправку k , если она дана. Обороты = импульсы : число магнитов. $1 \text{ м/с} = 3,6 \text{ км/ч}$.

Совет. Выпиши отдельными строчками: сколько импульсов, за сколько секунд, сколько магнитов. Потом получи обороты в секунду – и только после этого бери формулу. Саму формулу путают редко, а вот импульсы с оборотами – постоянно.

8.3. Сервомотор и DC-мотор: форточка и вентилятор

НА ПАЛЬЦАХ



Рисунок 8.4 – Сервомотор встаёт на заданный угол и держит его. DC-мотор крутится, пока через него идёт ток

Сервомотору говорят «куда»: встань на 60°. DC-мотору говорят «как сильно»: крутись на три четверти мощности.

В теплице двигаются две вещи: форточка и вентилятор. И моторы у них разные.

Сервомотор (коротко «серва») – мотор с шестерёнками и собственным «мозгом» внутри. Ему сообщают угол от 0 до 180°, и он поворачивает рычаг на этот угол, а потом держит его, даже если форточку качает ветер. Похоже на стрелку на циферблате: скажешь «на 60» – встанет на 60. Серва из школьного набора – SG90 весом 9 г – поворачивается на 60° за 0,1 с (по паспорту, при 4,8 В). У сервы три провода: питание, земля и сигнальный – его подключают к выводу платы.

```

#include <Servo.h>           // библиотека для сервомоторов
Servo okno;                  // сервомотор форточки

void setup() {
    okno.attach(6);          // сигнальный провод – на вывод 6
    okno.write(0);           // форточка закрыта
}

void loop() {
    for (int a = 0; a ≤ 90; a += 3) { // открываем плавно: шаг 3°
        okno.write(a);
        delay(40);
    }
    delay(5000);
    okno.write(0);           // закрыть сразу
    delay(5000);
}

```

Сама серва поворачивается быстро. Но распахнуть форточку за 0,15 с – значит хлопнуть рамой. Поэтому её открывают плавно: цикл `for` ставит серву на 0°, 3°, 6° ... и после каждого шага ждёт 40 мс. Читается он так: «начни с `a = 0`; повторяй, пока `a ≤ 90`; после каждого круга прибавляй к `a` три». Сколько раз выполнится тело цикла и сколько это займёт времени, посчитаем в задаче 8.3. Подсказка из главы 7: столбов на один больше, чем пролётов.

Угол от ручки. На занятиях серву часто крутят потенциометром – ручкой, как у громкости. Потенциометр на входе A0 даёт число от 0 до 1023 (`analogRead`), а серве нужен угол от 0 до 180. Пересчитывает команда `map`:

```

int a = map(analogRead(A0), 0, 1023, 0, 180); // число 0...1023 → угол 0...180
okno.write(a);

```

Внутри `map` та же пропорция, что и в главе 5: $\text{угол} = \text{число} \cdot 180 : 1023$. Но `map` считает в целых числах и **дробную часть отбрасывает**, не округляя. Ручка дала 700: $700 \cdot 180 : 1023 = 123,17$, серва встанет на 123°. Дала 511: $511 \cdot 180 : 1023 = 89,9$ – это уже 89°, а не 90°.

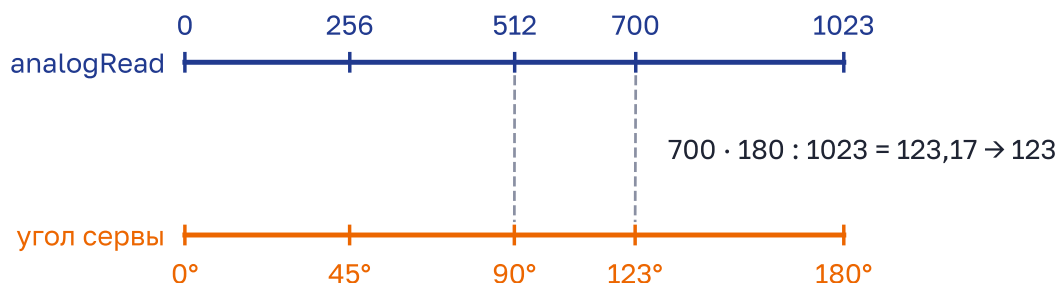


Рисунок 8.5 – Как `map` раскладывает числа от 0 до 1023 на углы от 0 до 180°

DC-мотор (мотор постоянного тока) устроен проще: подал напряжение – крутится, снял – остановился. Поменяй местами провода «плюс» и «минус» – закрутится в другую сторону. Чем больше напряжение, тем быстрее он крутится. На таком моторе держатся вентилятор и насос полива.

Напрямую к выводу Arduino мотор подключать нельзя. Вывод рассчитан на ток около 20 мА (в паспорте предел – около 40 мА, но на постоянной работе держат не больше 20) – хватит на светодиод, а моторчику вентилятора нужно в десятки раз больше: плата перегреется и может сгореть. Между платой и мотором ставят **драйвер** – «усилитель»: плата подаёт ему слабый сигнал «крутить», а он пропускает к мотору сильный ток от отдельного блока питания. Драйвер умеет и менять направление вращения.

НА ПАЛЬЦАХ



Рисунок 8.6 – Три команды `analogWrite` : вывод включён всё время, половину времени и четверть времени

Вывод щёлкает «вкл – выкл» сотни раз в секунду. Мотор не успевает заметить щелчки и крутится так, будто получил только ту долю мощности, какую долю времени вывод был включён.

Скорость задают **ШИМ** (широтно-импульсной модуляцией) – командой `analogWrite(вывод, число)` с числом от 0 до 255. Число – это доля времени «включено»: 255 из 255 – вывод включён всегда, 128 из 255 – примерно половину времени. Подробнее – в справочнике в конце книги. В задачах считают упрощённо: скорость вентилятора и количество воздуха, которое он прогоняет, **пропорциональны** числу ШИМ. В жизни это не совсем так: на маленьком числе мотор может вообще не стронуться с места.

Сколько воздуха. У вентилятора в паспорте пишут **производительность** – сколько кубометров воздуха он прогоняет за час ($\text{м}^3/\text{ч}$). Если объём теплицы 60 м^3 , а вентилятор даёт $300 \text{ м}^3/\text{ч}$, он сменит весь воздух за $60 : 300 = 0,2 \text{ ч}$ – это 12 минут. Тепличники говорят: «пять обменов в час».

Запомни. Серва: угол от 0 до 180° , `write(угол)`, держит положение. DC-мотор: крутится от напряжения, скорость – через ШИМ (0...255), направление – сменой полярности, подключается только через драйвер. `map` отбрасывает дробную часть. Время смены воздуха = объём теплицы : производительность вентилятора.

Совет. Процент мощности из числа ШИМ: число : $255 \cdot 100$. Число ШИМ из процентов: процент : $100 \cdot 255$. Держи под рукой: 255 – 100 %, 191 – около 75 %, 128 – около 50 %, 64 – около 25 %.

8.4. Солнце, панель и трекер

НА ПАЛЬЦАХ

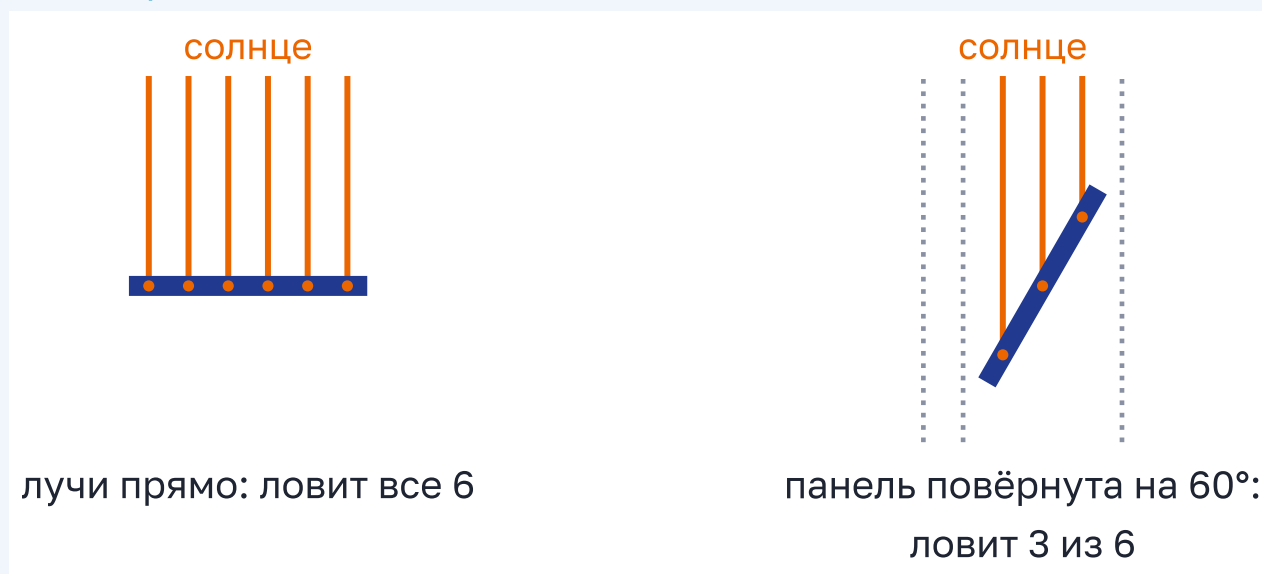


Рисунок 8.7 – Одни и те же солнечные лучи падают на панель, повернутую к солнцу, и на панель, повернутую боком

Повернутая к солнцу панель ловит все лучи, которые на неё летят. Наклонённая – только часть: остальные проходят мимо.

Как солнечная панель превращает свет в электричество, мы считали в главе 5: в ясный полдень на квадратный метр, **повёрнутый к солнцу**, приходит около 1000 Вт света, и панель забирает из этого 15–20 %. Здесь важны слова «повёрнутый к солнцу». Попробуй вечером посветить фонариком на лист бумаги: прямо – яркое круглое пятно, наискосок – пятно растягивается в длинный овал и тускнеет. Света столько же, а размазан он по большей площади, и каждому кусочку листа достаётся меньше.

С панелью так же. Угол между лучами и **перпендикуляром** к панели – прямой, которая торчит из панели под углом 90°, как мачта из палубы, – решает, какая доля света ей достанется:

Таблица 8.2

Угол между лучами и перпендикуляром	Доля света
0°	100 %
15°	97 %
30°	87 %
45°	71 %
60°	50 %
75°	26 %
90°	0 %

Пока угол небольшой, потери маленькие: на 15° теряется всего 3 %. Потом доля падает всё быстрее, на 60° остаётся половина, а если лучи скользят вдоль панели (90°), света ей не достаётся совсем.

Для любопытных. Числа в таблице – это косинус угла, умноженный на 100 %. Косинус проходят в 8–9 классе, а на инженерном калькуляторе в телефоне это кнопка $\cos$: $\cos 60^\circ = 0,5$ – та самая половина.

Высота солнца. Угол между горизонтом и направлением на солнце называют **высотой солнца**. В полдень солнце выше всего за день. Насколько высоко – зависит от того, как далеко город от экватора (эту величину называют **широтой**: у экватора 0° , у полюса 90°) и от времени года. Правило такое:

- в дни равноденствия (около 20 марта и 23 сентября) высота полуденного солнца $= 90^\circ - \text{широта}$;
- 22 июня, в самый длинный день, солнце в полдень выше на 23° ;
- 22 декабря, в самый короткий, – ниже на 23° .

Новосибирск стоит на широте около 55° . Значит, в полдень в равноденствие солнце стоит на $90^\circ - 55^\circ = 35^\circ$ над горизонтом, летом – на $35^\circ + 23^\circ = 58^\circ$, зимой – всего на $35^\circ - 23^\circ = 12^\circ$. Поэтому зимой в полдень тени такие длинные.

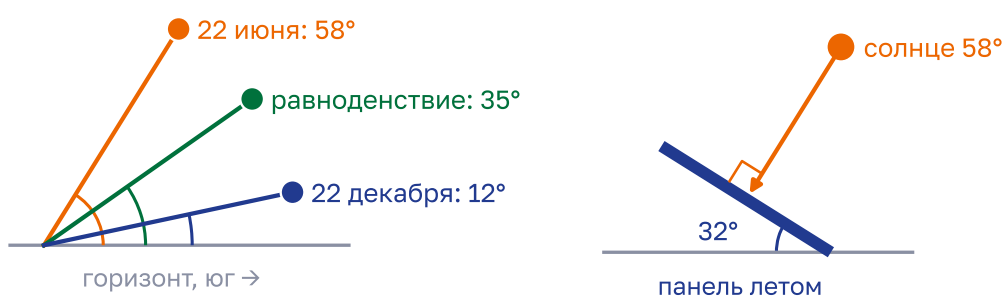


Рисунок 8.8 – Высота полуденного солнца в Новосибирске в разные дни года и наклон панели, при котором лучи падают на неё перпендикулярно

Какой угол выбрать. Чтобы лучи падали на панель перпендикулярно, панель наклоняют к горизонту на угол $90^\circ - \text{высота солнца}$. Солнце на 35° – наклон 55° , солнце высоко летом, на 58° , – наклон всего 32° : чем выше солнце, тем более плоско лежит панель. Зимой, когда солнце на 12° , панель ставят почти стоймя, на 78° . Кстати, снег с такой панели сползает сам.

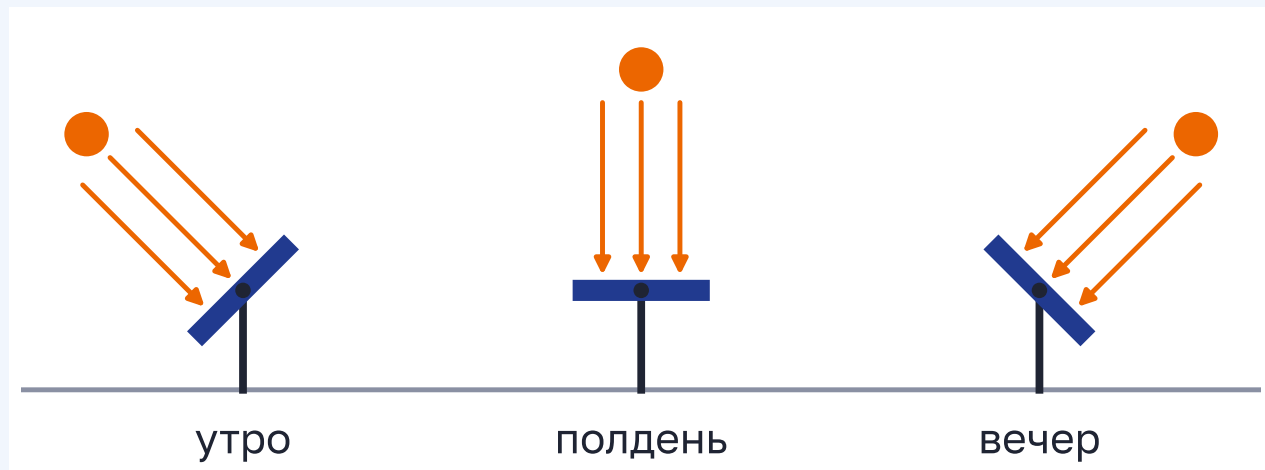


Рисунок 8.9 – Утром, в полдень и вечером трекер держит панель лицом к солнцу

Трекер ведёт себя как молодой подсолнух: утром смотрит на восток, в полдень – на юг, вечером – на запад. Лучи весь день падают на панель прямо.

Трекер. Солнце за сутки обходит полный круг – 360° , значит, за час оно смещается по небу на $360:24 = 15^\circ$. Неподвижная панель смотрит туда, где солнце в полдень, и утром и вечером лучи падают на неё наискосок. **Солнечный трекер** (от английского *track* – «следить») – механизм, который поворачивает панель вслед за солнцем, чаще всего сервомотором. Как он узнаёт, куда поворачивать? На панели ставят два фоторезистора, разделённых перегородкой: один смотрит чуть левее, другой – чуть правее. Левый видит больше света – повернуть на градус влево, правый – вправо, поровну – стоять. В коде ниже фоторезисторы включены так, что больше света – больше число. Если собрать их как в курсе (фоторезистор между A0 и GND), числа будут расти к темноте, и знаки в условиях придётся поменять.

```
// кусок из loop(); в начале программы объявлены Servo panel; и int povorot = 90;
int levo = analogRead(A0);           // фоторезистор слева
int pravo = analogRead(A1);          // фоторезистор справа
if (levo - pravo > 30 && povorot < 180) povorot = povorot + 1; // слева светлее
if (pravo - levo > 30 && povorot > 0) povorot = povorot - 1;  // справа светлее
panel.write(povorot);
```

Это кусок из `loop()`, а `panel` и `povorot` заведены в самом начале программы, как `окно` в примере с форточкой, – об этом комментарий в первой строке. В `povorot` хранится угол сервы трекера. Поворачиваем, только если разница больше 30. Сравнивай мы с нулём, панель дёргалась бы туда-сюда от любой мелкой дрожи показаний (помнишь шум из главы 7?).

Сколько даёт трекер. Возьмём упрощённую модель: солнце светит с 6:00 до 18:00 одинаково ярко, смещается на 15° в час, неподвижная панель смотрит на полуденное солнце. Разобьём день на куски по 2 часа и для каждого возьмём угол в середине куска: в 7:00 солнце ушло от полуденного направления на $5 \cdot 15^\circ = 75^\circ$, в 9:00 – на 45° , в 11:00 – на 15° . После полудня всё повторяется в обратном порядке.

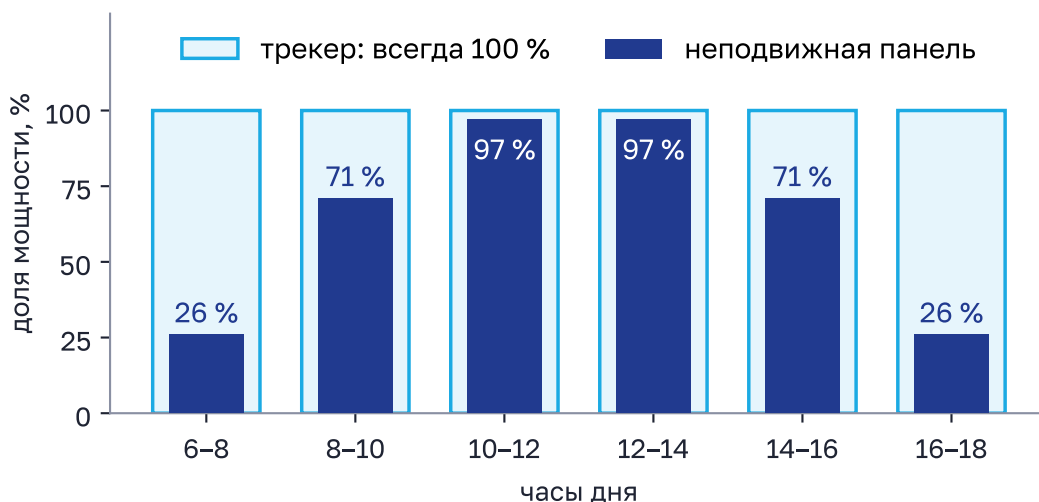


Рисунок 8.10 – Доля полной мощности у неподвижной панели и у трекера по двухчасовым кускам дня

У трекера все 12 часов – 100 %. Неподвижная панель за день набирает столько же энергии, сколько за $2 \cdot (26 + 71 + 97 + 97 + 71 + 26) : 100 = 7,76$ ч полной мощности. Трекер – 12 ч, в полтора с лишним раза больше. В жизни выигрыш скромнее: утром и вечером солнечный свет идёт к земле сквозь толстый слой воздуха и сам по себе слабее полуденного, так что те- рять неподвижной панели в эти часы почти нечего. К тому же серва трекера сама расходует энергию – это посчитаем в задаче 8.6.

Запомни. Панель берёт больше всего света, когда лучи падают на неё перпендикулярно; при отклонении на 60° – только половину. Высота солнца в полдень в равноденствие = 90° – широта, летом на 23° выше, зимой на 23° ниже. Наклон панели = 90° – высота солнца. Солнце смещается на 15° в час.

8.5. Метеостанция управляет теплицей

НА ПАЛЬЦАХ

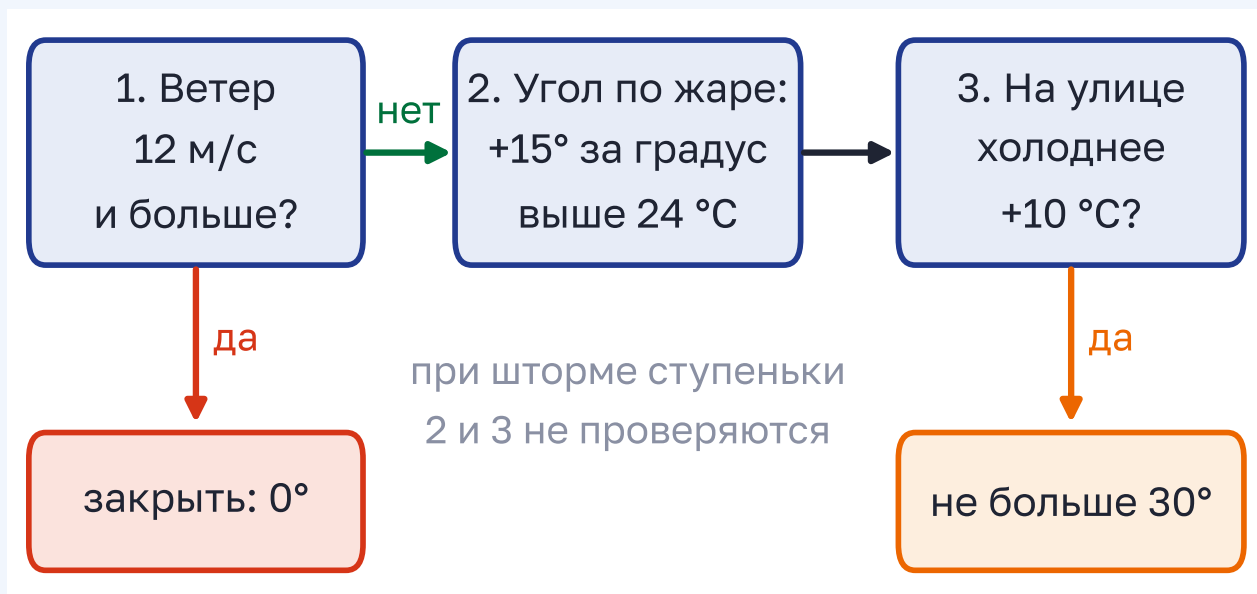


Рисунок 8.11 – Решение проходит три ступеньки по очереди: ветер, жара внутри, холод снаружи

Самое опасное проверяют первым. Если ветер штормовой, до вопроса «жарко ли внутри» дело даже не доходит.

Датчики есть, моторы есть, погоду мы видим. Осталось научить автоматику решать. Она получает числа от датчиков внутри (DHT11 под крышей) и от метеостанции снаружи (DHT22 и анемометр) и решает, на какой угол открыть форточку. Правила записывают по порядку, и программа проверяет их сверху вниз. Так работает цепочка `if ... else if` из главы 6: срабатывает первая подходящая ветка, остальные уже не проверяются.

Порядок правил – не мелочь. Возьмём два правила: «жарко внутри – открыть форточку» и «сильный ветер – закрыть форточку». Жаркий ветреный день: если первым стоит правило про жару, форточка распахнётся, и ветер её сорвёт. Поэтому первыми всегда идут правила безопасности – они сильнее остальных. Такое правило называют **приоритетным**.

Бывают правила помягче – **ограничения**. «На улице холоднее +10 °C – открывать не больше чем на 30°»: форточку открыть можно, но только приоткрыть: ледяной воздух не должен падать прямо на рассаду. В программе это выглядит так: сначала считаем угол по жаре, потом урезаем его, если он больше разрешённого.

Угол по температуре удобно задавать шагами: например, до 24 °C форточка закрыта, а за каждый градус выше 24 прибавляется 15°, но не больше 90°. При 27 °C: $27 - 24 = 3$ градуса сверх нормы, $3 \cdot 15 = 45^\circ$. При 32 °C вышло бы $8 \cdot 15 = 120^\circ$, но форточка дальше 90° не открывается, значит, 90°. Ограничение сверху в Arduino делает `constrain` или обычное `if`.

Запомни. Правила проверяют сверху вниз, правила безопасности (ветер, мороз) – первыми. Ограничение не выбирает угол, а только урезает слишком большой. Рассчитанный угол всегда сверяй с пределами механизма: от 0° до 90° у форточки, от 0° до 180° у сервы.

8.6. Разбираем задачи

Задача 8.1 Вертушка на крыше школы

разминка · анемометр, длина окружности, скорость

Школьный анемометр – вертушка с тремя чашками, от оси до центра каждой чашки 8 см. Ребята засекли по секундомеру: за 12 секунд вертушка сделала 36 оборотов. С какой скоростью движутся чашки? Поправку не учитывай. Ответ в м/с, округли до десятых.

Дано: $r = 8$ см; 36 оборотов за $t = 12$ с.

Найти: $v_{\text{ч}}$ – ?

Решение

Шаг 1. Радиус – в метры. $8 \text{ см} = 8 : 100 = 0,08 \text{ м}$.

Шаг 2. Обороты в секунду. $n = 36 : 12 = 3 \text{ об/с}$.

Шаг 3. Путь чашки за один оборот. Длина круга $2 \cdot 3,14 \cdot 0,08 = 0,5024 \text{ м}$ – примерно полметра.

Шаг 4. Путь за секунду – он же скорость. За секунду три круга: $v_{\text{ч}} = 0,5024 \cdot 3 = 1,507 \text{ м/с}$, округляем до 1,5 м/с.

Полтора метра в секунду – примерно шаг неспешного пешехода. Для лёгкого ветерка, при котором флаг еле шевелится, годится.

⚠ Типичная ошибка. Подставить радиус в сантиметрах: $2 \cdot 3,14 \cdot 8 \cdot 3 = 150,7$, и написать «150,7 м/с» – это скорость самолёта на взлёте. Вторая частая ошибка – взять 36 оборотов вместо 3 оборотов в секунду: формула просит обороты **за секунду**.

Ответ: 1,5 м/с.

Своя задача по программе сферы, модуль 8

Реши сам

8.1. У анемометра на даче радиус вертушки 12 см. За 20 секунд она сделала 30 оборотов. С какой скоростью движутся чашки? Ответ в м/с, округли до десятых.

8.2. Вертушка радиусом 6 см делает 5 оборотов в секунду. Поправочный коэффициент анемометра 2,5: ветер в 2,5 раза быстрее чашек. Какова скорость ветра в км/ч? Округли до целых.

Задача 8.2 Порог в импульсах

анемометр, импульсы и обороты, порог, округление

На вертушке анемометра два магнита, так что за каждый оборот геркон даёт 2 импульса. Радиус вертушки 7 см, поправочный коэффициент 3. Программа 5 секунд считает импульсы и по ним решает, закрывать ли форточки: их закрывают, когда ветер 10 м/с и сильнее. а) За 5 секунд пришло 60 импульсов. Какова скорость ветра? Ответ в м/с, округли до десятых. б) Программисту удобнее сравнивать не скорость, а сразу число импульсов. При каком наименьшем числе импульсов за 5 секунд программа должна закрыть форточки?

Дано: 2 импульса на оборот; $r = 7 \text{ см} = 0,07 \text{ м}$; $k = 3$; окно счёта 5 с; порог 10 м/с.

Найти: а) v при 60 импульсах – ?; б) наименьшее число импульсов для $v \geq 10 \text{ м/с}$ – ?

Решение

Шаг 1. Импульсы – в обороты. Два импульса – один оборот: $60 : 2 = 30$ оборотов за 5 с.

Шаг 2. Обороты в секунду. $n = 30 : 5 = 6 \text{ об/с}$.

Шаг 3. Скорость чашки. Длина круга $2 \cdot 3,14 \cdot 0,07 = 0,4396 \text{ м}$. $v_{\text{ч}} = 0,4396 \cdot 6 = 2,6376 \text{ м/с}$.

Шаг 4. Скорость ветра. $v = 3 \cdot 2,6376 = 7,9128 \text{ м/с}$, округляем: 7,9 м/с. Меньше 10 – форточки можно не закрывать.

Шаг 5. Пункт б): сколько «стоит» один импульс. Скорость ветра растёт вместе с числом импульсов: вдвое больше импульсов – вдвое сильнее ветер. Значит, каждый импульс добавляет одну и ту же долю скорости. В пункте а) 60 импульсов дали 7,9128 м/с (берём число до округления), на один импульс приходится $7,9128 : 60 = 0,1319 \text{ м/с}$.

Шаг 6. Сколько импульсов нужно на 10 м/с. $10 : 0,1319 = 75,8$ импульса.

Шаг 7. Округляем в нужную сторону. Импульсы бывают только целые. 75 импульсов – это ветер чуть меньше 10 м/с, закрывать ещё рано. 76 – уже чуть больше 10 м/с. Значит, порог – 76. Проверим прямым счётом: 76 импульсов – это 38 оборотов за 5 с, 7,6 об/с, $0,4396 \cdot 7,6 \cdot 3 = 10,02 \text{ м/с}$ – и правда чуть больше десяти.

⚠ Типичная ошибка. Отбросить дробную часть и ответить 75. При 75 импульсах ветер 9,89 м/с – меньше порога, и форточки останутся открытыми. Здесь вопрос «наименьшее целое, при котором уже не меньше 10», поэтому округляем только вверх. Вышло бы 75,2 – ответ всё равно был бы 76. Вторая ошибка – забыть про два магнита и получить порог 38 импульсов: тогда форточки захлопнутся уже при ветре около 5 м/с.

Ответ: а) 7,9 м/с; б) 76 импульсов.

Своя задача по программе сферы, модуль 8

Реши сам

8.3. На вертушке один магнит, радиус вертушки 9 см, поправочный коэффициент 2,8. За 10 секунд плата насчитала 45 импульсов. Какова скорость ветра? Ответ в м/с, округли до десятых.

8.4. Анемометр с двумя магнитами, радиус вертушки 8 см, поправочный коэффициент 3. Программа считает импульсы 4 секунды и при ветре 15 м/с и сильнее отправляет в чат тепличника предупреждение «шторм». При каком наименьшем числе импульсов за 4 секунды уйдёт предупреждение?

Задача 8.3 Плавная форточка

сервомотор, цикл for, время работы программы

Форточку открывает серва SG90. Чтобы рама не хлопала, её открывают плавно:

```
for (int a = 0; a ≤ 90; a += 3) {  
    okno.write(a);  
    delay(40);  
}
```

а) Сколько раз выполнится команда `okno.write(a)`? б) Сколько времени займёт открытие? Временем, которое тратят сами команды, пренебреги. Ответ в секундах. в) Во сколько раз это дольше, чем если бы серва сразу получила `write(90)` и повернулась со своей паспортной скоростью 0,1 с на 60°?

Дано: a от 0 до 90 включительно, шаг 3°; пауза 40 мс на шаге; серва – 0,1 с на 60°.

Найти: число шагов – ?; время – ?; во сколько раз дольше – ?

Решение

Шаг 1. Какие значения примет a . 0, 3, 6, ..., 87, 90. Условие $a \leq 90$ пропускает и само 90 – знак «меньше или равно».

Шаг 2. Считаем их. От 0 до 90 шагами по 3 – это $90:3 = 30$ промежутков. Значений на одно больше, как столбов в заборе: $30 + 1 = 31$. Команда `write` выполнится 31 раз.

Шаг 3. Время. После каждой команды – пауза 40 мс, в том числе после последней:
 $31 \cdot 40 = 1240$ мс. Переводим: $1240 \text{ мс} = 1240 : 1000 = 1,24$ с.

Шаг 4. Сколько бы заняла быстрая серва. 60° – за 0,1 с, значит, 90° – за полтора раза дольше:
 $0,1 \cdot 90:60 = 0,15$ с.

Шаг 5. Сравниваем. $1,24:0,15 = 8,27$, округлим до 8,3. Плавное открытие примерно в 8 раз дольше. Чуть больше секунды на открытие – как раз то, что нужно: форточка не хлопает, но и не ползёт минуту.

❗ **Типичная ошибка.** Разделить 90 на 3 и ответить «30 раз, 1,2 с». Забыт первый шаг, $a = 0$: серва сначала встанёт на 0° , и на это тоже уходит пауза. Такая «потеря единицы» – самая частая ошибка в задачах про циклы.

Ответ: а) 31 раз; б) 1,24 с; в) в 8,3 раза.

Своя задача по программе сферы, модуль 7

Проверить себя можно, запустив цикл в Python – он работает так же:

```
shagi = list(range(0, 91, 3)) # от 0 до 90 включительно, шаг 3
print(len(shagi))             # 31
print(len(shagi) * 40 / 1000) # 1.24 (секунды)
```

Реши сам

8.5. Вечером форточку закрывают так: `for (int a = 180; a ≥ 60; a -= 2) { okno.write(a); delay(25); }`. Сколько раз выполнится `okno.write(a)` и сколько миллисекунд займёт закрытие?

8.6. Угол сервы задаёт ручка потенциометра: `int a = map(analogRead(A0), 0, 1023, 0, 180);`. а) На какой угол встанет серва, если `analogRead` вернул 700? б) При каком наименьшем значении `analogRead` серва встанет на 90° ?

Задача 8.4 Вентилятор на 60 процентов

DC-мотор, ШИМ, производительность вентилятора, объём

Теплица – прямоугольный «домик» 4 м в ширину, 6 м в длину, средняя высота 2,5 м. Вентилятор с DC-мотором на полной мощности (`analogWrite(9, 255)`) прогоняет 480 м^3 воздуха в час. Автоматика включила его командой `analogWrite(9, 153)`. Считай, что поток воздуха пропорционален числу ШИМ. а) Сколько процентов мощности получил вентилятор? б) За сколько минут он сменит весь воздух в теплице один раз? в) Сколько раз в час сменится воздух?

Дано: $4 \times 6 \times 2,5 \text{ м}$; $480 \text{ м}^3/\text{ч}$ при 255; ШИМ 153.

Найти: процент – ?; время смены воздуха – ?; обменов в час – ?

Решение

Шаг 1. Процент мощности. $153:255 = 0,6$, то есть 60 %.

Шаг 2. Производительность при 60 %. $480 \cdot 0,6 = 288 \text{ м}^3/\text{ч}$.

Шаг 3. Объём теплицы. $4 \cdot 6 \cdot 2,5 = 60 \text{ м}^3$.

Шаг 4. Время смены воздуха. $60:288 = 0,2083 \text{ ч}$. Переводим в минуты: $0,2083 \cdot 60 = 12,5 \text{ мин}$.

Шаг 5. Обменов в час. В часе 60 мин, один обмен – 12,5 мин: $60:12,5 = 4,8$ раза. То же самое выйдет, если разделить производительность на объём: $288:60 = 4,8$.

Прикинь для сравнения полную мощность: $60:480 \cdot 60 = 7,5$ мин. На 60 % вентилятор работает медленнее, и 12,5 минуты вместо 7,5 выглядят честно.

⚠ Типичная ошибка. Оставить ответ в часах («0,2 мин») или умножить 0,2083 на 100 вместо 60 и получить «20,8 мин». Час – это 60 минут, а не 100. Ещё одна ловушка – считать процент от 100, а не от 255: `analogWrite(9, 153)` – это не 153 %, а 60 %.

Ответ: а) 60 %; б) 12,5 мин; в) 4,8 раза.

Своя задача по программе сферы, модуль 7

Реши сам

8.7. На полной мощности (`analogWrite 255`) вентилятор делает 1500 оборотов в минуту. Сколько оборотов в минуту он будет делать при `analogWrite(9, 102)`? Скорость считай пропорциональной числу ШИМ.

8.8. Теплица 3×5 м, средняя высота 2 м. Агроном хочет, чтобы воздух в ней сменялся каждые 10 минут. Вентилятор на полной мощности даёт $450 \text{ м}^3/\text{ч}$. Какое число надо написать в `analogWrite`, чтобы получить нужный поток? Поток считай пропорциональным числу ШИМ.

Задача 8.5 Панель на лето

высота солнца, широта, угол наклона панели

Солнечную панель метеостанции в Новосибирске (широта 55°) переставляют по сезонам. На какой угол к горизонту её надо наклонить, чтобы 22 июня в полдень лучи падали на неё перпендикулярно? Какой наклон понадобится 22 декабря? На сколько градусов надо изменить наклон при переходе от летней настройки к зимней?

Дано: широта 55° ; летом солнце в полдень на 23° выше, чем в равноденствие, зимой на 23° ниже.

Найти: наклон 22 июня – ?; наклон 22 декабря – ?; разница – ?

Решение

Шаг 1. Высота солнца в равноденствие. $90^\circ - 55^\circ = 35^\circ$.

Шаг 2. Высота солнца 22 июня. $35^\circ + 23^\circ = 58^\circ$.

Шаг 3. Наклон панели летом. Лучи должны идти перпендикулярно панели, значит, она отклонена от горизонта на $90^\circ - 58^\circ = 32^\circ$. Почему так, видно на рисунке к разделу 8.4: луч, панель и земля образуют треугольник. Угол между лучом и панелью 90° , между лучом и землёй 58° , а три угла треугольника в сумме дают 180° . На наклон панели остаётся $180^\circ - 90^\circ - 58^\circ = 32^\circ$.

Шаг 4. Зимой. Солнце: $35^\circ - 23^\circ = 12^\circ$. Наклон: $90^\circ - 12^\circ = 78^\circ$.

Шаг 5. Разница. $78^\circ - 32^\circ = 46^\circ$.

Зимой панель стоит почти вертикально – так и есть: зимнее солнце в Сибири низкое и светит сбоку, прямо в «стену».

❗ Типичная ошибка. Поставить панель на угол, равный высоте солнца: летом на 58° . Тогда лучи падают вовсе не перпендикулярно: между ними и перпендикуляром к панели выйдет $58^\circ - 32^\circ = 26^\circ$, и панель потеряет около десятой части света. Проверь себя рисунком: солнце высоко – панель лежит почти плоско.

Ответ: летом 32° , зимой 78° , изменить на 46° .

Своя задача по программе сферы, модуль 8

Реши сам

8.9. Город стоит на широте 54° . На какой угол к горизонту наклонить панель, чтобы в полдень в день равноденствия лучи падали на неё перпендикулярно?

8.10. Панель на теплице под Новосибирском (широта 55°) наклоняет серва: при `write(0)` панель лежит горизонтально, а угол сервы равен углу наклона панели. Летом в полдень (22 июня) стоит `write(32)`. Какое число надо написать в `write`, чтобы панель смотрела прямо на полуденное солнце в день осеннего равноденствия?

Задача 8.6*. Стоит ли ставить трекер

солнечный трекер, доля света, энергия за день, ватт-час

Метеостанции нужна солнечная панель, которая в полную мощность (лучи перпендикулярно) даёт 20 Вт. Можно поставить её неподвижно, чтобы она смотрела на полуденное солнце, а можно на трекер. Электроника и серва трекера расходуют в среднем 1,5 Вт всё светлое время. Возьми модель из раздела 8.4: солнце светит одинаково ярко с 6:00 до 18:00, смещается на 15° в час, день делим на куски по 2 часа и берём угол в середине куска. Долю света бери из таблицы в разделе 8.4. а) Сколько ватт-часов за день даст неподвижная панель? б) Сколько ватт-часов останется от трекера, если вычесть его собственный расход? в) На сколько процентов трекер выгоднее? Округли до целых.

Дано: $P = 20$ Вт; день 12 ч, куски по 2 ч; 15° в час; расход трекера 1,5 Вт; таблица долей.

Найти: $E_{\text{неп}}$ – ?; $E_{\text{тр}}$ – ?; выигрыш в % – ?

Решение

Шаг 1. Углы в серединах кусков. Середины кусков – 7, 9, 11, 13, 15 и 17 часов. От полудня до 7:00 – 5 часов, солнце в стороне на $5 \cdot 15^\circ = 75^\circ$. До 9:00 – 3 часа, $3 \cdot 15^\circ = 45^\circ$. До 11:00 – 1 час, 15° . После полудня – те же 15° , 45° , 75° .

Шаг 2. Доля света по таблице. 75° – 26 %, 45° – 71 %, 15° – 97 %. По кускам: 26, 71, 97, 97, 71, 26 %.

Шаг 3. Мощность неподвижной панели в каждом куске. 26 % от 20 Вт – это $20 \cdot 0,26 = 5,2$ Вт; дальше $20 \cdot 0,71 = 14,2$ Вт и $20 \cdot 0,97 = 19,4$ Вт. Каждая мощность держится 2 часа.

Шаг 4. Энергия неподвижной панели. Одну половину дня: $(5,2 + 14,2 + 19,4) \cdot 2 = 77,6$ Вт·ч. Вторая половина такая же: $E_{\text{неп}} = 77,6 \cdot 2 = 155,2$ Вт·ч.

Шаг 5. Трекер. Панель всё время смотрит на солнце: $20 \cdot 12 = 240$ Вт·ч. Собственный расход: $1,5 \cdot 12 = 18$ Вт·ч. Остаётся $E_{\text{тр}} = 240 - 18 = 222$ Вт·ч.

Шаг 6. Выигрыш. Разница $222 - 155,2 = 66,8$ Вт·ч. В процентах от неподвижной панели: $66,8 : 155,2 \cdot 100 = 43 \%$.

В модели трекер выгоднее на 43 %. В жизни меньше: наша модель считает утреннее солнце таким же ярким, как полуденное, а оно слабее. Так что это оценка «не больше», а не точное обещание.

 **Типичная ошибка.** Забыть про вторую половину дня и получить 77,6 Вт·ч. Другая ошибка – сравнивать проценты с разной основой: «трекер даёт 240, неподвижная 155,2, разница 84,8 – значит, на 84,8 %». Во-первых, забыт расход трекера, во-вторых, проценты считают от того, с чем сравниваем, – от 155,2 Вт·ч, а не от 100.

Ответ: а) 155,2 Вт·ч; б) 222 Вт·ч; в) на 43 %.

Своя задача по программе сферы, модуль 8

Реши сам

8.11. В той же модели маленькая панель на 5 Вт, а электроника трекера расходует 1 Вт всё светлое время (12 ч). Сколько ватт-часов за день даст неподвижная панель и сколько останется от трекера за вычетом его расхода? На сколько ватт-часов трекер выгоднее?

8.12*.* В той же модели панель на 10 Вт. При каком расходе электроники трекера (в ваттах, всё светлое время – 12 ч) трекер перестаёт давать выигрыш по сравнению с неподвижной панелью? Округли до десятых.

Задача 8.7*.*. Метеостанция решает за форточку

правила автоматики, приоритет, ограничение угла, выбор датчика

В теплице температуру меряет DHT11, на улице – метеостанция с анемометром и термометром. Угол форточки программа выбирает по правилам, проверяя их сверху вниз:

1. Ветер 12 м/с и сильнее – форточка закрыта (0°).
2. Иначе угол считается по температуре в теплице: до 24 °С включительно – 0°, за каждый градус выше 24 – плюс 15°, но не больше 90°.
3. Если на улице холоднее +10 °С, угол из правила 2 урезается до 30° (если он больше 30°).

а) Метеостанция стоит в Новосибирске круглый год, зимой морозы до –35 °С. Годится ли для уличного термометра датчик DHT11? Ответ «да» или «нет». б) Вот записи за один майский день. Найди угол форточки в каждый момент. Ответ – четыре числа через запятую.

Таблица 8.3

Время	В теплице, °C	На улице, °C	Ветер, м/с
8:00	22	6	3
11:00	28	12	5
14:00	31	15	13
17:00	27	9	6

Дано: три правила по порядку; таблица на 4 момента; DHT11 меряет от 0 до 50 °C.

Найти: годится ли DHT11 на улицу; угол форточки в 8:00, 11:00, 14:00, 17:00.

Решение

Шаг 1. Пункт а). По таблице из раздела 8.1 DHT11 меряет температуру от 0 °C. Зимой на улице до -35 °C – датчик выйдет за свой диапазон. Не годится, нужен DHT22 (он работает от -40 °C) или другой морозостойкий датчик.

Шаг 2. 8:00. Ветер 3 м/с – меньше 12, первое правило не срабатывает. В теплице 22 °C – не выше 24, по второму правилу 0°. На улице 6 °C – холоднее 10, но урезать нечего: 0° и так меньше 30°. Угол 0°.

Шаг 3. 11:00. Ветер 5 м/с – дальше. В теплице 28 °C: $28 - 24 = 4$ градуса, $4 \cdot 15 = 60^\circ$ – не больше 90°, оставляем. На улице 12 °C – не холоднее 10, урезать не нужно. Угол 60°.

Шаг 4. 14:00. Ветер 13 м/с – 12 и больше. Первое правило: форточка закрыта, остальные правила не смотрим. Угол 0°, хотя в теплице 31 °C. Проветривать придётся вентилятором.

Шаг 5. 17:00. Ветер 6 м/с – дальше. В теплице 27 °C: $3 \cdot 15 = 45^\circ$. На улице 9 °C – холоднее 10, угол урезается до 30°. Угол 30°.

Так выглядят те же правила в программе. Функция получает три числа и возвращает угол:

```
int ugo1(float tVnutri, float tUlica, float veter) {
    if (veter ≥ 12) return 0;           // правило 1: шторм - закрыто
    int a = 0;
    if (tVnutri > 24) a = (tVnutri - 24) * 15; // правило 2
    if (a > 90) a = 90;
    if (tUlica < 10 && a > 30) a = 30;      // правило 3: холодно снаружи
    return a;
}
```

return сразу возвращает ответ и выходит из функции, поэтому при сильном ветре до температуры дело не доходит.

⚠ Типичная ошибка. В 14:00 посчитать угол по температуре: $(31 - 24) \cdot 15 = 105^\circ$, урезать до 90° и ответить «90». Правило про ветер стоит первым и отменяет всё остальное. В 17:00 наоборот – забывают третье правило и пишут 45°.

Ответ: а) нет; б) 0, 60, 0, 30.

По мотивам задачи образовательного курса НТО Junior 2023/24, сфера «Технологии и среда обитания», «Программирование для умного сити-фермера»

Реши сам

8.13. Правила те же, что в задаче 8.7. Найди угол форточки в четыре момента: а) в теплице 33 °С, на улице 18 °С, ветер 4 м/с; б) 26 °С, 8 °С, 11 м/с; в) 29 °С, 7 °С, 3 м/с; г) 25 °С, 20 °С, 12 м/с. Ответ – четыре числа через запятую.

8.14*.* Правила те же, что в задаче 8.7, но скорость ветра программа получает от анемометра задачи 8.2: два магнита, радиус 7 см, поправка 3, счёт импульсов 5 секунд. В теплице 30 °С, на улице 14 °С. Какой угол выберет программа, если за 5 секунд пришло а) 80 импульсов; б) 95 импульсов?

Ответы и решения

Сначала реши задачу сам и только потом сверяйся. Не сошлось – найди в решении шаг, на котором твой путь разошёлся с нашим.

К главе 1. Растение как живая система

1.1. Ответ: 1356.

1.2. Ответ: 1246.

Подсказка: про утверждение 5 вспомни график насыщения из раздела 1.2 – слово «обязательно» его выдаёт.

1.3. Ответ: 21345 (анемометр лишний).

1.4. Ответ: БВА.

Подсказка: проверь каждую теплицу по всем трём столбцам таблицы норм – подходит только одна культура.

1.5. Ответ: 125.

Подсказка: ни углекислый газ, ни кислород, ни корни в этом опыте не проверяли – менялся только свет.

1.6. Ответ: воздух – банки 2 и 3; тепло – банки 2 и 4.

Подсказка: сравнивают две банки, которые отличаются одним-единственным условием; банка 1 отличается от банки 2 водой.

1.7. Ответ: тепло; 7,2 кг.

1.8. Ответ: а) свет и азот; б) 3,0 кг.

Подсказка: 3,4 кг из 4 кг – это 85 %, поэтому подтянуть нужно всё, что ниже 85 %; после ламп короткой доской станет азот.

1.9. Ответ: 9 ч 39 мин.

1.10. Ответ: в 21:46; 136 мин.

Подсказка: утренней досветки нет, поэтому 14 часов света отсчитывай от восхода.

1.11. Ответ: 20 ppm в час; 550 ppm.

1.12. Ответ: 6 ppm в минуту.

Подсказка: при свете датчик видит разницу: фотосинтез забирает газ, а дыхание в это же время добавляет 2 ppm в минуту, так что фотосинтез = 4 + 2.

1.13. Ответ: 70 %; при 14 °С.

Подсказка: здесь обратная задача – делим пар на предел: $12,1 : 17,3 \cdot 100 \approx 70 \%$.

1.14. Ответ: а) 22 °С; б) 150 г; в) да, выпадет.

Подсказка: пара в воздухе $23,0 \cdot 0,8 = 18,4$ г/м³, а при +20 °С предел только 17,3 г/м³.

К главе 2. Микроклимат теплицы: считаем

2.1. Ответ: 0,185 %.

2.2. Ответ: включатся; не хватает 700 лк.

Подсказка: сначала переведи порог в люксы: 4,5 клк = 4500 лк.

2.3. Ответ: 60 %; после остывания – 79 %.

2.4. Ответ: на 11 процентных пунктов; на 20 %.

2.5. Ответ: 9 ч 30 мин; 40 % суток.

2.6. Ответ: 5 ч 23 мин.

Подсказка: здесь света больше, чем нужно: найди длину дня и вычти из неё 12 ч.

2.7. Ответ: 67 %.

Подсказка: спрашивают долю замеров **в норме**; 27 °С – ещё норма.

2.8. Ответ: 6 ч; 25 %.

2.9. Ответ: 23.

Подсказка: в 23:10 лампы не горят, хотя темно: время вне промежутка с 6:00 до 22:00.

2.10. Ответ: 46.

Подсказка: $0,16 \% = 1600$ ppm; для «или» хватает одной верной части.

2.11. Ответ: 104 г.

2.12. Ответ: 37 мин.

Подсказка: $0,25$ л = 250 мл = 250 г пара в час; пара нужно добавить 151,5 г.

2.13. Ответ: включался 2 раза; работал 90 мин.

2.14. Ответ: 35 мин; 58 % часа.

Подсказка: при 84 % вентилятор ещё не включается, а после 13:50 работает до конца часа.

К главе 3. Графики и таблицы агронома

3.1. Ответ: на 18 °С; в среднем на 2,25 °С в час.

3.2. Ответ: с 6:00 до 8:00, по 3 °C в час.

Подсказка: найди изменение на каждом двухчасовом участке и раздели на 2 – самый крутой участок даёт самую большую скорость.

3.3. Ответ: средняя влажность 41 %, размах 12 %.

3.4. Ответ: 17 °C.

Подсказка: задача обратная: сумма пяти замеров равна $5 \cdot 14 = 70$, вычти из неё четыре известных.

3.5. Ответ: 60; 12; 3.

Подсказка: " ≥ 12 " считает и само число 12.

3.6. Ответ: 12.

Подсказка: в B2:B25 24 клетки; B2;B25 – всего две клетки; полусумма максимума и минимума – не среднее.

3.7. Ответ: 50.

Подсказка: $20 \cdot 20 = 400$ – мало, $20 \cdot 50 = 1000$ – уже хватает на 980, а 100 и 200 сделают график мельче.

3.8. Ответ: с 18:00 до 20:00, показание упало на 380.

Подсказка: участок с 12:00 до 14:00 почти такой же крутой (на 370) – сравни все шесть разностей.

3.9. Ответ: в 15:00.

3.10. Ответ: в 18:36.

Подсказка: 1,6 ч – это 1 ч 36 мин, а не 1 ч 60 мин.

3.11. Ответ: в теплице Б; в А потеплело на 6 °C, в Б – на 15 °C.

Подсказка: сначала цена клетки: $8 : 8 = 1$ °C у А и $40 : 8 = 5$ °C у Б.

3.12. Ответ: у Ани 23 °C, у Бори 19,25 °C; завышено у Ани.

Подсказка: у Ани шесть из девяти записей сделаны с 9:00 до 14:00, в самые тёплые часы, – дневное тепло посчитано много раз.

К главе 4. Вода и полив

4.1. Ответ: 24 л; слой 4 мм.

Подсказка: 1 л на 1 м^2 – это слой в 1 мм.

4.2. Ответ: 48 л в сутки; на 3 полных суток.

Подсказка: $150 : 48 = 3,125$ – спрашивают полные сутки, округляем вниз.

4.3. Ответ: 45 л.

4.4. Ответ: 21,2 мин.

Подсказка: радиус 30 см = 3 дм, высота 9 дм; объём около 254 л.

4.5. Ответ: на 30 %.

4.6. Ответ: 600 л; 210 л.

Подсказка: 390 л – это 65 % от прежнего расхода.

4.7. Ответ: 5880 Па.

4.8. Ответ: в баке А, в 1,6 раза.

Подсказка: литры не нужны – давление на дно зависит только от высоты воды.

4.9. Ответ: 2346.

4.10. Ответ: 17 л.

Подсказка: объём испарившегося слоя – площадь круга воды, умноженная на толщину слоя; удобно считать в дециметрах (радиус 3 дм, слой 0,06 дм).

4.11. Ответ: 60 мин.

4.12. Ответ: 19,7 мин; 24 %.

Подсказка: капельницы берут 2,4 л/мин; наполнение – 150 л при 7,6 л/мин, спад – 150 л при 2,4 л/мин.

К главе 5. Электричество, энергия и датчики

5.1. Ответ: 500 Ом.

Подсказка: 12 мА = 0,012 А.

5.2. Ответ: 13,6 мА – выдержит; с резистором 22 Ом – 136,4 мА, светодиод сгорит.

5.3. Ответ: 880 Ом последовательно; 55 Ом параллельно.

5.4. Ответ: 10 мА; 3 В на резисторе 300 Ом и 2 В на резисторе 200 Ом.

Подсказка: сначала общее сопротивление 500 Ом, потом ток, потом $U = I \cdot R$ для каждого резистора.

5.5. Ответ: –9,9 °С.

Подсказка: напряжение меньше 0,5 В – значит, на улице мороз и температура выйдет отрицательной.

5.6. Ответ: 56 %.

Подсказка: здесь во влажной земле число меньше, чем в сухой; подставь в формулу калибровки как есть: $(600 - 870) / (390 - 870) \cdot 100$.

5.7. Ответ: 730.

Подсказка: напряжение на фоторезисторе: $5 \cdot 25/35 = 3,571$ В.

5.8*. Ответ: 568; условие 1.

Подсказка: теперь А0 меряет напряжение на нижнем резисторе, а оно в темноте падает.

5.9. Ответ: 9 Вт; 54 Вт·ч.

5.10. Ответ: 13,3 ч; не хватит.

Подсказка: $2,4 \text{ А} \cdot \text{ч} = 2400 \text{ мА} \cdot \text{ч}$.

5.11. Ответ: 0,558 кВт·ч.

5.12. Ответ: 34.

Подсказка: при параллельном включении токи складываются: нужно 12 В и не меньше 1,05 А.

5.13. Ответ: 2000 Вт (2 кВт); 20 кВт·ч.

5.14*. Ответ: $-5,0$ °С; после утепления – $3,6$ °С.

Подсказка: задача обратная: найди, на сколько градусов 2000 Вт могут «перекрыть» разницу ($2000/100 = 20$ °С), и прибавь к -25 °С.

К главе 6. Программирование умной теплицы

6.1. Ответ: 1,5 л; робот окажется у горшка 11.

6.2. Ответ: программа А – горшки 1, 3, 4, 6, 800 мл; программа Б – горшки 3, 4, 6, 9, тоже 800 мл.

Подсказка: в программе Б робот сначала уезжает, поэтому горшок 1 он не проверяет, зато доезжает до горшка 9.

6.3. Ответ: 255, 100, 100, 0, 0.

Подсказка: ровно 800 не больше 800, поэтому первая ветка не сработает.

6.4. Ответ: 223; синий не загорится никогда.

Подсказка: всё, что меньше 18, меньше и 26, – такую температуру перехватит первая ветка.

6.5. Ответ: 400 мл; через 226 с.

Подсказка: круг с поливом длится 38 с, без полива – 30 с; до восьмой проверки 7 кругов, из них два с поливом.

6.6. Ответ: 20000 (20 с).

6.7. Ответ: строка 8, нужно `door = HIGH`; вентилятор не включится никогда, даже при закрытой двери.

Подсказка: `door = HIGH` кладёт в `door` значение `HIGH`, и условие каждый раз оказывается верным.

6.8. Ответ: после 2-й, 3-й и 4-й проверки (трёх из четырёх) – насос включился и больше не выключается; нужна ветка `else { digitalWrite(PUMP, LOW); }`.

6.9. Ответ: БГДЕАВЖ.

6.10. Ответ: 1780000 (1780 с).

Подсказка: 20 с работы увлажнителя тоже входят в 30 минут.

6.11. Ответ: 2 раза; после последнего замера увлажнитель выключен.

6.12. Ответ: а) 70 мин; б) 30 мин.

6.13. Ответ: 001122211101; режим менялся 5 раз.

6.14. Ответ: «тревога» – из неё не выходит ни одной стрелки. Выход, например: «нажата кнопка сброса / погасить красный» → «ждём».

К главе 7. Сигналы и обработка данных

7.1. Ответ: 37 отсчётов.

7.2. Ответ: на 10 суток; на 1 сутки.

Подсказка: $2880 \cdot 5 \text{ мин} = 14\,400 \text{ мин} = 240 \text{ ч}$; $2880 \cdot 30 \text{ с} = 86\,400 \text{ с}$ – это одни сутки.

7.3. Ответ: 4 числа: 1400, 1467, 1533, 1667 лк.

7.4. Ответ: стёртый отсчёт 420 лк; второе среднее 433,3 лк.

Подсказка: задача обратная: сумма первого окна равна $3 \cdot 420 = 1260$, из неё вычти два известных отсчёта.

7.5. Ответ: 2 ложных включения (3-я и 7-я минуты); настоящее – на 13-й минуте.

Подсказка: здесь насос включается, когда влажность становится **ниже** порога, так что ложные срабатывания дают выбросы вниз.

7.6. Ответ: на 8-й минуте; на 5 минут позже (по старому правилу – на 3-й минуте).

Подсказка: счётчик «подряд» сбрасывается, как только отсчёт оказался ниже 28 °С.

7.7. Ответ: 3 с; 20 капель в минуту.

7.8. Ответ: 150 оборотов в минуту.

Подсказка: фронты – на 3, 11, 19 и 27-м отсчётах; период 8 отсчётов = 0,4 с.

7.9. Ответ: 21 и 13.

7.10. Ответ: а) 7 битов; б) 8 битов; в) 14 с.

Подсказка: с шагом 0,5 % значений 201, а $2^7 = 128$ уже мало.

7.11. Ответ: 2,5 ч.

7.12. Ответ: 18,75 °С.

Подсказка: найди сумму всех отсчётов: $12 \cdot 23,5 = 282$, потом замени в ней 83 на медиану 26.

7.13. Ответ: а) суточное – $1/24$ раза в час (около 0,04), от форточки – 3 раза, дрожь датчика – 360 раз в час; б) 14 °С.

Подсказка: после фильтра остаётся одна суточная волна, а её размах – две амплитуды.

7.14. Ответ: окно 6; 507,5.

К главе 8. Метеостанция и автоматика

8.1. Ответ: 1,1 м/с.

8.2. Ответ: 17 км/ч.

Подсказка: скорость чашки 1,884 м/с, ветра – в 2,5 раза больше, потом умножь на 3,6.

8.3. Ответ: 7,1 м/с.

8.4. Ответ: 80 импульсов.

8.5. Ответ: 61 раз; 1525 мс.

Подсказка: цикл идёт вниз: от 180 до 60 – это 60 шагов по 2°, значений на одно больше.

8.6. Ответ: а) 123°; б) 512.

Подсказка: при 511 выходит 89,9, а `map` отбрасывает дробную часть – получается 89.

8.7. Ответ: 600 оборотов в минуту.

8.8. Ответ: 102.

Подсказка: задача обратная: объём 30 м³ за 10 мин – это 180 м³/ч, то есть 40 % от 450 м³/ч.

8.9. Ответ: 54°.

Подсказка: в равноденствие наклон панели равен широте.

8.10. Ответ: 55.

8.11. Ответ: неподвижная – 38,8 Вт·ч; трекер – 48 Вт·ч; выгоднее на 9,2 Вт·ч.

8.12. Ответ: 3,5 Вт.

Подсказка: неподвижная панель даёт 77,6 Вт·ч, трекер без расхода – 120 Вт·ч; разницу 42,4 Вт·ч раздели на 12 ч.

8.13. Ответ: 90, 30, 30, 0.

Подсказка: в б) по жару выходит 30°, а правило 3 урезает только то, что больше 30°; в г) ветер 12 м/с – это уже «12 и сильнее».

8.14. Ответ: а) 90° ; б) 0° .

Подсказка: один импульс – это 0,1319 м/с ветра (задача 8.2): 80 импульсов – около 10,6 м/с, ещё не шторм; 95 импульсов – около 12,5 м/с.

Справочник

Всё, что может понадобиться под рукой во время решения: числа и нормы, перевод единиц, формулы и правила из глав, мини-шпаргалки по Arduino, Tinkercad, ПРИМС и таблицам, словарь и все «Запомни» из глав на нескольких страницах.

Числа, которые используются в книге

Если в условии задачи на отборе дано другое значение, бери его из условия.

Таблица С.1

Величина	Значение
Arduino Uno: напряжение питания датчиков	5 В
analogRead – что возвращает	целое число от 0 (0 В) до 1023 (5 В)
analogWrite (ШИМ) – что принимает	целое число от 0 (выключено) до 255 (полная мощность)
Обычный светодиод: напряжение / допустимый ток	около 2 В / 20 мА
Термометр TMP36 в Tinkercad: что умеет измерять	от –40 до 125 °С
Плотность воды	1 кг/л = 1 г/мл = 1000 кг/м³
Ускорение свободного падения	$g = 9,8$ Н/кг
Углекислый газ в уличном воздухе	около 420–430 ppm (0,042–0,043 %)
Число π	3,14 (кнопка π на калькуляторе)

Микроклимат теплицы: ориентиры

Нормы для каждой культуры свои и меняются с возрастом растения. Ниже – ориентиры для многих тепличных овощей (томаты, огурцы, перцы), примерно. На отборе всегда бери нормы из условия задачи.

Таблица С.2

Параметр	Для многих культур примерно
Температура воздуха днём	20–28 °С
Температура воздуха ночью	на 5–10 °С ниже дневной
Относительная влажность воздуха	60–80 %
Углекислый газ при подкормке в теплице	600–1000 ppm

Жара опаснее, чем кажется: выше 30–32 °С пыльца у томатов развивается плохо, а при 35 °С и выше завязей почти не будет, даже если листья выглядят бодро.

Перевод единиц

Таблица С.3

Что	Как переводить
Время	$1 \text{ сут} = 24 \text{ ч} \cdot 1 \text{ ч} = 60 \text{ мин} = 3600 \text{ с} \cdot 1 \text{ с} = 1000 \text{ мс}$
Объём	$1 \text{ м}^3 = 1000 \text{ л} \cdot 1 \text{ л} = 1 \text{ дм}^3 = 1000 \text{ см}^3 = 1000 \text{ мл} \cdot 1 \text{ мл} = 1 \text{ см}^3$
Масса	$1 \text{ т} = 1000 \text{ кг} \cdot 1 \text{ кг} = 1000 \text{ г}$
Длина	$1 \text{ м} = 100 \text{ см} = 1000 \text{ мм} \cdot 1 \text{ км} = 1000 \text{ м}$
Площадь	$1 \text{ м}^2 = 10\,000 \text{ см}^2 \cdot 1 \text{ а (сотка)} = 100 \text{ м}^2 \cdot 1 \text{ га} = 10\,000 \text{ м}^2$
Напряжение	$1 \text{ В} = 1000 \text{ мВ}$
Ток	$1 \text{ А} = 1000 \text{ мА}$
Сопротивление	$1 \text{ кОм} = 1000 \text{ Ом}$
Мощность и энергия	$1 \text{ кВт} = 1000 \text{ Вт} \cdot 1 \text{ кВт} \cdot \text{ч} = 1000 \text{ Вт} \cdot \text{ч}$
Ёмкость аккумулятора	$1 \text{ А} \cdot \text{ч} = 1000 \text{ мА} \cdot \text{ч}$
Доли газа в воздухе	$1 \% = 10\,000 \text{ ppm} \cdot 1 \text{ ppm} = 0,0001 \%$
Давление	$1 \text{ кПа} = 1000 \text{ Па}$
Скорость	$1 \text{ м/с} = 3,6 \text{ км/ч}$

Совет. Перед тем как считать, выпиши все числа из условия и переведи их в одни единицы: миллилитры с литрами, минуты с секундами, миллиамперы с амперами не складываются и не делятся друг на друга.

Формулы и правила

Если в условии задачи правило записано по-своему, решай по условию.

Растение в двух строчках

Фотосинтез идёт в зелёных листьях и только на свету:

углекислый газ + вода → сахар (глюкоза) + кислород.

Дыхание идёт во всех живых клетках растения днём и ночью, со светом и без:

сахар + кислород → углекислый газ + вода + энергия.

Поэтому ночью в закрытой теплице углекислого газа становится больше, а днём, на солнце, растения его «съедают». Воду и растворённые минеральные вещества растение берёт кор-

нями из почвы – это корневое питание. Вода поднимается по стеблю и испаряется через листья.

Проценты и пропорции

Таблица С.4

Что	Как найти	Пример
p процентов от числа A	$A \cdot \frac{p}{100}$	15 % от 200 л: $200 \cdot 0,15 = 30$ л
Какую часть A составляет от B	$\frac{A}{B} \cdot 100\%$	18 л из 24 л: $\frac{18}{24} \cdot 100\% = 75\%$
На сколько процентов выросло	$\frac{\text{новое} - \text{старое}}{\text{старое}} \cdot 100\%$	с 60 до 75: $\frac{15}{60} \cdot 100\% = 25\%$
Пропорция: одно растёт во столько же раз, во сколько другое	$\frac{a_1}{b_1} = \frac{a_2}{b_2}$	3 л на 4 куста, на 10 кустов: $3 \cdot \frac{10}{4} = 7,5$ л

Проценты процентам рознь. Влажность воздуха поднялась с 60 % до 75 %. Она выросла на 15 процентных пунктов – это разность. А если спросят «на сколько процентов выросла», считай от прежнего значения: на 25 %. Читай вопрос внимательно.

Время и интервалы

Сколько прошло: из позднего времени вычесть раннее. Если минут не хватает, займи час: 1 ч = 60 мин. С 06:40 до 20:15: 20 ч 15 мин = 19 ч 75 мин, и 19 ч 75 мин – 6 ч 40 мин = 13 ч 35 мин.

Через полночь: до полуночи плюс после полуночи. С 22:30 до 05:00 – это 1 ч 30 мин + 5 ч = 6 ч 30 мин.

Сколько замеров за сутки: датчик опрашивают раз в 10 минут, в сутках $24 \cdot 60 = 1440$ минут, значит, замеров $1440:10 = 144$.

Среднее и графики

Таблица С.5

Что	Как найти
Среднее арифметическое	сложить все значения и разделить на их количество
Размах	наибольшее значение минус наименьшее
Сколько раз выше порога	пересчитать значения больше порога (или «не меньше» – смотри, что в условии)

Пример. Температура за пять замеров: 18, 21, 24, 27, 30 °С. Среднее $(18 + 21 + 24 + 27 + 30):5 = 120:5 = 24$ °С, размах $30 - 18 = 12$ °С.

Как читать график:

- Сначала подписи осей и единицы: что отложено по горизонтали (обычно время), что по вертикали.
- Потом цена деления: сколько единиц в одной клетке. Посчитай по двум подписанным делениям, не угадывай.
- Растёт линия – величина увеличивается, горизонтальная – не меняется, круче наклон – быстрее меняется.
- Норма на графике – горизонтальная полоса или линия. Время, когда величина вне нормы, – участки графика за её пределами.

Вода и полив

Таблица С.6

Что	Формула	Пример
Объём бака-бруска	$V = a \cdot b \cdot c$	бак 60 × 40 × 50 см: 120 000 см ³ = 120 л
Объём бака-цилиндра	$V = \pi r^2 h$	радиус 20 см, высота 50 см: ≈ 62 832 см ³ ≈ 62,8 л
Сколько воды прошло	объём = расход × время	2,5 л/мин за 12 мин: 30 л
Сколько времени лить	время = объём : расход	120 л при 2,5 л/мин: 48 мин
Масса воды	1 л весит 1 кг	120 л – это 120 кг

Давление – это сила, делённая на площадь, на которую она давит:

$$p = \frac{F}{S},$$

где p – давление в паскалях (Па), F – сила в ньютонах (Н), S – площадь в м². Сила 100 Н на площади 0,5 м² даёт давление 100 : 0,5 = 200 Па. Одна и та же сила на маленькой площади давит сильнее: поэтому иголка колет, а лыжи не проваливаются в сугроб.

Для любопытных: давление столба воды. Чем выше бак над грядкой, тем сильнее вода давит вниз:

$$p = \rho \cdot g \cdot h,$$

где ρ – плотность воды (1000 кг/м³), $g = 9,8$ Н/кг, h – высота столба воды в метрах. Бак на 2 м выше крана: 1000 · 9,8 · 2 = 19 600 Па, почти 20 кПа.

Испарение и конденсация. Вода испаряется быстрее, когда теплее, суше и ветренее. Пар снова становится каплями на холодном: на стекле теплицы холодной ночью, на крышке кастрюли, на бутылке из холодильника.

Электричество

Закон Ома: ток тем больше, чем больше напряжение, и тем меньше, чем больше сопротивление:

$$I = \frac{U}{R}, \quad U = I \cdot R, \quad R = \frac{U}{I},$$

где I – ток в амперах (А), U – напряжение в вольтах (В), R – сопротивление в омах (Ом). 5 В на резисторе 220 Ом: $I = 5 : 220 \approx 0,0227 \text{ А} \approx 22,7 \text{ мА}$. Перед делением переведи мА в А, а кОм в Ом.

Резистор для светодиода. Из 5 В около 2 В берёт себе светодиод, остальное должно «упасть» на резисторе. Ток нужен не больше $20 \text{ мА} = 0,02 \text{ А}$:

$$R = \frac{5 - 2}{0,02} = 150 \text{ Ом.}$$

Если такого номинала нет, берут ближайший **большой**: 220 Ом. Ток станет чуть меньше, светодиод будет гореть чуть тусклее, но не сгорит.

Таблица С.7

Соединение	Сопротивление	Ток и напряжение
Последовательно (одно за другим)	$R = R_1 + R_2 + \dots$	ток везде один, напряжения складываются
Параллельно (рядом, «лесенкой»)	$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \dots$	напряжение одно, токи складываются

Три резистора по 1 кОм последовательно – 3 кОм, параллельно – около 333 Ом, меньше, чем один. Две одинаковые кнопки последовательно работают как «И» (свет горит, только если нажаты обе), параллельно – как «ИЛИ» (хватит любой).

Мощность и энергия:

$$P = U \cdot I, \quad E = P \cdot t.$$

Солнечная панель даёт 6 В и 0,3 А: мощность $6 \cdot 0,3 = 1,8 \text{ Вт}$. За 5 часов она выработает $1,8 \cdot 5 = 9 \text{ Вт} \cdot \text{ч}$.

На сколько хватит аккумулятора: время = ёмкость : ток. Аккумулятор 2000 мА·ч, устройство берёт 250 мА: $2000 : 250 = 8 \text{ ч}$. Единицы должны совпасть: мА·ч делим на мА.

Датчик → число → градусы

Датчик превращает свет, тепло или влажность в напряжение. Arduino измеряет напряжение на аналоговом входе и выдаёт число N от 0 до 1023 – показание `analogRead`. Это ещё не градусы и не проценты, а номер ступеньки.

Таблица С.8

Что	Формула	Пример
Число → напряжение	$U = N \cdot \frac{5}{1023}$	$N = 512: U \approx 2,50 \text{ В}$
Напряжение → число	$N = \frac{U}{5} \cdot 1023$, дробную часть отбросить	$U = 0,75 \text{ В}: 153,45$, <code>analogRead</code> даст 153
TMP36: напряжение → температура	$t = (U - 0,5) \cdot 100$	$U = 0,75 \text{ В}: t = 25 \text{ °C}$
LM35: напряжение → температура	$t = U \cdot 100$	$U = 0,25 \text{ В}: t = 25 \text{ °C}$

Словами: у обоих термометров 10 мВ на каждый градус, но TMP36 при 0 °C выдаёт уже 0,5 В – так он может показывать и мороз. В некоторых примерах делят не на 1023, а на 1024 – разница в сотых долях градуса, решай так, как в условии.

Проверка на числах из Tinkercad: при 25 °C на термометре `analogRead` даёт 153, а программа по формулам выше показывает $(153 \cdot 5 : 1023 - 0,5) \cdot 100 \approx 24,8$ °C. Небольшая неточность – от того, что число целое.

Фоторезистор меняет сопротивление от света. Чтобы Arduino его «увидел», его ставят в делитель напряжения вместе с обычным резистором (часто 10 кОм), а середину соединяют с аналоговым входом. Зависимость показаний от освещённости неравномерная: на одном краю число прыгает сотнями, на другом едва меняется. В люксы его переводить не обязательно: пороги для теплицы подбирают прямо в числах `analogRead`.

Калибровка датчика влажности почвы. Замеряем показание в сухой почве $N_{\text{сух}}$ и в мокрой $N_{\text{мокр}}$, а дальше считаем долю пути от «сухо» до «мокро»:

$$\text{влажность в процентах} = \frac{N - N_{\text{сух}}}{N_{\text{мокр}} - N_{\text{сух}}} \cdot 100.$$

Сухо – 180, мокро – 820, сейчас – 500: $\frac{500 - 180}{820 - 180} \cdot 100 = \frac{320}{640} \cdot 100 = 50\%$. Формула работает и у датчиков, где мокрой почве соответствует меньшее число.

В Arduino то же самое делает `map(N, 180, 820, 0, 100)`. Учти: `map` считает в целых числах и **отбрасывает** дробную часть. `map(300, 0, 1023, 0, 100)` даёт 29, а не 29,3 и не 30.

Автоматика: пороги и ШИМ

Порог. Сравнили показание с числом – приняли решение. Внимательно со знаком: при пороге 28 °C запись `t > 28` при 28 градусах на термометре ничего не включит, а `t ≥ 28` – включит.

Два порога (гистерезис). Если форточку открывать при 28 °C и закрывать тоже при 28 °C, то около 28 она будет хлопать каждую секунду. Поэтому берут два порога: открыть при 28 °C и выше, закрыть только при 24 °C и ниже. Между ними ничего не меняем – форточка остаётся как была.

Цепочка «если – иначе если – иначе» проверяется сверху вниз, и срабатывает только первая верная ветка:

Таблица С.9

Условие	Цвет индикатора
$t < 20$	синий: холодно
иначе, если $t < 25$	зелёный: норма (здесь уже точно $20 \leq t < 25$)
иначе	красный: жарко ($t \geq 25$)

ШИМ. `analogWrite` не даёт «половину напряжения», а быстро-быстро включает и выключает вывод. Доля времени включения задаёт мощность:

$$\text{мощность в процентах} = \frac{\text{значение}}{255} \cdot 100, \quad \text{значение} = \frac{\text{мощность в процентах}}{100} \cdot 255.$$

`analogWrite(9, 191)` – около 75 %. Для 40 % нужно $0,4 \cdot 255 = 102$.

Сигналы

Таблица С.10

Что	Формула или правило	Пример
Период T – время одного полного колебания	$T = \frac{1}{f}$	$f = 50$ Гц: $T = 0,02$ с = 20 мс
Частота f – сколько колебаний в секунду, в герцах	$f = \frac{1}{T}$	$T = 0,5$ с: $f = 2$ Гц
Амплитуда	наибольшее отклонение от среднего уровня	сигнал от 1 до 4 В: амплитуда 1,5 В
Сколько разных значений у n битов	2^n	10 бит – 1024 (analogRead), 8 бит – 256 (ШИМ)
Детектирование по уровню	событие – когда сигнал перешёл порог (направление и строгость порога задаёт условие)	считаем, сколько раз показание стало выше 600

Сглаживание скользящим средним. Каждую точку заменяем средним её и соседних по окну. Показания 20, 21, 35, 22, 23 с окном 3: $(20 + 21 + 35) : 3 \approx 25,3$; $(21 + 35 + 22) : 3 = 26$; $(35 + 22 + 23) : 3 \approx 26,7$. Случайный скачок 35 превратился в плавный горбик. Так работает простейший фильтр низких частот: быстрые рывки гасит, медленные изменения пропускает.

Метеостанция и солнце

Анемометр – вертушка с чашечками. Датчик считает её обороты. Одна чашечка за оборот проходит длину окружности $2\pi r$, поэтому её скорость:

$$v = 2\pi r \cdot n,$$

где r – радиус вертушки в метрах, n – оборотов в секунду. $r = 0,1$ м, 2 оборота в секунду: $v = 2 \cdot 3,14 \cdot 0,1 \cdot 2 \approx 1,26$ м/с. Настоящий ветер дует чуть быстрее чашек, поэтому у реальных приборов есть поправочный коэффициент – если он дан в условии, умножь на него.

Солнечный трекер поворачивает панель туда, где светлее. Два фоторезистора смотрят чуть влево и чуть вправо: левый показывает больше света – повернуть панель влево, правый – вправо, почти поровну – стоять.

Роза ветров – рисунок, на котором длина луча по каждому направлению показывает, сколько дней (или замеров) ветер дул оттуда. Самый длинный луч – самый частый ветер.

Мини-шпаргалки по инструментам

Arduino: язык C++

Программа для Arduino (скетч) состоит из двух частей: `setup()` выполняется один раз при включении, `loop()` – повторяется по кругу, пока есть питание. После `//` идёт комментарий: плата его не читает.

Таблица С.11

Команда	Что делает
<code>pinMode(9, OUTPUT);</code>	вывод 9 будет выходом: к нему подключают светодиод, реле, драйвер мотора
<code>digitalWrite(9, HIGH);</code>	подать на вывод 9 напряжение 5 В; LOW – снять (0 В)
<code>digitalRead(2)</code>	прочитать кнопку на выводе 2: HIGH или LOW
<code>analogRead(A0)</code>	измерить напряжение на A0: число от 0 до 1023
<code>analogWrite(9, 128)</code>	ШИМ на выводе 9: от 0 до 255
<code>delay(1000);</code>	ждать 1000 мс, то есть 1 секунду
<code>millis()</code>	сколько миллисекунд прошло с включения платы
<code>Serial.begin(9600);</code>	включить связь с компьютером (пишется в setup)
<code>Serial.println(t);</code>	вывести значение в монитор порта и перейти на новую строку
<code>map(x, 0, 1023, 0, 100)</code>	пересчитать x из одного диапазона в другой, дробную часть отбрасывает
<code>constrain(x, 0, 100)</code>	не выпустить x за пределы от 0 до 100

ШИМ на Arduino Uno умеют только выводы 3, 5, 6, 9, 10 и 11 – на плате рядом с ними значок ~.

Переменные: `int` – целое число, `float` – дробное (пишется через точку: `0.5`), `bool` – «да/нет» (`true` / `false`). Сравнения: `=` равно, `≠` не равно, `<`, `>`, `≤`, `≥`. Связки: `&&` – «и», `||` – «или», `!` – «не».

Пример целиком: термометр TMP36 на A0 управляет вентилятором на выводе 9 с двумя порогами. Мы собрали и проверили эту программу.

```

const int SENSOR = A0;    // термометр TMP36 – на аналоговом входе A0
const int FAN = 9;        // вентилятор через драйвер – на выводе 9 (ШИМ)
bool fanOn = false;       // запоминаем, включён ли вентилятор

void setup() {
  pinMode(FAN, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int n = analogRead(SENSOR);    // число от 0 до 1023
  float u = n * 5.0 / 1023;      // переводим в вольты
  float t = (u - 0.5) * 100;     // вольты → градусы для TMP36
  Serial.println(t);

  if (t ≥ 28) fanOn = true;      // жарко – включить
  if (t ≤ 24) fanOn = false;    // остыло – выключить
  if (fanOn) {
    analogWrite(FAN, 191);      // примерно 75 % мощности
  } else {
    analogWrite(FAN, 0);
  }
  delay(10);                    // пауза для симулятора
}

```

Сервомотор поворачивается на заданный угол от 0 до 180 градусов – так открывают форточку:

```

#include <Servo.h>          // библиотека для сервомоторов
Servo window;              // сервомотор форточки

void setup() {
  window.attach(6);        // сигнальный провод – на вывод 6
}

void loop() {
  window.write(90);        // открыть: повернуть на 90 градусов
  delay(2000);
  window.write(0);         // закрыть
  delay(2000);
}

```

Совет. Ошибки, на которых спотыкаются чаще всего: забыли точку с запятой в конце команды; открыли фигурную скобку и не закрыли; в сравнении написали `=` вместо `==`; дробное число записали через запятую (0,5 вместо 0.5); посчитали `n * 5 / 1023` в целых числах и получили 0 – пиши `5.0`; забыли `pinMode(..., OUTPUT)`.

Tinkercad: что где

Так было в сентябре 2026 года, на вебинарах сферы. Интерфейс может обновиться – смысл останется.

- **Вход.** Ссылка на класс «Агротех» и инструкция – в курсе сферы, в модуле «Теория и практика по созданию инженерных агросистем». Общая учётная запись работает как одна сессия: проекты **не сохраняются**. Нужный код копируй в файл на компьютере, схему зарисуй в тетрадь.
- **Новая схема:** «Создать» → «Цепи» (Circuits). В панели компонентов справа выбери показ **всех** компонентов – они разложены по разделам: общие (резисторы), входы (датчики), выходы (светодиоды, моторы, дисплеи), питание, макетные платы, микроконтроллеры, приборы (мультиметр).
- **Провода** тянутся мышкой от контакта к контакту. Двойной щелчок по проводу – точка изгиба. Минус принято делать чёрным, плюс – красным: цвет выбирается в панели сверху.
- **Настройки детали:** щёлкни по ней – откроется окошко, где можно задать номинал резистора, цвет светодиода, число батареек. Там же справка по детали с примером подключения.
- **Макетная плата:** наведи мышь на отверстие – зелёным подсветятся все отверстия, соединённые с ним. Длинные ряды по краям – шины питания, короткие столбики по пять отверстий – для деталей.
- **Код:** кнопка «Код» открывает редактор. Сначала он показывает блоки – переключи на текст, чтобы писать на C++. Внизу окна кода – монитор последовательного порта, там же кнопка графика.
- **Запуск:** «Начать моделирование». Пока моделирование идёт, схему менять нельзя – как и в жизни, сначала выключи питание. Щёлкни во время моделирования по термометру, фоторезистору или датчику влажности – появится ползунок, им меняют температуру, свет и влажность.
- **Кнопка** остаётся нажатой, если щёлкнуть по ней с зажатой клавишей Shift – так можно нажать две кнопки сразу.
- **Предупреждение о токе** («допустимый ток 20 мА») значит, что в жизни деталь сгорела бы: не хватает резистора.
- В цикл `loop()` добавляй `delay(10)` – симулятору так легче. Если после ошибки в коде «Начать моделирование» не срабатывает, обнови страницу.
- **Детали для теплицы**, которые есть в Tinkercad: фоторезистор, датчик влажности почвы, термометр TMP36, газоанализатор, сервомотор, мотор постоянного тока, LCD-дисплей 16×2, RGB-светодиод, потенциометр, тактовая кнопка.

Машина состояний и ПРИМС

ПРИМС – «программирование расширенных иерархических машин состояний». Программу не пишут строчками, а рисуют схемой в Кибериаде IDE: из какого режима в какой переходит устройство и что при этом делает. По схеме редактор умеет сгенерировать код на C++.

Таблица С.12

Элемент	Что это	Пример для теплицы
Состояние	режим работы устройства	«форточка закрыта», «проветриваем»

Элемент	Что это	Пример для теплицы
Событие	то, что случилось и заставило реагировать	прошло 5 секунд, нажата кнопка, пришло новое показание
Условие	фильтр: переход сработает, только если оно верно	температура не ниже 28 °С
Действие	что устройство делает	открыть форточку, включить красный светодиод
Переход	смена состояния, рисуется стрелкой	«закрыта» → «проветриваем»

Подпись на стрелке удобно читать так: **событие [условие] / действие**. Например: «новое показание [$t \geq 28$] / открыть форточку».

- В каждый момент устройство находится **ровно в одном** состоянии.
- Действие на **входе** в состояние выполняется один раз, когда в него попали; на **выходе** – один раз, когда уходят.
- Если условие ложно, перехода нет: устройство остаётся, где было.
- Самый первый пример на курсе – мигание светодиодом: два состояния «горит» и «не горит», переходы между ними по времени. Светофор – три-четыре состояния по кругу.

Как проследить работу машины: заведи таблицу «шаг – текущее состояние – событие – условие верно? – новое состояние – действие» и заполняй строку за строкой.

Таблицы

Формулы в русской версии Excel, LibreOffice Calc и онлайн-таблиц начинаются со знака **=**. Аргументы разделяют точкой с запятой.

Таблица С.13

Формула	Что считает
=СРЗНАЧ(В2:В25)	среднее по клеткам от В2 до В25
=МАКС(В2:В25) , =МИН(В2:В25)	наибольшее и наименьшее
=СУММ(В2:В25)	сумму
=СЧЁТЕСЛИ(В2:В25; ">25")	сколько значений больше 25
=ОКРУГЛ(С2;1)	округлить до десятых
=(В2*5/1023-0,5)*100	перевести показание TMP36 из клетки В2 в градусы

Формулу из одной клетки можно протянуть вниз за уголок – адреса клеток сдвинутся сами. График: выдели столбец времени и столбец значений, затем меню «Вставка» → диаграмма, тип «график» (линия).

Словарик

Таблица С.14

Слово	Что значит
Агротехнологии	способы выращивать растения и животных с помощью техники, датчиков и автоматики
Актуатор (исполнительное устройство)	деталь, которая что-то делает: мотор, насос, вентилятор, лампа, сервомотор
Аналоговый сигнал	сигнал, который меняется плавно и может принимать любые значения: напряжение с термометра
Анемометр	прибор для измерения скорости ветра, вертушка с чашечками
АЦП (аналого-цифровой преобразователь)	часть Arduino, которая превращает напряжение в число от 0 до 1023
Биоритмы	повторяющиеся изменения в жизни растения: днём фотосинтез, ночью только дыхание
Гистерезис	два разных порога для включения и выключения, чтобы устройство не щёлкало туда-сюда
Датчик (сенсор)	деталь, которая превращает свет, тепло, влажность в электрический сигнал
Делитель напряжения	два резистора подряд; напряжение между ними зависит от их сопротивлений
Детектирование	обнаружение нужного события в сигнале, например перехода через порог
Дыхание растения	получение энергии из сахара с поглощением кислорода и выделением углекислого газа
Калибровка	настройка датчика по известным точкам: «вот так он показывает сухо, а вот так – мокро»
Кодирование	запись сообщения условными знаками: азбукой Морзе, нулями и единицами
Конденсация	превращение пара в капли воды на холодной поверхности
Макетная плата	пластиковая плата с отверстиями, соединёнными внутри, чтобы собирать схемы без пайки
Микроклимат	температура, влажность, освещённость и состав воздуха в теплице
Микроконтроллер	маленький компьютер на одной микросхеме; Arduino Uno – плата с микроконтроллером

Слово	Что значит
Модуляция	способ «посадить» сообщение на сигнал-переносчик (несущую), меняя его силу или частоту
Монитор порта	окно, куда Arduino выводит числа и текст; рядом – график этих чисел
Огибающая	плавная линия, которая обводит верхушки быстро меняющегося сигнала
Относительная влажность	насколько воздух насыщен паром, в процентах; при 100 % выпадает роса
Порог	число, при переходе через которое система принимает решение
Потенциометр	резистор с ручкой: крутишь – меняется напряжение; им задают пороги
ПРИМС	визуальный язык программирования машинами состояний; редактор – Кибериада IDE
Сервомотор	мотор, который поворачивается на заданный угол от 0 до 180 градусов
Сити-ферма	ферма в городе: на крыше, в здании, в контейнерах
Скетч	программа для Arduino
Спектр сигнала	из каких частот, медленных и быстрых, состоит сигнал
Фильтр низких частот	обработка, которая пропускает медленные изменения и гасит быстрые скачки
Фоторезистор	резистор, сопротивление которого зависит от освещённости
Фотосинтез	образование сахара из углекислого газа и воды в зелёных листьях на свету
Цифровой сигнал	сигнал со строго определёнными значениями, 0 и 1: есть напряжение или нет; помеха его не портит понемногу – он доходит целиком или пропадает
ШИМ	быстрое включение и выключение вывода, чтобы плавно менять яркость или скорость
DHT11	датчик, который измеряет температуру и влажность воздуха и передаёт их цифрой
LCD 1602	дисплей на 2 строки по 16 символов
ppm	миллионная доля: 400 ppm углекислого газа – это 400 частиц из миллиона, 0,04 %

Шпаргалка: всё «Запомни» из глав

Глава 1. Растение как живая система

Запомни. Факторов роста пять, и ни один не заменяет другой. Урожай держит на привязи тот, которого не хватает сильнее всего, – это закон минимума. У каждой культуры свои нормы: огурцу подавай тепло и сырость, томату – посуше, салату – прохладу.

Запомни. Фотосинтез идёт в листьях на свету: углекислый газ + вода → сахар + кислород. Хлорофилл берёт в основном красный и синий свет, зелёный отражает. После насыщения лишний свет почти ничего не прибавляет.

Запомни. Дыхание: сахар + кислород → углекислый газ + вода + энергия. Растение дышит всегда. Днём фотосинтез сильнее дыхания, ночью фотосинтеза нет. В закрытой теплице CO₂ растёт ночью и падает днём.

Запомни. Корни всасывают воду с минеральными солями через корневые волоски. N – листья, P – корни и цветы, K – плоды. Больше 90 % воды испаряется через устьица. Корням нужен кислород, поэтому перелив опасен так же, как засуха.

Запомни. Устьица у большинства растений открыты на свету и закрыты в темноте. Длина дня подсказывает растению время года. Длина дня = закат – восход, в часе 60 минут, а не 100.

Запомни. Микроклимат – погода внутри теплицы. У каждой его величины свой датчик и своя единица: градусы, проценты, люксы, ppm. Автоматике нужны числа, «на глаз» она не умеет.

Запомни. Относительная влажность – сколько пара есть, в процентах от того, сколько может поместиться. Остывая, воздух «сжимает стакан»: проценты растут, хотя пара не прибавилось. Остыл ниже точки росы – выпадает роса.

Глава 2. Микроклимат теплицы: считаем

Запомни. Температура – в °C, влажность воздуха и почвы – в %, свет – в люксах, углекислый газ – в ppm. Число без единицы – не ответ. Нормы и пороги бери из условия задачи.

Запомни. $1\% = 10\,000\text{ ppm}$, $1\text{ клк} = 1000\text{ лк}$, $1\text{ с} = 1000\text{ мс}$. В часе 60 минут, поэтому 1,5 ч – это 1 ч 30 мин.

Запомни. $\varphi = a : a_{\max} \cdot 100\%$, и обратно $a = a_{\max} \cdot \varphi : 100$. Воздух остывает – влажность растёт. Разность процентов – это процентные пункты; «на сколько процентов» – всегда от прежнего значения.

Запомни. Освещённость – в люксах; от лунной ночи до солнечного полдня она растёт в сотни тысяч раз. Отрезок через полночь – это две части: до 24:00 и после 0:00. Доля суток = часы : 24 · 100 %.

Запомни. Холоднее всего под утро, жарче всего после полудня. Доля замеров = число нужных замеров : число всех · 100 %. Время вне нормы = число таких замеров · промежуток между замерами.

Запомни. Проверь знак на границе ($>$ или $\geq$) и единицы. «И» – нужны обе части, «или» – хватит одной. С двумя порогами между ними устройство остаётся как было, поэтому иди по замерам по порядку.

Глава 3. Графики и таблицы агронома

Запомни. Порядок чтения: оси и единицы → цена деления → точки → форма линии. Точки – настоящие замеры, линия между ними – оценка. Круче участок – быстрее меняется величина.

Запомни. Среднее = сумма : количество. Размах = максимум – минимум. Среднее всегда лежит между минимумом и максимумом. С минусами считай внимательно: от -3 до $+13$ размах 16 градусов, а вовсе не 10.

Запомни. Строчка таблицы – точка графика. Масштаб выбирай по самому большому числу. Равное время – равные отрезки по оси. Непрерывно меняется – рисуй линию, отдельными порциями – столбики.

Запомни. Формула начинается с $=$. В диапазоне B2:B9 восемь клеток – обе крайние считаются. При протягивании вниз адреса в формуле сдвигаются сами. ">25" саму 25 не считает, " ≥ 25 " – считает.

Запомни. Порог на графике – горизонтальная линия. Момент, когда её пересекли, ищут по двум соседним замерам: сначала «сколько градусов в час», потом «сколько часов до порога». Доли часа переводят в минуты умножением на 60.

Запомни. Два графика рядом – сначала сравни оси, потом линии. Среднее за сутки не скажет, была ли жара в полдень: для решений в теплице смотри на крайние значения.

Глава 4. Вода и полив

Запомни. Вода на грядку = норма $\times$ число кустов (или $\times$ площадь) $\times$ число дней. Запас – округляем вверх, «сколько полных дней хватит» – вниз.

Запомни. $1 \text{ л} = 1 \text{ дм}^3 = 1000 \text{ см}^3 = 1000 \text{ мл}$; $1 \text{ м}^3 = 1000 \text{ л}$; $1 \text{ л воды} = 1 \text{ кг}$. Считай объём в дециметрах – сразу получишь литры. $1 \text{ мм осадков или полива} = 1 \text{ л на каждый квадратный метр}$.

Запомни. Объём = производительность $\times$ время; время = объём : производительность. Перед делением приведи л/ч и минуты (или мл и литры) к одним единицам. Вода и приходит, и уходит – считай разницу.

Запомни. Вылить = нужно корням : доля. Процент экономии – от старого расхода. «Стало меньше на 20 %» = «стало 80 % от прежнего».

Запомни. $p = F : S$, давление в паскалях. Вода: каждый метр глубины – плюс 9,8 кПа, $p = 9800 \cdot h \text{ Па}$, где h в метрах от поверхности воды. Ширина бака и число литров на давление не влияют.

Запомни. Испарение ускоряют тепло, сухой воздух, ветер и большая открытая поверхность; испаряясь, вода охлаждает. Конденсация идёт на холодном: пар из тёплого воздуха оседает каплями на холодном стекле. $1 \text{ г воды} = 1 \text{ мл}$.

Глава 5. Электричество, энергия и датчики

Запомни. Ток – поток зарядов, амперы. Напряжение – «толкание», вольты. Сопротивление – помеха току, омы. Ток идёт только по замкнутой цепи. $1\text{ A} = 1000\text{ mA}$, $1\text{ k}\Omega = 1000\text{ }\Omega$.

Запомни. $I = U/R$ – в вольтах, амперах и омах. Резистор для светодиода: $R = (5 - U_{\text{св}})/I$, номинал берём ближайший больший. Последовательно – сопротивления складываются, параллельно – общее меньше самого маленького (n одинаковых – в n раз меньше одного).

Запомни. Датчик превращает свет, тепло, влажность в напряжение, Arduino – напряжение в число 0...1023. $U = N \cdot 5/1023$; обратно $N = U/5 \cdot 1023$ с отброшенной дробью. ТМР36: $t = (U - 0,5) \cdot 100$. Почву калибруют по двум точкам: «сухо» и «мокро».

Запомни. $P = U \cdot I$ (ватты = вольты · амперы). $E = P \cdot t$ (Вт · ч = Вт·ч). Время работы аккумулятора = ёмкость : ток. КПД – доля полезной энергии, у солнечной панели около 15–20 %.

Запомни. Тепло идёт от тёплого к холодному: сквозь стенки, с воздухом, лучами. Потери = $k \cdot$ разность температур. Чтобы держать температуру, обогреватель должен покрывать потери. Мороз вычитай со знаком: между $+10\text{ }^{\circ}\text{C}$ и $-10\text{ }^{\circ}\text{C}$ двадцать градусов, а не ноль.

Глава 6. Программирование умной теплицы

Запомни. Алгоритм – точные шаги, исполнитель – тот, кто их выполняет без догадок. Из трёх кирпичиков (следование, ветвление, цикл) собирается любая программа. Что делает исполнитель, узнают только одним способом: проходят алгоритм шаг за шагом.

Запомни. `=` кладёт значение в переменную, `=` сравнивает. Целое, делённое на целое, теряет дробную часть. В цепочке `if ... else if` срабатывает только первая верная ветка. Что делает цикл, проверяй таблицей трассировки, а не в уме.

Запомни. `setup()` – один раз, `loop()` – по кругу. Схема любой автоматики: измерить (`analogRead`), сравнить с порогом (`if`), включить или выключить (`digitalWrite`). Время в `delay` – в миллисекундах: секунды умножь на 1000.

Запомни. Два порога = переменная-память + два условия: «выше верхнего – включить», «ниже нижнего – выключить». Если ни одно не сработало, память не трогаем. Голый `else` превращает два порога обратно в один.

Запомни. Машина состояний = состояния + переходы «событие [условие] / действие». Номер состояния хранят в одной переменной, переходы проверяет цепочка `if ... else if`. Вместо долгого `delay` – вопрос «сколько я уже в этом состоянии?»: `millis() - start`. За один замер – один переход. Из каждого состояния, кроме конечного, должна выходить хотя бы одна стрелка. Две стрелки с одним событием из одного состояния допустимы, только если условия у них не пересекаются.

Глава 7. Сигналы и обработка данных

Запомни. Сигнал – то, что меняется по делу, шум – мелкая беспорядочная дрожь поверх него. У волны период T – время одного повтора, частота $f = 1:T$ (в герцах – повторов в секунду), амплитуда – половина размаха.

Запомни. Отсчётов на один больше, чем промежутков, если считаешь с обоих концов. Время по отсчётам = число отсчётов · шаг. Два отсчёта на период и меньше – колебание может пропасть из записи.

Запомни. Событие – момент, когда сигнал перешагнул порог снизу вверх, а не каждый отсчёт выше порога. Период = (время последнего события – время первого) : (число событий – 1).

Запомни. Скользящее среднее – среднее из k последних отсчётов, окно сдвигается на один. Из n отсчётов получается $n - k + 1$ средних. Широкое окно – глаже, но с опозданием.

Запомни. Выброс – одиночный прыжок, невозможный в жизни. Среднее он портит, медиана из трёх его убирает. Против ложных срабатываний – медианный фильтр или «несколько отсчётов подряд».

Запомни. Спектр показывает, из каких частот сложен сигнал. ФНЧ пропускает медленное и убирает быстрое; скользящее среднее – простой ФНЧ. Окно длиной в период помехи убирает её целиком.

Запомни. Двоичное число переводят, складывая веса разрядов с единицами: 128, 64, 32, 16, 8, 4, 2, 1. n битов дают 2^n вариантов. Нужно N значений – бери такое наименьшее n , что $2^n \geq N$.

Глава 8. Метеостанция и автоматика

Запомни. Метеостанция нужна теплице, чтобы автоматика успела среагировать до того, как погода придёт внутрь. DHT11 меряет температуру (0...50 °C, ± 2 °C) и влажность воздуха (20...90 %) и отдаёт готовые числа; спрашивать его – не чаще раза в 1–2 секунды.

Запомни. Скорость чашки: $v_{\text{ч}} = 2\pi r \cdot n$ (радиус в метрах, n – оборотов в секунду). Скорость ветра – это скорость чашки, умноженная на поправку k , если она дана. Обороты = импульсы : число магнитов. 1 м/с = 3,6 км/ч.

Запомни. Серва: угол от 0 до 180°, `write(угол)`, держит положение. DC-мотор: крутится от напряжения, скорость – через ШИМ (0...255), направление – сменой полярности, подключается только через драйвер. `map` отбрасывает дробную часть. Время смены воздуха = объём теплицы : производительность вентилятора.

Запомни. Панель берёт больше всего света, когда лучи падают на неё перпендикулярно; при отклонении на 60° – только половину. Высота солнца в полдень в равноденствие = 90° – широта, летом на 23° выше, зимой на 23° ниже. Наклон панели = 90° – высота солнца. Солнце смещается на 15° в час.

Запомни. Правила проверяют сверху вниз, правила безопасности (ветер, мороз) – первыми. Ограничение не выбирает угол, а только урезает слишком большой. Рассчитанный угол всегда сверяй с пределами механизма: от 0° до 90° у форточки, от 0° до 180° у сервы.

Где что найти

Таблица С.15

Что	Где
Правила олимпиады, сферы, сроки	junior.ntcontest.ru
Личный кабинет участника	my.ntcontest.ru
Платформа отбора	orbita.education (заходить кнопкой из личного кабинета)

Что	Где
Курс сферы «Агротех» (за тест курса +5 баллов к отбору)	academy.junior.online/event/agrotech_junior , вход через «Талант»
Класс «Агротех» в Tinkercad	ссылка и инструкция – в курсе сферы
Кибериада IDE (ПРИМС)	ссылка – в курсе сферы
Tinkercad	tinkercad.com
Запасной симулятор Arduino	wokwi.com
Arduino IDE	по желанию, для настоящей платы; ставит педагог кружка
Программа сферы и кружка	junior.online/agrotech → «Образовательная программа «Агротех»»
Новости и вопросы по сфере	канал «Сфера Агротех» в мессенджере MAX (находится поиском по слову «Агротех»)
Памятки для Новосибирской области	nto.kvantorium54.ru

Об авторе-составителе



Артём Осиновский

Методист и разработчик детского технопарка «Кванториум» (Новосибирск). Придумывает образовательные программы и сам делает для них цифровые инструменты: платформу смены «Миссия на Луну» для фестиваля «Технопром-2026», сайт с памятками НТО Юниоры для Новосибирской области. Эту книгу собрал для школьников области, чтобы к отбору можно было подготовиться самостоятельно – даже без кружка рядом с

домом.

Нашли ошибку – пишите на osinovsky.aa@kvantorium54.ru (рабочая почта): сфера, версия книги и страница. По другим вопросам и проектам – личная: artem081033@gmail.com.