



НТО ЮНИОРЫ 2026/27 · СФЕРА

Медтех

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

Медтех

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

Автор-составитель – **Артём Осинковский**, 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 «Медтех» 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 Осинковский, 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	7
Что понадобится	7
План на месяц до отбора	27
Родителям и наставникам	27
Честно об этой книге	27
Глава 1 Тело человека как система сигналов	29
1.1. Тело глазами инженера	29
1.2. Сердце и кровеносная система	31
1.3. Дыхательная система	33
1.4. Нервная система и нейрон	34
1.5. Мышцы и опорно-двигательный аппарат	36
1.6. Органы чувств	37
1.7. Электричество в живых организмах	38
1.8. Какой датчик что меряет	40
1.9. Разбираем задачи	41
Глава 2 Пульс и сердце: считаем по графику	51
2.1. Сердце и пульс	51
2.2. Период и частота: удары в минуту и секунды между ударами	53
2.3. Читаем пульсограмму и ЭКГ	54
2.4. Пульс при нагрузке: сравниваем	58
2.5. Как браслет видит пульс	60
2.6. Разбираем задачи	61
Глава 3 Мышцы и мозг: ЭМГ и ЭЭГ	69
3.1. ЭМГ: электричество работающей мышцы	70
3.2. Порог: как сигнал становится командой	72
3.3. Бионическая рука и мио-армрестлинг	75
3.4. ЭЭГ: ритмы мозга	78
3.5. Биообратная связь: человек смотрит на свой сигнал	80
3.6. Разбираем задачи	81
Глава 4 Сигналы и обработка данных	93
4.1. Сигнал и шум	93
4.2. Что меряют у сигнала: размах, период, частота	95
4.3. Аналоговый и цифровой сигнал. Дискретизация	96
4.4. Детектирование: по уровню и по частоте	98
4.5. Сглаживание: скользящее среднее	102
4.6. Спектр и фильтр – на пальцах	104

4.7. Кодирование и передача	105
4.8. Разбираем задачи	107
Глава 5 Электричество и датчики	116
5.1. Цепь, ток и напряжение	116
5.2. Сопротивление и закон Ома	118
5.3. Светодиод и резистор к нему	119
5.4. Датчик – переводчик на язык вольт	121
5.5. Делитель напряжения	123
5.6. Аналоговый вход: из вольт в число и обратно	125
5.7. Батарейки и время работы прибора	126
5.8. Разбираем задачи	128
Глава 6 Программирование: Arduino и машины состояний	136
6.1. Алгоритм и исполнитель	137
6.2. Переменные: коробочки с именами	138
6.3. Условие: развилка в программе	139
6.4. Цикл: повторить много раз	141
6.5. Скетч Arduino: setup, loop и монитор порта	142
6.6. Блоки и текст: один алгоритм – два языка	144
6.7. Машина состояний: покой, измерение, тревога	146
6.8. Разбираем задачи	149
Глава 7 Таблицы, графики и статистика	167
7.1. Электронная таблица: ячейки и формулы	168
7.2. Среднее, наибольшее, наименьшее	170
7.3. Десятичные дроби и округление	171
7.4. Проценты, ppm и пороги: когда проветривать	173
7.5. Графики: построить и прочитав	175
7.6. Сравниваем группы и выбираем верный вывод	177
7.7. Разбираем задачи	178
Глава 8 Свет, звук и колебания в медицине	187
8.1. Колебания: период, частота, амплитуда	188
8.2. Звук: колебания, которые бегут	190
8.3. Эхо: как УЗИ заглядывает внутрь	192
8.4. Свет: что видит оптический датчик	195
8.5. Сила и рычаг: как устроена рука	197
8.6. Сервопривод: угол, нить и импульсы	198
8.7. Разбираем задачи	200
Ответы и решения	208
Справочник	215
Числа, которые используются в книге	215
Перевод единиц	216
Формулы и правила	216
Мини-шпаргалки по инструментам	220
Словарик	225

Как устроена эта книга

Эта книга – тренажёр для отборочного этапа сферы «Медтех» олимпиады «НТО Юниоры». Здесь есть всё, чтобы готовиться самостоятельно: теория с нуля, подробные разборы задач и сразу после каждого – похожие задачи «Реши сам» с ответами в конце книги. Набор с датчиками и знакомый врач не нужны: пульс, мышцы и мозг мы изучаем по графикам, на бумаге, с калькулятором и в бесплатных программах, а схемы собираем в симуляторе прямо в браузере.

Для кого эта книга

- Для школьников **5–7 классов**, которые выбрали (или думают выбрать) сферу «Медтех».
- Для тех, кто **не проходил физику и ни разу не программировал**. Школьная физика начинается в 7 классе, так что ток, напряжение и частоту мы объясняем с самого начала, а первую программу разбираем построчно.
- Для тех, у кого **нет кружка и оборудования**. Первая половина курса сферы, которая готовит к отбору, целиком проходит в онлайн-симуляторе. Книга устроена так же.
- Для **наставников и родителей**, которые хотят помочь. Для них есть отдельный раздел в конце введения.

Как устроен отбор

Таблица В.1

Когда	Что происходит
с 1 октября	демо-этап с пробными заданиями (по сообщению организаторов)
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, который проверяет компьютер: ввести число, выбрать верные варианты, сопоставить, расставить строки программы по порядку, прочитать график и сделать вывод.
- «Медтех» – новая сфера, её отборочных задач прошлых лет не существует. Зато разработчики сферы на вебинаре в сентябре 2026 года рассказали, из чего будет состоять

отбор. Задачи связаны с материалами курса, но они более творческие, чем тесты курса: на физиологию, биосигналы и обработку сигналов. Плюс практические задачи в симуляторе, где нужно собрать схему или написать программу. Задачки будут по схемотехнике и по обоим языкам курса – диаграммам состояний и C++. Поэтому книга построена по программе сферы «Медтех. Основы мониторинга и биоуправления», а в самой программе есть отдельный модуль «Разбор задач отборочного онлайн этапа».

- 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. И бонус засчитают, только если за задания отбора ты набрал больше нуля.
- Тесты внутри курса – это ещё не отбор. Они показывают, понял ли ты модуль. Чтобы курс засчитали пройденным, на тестах нужно набрать не меньше 70 %.
- Финал – **командный** и очный: команда из трёх человек (бывает из двух или четырёх) в своём регионе решает задачи за компьютерами на той же «Орбите». Там снова теория, творческие задачи и работа в симуляторе. Тема финала этого сезона, по словам разработчиков, – БОС-тренажёр для космонавтов. БОС – биологическая обратная связь: прибор показывает человеку его собственный пульс, дыхание или напряжение мышц, огоньком или героем игры, и человек учится ими управлять.
- Оборудование для отбора и общего финала не нужно. Отдельно от финала некоторые регионы проводят хакатон с набором «Медтех Юниор» – там уже работают с настоящими датчиками пульса, мышц и мозга.

Совет. Сроки и правила могут уточняться. Главный источник – сайт олимпиады junior.ntcontest.ru и памятка для Новосибирской области nto.kvantorium54.ru.

Что пригодится на отборе и где это в книге

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

Таблица В.2

Что пригодится на отборе	Глава
Анатомия и физиология человека, нервная и дыхательная системы, органы чувств	1. Тело человека как система сигналов
Электричество в живых организмах	1 и 3
Сердце и кровеносная система, временные интервалы	2. Пульс и сердце: считаем по графику
Мышцы и опорно-двигательный аппарат, механизмы	3. Мышцы и мозг: ЭМГ и ЭЭГ
Частота и период колебаний, обработка сигналов (из программы сферы)	4. Сигналы и обработка данных
Электричество и электрические цепи	5. Электричество и датчики

Что пригодится на отборе	Глава
Алгоритмы, переменные, управляющие конструкции, любой язык программирования	6. Программирование: Arduino и машины состояний
Графики, шкалы и диапазоны, проценты, среднее, максимум и минимум, таблицы	7. Таблицы, графики и статистика
Оптика и свет, колебания и волны	8. Свет, звук и колебания в медицине

Начни с главы 1: из неё понятно, какие сигналы даёт тело и как их ловят датчики, а дальше книга всё время на это опирается. Главы 2 и 3 разбирают конкретные сигналы – сердце, мышцы, мозг. Глава 4 собирает общие приёмы для любых сигналов, и после второй и третьей глав она пойдёт легче. Главу 5 прочитай раньше главы 6: сначала схема, потом программа для неё. Главу 7 можно открыть когда угодно – графики и средние нужны во всей книге. Главу 8 лучше читать после главы 4.

Как пользоваться книгой

В каждой главе теория и разборы идут вперемешку: за каждым разбором сразу следуют 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 карточки «на пальцах», а самое нужное собрано в рамках «Запомни». Все «Запомни» из всех глав повторяются в конце книги, в шпаргалке.
2. **Разбираем задачи.** К каждой задаче – решение по шагам: что дано, что делаем и почему, где часто ошибаются, какой ответ получился и можно ли ему верить.
3. **Реши сам.** Сразу после разбора – две задачи на тот же приём, но с другими числами и другим сюжетом.
4. **Ответы** – в конце книги, в разделе «Ответы и решения». Там только ответ и иногда короткая подсказка, если задача с подвохом. Полного решения нет специально: если подсказки не хватило, вернись к разбору перед задачей.

Хочешь пробежаться по всей программе быстро – есть отдельная короткая книжка **«Быстрый старт»**: из каждой главы самое нужное, один разбор и несколько задач «Реши сам». Её удобно пройти за неделю, а эту книгу держать под рукой как справочник.

У каждой задачи есть уровень – как в обычном задачнике: у трудных задач звёздочка рядом с номером («задача 3*»), в них нужно несколько идей сразу. Лёгкие задачи, в одно действие, подписаны словом «разминка». Звёздочка и «разминка» – это уровни нашей книги, а не баллы олимпиады. Сколько задач будет на отборе и сколько баллов стоит каждая, заранее не объявляют.

Совет. Разбор читай с листком бумаги. Закрой решение, попробуй решить сам, а потом сверяйся по шагам. Так ты запомнишь в три раза больше, чем если просто прочитаешь.

У каждой задачи внизу написано, откуда она взялась. «Своя задача по программе сферы» – мы придумали её по теме из программы 2026/27 года; таких в книге большинство. «По мотивам задачи НТО» – идею мы взяли из задач старшей олимпиады для 8–11 классов по профилю

«Нейротехнологии и когнитивные науки», сильно упростили, пересказали своими словами и поменяли числа.

Как вводить ответы на отборе

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

- **Только число**, без единиц измерения. Смотри, в каких единицах просят ответ: пульс – в ударах в минуту, время – в секундах или миллисекундах, частота – в герцах.
- **Дробную часть** на сайте обычно отделяют **точкой**: 0.75 , а не $0,75$. В книге мы пишем по-русски, через запятую.
- **Несколько верных вариантов** часто записывают строкой из их номеров подряд, без пробелов и запятых: выбрал варианты 1, 3 и 4 – вводишь 134 . Точный формат – в инструкции к задаче.
- **Округляй так, как сказано в условии**. «До целых» – ни одного знака после точки. Промежуточные результаты не округляй – только ответ.
- **Пульс – целое число**. Сердце не делает 72,4 удара. Если условие не говорит, как округлять, а получилось дробное число, перечитай условие: возможно, время нужно было перевести в секунды или минуты.
- **Попытки ограничены**. На задачу даётся от 3 до 5 вводов, и каждая неверная попытка снимает 30 % балла за эту задачу. Сначала реши на бумаге или проверь программу в симуляторе, и только потом вводи ответ.
- **«Сохранить» – ещё не отправка**. Ответ засчитан, когда нажата кнопка «Отправить».

Что понадобится

- Тетрадь в клетку, карандаш, линейка – графики пульса и сигналов удобно разбирать прямо по клеткам.
- Калькулятор. Хватит обычного на телефоне: проценты, дроби, деление.
- Компьютер с интернетом – для глав 5 и 6 и для задач с таблицами. Планшета мало: программы из списка сферы ставятся на компьютер.
- **Tinkercad** – онлайн-симулятор схем, в котором работает курс сферы. Открывается в браузере, ставить ничего не нужно. Входи по ссылке на класс из курса, логин там же. Схемы в общем классе между заходами не сохраняются – копируй код программы себе в файл.
- **Wokwi** (wokwi.com) – другой симулятор Arduino, тоже в браузере и без установки. Им удобно проверять программы из книги, если Tinkercad сегодня не открывается.
- **Arduino IDE** – программа для настоящей платы Arduino. Для отбора она не обязательна: код можно писать прямо в симуляторе. Но она есть в списке программ, которые организаторы называли для финала. Для подготовки к отбору хватит симулятора; программу для настоящей платы поможет поставить педагог кружка.
- **Кибериада IDE** – программа, где алгоритм рисуют схемой из состояний и стрелок (язык ПРИМС, глава 6). Где её скачать и как поставить, рассказывают в курсе сферы.
- **Табличный редактор** – любой: LibreOffice Calc, Excel, Google Таблицы. Он тоже в списке программ для финала.

- **Python** – по желанию: любой онлайн-интерпретатор, ничего ставить не нужно. Им удобно проверять расчёты по графикам и спискам чисел.

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/UKFECSX7N (класс курса «Медтех»; та же ссылка – в курсе сферы, модуль 5 «Теория и практика по созданию систем мониторинга и биоуправления. 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).

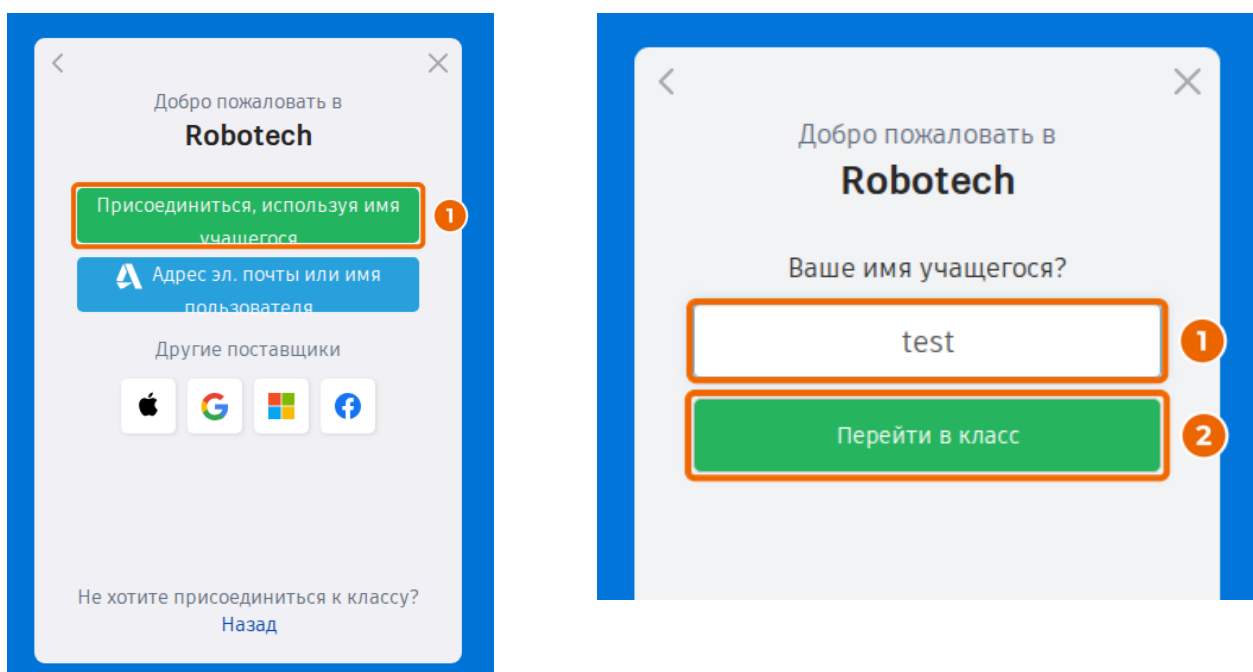


Рисунок В.1 – Вход в класс Tinkercad: слева – выбор способа входа, справа – имя учащегося

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

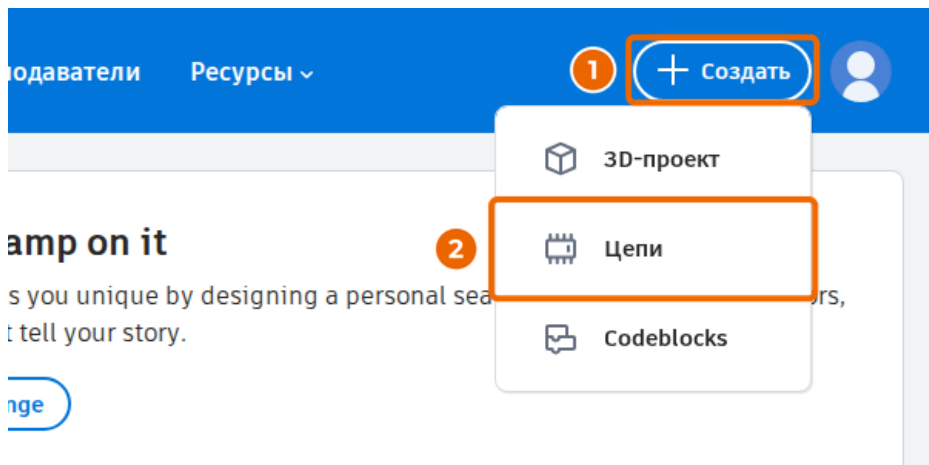


Рисунок В.2 – Новая схема: «Создать» → «Цепи»

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

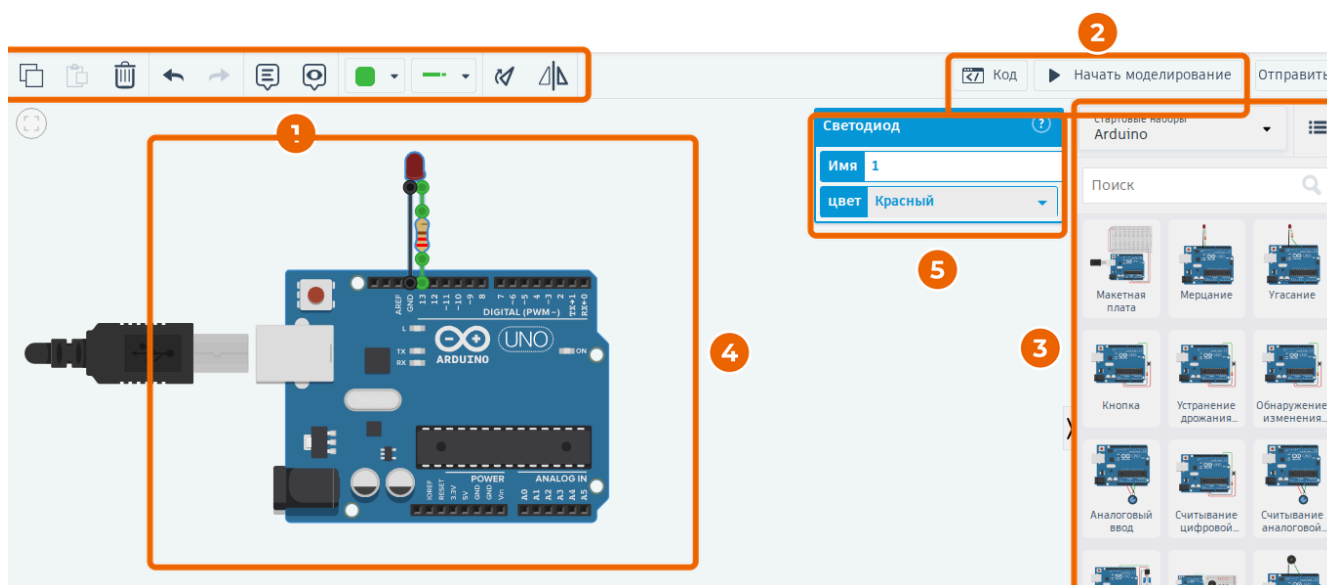


Рисунок В.3 – Редактор схем Tinkercad

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.

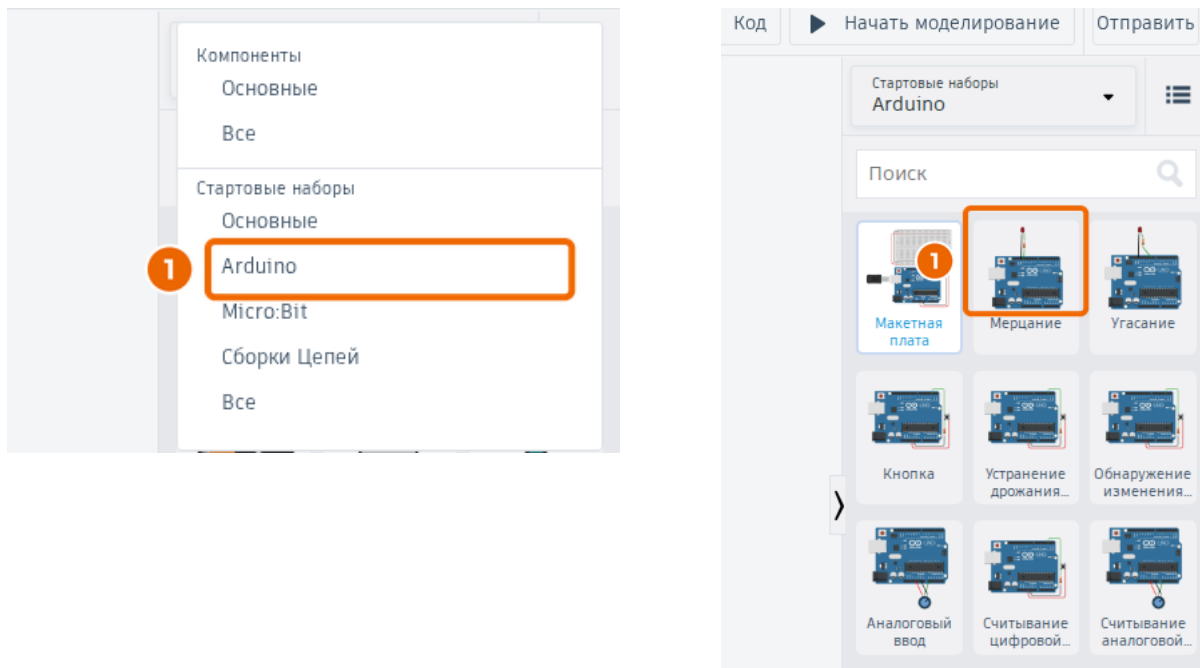


Рисунок В.4 – Стартовые наборы: слева – выбор набора «Arduino», справа – схема «Мерцание»

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.

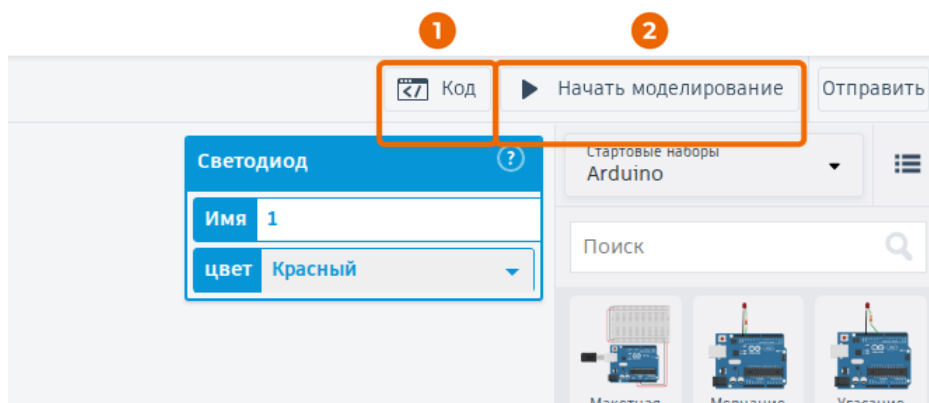


Рисунок В.5 – Кнопки «Код» и «Начать моделирование»

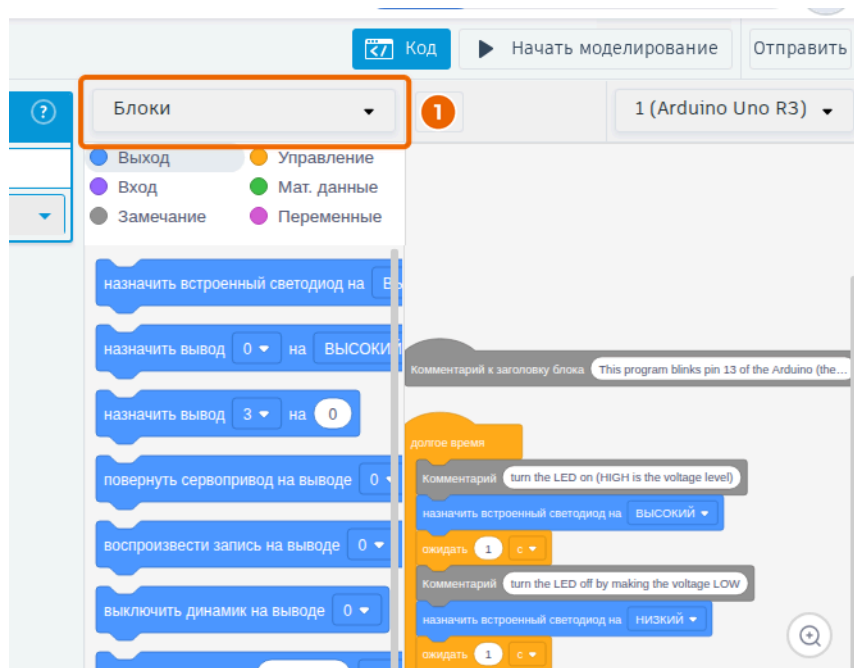


Рисунок В.6 – Программа «Мерцания» блоками

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

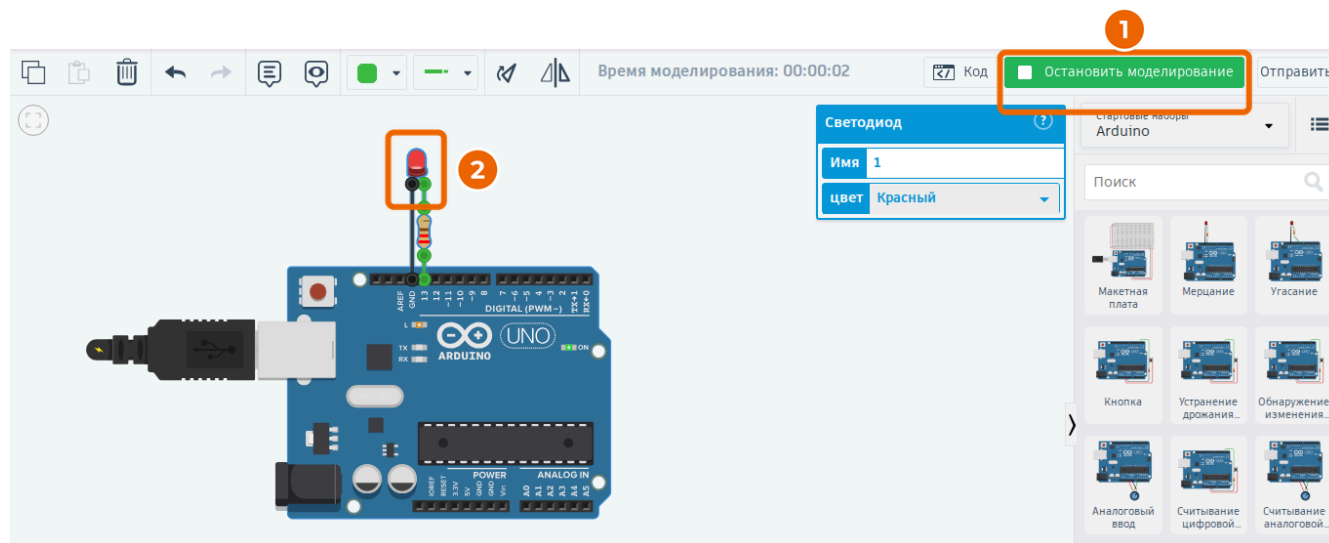


Рисунок В.7 – Схема работает

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/cyberiada 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).

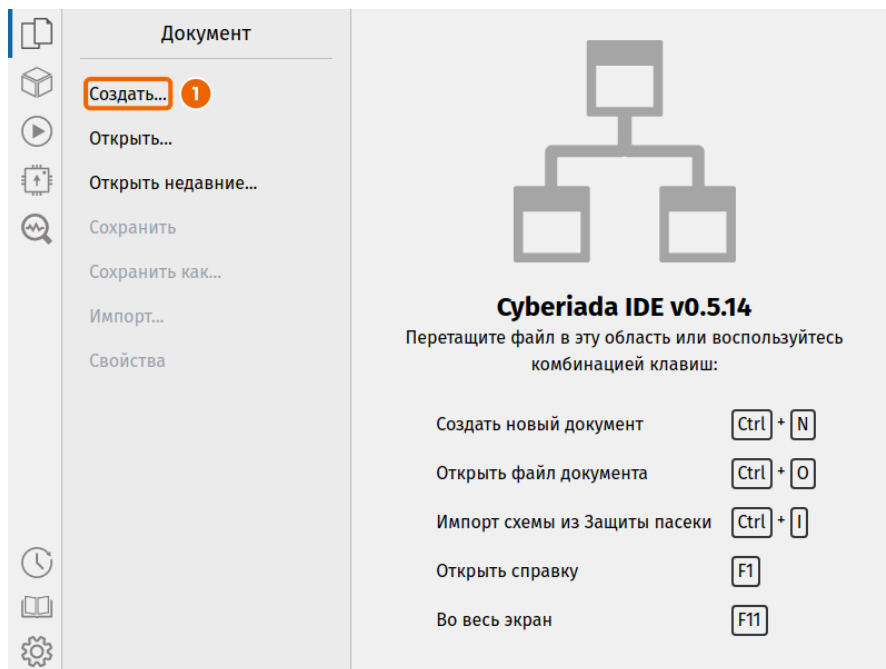


Рисунок В.8 – Стартовый экран Кибериады: «Документ» → «Создать...»

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).

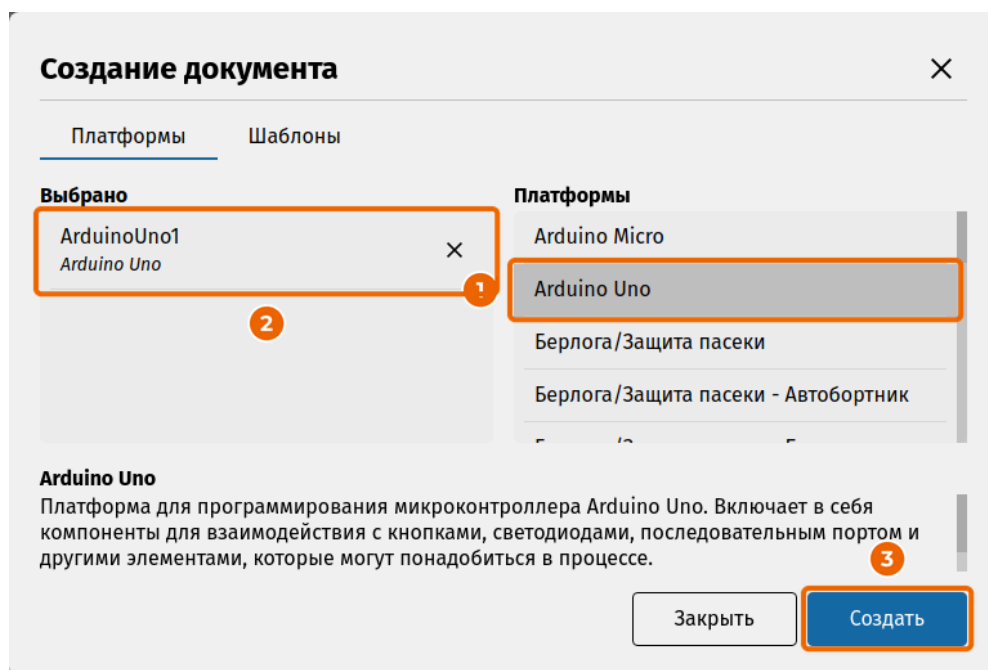


Рисунок В.9 – Выбор платформы для документа

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

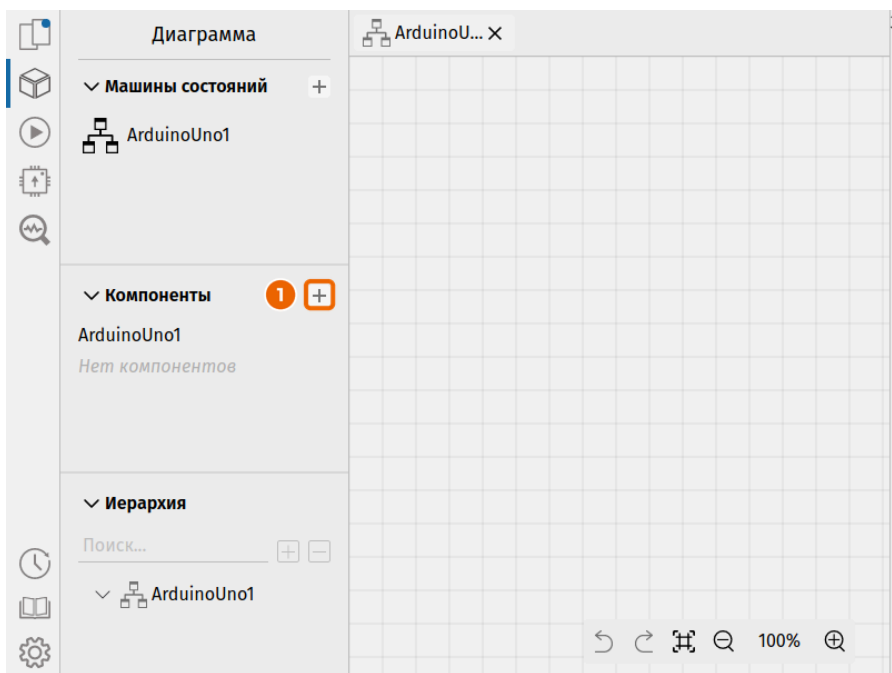


Рисунок В.10 – Окно диаграммы: плюс у «Компонентов»

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».

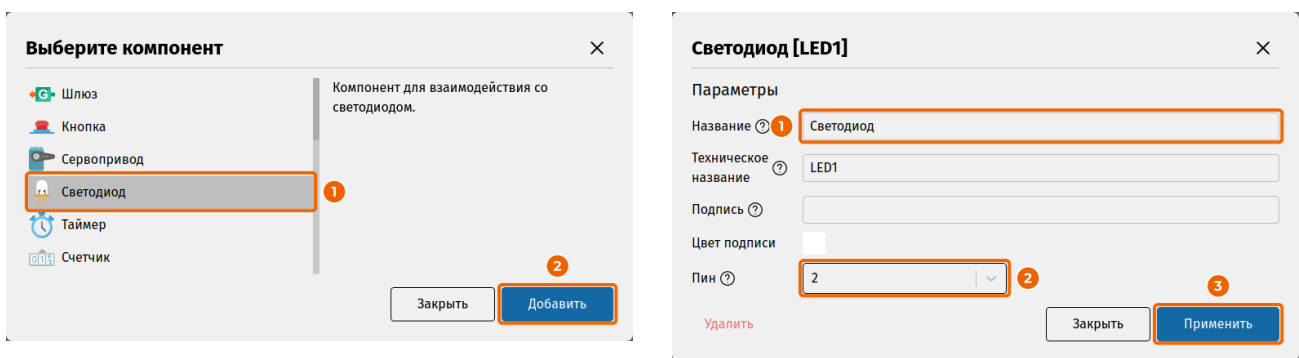


Рисунок В.11 – Новый компонент: слева – выбор «Светодиода», справа – его название и пин

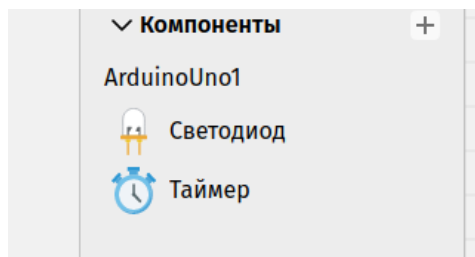


Рисунок В.12 – Оба компонента в списке

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.

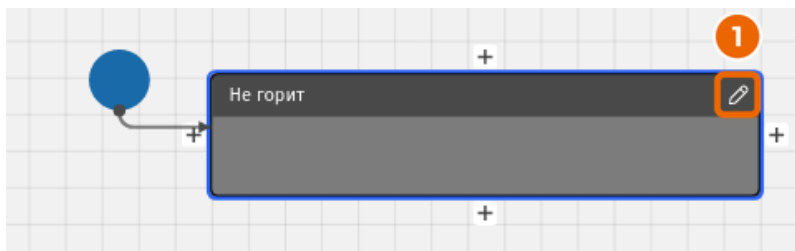


Рисунок В.13 – Новое состояние: карандаш – переименовать

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

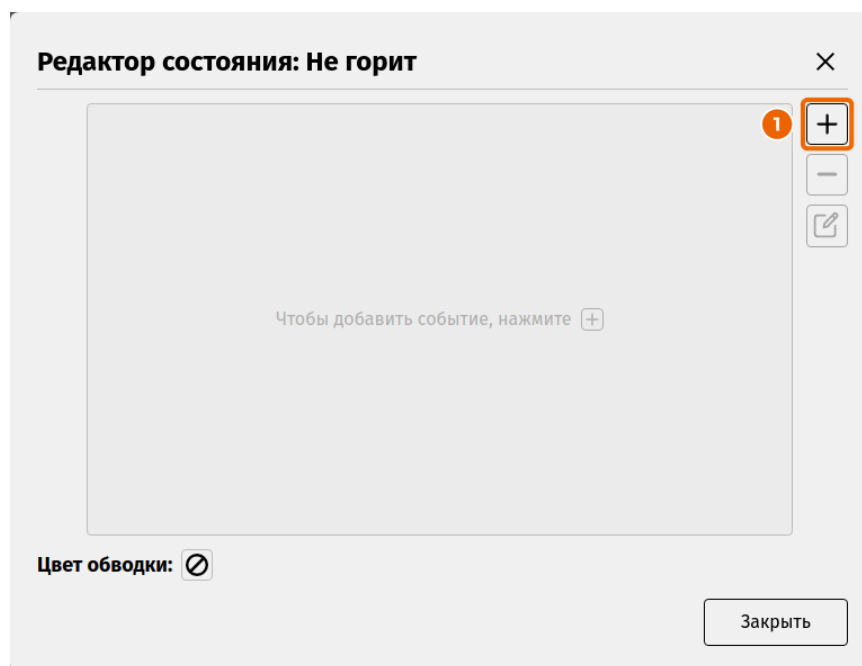


Рисунок В.14 – Редактор состояния

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).

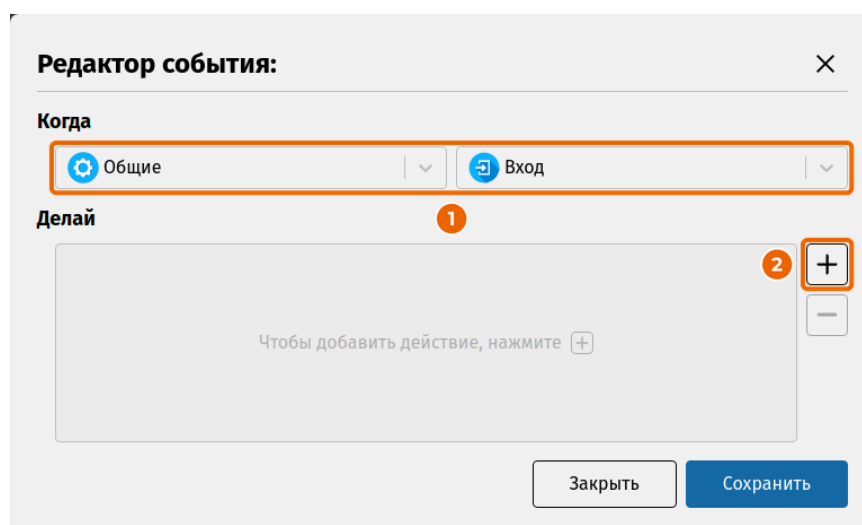


Рисунок В.15 – Редактор события: «Вход» и плюс у «Делай»

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

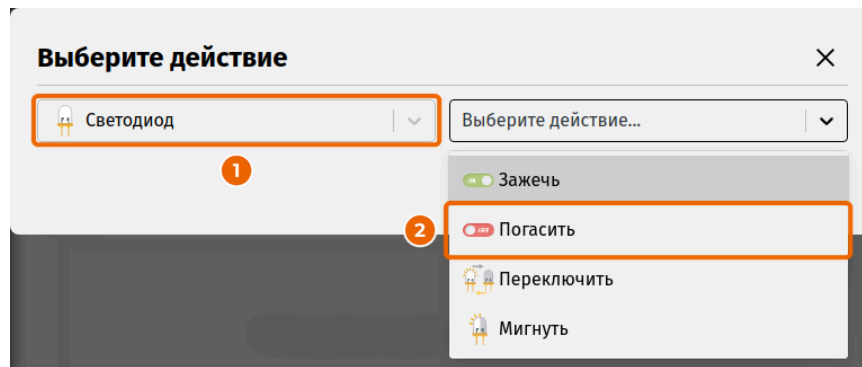


Рисунок В.16 – Действие «Светодиод – Погасить»

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.

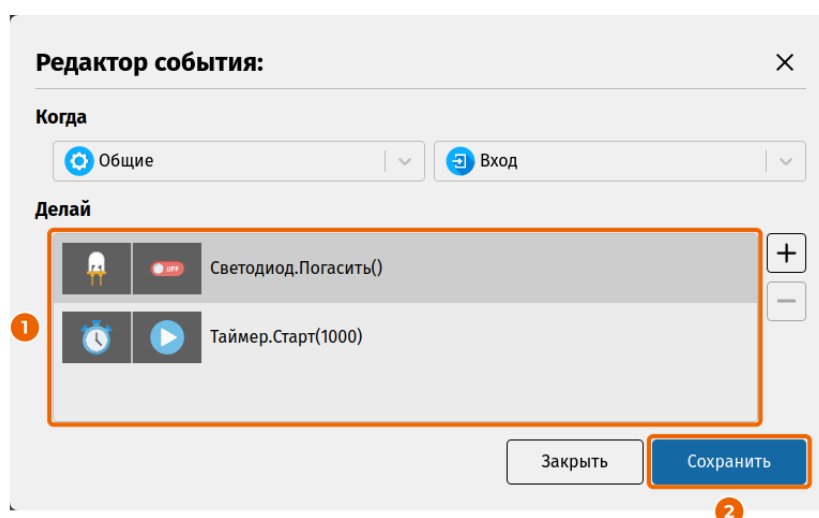


Рисунок В.18 – Событие «Вход» с двумя действиями

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.

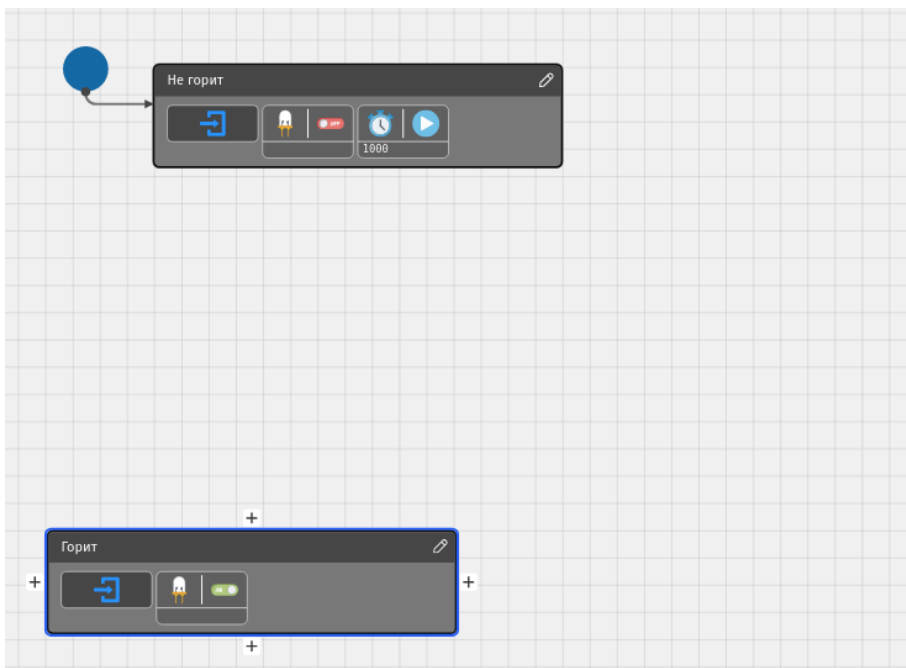


Рисунок В.19 – Два состояния готовы: на каждом видно, что оно делает при входе

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

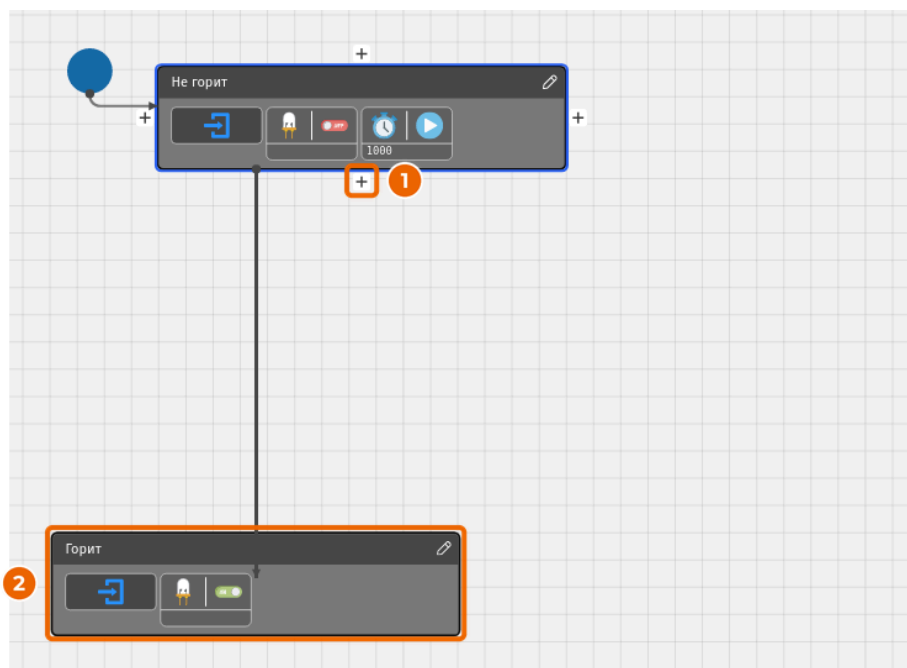


Рисунок В.20 – Тянем стрелку перехода

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).

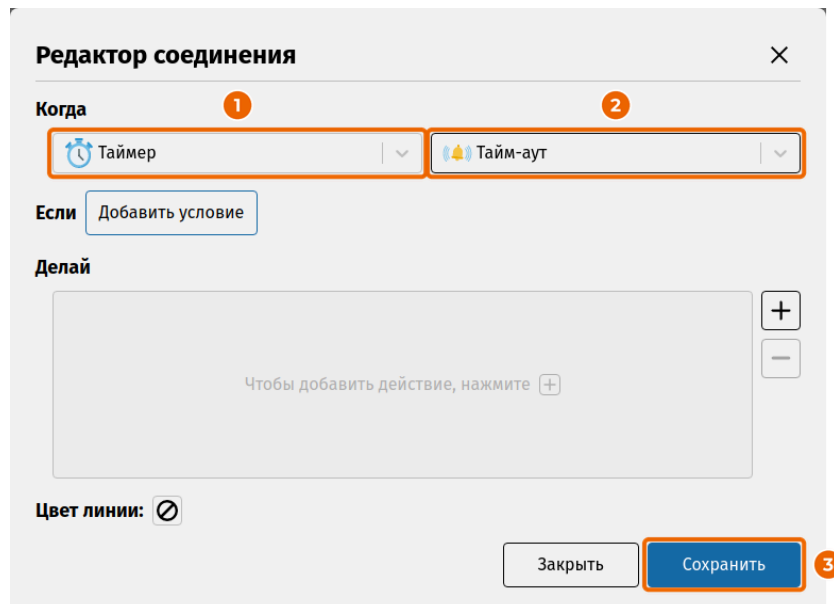


Рисунок В.21 – Переход по тайм-ауту таймера

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.

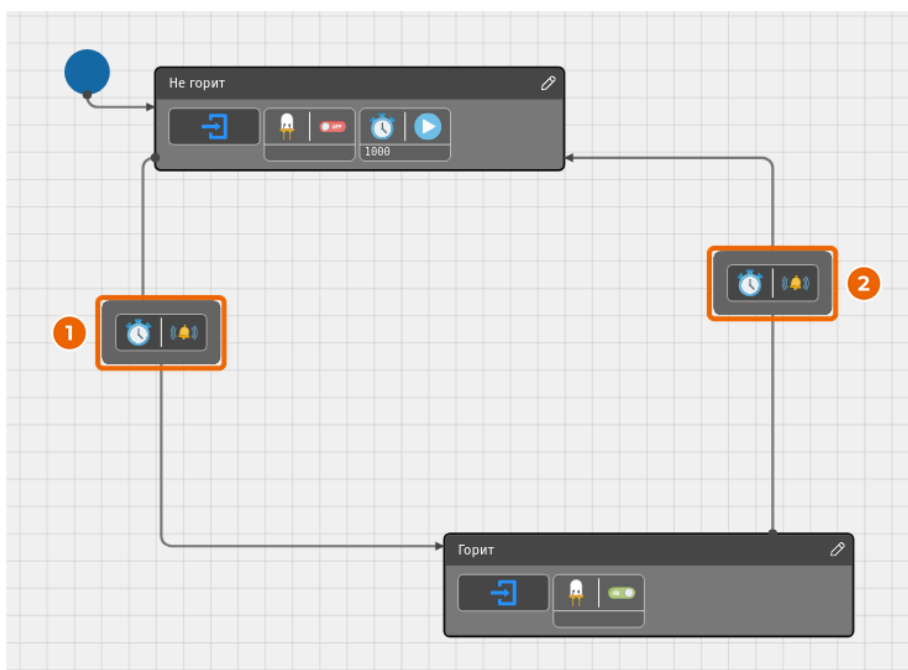


Рисунок В.22 – Готовая «мигалка»: переход «Не горит → Горит» (1) и обратный (2)

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.

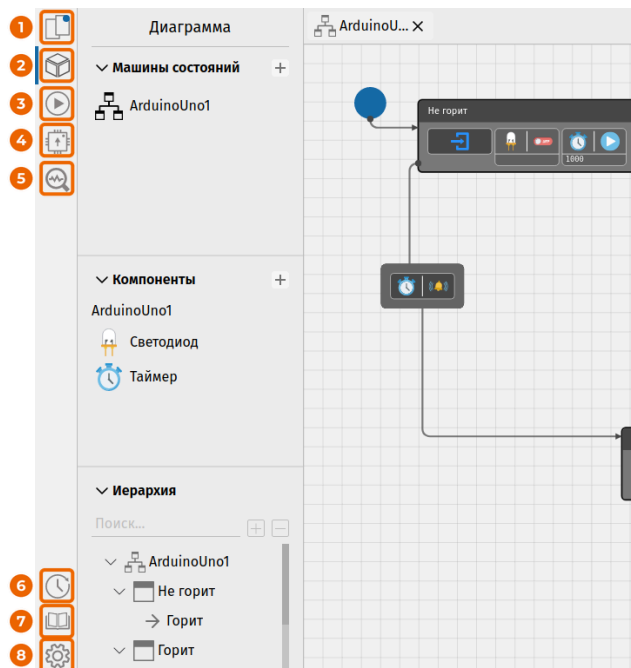


Рисунок В.23 – Значки левой панели

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.

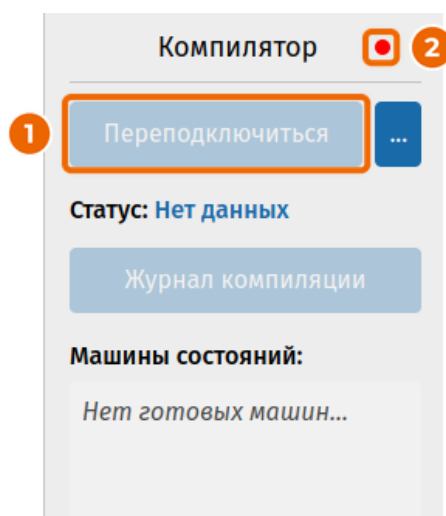


Рисунок В.24 – Панель «Компилятор»

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.

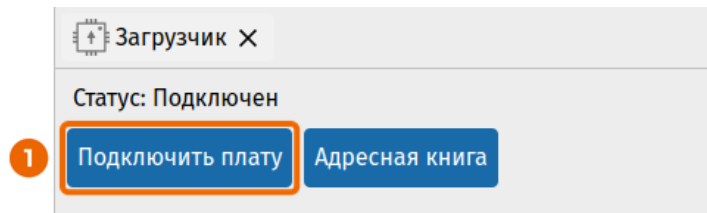


Рисунок В.25 – Панель «Загрузчик»

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).

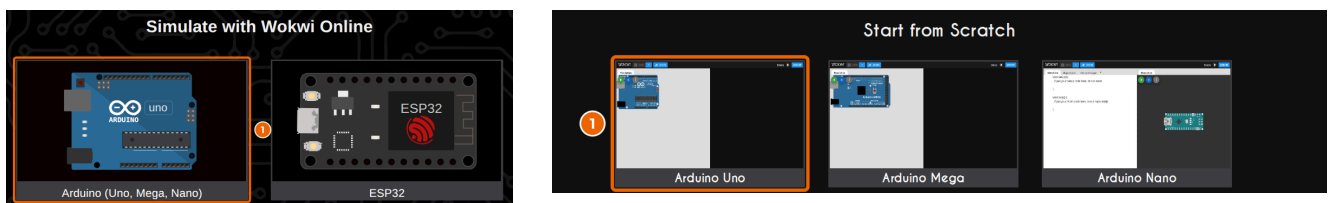


Рисунок В.26 – Как открыть чистый проект: слева – главная страница, справа – выбор платы

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.

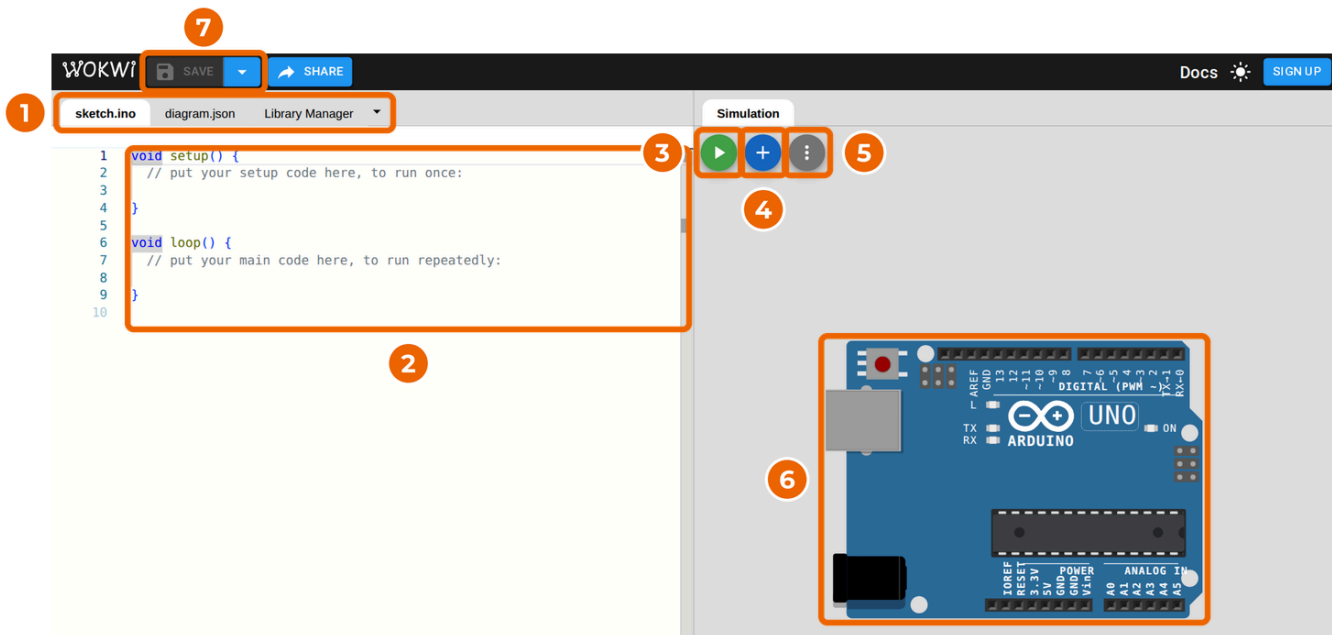


Рисунок В.27 – Окно Wokwi

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.

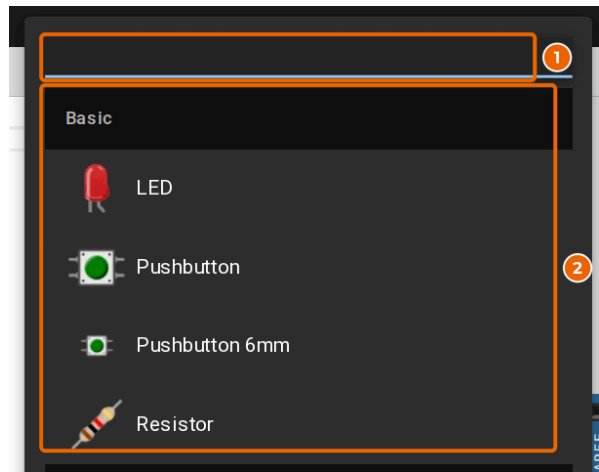


Рисунок В.28 – Список деталей: строка поиска (1) и раздел «Basic» (2)

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.

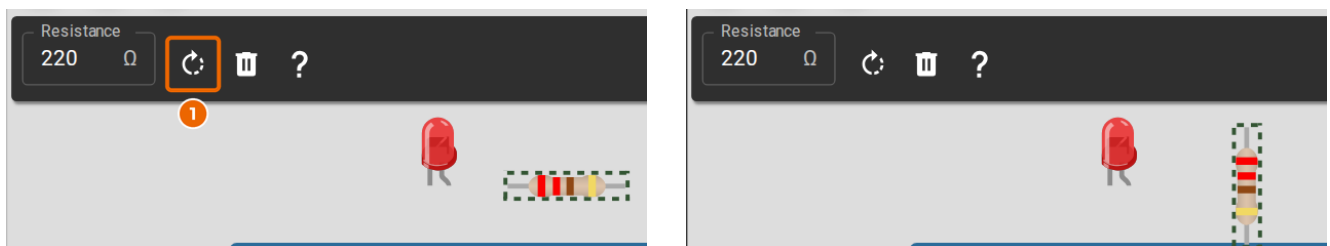


Рисунок В.30 – Поворот: слева резистор лежит, справа – после нажатия «Rotate» (1) он встал вертикально

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.

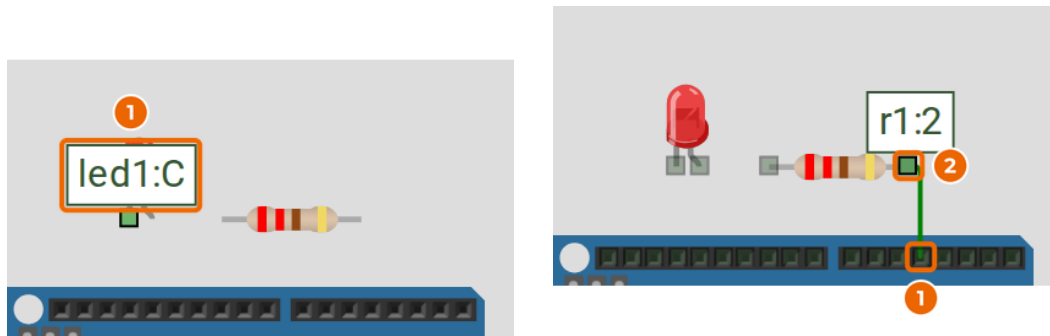


Рисунок В.31 – Слева – подпись ножки при наведении (1), справа – провод тянется от ножки 4 (1) к ножке r1:2 резистора (2)

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 скрыты: 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).

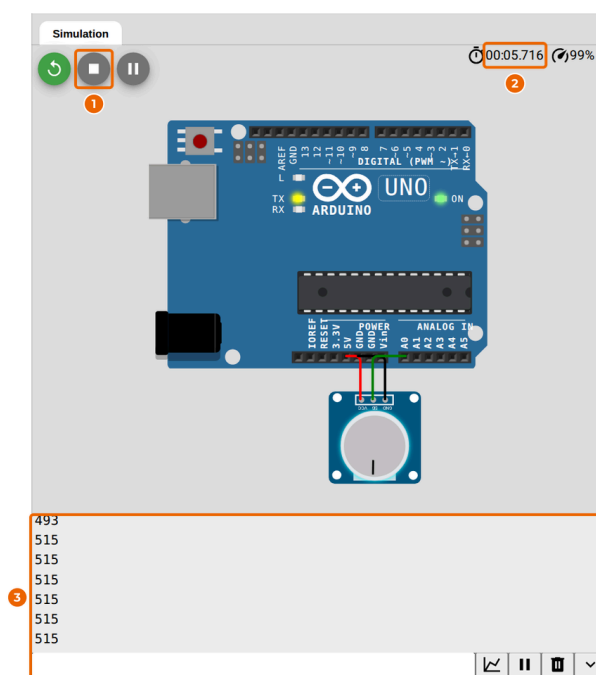


Рисунок В.32 – Идёт симуляция: «Stop the simulation» (1), время (2), монитор порта (3)

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`.

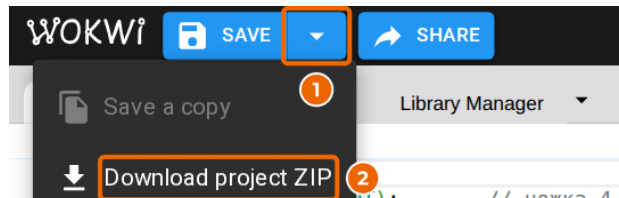


Рисунок В.33 – Меню рядом с «Save»

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.

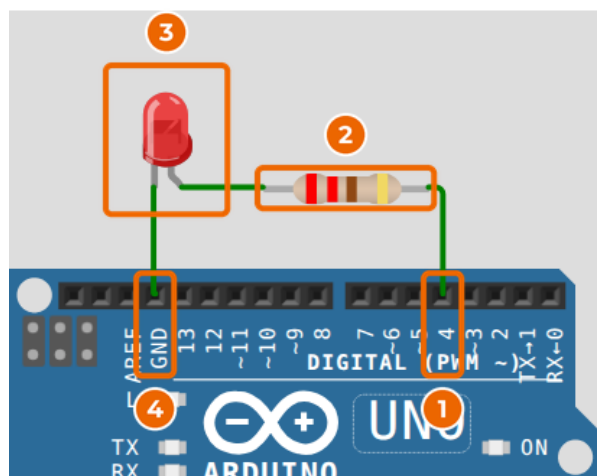


Рисунок В.34 – Схема мигалки

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
}
```

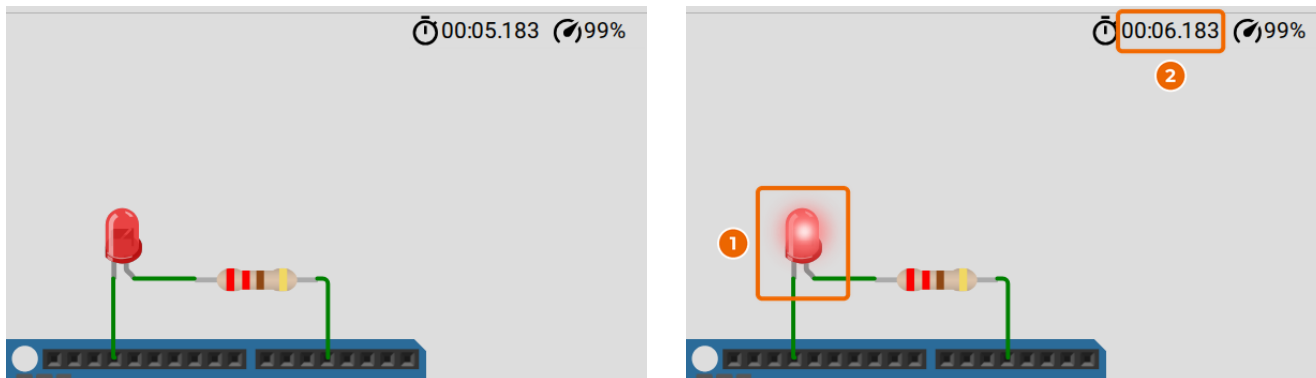


Рисунок В.35 – Слева светодиод не горит, справа горит (1); время симуляции – (2)

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).

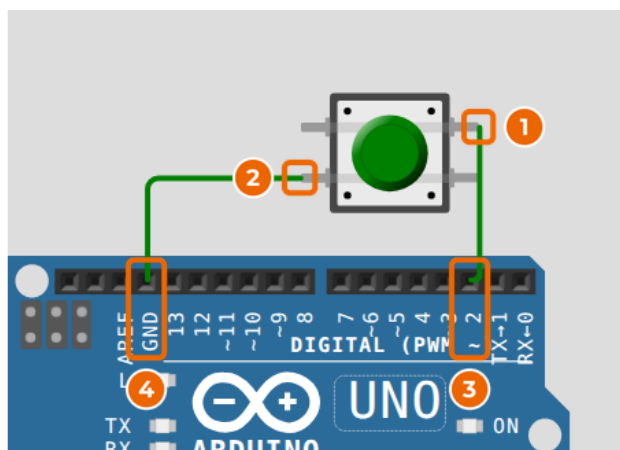


Рисунок В.36 – Схема с кнопкой

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.

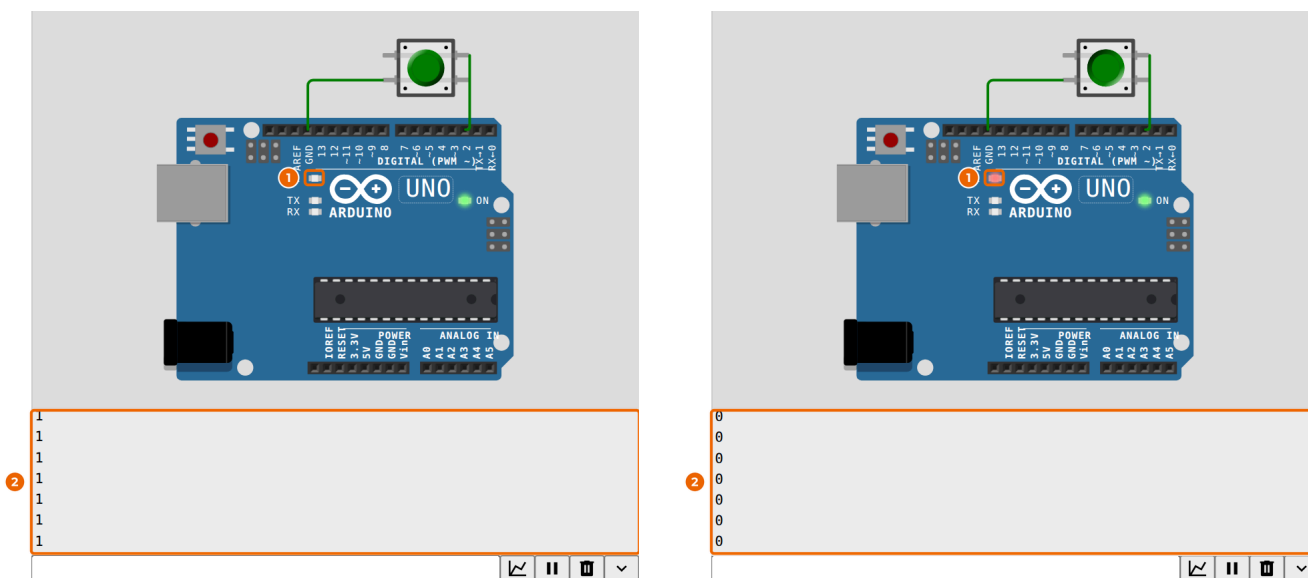


Рисунок В.37 – Слева кнопка отпущена, справа нажата: лампочка L (1) и монитор порта (2)

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).

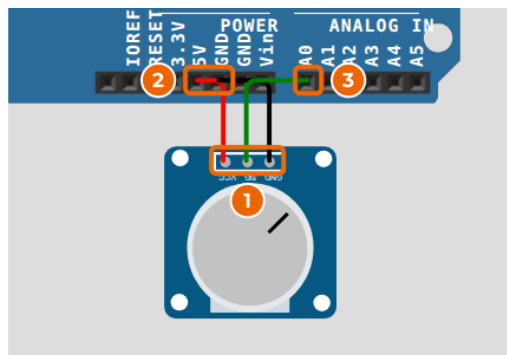


Рисунок В.38 – Схема с потенциометром

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.

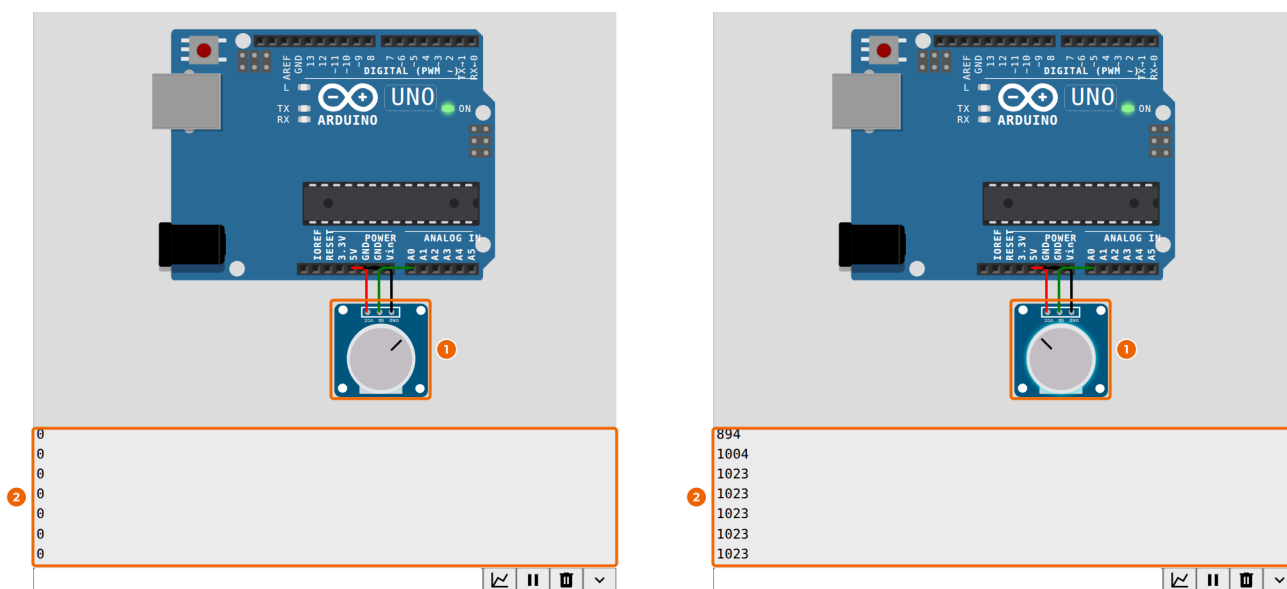


Рисунок В.39 – Слева ручка в одном крайнем положении, справа – в другом: ручка (1) и монитор порта (2)

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	Что делаем
1	главы 1–2	записаться на курс сферы, зайти в Tinkercad по ссылке из курса, посмотреть пробные задания
2	главы 3–4	каждый день 3–4 разбора с задачами «Реши сам»
3	главы 5–6	собрать схемы и программы из разборов в симуляторе
4	главы 7–8 и шпаргалка	заново решить задачи, где были ошибки; пройти тест курса сферы (+5 баллов)

Отбор длится до 10 ноября, и попыток две. Если первая попытка 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 чаще всего останавливает не сложность, а «ссылка не открывается», «схема пропала» или «непонятно, куда нажать».
- **Эта книга не про здоровье.** Нормы пульса и дыхания в ней примерные и нужны только для задач. Ни одна задача не ставит диагнозов, и по графикам из книги нельзя судить о здоровье ребёнка – с такими вопросами к врачу.
- **Самодельные датчики – только со взрослым.** Если дома есть Arduino и хочется снять сигнал с собственной руки, проверьте, что плата питается от батареек или от ноутбука, отключённого от розетки. Ничто, что соединено с розеткой, не должно касаться тела через провода.
- **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 зада-

чам. 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 «Медтех. Основы мониторинга и биоуправления» 2026/27, записям вебинаров сферы и сборнику задач старшей НТО по профилю «Нейротехнологии и когнитивные науки» 2025/26. Архива отборочных задач «Медтеха» нет – сфера появилась в этом сезоне, поэтому почти все задачи в книге наши, а задачи старшей олимпиады дали только идеи.

Книга написана до того, как открылся демо-этап с пробными заданиями. Когда пробные задания станут доступны, мы сверим с ними формат и темы и выпустим обновлённую версию. Если на пробном этапе тебе попало что-то, чего в книге нет, смело пиши – это поможет следующей версии.

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.

Тело человека как система сигналов



Датчики пульса, ЭМГ и ЭЭГ меряют не «здоровье вообще», а сигнал конкретного органа. Не зная, откуда сигнал взялся, график не прочитаешь. Поэтому начнём с тела и посмотрим на него глазами инженера, который разбирает схему робота: где тут насос, где провода, где моторы. Задачи к главе – сопоставить орган и его работу, выбрать верные утверждения, посчитать по паре чисел. Диагнозов и советов по здоровью здесь нет: мы не врачи, нас интересует устройство.

1.1. Тело глазами инженера

НА ПАЛЬЦАХ

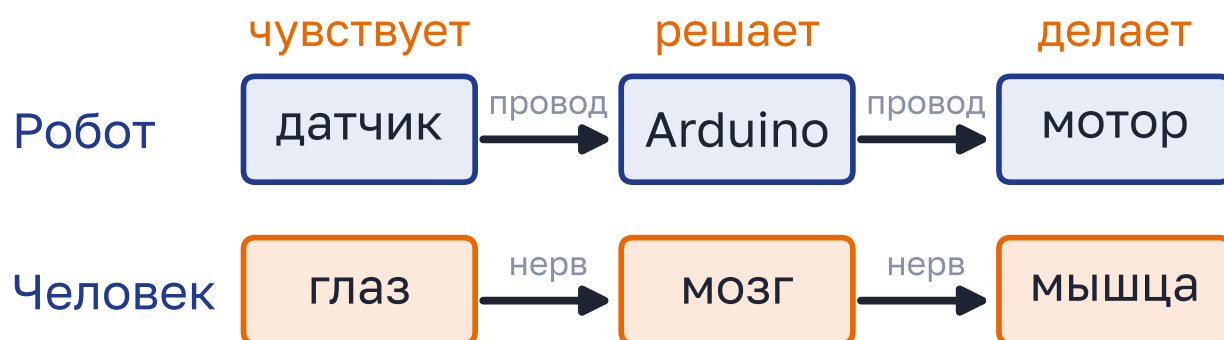


Рисунок 1.1 – Робот и человек устроены по одной схеме

Датчик заметил, контроллер решил, мотор сделал. У человека так же: датчик – глаз или кожа, провода – нервы, контроллер – мозг, мотор – мышца.

Собери в симуляторе схему «нажал кнопку – загорелся светодиод», и в ней окажутся три части. Кнопка – **датчик**: она замечает, что снаружи что-то произошло. Плата Arduino – **контроллер**: по программе решает, что делать. Светодиод или мотор – **исполнитель**: делает то, что решено. Провода соединяют всё это в одну цепочку.

Тело человека собрано по той же схеме, только частей больше. Каждая часть тела со своей работой называется **органом**: сердце, лёгкое, глаз, мышца. Органы, которые вместе делают одно большое дело, образуют **систему органов**. В сфере «Медтех» нам понадобятся пять систем.

Таблица 1.1

Система	Главные органы	Что делает	На что похожа
Кровеносная	сердце, сосуды	развозит кислород и питание	насос и трубы
Дыхательная	лёгкие, трахея	берёт кислород, отдаёт углекислый газ	вентиляция
Нервная	головной и спинной мозг, нервы	собирает сигналы, командует	контроллер и провода
Опорно-двигательная	кости, суставы, мышцы	держит тело и двигает его	каркас, шарниры, моторы
Органы чувств	глаз, ухо, нос, язык, кожа	ловят то, что снаружи	датчики

Системы не живут поодиночке. Ты побежал за школьным автобусом – ногам нужно больше кислорода. Нервная система это замечает и командует: сердцу – биться чаще, лёгким – дышать глубже. Прибежал, сел – и через пару минут всё успокаивается само. Инженер назвал бы это **обратной связью**: система смотрит на результат своей работы и сама себя подправляет.

На этой же идее построены **БОС-тренинги** (БОС – биологическая обратная связь), ради которых и собирают приборы в сфере «Медтех». Датчик снимает с человека сигнал, например дыхание, и показывает его на экране понятной картинкой: шкалой, цветом лампы, героем игры. Человек смотрит на экран и учится менять свой сигнал: дышать медленнее, расслабить мышцу. Кажется, что он управляет игрой, а тренирует он себя.

Запомни. Орган делает свою работу, система органов – общее дело. Инженерная «карта» тела: органы чувств – датчики, нервы – провода, мозг – контроллер, мышцы – моторы, сердце с сосудами – насос с трубами, лёгкие – вентиляция.

1.2. Сердце и кровеносная система

НА ПАЛЬЦАХ

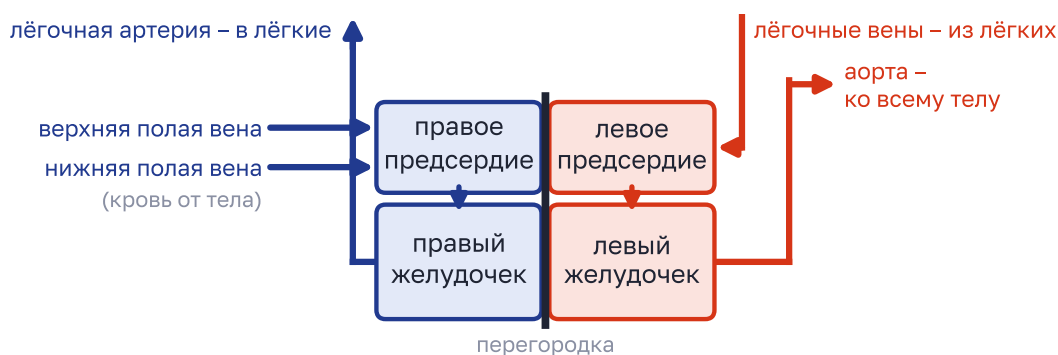


Рисунок 1.2 – Сердце – два насоса в одном

Насос 1 гонит кровь в лёгкие за кислородом, насос 2 – по всему телу. Кругов два, и кровь проходит их по очереди: тело, сердце, лёгкие, сердце, снова тело.

Кровь – это транспорт. От лёгких она везёт кислород ко всем клеткам тела, от клеток забирает **углекислый газ** – то, что остаётся после «сжигания» питания. Везёт она всё это по трубам – **сосудам**, а толкает её **сердце**.

Сердце – мышечный мешок размером примерно с кулак своего хозяина, масса у взрослого около 300 г. Внутри четыре **камеры**. Две верхние – **предсердия**: они принимают кровь. Две нижние – **желудочки**: они сжимаются и выталкивают её. Между предсердием и желудочком, а ещё на выходе из желудочков стоят **клапаны** – дверцы, которые пропускают кровь только в одну сторону, как ниппель у велосипедной камеры. Правая и левая половины сердца разделены глухой стенкой – **перегородкой**, и кровь из одной половины в другую напрямую не попадает.



Сердце показано как у человека, стоящего к тебе лицом: его правая половина – слева.

Рисунок 1.3 – Схема сердца. Синим – кровь, бедная кислородом, красным – богатая. Стрелки – куда она течёт

Проследим за одной каплей крови. Она отдала кислород где-то в пальце ноги и по венам вернулась к сердцу:

- по **полым венам** – в правое предсердие, оттуда в правый желудочек;
- правый желудочек толкает её по **лёгочной артерии** в лёгкие – здесь капля отдаёт углекислый газ и берёт кислород;
- по **лёгочным венам** она возвращается в левое предсердие, оттуда в левый желудочек;
- левый желудочек выталкивает её в **аорту** – самую толстую артерию, и дальше по всему телу.

Путь «правый желудочек – лёгкие – левое предсердие» называют **малым кругом кровообращения**, путь «левый желудочек – тело – правое предсердие» – **большим**. Левому желудочку приходится гнать кровь по всему телу, поэтому стенка у него заметно толще, чем у правого.

Сосуды бывают трёх видов:

- **артерии** несут кровь **от** сердца, у них толстые упругие стенки;
- **вены** несут кровь **к** сердцу;
- **капилляры** – самые тонкие сосуды, тоньше волоса; через их стенку кислород и питание уходят в клетки.

Здесь сидит любимая ловушка задач. Артерию и вену различают по направлению, кислород тут ни при чём. Лёгочная артерия – артерия, потому что идёт от сердца, хотя кровь в ней бедная кислородом. Лёгочные вены – вены, хотя кровь в них самая богатая кислородом во всём теле.

Пульс. При каждом сокращении желудочек выталкивает в аорту порцию крови, и по стенкам артерий пробегает толчок. Его можно нащупать на запястье со стороны большого пальца или на шее сбоку. Число ударов сердца за минуту называют **частотой сердечных сокращений** (ЧСС) и пишут в ударах в минуту (уд/мин). У взрослых в покое она примерно 60–90 уд/мин, у ребят 11–14 лет – примерно 70–100 уд/мин. При беге больше, во сне меньше. Для нас ЧСС – просто число, с которым считают; «хорошее» оно или «плохое», решает врач, а не задачник.

Сколько качает сердце. Порция, которую желудочек выталкивает за одно сокращение, называется **ударным объёмом**. У взрослого в покое это около 70 мл – треть стакана. За минуту сердце делает столько порций, какова ЧСС:

$$\text{объём за минуту} = \text{ударный объём} \cdot \text{ЧСС}.$$

При 70 мл и 70 уд/мин выходит $70 \cdot 70 = 4900$ мл, почти 5 л. Крови у взрослого как раз около 5 л, так что в покое вся она проходит через сердце примерно за минуту.

Что видят датчики. Сердечная мышца, сокращаясь, создаёт электрический сигнал. Его снимают электродами на коже груди или рук – это **ЭКГ**, электрокардиограмма. А датчик пульса из набора устроен иначе: он светит в палец и ловит свет, прошедший сквозь него или отражённый от него. Когда толчок крови доходит до пальца, крови там становится больше, она поглощает больше света, и до приёмника доходит меньше. Так получается **пульсовая волна**. Электричество сердца этот датчик не видит – он видит «эхо» удара в пальце, с небольшой задержкой. Как считать пульс по такому графику, разберём в главе 2.

Запомни. Артерия – от сердца, вена – к сердцу, и сколько в крови кислорода, здесь неважно. Правая половина сердца – кровь бедная кислородом, левая – богатая. Объём за минуту = ударный объём · ЧСС.

1.3. Дыхательная система

НА ПАЛЬЦАХ

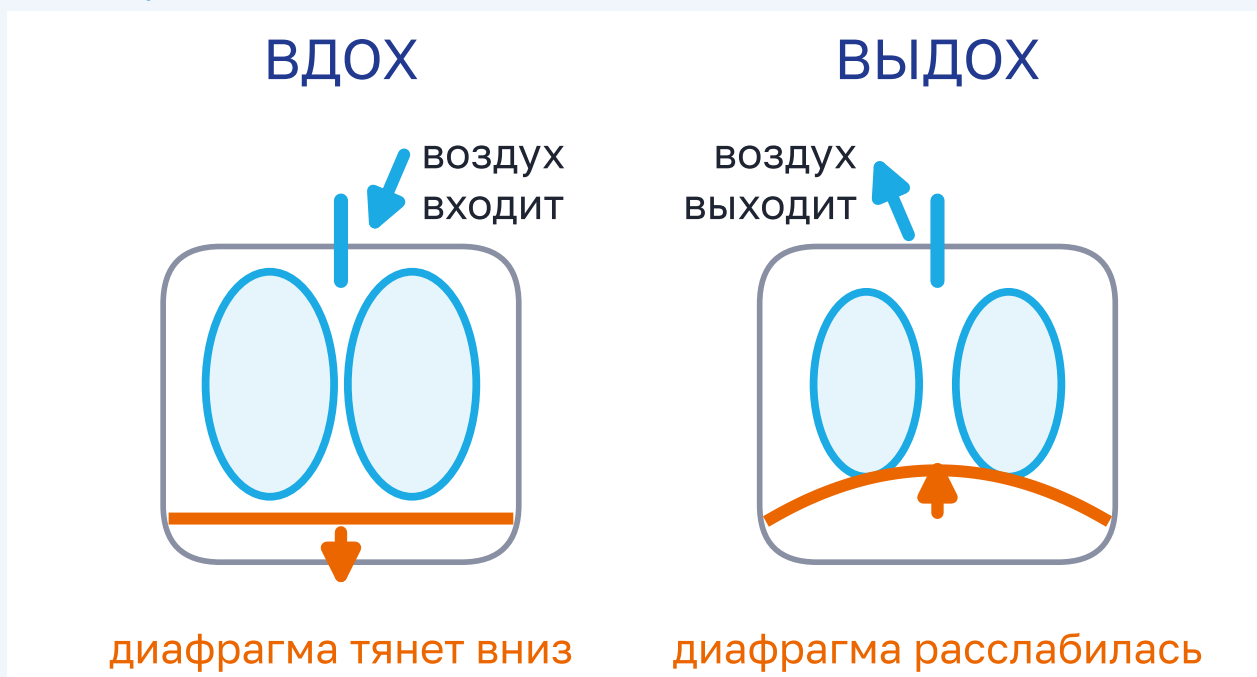


Рисунок 1.4 – Вдох и выдох

Лёгкие сами не раздуваются. Их растягивает диафрагма – мышца под лёгкими. Потянула вниз – воздух вошёл, расслабилась – вышел.

Воздух идёт по трубам с ветками, как дерево вверх ногами: **нос** (там воздух греется, очищается и увлажняется) → **гортань** → **трахея** → два **бронха** → всё более мелкие веточки в **лёгких**. На концах самых тонких веточек – **альвеолы**, крошечные пузырьки с очень тонкой стенкой. Каждый пузырёк оплетён капиллярами. Здесь и происходит **газообмен**: кислород из воздуха проходит в кровь, а углекислый газ из крови – в воздух. Альвеол в лёгких сотни миллионов, поэтому площадь, через которую идёт обмен, – десятки квадратных метров, примерно как пол в классе.

Вдох делают мышцы: диафрагма опускается, мышцы между рёбрами приподнимают грудную клетку, лёгкие растягиваются и засасывают воздух, как шприц, у которого тянут поршень. Спокойный выдох получается сам: мышцы расслабляются, и лёгкие, как растянутая резинка, сжимаются обратно.

Чем отличается воздух, который мы выдыхаем, от того, который вдыхаем:

Таблица 1.2

Газ	Во вдыхаемом воздухе	В выдыхаемом воздухе
Кислород	примерно 21 %	примерно 16 %
Углекислый газ	примерно 0,04 %	примерно 4 %

Кислорода мы забираем совсем немного, а углекислого газа в выдохе становится в 4: 0,04 = 100 раз больше. Поэтому датчик углекислого газа (CO_2 – его химическая формула),

поставленный у носа, «видит» дыхание: на каждом выдохе его показания подпрыгивают. По той же причине в закрытом классе к концу урока углекислого газа в воздухе больше, чем в начале, и класс проветривают.

Взрослый в покое делает примерно 12–20 вдохов в минуту, за один спокойный вдох – примерно 0,5 л воздуха. Объём воздуха за минуту считают так же, как объём крови: объём одного вдоха умножают на число вдохов в минуту.

Дыхание снимают разными датчиками. Можно надеть на грудь пояс с датчиком растяжения: грудь расширяется – пояс тянется. Можно поставить у носа датчик CO₂ или датчик температуры: выдыхаемый воздух тёплый. Есть у дыхания и одно отличие от сердца: сердцу не прикажешь «стучи медленнее», а дыхание можно задержать или замедлить по своей воле. Поэтому дыхание так часто берут для БОС-тренингов.

Запомни. Вдох делают мышцы, главная из них – диафрагма, спокойный выдох идёт сам. В альвеолах кислород уходит в кровь, углекислый газ – в воздух. Углекислого газа в выдохе примерно в 100 раз больше, чем во вдохе.

1.4. Нервная система и нейрон

НА ПАЛЬЦАХ

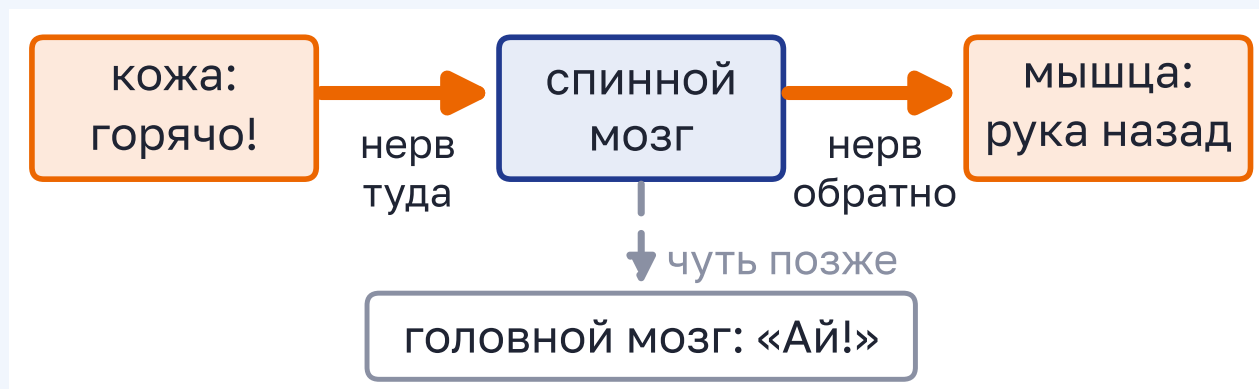


Рисунок 1.5 – Рефлекс: рука отдёргивается раньше, чем ты понял, что горячо

Сигнал бежит короткой дорогой – через спинной мозг. Головной мозг узнаёт о боли чуть позже, когда рука уже отдёрнута.

Нервная система – это контроллер и провода тела. Её центр – **головной мозг** и **спинной мозг**, который идёт внутри позвоночника. Вместе их называют центральной нервной системой. От центра по всему телу расходятся **нервы** – кабели, по которым бегут сигналы.

Строят нервную систему особые клетки – **нейроны**. В головном мозге их около 86 миллиардов. Нейрон не похож на «кругляшок» из учебника про клетку: у него есть отростки, и он умеет принимать и передавать сигналы. На отростки и смотрят, когда отличают нейрон от других клеток.



Рисунок 1.6 – Нейрон. Сигнал входит через дендриты, проходит тело клетки и уходит по аксону к следующей клетке

- **Дендриты** – короткие ветвистые отростки. Они принимают сигналы от других клеток.
- **Тело клетки** с ядром собирает всё, что пришло, и «решает», передавать ли сигнал дальше.
- **Аксон** – длинный отросток, по нему сигнал уходит. Аксон бывает длиной почти в метр: от спинного мозга до пальцев ноги.
- **Миелин** – оболочка вокруг многих аксонов, как изоляция у провода. С ней сигнал идёт намного быстрее.
- **Окончания** аксона подходят к следующему нейрону или к мышце. Место их контакта называют **синапсом**; сигнал передаётся там с помощью особого вещества.

По нейрону сигнал идёт только в одну сторону: от дендритов к окончаниям аксона. **Нерв** – это пучок из множества аксонов в общей оболочке, как кабель из многих проводов.

Нейроны делят по работе. **Чувствительные** несут сигнал от датчиков тела в мозг. **Двигательные** – от мозга к мышцам. **Вставочные** сидят внутри мозга между ними и передают сигнал с одного на другой.

Скорость. По толстым волокнам в миелине сигнал бежит примерно до 100 м/с – это 360 км/ч, быстрее «Сапсана». По тонким волокнам без миелина – около 1 м/с, скорость пешехода. Ток в проводе несравнимо быстрее, но телу хватает.

Рефлекс – автоматический ответ тела на раздражение, который идёт через нервную систему. Дорогу сигнала называют **рефлекторной дугой**: датчик в коже → чувствительный нейрон → спинной мозг → двигательный нейрон → мышца. Рефлексы бывают двух видов:

- **безусловные** – есть у человека с рождения и одинаковы у всех: моргнуть, когда в глаз летит соринка, отдернуть руку от горячего, дернуть ногой, когда врач стукнет молоточком под коленку;
- **условные** – вырабатываются в жизни, их надо «выучить»: кот мчится на кухню на звук открывающегося пакета с кормом, а ты тянешься к рюкзаку, едва прозвенел звонок с урока.

Когда реакция идёт через решение головного мозга, она медленнее. Заметить, что зажёгся светодиод, и нажать кнопку человек успевает примерно за 0,2–0,3 с. Игру «на реакцию» со светодиодом и кнопкой собирают на занятиях по программе сферы – проверь себя, когда соберёшь, у тебя выйдет что-то похожее.

Запомни. Дендриты принимают, аксон передаёт, по нейрону сигнал идёт в одну сторону. Нерв – пучок аксонов. Безусловный рефлекс есть с рождения, условный вырабатывается. Миелин ускоряет сигнал.

1.5. Мышцы и опорно-двигательный аппарат

НА ПАЛЬЦАХ



Рисунок 1.7 – Мышца умеет только тянуть

Толкать мышца не может. Поэтому мышцы работают парами: бицепс сгибает руку в локте, трицепс разгибает.

Скелет – каркас тела, у взрослого в нём 206 костей. Он держит форму, защищает (череп – головной мозг, рёбра – сердце и лёгкие) и служит рычагами для движения. Кости соединены **суставами** – подвижными соединениями. Локоть и колено сгибаются в одну сторону, как дверная петля. Плечо крутится почти во все стороны, как шарнир у штатива.

Мышцы двигают кости. Скелетных мышц у человека больше 600, к костям они крепятся прочными тяжами – **сухожилиями**. Мышечная ткань бывает трёх видов:

- **скелетная** – ей мы управляем сами: руки, ноги, язык, мышцы лица;
- **гладкая** – в стенках желудка, кишечника, сосудов; работает сама, приказать ей нельзя;
- **сердечная** – из неё сделано сердце; сокращается сама всю жизнь и не устаёт.

Мышца умеет одно: **сокращаться**, то есть становиться короче и тянуть за сухожилие. Обратное её растягивает другая мышца, которая тянет в противоположную сторону. Сгибаешь руку в локте – работает **бицепс** на передней стороне плеча, а **трицепс** сзади расслаблен. Разгибаешь – наоборот. У человека, который делает движение ловко и привычно, лишняя мышца пары почти не напрягается – так меньше устаёшь.

Как мышца «включается»? По двигательному нерву к ней приходят импульсы, и мышечные волокна сокращаются. По самим волокнам при этом тоже пробегают электрические импульсы. Их можно снять электродами с кожи над мышцей – это **ЭМГ**, электромиограмма. На графике ЭМГ линия дрожит вверх-вниз, и высоту этих «зубцов» называют **амплитудой**. Чем сильнее напрягаешь мышцу, тем больше волокон включается и тем чаще приходят импульсы – зубцы растут. Расслабленная мышца даёт почти ровную линию.

На этом устроена **бионическая рука** – модель протеза, которую собирают в сфере. Мышцы, которые сжимают пальцы, лежат на предплечье. ЭМГ-датчик на предплечье ловит их сигнал, контроллер сравнивает амплитуду с порогом, и сервомоторы сгибают пальцы модели. Подробно про ЭМГ – в главе 3.

Запомни. Мышца только тянет, поэтому мышцы работают парами. Бицепс сгибает руку в локте, трицепс разгибает. Сильнее напряжение – больше амплитуда ЭМГ, расслабленная мышца – почти ровная линия.

1.6. Органы чувств

НА ПАЛЬЦАХ



Рисунок 1.8 – Органы чувств – датчики тела

Каждый ловит своё: глаз не слышит, ухо не видит. Точно так же датчик освещённости не измерит громкость.

В каждом органе чувств есть **рецепторы** – клетки-датчики. Рецептор превращает то, что он поймал (свет, звук, запах), в нервные импульсы. Дальше импульсы бегут по нерву в свой участок мозга, и только там появляется картинка, звук или запах. Видим мы, по большому счёту, мозгом: глаз только ловит свет.

Глаз похож на фотокамеру. Роговица и хрусталик – линзы: они собирают изображение на задней стенке глаза, на **сетчатке**. Сетчатка – как матрица камеры, в ней два вида рецепторов.

Палочки работают даже в сумерках, но цвета не различают. **Колбочки** различают цвета, но им нужен хороший свет – поэтому вечером на даче все кусты кажутся серыми. **Зрачок** – отверстие, через которое свет входит в глаз: в темноте он расширяется, на ярком свете сужается. Изображение на сетчатке перевёрнуто, и мозг сам его «переворачивает».

Ухо ловит звук. Звук – это колебания воздуха. Они раскачивают **барабанную перепонку**, три крошечные слуховые косточки передают дрожание дальше, в **улитку**, и там рецепторы превращают его в импульсы. Сколько раз в секунду колеблется воздух, называют **частотой** и измеряют в **герцах** (Гц): 1 Гц – одно колебание в секунду. Человек слышит примерно от 20 до 20 000 Гц. Во внутреннем ухе, рядом с улиткой, сидит ещё и **вестибулярный аппарат** – датчик равновесия и наклона. Он работает, как датчик наклона из набора: чувствует, куда наклонена голова.

Нос чувствует запахи – молекулы веществ, которые долетели до него с воздухом. **Язык** различает вкусы: сладкий, солёный, кислый, горький и умами – вкус мясного бульона. **Кожа** – самый большой орган чувств: в ней рецепторы касания, давления, тепла, холода и боли.

Запомни. Рецептор превращает свет, звук, запах или касание в нервные импульсы, а ощущение складывается в мозге. Колбочки – цвет, палочки – сумерки. В ухе два датчика: улитка – звук, вестибулярный аппарат – равновесие.

1.7. Электричество в живых организмах

НА ПАЛЬЦАХ

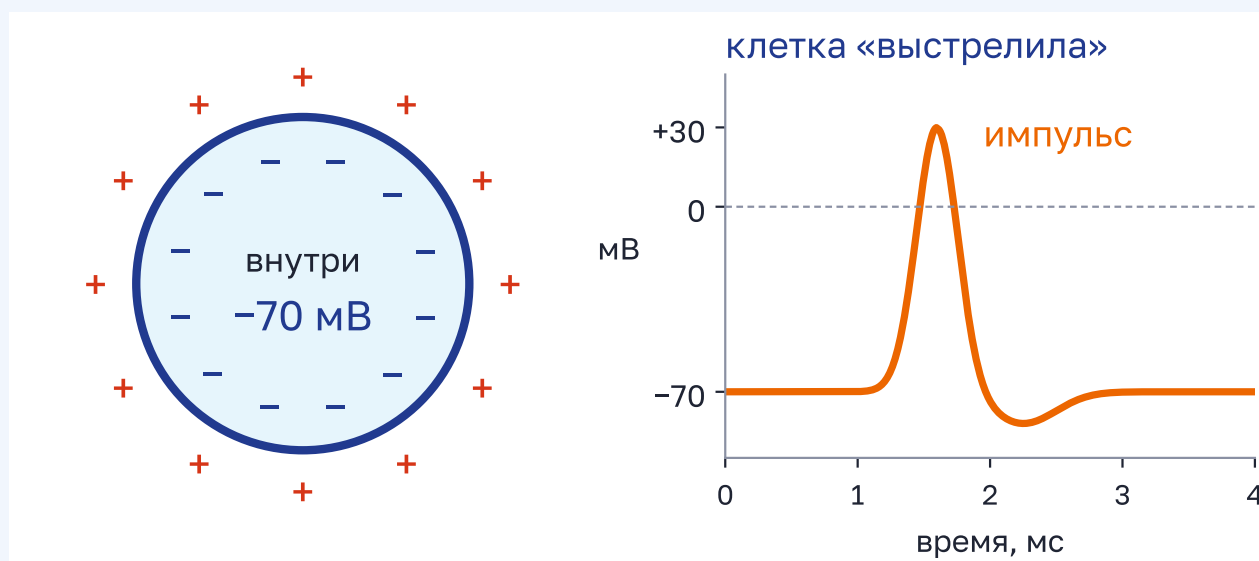


Рисунок 1.9 – Каждая клетка – маленькая батарейка

Внутри клетки – минус по сравнению с тем, что снаружи: около -70 мВ. Когда нервная клетка «стреляет», внутри на миг, меньше чем на тысячную долю секунды, становится плюс.

Сначала одно слово из физики, которой в школе у тебя, возможно, ещё не было. **Напряжение** – это разница «электрического давления» между двумя точками. Похоже на два бака с водой, соединённых трубкой: чем больше разница уровней, тем сильнее побежит вода. Напряжение измеряют в **вольтах** (В). У пальчиковой батарейки 1,5 В, у Arduino – 5 В.



Рисунок 1.10 – Напряжение – как разница уровней воды. Чем больше разница, тем сильнее напор в трубке

В теле напряжения гораздо меньше, поэтому нужны мелкие единицы:

1 мВ (милливольт) = 0,001 В, 1 мкВ (микровольт) = 0,001 мВ = 0,000 001 В.

Откуда в теле электричество? В жидкостях тела растворены соли, а в растворе соль распадается на **ионы** – частицы с зарядом, плюсовым или минусовым. Главные для нас – ионы натрия и калия. Оболочка клетки, **мембрана**, пропускает ионы выборочно, через особые «дверцы». Поэтому внутри клетки заряды распределены иначе, чем снаружи, и между внутренней и внешней стороной мембраны есть напряжение – около –70 мВ. Его называют **потенциалом покоя**: клетка отдыхает, но «заряжена», как батарейка.

Когда нервная или мышечная клетка получает сильный сигнал, в мембране на мгновение открываются дверцы для натрия. Ионы натрия влетают внутрь, и напряжение подскакивает примерно до +30 мВ. Примерно через тысячную долю секунды (1 мс) всё возвращается к –70 мВ. Такой всплеск называется **потенциалом действия**, или нервным импульсом. Он бежит по аксону, как «волна» на трибунах стадиона: каждый зритель лишь встаёт и садится, а волна обегает весь стадион.

Одна клетка даёт крошечный сигнал, и снаружи его не заметить. Но когда тысячи клеток срабатывают вместе – все волокна сердца или целая мышца, – их сигналы складываются, и часть доходит до кожи. Там его и ловят электроды. Чем глубже источник и чем больше ткани на пути, тем слабее сигнал на коже. Сердце большое и сокращается целиком, поэтому ЭКГ на коже – около 1 мВ. Сигнал мозга ослабевает, проходя через кость черепа, и ЭЭГ на коже головы – всего десятки микровольт.

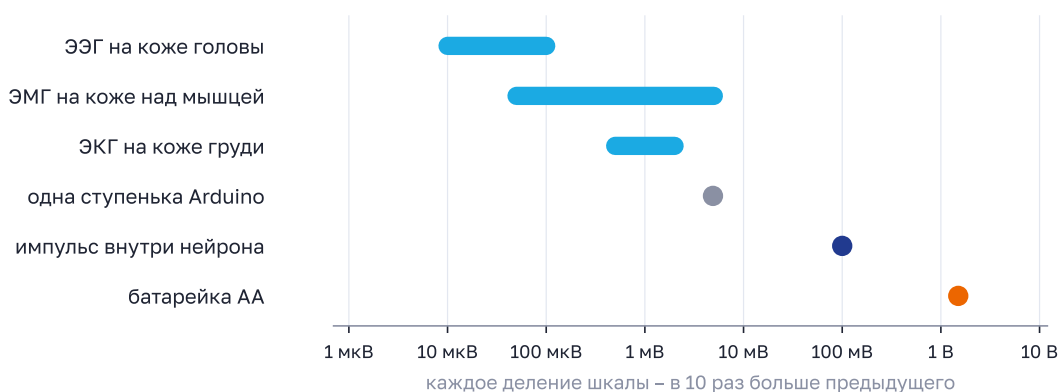


Рисунок 1.11 – Какие напряжения встречаются в теле и рядом с ним (примерно). Шкала необычная: каждое следующее деление в 10 раз больше предыдущего

Сравни с платой Arduino Uno. Напряжение на входе платы читает команда `analogRead` : она делит диапазон от 0 до 5 В на 1024 ступеньки и выдаёт номер ступеньки, число от 0 до 1023. Одна ступенька – примерно 4,9 мВ. Сигнал ЭЭГ в десятки и сотни раз меньше одной сту-

пеньки, так что напрямую плата его не заметит. Поэтому в каждом датчике биосигналов стоит **усилитель** – схема, которая увеличивает напряжение в сотни и тысячи раз. Как плата превращает напряжение в число, подробно разберём в главе 5.

Бывает в природе и настоящее «живое оружие»: электрический угорь из рек Южной Америки складывает напряжения тысяч особых клеток и бьёт разрядом в сотни вольт. Нашему телу до угря далеко: его сигналы приходится ловить очень чувствительными приборами.

Запомни. Внутри клетки в покое около -70 мВ; нервный импульс – всплеск примерно до $+30$ мВ длиной около 1 мс. На коже: ЭКГ – около 1 мВ, ЭМГ – от десятков микровольт до нескольких милливольт, ЭЭГ – десятки микровольт. $1 \text{ мВ} = 1000 \text{ мкВ}$.

Совет. Перед тем как сравнивать напряжения, переведи их в одни единицы. Самая частая беда – сравнить «70» из микровольт с «4,9» из милливольт и решить, что 70 больше.

1.8. Какой датчик что меряет

НА ПАЛЬЦАХ

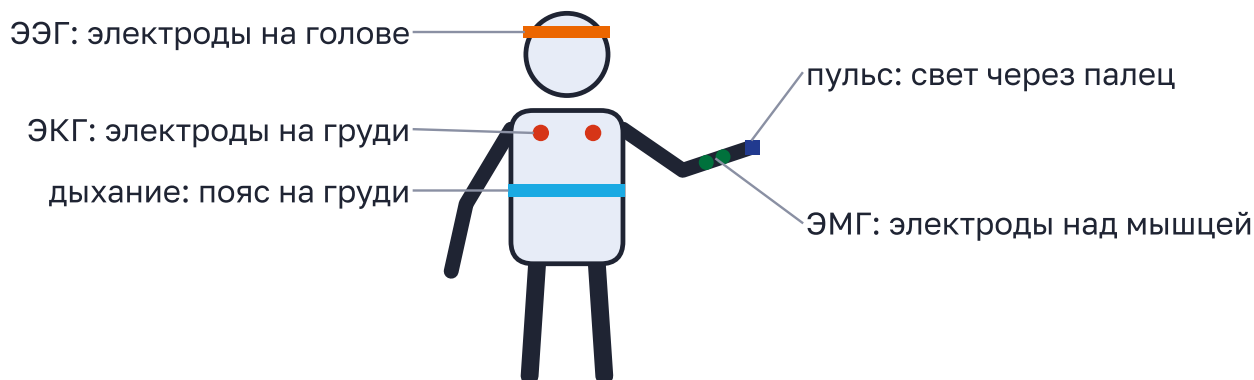


Рисунок 1.12 – Где ставят датчики биосигналов

Электроды ставят туда, где под кожей работает источник сигнала: на голову – для мозга, на грудь – для сердца, на предплечье – для мышц. Пульс и дыхание ловят по-другому – светом и растяжением.

Все датчики биосигналов делятся на две группы, и в задачах их то и дело просят различить.

Первая группа – электрические сигналы самого тела. Их вырабатывают клетки, а датчик только «слушает» электродами на коже и ничего не посылает в тело. Названия у всех трёх похожи, и их легко расшифровать: «электро-» – пишут электричество, «-грамма» – запись, а в середине – чья:

- **ЭКГ** – электрокардиограмма, «кардио» – сердце. Электроды на груди или на запястьях.
- **ЭМГ** – электромиограмма, «мио» – мышца. Электроды на коже над работающей мышцей.

- **ЭЭГ** – электроэнцефалограмма, «энцефало» – головной мозг. Электроды на голове, на ободке. В ЭЭГ ищут **ритмы** – колебания с определённой частотой. Например, альфа-ритм, 8–13 Гц, заметнее, когда человек спокойно сидит с закрытыми глазами (подробно – в главе 3).

Вторая группа – неэлектрические сигналы. Здесь тело само электричества не даёт, и датчик превращает в напряжение что-то другое:

- **пульс на пальце** – светодиод и фотоприёмник: меняется, сколько света проходит сквозь палец (раздел 1.2);
- **дыхание** – пояс с датчиком растяжения на груди, датчик CO₂ или датчик температуры у носа (раздел 1.3);
- **положение и наклон** – датчик наклона: сам по себе он к телу отношения не имеет, но закреплённый на голове или руке показывает её движение;
- **температура тела** – термодатчик.

Есть и обратная сторона. Датчики из первой группы ловят каждое движение: моргнул – на ЭЭГ скачок от мышц век, пошевелился – ЭКГ «поехала». Такие лишние всплески называют **помехами**, и бороться с ними – отдельное ремесло, о нём глава 4.

Запомни. ЭКГ – сердце, ЭМГ – мышцы, ЭЭГ – мозг: это электричество самого тела, его ловят электроды. Пульс на пальце – свет, дыхание – растяжение груди или CO₂ у носа: здесь датчик сам делает электрический сигнал из неэлектрического.

1.9. Разбираем задачи

Совет. В задачах «выбери все верные» проверяй каждое утверждение отдельно, как будто других нет: верных может оказаться одно, а может – все. Обведи слова «всегда», «только», «никогда» – одного исключения хватит, чтобы утверждение стало неверным. В «сопоставь» начинай с пар, в которых уверен, а спорные решай последними: к тому времени выбор сузится.

Задача 1.1 Кто за что отвечает

разминка · органы и их работа, сопоставление

Сопоставь органы и их работу. Одна работа лишняя.

Органы: **1** – сердце, **2** – альвеолы лёгких, **3** – спинной мозг, **4** – трицепс, **5** – сетчатка глаза, **6** – вестибулярный аппарат.

Работа:

- **А** – разгибает руку в локте;
- **Б** – качает кровь по сосудам;
- **В** – ловит свет и превращает его в нервные импульсы;

- **Г** – проводит сигналы между головным мозгом и телом, через него идут простые рефлексы;
- **Д** – сгибает руку в локте;
- **Е** – здесь кислород переходит из воздуха в кровь;
- **Ж** – чувствует наклон головы и равновесие.

Дано: 6 органов, 7 описаний работы (одно лишнее).

Найти: пары «орган – работа».

Решение

Шаг 1. Пары, в которых сомнений нет. Сердце – насос: 1 – Б. Сетчатка – «матрица» глаза, в ней рецепторы света: 5 – В.

Шаг 2. Лёгкие. Альвеолы – пузырьки на концах самых тонких веточек бронхов, оплетённые капиллярами. Через их стенку кислород уходит в кровь: 2 – Е.

Шаг 3. Нервная система. Спинной мозг лежит в позвоночнике, и по нему бегут сигналы от тела к головному мозгу и обратно. Через него же идёт рефлекс «отдёрнуть руку»: 3 – Г.

Шаг 4. Мышцы. Здесь два похожих описания – А и Д. Трицепс лежит на задней стороне плеча и разгибает руку: 4 – А. Описание Д подходит бицепсу, а его в списке органов нет.

Шаг 5. Вестибулярный аппарат. Он сидит во внутреннем ухе, но звуков не ловит – это датчик равновесия: 6 – Ж.

Шаг 6. Проверка. Шесть органов получили шесть разных букв, свободной осталась Д. В условии и сказано, что одна работа лишняя.

 **Типичная ошибка.** Перепутать бицепс и трицепс и поставить 4 – Д. Подсказка на всю жизнь: сгибаешь руку, чтобы показать «силу», – надувается бицепс спереди. Вторая ошибка – решить, что раз вестибулярный аппарат в ухе, то он про слух.

Ответ: 1 – Б, 2 – Е, 3 – Г, 4 – А, 5 – В, 6 – Ж (лишнее – Д).

Своя задача по программе сферы, модуль 2

Реши сам

1.1. Сопоставь орган чувств и то, что он ловит.

Органы: **1** – глаз, **2** – улитка внутреннего уха, **3** – нос, **4** – язык, **5** – кожа.

Что ловит: **А** – молекулы веществ в воздухе, **Б** – колебания воздуха, **В** – свет, **Г** – вещества, растворённые в слюне, **Д** – касание и тепло.

1.2. Сопоставь часть тела и деталь робота, на которую она похожа по работе. Одна деталь лишняя.

Части тела: **1** – сердце, **2** – нерв, **3** – локтевой сустав, **4** – мышца, **5** – головной мозг, **6** – кость.

Детали: **А** – кабель из многих проводов, **Б** – дверная петля, **В** – насос, **Г** – контроллер, **Д** – лебёдка, которая тянет трос, **Е** – жёсткий рычаг каркаса, **Ж** – вентилятор.

Задача 1.2 Где кровь богата кислородом

кровеносная система, круги кровообращения, артерии и вены

Выбери все части кровеносной системы, в которых течёт кровь, **богатая** кислородом. В ответ запиши номера выбранных вариантов в порядке возрастания, без пробелов.

- **1** – левый желудочек;
- **2** – правое предсердие;
- **3** – лёгочная артерия;
- **4** – лёгочные вены;
- **5** – аорта;
- **6** – нижняя полая вена.

Дано: 6 частей кровеносной системы; схема сердца из раздела 1.2.

Найти: номера частей с кровью, богатой кислородом.

Решение

Шаг 1. Где кровь берёт кислород. Только в лёгких, в капиллярах вокруг альвеол. Значит, кровь богата кислородом **после** лёгких и до тех пор, пока не отдаст кислород клеткам тела.

Шаг 2. Участок после лёгких. Из лёгких кровь идёт по лёгочным венам в левое предсердие, оттуда в левый желудочек и в аорту. Всё это – левая, «красная» половина: 4, 1, 5.

Шаг 3. Участок после тела. Отдав кислород в капиллярах тела, кровь возвращается по полым венам в правое предсердие, потом в правый желудочек и по лёгочной артерии – в лёгкие. Здесь она бедная кислородом: 6, 2, 3.

Шаг 4. Сверяемся со схемой. На рисунке к разделу 1.2 красные – лёгочные вены, левое предсердие, левый желудочек и аорта. Из нашего списка это 1, 4, 5. Три части «красные», три «синие» – по три на каждый участок пути.

 **Типичная ошибка.** Выучить «в артериях кровь с кислородом, в венах – без» и выбрать 1, 3, 5. Правило неверное: артерии и вены различают по направлению (от сердца или к сердцу). Лёгочная артерия несёт кровь в лёгкие, как раз **за** кислородом, а лёгочные вены несут её оттуда уже с кислородом.

Ответ: 145 (левый желудочек, лёгочные вены, аорта).

По мотивам задачи НТО 2025/26, профиль «Нейротехнологии и когнитивные науки», первый отборочный этап, биология 8–9 классов «Кровеносная система»

Реши сам

1.3. Выбери все верные утверждения. Ответ – номера в порядке возрастания, без пробелов.

- **1** – Артерии всегда несут кровь, богатую кислородом.
- **2** – По венам кровь течёт к сердцу.
- **3** – Стенка левого желудочка толще, чем стенка правого.
- **4** – Через стенки капилляров кислород уходит к клеткам.
- **5** – Клапаны сердца пропускают кровь в обе стороны.
- **6** – Из правого желудочка кровь идёт в лёгкие.

1.4. Капля крови вышла из правого желудочка. Расставь по порядку места, через которые она пройдёт дальше: **А** – левое предсердие, **Б** – лёгочная артерия, **В** – аорта, **Г** – капилляры лёгких, **Д** – левый желудочек, **Е** – лёгочные вены. Ответ – шесть букв подряд.

Задача 1.3 Сколько крови за урок

работа сердца, ударный объём, перевод единиц

Школьник сидит на уроке. Его сердце делает 80 ударов в минуту, и за каждое сокращение левый желудочек выталкивает в аорту 60 мл крови (считай эти числа постоянными весь урок). Сколько литров крови сердце перекачает в аорту за урок длиной 45 минут? Ответ – целое число литров.

Дано: ЧСС = 80 уд/мин; ударный объём $V_{\text{уд}} = 60$ мл; $t = 45$ мин.

Найти: V за урок, л – ?

Решение

Шаг 1. Что происходит. Каждый удар – одна порция в 60 мл. Значит, объём крови равен размеру порции, умноженному на число порций. Число порций – это число ударов за урок.

Шаг 2. Объём за минуту. За минуту 80 ударов по 60 мл: $60 \cdot 80 = 4800$ мл.

Шаг 3. В литры. В литре 1000 мл: $4800 : 1000 = 4,8$ л в минуту.

Шаг 4. За урок. Урок – 45 минут, каждую минуту по 4,8 л: $4,8 \cdot 45 = 216$ л.

Можно и в другом порядке: за урок сердце ударит $80 \cdot 45 = 3600$ раз, и $3600 \cdot 60 = 216\,000$ мл = 216 л. Ответ тот же.

Двести с лишним литров – больше двадцати вёдер по 10 л. Много, но сердце не наливает новую кровь, а гоняет по кругу одну и ту же: за урок вся кровь школьника успевает пройти через сердце не один десяток раз.

 **Типичная ошибка.** Забыть перевести миллилитры в литры и записать в ответ 216 000. Бывает и другое: ЧСС дана в минуту, а время урока переводят в секунды или часы – тогда в одном действии встречаются разные единицы времени. Проверяй, чтобы «в минуту» умножалось на минуты.

Ответ: 216 л.

Реши сам

1.5. У взрослого в покое ударный объём 70 мл, ЧСС 72 уд/мин. Сколько литров крови сердце выталкивает в аорту за одну минуту? Ответ округли до сотых.

1.6. Та же идея, только для дыхания. Школьник на перемене делает 16 вдохов в минуту, за каждый вдох – 0,5 л воздуха, и столько же выдыхает. В выдыхаемом воздухе примерно 4 % углекислого газа. Сколько литров углекислого газа будет в воздухе, который он выдохнет за 10 минут?

Задача 1.4 Что покажет ЭМГ трицепса

мышцы-пары, сгибание и разгибание, ЭМГ

На уроке технологии Лиза шкурит доску: наждачная бумага натянута на брусок, и Лиза водит его правой рукой вперёд-назад, прижимая к доске. От себя – рука в локте разгибается, на себя – сгибается. Шкурка трётся о доску в обе стороны, так что усилие нужно и туда, и обратно. Лиза делает это давно и ловко: лишние мышцы у неё не напрягаются. На трицепс правой руки Лизе наклеили электроды ЭМГ. Что будет видно на ЭМГ трицепса, пока Лиза тянет брусок **на себя**?

- **1** – амплитуда уменьшится;
- **2** – амплитуда увеличится;
- **3** – амплитуда будет много раз то расти, то падать за одно движение;
- **4** – амплитуда не изменится.

Ответ – номер варианта.

Дано: движение «на себя» – сгибание в локте; электроды на трицепсе; движения ловкие, без лишнего напряжения.

Найти: как меняется амплитуда ЭМГ трицепса при движении на себя.

Решение

Шаг 1. Какое это движение. Брусок едет к Лизе – рука сгибается в локте.

Шаг 2. Кто сгибает. Сгибает руку в локте бицепс: он сокращается и тянет предплечье вверх-к себе. Трицепс нужен для обратного движения – разгибания.

Шаг 3. Что делает трицепс. В паре мышц, когда одна работает, другая расслаблена – особенно у человека, который движется ловко. Значит, при движении на себя трицепс расслаблен.

Шаг 4. Что покажет ЭМГ. Расслабленная мышца получает мало импульсов, волокон работает меньше, и амплитуда ЭМГ падает. Подходит вариант 1.

А на следующем движении «от себя» трицепс включится, и амплитуда вырастет. Если записать ЭМГ за минуту работы, на графике будут чередоваться «пачки» больших колебаний (от

себя) и затишья (на себя).

❗ Типичная ошибка. Рассуждать «рука работает – значит, все мышцы руки напряжены» и выбрать 2. Вторая ловушка – вариант 3: рост и спад действительно чередуются, но от движения к движению, за много взмахов. Пока идёт одно движение на себя, амплитуда трицепса низкая.

Ответ: 1.

По мотивам задачи НТО 2025/26, профиль «Нейротехнологии и когнитивные науки», первый отборочный этап, инженерный тур «Взаимодействие нервных центров»

Реши сам

1.7. Кирилл положил руку на стол и расслабил, а потом поднимает гантель, сгибая руку в локте. Электроды ЭМГ стоят на его бицепсе. Как изменится амплитуда ЭМГ бицепса, когда Кирилл начнёт поднимать гантель? **1** – уменьшится; **2** – увеличится; **3** – не изменится.

1.8. Модель бионической руки сжимает пальцы, пока амплитуда ЭМГ с мышц предплечья больше порога, и разжимает их, когда амплитуда меньше порога. Порог настроен так, что спокойная рука его не превышает. Маша по очереди: **1** – держит руку расслабленной на столе; **2** – изо всех сил сжимает кулак; **3** – держит кулак сжатым; **4** – разжимает кулак и расслабляет руку; **5** – шевелит пальцами ног. На каких шагах модель руки будет сжата? Ответ – номера без пробелов.

Задача 1.5 Какой датчик что меряет

датчики биосигналов, электрические и неэлектрические сигналы

а) Сопоставь сигнал и способ, которым его снимают.

Сигналы: **1** – ЭЭГ, **2** – ЭМГ, **3** – ЭКГ, **4** – пульс на пальце, **5** – дыхание, **6** – наклон головы.

Способы:

- **А** – пояс с датчиком растяжения на груди;
- **Б** – электроды на ободке на голове;
- **В** – датчик наклона, закреплённый на голове;
- **Г** – электроды на коже над мышцей;
- **Д** – светодиод и фотоприёмник на пальце;
- **Е** – электроды на груди или на запястьях.

б) Какие из сигналов 1–6 – это электричество, которое вырабатывают клетки самого тела? Ответ – номера в порядке возрастания, без пробелов.

Дано: 6 сигналов, 6 способов их снять.

Найти: а) пары «сигнал – способ»; б) номера электрических сигналов тела.

Решение

Шаг 1. Расшифруем названия. ЭЭГ – электроэнцефалограмма, «энцефало» – мозг, значит, электроды на голове: 1 – Б. ЭМГ – «мио» – мышца: 2 – Г. ЭКГ – «кардио» – сердце: 3 – Е.

Шаг 2. Неэлектрические сигналы. Датчик пульса светит в палец: 4 – Д. Грудь при вдохе расширяется, пояс растягивается: 5 – А. Наклон ловит датчик наклона: 6 – В. Все шесть букв разные, лишних нет.

Шаг 3. Часть 6. Электричество самого тела – у сигналов с «электро-» в названии: ЭЭГ, ЭМГ, ЭКГ, то есть 1, 2, 3. Их снимают электродами, которые только слушают. Пульс на пальце связан с сердцем, но датчик ловит свет, и напряжение делает сам датчик. Дыхание и наклон – механика: растяжение и поворот.

 **Типичная ошибка.** В части 6 добавить пульс: «это же сердце, а сердце электрическое». Электрический сигнал сердца – ЭКГ. Датчик на пальце видит другое: как толчок крови меняет прохождение света. Ещё путают ЭМГ и ЭЭГ из-за похожих букв – выручает расшифровка «мио» и «энцефало».

Ответ: а) 1 – Б, 2 – Г, 3 – Е, 4 – Д, 5 – А, 6 – В; б) 123.

Своя задача по программе сферы, модуль 2

Реши сам

1.9. В каких случаях датчик ловит электрическую активность клеток тела? Ответ – номера в порядке возрастания, без пробелов.

- **1** – ЭМГ-браслет на предплечье управляет моделью протеза.
- **2** – Фитнес-браслет светит зелёным светом в запястье и считает пульс.
- **3** – Ободок с электродами для БОС-игры «успокой ракету».
- **4** – Электроды ЭКГ на груди у спортсмена на беговой дорожке.
- **5** – Градусник под мышкой.
- **6** – Датчик CO₂ у носа.

1.10. Команда придумывает устройства для БОС-тренажёра космонавта. Сопоставь задумку и датчик, который для неё нужен.

Задумки: **1** – модель протеза сжимает пальцы, когда напрягаешь мышцы предплечья; **2** – лампа мигает в такт ударам сердца; **3** – ракета на экране летит плавно, пока человек медленно дышит; **4** – игра по ритмам мозга проверяет, спокойно ли человек сидит с закрытыми глазами.

Датчики: **А** – датчик CO₂ у носа, **Б** – ЭЭГ-ободок, **В** – модуль ЭМГ, **Г** – модуль пульса.

Задача 1.6 Рефлекс быстрее мысли

нервная система, рефлекторная дуга, скорость нервного импульса

На даче ты случайно тронул пальцем горячую кастрюлю. Сигнал «горячо!» бежит по чувствительному нерву от пальца до спинного мозга – это 0,9 м. Внутри спинного мозга передача с нейрона на нейрон занимает 1 мс. Потом сигнал бежит по двигательному нерву к мышце, ко-

торая отдёргивает руку, – это 0,6 м. Скорость сигнала в обоих нервах – 60 м/с. Сколько миллисекунд проходит от касания до того, как сигнал дойдёт до мышцы? Время на сокращение самой мышцы не учитывай.

Дано: $L_1 = 0,9$ м; $L_2 = 0,6$ м; $v = 60$ м/с; задержка в спинном мозге $t_0 = 1$ мс.
Найти: t , мс – ?

Решение

Шаг 1. Из чего складывается время. Дорога туда, остановка в спинном мозге, дорога обратно: $t = t_1 + t_0 + t_2$. Время на каждом отрезке – как у пешехода: путь, делённый на скорость.

Шаг 2. Дорога туда. $t_1 = 0,9 : 60 = 0,015$ с. Метры делим на метры в секунду – получаются секунды.

Шаг 3. В миллисекунды. В секунде 1000 мс: $0,015 \cdot 1000 = 15$ мс.

Шаг 4. Дорога обратно. $t_2 = 0,6 : 60 = 0,01$ с = 10 мс.

Шаг 5. Всё вместе. $t = 15 + 1 + 10 = 26$ мс.

Сравним с реакцией «через голову»: заметить и осознанно нажать кнопку человек успевает за 0,2–0,3 с, то есть за 200–300 мс. Рефлекс быстрее примерно в десять раз. Глазом такое не уловить – моргание длится дольше.

 **Типичная ошибка.** Сложить секунды с миллисекундами: $0,015 + 1 + 0,01 = 1,025$ – и ответить «1 мс». Прежде чем складывать, переведи всё в одни единицы. Ещё забывают про задержку в спинном мозге или считают только дорогу в одну сторону.

Ответ: 26 мс.

Своя задача по программе сферы, модуль 2

Реши сам

1.11. Ты наступил на шишку. Сигнал бежит от пальца ноги до головного мозга 1,5 м со скоростью 60 м/с. Через сколько миллисекунд он доберётся до мозга?

1.12.* Ты ударился пальцем ноги о ножку стола. Сигнал «удар» бежит по толстому волокну в миелине со скоростью 50 м/с, а сигнал «боль» – по тонкому волокну без миелина со скоростью 1 м/с. Путь обоих сигналов до спинного мозга – 1 м. Через сколько миллисекунд после сигнала «удар» до спинного мозга доберётся сигнал «боль»? Это объясняет, почему сначала чувствуешь удар, а боль приходит следом.

Задача 1.7* Увидит ли Arduino работу мозга

Биопотенциалы, микровольты и милливольты, АЦП Arduino, усиление

Команда хочет записывать ЭЭГ на Arduino Uno. Электроды на лбу дают сигнал с размахом 70 мкВ (размах – разница между самым высоким и самым низким значением). Команда `analogRead` делит напряжение от 0 до 5 В на ступеньки, одна ступенька – примерно 4,9 мВ.

а) Сколько ступенек займёт размах сигнала, если подать его на плату без усилителя? Округли до тысячных.

б) Во сколько раз нужно усилить сигнал, чтобы его размах занял 100 ступенек?

Дано: размах ЭЭГ 70 мкВ; одна ступенька АЦП 4,9 мВ; нужно 100 ступенек.

Найти: а) число ступенек без усилителя; б) во сколько раз усилить.

Решение

Шаг 1. Одни единицы. Сигнал дан в микровольтах, ступенька – в милливольтах. Переведём сигнал: в 1 мВ – 1000 мкВ, значит, $70 \text{ мкВ} = 70 : 1000 = 0,07 \text{ мВ}$.

Шаг 2. Часть а. Делим размах сигнала на размер ступеньки: $0,07 : 4,9 = 0,0143$, после округления 0,014. Даже одной ступеньки не набирается: плата будет показывать одно и то же число, и на графике выйдет ровная линия.

Шаг 3. Сколько нужно в милливольтах. 100 ступенек по 4,9 мВ: $100 \cdot 4,9 = 490 \text{ мВ}$.

Шаг 4. Во сколько раз усилить. Было 0,07 мВ, нужно 490 мВ: $490 : 0,07 = 7000$ раз.

Шаг 5. Проверка. Усиленный сигнал: $0,07 \cdot 7000 = 490 \text{ мВ} = 0,49 \text{ В}$. Это меньше 5 В, в диапазон платы помещается с большим запасом. Настоящие усилители для ЭЭГ и правда увеличивают сигнал в тысячи раз, так что число 7000 не выглядит фантастикой.

Тот же расчёт можно поручить компьютеру. Если Python тебе пока незнаком, читай код как запись на калькуляторе: слева от знака `=` – имя «ячейки», справа – что в неё положить; `/` – деление; `round` округляет (цифра 3 – до тысячных), `print` печатает результат; после `#` идут пояснения для человека.

```
razmah_mV = 70
stupen_mV = 4.9
razmah_mV = razmah_mV / 1000          # 0.07 мВ
print(round(razmah_mV / stupen_mV, 3)) # а) 0.014
nuzhno_mV = 100 * stupen_mV           # 490 мВ
print(round(nuzhno_mV / razmah_mV))    # б) 7000
```

⚠ Типичная ошибка. Не перевести микровольты: $70 : 4,9 = 14,3$ «ступеньки», и кажется, что усилитель не нужен. Ответ вышел больше верного в 1000 раз. В части б та же беда даёт $490 : 70 = 7$ – «усилить в 7 раз».

Ответ: а) 0,014 ступеньки – меньше одной; б) в 7000 раз.

Своя задача по программе сферы, модуль 2

Реши сам

1.13. Размах ЭКГ на коже груди – 1,2 мВ, размах ЭЭГ на коже головы – 40 мкВ. а) Во сколько раз напряжение пальчиковой батарейки (1,5 В) больше размаха ЭКГ? б) Во сколько раз размах ЭКГ больше размаха ЭЭГ?

1.14.* Размах ЭМГ на коже над мышцей – 3 мВ. Усилитель можно переключить на усиление в 500, 1000, 1500 или 2000 раз. Усиленный сигнал попадёт в Arduino, если его размах не больше 5 В. Какое наибольшее усиление из списка можно выбрать? Сколько ступенек `analogRead` (по 4,9 мВ) займёт тогда размах сигнала? Округли до целых.

Пульс и сердце: считаем по графику



Сердце – самый честный датчик в теле: побежал за автобусом – оно тут же застучало чаще, сел отдышаться – стало успокаиваться. Здесь ты научишься переводить удары в минуту в секунды между ударами и обратно, считать пульс по графику с датчика и по ленте ЭКГ, сравнивать пульс до и после нагрузки в процентах. И узнаешь, как браслет на руке видит пульс обычным светом. На отборе такая задача – график или короткая история с числами. В ответ идёт число: удары в минуту, секунды, проценты. Если вопросов в задаче несколько, то и чисел несколько, по одному на пункт, а там, где надо сопоставить, ответ – пары вида «1 – Г, 2 – Б».

2.1. Сердце и пульс

НА ПАЛЬЦАХ

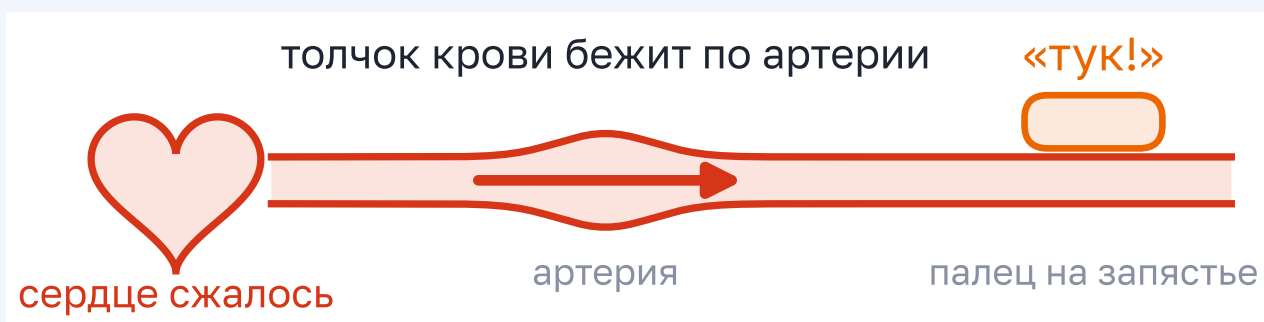


Рисунок 2.1 – Сердце сжалось – по артерии побежал толчок крови – палец на запястье его почувствовал

Пульс – это удар сердца, который добежал до твоего пальца по трубе-артерии.

Сердце – мышечный насос размером примерно с кулак (как оно устроено внутри, мы разбирали в главе 1). Оно сжимается, выталкивает порцию крови в большую артерию и расслабляется, чтобы снова набрать кровь. **Артерия** – сосуд, по которому кровь идёт от сердца. У неё упругие стенки, как у садового шланга. Каждая новая порция крови растягивает стенку, и этот толчок бежит по артерии дальше, до самых пальцев. Такой толчок и называется **пульсом**.

Где его нащупать? На запястье со стороны большого пальца: положи туда подушечки указательного и среднего пальцев и слегка надави. Можно на шее: сбоку от горла, чуть ниже угла нижней челюсти. Большим пальцем щупать не стоит – в нём свой заметный пульс, и ты будешь считать сам себя.

Сколько раз сердце сокращается за минуту, называют **частотой сердечных сокращений**, коротко **ЧСС**. Считают её в ударах в минуту: уд/мин, а в приборах часто пишут по-английски BPM (beats per minute). У здорового человека каждое сокращение сердца даёт один толчок на запястье, поэтому «пульс 72» и «ЧСС 72» в задачах означают одно и то же.

Сидеть и считать удары целую минуту скучно, да и ошибиться за минуту легко. Поэтому считают 15 или 30 секунд и пересчитывают на минуту. В минуте 60 секунд: насчитал N ударов за t секунд – умножь N на 60 и раздели на t .

$$\text{ЧСС} = \frac{N \cdot 60}{t},$$

где N – сколько ударов насчитал, t – за сколько секунд. За 15 секунд насчитал 18 ударов: $18 \cdot 60 / 15 = 72$ уд/мин. То же самое короче: 15 секунд – это четверть минуты, значит, умножаем на 4.

Есть у короткого счёта подводх. Сбился на один удар за 15 секунд – ответ уедет на 4 уд/мин, ведь каждый удар потом умножается на 4. Считал 30 секунд – ошибка всего 2 уд/мин. Поэтому врачи и медсёстры, когда нет прибора, считают подольше.

Какой пульс считают спокойным? Примерно, для задач: у взрослых в покое 60–90 уд/мин, у школьников 11–14 лет 70–100 уд/мин. Во сне сердце бьётся реже, на пробежке – заметно чаще, иногда до 150 и выше. Пульс меняется от волнения, жары, выпитого чая. Поэтому по одному числу никто не судит о здоровье – и мы в этой книге тоже не будем. Нормы здесь нужны, чтобы проверить себя: если у школьника в покое вышло 300 уд/мин, ошибка в расчёте, а не в сердце.

Запомни. ЧСС – удары за минуту. Считал 15 секунд – умножь на 4, считал 30 – умножь на 2, в общем случае $\text{ЧСС} = N \cdot 60 / t$. Спокойный пульс школьника – примерно 70–100 уд/мин.

2.2. Период и частота: удары в минуту и секунды между ударами

НА ПАЛЬЦАХ

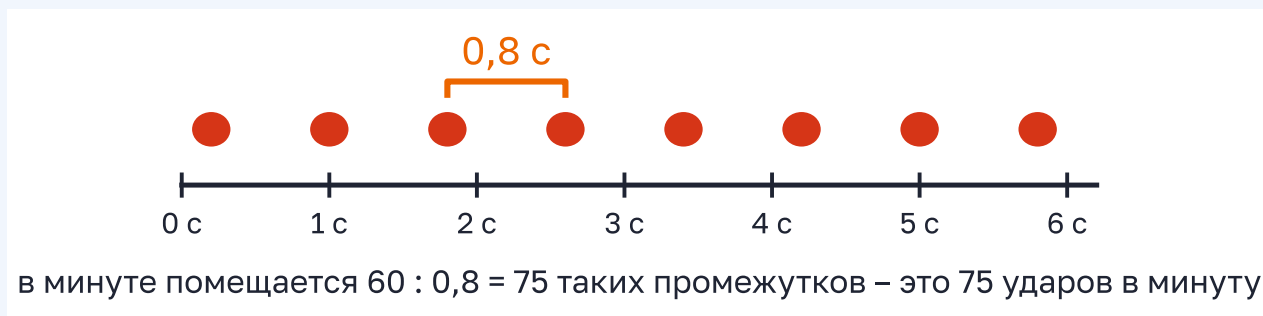


Рисунок 2.2 – Удары идут друг за другом через одинаковые промежутки

Знаешь, через сколько секунд бьёт сердце, – знаешь и сколько раз оно ударит за минуту.

Датчик пульса и прибор ЭКГ удары не «считают». Они видят, в какой момент был каждый удар, и измеряют время от одного удара до следующего. Это время называют **интервалом между ударами**, а у врачей – **RR-интервалом** (откуда такое название, станет ясно в разделе 2.3). В физике такое время называют **периодом** – это время одного полного повторения.

Как из интервала получить ЧСС? Спроси себя: сколько таких интервалов поместится в минуте? Если сердце бьёт каждые 0,8 с, то за 60 секунд оно ударит $60 : 0,8 = 75$ раз.

$$\text{ЧСС} = \frac{60}{T},$$

где T – интервал между ударами в **секундах**, а 60 – число секунд в минуте. Обратно, из ЧСС в интервал:

$$T = \frac{60}{\text{ЧСС}}.$$

Пульс 120 уд/мин – удар каждые $60 : 120 = 0,5$ с.

Приборы часто пишут интервал в **миллисекундах** (мс). Миллисекунда – тысячная доля секунды: 1 с = 1000 мс. Тогда удобно делить не 60, а 60 000 – столько миллисекунд в минуте:

$$\text{ЧСС} = \frac{60\,000}{T \text{ (в мс)}}.$$

Интервал 750 мс: $60\,000 : 750 = 80$ уд/мин. Можно и в два шага: 750 мс = 0,75 с, потом $60 : 0,75 = 80$.

Таблица 2.1

Интервал, с	Интервал, мс	ЧСС, уд/мин
1	1000	60
0,8	800	75
0,75	750	80
0,6	600	100

Интервал, с	Интервал, мс	ЧСС, уд/мин
0,5	500	120
0,4	400	150

Посмотри на таблицу сверху вниз: интервал становится короче, а пульс – чаще. Удары идут теснее, и за минуту их помещается больше. Интервал сократился вдвое (с 0,8 до 0,4 с) – пульс вырос вдвое (с 75 до 150). Такую связь называют **обратной пропорциональностью**.

В физике частоту считают не за минуту, а за секунду, в **герцах** (Гц): 1 Гц – одно колебание в секунду. Чтобы перевести пульс в герцы, раздели его на 60: 75 уд/мин – это $75:60 = 1,25$ Гц, чуть больше одного удара в секунду. Или сразу $1:T$: интервал 0,8 с даёт $1:0,8 = 1,25$ Гц. Про герцы, периоды и частоты любых сигналов подробно рассказано в главе 4.

Запомни. ЧСС = $60:T$, где T – в секундах, или $60\,000:T$, где T – в миллисекундах. Короче интервал – чаще пульс. Уд/мин в герцы – раздели на 60.

Совет. Перед делением посмотри на единицы интервала. Если в условии «640», а рядом «мс», то $60:640$ даст 0,09 «удара в минуту» – явная чепуха. Делить надо 60 000.

2.3. Читаем пульсограмму и ЭКГ

НА ПАЛЬЦАХ



Рисунок 2.3 – Пики – это удары. Считаем пики, меряем промежутки между ними

Ударов столько, сколько пиков, а промежутков на один меньше. Как столбов и пролётов в заборе.

Датчик выдаёт целый график: по горизонтали время, по вертикали – сила сигнала. Каждому удару сердца на графике соответствует **пик** – место, где линия резко поднимается и снова опускается. График пульса с датчика называют **пульсограммой**.

Первым делом разберись со шкалой. Посмотри, что подписано на горизонтальной оси и сколько секунд приходится на одну клетку. По вертикали для подсчёта пульса мерить ничего не нужно: нам важно, **когда** был пик. Высота пика на ответ не влияет.

Дальше два способа.

Способ 1 – по двум соседним пикам. Найди время двух пиков подряд, вычти одно из другого – получишь интервал T . Потом ЧСС = $60 : T$. Быстро, но если этот один интервал случайно вышел длиннее или короче других, ответ уедет.

Способ 2 – по многим пикам сразу. Он точнее. Возьми первый и последний пик на графике, найди время между ними и раздели на число промежутков:

$$T_{\text{ср}} = \frac{t_{\text{последнего}} - t_{\text{первого}}}{N - 1}.$$

Здесь N – число пиков, а $N - 1$ – число промежутков между ними. Пять столбов забора – четыре пролёта; пять пиков – четыре интервала. Если поделить на N , а не на $N - 1$, интервал выйдет меньше настоящего, а пульс – больше. Это ошибка номер один во всей главе.

Пример. Пики были в моменты 0,35; 1,15; 1,9; 2,7 и 3,5 с. Промежутки: 0,8; 0,75; 0,8; 0,8 с – почти одинаковые. Средний: $(3,5 - 0,35) : 4 = 3,15 : 4 = 0,7875$ с. ЧСС: $60 : 0,7875 = 76,2 \approx 76$ уд/мин.

Промежутки у живого сердца всегда чуть-чуть разные: на вдохе оно немного ускоряется, на выдохе замедляется. Поэтому средний интервал по многим ударам честнее, чем один.

Тот же расчёт на Python – если захочешь проверить себя на длинном списке пиков:

```
peaks = [0.35, 1.15, 1.9, 2.7, 3.5]      # когда были пики, с
for i in range(1, len(peaks)):
    print(round(peaks[i] - peaks[i - 1], 2)) # промежутки между соседними пиками
avg = (peaks[-1] - peaks[0]) / (len(peaks) - 1) # средний промежуток, с
print(round(avg, 4))
print(round(60 / avg))                    # ЧСС, уд/мин
```

Первая строка – **список** времён пиков. Цикл `for` проходит по номерам пиков начиная со второго (`i` от 1 до 4) и печатает разность соседей: `peaks[i]` – текущий пик, `peaks[i - 1]` – предыдущий. `peaks[-1]` – последний элемент списка, `peaks[0]` – первый, `len(peaks)` – сколько всего пиков (5). Программа напечатает 0.8, 0.75, 0.8, 0.8, потом средний промежуток 0.7875 и ЧСС 76.

ЭКГ: электрический портрет удара

НА ПАЛЬЦАХ



Рисунок 2.4 – Лента едет под пером с постоянной скоростью, перо рисует удары

На ленте ЭКГ расстояние – это время. При скорости 25 мм/с каждые 25 мм ленты – одна секунда.

ЭКГ (электрокардиограмма) – запись того, как по сердцу пробегает электрический сигнал. Прибор берёт его электродами с груди, рук и ног. Сигнал слабенький, около милливольт, зато на ЭКГ каждый удар выглядит одинаково, его ни с чем не спутаешь.



Рисунок 2.5 – Два удара на ЭКГ. Самый высокий острый зубец – R

Зубцы на ЭКГ врачи называют латинскими буквами:

- **P** – маленький горбик: сжимаются предсердия, верхние камеры сердца;
- **Q, R, S** – острая «иголка» вниз-вверх-вниз: сжимаются желудочки, главные насосы. **R** – самый высокий зубец;
- **T** – пологий горбик после иголки: желудочки восстанавливаются перед следующим ударом.

Для счёта пульса нужен только зубец R: он острый и высокий, его трудно спутать с чем-то ещё. Время от одного R до следующего и есть тот самый **RR-интервал** из раздела 2.2.

ЭКГ часто печатают на бумажной ленте в клетку. Лента едет под пером с постоянной скоростью, обычно 25 мм в секунду; скорость всегда указывают на ленте или в условии. При 25 мм/с каждый миллиметр – это $1:25 = 0,04$ с, а большая клетка в 5 мм – $5 \cdot 0,04 = 0,2$ с. Расстояние между зубцами R в 20 мм – это $20:25 = 0,8$ с, то есть 75 уд/мин.

Совет. Быстрый способ для ленты 25 мм/с: 300 раздели на число больших клеток между зубцами R. Между R четыре большие клетки – пульс $300:4 = 75$. Откуда 300? Одна большая клетка – 0,2 с, а $60:0,2 = 300$.

Пульсовая волна с пальца

Оптический датчик на пальце рисует другую кривую – **пульсовую волну**. Она плавнее ЭКГ: без острой иголки, с крутым подъёмом и медленным спуском.

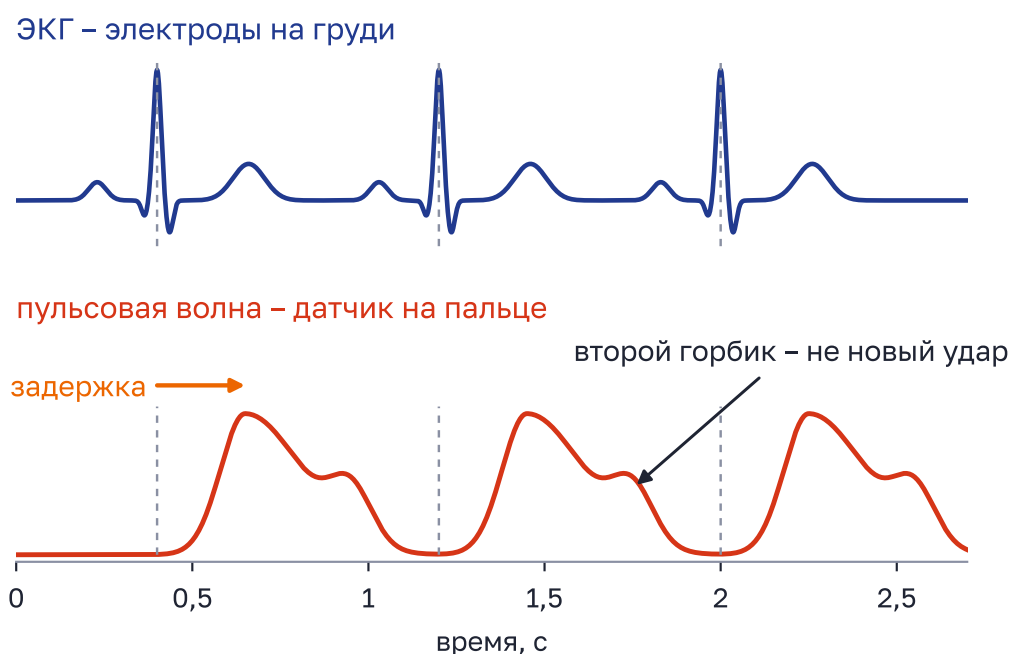


Рисунок 2.6 – Сверху ЭКГ, снизу пульсовая волна того же человека. Пунктир – моменты зубцов R

На рисунке видны две особенности. Пик пульсовой волны приходит **позже** зубца R: толчку крови нужно время, чтобы добежать от сердца до пальца. Но задержка одинаковая у каждого удара, поэтому интервалы между пиками те же, что на ЭКГ, и пульс получится тот же. А на спуске у каждого пика есть **второй горбик** поменьше. Это отголосок того же удара, а не новый удар. Кто его посчитает, получит пульс вдвое больше настоящего.

Запомни. N пиков – $N - 1$ промежутков. Средний интервал = (время последнего пика – время первого) : $(N - 1)$. На ЭКГ считаем зубцы R, на пульсовой волне – только высокие пики, без вторых горбиков.

2.4. Пульс при нагрузке: сравниваем

НА ПАЛЬЦАХ



Рисунок 2.7 – Пульс сидя, сразу после бега и после отдыха

Был 75, стал 150: прибавилось ещё столько же. Это вдвое больше, или на 100 %.

Когда ты бежишь на лыжах или поднимаешься на пятый этаж, мышцы работают сильнее и им нужно больше кислорода. Кислород приносит кровь, поэтому сердце начинает качать её быстрее: бьётся чаще и выталкивает за раз больше. Закончил нагрузку – пульс идёт вниз, но постепенно, за несколько минут.

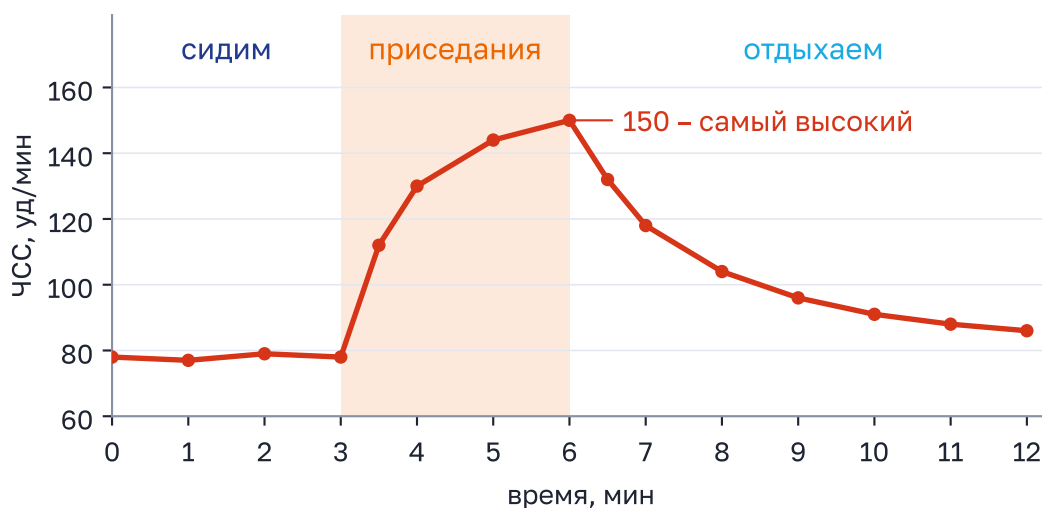


Рисунок 2.8 – ЧСС во время опыта: три минуты сидим, три минуты приседаем, потом отдыхаем. Точки – замеры

По такому графику задают вопросы трёх видов. Пусть было 78 уд/мин (сидим), стало 150 (конец приседаний).

- **На сколько вырос** – вычитаем: $150 - 78 = 72$ уд/мин.
- **Во сколько раз** – делим: $150 : 78 = 1,92$, почти вдвое.
- **На сколько процентов** – делим прибавку на то, что было до, и умножаем на 100:

$$\text{изменение, \%} = \frac{\text{новое} - \text{старое}}{\text{старое}} \cdot 100.$$

У нас $(150 - 78) : 78 \cdot 100 = 92,3 \%$.

Теперь ловушка, на которой спотыкаются многие взрослые. Через 6 минут отдыха (на 12-й минуте графика) пульс опустился до 86. На сколько процентов он снизился? «До» здесь – это 150, а вовсе не 78: снижение считаем от того места, откуда падали.

$(150 - 86) : 150 \cdot 100 = 42,7 \%$. Если поделить на 78, выйдет 82 % – число, которое к этому вопросу не имеет отношения.

Запомни. «На сколько» – вычитаем, «во сколько раз» – делим, «на сколько процентов» – прибавку делим на значение «до». Пульс падает после нагрузки – «до» равно самому высокому значению.

Сколько крови качает сердце

НА ПАЛЬЦАХ

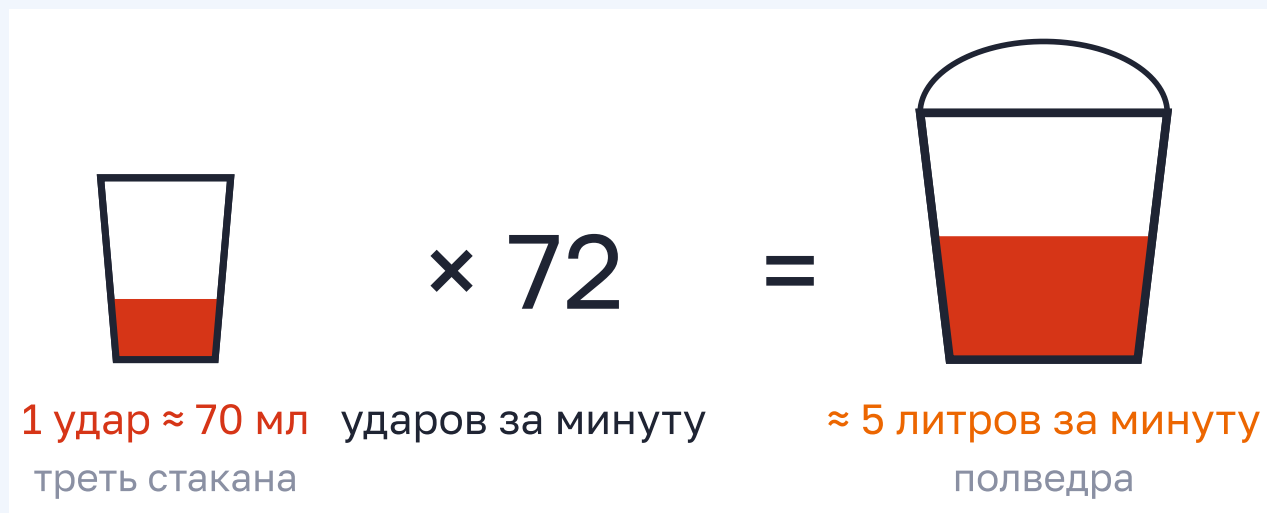


Рисунок 2.9 – Сколько крови сердце перекачивает за минуту в покое

Треть стакана за удар – вроде бы немного. Но за минуту сердце делает это 72 раза, и набирается полведра.

Сердце можно мерить не только ударами, но и литрами. За одно сокращение взрослое сердце в покое выталкивает около 70 мл крови – примерно треть стакана. Это **ударный объём**. Умножишь его на ЧСС – получишь **минутный объём**: сколько крови проходит через сердце за минуту.

$$\text{минутный объём} = \text{ударный объём} \cdot \text{ЧСС}.$$

При пульсе 72: $70 \cdot 72 = 5040$ мл за минуту. В литре 1000 мл, поэтому $5040 : 1000 = 5,04$ л – около пяти литров, полведра. За две минуты сердце перекачает почти целое ведро. При нагрузке растут оба множителя, и минутный объём увеличивается в несколько раз.

Запомни. Минутный объём = ударный объём · ЧСС. Ударный объём в миллилитрах даёт ответ в миллилитрах, для литров раздели на 1000.

2.5. Как браслет видит пульс

НА ПАЛЬЦАХ



Рисунок 2.10 – Светодиод светит сквозь палец, фотоприёмник ловит, сколько света прошло

При ударе в сосудах пальца крови чуть больше. Кровь поглощает свет, и до приёмника его доходит меньше.

Фитнес-браслет не колет палец и не ставит электроды. Внутри у него **светодиод** – маленькая лампочка – и **фотоприёмник** – деталь, сигнал которой зависит от того, сколько света на неё падает. Такой способ называется **фотоплетизмографией** (ФПГ). Слово длинное, а смысл простой: «записывать светом, как наполняется орган».

Кровь хорошо поглощает свет. После каждого удара сердца в мелкие сосуды под кожей приходит новая порция крови, и они чуть-чуть раздуваются. Крови в этом месте становится больше – света до приёмника доходит меньше. Кровь уходит – света снова больше. Так яркость на приёмнике «дышит» в такт сердцу, а программа рисует из неё пульсовую волну.

Датчики бывают двух видов.

- **На просвет:** светодиод с одной стороны пальца, приёмник с другой. Так устроена прищепка, которую надевают на палец в больнице. В ней светят красный и невидимый инфракрасный светодиоды, и заодно прибор оценивает, сколько в крови кислорода.
- **На отражение:** светодиод и приёмник лежат рядом, свет уходит в кожу и частично отражается обратно. Так устроены браслеты и умные часы, у многих из них светодиоды зелёные – загляни под свои часы, если они у тебя есть.

Дальше работает электроника. Приёмник выдаёт напряжение, плата Arduino измеряет его командой `analogRead` и получает число от 0 до 1023. Числа летят в компьютер, и **плоттер** (онограф в Arduino IDE и в симуляторе) рисует из них кривую. Программа ищет пики: удар зачитывает, когда сигнал поднялся выше выбранного **порога**, и замечает время удара. Время

между ударами в миллисекундах, 60 000 разделить на него – и на экране пульс. Как выбрать порог и почему второй горбик ломает счёт, разобрано в главе 4, а готовая программа-пульсометр – в справочнике.

В симуляторе, где занимается курс сферы, настоящего пальца нет. Там пульсовую волну делают программой: синусоида – плавная волна, у которой крутилками-потенциометрами меняют высоту и частоту. Считать пики по такой «идеальной» волне легко. У живого человека сигнал неровный, и стоит пошевелить рукой, как график прыгает сильнее, чем от любого удара. Поэтому пульс браслетом меряют, когда рука лежит спокойно.

Запомни. Оптический датчик – светодиод плюс фотоприёмник. Удар – в сосудах больше крови – света до приёмника доходит меньше. Пики на графике – удары, время между пиками даёт пульс.

2.6. Разбираем задачи

Задача 2.1 Пульс на уроке

разминка · ЧСС по числу ударов

На уроке биологии Лиза нащупала пульс на запястье и за 15 секунд насчитала 21 удар. Какова её ЧСС? Ответ дай в ударах в минуту.

Дано: $N = 21$ удар; $t = 15$ с.

Найти: ЧСС – ?

Решение

Шаг 1. Какую долю минуты считала Лиза. В минуте 60 с. $60 : 15 = 4$ – значит, 15 секунд – это четверть минуты, и за целую минуту ударов будет вчетверо больше.

Шаг 2. Пересчёт на минуту. $21 \cdot 4 = 84$ уд/мин. По формуле то же самое:
 $21 \cdot 60 / 15 = 1260 / 15 = 84$.

Для девочки 11–14 лет в покое это обычный пульс, он попадает в диапазон примерно 70–100.

 **Типичная ошибка.** Сдать ответ «21». Это удары за 15 секунд, а спрашивали про минуту. Другой вариант той же ошибки – умножить на 15 вместо 4: 315 уд/мин у человека не бывает даже на самой тяжёлой тренировке.

Ответ: 84 уд/мин.

Своя задача по программе сферы, модуль 2

Реши сам

2.1. Перед контрольной Миша посчитал пульс за 30 секунд и насчитал 38 ударов. Какова его ЧСС в ударах в минуту?

2.2. Браслет показывает пульс 90 уд/мин. Сколько ударов насчитает человек, если будет считать пульс 20 секунд?

Задача 2.2 Что показывает браслет

разминка · интервал между ударами, ЧСС, герцы

Приложение фитнес-браслета показывает время между последними двумя ударами сердца Егора: 625 мс. а) Какова ЧСС Егора в уд/мин? б) Какова частота ударов в герцах?

Дано: $T = 625$ мс.

Найти: ЧСС в уд/мин – ?; частоту f в Гц – ?

Решение

Шаг 1. Переводим в секунды. В секунде 1000 мс: $625 : 1000 = 0,625$ с.

Шаг 2. Сколько интервалов в минуте. ЧСС = $60 : 0,625 = 96$ уд/мин. Можно и без перевода: $60\,000 : 625 = 96$ – тот же ответ.

Шаг 3. Герцы. Частота в герцах – удары за секунду: $96 : 60 = 1,6$ Гц. Или $1 : 0,625 = 1,6$ Гц.

Интервал чуть больше полсекунды – значит, ударов в секунду чуть меньше двух. Полтора с небольшим как раз подходит.

⚠ Типичная ошибка. Разделить 60 на 625 и получить 0,096. Миллисекунды не переведены в секунды. Если у тебя пульс вышел меньше одного удара в минуту, остановись – проверь единицы.

Ответ: а) 96 уд/мин; б) 1,6 Гц.

Своя задача по программе сферы, модуль 6

Реши сам

2.3. На пульсограмме соседние пики отстоят друг от друга на 0,48 с. Найди ЧСС в ударах в минуту.

2.4. После пробежки у Веры пульс 120 уд/мин. Через сколько миллисекунд следует один удар за другим? Какова частота ударов в герцах?

Задача 2.3 Пульс по клеткам

пульсограмма, пики, средний интервал

Маша надела датчик пульса на палец сразу после перемены. На рисунке – запись с датчика. Одна клетка по горизонтали – 0,2 с. Найди ЧСС Маши в ударах в минуту.



Рисунок 2.11 – Запись оптического датчика пульса. По горизонтали время, одна клетка – 0,2 с

Дано: график; 1 клетка = 0,2 с.

Найти: ЧСС – ?

Решение

Шаг 1. Какие пики считать. У каждого удара на рисунке высокий пик и сразу за ним горбик пониже. Горбик – отголосок того же удара (раздел 2.3), его не считаем. Высоких пиков – **7**.

Шаг 2. Время первого и последнего пика. Первый пик – на второй линии сетки после нуля: $2 \cdot 0,2 = 0,4$ с. Последний – на отметке 4 с. Не все промежуточные пики стоят точно на линиях сетки, поэтому берём крайние – они на линиях.

Шаг 3. Сколько промежутков. 7 пиков – $7 - 1 = 6$ промежутков.

Шаг 4. Средний интервал. $(4,0 - 0,4) : 6 = 3,6 : 6 = 0,6$ с. Это три клетки – на рисунке соседние пики и правда отстоят примерно на три клетки.

Шаг 5. ЧСС. $60 : 0,6 = 100$ уд/мин.

Пульс 100 сразу после беготни на перемене – жизненное число. Посиди Маша минут пять за партой, датчик показал бы ближе к 80.

⚠ Типичная ошибка. Здесь их сразу три. Посчитать горбики за удары – 14 «пиков», и пульс выйдет вдвое больше, за 200. Поделить 3,6 с на 7 вместо 6 – выйдет 117. Или взять формулу «удары за время» для всего окна: $7 \cdot 60 : 4,6 = 91$ – здесь мешает то, что до первого пика и после последнего на графике есть «пустые» кусочки, которые к интервалам не относятся.

Ответ: 100 уд/мин.

Своя задача по программе сферы, модуль 6

Реши сам

2.5. ЭКГ записали на ленту, которая движется со скоростью 50 мм/с (вдвое быстрее обычного). Расстояние между соседними зубцами R – 40 мм. Найди ЧСС в ударах в минуту.

2.6. На пульсограмме 7 высоких пиков. Первый был в момент 0,6 с, последний – в момент 4,44 с. Найди средний интервал между ударами в секундах и ЧСС в ударах в минуту (ЧСС округли до целых).

Задача 2.4 Лента ЭКГ

лента ЭКГ, мм в секунды, среднее, максимум

Кусок ленты ЭКГ записан при скорости 25 мм/с. На рисунке подписаны расстояния между соседними зубцами R. а) Найди средний RR-интервал в секундах и среднюю ЧСС. б) Найди самую большую «мгновенную» ЧСС – посчитанную по одному интервалу. Округли до целых.

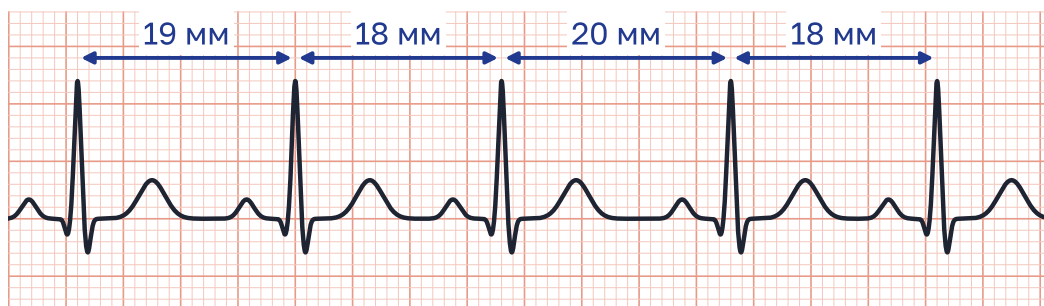


Рисунок 2.12 – Кусок ленты ЭКГ, 25 мм/с. Над зубцами R – расстояния между ними

Дано: скорость ленты 25 мм/с; расстояния 19, 18, 20, 18 мм.

Найти: $T_{\text{ср}}$, средняя ЧСС, наибольшая ЧСС – ?

Решение

Шаг 1. Сколько секунд в одном миллиметре. Лента за 1 с проезжает 25 мм, значит, 1 мм – это $1:25 = 0,04$ с.

Шаг 2. Средний интервал. Зубцов R пять, промежутков четыре. Всего $19 + 18 + 20 + 18 = 75$ мм, в среднем $75:4 = 18,75$ мм. В секундах: $18,75 \cdot 0,04 = 0,75$ с.

Шаг 3. Средняя ЧСС. $60:0,75 = 80$ уд/мин. Проверка по совету из раздела 2.3: $18,75$ мм – это $18,75:5 = 3,75$ больших клетки, и $300:3,75 = 80$ – оба способа дали одно число.

Шаг 4. Самая большая мгновенная ЧСС. Пульс тем чаще, чем **короче** интервал. Самый короткий – 18 мм, то есть $18 \cdot 0,04 = 0,72$ с. ЧСС: $60:0,72 = 83,3 \approx 83$ уд/мин.

Мгновенные значения гуляют от 75 (по интервалу 20 мм) до 83, а среднее 80 оказалось между ними. Выскочи среднее за эти пределы – ищи ошибку в счёте.

⚠ Типичная ошибка. Для «самой большой ЧСС» взять самый большой интервал, 20 мм, и получить 75. Большой интервал – это медленный пульс. Вторая ловушка – умножить миллиметры на 25 вместо деления: $18,75 \cdot 25 = 468$ «секунд» между ударами, почти восемь минут.

Ответ: а) 0,75 с, 80 уд/мин; б) 83 уд/мин.

Реши сам

2.7. ЭКГ спящего человека записана при скорости 25 мм/с. Расстояния между соседними зубцами R: 24, 26, 25 и 25 мм. Найди среднюю ЧСС и самую маленькую мгновенную ЧСС (округли до целых).

2.8. Браслет записал пять интервалов между ударами: 800, 750, 820, 780 и 850 мс. Найди среднюю ЧСС по среднему интервалу. Потом найди наибольшую и наименьшую мгновенные ЧСС (каждую округли до целых).

Задача 2.5 Тридцать приседаний

проценты, сравнение до и после нагрузки

На физкультуре Даня трижды посчитал пульс, каждый раз по 15 секунд: сидя – 20 ударов, сразу после 30 приседаний – 30 ударов, через 2 минуты отдыха – 23 удара. а) На сколько процентов пульс вырос после приседаний? б) На сколько процентов он снизился за 2 минуты отдыха? Округли до десятых.

Дано: за 15 с: 20, 30, 23 удара.

Найти: рост в %; снижение в %.

Решение

Шаг 1. Все замеры – в уд/мин. 15 секунд – четверть минуты, умножаем на 4: сидя $20 \cdot 4 = 80$, после приседаний $30 \cdot 4 = 120$, после отдыха $23 \cdot 4 = 92$ уд/мин.

Шаг 2. Рост. Прибавка $120 - 80 = 40$ уд/мин. Сравниваем с тем, что было до приседаний, с 80: $40 : 80 \cdot 100 = 50$ %.

Шаг 3. Снижение. Теперь «до» – это 120, с него пульс и падал. Убавилось $120 - 92 = 28$ уд/мин: $28 : 120 \cdot 100 = 23,33$ % $\approx 23,3$ %.

Проценты можно было считать и прямо по ударам за 15 секунд: $(30 - 20) : 20 = 0,5$ – те же 50 %. Умножение на 4 при делении сокращается.

⚠ Типичная ошибка. Во втором вопросе поделить на 80: $28 : 80 \cdot 100 = 35$ %. Но за две минуты пульс упал со 120, а не с 80. Одни и те же 28 ударов – это 35 % от меньшего числа и только 23,3 % от большего.

Ответ: а) на 50 %; б) на 23,3 %.

Своя задача по программе сферы, модуль 2

Реши сам

2.9. Сидя за партой, Оля насчитала за 15 секунд 18 ударов пульса, а после лыжного круга – 30 ударов за 15 секунд. На сколько процентов вырос её пульс? Округли до целых.

2.10. В конце забега пульс спортсмена был 150 уд/мин. За первую минуту отдыха он снизился на 20 %, за вторую – ещё на 10 % от того, что было после первой минуты. Каким стал пульс? На сколько процентов он снизился за две минуты по сравнению со 150?

Задача 2.6 Чьё это сердце?

интервал и ЧСС, сопоставление

Зоолог записал ЭКГ четырёх животных и выписал для каждого интервал между ударами сердца:

1. 0,3 с; 2) 2,1 с; 3) 120 мс; 4) 800 мс.

Примерная ЧСС в покое (по данным для этой задачи): **А** – человек, 60–90 уд/мин; **Б** – слон, 25–35 уд/мин; **В** – домовая мышь, 450–600 уд/мин; **Г** – кролик, 150–250 уд/мин.

Сопоставь номера записей и животных. Ответ запиши в виде «1 – А, 2 – Б, ...» по возрастанию номеров.

Дано: четыре интервала; четыре диапазона ЧСС.

Найти: пары «запись – животное».

Решение

Шаг 1. Все интервалы – в секундах. 120 мс = 0,12 с; 800 мс = 0,8 с. Первые два уже в секундах.

Шаг 2. ЧСС каждой записи. $ЧСС = 60 : T$:

- запись 1: $60 : 0,3 = 200$ уд/мин;
- запись 2: $60 : 2,1 = 28,6$ уд/мин;
- запись 3: $60 : 0,12 = 500$ уд/мин;
- запись 4: $60 : 0,8 = 75$ уд/мин.

Шаг 3. Сопоставляем с диапазонами. 200 – между 150 и 250, это кролик (Г). 28,6 – слон (Б). 500 – мышь (В). 75 – человек (А).

Можно было угадать и без расчёта: чем меньше животное, тем чаще бьётся его сердце. Самый короткий интервал – у мыши, самый длинный – у слона. Расчёт просто подтверждает догадку.

⚠ Типичная ошибка. Не заметить «мс» у записей 3 и 4. Тогда $60 : 120 = 0,5$ удара в минуту, и мышь с человеком не подходят ни под один диапазон. Когда числа в одном списке записаны в разных единицах, переведи всё к одним.

Ответ: 1 – Г, 2 – Б, 3 – В, 4 – А.

Реши сам

2.11. Возьми из задачи 2.6 диапазон ЧСС мышцы: от 450 до 600 уд/мин. В каких пределах лежит интервал между ударами её сердца? Ответ дай в миллисекундах, округли до целых.

2.12. Браслет папы делит пульс на зоны: **А** – «сон», 50–65 уд/мин; **Б** – «покой», 65–90 уд/мин; **В** – «бег», 130–170 уд/мин. За день браслет записал три интервала между ударами: 1) 0,42 с; 2) 1,1 с; 3) 0,8 с. Сопоставь номера интервалов и зоны. Ответ в виде «1 – А, 2 – Б, ...».

Задача 2.7* Сколько крови качает сердце

минутный объём, интервал, единицы, во сколько раз

Ударный объём – сколько крови сердце выталкивает за одно сокращение. **Минутный объём** – сколько за минуту. Папа надел браслет и пошёл на лыжную гонку. В покое браслет показывал интервал между ударами 0,8 с, а ударный объём у папы (по данным врача для этой задачи) – 70 мл. На дистанции интервал сократился до 0,4 с, а ударный объём вырос до 100 мл. а) Каков минутный объём в покое, в литрах? б) Каков он на дистанции, в литрах? в) Во сколько раз он вырос? Этот ответ округли до десятых.

Дано: покой – $T_1 = 0,8$ с, $V_1 = 70$ мл; гонка – $T_2 = 0,4$ с, $V_2 = 100$ мл.

Найти: минутный объём в покое и на гонке (л); во сколько раз вырос.

Решение

Шаг 1. Строим цепочку. Минутный объём = ударный объём · ЧСС. Ударный объём дан, а ЧСС нет – её получим из интервала. Выходит: интервал → ЧСС → минутный объём в мл → в литры.

Шаг 2. ЧСС в покое и на гонке. $60:0,8 = 75$ уд/мин; $60:0,4 = 150$ уд/мин.

Шаг 3. Минутный объём в покое. $70 \cdot 75 = 5250$ мл за минуту. В литре 1000 мл:
 $5250:1000 = 5,25$ л. Это близко к тем «около 5 литров», о которых шла речь в разделе 2.4.

Шаг 4. Минутный объём на гонке. $100 \cdot 150 = 15\,000$ мл = 15 л.

Шаг 5. Во сколько раз. Делим точные значения, не округлённые: $15:5,25 = 2,857 \approx 2,9$.

Проверка другим путём. Пульс вырос в $150:75 = 2$ раза, ударный объём – в $100:70 = 1,43$ раза. Минутный объём – произведение, значит, растёт в $2 \cdot 1,43 = 2,86$ раза, как и получилось в шаге 5.

⚠ Типичная ошибка. Умножить ударный объём на интервал: $70 \cdot 0,8 = 56$. Интервал – это секунды между ударами, а нам нужно число ударов за минуту. Ещё одна: заранее округлить 5,25 л до 5,3 л и поделить $15:5,3 = 2,83 \approx 2,8$ – округление в середине решения испортило последнюю цифру.

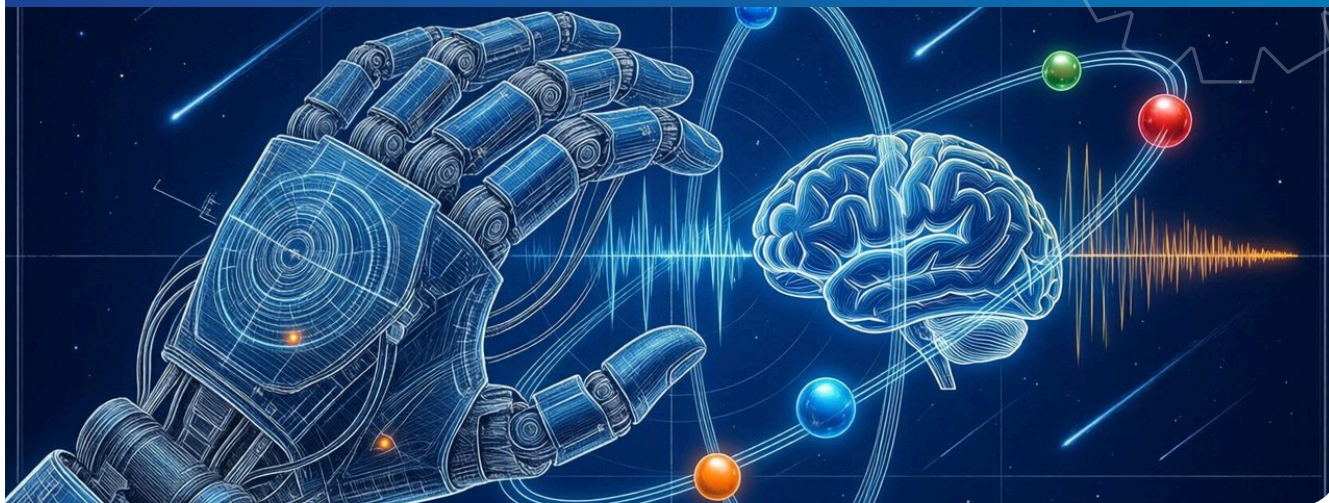
Ответ: а) 5,25 л; б) 15 л; в) в 2,9 раза.

Реши сам

2.13.* У семиклассника в покое интервал между ударами 0,75 с, ударный объём 60 мл. На тренировке ЧСС 160 уд/мин, ударный объём 85 мл. а) Во сколько раз вырос минутный объём? Округли до десятых. б) Сколько литров крови перекачает его сердце за 5 минут тренировки?

2.14.* В покое минутный объём у человека 6,3 л при ЧСС 75 уд/мин. При нагрузке – 18 л при ЧСС 150 уд/мин. Найди ударный объём в покое и при нагрузке в миллилитрах. На сколько процентов вырос ударный объём? Округли до десятых.

Мышцы и мозг: ЭМГ и ЭЭГ



Сожми кулак – и модель руки на столе сожмёт пальцы. Закрой глаза и успокойся – и на экране поднимется ракета. Оба фокуса держатся на электричестве мышц и мозга и на одном приёме: сравнить сигнал с порогом. Задачи здесь – график или столбик чисел и короткий вопрос: какой порог поставить, сколько раз сработает, на какой угол повернётся сервопривод, какой это ритм.

3.1. ЭМГ: электричество работающей мышцы

НА ПАЛЬЦАХ

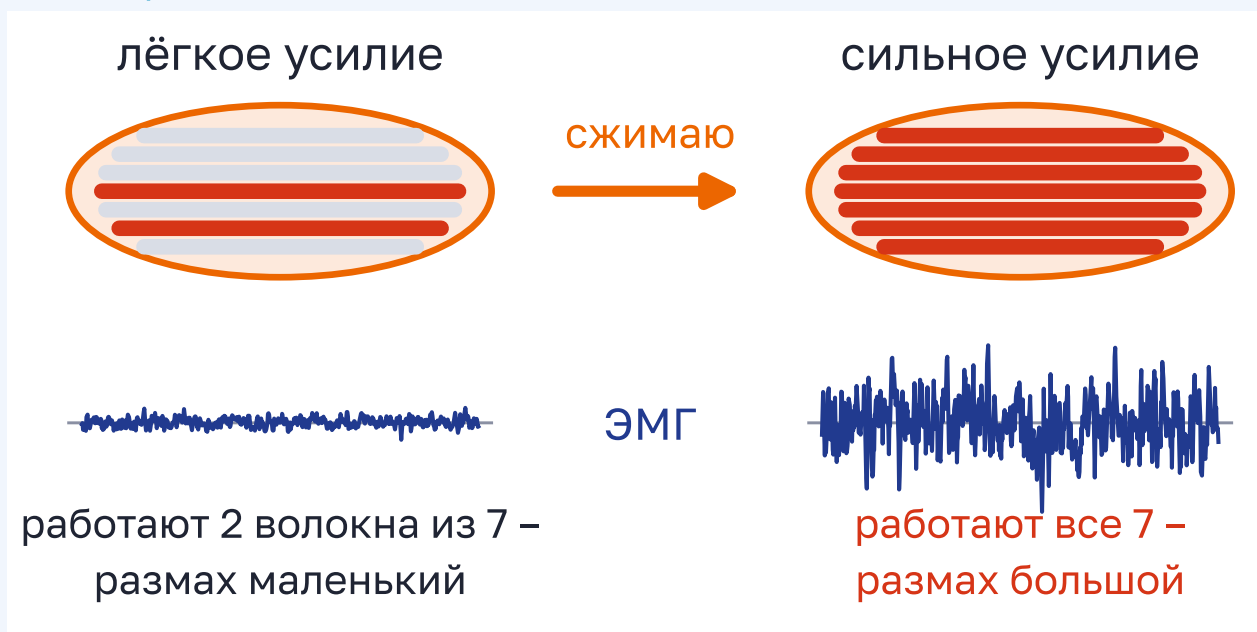


Рисунок 3.1 – Мышца – пучок волокон. Чем больше волокон работает, тем больше размах ЭМГ

Поднимаешь ручку – работают несколько волокон, тащишь ведро воды на даче – почти все. Электроды на коже слышат их всех разом: чем больше волокон включилось, тем сильнее «шумит» сигнал.

В главе 1 мы выяснили, что мышца сокращается по команде нерва и по её волокнам при этом бегут электрические импульсы. Каждое волокно даёт совсем крошечный всплеск. Электроды на коже над мышцей ловят сумму всплесков от сотен волокон сразу – эта запись и называется **ЭМГ**, электромиограмма.

Как выглядит ЭМГ? Совсем не так, как пульс. У пульса – аккуратные пики через равные промежутки. У ЭМГ – «лохматая» полоса: сигнал сотни раз в секунду дёргается вверх и вниз вокруг среднего уровня, и никакого порядка в этих дёрганиях нет. Волокна включаются вразнобой, каждое в свой момент. Поэтому смотрят на ЭМГ не по отдельным пикам, а по **размаху** – насколько широкая полоса.

А размах зависит от усилия:

- рука лежит расслабленно – полоса почти плоская, остаётся только слабый шум;
- сжимаешь кулак легонько – полоса становится шире;
- сжимаешь изо всех сил – полоса самая широкая: в работу вступили почти все волокна, и импульсы по ним идут чаще.

На этом свойстве ЭМГ стоит вся первая половина главы: **сильнее напряжена мышца – больше амплитуда ЭМГ**. Точной формулы здесь нет – у каждого человека и при каждой установке электродов свои числа. Но правило «больше усилие – больше размах» работает обычно. Это не точные весы: в два раза выше ЭМГ не значит, что мышца в два раза сильнее. Размах зависит ещё от положения тела, усталости и того, как лежат электроды.

Огибающая: плавная линия вместо «лохматой» полосы

НА ПАЛЬЦАХ

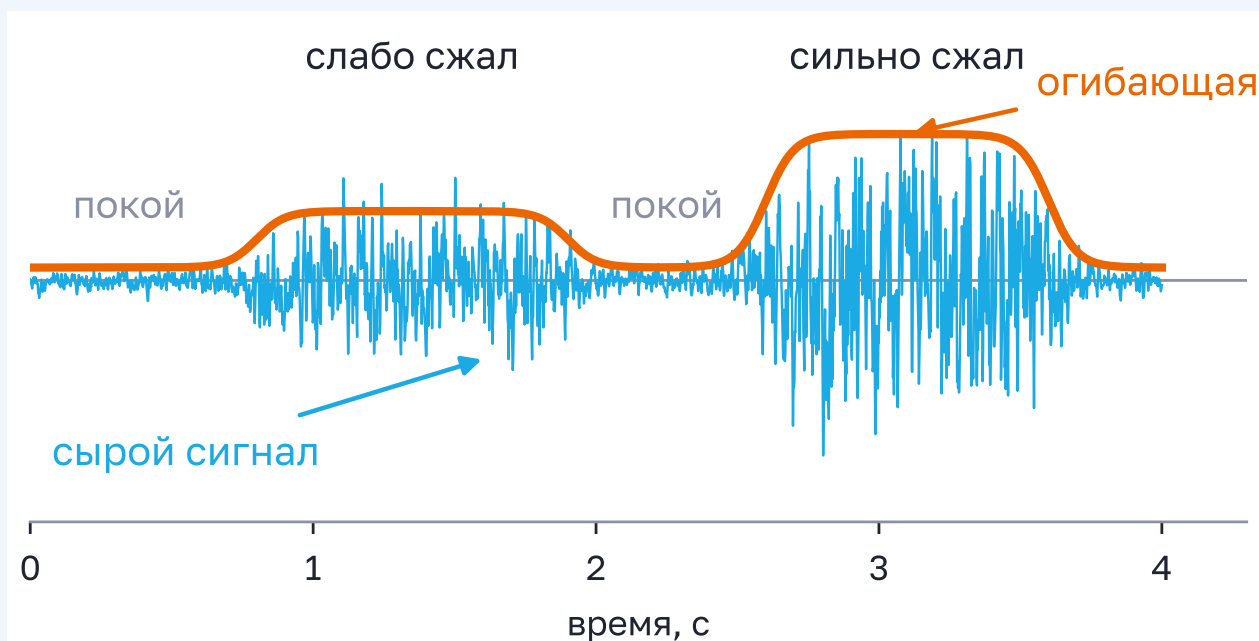


Рисунок 3.2 – Голубое – сырой сигнал ЭМГ, оранжевое – огибающая. По огибающей сразу видно, где рука отдыхала, а где её сжимали слабо и сильно

Пропусти полосу через сито: зубчики пропали, а главное осталось – где полоса широкая, где узкая. Линия идёт вдоль верхнего края полосы, но чуть ниже самых верхушек.

С «сырым» сигналом ЭМГ неудобно работать: он скачет и вверх, и вниз, и в любой момент может оказаться около нуля, даже если кулак сжат изо всех сил. Поэтому из него делают **огибающую** – плавную линию, которая показывает, насколько широка полоса в каждый момент. Огибающая поднимается, когда мышца напрягается, и опускается, когда расслабляется.

Получают огибающую в два шага.

Шаг 1. «Выпрямить» сигнал. Для каждого значения смотрим, насколько оно отошло от середины, не важно вверх или вниз (берём модуль). Если середина 500, то и 540, и 460 отошли на 40.

Шаг 2. Сгладить отклонения окном длиной в один период быстрых колебаний (несущей). Тогда в окно попадает целая волна, и среднее меняется плавно. Его и называют огибающей.

Пример. Середина 500, пять значений подряд, которые укладываются в один период: 540, 460, 580, 430, 550. Отклонения: 40, 40, 80, 70, 50. Среднее: $(40 + 40 + 80 + 70 + 50) : 5 = 280 : 5 = 56$. Огибающая в этот момент – 56, хотя сигнал доходил до 80. Так и должно быть: среднее всегда меньше самого большого значения, поэтому огибающая идёт ниже верхушек.

Некоторые модули ЭМГ считают огибающую сами и отдадут плате уже плавную линию, некоторые отдают сырой сигнал, и огибающую считает программа. В задачах почти всегда даётся огибающая – в **условных единицах**: это числа, которые вернула команда `analogRead`, от 0 до 1023. Сравнивать их можно только между собой, в одной и той же записи.

Запомни. ЭМГ ловит сразу сотни волокон, поэтому она «лохматая» и без чётких пиков. Сильнее напряг мышцу – шире полоса. Огибающая рисует ширину полосы одной плавной линией: по ней и работают все программы этой главы.

Совет. В симуляторе (Wokwi или Tinkercad) живой мышцы нет, и ЭМГ там изображают. Ручка потенциометра играет роль силы сжатия, а случайные числа добавляют «лохматость». Покрутил ручку – «мышца напряглась».

3.2. Порог: как сигнал становится командой

НА ПАЛЬЦАХ



Рисунок 3.3 – Огибающая ЭМГ и порог. Пока линия выше порога, лампочка горит

Порог – как планка у прыгуна в высоту. Перепрыгнул – есть команда, не дотронул – ничего не происходит.

Контроллер не понимает, что значит «рука напряжена». Он умеет сравнивать числа. Поэтому инженер выбирает число – **порог** – и пишет правило: огибающая выше порога – мышца напряжена, включаем светодиод или сжимаем пальцы модели; не выше – мышца отдыхает, выключаем. Это называют **детектированием по уровню** (амплитудным), или по порогу. В главе 2 тем же приёмом ловили удары сердца.

Вот такая программа для Arduino зажигает светодиод, когда мышца напряжена. Если ты ещё не писал программ, прочитай комментарии после `//` – в них сказано, что делает каждая строка.

```

const int EMG = A0;           // ЭМГ-модуль – на аналоговом входе A0
const int LED = 13;           // светодиод – на выводе 13
const int POROG = 300;        // порог: выше него мышца «напряжена»

void setup() {
  pinMode(LED, OUTPUT);       // вывод 13 работает как выход
  Serial.begin(9600);          // связь с компьютером – чтобы видеть график
}

void loop() {
  int emg = analogRead(EMG);   // огибающая ЭМГ: число от 0 до 1023
  Serial.println(emg);          // отправляем число на график
  if (emg > POROG) {            // сигнал выше порога?
    digitalWrite(LED, HIGH);    // да – зажечь светодиод
  } else {
    digitalWrite(LED, LOW);     // нет – погасить
  }
  delay(10);                   // подождать 10 мс и измерить снова
}

```

Часть `setup()` плата выполняет один раз при включении, а `loop()` – снова и снова, по кругу, сотню раз в секунду. Слово `const int` значит «целое число, которое не меняется»: так удобно дать порогу имя и потом поменять его в одном месте. Команда `if ... else` – развилка: если условие в скобках верно, выполняется первая часть, иначе вторая.

Как выбрать порог: калибровка

Какое число поставить в `POROG`? У Маши в покое огибающая гуляет около 50, у Пети – около 150: у каждого своя кожа, свои мышцы и электроды стоят чуть по-разному. Поэтому порог подбирают под человека. Эта подстройка называется **калибровкой**.

Делают её так. Человек несколько секунд сидит спокойно – записываем самое **большое** значение огибающей в покое. Потом несколько секунд сжимает кулак – записываем самое **маленькое** значение при сжатии. Порог должен лечь между ними: выше всего, что бывает в покое, и ниже всего, что бывает при сжатии. Надёжнее всего – посередине:

$$\text{порог} = \frac{\text{наибольшее в покое} + \text{наименьшее при сжатии}}{2}.$$

Пример: в покое огибающая не поднималась выше 110, при сжатии не опускалась ниже 430. Порог: $(110 + 430) : 2 = 540 : 2 = 270$. От него до покоя $270 - 110 = 160$ единиц запаса, до сжатия – тоже 160.

Что будет, если ошибиться с порогом:

- **порог слишком низкий** – в покое случайный всплеск шума перепрыгнет планку, и рука модели дёрнется сама собой;
- **порог слишком высокий** – чтобы сработало, придётся сжимать кулак изо всех сил, рука быстро устанет, а слабое сжатие не заметят вовсе.

Сколько раз работает

В задачах часто спрашивают: «сколько раз включится светодиод?» Включение – это **момент, когда сигнал перешёл через порог снизу вверх**. Пока огибающая остаётся наверху, свето-

диод просто горит, и новых включений нет. Одно долгое сжатие – одно включение, сколько бы точек ни оказалось над планкой.

И внимательно смотри на слово в условии. «Выше порога» или «больше порога» – значит, значение, **равное** порогу, ещё не считается. Если в условии «не меньше порога», равное уже считается.

Когда сигнал дрожит у порога: два порога

НА ПАЛЬЦАХ

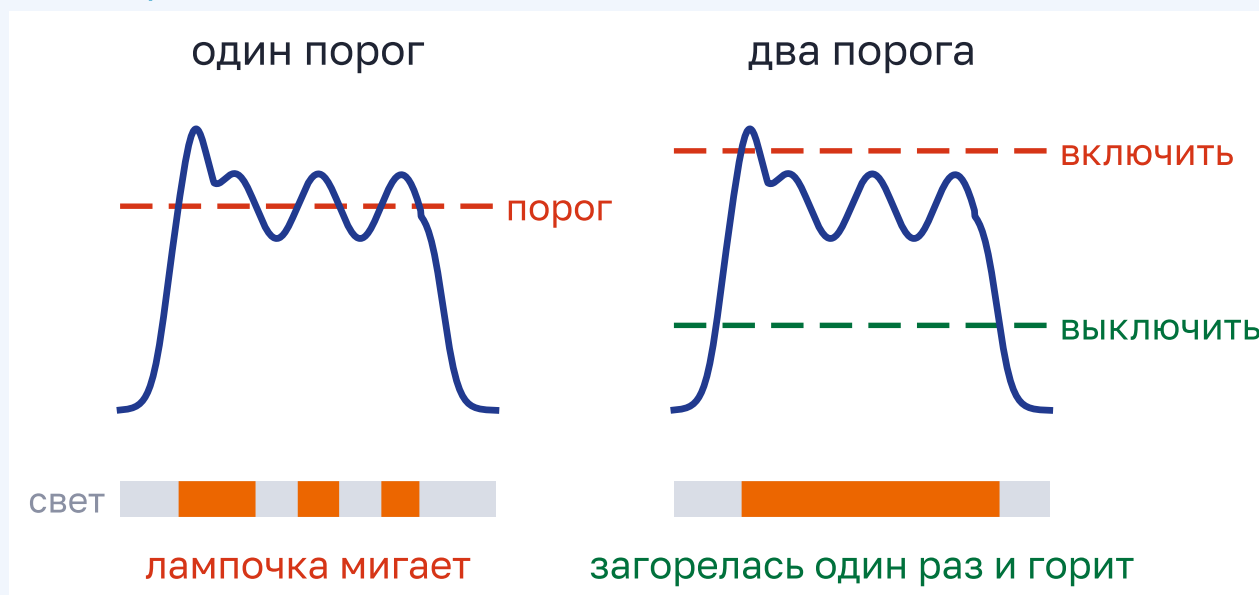


Рисунок 3.4 – Один и тот же сигнал дрожит у планки. С одним порогом лампочка мигает, с двумя – загорается один раз и горит до конца сжатия

Один порог – как автоматическая дверь в магазине, если встать на самом краю: качнулся вперёд – открылась, назад – закрылась, и так без конца. Два порога говорят: включай высоко, выключай низко, а посередине ничего не трогай.

Огибающая никогда не бывает идеально гладкой. Если человек держит кулак сжатым не сильно и огибающая колыхается у самого порога – 390, 410, 395, 420, – светодиод начинает мигать, а рука модели – дёргаться: сжалась, разжалась, снова сжалась.

Лечат это **двумя порогами**. Верхний – чтобы включить: сигнал поднялся выше него – сжимаем. Нижний – чтобы выключить: сигнал опустился ниже него – разжимаем. Пока сигнал между порогами, рука остаётся как была. Так устроен термостат в обогревателе: включается, когда в комнате остыло до одной температуры, а выключается, когда нагрелось до другой, повыше, – иначе он щёлкал бы каждую секунду. У инженеров для такого приёма есть длинное слово **гистерезис**, но запоминать его не обязательно.

Чтобы работать с двумя порогами, программе нужно помнить, в каком **состоянии** рука сейчас: сжата или разжата. Поэтому в задачах на два порога всегда проходи по сигналу по порядку и после каждого значения записывай состояние. Это маленькая машина состояний, о них подробно – в главе 6.

Запомни. Порог ставят между покоем и сжатием: выше самого большого значения в покое и ниже самого маленького при сжатии, лучше посередине. Срабатывания считают по переходам через порог снизу вверх. Сигнал дрожит у порога – бери два порога: включать по верхнему, выключать по нижнему.

Совет. Перед подсчётом срабатываний выпиши под каждым числом букву: В – выше порога, Н – не выше. Срабатывание – это место, где после Н стоит В. Посчитать такие места в строке «Н Н В В Н В» легче, чем в столбике трёхзначных чисел.

3.3. Бионическая рука и мио-армрестлинг

НА ПАЛЬЦАХ

сильнее сжал кулак – сильнее согнулся палец модели

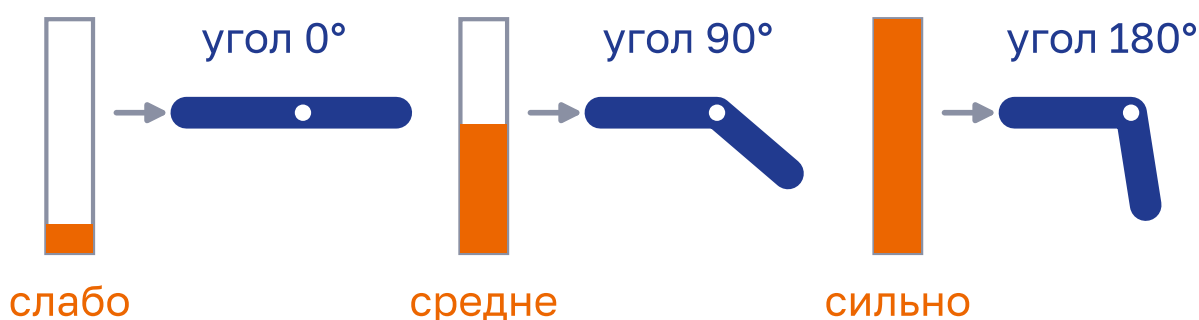


Рисунок 3.5 – Чем сильнее сжат кулак, тем больше угол сервопривода и тем сильнее согнут палец модели

Как регулятор громкости: чуть повернул – чуть громче, до упора – на полную.
Здесь «ручку» крутит мышца.

Бионическая рука – модель протеза: пальцы из пластика, а сгибают их **сервоприводы**. Сервопривод – моторчик, который поворачивает свой вал на заданный угол, от 0 до 180°, и держит его. Скажешь ему 90 – повернётся на 90° и остановится.

Управлять рукой можно двумя способами.

По порогу – «да или нет». Огибающая выше порога – пишем в сервопривод 180° (палец согнут), ниже – 0° (палец выпрямлен). Просто и надёжно, но рука умеет только хватать и отпускать.

Пропорционально – «насколько». Чем сильнее сжат кулак, тем сильнее согнут палец. Для этого огибающую нужно пересчитать в угол. После калибровки мы знаем два числа: огибающую в покое (пусть 80) и при сильном сжатии (пусть 680). Покою должен соответствовать угол 0°, сильному сжатию – 180°, а всему, что между, – углы между ними.

Считают так. Весь диапазон огибающей – от 80 до 680, это $680 - 80 = 600$ единиц. На эти 600 единиц приходится 180° . Огибающая 380 отошла от покоя на $380 - 80 = 300$ единиц – это половина диапазона, значит, и угол – половина от 180° :

$$\text{угол} = \frac{(\text{огибающая} - \text{покой}) \cdot 180}{\text{сжатие} - \text{покой}} = \frac{(380 - 80) \cdot 180}{680 - 80} = \frac{300 \cdot 180}{600} = 90^\circ.$$

В Arduino этот пересчёт делает одна команда `map`:

```
#include <Servo.h>           // библиотека для сервопривода
Servo finger;                // сервопривод пальца

const int POKOY = 80;        // огибающая в покое (узнали при калибровке)
const int SZHAT = 680;       // огибающая при сильном сжатии

void setup() {
  finger.attach(9);          // сигнальный провод сервопривода – на вывод 9
}

void loop() {
  int emg = analogRead(A0);   // огибающая ЭМГ
  int angle = map(emg, POKOY, SZHAT, 0, 180); // пересчёт в угол
  angle = constrain(angle, 0, 180); // не выйти за 0...180°
  finger.write(angle);        // повернуть палец
  delay(15);
}
```

Здесь две ловушки, и обе любят составители задач.

- `map` **отбрасывает дробную часть**, а не округляет. Вышло 65,1 – будет 65. Вышло 94,9 – будет 94, а не 95.
- `map` **не знает границ**. Если человек сжал кулак сильнее, чем при калибровке, и огибающая стала 720, формула выдаст $(720 - 80) \cdot 180 : 600 = 192^\circ$. А сервопривод больше 180° не умеет. Если огибающая ниже покоя, выйдет отрицательный угол. Поэтому следом стоит `constrain(angle, 0, 180)` – «зажим»: всё, что меньше 0, становится 0, всё, что больше 180, становится 180.

Мио-армрестлинг

НА ПАЛЬЦАХ

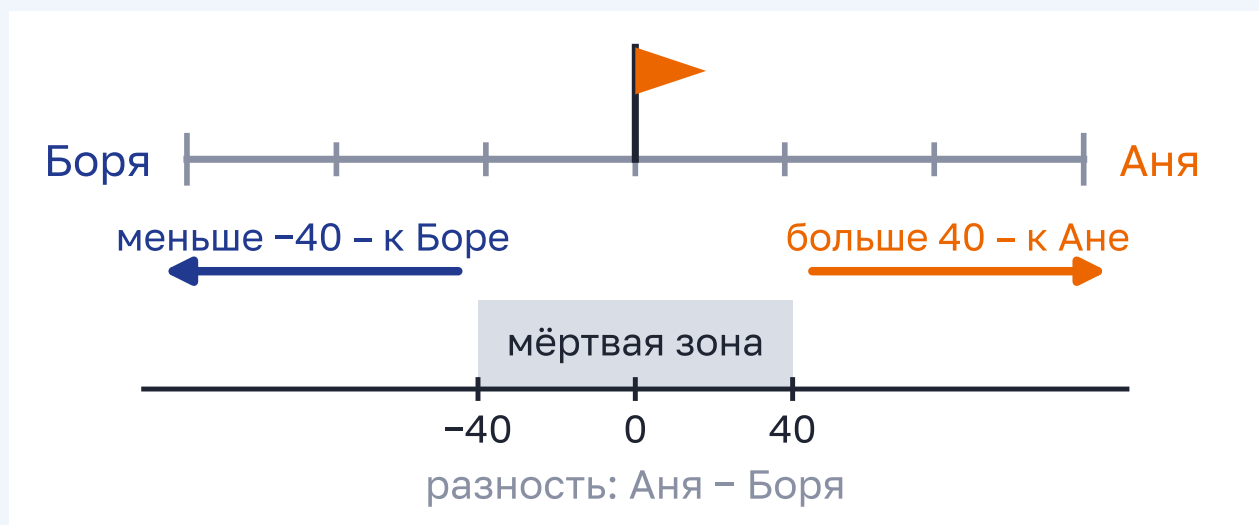


Рисунок 3.6 – Флажок стоит в центре дорожки. Он двигается, только когда разность сигналов вышла из серой «мёртвой зоны»

Перетягивание каната, где руки лежат на столе, а тянут сигналы мышц. Если оба тянут почти одинаково, канат стоит на месте.

Мио-армрестлинг – соревнование, в котором руки не касаются друг друга. Двое садятся друг напротив друга, у каждого на предплечье ЭМГ-датчик. Arduino сравнивает две огибающие, и мотор тянет флажок (или ползунок, или стрелку) в сторону того, чья мышца напряжена сильнее. Кто дотянул флажок до своей отметки – победил.

Сравнивают сигналы через **разность**: из огибающей первого игрока вычитают огибающую второго. Разность больше нуля – сильнее первый, меньше нуля – второй. Но две живые руки никогда не дают одинаковых чисел, даже если оба игрока расслаблены: у одного 55, у другого 70. Чтобы флажок не ползал от такой мелочи, вводят **мёртвую зону** – полосу вокруг нуля, в которой ничего не происходит. Например: флажок двигается, только если разность больше 40 или меньше -40.

Запомни. Пропорциональное управление:

$$\text{угол} = (\text{огibaющая} - \text{покой}) \cdot 180 : (\text{сжатие} - \text{покой})$$
. Arduino-команда `map` отбрасывает дробную часть, а `constrain` не пускает угол за 0 и 180°. В мио-армрестлинге смотрят на разность двух сигналов, а маленькую разность гасит мёртвая зона.

3.4. ЭЭГ: ритмы мозга

НА ПАЛЬЦАХ



Рисунок 3.7 – Слева – спокойный человек с закрытыми глазами, справа – человек с открытыми глазами занят делом

Миллиарды нейронов как стадион болельщиков. Когда все хлопают в такт, хлопki складываются в мощные медленные волны. Когда каждый занят своим, общий шум мелкий и частый.

ЭЭГ, электроэнцефалограмма, – запись электрической активности мозга с поверхности головы. Нейроны коры работают большими группами, и когда тысячи клеток включаются и выключаются в одном темпе, их сигналы складываются в **волны**. Волны доходят до кожи головы очень ослабленными – через кости черепа. На коже это всего десятки микровольт, в десятки раз слабее, чем ЭМГ (а 1 мкВ – это миллионная доля вольта). Поэтому в ЭЭГ-модуле стоит мощный усилитель, а электроды прижимают к голове ободком.

В 1929 году немецкий врач Ганс Бергер опубликовал первые записи ЭЭГ человека. Он заметил: когда человек спокойно сидит с закрытыми глазами, на записи появляются ровные волны около десяти в секунду. Он назвал их **альфа-волнами**. Стоило человеку открыть глаза или начать решать задачу в уме, альфа-волны пропадали, и оставались мелкие частые колебания – их Бергер назвал **бета-волнами**.

Ритмы ЭЭГ различают по **частоте** – сколько волн проходит за одну секунду. Частоту меряют в **герцах**: 1 Гц – одна волна в секунду.

Таблица 3.1

Ритм	Частота	Когда сильнее
дельта	примерно 0,5–4 Гц	в глубоком сне
тета	примерно 4–8 Гц	в дремоте, на грани сна
альфа	примерно 8–13 Гц	покой, глаза закрыты

Ритм	Частота	Когда сильнее
бета	примерно 13–30 Гц	глаза открыты, внимание, работа

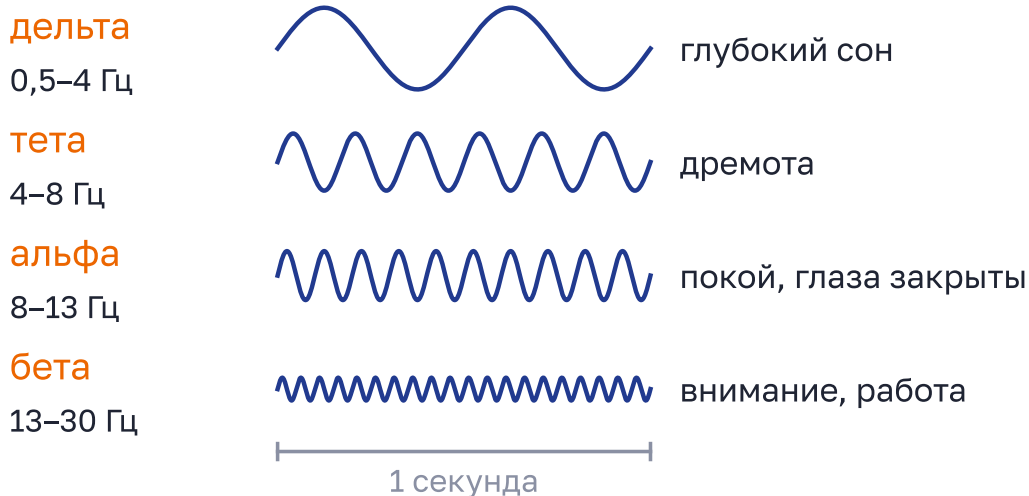


Рисунок 3.8 – Четыре ритма ЭЭГ на одной и той же секунде. Чем выше частота, тем теснее волны

Границы ритмов в разных книгах чуть-чуть отличаются: где-то альфа – 8–12 Гц, где-то 8–13 Гц, а тета – 4–7 или 4–8 Гц. В этой книге мы берём 8–13 и 13–30, а на отборе бери те границы, которые даны в условии. Частоту, которая лежит на самой границе, в задачах почти не дают.

Как узнать ритм по графику

Ритм определяют в два шага: сначала находят частоту, потом смотрят в таблицу.

Способ 1: посчитать волны. Считаем, сколько полных волн (например, сколько вершин) уместилось на отрезке времени, и делим на длину отрезка в секундах:

$$f = \frac{N}{t},$$

где f – частота в герцах, N – число волн, t – время в секундах. Насчитал 20 волн за 2 с: $f = 20 : 2 = 10$ Гц – альфа. Насчитал 6 волн за 0,25 с: $f = 6 : 0,25 = 24$ Гц – бета.

Способ 2: по периоду. Период T – время одной волны, от вершины до следующей вершины. Частота – единица, делённая на период:

$$f = \frac{1}{T}.$$

Период 0,1 с: $f = 1 : 0,1 = 10$ Гц. Если период дан в миллисекундах, сначала переведи в секунды: 50 мс = 0,05 с, $f = 1 : 0,05 = 20$ Гц.

Спокойствие и внимание

Вот как меняется ЭЭГ в разных состояниях – это и проверяют в задачах «что будет, если...»:

- **закрыв глаза и расслабился** – альфа-ритм усиливается, волны большие и ровные. Сильнее всего он над затылком, где находятся зрительные отделы коры;
- **открыл глаза** – альфа-ритм слабеет: в мозг хлынула картинка, и нейроны перестали «петь хором»;

- **решаешь задачу в уме, сосредоточен** – альфа тоже слабеет, а бета-ритм становится заметнее. Бета-волны мельче альфа-волн, так что общий размах записи при этом уменьшается;
- **засыпаешь** – появляются медленные тета-волны, а в глубоком сне – ещё более медленные дельта.

На ЭЭГ попадает и то, что к мозгу отношения не имеет. Моргнул – на записи большой медленный скачок от движения глаз. Стиснул зубы – запись забивает ЭМГ жевательных мышц: она во много раз сильнее ЭЭГ. Такие лишние сигналы называют **помехами** (или артефактами). Поэтому во время записи ЭЭГ сидят спокойно, не жуют и стараются не моргать слишком часто.

Запомни. Альфа – примерно 8–13 Гц, покой с закрытыми глазами. Бета – примерно 13–30 Гц, внимание и работа. Частота: $f = N:t$ (волны за время) или $f = 1:T$ (через период в секундах). Открыл глаза или задумался – альфа слабеет.

Совет. Если отрезок на графике короче секунды, не забудь поделить на его длину. Шесть волн на отрезке 0,5 с – это 12 Гц, а не 6. Хорошая самопроверка: частота альфа-ритма – около десяти, и если у спокойного человека с закрытыми глазами вышло 5 или 50, где-то ошибка.

3.5. Биообратная связь: человек смотрит на свой сигнал

НА ПАЛЬЦАХ

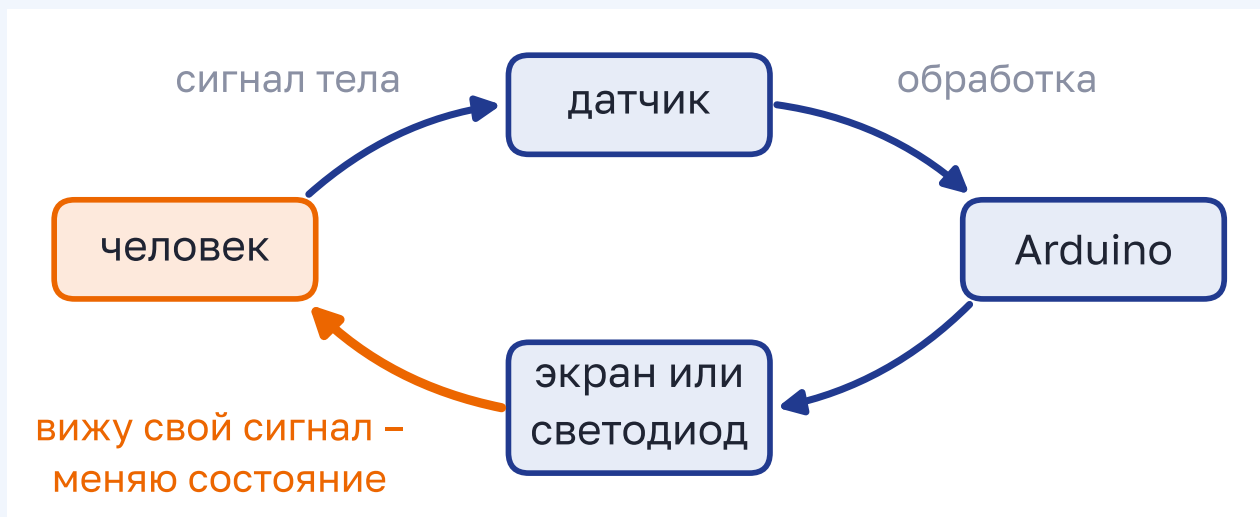


Рисунок 3.9 – Петля биологической обратной связи: сигнал тела – датчик – обработка – экран – глаза человека – и снова тело

Как зеркало в танцевальном зале: видишь себя – и сразу поправляешь движение. Только здесь в «зеркале» не руки и ноги, а пульс, мышцы или ритмы мозга.

Обычно мы не чувствуем, насколько напряжены мышцы плеч или сколько альфа-волн сейчас в мозге. Но если прибор покажет это на экране, человек начинает замечать связь: «ага, когда я опускаю плечи, полоска падает». Через какое-то время он учится управлять своим состоя-

нием по желанию. Это называется **биологической обратной связью**, сокращённо **БОС**, а занятия с ней – **БОС-тренингами**.

Любой БОС-тренажёр – петля из четырёх звеньев: датчик снимает сигнал, контроллер его обрабатывает (считает огибающую, частоту, сравнивает с порогом), экран или светодиоды показывают результат, а человек смотрит и меняет своё состояние. Замкнулась петля – человек видит, как изменился сигнал, и пробует снова.

Примеры тренажёров из этой главы:

- **«Подними ракету»** – ЭЭГ: пока альфа-ритм сильный, ракета на экране поднимается. Чтобы взлететь, надо успокоиться.
- **«Расслабь плечи»** – ЭМГ с мышцы плеча: чем выше огибающая, тем больше горит светодиодов на шкале. Задача – погасить их все.
- **«Держи коридор»** – ЭМГ: зелёный свет горит, пока огибающая в заданном коридоре, не ниже одного порога и не выше другого. Учит точно дозировать усилие – пригодится тому, кто будет управлять протезом.

Правила игры в БОС-тренажёре – это почти всегда пороги и счётчики из этой главы: «каждую секунду, когда сигнал выше порога, ракета поднимается на столько-то». И пороги здесь тоже калибруют под человека: слишком лёгкая игра ничему не учит, а слишком трудная надоест через минуту. Кстати, задача финала этого сезона, как рассказывали разработчики сферы, – как раз БОС-тренажёр для космонавтов: следить за своим состоянием в стрессе на орбите нужно не меньше, чем на Земле.

Запомни. БОС – человек видит свой биосигнал и учится им управлять. Петля: датчик → обработка → экран → человек → датчик. В задачах про тренажёр иди по секундам по порядку и сверяйся с условием: «меньше» там или «не меньше», может ли ракета уйти ниже земли.

3.6. Разбираем задачи

Задача 3.1 Порог для бионической руки

разминка · ЭМГ, выбор порога по графику, калибровка

Лена калибрует бионическую руку. На рисунке – огибающая ЭМГ с её предплечья: сначала 3 секунды рука лежит спокойно, потом 3 секунды Лена сжимает кулак, и так два раза. Рука модели сжимается, когда огибающая **выше** порога. Какие пороги подойдут – рука будет сжата всё время, пока Лена сжимает кулак, и ни разу не сожмётся в покое? **1** – 80; **2** – 200; **3** – 300; **4** – 450; **5** – 600. Ответ – номера подходящих вариантов без пробелов, по возрастанию.

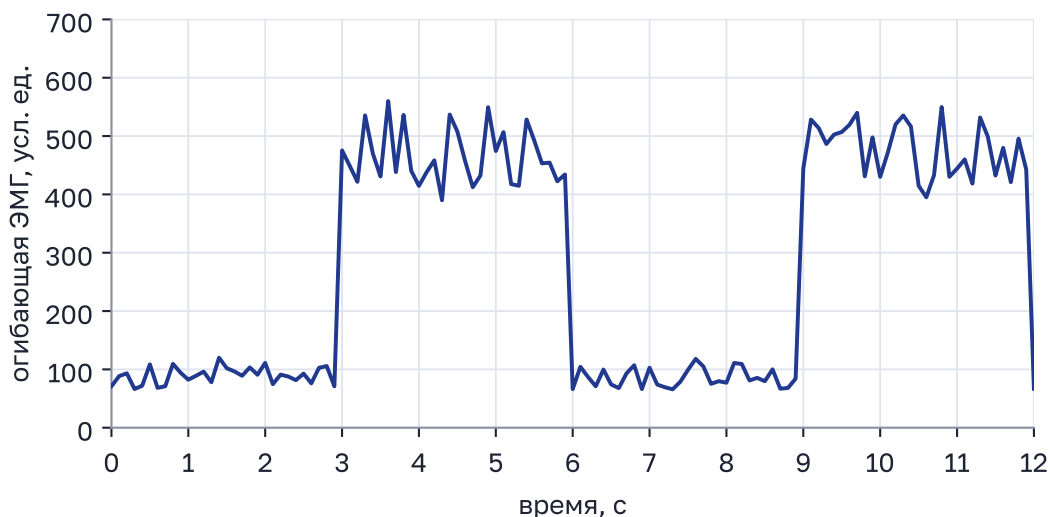


Рисунок 3.10 – Огибающая ЭМГ во время калибровки. По вертикали – условные единицы

Дано: график огибающей; рука сжата, если огибающая выше порога; пороги 80, 200, 300, 450, 600.

Найти: номера подходящих порогов.

Решение

Шаг 1. Что нужно от порога. Два условия сразу. В покое огибающая не должна подниматься выше порога ни на миг – значит, порог выше самого большого значения в покое. При сжатии огибающая не должна опускаться до порога – значит, порог ниже самого маленького значения при сжатии.

Шаг 2. Покой. Участки 0–3 с и 6–9 с. Самый высокий зубец – около 1,4 с, чуть выше линии 100: около 120. Всё остальное в покое ещё ниже.

Шаг 3. Сжатие. Участки 3–6 с и 9–12 с. Самый глубокий провал – около 4,3 с: чуть ниже линии 400, около 390.

Шаг 4. Коридор для порога. Порог должен лечь между 120 и 390.

Шаг 5. Проверяем варианты.

- 80 – ниже 120: зубцы покоя его перепрыгнут, рука сожмётся сама. Не подходит.
- 200 – между 120 и 390. Подходит.
- 300 – между 120 и 390. Подходит.
- 450 – выше 390: в провалах при сжатии огибающая опустится ниже 450, и рука на мгновение разожмётся. Не подходит.
- 600 – выше всего, что было на графике: рука не сожмётся ни разу. Не подходит.

Середина коридора – $(120 + 390) : 2 = 255$. Порог 200 чуть ближе к покою, 300 – к сжатию, но запас у обоих не меньше 80 единиц в каждую сторону.

⚠ Типичная ошибка. Смотреть на средний уровень вместо крайних точек. На глаз сжатие держится около 450–500, и кажется, что порог 450 годится. Но порогу важен самый глубокий провал: в момент 4,3 с огибающая опускается до 390, и рука с порогом 450 разожмётся посреди сжатия.

Ответ: 23.

Своя задача по программе сферы, модуль 7

Реши сам

3.1. При калибровке огибающая ЭМГ Тимура в покое менялась от 30 до 90, а при сжатии кулака – от 310 до 480. Найди порог по правилу середины: посередине между самым большим значением в покое и самым маленьким при сжатии.

3.2. Руку Тимура из предыдущей задачи откалибровали и поставили порог 200. На следующий день электрод немного сполз, и сигнал ослаб: в покое огибающая теперь от 20 до 60, при сжатии – от 150 до 260. а) Будет ли рука со старым порогом 200 сжата всё время, пока Тимур сжимает кулак? Ответь «да» или «нет». б) Найди новый порог по правилу середины.

Задача 3.2 Глаза открыты или закрыты

разминка · ЭЭГ, частота по числу волн, альфа и бета

Саша записал свою ЭЭГ два раза: один раз спокойно сидел с закрытыми глазами, другой – с открытыми глазами читал задачу с экрана. На рисунке – по полсекунды каждой записи. а) Найди частоту волн на записях А и Б в герцах. б) Какая запись сделана с закрытыми глазами? Ритмы: альфа – 8–13 Гц, бета – 13–30 Гц.

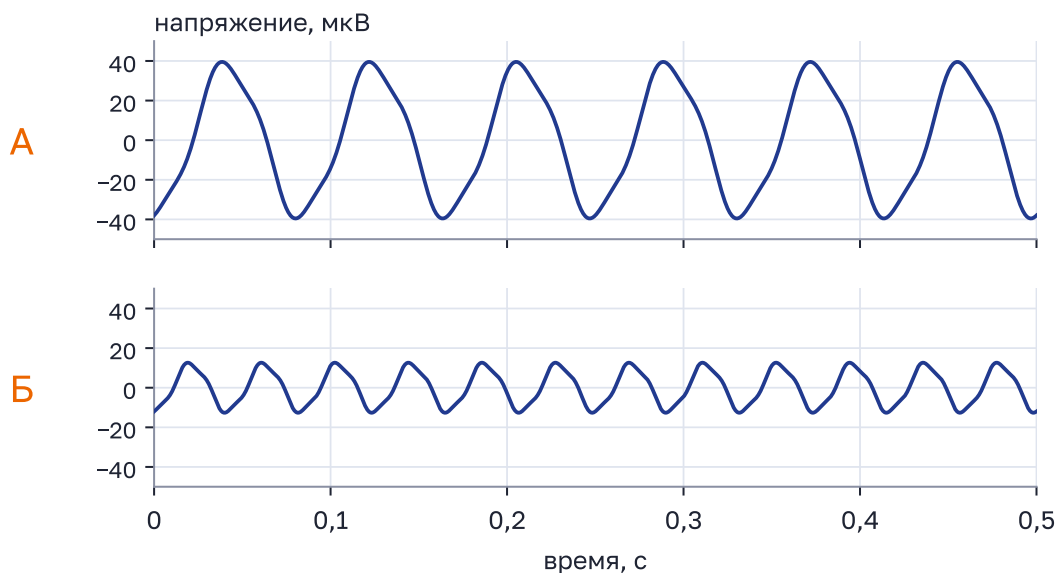


Рисунок 3.11 – Две записи ЭЭГ по 0,5 с

Дано: две записи по 0,5 с; альфа 8–13 Гц, бета 13–30 Гц.

Найти: f_A , f_B – ?; запись с закрытыми глазами – ?

Решение

Шаг 1. Считаем волны на записи А. Удобнее всего считать вершины. На записи А вершин 6, и все они целиком внутри отрезка.

Шаг 2. Частота записи А. 6 волн за 0,5 с: $f_A = 6 : 0,5 = 12$ Гц. Проверим через период: волна повторяется примерно через 0,083 с (чуть меньше одной клетки 0,1 с), $1 : 0,083 \approx 12$ Гц.

Шаг 3. Считаем волны на записи Б. Вершин **12** – вдвое больше, волны вдвое теснее.
 $f_B = 12 : 0,5 = 24$ Гц.

Шаг 4. Какие это ритмы. 12 Гц лежит в полосе 8–13 Гц – это альфа. 24 Гц – в полосе 13–30 Гц, бета.

Шаг 5. Какая запись с закрытыми глазами. Альфа-ритм сильнее всего у спокойного человека с закрытыми глазами. На записи А он есть, и волны большие – размах около 80 мкВ, от –40 до +40. На записи Б волны мелкие и частые – так выглядит ЭЭГ, когда глаза открыты и человек читает.

 **Типичная ошибка.** Забыть, что на рисунке всего полсекунды, и записать «6 Гц» и «12 Гц». Тогда запись А попадёт в тета-ритм, а Б – в альфа, и ответ на пункт б) тоже перевернётся. Прежде чем считать волны, посмотри, какой отрезок времени показан.

Ответ: а) А – 12 Гц, Б – 24 Гц; б) запись А.

По мотивам задачи НТО 2025/26, профиль «Нейротехнологии и когнитивные науки», заключительный этап, инженерный тур «Стандартная функциональная проба»

Реши сам

3.3. На записи ЭЭГ от одной вершины волны до следующей проходит 0,05 с. Какой это ритм? **1** – дельта (0,5–4 Гц); **2** – тета (4–8 Гц); **3** – альфа (8–13 Гц); **4** – бета (13–30 Гц). Ответ – номер варианта.

3.4. ЭЭГ-модуль передаёт в компьютер 250 чисел (отсчётов) в секунду. На записи от одной вершины волны до следующей – 25 отсчётов. Найди частоту волн в герцах и определи ритм по вариантам из предыдущей задачи.

Задача 3.3 Сколько раз загорится светодиод

детектирование по порогу, подсчёт срабатываний, время работы

В симуляторе Wokwi собрали модель ЭМГ-браслета: светодиод горит, пока огибающая **больше 300**. Программа измеряет огибающую каждые 0,1 с. Вот 20 значений подряд:

120, 150, 340, 410, 380, 290, 310, 180, 140, 300, 260, 520, 560, 530, 120, 90, 330, 110, 305, 100.

а) Сколько раз светодиод **включится** (перейдёт из «не горит» в «горит»)? б) Сколько секунд он горел в сумме? Считай, что после каждого измерения светодиод горит или не горит до следующего измерения.

Дано: 20 значений через 0,1 с; горит, если значение больше 300.

Найти: число включений; время горения – ?

Решение

Шаг 1. Помечаем каждое значение. Выше 300 – В, не выше – Н. Значение, равное 300, – не выше, это Н.

Таблица 3.2

Значения 1–5	Значения 6–10	Значения 11–15	Значения 16–20
120 Н	290 Н	260 Н	90 Н
150 Н	310 В	520 В	330 В
340 В	180 Н	560 В	110 Н
410 В	140 Н	530 В	305 В
380 В	300 Н	120 Н	100 Н

Получилась строчка: Н Н В В В Н В Н Н Н В В В Н Н В Н В Н.

Шаг 2. Считаем включения. Включение – место, где после Н стоит В. Идём по строчке: 150 → 340 (первое), 290 → 310 (второе), 260 → 520 (третье), 90 → 330 (четвёртое), 110 → 305 (пятое). Всего **5**. Значение 300 на десятом месте включения не дало: оно не больше порога.

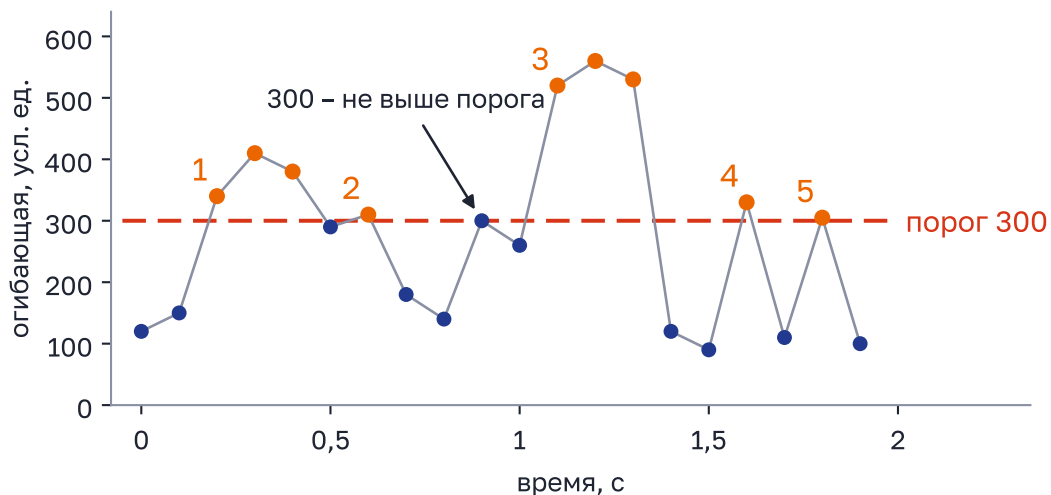


Рисунок 3.12 – Решение задачи 3.3: оранжевые точки – выше порога, цифры – номера включений

Шаг 3. Сколько значений с буквой В. 340, 410, 380, 310, 520, 560, 530, 330, 305 – девять штук.

Шаг 4. Время горения. Каждое значение «отвечает» за 0,1 с. $9 \cdot 0,1 = 0,9$ с.

Проверить можно на Python – программа проходит по списку и ищет те же места «было не выше – стало выше»:

```
vals = [120, 150, 340, 410, 380, 290, 310, 180, 140, 300,
        260, 520, 560, 530, 120, 90, 330, 110, 305, 100]
porog = 300
vkl = 0 # сколько раз включился
for i in range(1, len(vals)):
    if vals[i] > porog and vals[i - 1] ≤ porog: # был не выше, стал выше
        vkl += 1
gorel = sum(1 for v in vals if v > porog) * 0.1 # секунд горел
print(vkl, round(gorel, 1)) # 5 0.9
```

Пять включений за две секунды – для браслета многовато. Если человек сжимал кулак один-два раза, то включения 2, 4 и 5 – короткие вспышки, когда огибающая еле-еле переваливала через 300. Порог стоит слишком близко к сигналу.

❗ Типичная ошибка. Первая – ответить «9»: посчитать все значения выше порога. Это число измерений, когда светодиод горел, а включений меньше. Вторая – засчитать 300 как «выше порога» и получить лишнее, шестое включение на переходе 140 → 300. Слово «больше» в условии значит, что равное не считается.

Ответ: а) 5; б) 0,9 с.

Своя задача по программе сферы, модуль 2

Реши сам

3.5. Светодиод горит, пока огибающая ЭМГ больше 250. Огибающую измеряют каждые 0,2 с: 80, 210, 270, 240, 95, 60, 250, 300, 310, 120, 70, 265, 90, 250, 251, 40. а) Сколько раз светодиод включится? б) Сколько секунд он горел в сумме?

3.6. Ваня три раза сжал кулак, а между сжатиями пару раз случайно пошевелил пальцами. Огибающая ЭМГ: 90, 160, 420, 450, 380, 130, 220, 180, 390, 410, 140, 210, 100, 470, 120, 80. а) Сколько раз сработает счётчик сжатий с порогом 200 и сколько – с порогом 350? Срабатывание – переход из «не больше порога» в «больше порога». б) Какой из двух порогов честно посчитал сжатия Вани?

Задача 3.4 На сколько градусов согнётся палец

пропорциональное управление, map, constrain, сервопривод

Палец бионической руки управляется по программе из раздела 3.3: при калибровке огибающая в покое была 80, при сильном сжатии – 680.

```
int angle = map(emg, 80, 680, 0, 180);
angle = constrain(angle, 0, 180);
finger.write(angle);
```

На какой угол повернётся сервопривод, если огибающая равна а) 297; б) 720? Помни, что map отбрасывает дробную часть. Ответ – в градусах.

Дано: покой 80, сжатие 680; углы от 0 до 180°; огибающая 297 и 720.

Найти: угол в каждом случае – ?

Решение

Шаг 1. Ширина диапазона. От покоя до сжатия $680 - 80 = 600$ единиц. На них приходится 180°.

Шаг 2. Пункт а): насколько огибающая отошла от покоя. $297 - 80 = 217$ единиц.

Шаг 3. Угол. $217 \cdot 180 : 600 = 39\,060 : 600 = 65,1$. Команда `map` отбрасывает дробную часть: 65. Команда `constrain` ничего не меняет – 65 лежит между 0 и 180. Угол **65°**.

Шаг 4. Пункт б). $720 - 80 = 640$, $640 \cdot 180 : 600 = 115\,200 : 600 = 192$. Это больше 180 – человек сжал кулак сильнее, чем при калибровке. `constrain` «зажимает» 192 до 180. Угол **180°**.

Прикинем на глаз: 297 – примерно треть пути от 80 до 680, и 65° – примерно треть от 180°. А 720 за пределами калибровки, так что палец и должен согнуться до упора.

❗ **Типичная ошибка.** Забыть вычесть покой и поделить 297 на 680: $297 \cdot 180 : 680 \approx 78,6$ – получится 78°. Формула `map` считает от начала диапазона, то есть от 80, а не от нуля. Вторая ловушка – ответить в пункте б) «192»: сервопривод на такой угол не повернётся – для того в программе и стоит `constrain`.

Ответ: а) 65°; б) 180°.

Своя задача по программе сферы, модуль 7

Реши сам

3.7. У Вероники при калибровке огибающая в покое 100, при сильном сжатии – 900. Программа: `angle = map(emg, 100, 900, 0, 180); angle = constrain(angle, 0, 180);`. На какой угол повернётся сервопривод при огибающей а) 460; б) 60?

3.8. Программа – как в разобранный задаче 3.4 (покой 80, сжатие 680). Какое **наименьшее** целое значение огибающей даст угол 90°? Помни, что `map` отбрасывает дробную часть.

Задача 3.5 Мио-армрестлинг

мио-армрестлинг, разность сигналов, мёртвая зона

Аня и Боря играют в мио-армрестлинг. Каждые 0,5 с Arduino измеряет обе огибающие ЭМГ и вычитает из Аниной Борину. Если разность **больше 40** – флажок сдвигается на одно деление к Ане, если **меньше -40** – на одно деление к Боре, иначе стоит на месте. Флажок стартует в центре. Кто первым дотянет флажок на **3 деления** к себе, тот победил, и игра заканчивается. Первый замер – через 0,5 с после старта. Кто победил и через сколько секунд после старта?

Таблица 3.3

Замер	Аня	Боря
1	310	350
2	420	360
3	380	400
4	450	380
5	300	390

Замер	Аня	Боря
6	470	410
7	520	450
8	490	430
9	500	420
10	480	400

Дано: 10 замеров через 0,5 с; сдвиг при разности больше 40 или меньше -40; победа – 3 деления.

Найти: победитель и время победы – ?

Решение

Шаг 1. Договоримся о знаках. Положение флажка – число: 0 в центре, плюс – к Ане, минус – к Боре. Победа Ани – +3, Бори – -3.

Шаг 2. Идём по замерам и после каждого записываем разность и положение.

Таблица 3.4

Замер	Аня – Боря	Сдвиг	Положение
1	-40	0	0
2	60	+1	+1
3	-20	0	+1
4	70	+1	+2
5	-90	-1	+1
6	60	+1	+2
7	70	+1	+3

В первом замере разность -40 – это не меньше -40, поэтому флажок стоит. В третьем -20 попадает в мёртвую зону.

Шаг 3. Конец игры. После 7-го замера флажок на +3 – Аня победила, дальше не считаем.

Шаг 4. Время. Замеры идут через 0,5 с, седьмой – через $7 \cdot 0,5 = 3,5$ с после старта.

 **Типичная ошибка.** В первом замере разность равна -40, и рука сама тянется сдвинуть флажок к Боре. Но по правилу флажок двигается, только если разность **меньше** -40. С лишним шагом к Боре Аня дошла бы до +3 только на девятом замере. Вторая ошибка – до-считать все 10 замеров и ответить про положение в конце: игра закончилась раньше.

Ответ: Аня, через 3,5 с.

Своя задача по программе сферы, модуль 7

Реши сам

3.9. Катя и Дима играют в мио-армрестлинг по правилам разобранной задачи 3.5, но мёртвая зона другая: флажок сдвигается, если разность Катя – Дима больше 30 (к Кате) или меньше –30 (к Диме). Победа – 3 деления. Огибающие Кати по замерам: 200, 260, 300, 280, 350, 330, 310, 360; Димы: 240, 250, 220, 330, 280, 260, 300, 250. Где окажется флажок после 8 замеров? Ответ – число делений: со знаком «плюс», если к Кате, со знаком «минус», если к Диме.

3.10. Правила – как в разобранной задаче 3.5. В очередном замере огибающая Ани равна 450. Какое **наименьшее** значение должна показать огибающая Бори, чтобы флажок в этом замере не сдвинулся к Ане?

Задача 3.6 Ракета на альфа-ритме

БОС-тренинг, альфа-ритм, подсчёт по секундам

БОС-тренажёр «Подними ракету» раз в секунду измеряет силу альфа-ритма Лены (в условных единицах). Если сила альфа-ритма **не меньше 50**, ракета поднимается на 4 м, если меньше 50 – опускается на 1 м (ниже земли она не опускается). Ракета стартует с земли. Первые 4 секунды Лена сидит с закрытыми глазами и старается расслабиться, с 5-й по 7-ю секунду тренер просит её умножить в уме 17 на 23, потом она снова расслабляется. Сила альфа-ритма по секундам:

62, 58, 71, 66, 35, 28, 41, 55, 49, 68.

На какой высоте окажется ракета через 10 секунд?

Дано: 10 значений; не меньше 50 – вверх на 4 м, меньше 50 – вниз на 1 м; старт с земли.

Найти: высота через 10 с – ?

Решение

Шаг 1. Помечаем секунды. Не меньше 50 – «вверх», меньше 50 – «вниз»: 62 вверх, 58 вверх, 71 вверх, 66 вверх, 35 вниз, 28 вниз, 41 вниз, 55 вверх, 49 вниз, 68 вверх.

Шаг 2. Следим за высотой по секундам. 4, 8, 12, 16 м – четыре подъёма подряд. Потом 15, 14, 13 м – счёт в уме погасил альфа-ритм, три секунды вниз. Дальше 17 м (55 – вверх), 16 м (49 – вниз), 20 м (68 – вверх).

Шаг 3. Проверка счётом. Подъёмов 6, спусков 4: $6 \cdot 4 - 4 \cdot 1 = 24 - 4 = 20$ м. Такой подсчёт годится, только если ракета ни разу не коснулась земли, – а она не коснулась: самая низкая точка после старта 13 м.

⚠ Типичная ошибка. Посчитать 49 как «вверх», потому что «почти 50». Правило – «не меньше 50», а 49 меньше. Тогда выйдет $7 \cdot 4 - 3 \cdot 1 = 25$ м. Вторая – считать «сколько секунд вверх» и умножать только на 4, забыв про спуски: получится 24 м.

Ответ: 20 м.

По мотивам задачи НТО 2025/26, профиль «Нейротехнологии и когнитивные науки», заключительный этап, инженерный тур «Усложнённая функциональная проба»

Реши сам

3.11. В другом тренажёре ракета поднимается на 3 м в каждую секунду, когда сила альфа-ритма не меньше 45, и опускается на 2 м, когда меньше 45. Ниже земли ракета не опускается, стартует с земли. Сила альфа-ритма по секундам: 30, 38, 50, 52, 47, 44, 60, 61. На какой высоте окажется ракета через 8 секунд?

3.12. Тренажёр как в разобранной задаче 3.6, но за «хорошую» секунду ракета поднимается на 5 м, а за «плохую» опускается на 2 м. Игра длится 20 секунд. Ракета закончила игру на высоте 58 м и ни разу не касалась земли после старта. Сколько секунд из двадцати альфа-ритм был не меньше порога?

Задача 3.7* Два порога вместо одного

два порога, состояние сжата/разжата, дребезг

Паша дважды сжал кулак, но держал его несильно, и огибающая ЭМГ дрожала. Значения через каждые 0,1 с:

100, 180, 420, 380, 430, 390, 470, 410, 360, 300, 220, 150, 260, 410, 380, 480, 410, 380, 190, 120.

Сколько раз сожмётся рука модели а) с одним порогом: сжата, если огибающая больше 400, разжата, если не больше; б) с двумя порогами: разжатая рука сжимается, когда огибающая становится **больше 450**, сжатая разжимается, когда огибающая становится **меньше 250**, а в остальных случаях остаётся как была? В начале рука разжата. Ответ – два числа.

Дано: 20 значений; а) один порог 400; б) сжать при больше 450, разжать при меньше 250; в начале разжата.

Найти: число сжатий в а) и б) – ?

Решение

Шаг 1. Один порог. Сжатие – переход из «не больше 400» в «больше 400». Помечаем: 100 Н, 180 Н, 420 В, 380 Н, 430 В, 390 Н, 470 В, 410 В, 360 Н, 300 Н, 220 Н, 150 Н, 260 Н, 410 В, 380 Н, 480 В, 410 В, 380 Н, 190 Н, 120 Н. Места «Н → В»: 180 → 420, 380 → 430, 390 → 470, 260 → 410, 380 → 480. Рука сожмётся **5** раз, хотя Паша сжимал кулак всего дважды: в первом сжатии она дёрнулась трижды, во втором – дважды.

Шаг 2. Два порога: что помнить. Теперь одного значения мало, чтобы понять, что делает рука. Нужно помнить её состояние. Правила такие:

- рука **разжата** – смотрим только на верхний порог: больше 450 – сжать, иначе остаётся разжатой;
- рука **сжата** – смотрим только на нижний порог: меньше 250 – разжать, иначе остаётся сжатой.

Шаг 3. Идём по значениям.

- 100, 180, 420, 380, 430, 390 – рука разжата, ни одно не больше 450. Остаётся разжатой.
- 470 – больше 450: **сжатие номер 1**.

- 410, 360, 300 – рука сжата, ни одно не меньше 250. Остаётся сжатой, хоть огибающая и упала ниже 400.
- 220 – меньше 250: разжать.
- 150, 260, 410, 380 – рука разжата, ни одно не больше 450.
- 480 – больше 450: **сжатие номер 2**.
- 410, 380 – сжата, не меньше 250.
- 190 – меньше 250: разжать. 120 – остаётся разжатой.

Рука сожмётся **2** раза – столько же, сколько сжимал Паша.

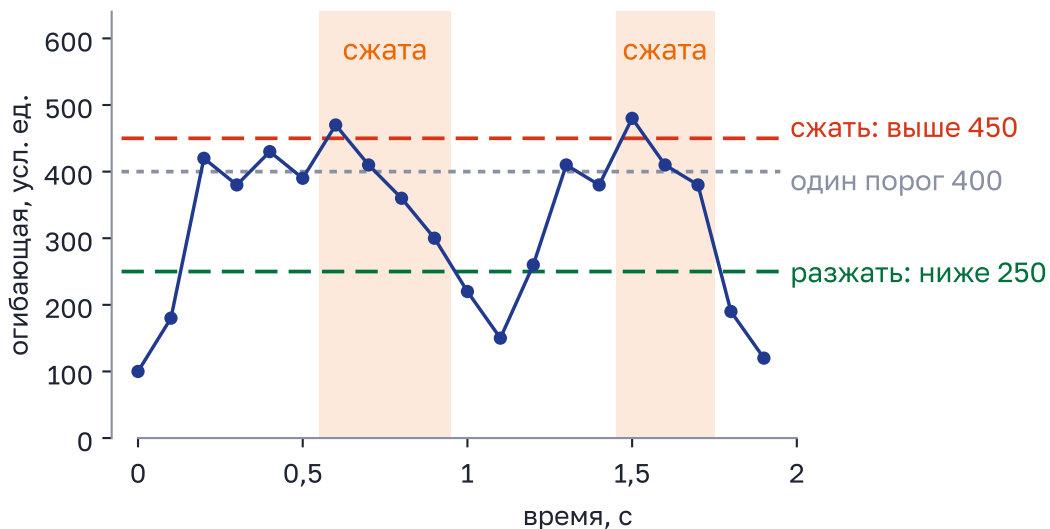


Рисунок 3.13 – Решение задачи 3.7: оранжевым закрашено время, когда рука сжата при двух порогах

Шаг 4. Как это выглядит в программе. Переменная `szhata` помнит состояние, и каждое новое значение проходит через такую функцию:

```
const int SZHAT_POROG = 450; // сжать, если огибающая выше
const int RAZZHAT_POROG = 250; // разжать, если ниже
bool szhata = false; // рука сейчас сжата?

void obrabotat(int emg) {
    if (!szhata && emg > SZHAT_POROG) { // была разжата и сигнал выше верхнего порога
        szhata = true; // сжимаем
    } else if (szhata && emg < RAZZHAT_POROG) { // была сжата и сигнал ниже нижнего
        szhata = false; // разжимаем
    }
}
```

Знак `!` значит «не»: `!szhata` – «рука не сжата». Знак `&&` – «и»: должны выполняться оба условия.

⚠ Типичная ошибка. В пункте б) смотреть на оба порога сразу, забыв про состояние. Например, решить, что при значении 410 после 470 рука разожмётся, «потому что 410 не больше 450». Но сжатая рука верхний порог уже не проверяет – она ждёт значения меньше 250. Чтобы не запутаться, выпиши рядом с каждым числом состояние руки после него: «разж», «сж».

Ответ: а) 5; б) 2.

Своя задача по программе сферы, модуль 7

Реши сам

3.13.* Огибающая ЭМГ через каждые 0,1 с: 90, 300, 520, 480, 510, 470, 530, 300, 190, 80, 350, 480, 500, 470, 260, 110. Сколько раз сожмётся рука а) с одним порогом: сжата, если огибающая больше 480; б) с двумя порогам: разжатая сжимается при огибающей больше 500, сжатая разжимается при огибающей меньше 200? В начале рука разжата.

3.14. Рука сжимается, когда огибающая становится больше 450. Известно: в покое огибающая держится от 120 до 180, а пока человек держит кулак сжатым, она дрожит от 400 до 520. Какой порог **разжатия** выбрать, чтобы на каждое сжатие кулака рука сжималась один раз и разжималась, когда человек расслабился? **1** – 500; **2** – 420; **3** – 300; **4** – 100. Ответ – номер варианта.

Сигналы и обработка данных



Датчик не знает, что такое пульс или сжатая мышца. Он выдаёт длинный ряд чисел, и дальше работает программа: ищет в этих числах удары, отсеивает помехи, считает частоту. В главах 2 и 3 мы разглядывали пульс, ЭМГ и ЭЭГ по отдельности. Здесь – приёмы, которым всё равно, откуда пришли числа: с пальца, с руки или с головы. На отборе такая задача выглядит просто – ряд чисел или график с датчика, а ответ одно число. Трудность в том, чтобы не сбиться при счёте.

4.1. Сигнал и шум

НА ПАЛЬЦАХ

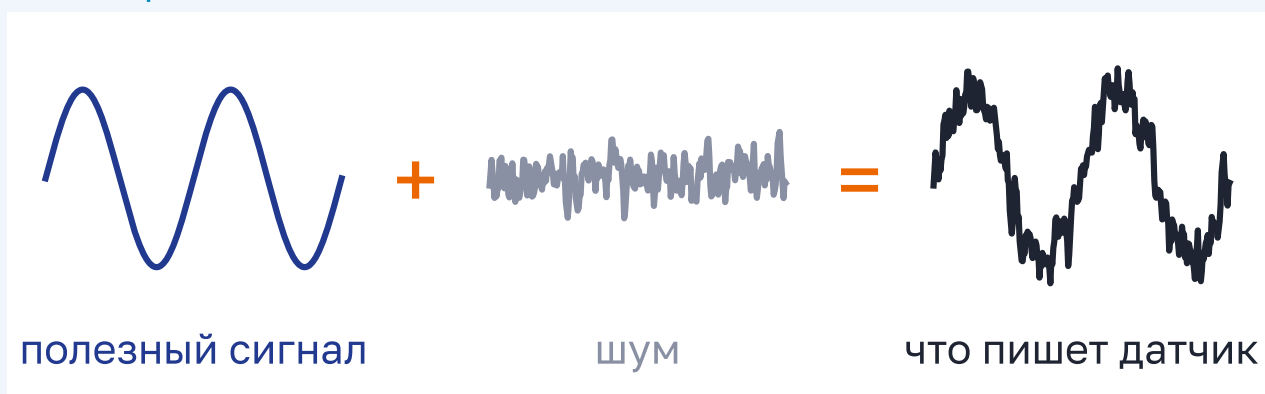


Рисунок 4.1 – Полезный сигнал плюс шум – так выглядит запись любого датчика

Ровная волна – то, что мы хотим услышать. Мелкая дрожь – помеха. Датчик ловит их вместе, и разделять их – забота программы.

Сигнал – это величина, которая меняется со временем и этим что-то сообщает. Звонок с урока, мигание поворотника у автобуса, стрелка весов, которая качнулась, когда ты на них встал. У датчика сигнал – напряжение на выходе: палец наполнился кровью – напряжение изменилось, мышца напряглась – тоже.

Сообщение несут **изменения**. Светофор, который всегда горит зелёным, водителю ничего не говорит. Датчик пульса, который минуту подряд показывает одно и то же число, тоже ничего не сообщает – скорее всего, он соскочил с пальца. Поэтому в сигнале нас интересует, как он меняется: насколько сильно, как часто, в какие моменты.

Шум – всё, что примешалось к полезному сигналу и мешает его читать. Источников у шума много:

- пошевелил пальцем – на пульсовой волне скачок выше любого удара;
- моргнул – на ЭЭГ всплеск от мышц век;
- провода работают как маленькие антенны и ловят «гул» от электросети. Ток в розетках переменный, он делает 50 колебаний в секунду, поэтому такая помеха – быстрые колебания частотой 50 Гц;
- электрод плохо прилегает к коже – сигнал то пропадает, то прыгает.

Шум есть всегда, вопрос в том, насколько он силён по сравнению с сигналом. Пульсовая волна с пальца крупная, и мелкая дрожь ей почти не мешает. ЭЭГ – самый слабый из биосигналов, десятки микровольт, и там шум бывает сильнее самого сигнала. Поэтому у каждого прибора два способа бороться: не пускать шум (хорошо закрепить датчик, не шевелиться) и убирать его программой. Второму посвящена почти вся эта глава.

Запомни. Сигнал – то, что меняется и этим несёт сообщение. Шум – лишнее, что к нему примешалось: движение, моргание, наводка от сети 50 Гц, плохой контакт. Датчик пишет сумму того и другого.

4.2. Что меряют у сигнала: размах, период, частота

НА ПАЛЬЦАХ

период 5 с – один вдох и выдох

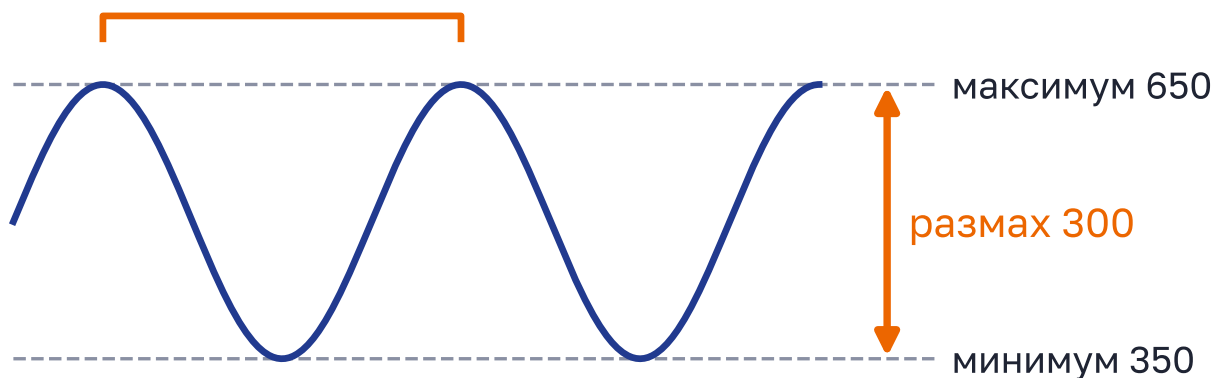


Рисунок 4.2 – Дыхательный пояс на груди: вдох – число растёт, выдох – падает

Как высоко качаются качели – это размах. Сколько секунд занимает один взмах туда-обратно – это период.

С любого сигнала, который качается вверх-вниз, первым делом снимают несколько чисел.

- **Максимум** – самое большое значение за время записи, **минимум** – самое маленькое.
- **Размах** – максимум минус минимум: на сколько сигнал поднимается от самой нижней точки до самой верхней.
- **Средний уровень** – середина между максимумом и минимумом: $(\max + \min) : 2$. Около него сигнал и качается.
- **Амплитуда** – на сколько сигнал отходит от среднего уровня вверх или вниз. У ровной волны это половина размаха (если сигнал качается около нуля, это то же самое, что расстояние от нуля до вершины):

$$A = \frac{\max - \min}{2}.$$

На рисунке пояс показывает от 350 до 650. Размах $650 - 350 = 300$, средний уровень $(650 + 350) : 2 = 500$, амплитуда $300 : 2 = 150$. Проверь себя: $500 + 150 = 650$ и $500 - 150 = 350$ – снова максимум и минимум.

Теперь время. **Период** T – сколько секунд длится одно полное колебание: от гребня до следующего гребня или от впадины до следующей впадины. **Частота** f – сколько колебаний укладывается в одну секунду. Её меряют в **герцах** (Гц): 1 Гц – одно колебание в секунду. Период и частота – две стороны одного и того же:

$$f = \frac{1}{T}, \quad T = \frac{1}{f}.$$

Если на записи насчитал N полных колебаний за t секунд, частота $f = N : t$. У дыхания на рисунке период 5 с, частота $1 : 5 = 0,2$ Гц. Для дыхания и пульса удобнее счёт «в минуту»: в минуте 60 с, значит, $60 : 5 = 12$ вдохов в минуту. Для пульса мы уже делали то же самое в главе 2: ЧСС = $60 : T$.

Частоты у биосигналов очень разные. Дыхание – доли герца, пульс – около одного-двух герц, альфа-ритм ЭЭГ – примерно 8–13 Гц, бета-ритм – примерно 13–30 Гц, быстрые колебания ЭМГ – десятки и сотни герц. Если у тебя «пульс» вышел 40 Гц, это 2400 ударов в минуту, и где-то в расчёте ошибка.

Запомни. Размах = максимум – минимум. Амплитуда – половина размаха, средний уровень – середина между максимумом и минимумом. Период – секунды на одно колебание, частота – колебания за секунду: $f = 1:T$.

Совет. Слова «размах» и «амплитуда» легко перепутать. Перечитай вопрос: если спрашивают амплитуду, а ты нашёл размах, осталось поделить на 2. Если в условии про ЭМГ сказано, как считать амплитуду, делай как сказано.

4.3. Аналоговый и цифровой сигнал. Дискретизация

НА ПАЛЬЦАХ



Рисунок 4.3 – Плавная линия – сигнал. Точки – то, что успела измерить плата

Кино – это 24 фотографии в секунду, а нам кажется, что всё движется. Плата тоже «фотографирует» сигнал: много раз в секунду записывает, чему он равен.

Аналоговый сигнал меняется плавно и может принять любое значение: 2,5 В, 2,51 В, 2,513 В... Так ведёт себя напряжение на выходе датчика, так движется стрелка у часов на стене. **Цифровой сигнал** состоит из набора строго определённых значений, обычно это 0 и 1. Так ведёт себя выключатель на стене: свет либо горит, либо нет, середины не бывает. И числа, которые плата записывает через промежутки времени, тоже идут ступеньками, как электронные часы: показывают 7:14, потом сразу 7:15, а всё, что между, пропускают. Есть и вторая разница, про помехи. Аналоговый сигнал портится понемногу: на старой записи слышно шипение, но слова разобрать можно. Цифровой сигнал либо доходит как надо, либо пропадает целиком: Wi-Fi передаёт картинку нулями и единицами, и она или чёткая, или с выпавшим куском.

Превращает одно в другое **АЦП** – аналого-цифровой преобразователь. У Arduino Uno он встроен в плату: команда `analogRead` измеряет напряжение от 0 до 5 В и выдаёт целое число

от 0 до 1023. Как устроены ступеньки и как перевести число обратно в вольты, разобрано в главе 5. Здесь нам важно другое: в какие моменты плата делает измерения.

Измерять можно только в отдельные моменты времени. Одно измерение называется **отсчётом**, а сам процесс – **дискретизацией** (от слова «дискретный» – раздельный, по кусочкам). Сколько отсчётов плата делает за одну секунду, называют **частотой дискретизации** f_d . Её тоже пишут в герцах, только здесь 1 Гц – один отсчёт в секунду.

Зная частоту дискретизации, легко найти ещё два числа.

- **Шаг между отсчётами:** $\Delta t = 1 : f_d$. При 100 Гц это $1 : 100 = 0,01$ с, или 10 мс.
- **Сколько отсчётов за время записи:** $n = f_d \cdot t$. За полминуты при 100 Гц – $100 \cdot 30 = 3000$ отсчётов.

Как часто нужно измерять? Посмотри на рисунок ниже. Серая волна делает 2 колебания в секунду. Слева её измеряют 20 раз в секунду, и точки повторяют волну. Справа – всего 3 раза в секунду. Точки получились, но соединить их можно только в медленную волну, которой в сигнале нет: 1 колебание в секунду вместо двух.

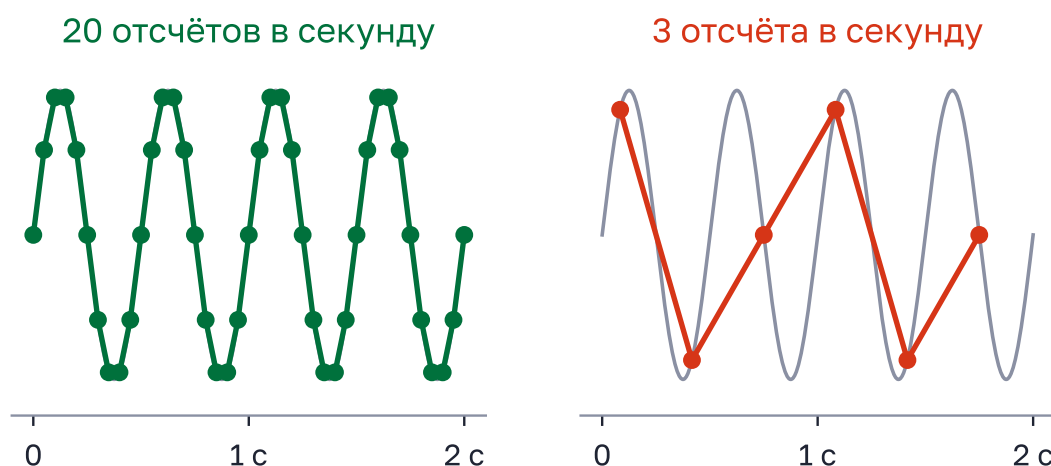


Рисунок 4.4 – Одна и та же волна частотой 2 Гц при частой и при слишком редкой дискретизации

Отсюда правило: чтобы не потерять колебания частотой f , отсчётов в секунду должно быть **больше чем $2f$** . На каждое колебание нужно успеть поймать хотя бы гребень и впадину, а лучше больше точек. Это правило называют **теоремой Котельникова**, в честь советского учёного Владимира Котельникова. Название звучит грозно, а считать тут одно умножение: хочешь записать альфа-ритм до 13 Гц – нужно больше $2 \cdot 13 = 26$ отсчётов в секунду. На практике берут с большим запасом: в задачах старшей НТО по нейротехнологиям ЭЭГ записывали с частотой 250 Гц. Но и частить без толку не стоит – каждый отсчёт занимает память, об этом раздел 4.7.

Запомни. Отсчёт – одно измерение. Частота дискретизации – отсчётов за секунду. Шаг $\Delta t = 1 : f_d$, отсчётов за время t : $n = f_d \cdot t$. Измерять надо чаще, чем $2f$, где f – самые быстрые колебания, которые нужно сохранить.

Совет. Если шаг дан в миллисекундах, сначала переведи его в секунды: $20 \text{ мс} = 0,02 \text{ с}$. Тогда $1 : 0,02 = 50 \text{ Гц}$. Разделишь 1 на 20 – получишь 0,05 «отсчёта в секунду», и это сразу видно как ошибку.

4.4. Детектирование: по уровню и по частоте

НА ПАЛЬЦАХ



Рисунок 4.5 – Порог – планка поперёк графика. Засчитываем удар, когда сигнал перелез через неё снизу вверх

Как прыжки в высоту: засчитывают того, кто перелетел планку. Невысокий горбик до неё не дотянулся – его нет.

Детектирование – поиск в сигнале нужного события: удара сердца, сжатия кулака, вдоха. У него два вида. **Детектирование по уровню (амплитудное)** смотрит, насколько высоко поднялся сигнал. **Детектирование по частоте (частотное)** считает, сколько событий (импульсов) набралось за единицу времени, или сколько времени прошло между событиями. В программе кружка второй вид назван «по периоду»: это то же самое, период и частота – две стороны одного и того же. Бывают и приборы, которые ничего не детектируют: термометр просто показывает число.

По уровню: порог

Выбираем **порог** – число, выше которого считаем, что событие случилось. **Срабатывание** засчитываем в тот момент, когда сигнал был не выше порога и стал выше. Пока сигнал остаётся выше порога, новых срабатываний нет: один широкий пик занимает несколько отсчётов подряд, и если считать каждый отсчёт, из одного удара получится пять.

Разберём ряд из десяти отсчётов с порогом 600:

320, 480, 710, 650, 390, 350, 620, 580, 640, 360.

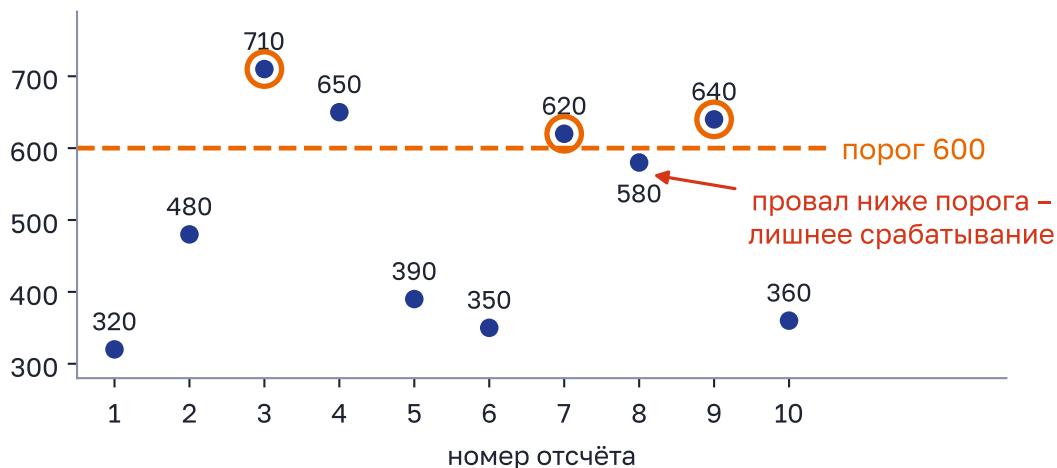


Рисунок 4.6 – Срабатывания порога 600 обведены. Провал на восьмом отсчёте дал лишнее срабатывание

Идём слева направо. Третий отсчёт (710) – сигнал перелез через 600, это срабатывание номер один. Четвёртый (650) ещё выше порога – продолжение того же пика. Пятый и шестой ниже. Седьмой (620) – снова выше, срабатывание номер два. Восьмой (580) на двадцать единиц ниже порога, девятый (640) опять выше – срабатывание номер три. Итого три.

Посмотри на седьмой, восьмой и девятый отсчёты вместе. Скорее всего, это один пик, в середину которого затесалась помеха. Программа честно насчитала три срабатывания, а ударов было два. Порог не отличает настоящий пик от дрожи около планки, и против этого есть три средства: два порога вместо одного (глава 3), детектирование по периоду (ниже) и сглаживание (раздел 4.5).

У пульсовой волны с пальца есть своя ловушка. За каждым высоким пиком идёт горбик пониже (глава 2). Поставишь порог ниже горбика – каждый удар посчитается дважды, и пульс выйдет вдвое больше настоящего. Поставишь выше самых слабых пиков – часть ударов потеряется. Порог должен лечь между горбиками и пиками, а как подбирать его под конкретного человека (калибровка), разобрано в главе 3.

Так считает срабатывания программа на Python. Если Python тебе незнаком, читай по строчкам: после `#` идёт пояснение.

```
signal = [320, 480, 710, 650, 390, 350, 620, 580, 640, 360] # ряд с датчика
porog = 600
count = 0 # сколько срабатываний насчитали
vyshe = False # сигнал сейчас выше порога?
for v in signal: # берём отсчёты по одному
    if v > porog and not vyshe: # был не выше – стал выше
        count = count + 1
        vyshe = True
    if v < porog: # опустился ниже – можно ждать новый пик
        vyshe = False
print(count) # 3
```

Совет. У этой программы есть слабое место – самое начало записи. В начале `vyshе = False`, поэтому если первый же отсчёт выше порога, она его засчитает. А ведь сигнал никуда не перелезал: запись просто началась, когда он уже был наверху. По правилу «был не выше – стал выше» такой отсчёт не считается, сравнить его не с чем. Чтобы программа считала так же, как ты на бумаге, начни с `vyshе = signal[0] > порог`.

Время выше порога. Порог отвечает и на вопрос «сколько времени». Считаем, что каждый отсчёт «действует» до следующего. Тогда число отсчётов выше порога, умноженное на шаг между отсчётами, – это время в секундах. Так БОС-тренажёр считает, сколько секунд мышца была напряжена.

Зона. Иногда одной планки мало и нужен коридор с двумя границами: «пульс от 70 до 80», «взгляд на картинке». Тогда для каждого отсчёта спрашиваем «внутри или снаружи». Время в зоне – отсчёты внутри, умноженные на шаг. **Вход в зону** – отсчёт внутри, перед которым был отсчёт снаружи. Войти в коридор можно с двух сторон: снизу, как через порог, и сверху, когда сигнал опускается. Про границы читай условие внимательно: «от 70 до 80 включительно» значит, что 70 и 80 – ещё внутри.

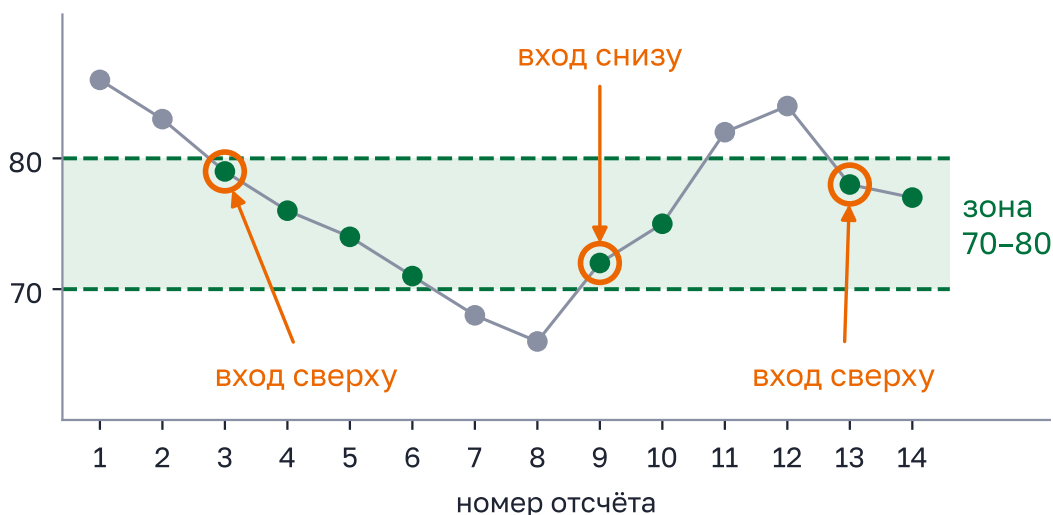


Рисунок 4.7 – Коридор от 70 до 80. Зелёные точки внутри, серые снаружи. Входов три, и два из них – сверху

По частоте (по периоду)

НА ПАЛЬЦАХ

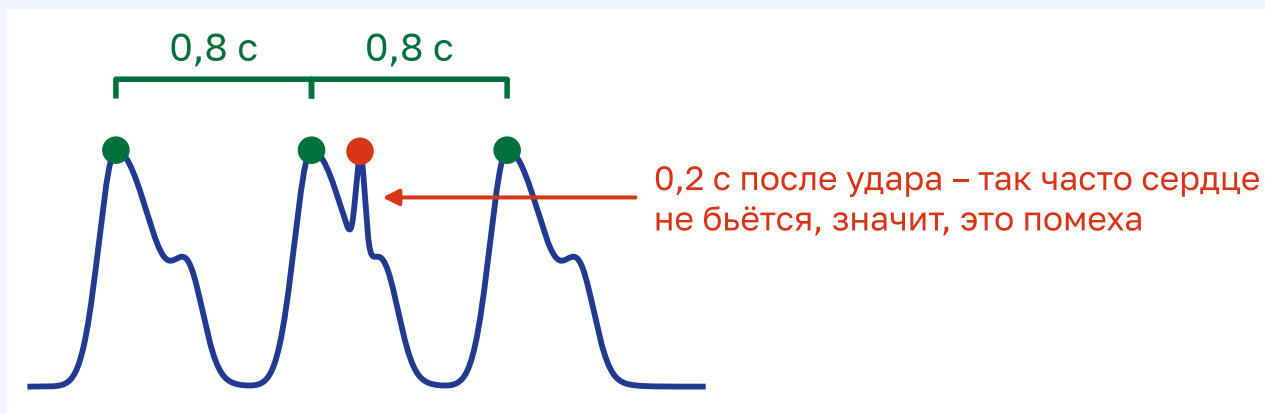


Рисунок 4.8 – Удары сердца идут через одинаковые промежутки. Всплеск, который пришёл слишком рано, – не удар

Следы на снегу идут шаг за шагом, примерно на одном расстоянии. Если между двумя «следами» ширина ладони, это не шаг – кто-то уронил варезку.

Детектированию по периоду не важна высота пика, оно смотрит на время между событиями. Пригодается оно в трёх случаях.

Найти частоту. Программа знает номера отсчётов, на которых сработал порог. Номер отсчёта – это та же шкала времени, только в «шагах». Разница номеров, делённая на частоту дискретизации, – время в секундах. Пусть при 50 Гц пики нашлись на отсчётах 12 и 52. Между ними $52 - 12 = 40$ отсчётов, это $40 : 50 = 0,8$ с, а пульс $60 : 0,8 = 75$ уд/мин. Если пиков много, бери первый и последний, а промежутков считай на один меньше, чем пиков (как с пролётами забора в главе 2).

Считать обороты по импульсам. Датчик колеса даёт импульс на каждый зуб шестерёнки. У колеса 48 зубьев, за секунду пришло 96 импульсов: $96 : 48 = 2$ оборота в секунду. Высота импульсов тут никого не волнует, важно только, сколько их.

Отсеять невозможное. Разработчик пульсометра решает, например: ударов чаще 200 в минуту наш прибор не ждёт. Значит, между ударами не меньше $60 : 200 = 0,3$ с, и срабатывание, которое пришло раньше, – помеха. В ряду выше, если шаг 0,1 с, седьмой и девятый отсчёты разделяет всего 0,2 с, и третье срабатывание можно выбросить.

Запомни. Срабатывание – сигнал был не выше порога и стал выше. Один пик – одно срабатывание, сколько бы отсчётов он ни занимал. Время выше порога = число отсчётов выше порога · шаг. Равное порогу значение считается или нет – решает условие («выше» – не считается, «не меньше» – считается). Частота по номерам отсчётов: разница номеров делится на f_d – получаются секунды.

4.5. Сглаживание: скользящее среднее

НА ПАЛЬЦАХ



окошко из трёх чисел сдвигается на одно; результат пишем под средним числом окошка

Рисунок 4.9 – Окошко из трёх чисел едет по ряду, и каждый раз мы берём среднее того, что в него попало

Одна двойка среди пятёрок сильно портит день, но в средней за неделю она почти не видна. Сглаживание работает так же.

Самый простой способ убрать шум – **скользящее среднее**. Берём первые три отсчёта и находим их среднее: сложить и разделить на 3. Результат записываем **под средней точкой окошка** – так окно называют центрированным. Для окна из трёх чисел это сама точка и два соседа, слева и справа: среднее для отсчёта 2 берёт отсчёты 1, 2 и 3. Потом сдвигаем «окошко» на одно число вправо и снова берём среднее трёх. И так до конца ряда. Центрирование нужно, чтобы линия не сдвигалась по времени: сглаженный пик стоит там же, где был исходный.

Ряд с датчика: 480, 510, 810, 540, 510, 540. Число 810 выбивается – похоже на помеху.

- под отсчётом 2: $(480 + 510 + 810) : 3 = 1800 : 3 = 600$;
- под отсчётом 3: $(510 + 810 + 540) : 3 = 1860 : 3 = 620$;
- под отсчётом 4: $(810 + 540 + 510) : 3 = 1860 : 3 = 620$;
- под отсчётом 5: $(540 + 510 + 540) : 3 = 1590 : 3 = 530$.

Сглаженный ряд: у отсчётов 2, 3, 4, 5 значения 600, 620, 620, 530. У отсчётов 1 и 6 среднего нет: у первого нет соседа слева, у последнего – справа. Выброс 810 пропал, вместо него – небольшой бугорок 620. Зато его след попал в три соседние средние: помеха не исчезла, её «размазало» тонким слоем.

Сколько чисел получится? Окошко из трёх чисел можно поставить на ряд из шести чисел четырьмя способами. В общем случае из n чисел окошком по k выходит $n - k + 1$ средних: по три – на два меньше, по пять – на четыре меньше. Пропадают крайние значения, по $(k - 1) : 2$ с каждого края: для окна 3 – по одному, для окна 5 – по два (среднее для отсчёта i берёт отсчёты от $i - 2$ до $i + 2$).

Нумерация зависит от условия задачи. В курсе и в этой книге среднее подписывают номером средней точки окошка. Если в условии сказано иначе («по последним трём отсчётам»), делай так, как сказано.

Размер окошка – это выбор. Чем оно шире, тем глаже линия, но у сглаживания есть цена.

- **Пики становятся ниже.** Острая вершушка усредняется с соседями и проседает. Если после сглаживания оставить тот же порог, слабый пик может под него не дотянуться.
- **В живом режиме линия запаздывает.** Окно с центром в точке требует отсчётов из будущего, а плате, которая считает на ходу, их взять негде. Поэтому для живого сигнала берут окно «по последним k отсчётам» и узнают о подъёме сигнала позже, чем он случился. Окошко из 5 отсчётов при 50 Гц – это 0,1 с, для пульса терпимо, а для игры «сожми кулак» уже заметно. Центрированное окно годится, когда запись уже лежит целиком, как в таблице.



Рисунок 4.10 – Шумная пульсовая волна и она же после скользящего среднего по 5 отсчётам

Если сдвиг одинаковый для всех пиков (у окна «по последним»), частоту он не портит: все удары опаздывают на одно и то же время, и промежутки между ними не меняются. Поэтому пульс по сглаженному ряду считают так же, как по сырому.

Скользящее среднее на Python:

```
a = [480, 510, 810, 540, 510, 540]
k = 3                                # ширина окошка
smooth = []
for i in range(len(a) - k + 1):      # сколько раз можно поставить окошко
    okno = a[i:i + k]                # k чисел, начиная с номера i
    smooth.append(sum(okno) / k)      # это среднее стоит под средней точкой окошка: номер i
+ 2 с единицы
print(smooth)                        # [600.0, 620.0, 620.0, 530.0]
```

Запомни. Скользящее среднее: окошко из k чисел едет по ряду, каждое положение даёт одно среднее, и его пишут под средней точкой окошка (для окна 3 – сама точка и соседи слева и справа). Из n чисел получается $n - k + 1$, на краях средних нет. Шире окошко – глаже линия, ниже пики; окно «по последним» ещё и запаздывает.

Совет. Считая вручную, не складывай заново все три числа. Сдвинул окошко – вычти число, которое вышло слева, и прибавь то, что вошло справа: $1800 - 480 + 540 = 1860$.

4.6. Спектр и фильтр – на пальцах

НА ПАЛЬЦАХ



Рисунок 4.11 – Сложный сигнал раскладывается на простые волны. Спектр показывает, какие в нём частоты и сколько каждой

Аккорд на гитаре звучит одним звуком, но в нём несколько нот. Спектр – это список нот в аккорде и громкость каждой.

Сигнал с датчика редко похож на ровную волну. Но его почти всегда можно представить как **сумму** простых волн разной частоты: медленной, побыстрее, совсем быстрой. На рисунке медленная волна в 1 Гц плюс мелкая рябь в 12 Гц дают кривую, похожую на настоящую запись.

Спектр сигнала – это ответ на вопрос «из каких частот он сложен и какой вклад у каждой». Его рисуют столбиками: по горизонтали частота, по вертикали – насколько сильна волна этой частоты. На рисунке высокий столбик у 1 Гц и низкий у 12 Гц: медленная волна главная, рябь слабая. Как компьютер раскладывает сигнал на волны, учат уже в институте, и считать спектр вручную в этой книге не придётся. Достаточно уметь читать столбики.

Зачем это медику? ЭЭГ на глаз – беспорядочные каракули. А в спектре видно, какой ритм сильнее. Человек закрыл глаза и расслабился – растёт столбик в районе 8–13 Гц, то есть альфа-ритм (подробнее в главе 3). На этом построены БОС-тренинги «на расслабление»: игра следит за высотой альфа-столбика.

Фильтр – обработка, которая убирает из сигнала часть частот. Работает как сито.

- **Фильтр низких частот** пропускает медленные колебания и гасит быстрые. Скользящее среднее из раздела 4.5 – самый простой фильтр низких частот: быстрая дрожь в окошке усредняется в ноль, а медленная волна остаётся.
- **Фильтр высоких частот** делает наоборот: гасит медленное. Его ставят, когда вся линия сигнала медленно «уплывает» вверх или вниз, например из-за подсыхающего электрода.
- **Полосовой фильтр** оставляет только полосу частот. Хочешь смотреть на альфа-ритм – оставь 8–13 Гц.
- **Фильтр 50 Гц** вырезает наводку от электросети, про которую говорили в разделе 4.1.

В задачах про фильтр всё решает граница: что по одну её сторону, проходит, что по другую – гасится. Какие колебания останутся, узнаёшь простым сравнением чисел.

Запомни. Спектр – из каких частот сложен сигнал и сколько каждой. Фильтр низких частот пропускает медленное и гасит быстрое, высоких – наоборот, полосовой оставляет полосу. Скользящее среднее – простейший фильтр низких частот.

4.7. Кодирование и передача

НА ПАЛЬЦАХ



Рисунок 4.12 – Число 600 записали нулями и единицами и передали вспышками светодиода

Ночью в походе можно переговариваться фонариком: короткая вспышка, длинная, пауза. Плата делает то же самое, только очень быстро и с одними нулями и единицами.

Сообщение само по проводу не пройдёт – его надо перевести в сигнал. Правило, по которому сообщение превращают в сигнал, называют **кодом**. Старый пример – азбука Морзе: каждая буква – набор коротких и длинных сигналов. Сигнал бедствия SOS – три коротких, три длинных, три коротких. Его можно простучать, просигнализировать фонарём или пропищать.

Компьютер и плата пользуются **двоичным кодом**: только 0 и 1. Одна такая цифра называется **битом**. Один бит – два варианта: 0 или 1. Два бита – четыре варианта (00, 01, 10, 11). Каждый новый бит удваивает число вариантов.

Таблица 4.1

Битов	Вариантов (уровней)
1	2
2	4
8	256
10	1024
12	4096
16	65 536

Теперь понятно, откуда у Arduino числа от 0 до 1023: её АЦП выдаёт **10 бит** на отсчёт, а это 1024 варианта. Ноль тоже вариант, поэтому последний номер – 1023.

Для любопытных: как прочитать 1001011000. Переводить числа из двоичной записи учат на информатике классом-двумя позже, и в задачах этой главы это не понадобится. Но разгадать картинку в начале раздела можно уже сейчас. У каждой позиции свой «вес»: справа на лево 1, 2, 4, 8, 16 и так далее, каждый следующий вдвое больше. Единица значит «вес берём», ноль – «не берём». Для десяти битов веса слева направо такие: 512, 256, 128, 64, 32, 16, 8, 4, 2, 1. В записи 1001011000 единицы стоят на местах с весами 512, 64, 16 и 8:
 $512 + 64 + 16 + 8 = 600$.

Сколько данных выходит. Каждый отсчёт – сколько-то битов, отсчётов много, а каналов (датчиков) может быть несколько. Отсюда **поток данных** – сколько битов в секунду выдаёт прибор:

$$\text{поток (бит/с)} = \text{каналы} \cdot \text{бит на отсчёт} \cdot f_{\text{д}}.$$

Пульсометр с одним датчиком, 10 бит на отсчёт, 50 отсчётов в секунду: $1 \cdot 10 \cdot 50 = 500$ бит/с. За минуту $500 \cdot 60 = 30\,000$ бит. Объём файлов считают в **байтах**: 1 байт = 8 бит, 1 Кбайт = 1024 байта. Выходит $30\,000 : 8 = 3750$ байт – меньше четырёх килобайт, примерно как страница текста.

Для любопытных: несущая и модуляция. Слово «модуляция» есть в программе кружка, но задач на неё в этой главе нет. По радио сообщение передают так: берут быструю ровную волну – **несущую** – и меняют её, чтобы она «несла» сообщение. Меняют размах – это **амплитудная модуляция**, меняют частоту – **частотная**. Радио FM так и называется: frequency modulation, частотная модуляция. Мигание фонарика – самый простой случай: несущую то включают, то выключают. Плавную линию вдоль верхушек быстрых колебаний называют **огibaющей**, мы встречали её в главе 3 у ЭМГ. Получают её выпрямлением (берут модуль) и сглаживанием окном длиной в один период несущей, поэтому она идёт ниже верхушек. Мышца даёт быструю дрожь, а сила сжатия спрятана в огibaющей.

Запомни. Бит – одна двоичная цифра, b бит дают 2^b уровней, 10 бит – 1024 (от 0 до 1023). Поток = каналы · бит на отсчёт · отсчётов в секунду. 1 байт = 8 бит, 1 Кбайт = 1024 байта.

Совет. В задачах на поток данных выпиши три множителя отдельными строчками: каналы, биты, отсчёты в секунду. Потом посмотри, в чём просят ответ – в битах или байтах, за секунду или за всю запись.

4.8. Разбираем задачи

Задача 4.1 Размах пульсовой волны

разминка · максимум, минимум, размах, амплитуда

Лена надела на палец оптический датчик пульса. Программа вывела 12 отсчётов подряд (в условных единицах АЦП):

486, 530, 611, 648, 597, 522, 470, 454, 489, 541, 620, 639.

Найди: а) наибольшее и наименьшее значение; б) размах; в) амплитуду, если считать её половиной размаха.

Дано: 12 отсчётов.

Найти: $\max$, $\min$, размах, амплитуду A .

Решение

Шаг 1. Максимум. Проходим ряд и держим в голове самое большое число: 486, потом 530, потом 611, потом 648. Дальше ничего больше 648 нет (ближе всех 639 в конце). $\max = 648$.

Шаг 2. Минимум. Так же ищем самое маленькое: 486, потом 470, потом 454. Меньше 454 в ряду нет. $\min = 454$.

Шаг 3. Размах. $648 - 454 = 194$.

Шаг 4. Амплитуда. Половина размаха: $194 : 2 = 97$.

Прикинем в вольтах: у Arduino одна ступенька – примерно 4,9 мВ, значит, размах около $194 \cdot 4,9 \approx 950$ мВ, почти вольт. Для датчика пульса с усилителем это нормальная величина.

 **Типичная ошибка.** Взять за максимум и минимум первое и последнее число или «на глаз» самое большое в начале ряда. В этом ряду второй пик (639) почти догоняет первый – проверь каждое число, прежде чем решить. Вторая ошибка – сдать в пункте в) 648, то есть максимум: амплитуда отсчитывается от среднего уровня, а не от нуля.

Ответ: а) 648 и 454; б) 194; в) 97.

Своя задача по программе сферы, модуль 3

Реши сам

4.1. Датчик ЭМГ на предплечье выдал 10 отсчётов: 512, 538, 590, 471, 438, 560, 602, 455, 498, 531. Найди размах и амплитуду (половину размаха).

4.2. В симуляторе пульсовую волну делают программой: ровная волна качается вокруг среднего уровня 500 с амплитудой 120. Какое наибольшее и какое наименьшее число покажет монитор порта? Каков размах?

Задача 4.2 Сколько измерений за тренировку

разминка · частота дискретизации, шаг, число отсчётов

Программа пульсометра измеряет сигнал датчика каждые 20 мс. а) Какова частота дискретизации в герцах? б) Сколько отсчётов наберётся за 3 минуты записи?

Дано: $\Delta t = 20$ мс; $t = 3$ мин.

Найти: f_d – ?; n – ?

Решение

Шаг 1. Шаг в секундах. В секунде 1000 мс: $20:1000 = 0,02$ с.

Шаг 2. Частота дискретизации. Сколько раз по 0,02 с помещается в секунде:
 $f_d = 1:0,02 = 50$ Гц, то есть 50 отсчётов в секунду. Без перевода тоже можно:
 $1000:20 = 50$.

Шаг 3. Время записи в секундах. $3 \cdot 60 = 180$ с.

Шаг 4. Число отсчётов. $n = f_d \cdot t = 50 \cdot 180 = 9000$.

50 измерений в секунду для пульса с запасом: удар длится больше десятой доли секунды, и на каждый пик придётся несколько точек.

⚠ Типичная ошибка. Забыть перевести минуты в секунды и получить $50 \cdot 3 = 150$ отсчётов. Проверь: 150 отсчётов плата набрала бы за 3 секунды, а не за 3 минуты.

Ответ: а) 50 Гц; б) 9000 отсчётов.

Своя задача по программе сферы, модуль 3

Реши сам

4.3. Учебный ЭЭГ-ободок делает 250 отсчётов в секунду. Сколько миллисекунд проходит между соседними отсчётами? Сколько отсчётов он запишет за 12 секунд?

4.4. Команда хочет записывать альфа-ритм ЭЭГ, считая, что его самые быстрые колебания – 13 Гц. Плату можно настроить на 10, 25, 40 или 100 отсчётов в секунду. Какую **наименьшую** частоту дискретизации из списка можно выбрать, чтобы по теореме Котельникова не потерять альфа-ритм?

Задача 4.3 Пульс в зелёной зоне

зона, время в зоне, входы в зону

Артём проходит БОС-тренинг на расслабление. Браслет каждые 2 с присылает пульс, и пока он в зелёной зоне – от 70 до 80 уд/мин включительно, – в игре распускаются цветы. Вот 20 значений подряд:

92, 88, 85, 81, 79, 77, 78, 82, 80, 76, 74, 72, 69, 71, 73, 83, 78, 75, 74, 76.

Считай, что каждое значение держится до прихода следующего. а) Сколько секунд пульс был в зелёной зоне? б) Сколько раз пульс входил в зелёную зону?

Дано: 20 значений; шаг $\Delta t = 2$ с; зона от 70 до 80 включительно.

Найти: время в зоне – ?; число входов – ?

Решение

Шаг 1. Помечаем каждое значение. Внутри зоны – 3, снаружи – С. Граница 80 – внутри, 69 и 81 – снаружи.

92 С, 88 С, 85 С, 81 С, 79 З, 77 З, 78 З, 82 С, 80 З, 76 З, 74 З, 72 З, 69 С, 71 З, 73 З, 83 С, 78 З, 75 З, 74 З, 76 З.

Шаг 2. Время в зоне. Букв З: $3 + 4 + 2 + 4 = 13$. Каждое значение держится 2 с: $13 \cdot 2 = 26$ с.

Шаг 3. Входы в зону. Вход – место, где после С стоит З. Таких мест четыре: $81 \rightarrow 79$, $82 \rightarrow 80$, $69 \rightarrow 71$ и $83 \rightarrow 78$. Третий вход – снизу: пульс провалился до 69 и снова поднялся в зону. Остальные три – сверху.

Вся запись – $20 \cdot 2 = 40$ с, и две трети её Артём был в зоне. Для начала тренировки, когда пульс только успокаивается после 92, вполне правдоподобно.

 **Типичная ошибка.** Считать зону как порог: «вошёл – значит, поднялся выше 70». Тогда считаешь только подъём $69 \rightarrow 71$, а три входа сверху потеряешь. Самое первое 92 подъёмом через 70 не считай: сравнить его не с чем, запись просто началась выше планки. Программа из раздела 4.4 без правки из совета после неё его засчитает и выдаст 2 вместо 1. Вторая ошибка – выкинуть 80 из зоны: слово «включительно» в условии говорит, что граница внутри. Без неё время выйдет 24 с.

Ответ: а) 26 с; б) 4 раза.

По мотивам задачи второго отборочного этапа НТО «Нейротехнологии и когнитивные науки» 2025/26, инженерный тур, «Айтрекинг на параллельных стимулах»

Реши сам

4.5. Айтрекер – прибор, который следит, куда смотрит человек, – 20 раз в секунду пишет положение взгляда по горизонтали: 0 – левый край экрана, 1 – правый. Картинка занимает полосу от 0,6 до 0,9. Запись: 0,21; 0,35; 0,62; 0,71; 0,74; 0,69; 0,55; 0,41; 0,38; 0,66; 0,78; 0,85; 0,88; 0,93; 0,81; 0,52; 0,47; 0,64; 0,70; 0,33. Сколько секунд человек смотрел на картинку? Сколько раз взгляд на неё попадал?

4.6. Пульсовая волна с пальца, у каждого удара высокий пик и за ним горбик пониже: 380, 420, 690, 740, 560, 440, 520, 410, 390, 430, 700, 760, 580, 445, 530, 400. а) Сколько срабатываний насчитает программа с порогом 450? б) С порогом 600? в) Какой из двух порогов считает удары правильно?

Задача 4.4 Пульс по номерам отсчётов

детектирование по периоду, частота по номерам отсчётов

Пульсометр делает 100 отсчётов в секунду. Детектор нашёл удары сердца на отсчётах с номерами 18, 98, 177, 259, 338. Найди ЧСС в ударах в минуту и частоту ударов в герцах.

Дано: $f_d = 100$ Гц; удары на отсчётах 18, 98, 177, 259, 338.

Найти: ЧСС – ?; f – ?

Решение

Шаг 1. Сколько промежутков. Ударов 5, промежутков между ними на один меньше: 4.

Шаг 2. Сколько отсчётов от первого удара до последнего. $338 - 18 = 320$ отсчётов.

Шаг 3. Средний промежуток в отсчётах. $320 : 4 = 80$ отсчётов. Можно проверить по соседям: $98 - 18 = 80$, $177 - 98 = 79$, $259 - 177 = 82$, $338 - 259 = 79$. Промежутки чуть разные (сердце не метроном), а в среднее – 80.

Шаг 4. Переводим в секунды. Один отсчёт – $1 : 100 = 0,01$ с. $80 \cdot 0,01 = 0,8$ с. Или сразу $80 : 100 = 0,8$ с.

Шаг 5. ЧСС. $60 : 0,8 = 75$ уд/мин.

Шаг 6. Частота в герцах. $1 : 0,8 = 1,25$ Гц. Или $75 : 60 = 1,25$ Гц.

75 ударов в минуту – спокойный пульс, чуть больше удара в секунду. Номера отсчётов тоже согласны: от первого до последнего удара $320 : 100 = 3,2$ с, за это время сердце ударило 4 раза после первого.

 **Типичная ошибка.** Принять номера отсчётов за миллисекунды: «80 мс между ударами, пульс 750». При 100 отсчётах в секунду шаг – 10 мс, а не 1 мс. Другая ошибка – делить 320 на 5 ударов вместо 4 промежутков: выйдет 64 отсчёта и пульс 94 уд/мин.

Ответ: 75 уд/мин; 1,25 Гц.

Своя задача по программе сферы, модуль 6

Реши сам

4.7. ЭЭГ записывают с частотой 250 Гц. Программа нашла гребни одной и той же волны на отсчётах 30, 55, 80, 104, 130, 155. Найди частоту этой волны в герцах. Какой это ритм: альфа (примерно 8–13 Гц) или бета (примерно 13–30 Гц)?

4.8. Пульсометр делает 40 отсчётов в секунду. Пульс у Димы 75 уд/мин. Через сколько отсчётов друг от друга будут стоять соседние пики на записи?

Задача 4.5 Уберём выброс

скользящее среднее, выброс

Костя дёрнул пальцем, и в запись датчика пульса попал выброс. Ряд: 510, 540, 870, 570, 600, 630, 600, 630. Сгладь его скользящим средним по трём точкам. а) Сколько чисел будет в сглаженном ряду? б) Запиши сглаженный ряд. в) На сколько наибольшее значение сглаженного ряда меньше наибольшего значения исходного?

Дано: 8 чисел; окошко $k = 3$.

Найти: число средних – ?; сглаженный ряд – ?; разницу максимумов – ?

Решение

Шаг 1. Сколько средних. Из $n = 8$ чисел окошком по 3 выходит $8 - 3 + 1 = 6$ средних.

Шаг 2. Считаем суммы. Первую – целиком, дальше сдвигаем окошко: вычитаем ушедшее число, прибавляем пришедшее.

- $510 + 540 + 870 = 1920$;
- $1920 - 510 + 570 = 1980$;
- $1980 - 540 + 600 = 2040$;
- $2040 - 870 + 630 = 1800$;
- $1800 - 570 + 600 = 1830$;
- $1830 - 600 + 630 = 1860$.

Шаг 3. Делим каждую сумму на 3. 640, 660, 680, 600, 610, 620.

Шаг 4. Сравниваем максимумы. В исходном ряду наибольшее 870, в сглаженном – 680. Разница $870 - 680 = 190$.

Эти шесть средних стоят под отсчётами со 2-го по 7-й (среднее 640 – под отсчётом 2, 620 – под отсчётом 7); у 1-го и 8-го отсчётов среднего нет. Выброс не пропал бесследно: он поднял первые три средних (640, 660, 680), а дальше ряд спокойно идёт около 600–620, как и настоящий сигнал. Сглаженный ряд лежит между самым маленьким и самым большим числом исходного – у средних иначе не бывает.

⚠ Типичная ошибка. Сдвигать окошко не на одно число, а на три: (510, 540, 870), потом (570, 600, 630) – и получить всего два средних. Скользящее окошко шагает по одному числу, соседние окошки перекрываются.

Ответ: а) 6; б) 640, 660, 680, 600, 610, 620; в) на 190.

Своя задача по программе сферы, модуль 3

Реши сам

4.9. Сгладь ряд скользящим средним по трём точкам: 300, 330, 360, 540, 390, 420, 450.

4.10. Фильтр низких частот настроен так: колебания медленнее 20 Гц пропускает, быстрее – гасит. В сигнал с датчика попали: 1) дыхание, около 0,3 Гц; 2) пульс, около 1,2 Гц; 3) альфа-ритм ЭЭГ, около 10 Гц; 4) наводка от электросети, 50 Гц. Какие колебания останутся после фильтра? Ответ – номера по возрастанию без пробелов.

Задача 4.6 Сколько весит запись ЭЭГ

двоичный код, поток данных, биты и байты

Учебный ЭЭГ-ободок записывает 2 канала. Каждый отсчёт АЦП занимает 10 бит, отсчётов – 250 в секунду на каждый канал. а) Сколько бит в секунду передаёт ободок в компьютер? б) Сколько байт займёт запись длиной 1 минута?

Дано: 2 канала; 10 бит на отсчёт; $f_d = 250$ Гц; $t = 1$ мин.

Найти: поток в бит/с – ?; объём в байтах – ?

Решение

Шаг 1. Один канал за секунду. 250 отсчётов по 10 бит: $250 \cdot 10 = 2500$ бит.

Шаг 2. Два канала. $2500 \cdot 2 = 5000$ бит/с. По формуле: $2 \cdot 10 \cdot 250 = 5000$.

Шаг 3. Минута в секундах. 60 с.

Шаг 4. Биты за минуту. $5000 \cdot 60 = 300\,000$ бит.

Шаг 5. Биты в байты. В байте 8 бит: $300\,000 : 8 = 37\,500$ байт.

Около 37 килобайт – фотография с телефона весит в десятки раз больше. А вот у прибора с 19 каналами при тех же настройках поток вышел бы почти в десять раз больше.

 **Типичная ошибка.** Забыть один из множителей – каналы или биты. Первое даёт 2500 бит/с, второе – 500 «бит» в секунду. Ещё одна частая ошибка в конце: умножить на 8 вместо деления. В байтах число всегда **меньше**, чем в битах.

Ответ: а) 5000 бит/с; б) 37 500 байт.

По мотивам задачи первого отборочного этапа НТО «Нейротехнологии и когнитивные науки» 2025/26, инженерный тур, «Приём данных»

Реши сам

4.11. Браслет для бионической руки снимает ЭМГ с 4 мышц (4 канала). Отсчёт – 12 бит, 500 отсчётов в секунду на канал. Найди поток данных в бит/с и объём записи длиной 10 секунд в байтах.

4.12.* Самодельный регистратор пульса пишет 1 канал: 10 бит на отсчёт, 50 отсчётов в секунду. Под запись на карте памяти отвели 16 Кбайт (1 Кбайт = 1024 байта). На сколько **полных** секунд записи хватит этого места?

Задача 4.7* Шум мешает считать пульс

шум, порог, скользящее среднее, ЧСС по отсчётам

Пульсометр делает отсчёт каждые 0,1 с. Удар засчитывается, когда сигнал переходит порог 600 снизу вверх. Запись из 26 отсчётов (на рисунке – те же числа):

430, 450, 620, 710, 650, 520, 470, 440, 430, 460, 640, 580, 660, 530, 480, 450, 440, 460, 610, 720, 640, 530, 620, 450, 440, 460.

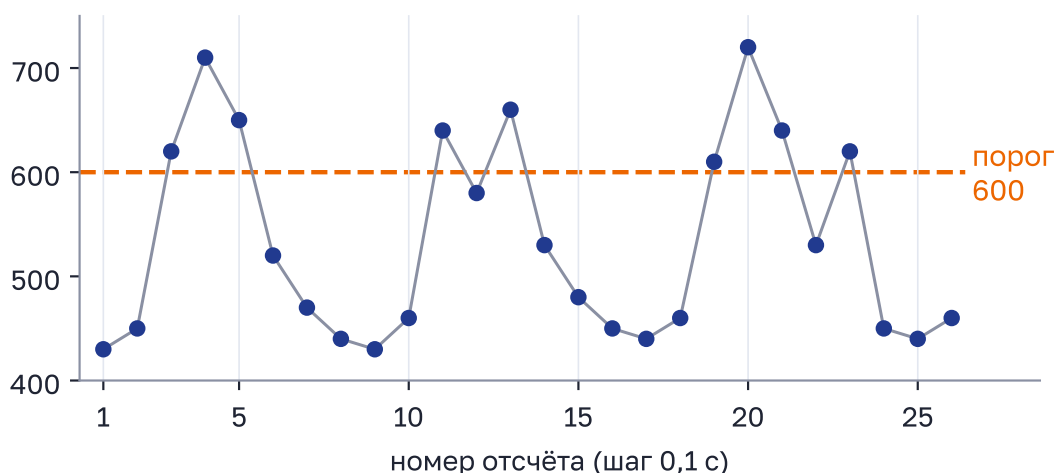


Рисунок 4.13 – Запись к задаче 4.7: 26 отсчётов и порог 600

а) Сколько срабатываний насчитает программа по этому ряду? б) Программу доработали: сначала она сглаживает ряд скользящим средним по трём точкам, а потом ищет срабатывания того же порога. Сколько срабатываний теперь? в) Найди ЧСС по сглаженному ряду: по номерам первого и последнего срабатывания. Ответ в ударах в минуту.

Дано: 26 отсчётов; $\Delta t = 0,1$ с; порог 600; окошко 3.

Найти: срабатывания по сырому ряду – ?; по сглаженному – ?; ЧСС – ?

Решение

Шаг 1. Срабатывания по сырому ряду. Ищем места, где число было ниже 600, а следующее выше.

- отсчёт 3 (450 → 620) – первый пик;
- отсчёт 11 (460 → 640) – второй пик начался;
- отсчёт 13 (580 → 660) – провал до 580 посреди второго пика, и программа засчитала его ещё раз;
- отсчёт 19 (460 → 610) – третий пик;
- отсчёт 23 (530 → 620) – одиночный всплеск на спаде третьего пика.

Срабатываний 5, а высоких пиков на рисунке три.

Шаг 2. Сглаживаем с умом. Все 24 средних считать не обязательно. Срабатывания могут появиться только там, где сигнал близок к порогу, поэтому посчитаем средние вокруг трёх пиков и вокруг обоих подозрительных мест. Средние нумеруем по средней точке окошка: среднее номер 4 – это отсчёты 3, 4, 5 (у отсчётов 1 и 26 среднего нет).

- Первый пик: №3 (450 + 620 + 710): 3 = 593,3 – ниже 600; №4 (620 + 710 + 650): 3 = 660 – выше. Срабатывание на №4. Дальше №5 = 626,7, №6 = 546,7 – опустились.
- Второй пик: №11 (460 + 640 + 580): 3 = 560 – ниже; №12 (640 + 580 + 660): 3 = 626,7 – выше, срабатывание на №12; №13 (580 + 660 + 530): 3 = 590 – ниже; №14 = 556,7 – ниже. Провал 580 растворился: одно срабатывание вместо двух.

- Третий пик: №19 $(460 + 610 + 720) : 3 = 596,7$ – ниже; №20 $(610 + 720 + 640) : 3 = 656,7$ – выше, срабатывание на №20; №21 = 630; №22 $(640 + 530 + 620) : 3 = 596,7$ – ниже; №23 $(530 + 620 + 450) : 3 = 533,3$ – ниже. Всплеск 620 утонул в соседях.

Остальные средние далеко от 600 (от 443 до 510), их можно не проверять. Срабатываний 3 – по одному на пик.

Шаг 3. Промежутки. Срабатывания на средних №4, №12, №20. Все средние сдвинуты одинаково, поэтому промежутки между ними верные. От первого до последнего $20 - 4 = 16$ шагов, промежутков 2, на один $16 : 2 = 8$ шагов.

Шаг 4. В секунды. $8 \cdot 0,1 = 0,8$ с.

Шаг 5. ЧСС. $60 : 0,8 = 75$ уд/мин.

Проверим детектированием по периоду. В сыром ряду срабатывание 13 пришло через $(13 - 11) \cdot 0,1 = 0,2$ с после 11 – это был бы пульс $60 : 0,2 = 300$ уд/мин, такого не бывает, значит, помеха. А вот всплеск 23 пришёл через 0,4 с после 19, «пульс» 150 – по одному времени его не распознать. Сглаживание убрало обе помехи сразу.

Вся проверка на Python:

```
raw = [430, 450, 620, 710, 650, 520, 470, 440, 430, 460, 640, 580, 660,
       530, 480, 450, 440, 460, 610, 720, 640, 530, 620, 450, 440, 460]

def srbatyvaniya(ryad, porog, sdvig=0):
    nomera = [] # номера, где сработал порог (sdvig = 1 для средних: у первого
    среднего номер 2)
    vyshe = False
    for i in range(len(ryad)):
        if ryad[i] > porog and not vyshe:
            nomera.append(i + 1 + sdvig) # нумеруем с единицы, как в условии
            vyshe = True
        if ryad[i] < porog:
            vyshe = False
    return nomera

smooth = [(raw[i] + raw[i + 1] + raw[i + 2]) / 3 for i in range(len(raw) - 2)]
print(srbatyvaniya(raw, 600)) # [3, 11, 13, 19, 23]
s = srbatyvaniya(smooth, 600, 1)
print(s) # [4, 12, 20]
T = (s[-1] - s[0]) / (len(s) - 1) * 0.1
print(round(60 / T)) # 75
```

⚠ Типичная ошибка. Посчитать пульс по сырому ряду: 5 срабатываний за 2,6 с записи – это $5 \cdot 60 : 2,6 \approx 115$ уд/мин вместо 75. Ещё одна ловушка – после сглаживания взять номера отсчётов сырого ряда, а не средних. Для частоты это неважно (промежутки те же), но в пункте б) считать надо по сглаженному ряду, иначе провал 580 снова даст лишнее срабатывание.

Ответ: а) 5; б) 3; в) 75 уд/мин.

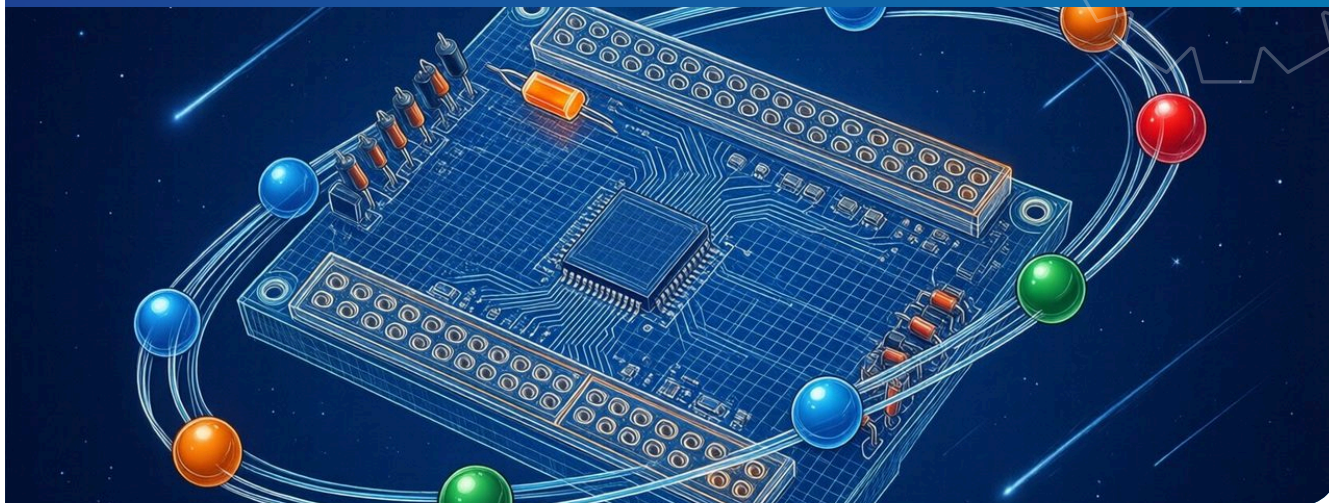
Своя задача по программе сферы, модуль 6

Реши сам

4.13.* Дыхательный пояс делает отсчёт каждые 0,5 с. Вдох засчитывается, когда сигнал переходит порог 500 снизу вверх. В записи – кашель: 380, 420, 530, 610, 640, 580, 460, 590, 390, 400, 540, 620, 650, 570, 450, 390, 370, 360. а) Сколько срабатываний по сырому ряду? б) Сколько после скользящего среднего по трём точкам? в) Найди частоту дыхания во вдохах в минуту по сглаженному ряду.

4.14.* Катя пошевелила пальцем, и в запись пульса попал выброс: 600, 615, 630, 1110, 645, 660, 675, 690, 705. Сгладь ряд скользящим средним по трём точкам, а потом отдельно – по пяти точкам (среднее пиши под средней точкой окошка). Сколько чисел в каждом сглаженном ряду и какое в нём наибольшее значение?

Электричество и датчики



Разбери пульсометр, фитнес-браслет или ЭМГ-руку – внутри найдёшь одно и то же: датчик превращает что-то живое в напряжение, а плата меряет это напряжение и получает число. Физику в школе ещё не проходили? Не страшно: ток, напряжение и сопротивление здесь начинаются с нуля, с воды в трубах. Потом – резистор для светодиода, датчики, перевод числа `analogRead` в вольты и срок жизни аккумулятора. В задачах на эту тему – схема или прибор, несколько чисел с единицами, а в ответ – одно число.

5.1. Цепь, ток и напряжение

НА ПАЛЬЦАХ

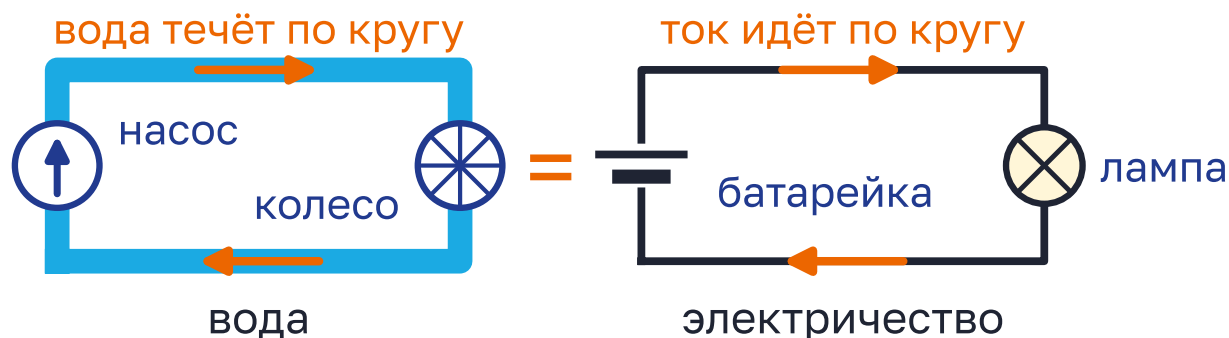


Рисунок 5.1 – Вода в трубах и ток в проводах устроены одинаково

Насос гонит воду по кругу, и вода крутит колесо. Батарейка гонит по проводам ток, и загорается лампа. Разорви трубу или провод – всё остановится.

Внутри проводов есть крошечные заряженные частицы – **электроны**. Пока провод просто лежит на столе, они толкуются на месте. Подключи провод к батарейке – и они все вместе двинутся в одну сторону. Такое дружное движение зарядов называют **электрическим током**.

Ток идёт только по **замкнутой цепи** – по кругу без разрывов: от одного вывода батарейки через провода и лампу к другому её выводу. Выключатель на стене ничего не «выключает» внутри лампочки, он просто разрывает цепь. Вода в трубе, которую перекрыли краном, тоже перестаёт течь по всему кругу сразу, а не только возле крана.

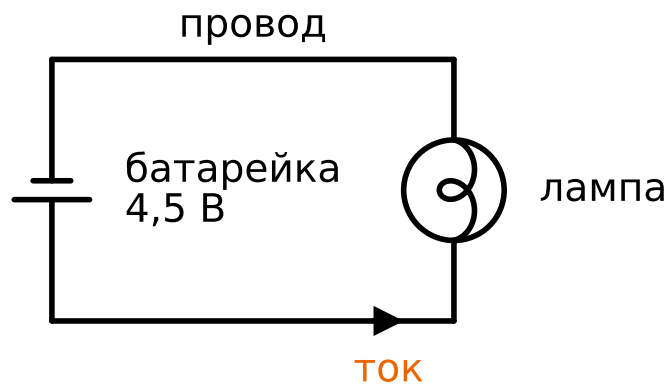


Рисунок 5.2 – Простая цепь: батарейка, провода и лампа. Длинная пластинка батарейки – «плюс», короткая – «минус»

Чтобы ток пошёл, заряды надо толкать. Толкает их батарейка, и сила этого толчка называется **напряжением**. В нашем сравнении с водой напряжение – это напор насоса: чем он сильнее, тем бодрее бежит вода. Напряжение обозначают буквой U и измеряют в **вольтах** (В). Насколько знакомых напряжений:

- пальчиковая батарейка – 1,5 В;
- плата Arduino Uno и USB-зарядка телефона – 5 В;
- домашняя розетка – больше двухсот вольт, это опасно для жизни, и с розеткой мы в кружке не работаем.

Сила тока – сколько заряда проходит по проводу за секунду, в нашем сравнении – сколько литров воды протекает по трубе. Её обозначают буквой I и измеряют в **амперах** (А). Для наших приборов ампер – очень много, поэтому чаще пишут **миллиамперы**: 1 мА = 0,001 А, то есть 1 А = 1000 мА. Светодиоду хватает 10–20 мА, плате Arduino – несколько десятков миллиампер.

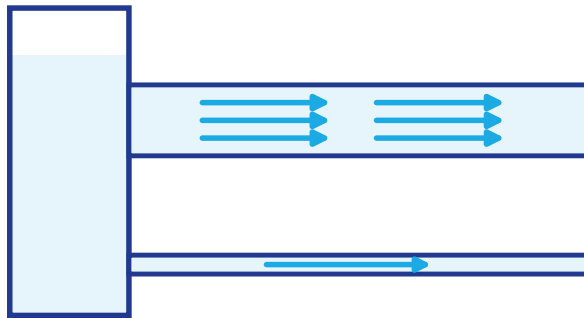
Напряжение всегда меряют **между двумя точками**: между «плюсом» и «минусом» батарейки, между выводом платы и её «землёй». Общий «минус» у Arduino подписан **GND** (от английского ground – земля), его напряжение считают равным нулю. Когда говорят «на выводе A0 3 В», имеют в виду: между A0 и GND 3 В.

Запомни. Ток идёт только по замкнутой цепи. Напряжение U – «напор», в вольтах (В). Сила тока I – «сколько течёт», в амперах (А) и миллиамперах: 1 А = 1000 мА.

5.2. Сопротивление и закон Ома

НА ПАЛЬЦАХ

один и тот же бак



широкая труба: сопротивление
маленькое – ток большой

узкая труба: сопротивление
большое – ток маленький

Рисунок 5.3 – Бак один и тот же, а вода по двум трубам течёт по-разному

Узкая труба мешает воде. Деталь с большим сопротивлением так же мешает току: напор тот же, а течёт меньше.

Всё, через что идёт ток, ему немного мешает. Медный провод почти не мешает, волосок в лампе накаливания мешает сильно и от этого раскаляется. Эту «помеху току» называют **сопротивлением**, обозначают буквой R и измеряют в **омах** (Ом). Большие сопротивления пишут в килоомах и мегаомах: $1 \text{ кОм} = 1000 \text{ Ом}$, $1 \text{ МОм} = 1000 \text{ кОм} = 1\,000\,000 \text{ Ом}$.

Резистор – деталь, которую делают ради её сопротивления. Выглядит как маленький бочонок с цветными полосками и двумя ножками, а на схемах его рисуют прямоугольником. Резистор нужен, чтобы ограничить ток или разделить напряжение, – обе работы мы увидим в этой главе.

Как связаны напор, ширина трубы и поток воды? Чем сильнее напор, тем больше течёт; чем уже труба, тем меньше. С током то же самое, и это правило называется **законом Ома** (по имени немецкого физика Георга Ома). Словами: **сила тока равна напряжению, делённому на сопротивление**. Буквами:

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

где I – сила тока в амперах, U – напряжение в вольтах, R – сопротивление в омах.

Из этой формулы получаются ещё две – для задач, где неизвестно другое:

- **напряжение** – ток, умноженный на сопротивление: $U = I \cdot R$;
- **сопротивление** – напряжение, делённое на ток: $R = U : I$.

Пример 1. Резистор 1 кОм подключили к 5 В . Переводим: $1 \text{ кОм} = 1000 \text{ Ом}$. Ток $I = 5 : 1000 = 0,005 \text{ А} = 5 \text{ мА}$.

Пример 2. Через резистор 220 Ом идёт ток 10 мА . Переводим: $10 \text{ мА} = 0,01 \text{ А}$. Напряжение на резисторе $U = 0,01 \cdot 220 = 2,2 \text{ В}$.

Резисторы подряд. Если поставить резисторы друг за другом, в одну цепочку (такое соединение называют **последовательным**), ток идёт через все по очереди, и их сопротивления

складываются: $R = R_1 + R_2$. Три резистора по 1 кОм подряд – это 3 кОм, как одна длинная узкая труба.

Запомни. Закон Ома: $I = U : R$, $U = I \cdot R$, $R = U : I$. Подставляй вольты, амперы и омы. Последовательно стоящие сопротивления складываются.

Совет. Есть короткий путь без перевода единиц: килоомы, умноженные на миллиамперы, дают вольты. $2 \text{ кОм} \cdot 3 \text{ мА} = 6 \text{ В}$; $5 \text{ В} : 1 \text{ кОм} = 5 \text{ мА}$. Работает, потому что «кило» (тысяча) и «милли» (тысячная) гасят друг друга. Только не смешивай: омы с миллиамперами так не перемножаются.

Для любопытных: резисторы рядом. А если те же три резистора по 1 кОм поставить рядом, «лесенкой» (такое соединение называют **параллельным**), у тока появятся три дороги сразу. Общее сопротивление станет меньше, чем у одного резистора: около 333 Ом, втрое меньше. На вебинаре сферы это проверяли мультиметром в симуляторе. Точную формулу для разных резисторов рядом проходят в 8 классе, а в задачах этой главы такое соединение не встречается.

5.3. Светодиод и резистор к нему

НА ПАЛЬЦАХ

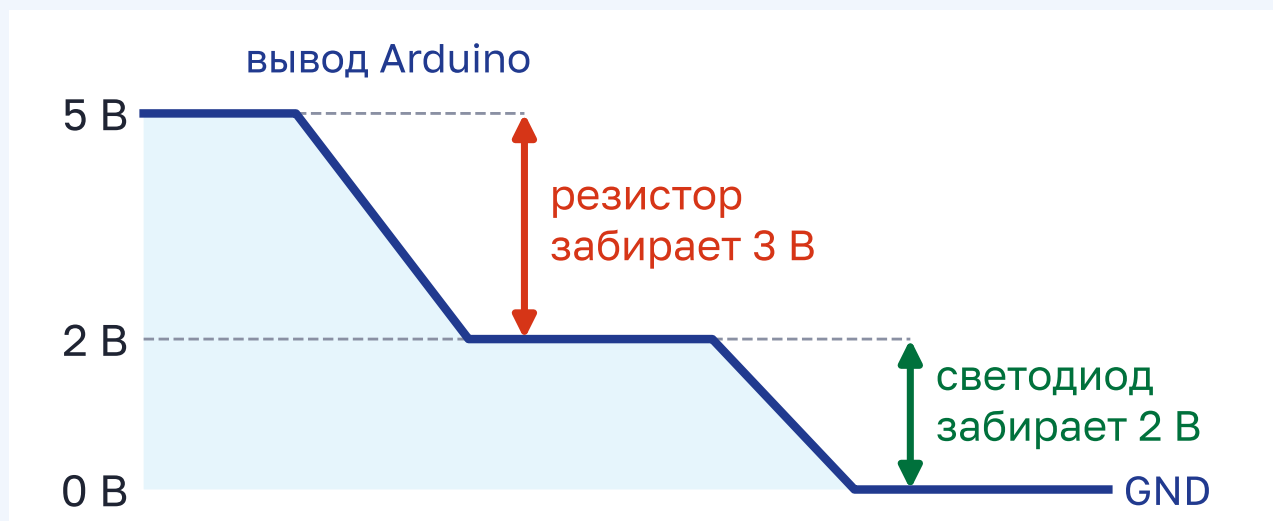


Рисунок 5.4 – Пять вольт от вывода платы по пути к GND делятся между резистором и светодиодом

Светодиод всегда «забирает» себе примерно одно и то же напряжение. Всё, что осталось, должен забрать резистор.

Светодиод – маленькая лампочка, которая светится, когда через неё идёт ток. У него есть «плюс» и «минус»: ток проходит только в одну сторону. Длинная ножка (**анод**) – к плюсу, короткая (**катод**) – к минусу, к GND. Провернёшь – не загорится, но и не сгорит.

У светодиода есть привычка: пока горит, он забирает себе почти одно и то же напряжение – у красного около 2 В, у синего и белого около 3 В (точное число пишут в описании детали или дают в задаче). И у него есть предел тока: для обычного учебного светодиода это около 20 мА. Больше – перегреется и сгорит.

Подключим красный светодиод прямо к выводу Arduino, на котором 5 В. Светодиоду нужно 2 В, а лишние 3 В забрать некому: провод почти не мешает току. Ток станет во много раз больше допустимого. На вебинаре сферы так и сделали в симуляторе: программа предупредила, что через светодиод идёт почти 500 мА при допустимых 20 мА. В симуляторе это только предупреждение, а настоящий светодиод перестанет светить навсегда.

Поэтому последовательно со светодиодом ставят резистор. Он забирает на себя лишнее напряжение и не пускает ток выше нормы. Как подобрать его сопротивление? На резистор приходится то, что осталось от питания: $5 - 2 = 3$ В. Ток через него тот же, что через светодиод. По закону Ома:

$$R = \frac{U_{\text{пит}} - U_{\text{св}}}{I},$$

где $U_{\text{пит}}$ – напряжение питания (у вывода Arduino 5 В), $U_{\text{св}}$ – напряжение светодиода, I – ток, который мы разрешаем, в амперах. Для красного светодиода и 20 мА:

$$R = (5 - 2) : 0,02 = 150 \text{ Ом}.$$

Резисторы на любое сопротивление не выпускают. В магазине и в наборах лежат **номиналы** из стандартного ряда, а в учебных наборах – совсем немного видов: 220 Ом, 1 кОм, 10 кОм. Берём **ближайший больший** номинал: с большим сопротивлением ток станет чуть меньше, светодиод будет гореть чуть тусклее, но останется цел. Для нашего примера это 220 Ом – поэтому на курсе его и советуют. Ток тогда $3 : 220 \approx 0,0136$ А, около 13,6 мА.

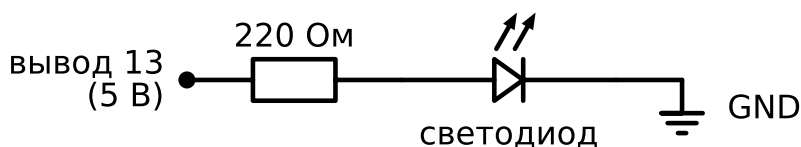


Рисунок 5.5 – Светодиод на выводе 13: резистор 220 Ом, светодиод, GND

Резистор можно поставить и до светодиода, и после – в цепочке ток один и тот же, результат не изменится.

Запомни. Светодиод – только с резистором. $R = (U_{\text{пит}} - U_{\text{св}}) : I$, ток – в амперах. Из набора бери ближайший **больший** номинал.

5.4. Датчик – переводчик на язык вольт

НА ПАЛЬЦАХ

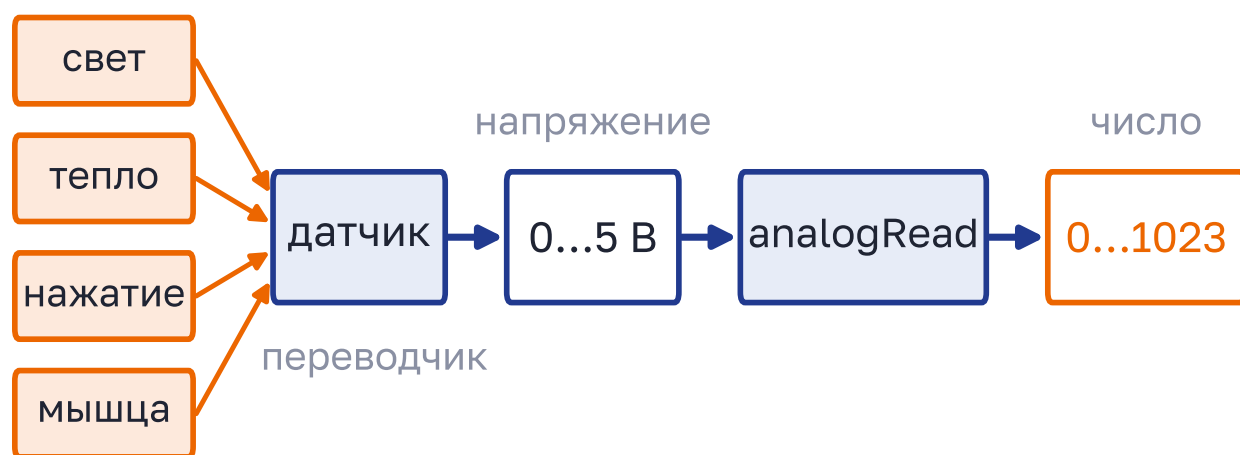


Рисунок 5.6 – Датчик переводит свет, тепло, нажатие или работу мышцы в напряжение, а плата – напряжение в число

Плата не видит света и не чувствует тепла. Она умеет одно – мерить напряжение. Всё остальное ей переводит датчик.

Датчик (его ещё называют сенсором) – деталь, которая превращает какую-то величину из внешнего мира в электрический сигнал. Свет, температура, сила нажатия, работа мышцы – на выходе у датчика напряжение, которое меняется вслед за этой величиной. Дальше плата измеряет напряжение и получает число.

Датчики бывают двух видов. Одни **сами выдают напряжение**. Другие только **меняют своё сопротивление**, и тогда к ним добавляют резистор, чтобы сопротивление превратилось в напряжение (как – в разделе 5.5). Вот те, что встречаются в курсе сферы и в задачах:

- **Фоторезистор** – резистор, который чувствует свет: чем светлее, тем меньше его сопротивление. На вебинаре сферы из него собирали в симуляторе «датчик пульса»: двигаешь ползунок освещения – и на графике бегут «удары».
- **Термистор** – резистор, чьё сопротивление меняется от тепла. Бывают и готовые **датчики температуры**, которые сами выдают напряжение по простому правилу, например «0,5 В при 0 °C и ещё 0,01 В на каждый градус». Правило всегда берут из описания датчика или из условия задачи.
- **Датчик силы нажатия** – плёнка, которая при сильном нажатии пропускает ток лучше: её сопротивление падает.
- **Оптический датчик пульса** – светодиод и приёмник света рядом. Кровь в пальце то прибывает, то убывает, и до приёмника доходит то больше, то меньше света (подробно – в главе 2).
- **Электроды ЭМГ, ЭКГ, ЭЭГ**. Тут датчиком служит само тело: мышцы, сердце и мозг дают напряжение. Но оно крошечное: у сердца около 1 мВ, у мозга – десятки микровольт (глава 1). Поэтому за электродами всегда стоит **усилитель**.

Усилитель – схема, которая увеличивает напряжение во много раз. Во сколько раз, называют **коэффициентом усиления**, его обозначают буквой K . Словами: напряжение на выходе равно напряжению на входе, умноженному на коэффициент усиления: $U_{\text{вых}} = K \cdot U_{\text{вх}}$. Сигнал сердца на коже около 1 мВ; усилитель с $K = 1000$ превратит его в $1 \cdot 1000 = 1000 \text{ мВ} = 1 \text{ В}$, а такое напряжение плата уже хорошо видит.

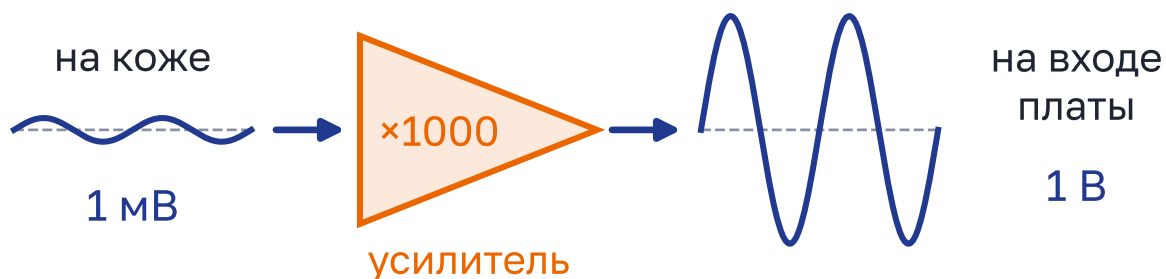


Рисунок 5.7 – Усилитель похож на лупу: форма сигнала та же, только всё в 1000 раз крупнее

Обратная дорога тоже пригодится: если после усилителя сигнал качнулся на 1 В, на коже он качнулся в K раз меньше – на 1 мВ. Чтобы вернуться с платы на кожу, делят на K .

Запомни. Датчик переводит величину (свет, тепло, нажатие, биосигнал) в напряжение. Если датчик меняет сопротивление, к нему добавляют резистор. Биосигналы крошечные – перед платой их усиливают: $U_{\text{вых}} = K \cdot U_{\text{вх}}$.

5.5. Делитель напряжения

НА ПАЛЬЦАХ

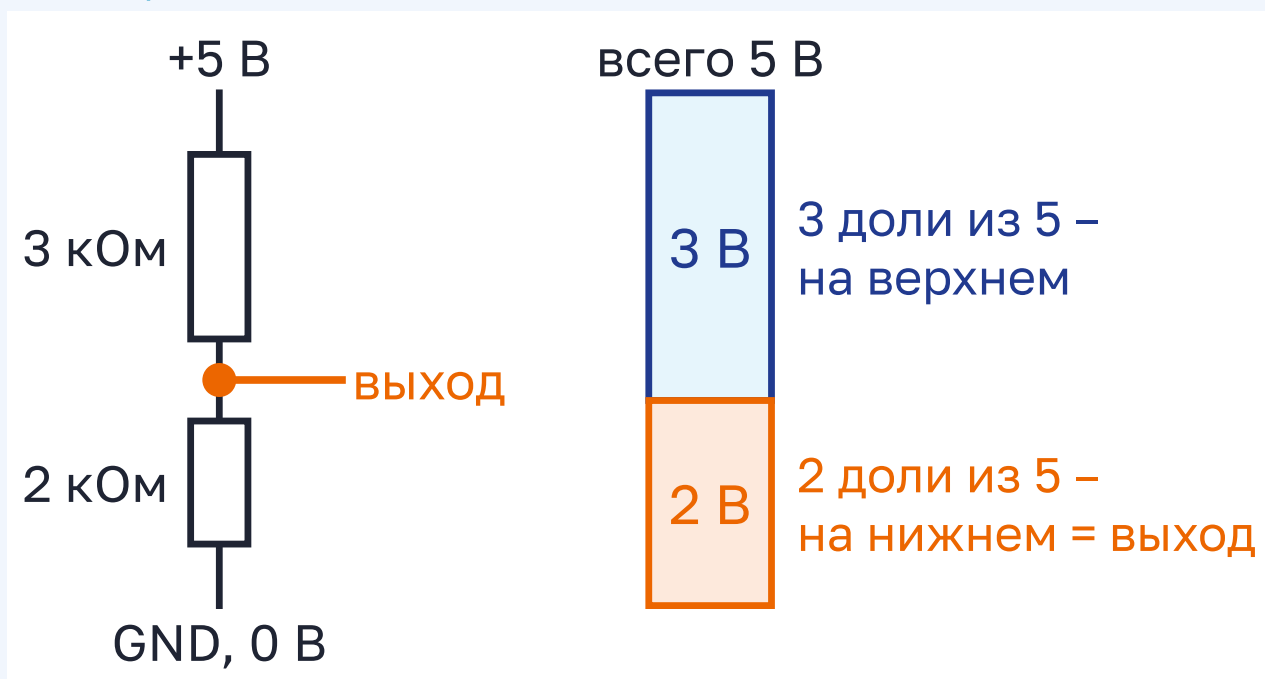


Рисунок 5.8 – Два резистора подряд делят 5 В на части – пропорционально своим сопротивлениям

Всего 5 кОм – это 5 долей. Верхний резистор – 3 доли, ему достаётся 3 В. Нижний – 2 доли, у него 2 В. Точка между ними и есть выход.

Поставим два резистора подряд между 5 В и GND. Напряжение 5 В разделится между ними, как пирог между двумя едоками: кто «больше» (у кого больше сопротивление), тому и кусок больше. Точку между резисторами называют **выходом** делителя. Напряжение на выходе – то, что досталось **нижнему** резистору, который стоит между выходом и GND.

Словами: **выход равен напряжению питания, умноженному на долю нижнего резистора в общем сопротивлении**. Буквами:

$$U_{\text{вых}} = U \cdot \frac{R_2}{R_1 + R_2},$$

где U – питание, R_1 – верхний резистор (от питания до выхода), R_2 – нижний (от выхода до GND).

Пример. 5 В, сверху 3 кОм, снизу 2 кОм: $U_{\text{вых}} = 5 \cdot \frac{2}{3+2} = 5 \cdot 0,4 = 2 \text{ В}$. Если оба резистора одинаковые, на выходе половина: 2,5 В.

Сопротивления в формуле можно брать в килоомах, лишь бы оба в одних единицах: единицы в дроби сокращаются.

Потенциометр – это делитель в одной детали. У него три ножки: крайние подключают к 5 В и GND, средняя соединена с ползунком, который едет по полоске из вещества с сопротивлением. Ползунок делит полоску на две части – верхнюю и нижнюю. Крутишь ручку – меняется доля нижней части, и напряжение на средней ножке плавно ходит от 0 до 5 В. В курсе сферы потенциометром заменяют датчик, которого нет в симуляторе: покрутил ручку – «сигнал» вырос.

Делитель с датчиком. Плата не умеет мерить сопротивление, она мерит напряжение. Поэтому датчик, который меняет сопротивление, ставят в делитель вместо одного из резисторов. Самая частая схема – фоторезистор сверху и постоянный резистор 10 кОм снизу.

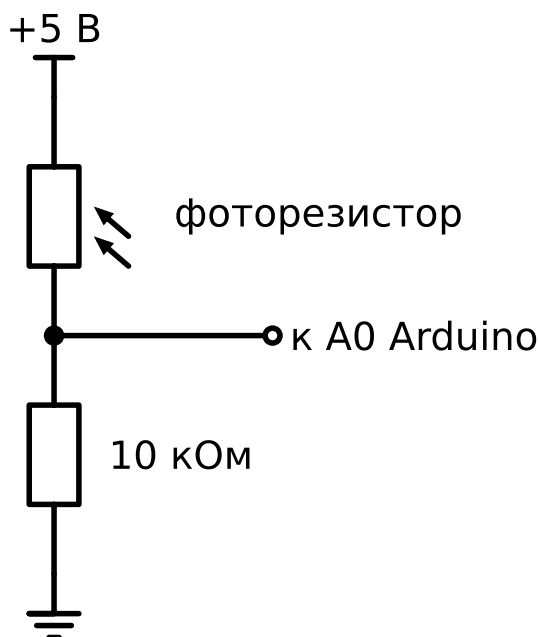


Рисунок 5.9 – Делитель на фоторезисторе: чем светлее, тем меньше сопротивление фоторезистора и тем больше напряжение на A0

Посветили на фоторезистор – его сопротивление упало. На верхний резистор теперь приходится меньшая доля, на нижний – большая, и напряжение на A0 растёт. Затемнили – напряжение падает. Если поменять фоторезистор и 10 кОм местами, всё пойдёт наоборот: свет будет *уменьшать* напряжение. Поэтому в задаче первым делом смотри, какой резистор внизу.

Запомни. $U_{\text{вых}} = U \cdot R_2 : (R_1 + R_2)$, где R_2 – резистор между выходом и GND. Одинаковые резисторы делят напряжение пополам.

Совет. Проверка без формулы: выход всегда между 0 и напряжением питания, и он ближе к тому концу, где резистор *меньше*. Если нижний резистор намного больше верхнего, на выходе почти всё питание.

5.6. Аналоговый вход: из вольт в число и обратно

НА ПАЛЬЦАХ

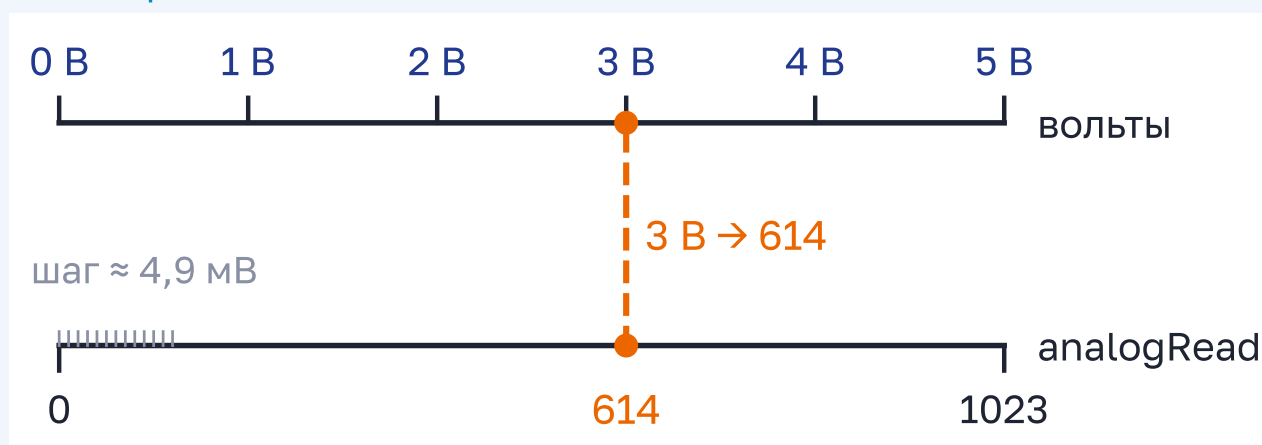


Рисунок 5.10 – Две линейки одной длины: сверху вольты от 0 до 5, снизу числа от 0 до 1023

Плата прикладывает напряжение к верхней линейке и читает число с нижней. 5 В – это 1023, половина от 5 В – половина от 1023.

У Arduino Uno шесть **аналоговых входов**, A0–A5. Команда `analogRead(A0)` измеряет напряжение на входе A0 и возвращает целое число от 0 до 1023. Делает это **АЦП** – аналого-цифровой преобразователь, встроенный в плату (о том, как часто он измеряет, – глава 4). 0 В даёт число 0, 5 В – число 1023. Разных чисел 1024, а шагов между ними 1023: на школьной линейке от 0 до 10 см тоже одиннадцать штрихов, но десять сантиметровых промежутков. Одна ступенька – это

$$5 : 1023 \approx 0,0049 \text{ В} = 4,9 \text{ мВ}.$$

Из числа в вольты. Сколько ступенек, столько раз по 4,9 мВ. Точнее – через пропорцию: число N так относится к 1023, как напряжение U к 5 В:

$$U = N \cdot \frac{5}{1023}.$$

Пример: `analogRead` вернул 818. $U = 818 \cdot 5 : 1023 = 4090 : 1023 \approx 4,0$ В.

Из вольт в число. Обратное действие:

$$N = U \cdot \frac{1023}{5}.$$

Пример: на входе 3 В. $N = 3 \cdot 1023 : 5 = 613,8$, после округления 614.

Настоящий АЦП дробную часть отбрасывает, поэтому плата может показать и 613. В задачах такое место оговаривают («округли до целых»), и ответ нужно дать по условию. Бывает и так, что в условии делят не на 1023, а на 1024, – разница крошечная, но считать надо по условию.

В программе перевод в вольты выглядит так:

```

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;    // то же напряжение в вольтах
  Serial.println(u);           // печатаем в монитор порта
  delay(200);                  // ждём 0,2 с
}

```

Строка `int n = analogRead(A0);` заводит **переменную** – ячейку памяти с именем `n` – и кладёт в неё число с входа A0. Слово `int` значит «целое число». В следующей строке `float` – число с дробной частью: вольты бывают дробными. `Serial.println` отправляет число в **монитор порта** – окно в Arduino IDE и в симуляторе, где видно, что прислала плата; дробное число там печатается с двумя знаками после точки, например `3.00`.

Обрати внимание на `5.0`, а не `5`. Если написать `n * 5 / 1023`, все числа в строке целые, и плата разделит нацело: для 818 получится $4090 : 1023 = 3$ с отброшенным остатком, и в монитор уйдёт `3.00` вместо `4.00`. Точка в `5.0` говорит плате: считай с дробными.

Запомни. `analogRead` : 0 В → 0, 5 В → 1023, ступенька около 4,9 мВ. Из числа в вольты: $U = N \cdot 5 : 1023$. Из вольт в число: $N = U \cdot 1023 : 5$.

Совет. Прикидка в уме: 1 В – это примерно 205 ступенек. 2 В – около 409, 4 В – около 818. Получил для 1,5 В число 900 – где-то ошибся, должно быть около 307.

5.7. Батарейки и время работы прибора

НА ПАЛЬЦАХ



время = запас : расход

$$2000 : 50 = 40 \text{ ч}$$

Рисунок 5.11 – Аккумулятор – бак с запасом, прибор – открытый кран

Чем больше бак и чем тоньше струя, тем дольше течёт. Время работы – запас, делённый на расход.

Пульсометр на прогулке, браслет на руке, ЭЭГ-ободок – всё это работает от аккумулятора или батарейки. У старших ребят на финале НТО по нейротехнологиям плату с ЭЭГ по правилам безопасности питали от аккумулятора, а не от розетки. Значит, надо уметь прикинуть, на сколько его хватит.

На аккумуляторе пишут его **ёмкость** – сколько заряда в нём запасено. Её измеряют в **миллиампер-часах** (мА·ч). 2000 мА·ч значит: аккумулятор может отдавать ток 2000 мА в течение 1 часа, или 1000 мА – 2 часа, или 100 мА – 20 часов. Чем меньше ток, тем дольше.

Словами: **время работы равно ёмкости, делённой на ток, который берёт прибор**. Буквами:

$$t = \frac{C}{I},$$

где C – ёмкость в мА·ч, I – ток в мА, t – время в часах. Миллиамперы в миллиампер-часах сокращаются, остаются часы.

Пример. Прибор берёт 50 мА, аккумулятор 2000 мА·ч: $t = 2000 : 50 = 40$ ч.

Если прибор берёт ток в амперах, сначала переведи в миллиамперы ($0,2 \text{ А} = 200 \text{ мА}$), иначе ответ уедет в тысячу раз.

Если ток меняется. Многие приборы то работают, то «спят». Браслет включает датчик пульса на минуту и отключает на несколько минут. Тогда считают **средний ток** – как среднее значение в главе 7, только каждое значение тока берут столько раз, сколько минут он длится. Об этом задача 5.7.

На деле аккумулятор отдаёт немного меньше, чем написано на корпусе, особенно на холоде: зимний телефон на морозе садится быстрее. В задачах считаем по надписи, если условие не говорит другого.

Запомни. $t = C : I$: ёмкость в мА·ч делим на ток в мА – получаем часы. Если ток меняется, сначала найди средний ток.

Совет. Смотри, что спрашивают: «сколько проработает» – можно с дробью, а «на сколько полных дней хватит» – округляй вниз: неполный день прибор до вечера не дотянет. А если нужно подобрать аккумулятор под задачу – округляй вверх, до ближайшего подходящего.

Для любопытных: ватты. На зарядке телефона и на лампочке пишут **мощность** – сколько энергии прибор тратит за секунду. Её находят умножением напряжения на силу тока: $P = U \cdot I$, в ваттах (Вт), если напряжение в вольтах, а ток в амперах. Светодиод при 5 В и 0,02 А: $P = 5 \cdot 0,02 = 0,1$ Вт. Подробно мощность проходят в 8 классе; в задачах этой главы она не нужна.

5.8. Разбираем задачи

Задача 5.1 Что показал мультиметр

разминка · закон Ома, миллиамперы

На кружке Вика собрала цепь со светодиодом и резистором 470 Ом. **Мультиметр** – прибор, который меряет напряжение, ток и сопротивление, – показал, что на резисторе 2,35 В. Какой ток идёт через резистор? Ответ дай в миллиамперах.

Дано: $R = 470 \text{ Ом}$; $U = 2,35 \text{ В}$.

Найти: I – ? (в мА)

Решение

Шаг 1. Какая формула. Известны напряжение и сопротивление, нужен ток. Закон Ома:
 $I = U : R$.

Шаг 2. Подставляем. Напряжение в вольтах, сопротивление в омах – переводить ничего не надо: $I = 2,35 : 470 = 0,005 \text{ А}$.

Шаг 3. Переводим в миллиамперы. В ампере 1000 мА: $0,005 \cdot 1000 = 5 \text{ мА}$.

5 мА – спокойный ток для учебного светодиода, до предела в 20 мА далеко. Светить он будет, но не во всю силу.

 **Типичная ошибка.** Сдать ответ 0,005: задача просит миллиамперы, а это амперы. Вторая ошибка – перемножить $2,35 \cdot 470 \approx 1105$: так считают напряжение, а не ток.

Ответ: 5 мА.

Своя задача по программе сферы, модуль 1

Реши сам

5.1. К батарейке 4,5 В подключили резистор 1,5 кОм. Какой ток пойдёт через него? Ответ в миллиамперах.

5.2. В цепи от 5 В последовательно стоят два резистора. Один из них – 1 кОм, сопротивление второго неизвестно. Мультиметр показал ток 2 мА. Каково сопротивление второго резистора? Ответ в килоомах.

Задача 5.2 Синий светодиод для индикатора пульса

резистор для светодиода, номинал, закон Ома

Команда делает индикатор: на каждый удар сердца мигает синий светодиод. Он подключён к выводу Arduino, где 5 В. Синему светодиоду нужно 3,2 В, и ток через него не должен превышать 15 мА. В наборе есть резисторы 100 Ом, 220 Ом, 470 Ом и 1 кОм.

а) Какое наименьшее сопротивление резистора допустимо? б) Какой резистор из набора взять? в) Какой ток пойдёт через светодиод с этим резистором? Ответ в миллиамперах, округли до десятых.

Дано: $U_{\text{пит}} = 5 \text{ В}$; $U_{\text{св}} = 3,2 \text{ В}$; $I_{\text{max}} = 15 \text{ мА}$; набор: 100, 220, 470, 1000 Ом.

Найти: R_{min} – ?; номинал – ?; I – ?

Решение

Шаг 1. Сколько напряжения достаётся резистору. Светодиод забирает 3,2 В, резистору остаётся $5 - 3,2 = 1,8 \text{ В}$.

Шаг 2. Ток – в амперы. $15 : 1000 = 0,015 \text{ А}$.

Шаг 3. Наименьшее сопротивление. $R = 1,8 : 0,015 = 120 \text{ Ом}$. Если резистор меньше 120 Ом, ток станет больше 15 мА.

Шаг 4. Выбираем из набора. Нужен номинал не меньше 120 Ом. 100 Ом не годится – он меньше. Ближайший больший – 220 Ом.

Шаг 5. Ток с этим резистором. На резисторе по-прежнему 1,8 В: $I = 1,8 : 220 \approx 0,00818 \text{ А}$. В миллиамперах: $0,00818 \cdot 1000 \approx 8,2 \text{ мА}$.

Ток получился меньше разрешённых 15 мА, запас почти вдвое – светодиод будет гореть чуть тусклее, зато проживёт долго.

 **Типичная ошибка.** Делить все 5 В на ток: $5 : 0,015 \approx 333 \text{ Ом}$. Про напряжение светодиода забыли, и резистор вышел больше нужного. Вторая ошибка – взять из набора ближайший номинал в любую сторону: 100 Ом ближе к 120, чем 220, но с ним ток будет $1,8 : 100 = 0,018 \text{ А} = 18 \text{ мА}$ – больше разрешённого.

Ответ: а) 120 Ом; б) 220 Ом; в) 8,2 мА.

Своя задача по программе сферы, модуль 1

Реши сам

5.3. Жёлтый светодиод на 2,1 В подключают к 5 В, ток должен быть не больше 10 мА. Каково наименьшее допустимое сопротивление резистора? Какой резистор взять из набора 220 Ом, 330 Ом, 470 Ом?

5.4. Миша подключил красный светодиод (2 В, предел 20 мА) к выводу 5 В через резистор 100 Ом. а) Какой ток пойдёт через светодиод, в миллиамперах? б) Выдержит ли светодиод? Ответь «да» или «нет». в) Какой наименьший резистор из набора 100 Ом, 220 Ом, 470 Ом подойдёт?

Задача 5.3 Ночник у кровати

делитель напряжения, фоторезистор, analogRead

В больничной палате ночник должен сам загораться, когда темно. Датчик собран по схеме с рисунка в разделе 5.5: фоторезистор сверху (от 5 В до А0), резистор 10 кОм снизу (от А0 до GND). При дневном свете сопротивление фоторезистора 2,5 кОм, в сумерках – 15 кОм.

а) Какое число вернёт `analogRead(A0)` днём? б) А в сумерках? Округли до целых. в) Программа включает ночник, если число меньше 600. Загорится ли ночник в сумерках? Ответь «да» или «нет».

Дано: $U = 5$ В; R_1 (фоторезистор) = 2,5 кОм днём и 15 кОм в сумерках; $R_2 = 10$ кОм; порог 600.

Найти: $N_{\text{день}}$ – ?; $N_{\text{сумерки}}$ – ?; загорится ли ночник.

Решение

Шаг 1. Какой резистор нижний. Выход – вход А0. Между А0 и GND стоит 10 кОм, это R_2 . Фоторезистор – верхний, R_1 .

Шаг 2. Напряжение днём. Всего $2,5 + 10 = 12,5$ кОм. Доля нижнего: $10 : 12,5 = 0,8$.
 $U_{\text{вых}} = 5 \cdot 0,8 = 4$ В.

Шаг 3. Число днём. $N = 4 \cdot 1023 : 5 = 818,4$, округляем до 818.

Шаг 4. Напряжение в сумерках. Всего $15 + 10 = 25$ кОм. Доля нижнего: $10 : 25 = 0,4$.
 $U_{\text{вых}} = 5 \cdot 0,4 = 2$ В.

Шаг 5. Число в сумерках. $N = 2 \cdot 1023 : 5 = 409,2$, округляем до 409.

Шаг 6. Сравниваем с порогом. $409 < 600$ – ночник загорится. Днём $818 > 600$, и он не горит.

Проверим направление: стало темнее – сопротивление фоторезистора выросло – число упало с 818 до 409. При фоторезисторе сверху свет число увеличивает, темнота уменьшает – с разделом 5.5 не спорит.

 **Типичная ошибка.** Поставить в числитель фоторезистор: $5 \cdot 2,5 : 12,5 = 1$ В днём. Это напряжение на верхнем резисторе, а плата меряет между А0 и GND – на нижнем. Выручает проверка: 1 В днём и 3 В в сумерках значило бы, что свет уменьшает число, а при такой схеме он его увеличивает.

Ответ: а) 818; б) 409; в) да.

Своя задача по программе сферы, модуль 2

Реши сам

5.5. Потенциометр 10 кОм подключён крайними ножками к 5 В и GND, средняя ножка – к А0. Ручку повернули так, что между средней ножкой и GND 7 кОм, а между средней ножкой и 5 В – 3 кОм. Какое напряжение на А0 и какое число вернёт `analogRead`? Число округли до целых.

5.6. Датчик силы нажатия стоит в делителе сверху (от 5 В до A0), а снизу (от A0 до GND) нужно поставить резистор. Когда палец жмёт на датчик, его сопротивление 6 кОм. Какой резистор из набора 1 кОм, 2 кОм, 4 кОм, 10 кОм выбрать, чтобы при нажатии на A0 было 2 В?

Задача 5.4 Градусник на Arduino

analogRead в вольты, датчик температуры, правило датчика

Датчик температуры выдаёт напряжение по правилу: при 0 °C – 0,5 В, и на каждый градус выше нуля прибавляется 0,01 В. Ученица держит датчик в ладони, `analogRead` показывает 177. Какую температуру измерил датчик? Ответ округли до десятых градуса.

Дано: $N = 177$; правило датчика: 0,5 В при 0 °C, +0,01 В на 1 °C.

Найти: T – ? (°C)

Решение

Шаг 1. Число – в вольты. $U = 177 \cdot 5 : 1023 = 885 : 1023 \approx 0,8651$ В. Округлять здесь рано: если взять 0,87 В, температура уедет на полградуса.

Шаг 2. Сколько вольт прибавилось к нулевому уровню. $0,8651 - 0,5 = 0,3651$ В.

Шаг 3. Сколько это градусов. Каждый градус – 0,01 В, поэтому делим: $0,3651 : 0,01 = 36,51$ °C.

Шаг 4. Округляем. До десятых: 36,5 °C.

Температура руки чуть ниже 37 °C – как у градусника, который недодержали под мышкой.

 **Типичная ошибка.** Забыть про 0,5 В при нуле и сразу поделить $0,8651 : 0,01 = 86,5$ °C. Такая «температура» в ладони – повод остановиться и перечитать правило датчика. Вторая ошибка – округлить напряжение до десятых (0,9 В): тогда выйдет 40 °C, мимо на три с половиной градуса.

Ответ: 36,5 °C.

Своя задача по программе сферы, модуль 2

Реши сам

5.7. `analogRead` вернул 921. Какое напряжение на входе? Ответ в вольтах, округли до сотых.

5.8. Датчик температуры из задачи 5.4 опустили в тёплую воду, 38,0 °C. Какое число покажет `analogRead`? Округли до целых.

Задача 5.5 Пульсометр в летнем лагере

ёмкость аккумулятора, время работы, округление вниз

Отряд едет в лагерь на Обском море и берёт самодельный пульсометр: плата, датчик и экранчик вместе берут 60 мА. Питают его от пауэрбанка ёмкостью 3000 мА·ч. Каждый день пульсометр включают на 4 часа.

а) Сколько часов пульсометр проработает без подзарядки? б) На сколько полных дней хватит пауэрбанка?

Дано: $I = 60$ мА; $C = 3000$ мА·ч; 4 ч в день.

Найти: t – ? (ч); число полных дней – ?

Решение

Шаг 1. Единицы. Ёмкость в мА·ч, ток в мА – всё согласовано, ответ получится в часах.

Шаг 2. Время работы. $t = C : I = 3000 : 60 = 50$ ч.

Шаг 3. Сколько дней по 4 часа. $50 : 4 = 12,5$.

Шаг 4. Полных дней. На 12 дней хватает целиком, а на 13-й останется только половина дневной работы – 2 часа из 4. Полных дней – 12.

Двенадцать дней – почти две трети лагерной смены в 21 день. Так что зарядку для пауэрбанка всё-таки положи в рюкзак.

 **Типичная ошибка.** Округлить 12,5 по правилам математики до 13. Но на 13-й день прибор выключится посередине занятия – для вопроса «на сколько полных дней хватит» округляем только вниз.

Ответ: а) 50 ч; б) 12 дней.

Своя задача по программе сферы, модуль 6

Реши сам

5.9. Датчик CO₂ в классе берёт 80 мА и питается от аккумулятора 2400 мА·ч. Сколько часов он проработает?

5.10. Прибор для контроля дыхания берёт 45 мА. Он должен работать 3 дня по 8 часов без подзарядки. а) Какая ёмкость нужна, в мА·ч? б) Какой аккумулятор выбрать из 800, 1000, 1200 и 2000 мА·ч, чтобы хватило и чтобы он был самым лёгким (наименьшим)?

Задача 5.6* ЭЭГ по одному байту

усилитель, АЦП, байт 0–255, микровольты

Плата с ЭЭГ-модулем, чтобы быстрее передавать данные на компьютер, отправляет каждое измерение одним **байтом** – целым числом от 0 до 255, а не числом `analogRead` от 0 до 1023. Шкала байта такая: байт 0 – это 0 В на входе платы, байт 255 – это 5 В, между ними всё поровну. Перед платой стоит усилитель, который увеличивает сигнал с кожи в 2000 раз.

а) Компьютер принял байт 102. Какое напряжение было на входе платы? Ответ в вольтах, до десятых. б) На сколько милливольт отличаются соседние значения байта? Округли до десятых. в) Какое самое маленькое изменение напряжения на коже программа на компьютере ещё может заметить? Ответ в микровольтах, округли до десятых.

Дано: байт B от 0 до 255 $\leftrightarrow$ от 0 до 5 В; $B = 102$; $K = 2000$.

Найти: U – ? (В); шаг байта – ? (мВ); шаг на коже – ? (мкВ)

Решение

Шаг 1. Байт – в вольты. Та же пропорция, что для `analogRead`, только вместо 1023 стоит 255:
 $U = 102 \cdot 5 : 255 = 510 : 255 = 2,0$ В.

Откуда вообще 255? Проще всего получить байт из числа `analogRead`, поделив его на 4 и отбросив остаток: $1023 : 4$ – это 255 с хвостиком. Каждые четыре соседние ступеньки АЦП сливаются в одну ступеньку байта, линейка стала вчетверо грубее. Считать по шкале «0–255 $\leftrightarrow$ 0–5 В» можно смело: с точным делением на 4 она расходится меньше чем на одну ступеньку байта, и на ответы до десятых это не влияет.

Шаг 2. Одна ступенька байта. Весь диапазон 5 В поделён на 255 ступенек: $5 : 255 \approx 0,0196$ В. Переводим в милливольты: $0,0196 \cdot 1000 \approx 19,6$ мВ.

Шаг 3. Что это значит на коже. Усилитель увеличил сигнал в 2000 раз. Значит, изменению на входе платы в 19,6 мВ на коже соответствует изменение в 2000 раз меньше:
 $19,6 : 2000 = 0,0098$ мВ.

Шаг 4. Милливольты – в микровольты. В милливольте 1000 микровольт: $0,0098 \cdot 1000 = 9,8$ мкВ.

Сравним. Если бы данные шли полным числом `analogRead`, ступенька была бы 4,9 мВ, на коже – $4,9 : 2000 \approx 2,4$ мкВ, вчетверо мельче. А ЭЭГ на коже – десятки микровольт (глава 1). Волна в 50 мкВ при байтовой передаче займёт всего около пяти ступенек: форму её не разглядеть, но заметить волну программа успеет.

 **Типичная ошибка.** Поделить на 1023 по привычке: $102 \cdot 5 : 1023 \approx 0,5$ В. Число пришло байтом, а у байта своя линейка – до 255. Другая ошибка – в пункте в) умножить на 2000 вместо деления: усилитель увеличивает сигнал, поэтому на коже всё в 2000 раз меньше, чем на плате.

Ответ: а) 2,0 В; б) 19,6 мВ; в) 9,8 мкВ.

По мотивам задачи заключительного этапа НТО «Нейротехнологии и когнитивные науки» 2025/26, инженерный тур, «Дата рождения и P300»

Реши сам

5.11. Та же плата, что в задаче 5.6. а) Компьютер принял байт 204. Какое напряжение было на входе платы? Ответ в вольтах, до десятых. б) Какой байт придёт, если на входе платы 2,2 В? Округли до целых.

5.12.* Сигнал ЭМГ на коже предплечья качается с размахом 1,5 мВ. Усилитель увеличивает его в 500 раз. а) Сколько ступенек `analogRead` займёт размах сигнала на входе платы? б) А сколько ступенек байта, если слать данные, как в задаче 5.6? Оба ответа округли до целых.

Задача 5.7* Браслет, который спит

средний ток, ёмкость, время работы

Фитнес-браслет раз в 10 минут включает датчик пульса на 1 минуту и в это время берёт ток 25 мА. Остальные 9 минут он «спит» и берёт 1 мА. Аккумулятор браслета – 170 мА·ч.

а) Какой средний ток берёт браслет? б) Сколько часов он проработает? в) Сколько часов он проработал бы, если бы мерил пульс без перерыва?

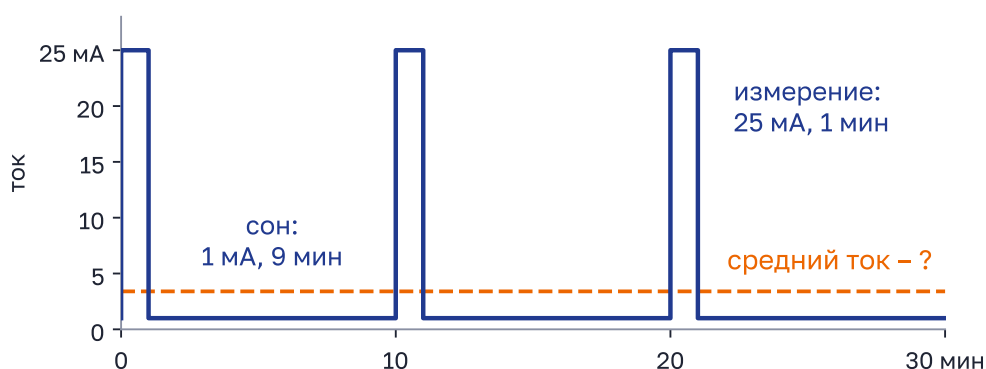


Рисунок 5.12 – Ток браслета за полчаса: каждые 10 минут – одна минута измерения

Дано: 1 мин по 25 мА и 9 мин по 1 мА в каждом 10-минутном круге; $C = 170$ мА·ч.

Найти: $I_{\text{ср}}$ – ?; t – ?; $t_{\text{без сна}}$ – ?

Решение

Шаг 1. Что повторяется. Режим браслета – круг из 10 минут: 1 минута измерения и 9 минут сна. Хватит посчитать один круг: дальше всё то же самое.

Шаг 2. Средний ток за круг. Каждый ток умножаем на то, сколько минут он длится, и складываем: $25 \cdot 1 + 1 \cdot 9 = 25 + 9 = 34$. Делим на длину круга: $34 : 10 = 3,4$ мА. Такой постоянный ток за 10 минут истратил бы столько же, сколько настоящий браслет со своими скачками.

Шаг 3. Время работы. $t = 170 : 3,4 = 50$ ч.

Шаг 4. Без перерыва. Тогда ток всё время 25 мА: $t = 170 : 25 = 6,8$ ч.

Сон растягивает работу больше чем в семь раз: 50 часов против 6,8. Отсюда и правило для носимых приборов – включать датчик только тогда, когда он нужен.

⚠ Типичная ошибка. Сложить два тока и поделить пополам: $(25 + 1) : 2 = 13$ мА. Так было бы, если бы браслет мерил и спал поровну, по 5 минут. А он спит в 9 раз дольше, и средний ток куда ближе к 1 мА, чем к 25.

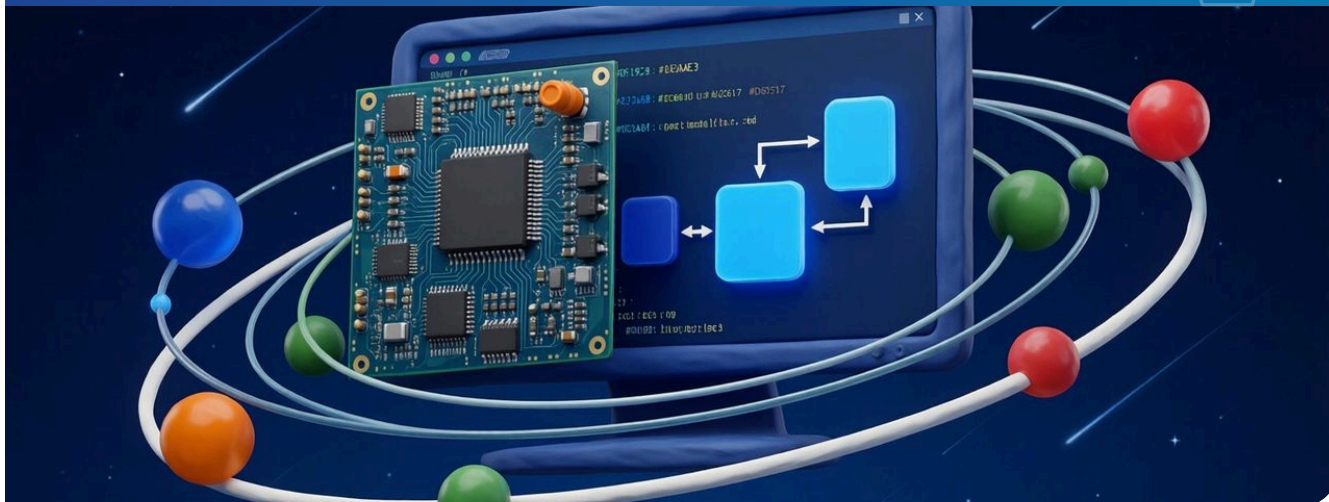
Ответ: а) 3,4 мА; б) 50 ч; в) 6,8 ч.

Реши сам

5.13. Прибор для контроля CO_2 в классе каждые 5 минут одну минуту греет датчик и берёт 150 мА, а остальные 4 минуты берёт 10 мА. а) Каков средний ток? б) Сколько полных часов проработает прибор от пауэрбанка 5000 мА·ч?

5.14.* Браслет должен проработать 5 суток на аккумуляторе 240 мА·ч. Во сне он берёт 1 мА, при измерении пульса – 21 мА. Сколько минут в каждом часе, самое большее, браслет может мерить пульс, чтобы аккумулятора хватило на 5 суток?

Программирование: Arduino и машины состояний



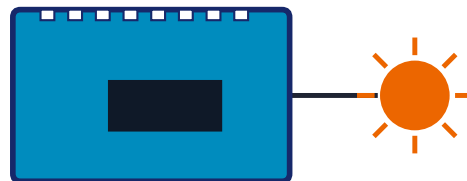
Датчик пульса без программы – просто лампочка и фотоприёмник. Решать, когда включить красный светодиод, сколько раз засчитать сжатие кулака и когда поднять тревогу, будет программа. Здесь ты научишься читать такие программы на C++ для платы Arduino, собирать их из блоков и рисовать схемой из состояний и стрелок – на языке ПРИМС, который есть в курсе сферы. В задачах этой книги программа уже написана: спрашивается, что она выведет, где в ней ошибка, в каком порядке поставить перемешанные строки или в каком состоянии окажется устройство. Для всего этого нужен один навык – следить за программой пальцем, строка за строкой.

6.1. Алгоритм и исполнитель

НА ПАЛЬЦАХ

1. включи светодиод
2. подожди 1 секунду
3. выключи светодиод

алгоритм –
шаги по порядку



исполнитель выполняет
их по одному

Рисунок 6.1 – Алгоритм – список шагов, исполнитель – тот, кто их выполняет

Плата Arduino не думает. Она берёт первый шаг, выполняет, берёт следующий – и так до конца списка.

Алгоритм – точное описание того, что и в каком порядке делать, чтобы получить результат. Инструкция к электронному градуснику – алгоритм: «нажми кнопку, поставь под мышку, дождись писка, посмотри на экран». Дорога от дома до школы, которую ты объясняешь другу, – тоже алгоритм.

Исполнитель – тот, кто выполняет алгоритм: человек, робот, плата Arduino, персонаж на экране. У каждого исполнителя есть **система команд** – список того, что он умеет. Друг поймёт слова «поверни у аптеки налево». Плата Arduino их не поймёт никогда: она умеет подать напряжение на вывод, прочитать датчик, подождать. Если в алгоритме есть команда не из списка, исполнитель встанет.

Главное свойство компьютерного исполнителя: он делает **то, что написано, и только это** – по одной команде и по порядку. Ничего не додумывает и не пропускает. Напишешь «выключи светодиод» раньше, чем «включи», – он выключит уже выключенный, потом включит и оставит гореть. Поэтому почти все ошибки в программах – это не «компьютер сломался», а «я написал не то, что хотел».

В задачах на отборе часто встречается учебный исполнитель с двумя-тремя командами. Например, исполнитель **Яркость** хранит одно число – яркость светодиода – и умеет только это:

- команда **1** – «прибавь 20»;
- команда **2** – «умножь на 2».

Программа для него – строка из номеров команд, её читают слева направо. Было 5, программа **121**: прибавляем 20 – получается 25, умножаем на 2 – 50, прибавляем 20 – 70.

Бывает и обратный вопрос: какой программой получить из одного числа другое? Тогда удобно идти **с конца**. Если в конце стоит 70, последней командой могла быть «прибавь 20» (тогда до неё было 50) или «умножь на 2» (было 35). Дальше тот же вопрос задаём для 50 и для 35, пока не дойдём до начального числа.

Запомни. Алгоритм – шаги по порядку, исполнитель – тот, кто их выполняет и умеет только свои команды. Программу для учебного исполнителя проверяй прогоном: выпиши число после каждой команды. Ищешь программу – иди от конца к началу.

6.2. Переменные: коробочки с именами

НА ПАЛЬЦАХ

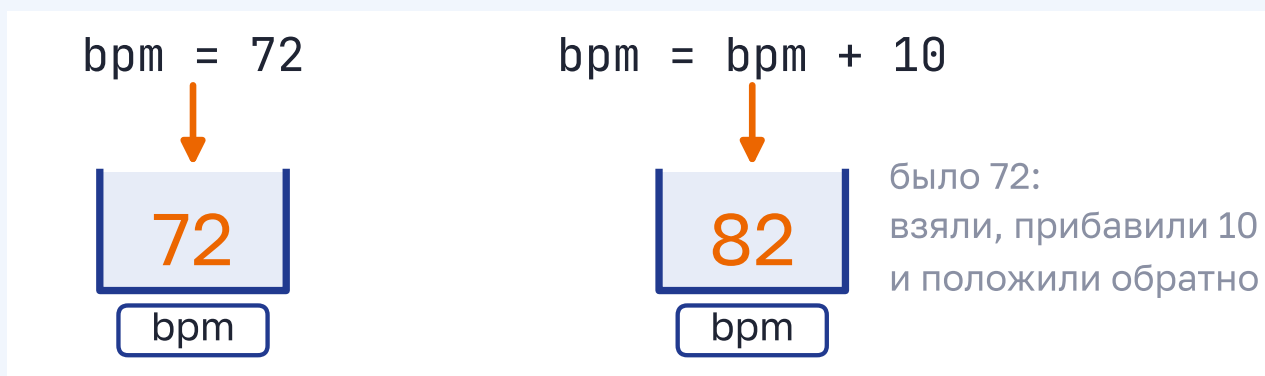


Рисунок 6.2 – Переменная – коробочка с наклейкой. Знак = кладёт в неё число

`bpm = bpm + 10` – не уравнение. Это команда: достань число, прибавь 10 и положи обратно.

Программе нужно где-то держать числа: последнее показание датчика, сколько насчитано ударов, горит ли сейчас тревога. Для этого есть **переменные**. Переменная – коробочка в памяти платы: у неё есть имя, и в ней лежит одно значение. Положишь новое – старое пропадёт.

В C++ – языке, на котором пишут программы для Arduino, – коробочку сначала **объявляют**: называют её и говорят, что в ней будет лежать. Что в ней лежит, называется **типом**:

- `int` – целое число: показание датчика, число ударов, номер вывода. `int bpm = 72;`
- `float` – дробное число: напряжение, температура. Дробную часть отделяют **точкой**: `float volt = 3.3;`
- `bool` – «да» или «нет», по-программистски `true` или `false`: `bool alarm = false;`

Знак `=` в программе значит **«положи в коробочку»**, а вовсе не «равно». Строчку `bpm = bpm + 10;` компьютер читает справа налево: взять то, что лежит в `bpm`, прибавить 10, результат положить обратно в `bpm`. Было 72 – стало 82. Каждая команда в C++ заканчивается точкой с запятой. Текст после `//` – **комментарий**: плата его не читает, он для людей.

Самая коварная вещь в арифметике C++ – деление целых чисел. Если оба числа целые, результат тоже целый, а дробная часть **отбрасывается**. Не округляется, а просто исчезает: `82 / 4` даёт 20, хотя на калькуляторе 20,5. `9 / 10` даёт 0. Остаток от деления даёт знак `%`: `82 % 4` равно 2, потому что $82 = 4 \cdot 20 + 2$.

```

void setup() {
  Serial.begin(9600);
  int beats = 18;           // ударов за 15 секунд
  int bpm = beats * 4;      // ударов за минуту
  Serial.println(bpm);      // 72
  bpm = bpm + 10;          // после десяти приседаний
  Serial.println(bpm);      // 82
  Serial.println(bpm / 4);  // 20 – дробная часть отброшена
  Serial.println(bpm % 4);  // 2 – остаток от деления
  bool fast = bpm > 100;    // 82 > 100? нет – «ложь»
  Serial.println(fast);     // 0
}

void loop() {
}

```

Команда `Serial.println(x)` отправляет значение `x` на компьютер и переводит строку, подробно о ней – в разделе 6.5. Обрати внимание на последнюю строку: «ложь» печатается как 0, а «истина» – как 1.

Совет. Коробочка `int` на Arduino Uno маленькая: в неё помещаются числа только от -32 768 до 32 767. Для больших чисел, например для времени в миллисекундах, берут тип `unsigned long` – он встретится в разделе 6.7.

Запомни. `=` кладёт в переменную, старое значение пропадает. Целое делим на целое – получаем целое, дробь отбрасывается: `82 / 4` даёт 20. `%` – остаток: `82 % 4` даёт 2.

6.3. Условие: развилка в программе

НА ПАЛЬЦАХ

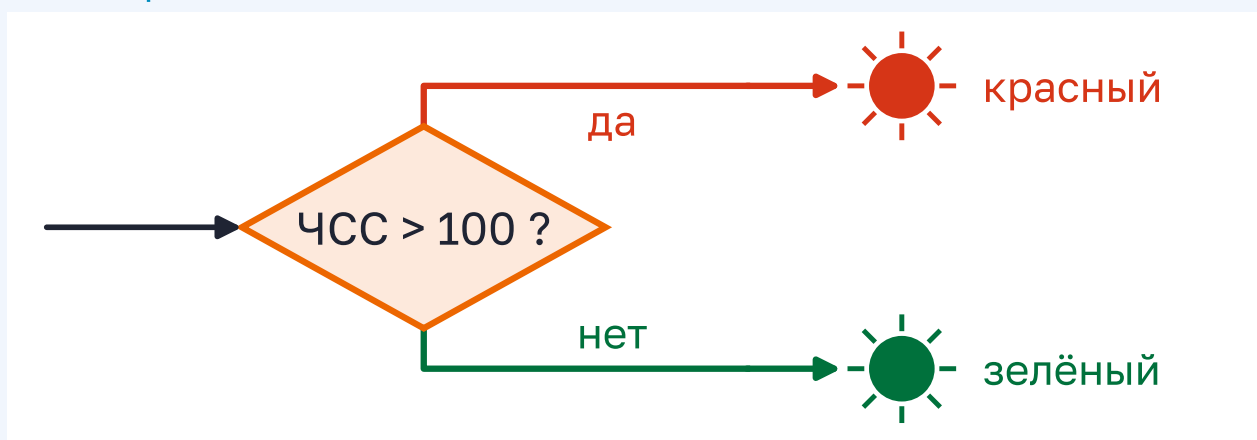


Рисунок 6.3 – Условие – развилка: по одной дороге, если ответ «да», по другой – если «нет»

Программа задаёт вопрос, на который можно ответить только «да» или «нет», и в зависимости от ответа выбирает дорогу.

Прибор, который всегда делает одно и то же, мало на что годится. Пульсометр должен по-разному реагировать на 70 и на 130 ударов в минуту. Для этого в программе есть **условие** – команда `if` («если»):

```
if (bpm > 100) {  
    digitalWrite(LED, HIGH);    // да - включить красный  
} else {  
    digitalWrite(LED, LOW);     // нет - включить зелёный  
}
```

В круглых скобках после `if` – вопрос. Если ответ «да», выполняются команды в первых фигурных скобках, если «нет» – в скобках после `else` («иначе»). Часть с `else` можно не писать: тогда при ответе «нет» программа просто идёт дальше.

Знаки сравнения: `>` больше, `<` меньше, `>=` больше или равно, `<=` меньше или равно, `=` равно, `!=` не равно. Равенство проверяют **двойным** знаком `=`. Одинарный `=` кладёт число в переменную, и `if (level = 2)` – это не вопрос «равен ли уровень двум?», а команда «положи в уровень 2». Компьютер её честно выполнит, и условие всегда окажется верным. Эта ошибка встретится нам в задачах.

Два условия сразу соединяют так: `&&` – «и» (нужны оба), `||` – «или» (хватит одного). «Открой окно, если душно и окно закрыто» – это `if (co2 > 700 && windowClosed)`.

Когда вариантов больше двух, пишут цепочку:

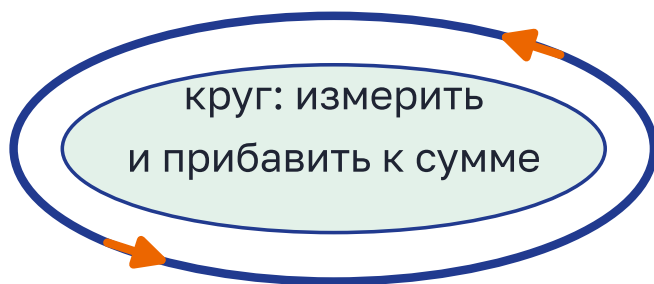
```
if (bpm > 100) {  
    Serial.println("частый");  
} else if (bpm < 60) {  
    Serial.println("редкий");  
} else {  
    Serial.println("обычный");  
}
```

Компьютер проверяет условия сверху вниз и выполняет **только первую** ветку, где ответ «да». Остальные даже не смотрит. При `bpm = 112` напечатается «частый». Слова здесь условные: для задачи мы сами решили, где граница. Про нормы пульса – в главе 2, и все они примерные.

Запомни. `if (вопрос) { ... } else { ... }` – развилка на две дороги. `=` сравнивает, `=` кладёт. В цепочке `if ... else if ... else` срабатывает только первая подходящая ветка. Внимательно смотри на границу: `> 100` и `>= 100` по-разному поступают с числом 100.

6.4. Цикл: повторить много раз

НА ПАЛЬЦАХ



1 2 3 4 5

счётчик i: 1, 2, 3, 4, 5

после пятого круга – стоп

Рисунок 6.4 – Цикл – круги по стадиону: на каждом круге одно и то же, счётчик считает круги

Тренер говорит: «Пять кругов». Ты бежишь, считаешь круги и после пятого останавливаешься. Цикл `for` делает то же самое.

Датчик шумит, и одному измерению верить нельзя. Надёжнее измерить пять раз и взять среднее. Писать пять одинаковых строчек скучно, поэтому есть **цикл** – команда «повтори».

Цикл `for` – «повтори столько-то раз»:

```
int sum = 0; // копилка для суммы
for (int i = 1; i ≤ 5; i++) { // i = 1, 2, 3, 4, 5
    int v = analogRead(A0); // одно измерение
    sum = sum + v;
}
Serial.println(sum / 5); // среднее из пяти
```

В скобках у `for` три части через точку с запятой. `int i = 1` выполняется один раз – заводим счётчик кругов. `i ≤ 5` – условие: пока оно верно, цикл крутится. `i++` – то же, что `i = i + 1`, выполняется после каждого круга. Если датчик выдал 500, 520, 510, 530 и 540, сумма растёт так: 500, 1020, 1530, 2060, 2600. Среднее: $2600 : 5 = 520$.

Цикл `while` – «повторяй, пока верно условие». Он удобен, когда заранее неизвестно, сколько будет кругов. Например, прибор ждёт, пока на датчик положат палец:

```
int tries = 0;
while (analogRead(A0) < 100) { // пальца на датчике нет
    delay(500); // подождать полсекунды
    tries = tries + 1;
}
Serial.println(tries);
```

Условие проверяется **в начале** каждого круга. Допустим, датчик отвечает 20, 35, 40, а потом 610. Три раза условие верно – три раза ждём и прибавляем. На четвёртой проверке 610 не меньше 100, цикл заканчивается, и печатается 3.

Как узнать, что напечатает программа с циклом? Не угадывать, а **проследить**. Заведи таблицу: столбец на каждую переменную, строка на каждый круг. Выполняй программу паль-

цем и записывая новые значения. Такую табличку называют **трассировкой**, и в задачах «что выведет программа» она выручает всегда.

Запомни. `for` – повторить заданное число раз, `while` – повторять, пока условие верно. Условие проверяется перед каждым кругом: когда оно стало ложным, цикл заканчивается сразу. Чтобы узнать, что выведет цикл, води трассировку.

6.5. Скетч Arduino: `setup`, `loop` и монитор порта

НА ПАЛЬЦАХ

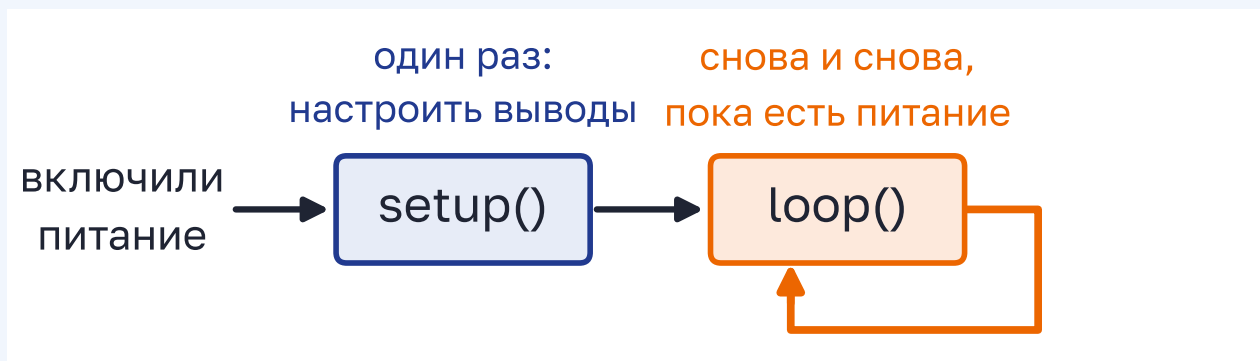


Рисунок 6.5 – Как работает скетч: сначала один раз `setup()`, потом `loop()` по кругу, пока есть питание
`setup()` – утренняя зарядка, её делают один раз. `loop()` – дворник со снегом: до-
чистил дорожку до конца – идёт в начало.

Arduino Uno – небольшая плата с **микроконтроллером**, крошечным компьютером в одной микросхеме. У платы есть пронумерованные **выводы** (их ещё называют пинами): **цифровые** 0–13 и **аналоговые входы** A0–A5. К выводам подключают светодиоды, датчики, моторы. Как устроены цепь, светодиод и датчик, рассказано в главе 5, здесь – только программа. Программу для Arduino называют **скетчем**.

В каждом скетче есть две обязательные части. `setup()` выполняется один раз сразу после включения питания. В нём настраивают выводы и включают связь с компьютером. `loop()` выполняется потом снова и снова, пока плату не выключат: дошла до последней строки – начинает с первой. Если в ней нет пауз, таких кругов набегает тысячи в секунду.

```

const int LED = 13;           // светодиод - на выводе 13

void setup() {                // выполняется один раз при включении
  pinMode(LED, OUTPUT);       // вывод 13 работает как выход
  Serial.begin(9600);         // включаем связь с компьютером
}

void loop() {                 // повторяется по кругу, пока есть питание
  int v = analogRead(A0);     // напряжение на A0 - число от 0 до 1023
  Serial.println(v);          // отправляем число в монитор порта
  digitalWrite(LED, HIGH);    // 5 В на вывод 13 - светодиод горит
  delay(100);                 // ждём 100 мс
  digitalWrite(LED, LOW);     // 0 В - погас
  delay(400);                 // ждём ещё 400 мс
}

```

Разберём строчки.

- `const int LED = 13;` – **константа**: коробочка, содержимое которой менять нельзя. Имя `LED` читать легче, чем голое число 13, а если светодиод переедет на другой вывод, поправишь одну строку.
- `pinMode(LED, OUTPUT)` – вывод 13 будет **выходом**: плата сама подаёт на него напряжение. Для датчика или кнопки вывод делают входом: `INPUT`.
- `digitalWrite(LED, HIGH)` – подать на вывод 5 В, светодиод загорается. `LOW` – 0 В, гаснет. Других вариантов у цифрового вывода нет, только эти два.
- `analogRead(A0)` – измерить напряжение на аналоговом входе A0. Ответ – целое число от 0 (это 0 В) до 1023 (это 5 В). Так Arduino читает потенциометр, датчик пульса, огибающую ЭМГ.
- `delay(100)` – ничего не делать 100 **миллисекунд**. В одной секунде 1000 мс, поэтому `delay(1000)` – одна секунда.
- `Serial.begin(9600)` – в `setup()` включить связь с компьютером по кабелю, 9600 – её скорость. Без этой строки монитор порта будет молчать.
- `Serial.println(v)` – отправить число на компьютер и перейти на новую строку. `Serial.print(v)` делает то же, но строку не переводит: следующее число встанет рядом.

Числа, которые присылает плата, видно в **мониторе порта** – окне внизу под кодом в симуляторе и в Arduino IDE. Рядом есть **плоттер**: он рисует из этих чисел график, и сигнал становится видно глазами. Так на курсе сферы смотрят пульсовую волну и ЭМГ.

Сколько раз в секунду этот скетч измеряет сигнал? Один круг `loop()` – это $100 + 400 = 500$ мс, остальные команды выполняются за доли миллисекунды. Значит, два измерения в секунду, и светодиод мигает дважды в секунду – как маячок «прибор работает».

Совет. Любой скетч из этой главы можно запустить в симуляторе Wokwi (wokwi.com): выбираешь плату Arduino Uno, вставляешь код, жмёшь запуск. На курсе сферы работают в Tinkercad: как войти в класс без регистрации – во введении, «Tinkercad: вход в класс и первая схема». Сомневаешься, что лежит в переменной, – добавь `Serial.println` и посмотри своими глазами.

Запомни. `setup()` – один раз, `loop()` – по кругу. `digitalWrite` – только HIGH (5 В) или LOW (0 В), `analogRead` – число от 0 до 1023. Время у Arduino в миллисекундах. Длительность круга `loop()` – сумма всех `delay` в нём.

6.6. Блоки и текст: один алгоритм – два языка

НА ПАЛЬЦАХ

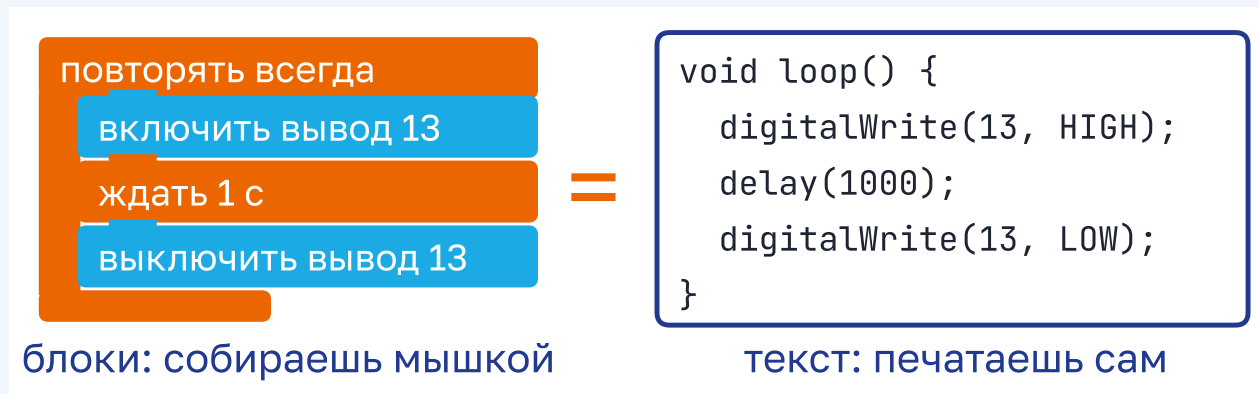


Рисунок 6.6 – Слева программа из блоков, справа та же программа текстом

Блоки собирают мышкой, как конструктор, текст набирают на клавиатуре. Плата получит одно и то же.

Программу можно не только печатать, но и собирать из цветных **блоков**. Каждый блок – одна команда, надпись на нём по-русски. Блок «повторять всегда» похож на скобу, внутрь которой вставляют другие блоки. Блок «если ... то ... иначе» – скоба с двумя отделениями. Защёлкнуть блок не туда, куда он подходит, нельзя, поэтому ошибок вроде забытой точки с запятой в блоках не бывает.

Блоки для Arduino есть в mBlock и в Tinkercad (там окно кода открывается сразу с блоками). Одним щелчком можно переключиться на текст, и среда покажет ту же программу строчками на C++. Для учёбы это удобно: собрал из блоков – посмотрел, как это пишется словами. На курсе сферы дальше работают с текстом: длинную программу из блоков собирать долго, а текст легко поправить и скопировать.

Вот программа «Свежий воздух» из блоков. Она раз в секунду читает датчик углекислого газа и зажигает красный светодиод, если показание больше 700, а иначе – зелёный:

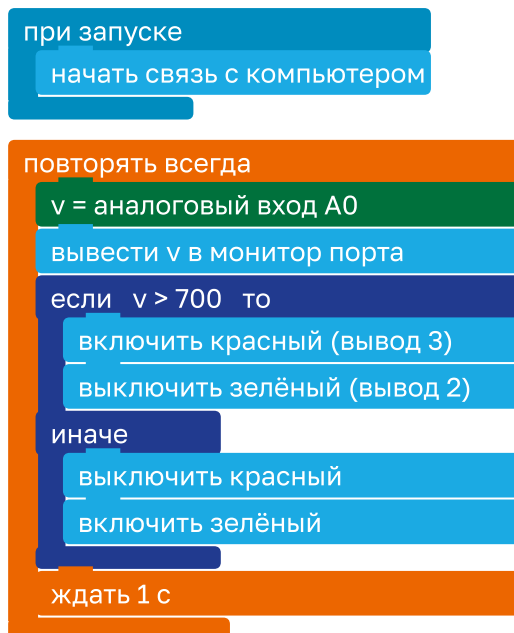


Рисунок 6.7 – Программа из блоков. Надписи на блоках в разных средах немного отличаются, но устроены они одинаково

А вот она же текстом:

```
const int GREEN = 2;
const int RED = 3;

void setup() {
  pinMode(GREEN, OUTPUT);
  pinMode(RED, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int v = analogRead(A0);
  Serial.println(v);
  if (v > 700) {
    digitalWrite(RED, HIGH);
    digitalWrite(GREEN, LOW);
  } else {
    digitalWrite(RED, LOW);
    digitalWrite(GREEN, HIGH);
  }
  delay(1000);
}
```

Сравни их построчно. Блок «при запуске» – это `setup()`, «повторять всегда» – `loop()`. Скоба «если ... то ... иначе» превратилась в `if (...) { ... } else { ... }`, а блоки внутри скобы – в строки внутри фигурных скобок. «Ждать 1 с» стало `delay(1000)`: в блоках время часто пишут в секундах, в тексте – всегда в миллисекундах. Отдельно посмотри, где стоит «ждать 1 с»: **под** скобой «если», а не внутри неё. Значит, пауза нужна в обоих случаях. Засунешь её в ветку «иначе» – и при высоком показании программа перестанет ждать.

Настройку выводов (`pinMode`) в блоках иногда делать не нужно: среда добавляет её сама. В тексте её пишут руками, и если забыть, светодиод на выводе будет гореть еле-еле или не загорится.

Запомни. «При запуске» = `setup()` , «повторять всегда» = `loop()` , скоба «если – иначе» = `if ... else` . Что стоит внутри скобы, в тексте стоит внутри фигурных скобок. В тексте время – в миллисекундах.

6.7. Машина состояний: покой, измерение, тревога

НА ПАЛЬЦАХ



Рисунок 6.8 – Уличный фонарь с датчиком света: два состояния и два события

Фонарь всегда в одном состоянии: либо «день», либо «ночь». Переключает его событие: стемнело или рассвело.

Настоящий прибор не просто повторяет одно и то же. Он живёт в разных **режимах**: ждёт, измеряет, бьёт тревогу – и в каждом режиме реагирует на одно и то же по-разному. Такой прибор удобно описывать **машиной состояний**. Словарь у неё короткий:

- **состояние** – режим, в котором устройство находится прямо сейчас: «покой», «измерение», «тревога». В каждый момент оно одно, двух сразу не бывает;
- **событие** – то, что случилось и на что нужно ответить: «прошло 10 секунд», «пришло показание датчика»;
- **условие** – проверка в квадратных скобках: переход сработает, только если она верна, например `[показание > 600]`;
- **переход** – стрелка из одного состояния в другое. Подпись на стрелке: «событие [условие]».

У состояния бывает **действие на входе**: его выполняют один раз, когда машина в это состояние попала. На схеме его пишут внутри прямоугольника: «вход / горит красный». Начальное состояние отмечают стрелкой из чёрной точки.

Так программы рисуют на языке **ПРИМС** – расширенных иерархических машин состояний. В курсе сферы на нём работают в среде **Кибериада IDE**: схему рисуют мышкой, а код на C++ для Arduino среда делает по ней сама. «Иерархические» – потому что внутри одного состоя-

ния может жить своя маленькая машина. В задачах этой книги такого нет. Если платы и Кибериады под рукой нет, схему можно рисовать на бумаге: логика от этого не меняется.

Вот машина «Свежий воздух» для класса. Датчик углекислого газа выдаёт число: чем больше в воздухе углекислого газа, тем оно больше. Выше 600 – пора проветрить (эту границу мы задали сами, для задачи).

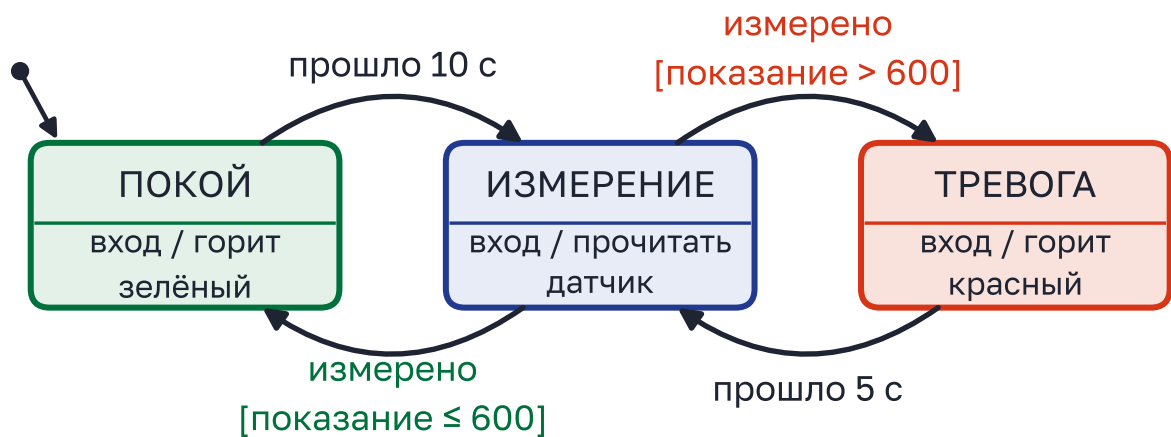


Рисунок 6.9 – Машина «Свежий воздух»: покой – измерение – тревога

Как она работает. Включили – машина в **покое**, горит зелёный. Прошло 10 секунд – переход в **измерение**, на входе прибор читает датчик. Из измерения две стрелки, и выбирает между ними условие: показание больше 600 – в **тревогу** (горит красный), иначе обратно в покой. Из тревоги через 5 секунд машина снова идёт мерить. Если в классе всё ещё душно – опять тревога, если проветрили – покой.

Проследим, что будет, если измерения дадут по очереди 450, 580, 640, 700 и 520.

Таблица 6.1

Время, с	Показание	Куда перешла машина	Что горит
0	–	покой (начало)	зелёный
10	450	покой	зелёный
20	580	покой	зелёный
30	640	тревога	красный
35	700	тревога	красный
40	520	покой	зелёный

Заметь: после тревоги прибор мерит не через 10, а через 5 секунд. Разные состояния – разные правила, в этом весь смысл машины.

А вот та же машина на C++. Номер состояния хранится в одной переменной `state`, момент входа в состояние – в `start`. Функция `enter` делает переход: запоминает новое состояние, запускает часы с нуля и выполняет действие на входе.

```

const int GREEN = 2;           // зелёный светодиод
const int RED = 3;            // красный светодиод
const int LIMIT = 600;        // выше этого показания - душно

const int REST = 0;           // состояние «покой»
const int MEASURE = 1;        // состояние «измерение»
const int ALARM = 2;          // состояние «тревога»

int state = REST;             // в каком состоянии машина сейчас
unsigned long start = 0;       // когда вошли в это состояние, мс
int v = 0;                    // последнее показание датчика

void enter(int newState) {     // переход в новое состояние
    state = newState;
    start = millis();          // часы состояния - с нуля
    if (state == REST) {       // вход в покой: горит зелёный
        digitalWrite(RED, LOW);
        digitalWrite(GREEN, HIGH);
    }
    if (state == MEASURE) {    // вход в измерение: прочитать датчик
        v = analogRead(A0);
        Serial.println(v);
    }
    if (state == ALARM) {      // вход в тревогу: горит красный
        digitalWrite(GREEN, LOW);
        digitalWrite(RED, HIGH);
    }
}

void setup() {
    pinMode(GREEN, OUTPUT);
    pinMode(RED, OUTPUT);
    Serial.begin(9600);
    enter(REST);               // начальное состояние - покой
}

void loop() {
    unsigned long t = millis() - start; // сколько мс длится текущее состояние
    if (state == REST && t ≥ 10000) {    // прошло 10 с
        enter(MEASURE);
    } else if (state == MEASURE && v > LIMIT) { // измерено [показание > 600]
        enter(ALARM);
    } else if (state == MEASURE) {      // измерено [показание ≤ 600]
        enter(REST);
    } else if (state == ALARM && t ≥ 5000) { // прошло 5 с
        enter(MEASURE);
    }
}

```

Посмотри на `loop()`: каждая её ветка – одна стрелка со схемы. «Если сейчас покой и прошло 10 000 мс – переходим в измерение». Новое здесь – `millis()`: сколько миллисекунд прошло с включения платы. Это часы контроллера. Разность `millis() - start` показывает, сколько мы уже сидим в текущем состоянии. Для времени берут тип `unsigned long` – целое число без минуса, в которое влезают миллиарды: в `int` миллисекунды перестали бы помещаться уже через 33 секунды.

Почему не `delay(10000)` ? Пока идёт `delay` , контроллер спит и ничего не замечает. Машина со своими часами `loop()` не останавливает: она крутится тысячи раз в секунду и каждый раз только спрашивает «не пора ли?». Поэтому в неё легко добавить ещё одно событие – например, реакцию на новый датчик, – и оно не будет ждать, пока закончится пауза. Запусти скетч в Wokwi с потенциометром на A0 и следи за монитором порта: пока ручка внизу, числа приходят раз в 10 секунд, а подкрутишь выше 600 – раз в 5.

Когда проверяешь чужую машину, задай себе два вопроса. Есть ли **тупик** – состояние, из которого не выходит ни одна стрелка? Попала машина в тупик – прибор «завис». И не могут ли две стрелки из одного состояния сработать одновременно? Если могут, нужно правило, какая главнее, иначе непонятно, куда идти.

Совет. Задачу «проследи машину» решай таблицей: шаг – текущее состояние – событие – какое условие верно – новое состояние. Одно событие – не больше одного перехода. Попал в новое состояние – следующую стрелку ищи уже из него и только на следующем событии.

Запомни. Машина состояний: в каждый момент одно состояние, переходы по событиям, условие в квадратных скобках решает, сработает ли стрелка. В коде – переменная `state` и цепочка `if ... else if` в `loop()` , время – через `millis() - start` , а не через `delay` .

6.8. Разбираем задачи

Задачи с кодом в этой главе – четырёх видов, и к каждому есть свой ключ.

- **«Что выведет программа».** Не угадывай – води трассировку: столбец на переменную, строка на шаг. Особенно внимательно – с целым делением и с последним кругом цикла.
- **«Расставь строки».** Ищи зацепки: переменную заводят раньше, чем используют; `if` раньше `else` ; у каждой `{` есть своя `}` . Если две строки можно поменять без вреда, порядок подскажет условие.
- **«Найди ошибку».** Подставь в программу числа из условия, особенно граничные: число на самом пороге, ноль, самое большое. Частые ошибки: `=` вместо `==` , `>` вместо `>=` , лишняя или пропущенная проверка, пауза не в той ветке.
- **«Проследи машину состояний».** Таблица по шагам, одно событие – один переход.

Задача 6.1 Исполнитель Яркость

разминка · алгоритм, исполнитель, система команд, обратный ход

Исполнитель Яркость хранит одно число – яркость светодиода – и знает две команды: **1** – «прибавь 20», **2** – «умножь на 2». Программа – строка из номеров команд, читается слева направо. а) Яркость была 5. Какой она станет после программы 12211? б) Найди программу из трёх команд, которая превращает 5 в 60. Ответ – строка из цифр.

Дано: команда 1: $x + 20$; команда 2: $x \cdot 2$; а) начальное 5, программа 12211; б) из 5 в 60 за 3 команды.

Найти: а) результат; б) программу.

Решение

Шаг 1. Прогон программы 12211. Пишем число после каждой команды, не перескакивая:

- 1: $5 + 20 = 25$;
- 2: $25 \cdot 2 = 50$;
- 2: $50 \cdot 2 = 100$;
- 1: $100 + 20 = 120$;
- 1: $120 + 20 = 140$.

Шаг 2. Обратный ход для 60. Последней командой могла быть «прибавь 20» – тогда перед ней было $60 - 20 = 40$. Или «умножь на 2» – тогда было $60 : 2 = 30$. Осталось две команды, и нужно проверить, можно ли за них получить из 5 число 40 или 30.

Шаг 3. Что получается из 5 за две команды. Вариантов всего четыре: 11 даёт $5 + 20 + 20 = 45$; 12 даёт $(5 + 20) \cdot 2 = 50$; 21 даёт $5 \cdot 2 + 20 = 30$; 22 даёт $5 \cdot 2 \cdot 2 = 20$. Числа 40 среди них нет, а 30 есть – это программа 21.

Шаг 4. Собираем. Из 5 программой 21 получаем 30, потом команда 2: $30 \cdot 2 = 60$. Вся программа – 212.

Прогоним её для верности: $5 \cdot 2 = 10$, $10 + 20 = 30$, $30 \cdot 2 = 60$.

 **Типичная ошибка.** В пункте а) выполнить сначала все «умножь», а потом все «прибавь», как в школьном примере с порядком действий. Исполнитель так не умеет: он делает команды строго по очереди, слева направо. Если сначала два раза умножить, а потом три раза прибавить, выйдет $5 \cdot 4 + 60 = 80$ вместо 140.

Ответ: а) 140; б) 212.

Своя задача по программе сферы, модуль 1

Реши сам

6.1. Исполнитель Громкость знает две команды: **1** – «прибавь 3», **2** – «умножь на 3». Громкость была 2. Какой она станет после программы 21211?

6.2. Исполнитель Удвоитель знает команды **1** – «прибавь 5» и **2** – «умножь на 2». Найди **самую короткую** программу, которая превращает 1 в 49. Такая программа одна. Ответ – строка из цифр.

Задача 6.2 Датчик дыхания

разминка · скетч, `analogRead`, `Serial.println`, `if - else`, расставь строки

На курсе сферы датчик дыхания часто заменяют потенциометром: крутишь ручку – «сигнал» меняется. Программа должна раз за разом читать вход A0, отправлять число в монитор порта, а потом зажигать светодиод на выводе 13, если показание больше 600, и гасить его, если нет. Строки внутри `loop()` перемешались. Расставь их по порядку. Число уходит в монитор порта **раньше**, чем программа решает, что делать со светодиодом. Ответ – последовательность букв.

```
void setup() {
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // сюда - семь строк:
  // А } else {
  // Б   digitalWrite(13, LOW);
  // В int v = analogRead(A0);
  // Г if (v > 600) {
  // Д   Serial.println(v);
  // Е }
  // Ж   digitalWrite(13, HIGH);
}
```

Дано: семь строк А–Ж; сначала вывод числа, потом светодиод; светодиод горит при показании больше 600.

Найти: порядок строк.

Решение

Шаг 1. Что должно быть первым. И вывод, и условие работают с переменной `v`. Коробочку нужно завести и наполнить раньше, чем ею пользоваться. Первая строка – В: `int v = analogRead(A0);`.

Шаг 2. Вывод числа. По условию число уходит в монитор порта раньше решения про светодиод. Вторая строка – Д.

Шаг 3. Развилка. Дальше вопрос – строка Г `if (v > 600) {`. Сразу за ней ветка «да»: при показании больше 600 светодиод горит – это Ж. Потом `} else {` – строка А, за ней ветка «нет» – Б. Закрывает развилку скобка Е.

Шаг 4. Проверка скобками. Открывающих фигурных скобок в наших строках две – в Г и в А. Закрывающих тоже две – в А и в Е. Скобка в А закрывает ветку «да» и сразу открывает «иначе».

Подставь пару чисел. При 300 светодиод тёмный, при 650 горит. А при 600? Шестьсот не больше шестисот, так что на самой границе он тоже тёмный – загорится только с 601.

❗ **Типичная ошибка.** Поставить строку Д (вывод) первой – «потому что про неё сказано в начале». Но `v` ещё не существует, и программа не соберётся: симулятор подчеркнёт строку и напишет, что такое имя не объявлено. Другая ошибка – поменять местами Ж и Б. Скetch соберётся, но светодиод будет гореть наоборот, при низком показании.

Ответ: В, Д, Г, Ж, А, Б, Е.

Своя задача по программе сферы, модуль 2

Реши сам

6.3. Тревожный маячок на выводе 9 должен вспыхивать на 200 мс и гаснуть на 800 мс. В момент каждой вспышки, сразу после включения, в монитор порта уходит слово «тревога». Сразу после включения платы маячок загорается. Вывод 9 уже настроен в `setup()`. Расставь пять строк функции `loop()`: А `delay(800)`; Б `digitalWrite(9, HIGH)`; В `Serial.println("тревога");` Г `delay(200)`; Д `digitalWrite(9, LOW)`; . Ответ – последовательность букв.

6.4. Программа из блоков (надписи блоков записаны словами, отступ показывает, что стоит внутри скобы):

```
при запуске
  сделать вывод 12 выходом
повторять всегда
  v = аналоговый вход A0
  если v < 200 то
    включить вывод 12
  иначе
    выключить вывод 12
  ждать 0,5 с
```

Какой из вариантов функции `loop()` делает то же самое? Ответ – номер варианта.

```
// Вариант 1
int v = analogRead(A0);
if (v < 200) { digitalWrite(12, HIGH); } else { digitalWrite(12, LOW); delay(500); }

// Вариант 2
int v = analogRead(A0);
if (v > 200) { digitalWrite(12, HIGH); } else { digitalWrite(12, LOW); }
delay(500);

// Вариант 3
int v = analogRead(A0);
if (v < 200) { digitalWrite(12, HIGH); } else { digitalWrite(12, LOW); }
delay(500);

// Вариант 4
int v = analogRead(A0);
if (v < 200) { digitalWrite(12, HIGH); } else { digitalWrite(12, LOW); }
delay(0.5);
```

Задача 6.3 Пульс после бега

цикл `while`, целое деление, трассировка, что выведет

Упрощённая модель: сразу после бега пульс 150 уд/мин, и каждую минуту отдыха он уменьшается на пятую часть. Программа считает, через сколько минут пульс опустится до 90 или ниже. Что она выведет? Ответ – два числа по порядку.

```
void setup() {
  Serial.begin(9600);
  int pulse = 150;           // пульс сразу после бега, уд/мин
  int minutes = 0;           // сколько минут прошло
  while (pulse > 90) {
    pulse = pulse - pulse / 5;
    minutes = minutes + 1;
  }
  Serial.println(minutes);
  Serial.println(pulse);
}

void loop() {
}
```

Дано: начальное `pulse = 150`; за круг `pulse = pulse - pulse / 5`; цикл идёт, пока `pulse > 90`; все переменные целые.

Найти: два напечатанных числа.

Решение

Шаг 1. Где тут ловушка. Переменные типа `int`, значит, `pulse / 5` – целое деление, дробная часть отбрасывается. Условие `pulse > 90` проверяется в начале каждого круга.

Шаг 2. Трассировка. Строка таблицы – один круг цикла.

Таблица 6.2

Круг	pulse до	pulse > 90 ?	pulse / 5	pulse после
1	150	да	30	120
2	120	да	24	96
3	96	да	19	77
–	77	нет – выход	–	–

После каждого круга `minutes` растёт на 1, так что после третьего круга в ней 3. На третьем круге $96/5$ даёт 19, а не 19,2, поэтому пульс становится $96 - 19 = 77$.

Шаг 3. Что печатается. После цикла – сначала `minutes`, потом `pulse`: 3 и 77.

Пульс, который за три минуты отдыха падает со 150 почти до 80, для школьника после пробежки выглядит правдоподобно. Но это модель для задачи, а не правило для настоящего сердца.

❗ **Типичная ошибка.** Считать на калькуляторе: $96 - 96:5 = 76,8$ и записать 76,8 или 77 «по округлению». Число 77 здесь совпало случайно, а рассуждение неверное: при других числах оно подведёт. Вторая ошибка – остановиться на 96, решив, что 96 «уже почти 90». Цикл крутится, пока условие верно, а $96 > 90$ – верно.

Ответ: 3 и 77.

Своя задача по программе сферы, модуль 1

Реши сам

6.5. Что выведет программа? Ответ – три числа по порядку.

```
void setup() {
  Serial.begin(9600);
  int breaths = 20;           // вдохов за первую минуту
  int total = 0;              // сколько всего вдохов
  for (int m = 1; m ≤ 4; m++) {
    total = total + breaths;
    breaths = breaths - 2;
  }
  Serial.println(total);
  Serial.println(total / 4);
  Serial.println(breaths);
}

void loop() {
}
```

6.6. Сигнал тревоги стихает: каждый раз громкость уменьшается вдвое. Сколько чисел программа напечатает в первой строке, какое из них последнее и что будет во второй строке?

```
void setup() {
  Serial.begin(9600);
  int volume = 1000;          // громкость, условные единицы
  int n = 0;                  // сколько раз убавили
  while (volume ≥ 100) {
    volume = volume / 2;
    n = n + 1;
    Serial.print(volume);
    Serial.print(" ");
  }
  Serial.println();
  Serial.println(n);
}

void loop() {
}
```

Задача 6.4 Сколько раз сжали кулак

найди ошибку, порог, пересечение снизу вверх, флаг

Программа должна считать сжатия кулака по огибающей ЭМГ: сжатие засчитывается, когда сигнал **пересёк** порог 400 снизу вверх. Пока сигнал остаётся выше порога, это то же самое сжатие. Номера строк стоят в комментариях справа.

```
const int THRESHOLD = 400;           // (1) порог
int count = 0;                       // (2) сколько сжатий насчитали
bool above = false;                  // (3) сигнал сейчас выше порога?

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

void loop() {
    int v = analogRead(A0);           // (4) огибающая ЭМГ
    if (v > THRESHOLD) {               // (5)
        count = count + 1;            // (6)
        above = true;                 // (7)
    }
    if (v ≤ THRESHOLD) {              // (8)
        above = false;                // (9)
    }
    Serial.println(count);             // (10)
    delay(100);
}
```

Датчик выдал 12 значений: 80, 120, 450, 520, 480, 150, 90, 430, 470, 110, 500, 100. а) Какое число программа напечатает последним? б) Сколько раз человек сжал кулак? в) В какой строке ошибка? Ответ – три числа.

Дано: порог 400; 12 значений огибающей; сжатие – переход через порог снизу вверх.

Найти: последнее напечатанное число; настоящее число сжатий; номер строки с ошибкой.

Решение

Шаг 1. Что делает программа. Строка (5) спрашивает только «выше ли порога?». Значит, `count` растёт на каждом значении выше 400, сколько бы их ни шло подряд. Переменная `above` старательно ставится в `true` и `false`, но нигде не проверяется – она ни на что не влияет.

Шаг 2. Считаем, что напечатает программа. Значения выше 400: 450, 520, 480, 430, 470, 500 – шесть штук. Последнее напечатанное число – 6.

Шаг 3. Сколько было сжатий. Отмечаем места, где сигнал перешёл через порог снизу вверх:

- 120 → 450 – первое сжатие (дальше 520 и 480 – то же самое);
- 90 → 430 – второе (470 – оно же);
- 110 → 500 – третье.

Сжатий три.

Шаг 4. Где ошибка. Счётчик должен расти, только если сигнал выше порога **и** до этого был не выше. Для этого и заведена `above`. Исправляем строку (5):

```
if (v > THRESHOLD && !above) {    // (5) выше порога и только что был ниже
```

Знак `!` значит «не»: `!above` – «сигнал раньше не был выше порога». Пройди исправленную программу по тем же 12 числам: счётчик вырастет только на 450, 430 и 500, и последним напечатается 3 – столько, сколько было сжатий.

❗ Типичная ошибка. Искать ошибку в строках (8)–(9) – ведь «сбрасывается» что-то там. Но они как раз правильные: когда сигнал опустился, флаг нужно сбросить, чтобы поймать следующее сжатие. Ещё одна ловушка – ответить на вопрос а) числом 3: на вопрос «что напечатает программа» отвечают про программу как есть, со всеми её ошибками.

Ответ: а) 6; б) 3; в) строка 5, нужно `if (v > THRESHOLD && !above)`.

Своя задача по программе сферы, модуль 2

Реши сам

6.7. Прибор в классе должен писать «проветрить», только когда уровень равен 2. Датчик выдал по очереди 350, 520, 810, 640, 380. Сколько раз программа напечатает «проветрить»? В какой строке ошибка и сколько раз слово напечаталось бы без неё?

```
int level = 0;                                // (1) 0 - свежо, 1 - душновато, 2 - душно

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

void loop() {
    int v = analogRead(A0);                    // (2) показание датчика CO2
    if (v > 700) {                              // (3)
        level = 2;                             // (4)
    } else if (v > 400) {                        // (5)
        level = 1;                             // (6)
    } else {
        level = 0;                             // (7)
    }
    if (level == 2) {                           // (8)
        Serial.println("проветрить");           // (9)
    } else {
        Serial.println("норма");               // (10)
    }
    delay(1000);
}
```

6.8. Программа должна считать секунды, когда пульс был **100 уд/мин и выше** (граница условная, для задачи). Пульсометр раз в секунду выдаёт: 92, 100, 104, 99, 100, 100, 97, 101. Какое число программа напечатает последним? В какой строке ошибка и какое число должно было получиться?

```
int seconds = 0; // (1) сколько секунд пульс был высоким

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

void loop() {
  int bpm = analogRead(A0); // (2) ЧСС от пульсометра, уд/мин
  if (bpm > 100) { // (3)
    seconds = seconds + 1; // (4)
  }
  Serial.println(seconds); // (5)
  delay(1000);
}
```

Задача 6.5 Пульсометр-тренажёр

машина состояний, два порога, трассировка переходов

Тренажёр для спокойного дыхания устроен как машина состояний на рисунке. Раз в секунду приходит показание: либо «пальца нет», либо ЧСС в уд/мин. Правила: за одно показание машина делает не больше одного перехода; если ни одна стрелка из текущего состояния не подходит, машина остаётся на месте. Начало – в состоянии 1.

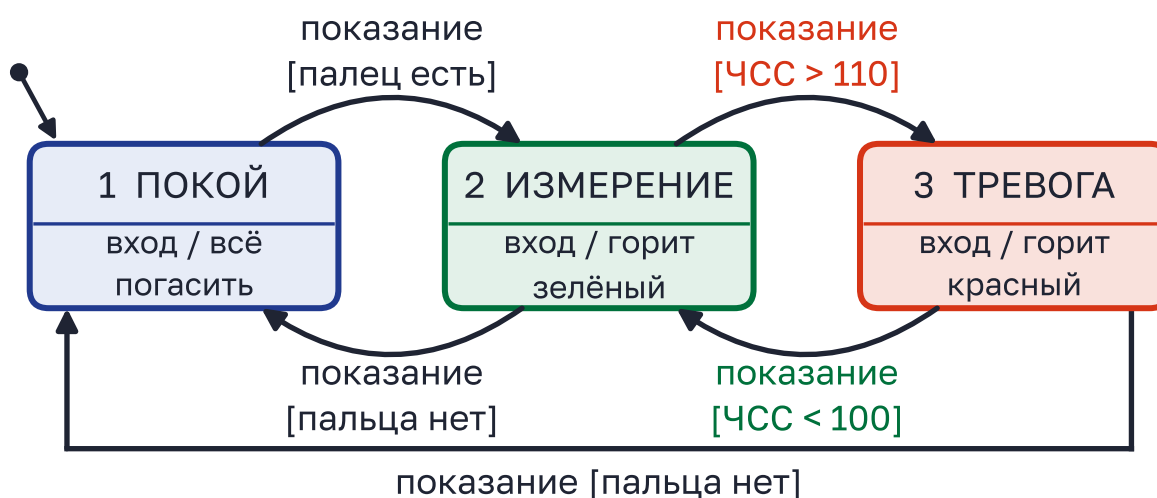


Рисунок 6.10 – Машина тренажёра. Номера состояний стоят перед названиями

За 12 секунд пришли показания: нет, 80, 95, 112, 118, 105, 99, 115, нет, 120, 121, 90 («нет» – пальца нет). а) Запиши номера состояний после каждого показания одной строкой из 12 цифр. б) Сколько раз загорался красный?

Дано: схема машины; 12 показаний; начальное состояние 1; одно показание – не больше одного перехода.

Найти: строку из 12 номеров состояний; число включений красного.

Решение

Шаг 1. Выписываем стрелки. Из 1 (покой): [палец есть] → 2. Из 2 (измерение): [пальца нет] → 1; [ЧСС > 110] → 3. Из 3 (тревога): [пальца нет] → 1; [ЧСС < 100] → 2. Порогов два: в тревогу – выше 110, обратно – ниже 100. Между ними, от 100 до 110, машина остаётся где была.

Шаг 2. Трассировка.

Таблица 6.3

Секунда	Показание	Было	Стало
1	нет	1	1
2	80	1	2
3	95	2	2
4	112	2	3
5	118	3	3
6	105	3	3
7	99	3	2
8	115	2	3
9	нет	3	1
10	120	1	2
11	121	2	3
12	90	3	2

В первую секунду пальца нет, и из покоя машина не уходит. Дальше каждая строка читается по стрелкам из шага 1. Самые скользкие места – 6-я и 10-я секунды, о них ниже.

Шаг 3. Ответ. Строка номеров: 122333231232. Красный загорался при каждом входе в состояние 3, а входов три – на 4-й, 8-й и 11-й секундах.

Посмотри на 6-ю секунду. Пульс 105 уже не такой высокий, но тревога не снимается: машина ждёт, пока он опустится ниже 100. Будь порог один, на 110, красный на пульсе около 110 мигал бы каждую секунду. Такие два порога мы уже видели в главе 3, у бионической руки.

 **Типичная ошибка.** На 10-й секунде рассуждать «пульс 120 – значит, тревога» и сразу записать 3. Но машина была в покое, и за одно показание она успевает только перейти в измерение. Вторая ошибка – на 6-й секунде вернуться в 2, потому что 105 «меньше, чем 110». Из тревоги ведёт стрелка [ЧСС < 100], а не [ЧСС ≤ 110].

Ответ: а) 122333231232; б) 3 раза.

Своя задача по программе сферы, модуль 1

Реши сам

6.9. Машина та же, что в задаче 6.5, начало – в состоянии 1. Показания за 12 секунд: 85, нет, нет, 117, 114, 108, 101, 100, 96, 113, нет, 104. Запиши номера состояний после каждого показания строкой из 12 цифр. Сколько раз загорался красный?

6.10.* Машина «Свежий воздух» из раздела 6.7 включилась в момент 0. Измерения по порядку дали 620, 650, 580, 590, 700, 710, 560, 540. Сколько секунд из первой минуты после включения горел красный светодиод? Время на само измерение не учитывай.

Задача 6.6 Стимулы для опыта на внимание

машина состояний в коде, millis, таймер, что выведет

В опыте на внимание человеку показывают вспышки светодиода – **стимулы**, а прибор записывает, как на них отвечает мозг. Стимул горит 600 мс, потом пауза 400 мс, и так 6 раз. После шестого стимула пауза уже не нужна: машина сразу переходит в состояние «конец» и печатает время. а) Что напечатает программа? б) Через сколько миллисекунд после включения загорится четвёртый стимул? Время на выполнение команд, кроме ожидания, не учитывай.

```

const int LED = 8;           // светодиод-стимул
const int SHOW_MS = 600;    // сколько мс горит стимул
const int PAUSE_MS = 400;   // сколько мс длится пауза
const int TOTAL = 6;        // сколько всего стимулов

const int SHOW = 0;         // состояние «стимул»
const int PAUSE = 1;        // состояние «пауза»
const int DONE = 2;        // состояние «конец»

int state = SHOW;
int shown = 1;              // сколько стимулов уже зажгли
unsigned long start = 0;    // когда вошли в текущее состояние, мс

void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
  digitalWrite(LED, HIGH);  // первый стимул горит сразу
  start = millis();
}

void loop() {
  unsigned long t = millis() - start;
  if (state == SHOW && t ≥ SHOW_MS) {
    digitalWrite(LED, LOW);
    if (shown == TOTAL) {    // это был последний стимул
      state = DONE;
      Serial.println(millis());
    } else {
      state = PAUSE;
    }
    start = millis();
  } else if (state == PAUSE && t ≥ PAUSE_MS) {
    digitalWrite(LED, HIGH);
    shown = shown + 1;
    state = SHOW;
    start = millis();
  }
}

```

Дано: стимул 600 мс, пауза 400 мс, 6 стимулов; после последнего стимула – сразу «конец»; первый стимул горит с момента 0.

Найти: напечатанное число; момент включения 4-го стимула.

Решение

Шаг 1. Состояния и переходы. Состояний три. «Стимул» → «пауза» через 600 мс, если зажгли ещё не все. «Пауза» → «стимул» через 400 мс, при этом `shown` растёт на 1. «Стимул» → «конец» через 600 мс, если `shown` уже равен 6. Из «конца» стрелок нет – это тупик, но здесь он нужен: опыт окончен.

Шаг 2. Прослеживаем по времени.

Таблица 6.4

Время, мс	Состояние	Какой стимул горит	shown
-----------	-----------	--------------------	-------

Время, мс	Состояние	Какой стимул горит	shown
0	стимул	1-й	1
600	пауза	–	1
1000	стимул	2-й	2
2000	стимул	3-й	3
3000	стимул	4-й	4
4000	стимул	5-й	5
5000	стимул	6-й	6
5600	конец	–	6

Паузы между стимулами (в 1600, 2600 мс и так далее) в таблице не выписаны, они такие же, как первая. Один цикл «стимул + пауза» длится $600 + 400 = 1000$ мс, поэтому k -й стимул загорается в момент $(k - 1) \cdot 1000$ мс. Четвёртый – в $3 \cdot 1000 = 3000$ мс.

Шаг 3. Когда конец. Шестой стимул загорается в $5 \cdot 1000 = 5000$ мс и горит 600 мс. Паузы после него нет: $5000 + 600 = 5600$ мс. Программа печатает `millis()` в момент перехода, то есть 5600. Шесть вспышек укладываются чуть больше чем в пять с половиной секунд – на секундомере телефона это легко проверить.

⚠ Типичная ошибка. Умножить $6 \cdot 1000$ и ответить 6000. Так было бы, если бы после шестого стимула тоже шла пауза. Прочитай ветку `if (shown == TOTAL)` : в последний раз машина идёт не в «паузу», а сразу в «конец». Ещё ошибка – считать, что четвёртый стимул загорается в 4000 мс: первый горит с нуля, а не с 1000.

Ответ: а) 5600; б) 3000 мс.

По мотивам задачи заключительного этапа НТО «Нейротехнологии и когнитивные науки» 2025/26, инженерный тур, «Базовая программа на PsychoPy»

Реши сам

6.11. Дыхательный тренажёр: светодиод горит 4 с – вдох, гаснет на 6 с – выдох. После третьего выдоха машина переходит в «конец» и печатает время. а) Что напечатает программа? б) В каком состоянии машина через 17 с после включения? в) Сколько секунд всего горел светодиод?

```
const int LED = 8;           // горит - вдох, погас - выдох
const int INHALE = 0;        // состояние «вдох»
const int EXHALE = 1;        // состояние «выдох»
const int DONE = 2;          // состояние «конец»

int state = INHALE;
int cycles = 0;               // сколько полных циклов «вдох - выдох» прошло
unsigned long start = 0;

void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
  digitalWrite(LED, HIGH);
  start = millis();
}

void loop() {
  unsigned long t = millis() - start;
  if (state == INHALE && t ≥ 4000) {
    digitalWrite(LED, LOW);
    state = EXHALE;
    start = millis();
  } else if (state == EXHALE && t ≥ 6000) {
    cycles = cycles + 1;
    if (cycles == 3) {
      state = DONE;
      Serial.println(millis());
    } else {
      digitalWrite(LED, HIGH);
      state = INHALE;
    }
    start = millis();
  }
}
```

6.12. Плата пишет сигнал ЭЭГ и рядом – счётчик, чтобы потом было видно, не потерялись ли строки. Одна строка – каждые 4 мс, на остальное время не уходит.

```
int counter = 0;           // счётчик строк

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

void loop() {
  int v = analogRead(A0);   // сигнал ЭЭГ
  counter = counter + 1;
  if (counter == 100) {
    counter = 0;
  }
  Serial.print(counter);
  Serial.print(" ");
  Serial.println(v);
  delay(4);
}
```

а) Какое значение счётчика будет в 250-й строке? б) Через сколько миллисекунд счётчик возвращается к тому же числу (на графике он похож на пилу)? в) В записи после строки со счётчиком 37 сразу идёт строка со счётчиком 41. Сколько строк потерялось?

Задача 6.7* Рука, которая не дёргается

машина состояний в коде, счётчик подряд, два порога, что выведет

Модель бионической руки сжимается по сигналу ЭМГ. Чтобы случайный всплеск шума не сжимал руку, программа ждёт **трёх показаний подряд** выше 500. Разжимается рука, когда сигнал опустится ниже 300. Показание приходит раз в 50 мс, первое – в момент 0.

```

const int REST = 0;           // рука разжата
const int CHECK = 1;         // проверяем: вдруг это случайный всплеск
const int GRIP = 2;          // рука сжата

int state = REST;
int count = 0;               // сколько показаний подряд выше 500

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

void loop() {
  int v = analogRead(A0);    // огибающая ЭМГ
  if (state == REST) {
    if (v > 500) {
      state = CHECK;
      count = 1;
    }
  } else if (state == CHECK) {
    if (v > 500) {
      count = count + 1;
      if (count == 3) {
        state = GRIP;
        Serial.println("сжать");
      }
    } else {
      state = REST;
    }
  } else if (state == GRIP) {
    if (v < 300) {
      state = REST;
      Serial.println("разжать");
    }
  }
  delay(50);
}

```

Датчик выдал 16 показаний: 120, 560, 180, 540, 610, 650, 700, 420, 350, 280, 520, 530, 150, 600, 640, 660. а) Какие слова и в каком порядке напечатает программа? б) На каком по счёту показании рука сожмётся впервые и сколько миллисекунд пройдёт к этому моменту от первого показания?

Дано: состояния «разжата», «проверка», «сжата»; сжатие – 3 показания подряд выше 500; разжатие – ниже 300; 16 показаний, шаг 50 мс.

Найти: напечатанные слова; номер показания и время первого сжатия.

Решение

Шаг 1. Правила словами. Из «разжата»: показание выше 500 – в «проверку», счётчик 1. Из «проверки»: выше 500 – счётчик +1, дошёл до 3 – в «сжата» и печать «сжать»; не выше 500 – назад в «разжата», всплеск оказался случайным. Из «сжата»: ниже 300 – в «разжата» и печать «разжать». Значения от 300 до 500 руку в «сжата» не трогают.

Шаг 2. Трассировка.

Таблица 6.5

№	Показание	Состояние после	count	Печать
1	120	разжата	0	
2	560	проверка	1	
3	180	разжата	1	
4	540	проверка	1	
5	610	проверка	2	
6	650	сжата	3	сжать
7	700	сжата	3	
8	420	сжата	3	
9	350	сжата	3	
10	280	разжата	3	разжать
11	520	проверка	1	
12	530	проверка	2	
13	150	разжата	2	
14	600	проверка	1	
15	640	проверка	2	
16	660	сжата	3	сжать

Показание 2 (560) – как раз случайный всплеск: следующее уже 180, и рука не дёрнулась. Показания 11–12 тоже оборвались на втором.

Шаг 3. Время. Рука впервые сжалась на 6-м показании. Показания идут через 50 мс, первое – в момент 0, значит, шестое – через пять промежутков: $5 \cdot 50 = 250$ мс.

Настоящее сжатие началось на 4-м показании, а рука ответила на 6-м – на 100 мс позже. Человек такой задержки почти не заметит, зато рука больше не вздрагивает от каждого всплеска.

 **Типичная ошибка.** На 8-м и 9-м показаниях разжать руку: 420 и 350 меньше 500, «значит, сигнал пропал». Но разжимает рука только ниже 300 – второй порог нужен как раз для этого. Ещё ошибка – во 2-й строке таблицы сразу перейти в «сжата», забыв про проверку, или сосчитать время как $6 \cdot 50 = 300$ мс: промежутков между шестью показаниями пять, а не шесть.

Ответ: а) «сжать», «разжать», «сжать»; б) на 6-м показании, через 250 мс.

Своя задача по программе сферы, модуль 2

Реши сам

6.13.* Программа из задачи 6.7, новые 16 показаний: 90, 520, 540, 310, 560, 580, 620, 640, 330, 290, 510, 470, 530, 560, 590, 240. Сколько раз программа напечатает «сжать»? Через сколько миллисекунд от первого показания рука сожмётся впервые?

6.14.* В программе из задачи 6.7 число 3 в строке `if (count == 3)` заменили. Что напечатает программа на показаниях задачи 6.7, если поставить 4? А если 5?

Таблицы, графики и статистика



Датчик CO₂ в классе за один урок выдаёт сотню чисел, пульсометр за минуту – тысячи. Глазами в таком столбце ничего не разглядишь. Поэтому числа складывают в таблицу, сворачивают до нескольких главных – среднего, наибольшего, наименьшего – и рисуют по ним график. Почти все примеры тут – про углекислый газ в классе и про пульс: с этими датчиками работают в программе сферы. На отборе такая задача – таблица или график и вопрос к ним, а в ответ вводят число или номера верных утверждений.

7.1. Электронная таблица: ячейки и формулы

НА ПАЛЬЦАХ

	А	В	С
1	мин	ppm	
2	0	480	
3	10	610	
4	20	760	
5	30	890	

Рисунок 7.1 – Адрес ячейки – буква столбца и номер строки

Как клетка «В3» в морском бое: сначала буква, потом цифра. В ячейке В3 лежит число 610.

Электронная таблица – программа, в которой лист расчерчен на клетки, как тетрадь по математике. Подойдёт любая: LibreOffice Calc (бесплатная, ставится на компьютер), Excel, Google Таблицы в браузере. Табличный редактор есть и в списке программ, которые организаторы назвали для финала Медтеха, – так что знакомство пригодится не только на отборе.

Клетка таблицы называется **ячейкой**. Столбцы подписаны латинскими буквами А, В, С..., строки – числами 1, 2, 3... **Адрес** ячейки – буква её столбца и номер строки: В3 – столбец В, строка 3. Буква всегда первая, и она латинская: русская «В» вместо латинской В – частая причина, почему формула не работает.

В ячейку можно записать текст («время, мин»), число (610) или **формулу**. Формула начинается со знака `=`. Увидев его, таблица считает и показывает результат вместо напечатанного. Напишешь в ячейку `=2+3` – увидишь 5. Напишешь `=B3-B2` – таблица возьмёт число из В3, вычитет число из В2 и покажет разность. Поменяешь число в В3 – разность пересчитается сама. Ради этого таблицы и придумали.

Вот запись датчика CO₂ в кабинете биологии: в столбце А – минуты от начала урока, в столбце В – показания. Что такое ppm, разберём в разделе 7.4, пока это просто числа.

Таблица 7.1

Строка	А (время, мин)	В (CO ₂ , ppm)	С (прирост)
2	0	480	
3	10	610	=B3-B2 → 130
4	20	760	=B4-B3 → 150

Строка	А (время, мин)	В (CO ₂ , ppm)	С (прирост)
5	30	890	=B5-B4 → 130
6	40	1010	=B6-B5 → 120

Формулу в С3 пишут руками только один раз. Потом берут ячейку за маленький квадратик в правом нижнем углу и тянут вниз до С6. Таблица копирует формулу и в каждой строке сдвигает адреса на столько же строк: в С4 окажется =B4-B3, в С6 – =B6-B5. Такие адреса называют **относительными**: для формулы в С3 запись В3 значит «столбец В, моя строка».

Для нескольких чисел сразу есть готовые **функции**. Кусок столбца от В2 до В6 записывают через двоеточие: В2:В6. Это **диапазон** – все ячейки от первой до последней, включая обе.

Таблица 7.2

Формула	Что считает	Результат
=СУММ(В2:В6)	сумму	3750
=СРЗНАЧ(В2:В6)	среднее	750
=МАКС(В2:В6)	наибольшее	1010
=МИН(В2:В6)	наименьшее	480
=МАКС(В2:В6)-МИН(В2:В6)	размах	530
=СЧЁТЕСЛИ(В2:В6;">800")	сколько чисел больше 800	2

Это русские имена функций. В английской версии программы они называются SUM, AVERAGE, MAX, MIN, COUNTIF. Если таблица не узнаёт русское имя, попробуй английское. Условие в СЧЁТЕСЛИ пишут в кавычках: ">800" – больше 800, "≥800" – 800 и больше.

Сколько ячеек в диапазоне? Номер последней строки минус номер первой **плюс один**: в В2:В6 их 6 – 2 + 1 = 5. Забыть «плюс один» – как посчитать столбы в заборе по числу пролётов.

Запомни. Адрес – латинская буква столбца и номер строки. Формула начинается с =. В2:В6 – ячейки с В2 по В6, обе включены, всего 5. Протянул формулу вниз – адреса в ней съехали на столько же строк.

Совет. В русской версии таблицы десятичная дробь пишется через запятую: 36,6. Набери 36.6 с точкой – и таблица может принять число за текст или за дату, а формулы его не посчитают. Узнать такое «число» легко: оно прижимается к левому краю ячейки, а настоящие числа стоят у правого.

Совет. Иногда адрес при протягивании должен стоять на месте – например, когда все строки делят на один и тот же порог из ячейки В8. Тогда пиши =\$B\$8: знак \$ закрепляет букву и номер, и формула в каждой строке смотрит на В8.

7.2. Среднее, наибольшее, наименьшее

НА ПАЛЬЦАХ

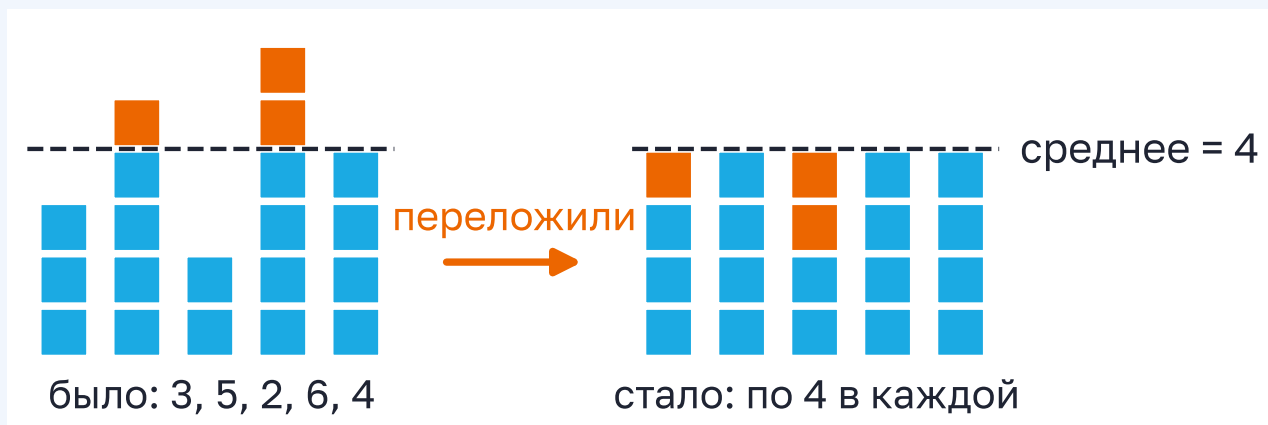


Рисунок 7.2 – Пять стопок кубиков выравнили: лишнее с высоких переложили на низкие

Было 3, 5, 2, 6 и 4 кубика. Выравнили – в каждой стопке по 4. Это и есть среднее.

Среднее арифметическое – число, которое получится у каждого, если всё поделить поровну. Словами: **сложи все числа и раздели сумму на их количество**. Буквами:

$$\text{среднее} = \frac{x_1 + x_2 + \dots + x_n}{n},$$

где $x_1, x_2, \dots$ – сами числа, n – сколько их. Для стопок с картинки:

$$(3 + 5 + 2 + 6 + 4) : 5 = 20 : 5 = 4.$$

Рядом со средним почти всегда спрашивают ещё три числа:

- **максимум** – наибольшее значение, **минимум** – наименьшее;
- **размах** – максимум минус минимум: насколько сильно гуляли числа.

Среднее всегда лежит между минимумом и максимумом. Это бесплатная проверка: вышло среднее меньше самого маленького числа или больше самого большого – ты ошибся в сумме или поделил не на то количество.

У среднего есть слабое место: оно боится **выбросов** – чисел, которые резко отличаются от остальных. Датчик пульса соскочил с пальца и на секунду показал 0. Датчик CO₂ при включении ещё не прогрелся и выдал странное число. Одно такое значение сильно тянет среднее к себе.

Пример. Пять показаний датчика CO₂: 720, 740, 730, 0, 750. Ноль – сбой, в комнате с людьми CO₂ не бывает меньше, чем на улице. Среднее со сбоем:

$$(720 + 740 + 730 + 0 + 750) : 5 = 2940 : 5 = 588 \text{ – меньше любого настоящего показания.}$$

Без сбоя: $2940 : 4 = 735$. Сумма осталась той же, 2940 (ноль к ней ничего не прибавил), зато делим теперь на 4.

На такой случай есть **медиана** – число посередине. Выстрой числа по возрастанию и возьми то, что стоит в центре: 0, 720, **730**, 740, 750 – медиана 730. Если чисел чётное количество, в середине стоят два, и медиана – их среднее. Сбой медиану почти не сдвинул: 730 рядом с честным средним 735.

Когда выброс можно выбрасывать? Когда понятно, что это сбой прибора: пульс 0 у живого человека, пульс 250 у школьника за партой, CO₂ меньше уличного в классе. В задаче такое правило дают прямо: «значения меньше ... считать сбоем». Если такого правила нет, а число просто большое, – это может быть настоящее событие, например, выдох прямо на датчик. Его не выкидывают молча.

Запомни. Среднее = сумма : количество, и оно всегда между минимумом и максимумом. Размах = максимум – минимум. Медиана – середина упорядоченного ряда, её сбой почти не сдвигает. Убрал число из ряда – дели на новое количество.

Совет. Среднее легко проверить в Python – любой онлайн-интерпретатор, ничего ставить не надо.

```
co2 = [720, 740, 730, 0, 750]
print(sum(co2) / len(co2))      # 588.0 - среднее со сбоем
good = [v for v in co2 if v > 0] # оставляем только показания больше нуля
print(sum(good) / len(good))    # 735.0 - среднее без сбоя
print(sorted(co2)[2])           # 730 - медиана: средний из пяти после сортировки
```

len – сколько чисел в списке, sum – их сумма, sorted – список по возрастанию. Счёт мест в Python идёт с нуля, поэтому третье число – это [2] .

7.3. Десятичные дроби и округление

НА ПАЛЬЦАХ



Рисунок 7.3 – Число 0,746 на линейке с делениями по одной тысячной

До отметки 0,75 – четыре деления, до 0,74 – шесть. Округлить до сотых – перейти к ближайшей сотой отметке.

Датчики и секундомеры редко выдают круглые числа. Реакция – 0,287 с, пульс в среднем 76,4 уд/мин, доля душных минут – 33,33... %. Чтобы с такими числами не путаться, вспомним, как устроена десятичная дробь. В числе 0,746 после запятой стоят **десятые** (7), **сотые** (4) и **тысячные** (6). Значит, 0,746 – это 746 тысячных.

Сравнивать десятичные дроби удобно, если уравнивать число знаков после запятой, дописав нули в конце: от этого число не меняется. Что больше, 0,3 или 0,29? Допишем ноль: 0,30 и 0,29 – тридцать сотых больше двадцати девяти. Ловушка тут в том, что «29 длиннее 3», и рука сама ставит 0,29 выше. Сравнивают разряды слева направо, длина записи тут ни при чём.

Перевод миллисекунд в секунды – деление на 1000, то есть запятая уезжает на три знака влево: 287 мс = 0,287 с, 45 мс = 0,045 с (не 0,45!). Обратно – умножение на 1000: 0,3 с = 300 мс.

Округление. «Округли до сотых» – оставь две цифры после запятой. Смотрим на первую цифру, которую отбрасываем: 5 и больше – последнюю оставшуюся увеличиваем на 1, меньше 5 – оставляем как есть. 0,746 → 0,75 (отбросили 6); 0,7438 → 0,74 (отбросили 3); 76,45 → 76,5 до десятых; 0,396 → 0,40 до сотых (девятка превратилась в десятку, и ноль на конце оставляем – он показывает, что точность до сотых).

Округляют **только ответ**. Промежуточные числа держи полностью или хотя бы с запасом в пару знаков. Иначе ошибки копятся: среднее из трёх уже округлённых чисел может отличаться от честного в последней цифре – а на отборе такой ответ будет неверным.

В таблице есть своя ловушка: она может **показывать** меньше знаков, чем **хранит**. Кнопки, которые убирают знаки после запятой, меняют только вид: в ячейке видно 0,3, а внутри лежит 0,308, и все формулы дальше считают с 0,308. Настоящее округление делает функция `=ОКРУГЛ(В7;2)` – число и количество знаков после запятой.

Запомни. Сравнивая дроби, уравнивай число знаков нулями: $0,30 > 0,29$. Миллисекунды → секунды: делим на 1000. Округляют по первой отброшенной цифре (5 и больше – вверх) и только в самом конце.

7.4. Проценты, ppm и пороги: когда проветривать

НА ПАЛЬЦАХ



Рисунок 7.4 – Шкала датчика CO₂: зелёная, оранжевая и красная зоны

Урок начался в зелёной зоне, а к звонку показание уехало в красную. Как у светофора: цвет подсказывает, что делать.

Из главы 1 ты помнишь: в выдохе углекислого газа примерно в 100 раз больше, чем во вдохе. Двадцать пять человек в закрытом кабинете выдыхают его весь урок, и доля CO₂ в воздухе растёт. Открыли окно – уличный воздух её разбавляет. За этим числом и следит датчик CO₂ из модуля 6 программы сферы. Хотя к концу урока в душном классе многие и без датчика чувствуют: клонит в сон.

Датчики показывают долю газа в **ppm** – от английского parts per million, «частей на миллион». 400 ppm значит: из миллиона частичек воздуха 400 – углекислый газ. Проценты – это части на сотню, поэтому

$$1\% = \frac{1\,000\,000}{100} = 10\,000\text{ ppm}.$$

На улице сейчас чуть больше 420 ppm, это 0,042 %, и каждый год понемногу прибавляется. В задачах книги для лёгкого счёта берём круглые **400 ppm** (0,04 %), а если в задаче уличное значение дано – считаем с ним. В выдохе – около 4 %, то есть 40 000 ppm. Из ppm в проценты – делим на 10 000: 1200 ppm = 0,12 %. Зачем вообще придумали ppm, видно сразу: «1200» читать легче, чем «0,0012 от всего воздуха».

Порог – граница, после которой прибор что-то делает: меняет цвет лампочки, пищит, пишет «проветрите». Какие пороги ставить, решают по-разному: в разных приборах и документах границы свои. В этой книге датчик настроен так:

Таблица 7.3

Показание, ppm	Цвет	Что делать
800 и меньше	зелёный	всё хорошо
больше 800, до 1000	оранжевый	скоро проветривать
больше 1000	красный	проветрить сейчас

На отборе бери пороги **из условия**, даже если они другие. И внимательно читай слова: «больше 1000» и «не меньше 1000» расходятся на одно-единственное значение 1000. Если датчик показал 1000, а правило «больше 1000», красный не горит.



Рисунок 7.5 – CO₂ в одном кабинете за четыре урока. Пунктир – порог 1000 ppm

На графике пила: на уроках CO₂ растёт, на переменах с открытым окном падает. На третьей перемене окно не открыли – и четвёртый урок начался сразу у порога, а к концу ушёл к 1500.

Проценты у датчика спрашивают тремя способами, и в каждом делят на своё число.

- **Какая доля измерений выше порога.** Посчитай, сколько показаний больше порога, раздели на число всех показаний и умножь на 100. Из 20 измерений 7 выше 1000 – это $7:20 \cdot 100 = 35\%$. В таблице: `=СЧЁТЕСЛИ(B2:B21;">1000")/20*100`.
- **На сколько процентов показание выше порога.** Проценты считаем **от порога**: $(1250 - 1000):1000 \cdot 100 = 25\%$.
- **На сколько процентов выросло за урок.** Проценты от того, что было **в начале**: с 500 до 1200 ppm – $(1200 - 500):500 \cdot 100 = 140\%$. Больше ста процентов значит, что величина выросла больше чем вдвое.

А «во сколько раз» – просто деление: 1200 ppm в классе против 400 на улице – в $1200:400 = 3$ раза больше.

Запомни. 1 % = 10 000 ppm. На улице чуть больше 400 ppm, в задачах берут 400. Доля измерений выше порога = (сколько выше : сколько всего) · 100. «На сколько процентов больше» считают от того, с чем сравнивают: от порога, от начального значения.

Совет. Перед подсчётом «сколько выше порога» реши, куда попадает само пороговое значение. Подчеркни в условии слова «больше», «не меньше», «не больше» – они решают судьбу числа, равного порогу.

7.5. Графики: построить и прочитать

НА ПАЛЬЦАХ

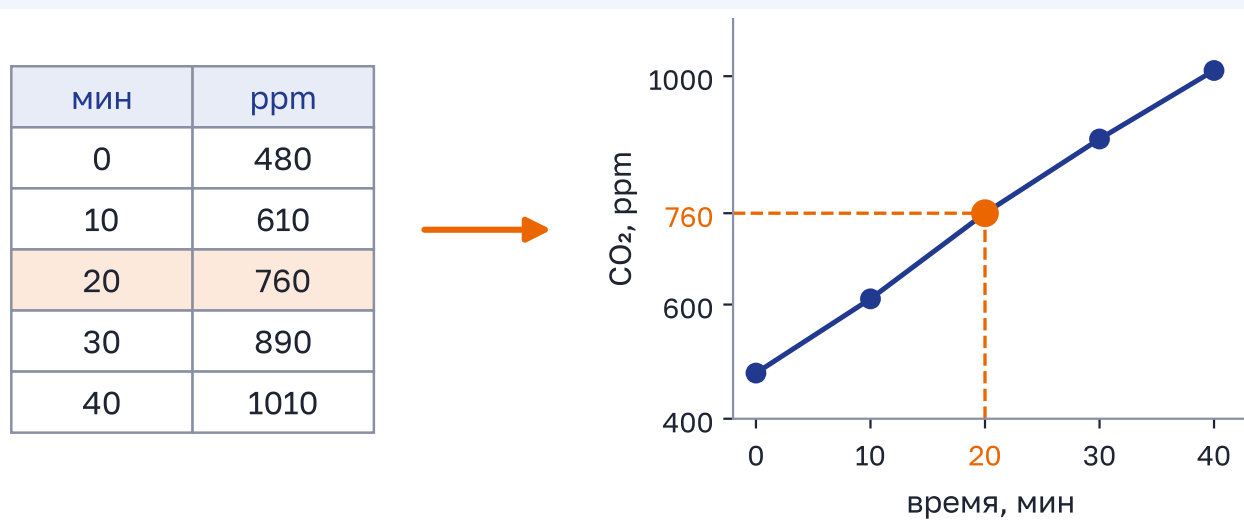


Рисунок 7.6 – Та же таблица из раздела 7.1, нарисованная точками

Каждая строка таблицы – одна точка: время отложили вправо, показание – вверх.
Соединили точки – получился график.

График – рисунок, на котором видно, как одна величина меняется вместе с другой. Время почти всегда откладывают по горизонтальной оси, а по вертикальной – то, что измеряли. У каждой оси должны быть подпись и единица: «время, мин», «CO₂, ppm». Без единиц график – просто красивая линия.

Построить график по таблице на бумаге несложно.

1. Реши, что по какой оси. Время – по горизонтали.
2. Выбери **масштаб**: сколько единиц в одной клетке. В таблице из раздела 7.1 время от 0 до 40 мин – удобно 5 мин в клетке, 8 клеток. Показания от 480 до 1010 – удобно начать ось с 400 и брать 100 ppm в клетке.
3. Для каждой строки таблицы найди точку: вправо на время, вверх на показание.
4. Соедини точки по порядку.

В электронной таблице то же делается мышкой: выдели два столбца с числами и вставь диаграмму из меню «Вставка». Для изменений во времени выбирай тип «линия» (в Excel он называется «График»). Для сравнения нескольких отдельных значений – столбики.

Читать график учатся по тем же шагам, только наоборот.

- **Значение в момент времени.** Найди время на горизонтальной оси, поднимись до линии и посмотри влево, на вертикальную ось. На картинке «на пальцах»: 20 мин → 760 ppm.
- **Когда достигли значения.** Найди значение на вертикальной оси, пройди вправо до линии и опусти вниз.
- **Цена деления.** Если подписана не каждая линия сетки, сначала найди, сколько стоит одна клетка: разница соседних подписей, делённая на число клеток между ними. Подписи 400 и 600, между ними две клетки – клетка стоит 100.

- **Как быстро растёт.** Скорость изменения – на сколько изменилось, делённое на время: $(1010 - 480) : 40 = 530 : 40 = 13,25$ ppm в минуту. Чем круче линия идёт вверх, тем быстрее рост. Линия вниз – величина уменьшается, горизонтальная – не меняется.
- **Между точками.** Если датчик мерил раз в 10 минут, а спрашивают про 15-ю минуту, берут точку на отрезке между соседними измерениями. Это оценка: что происходило между замерами, датчик не видел.

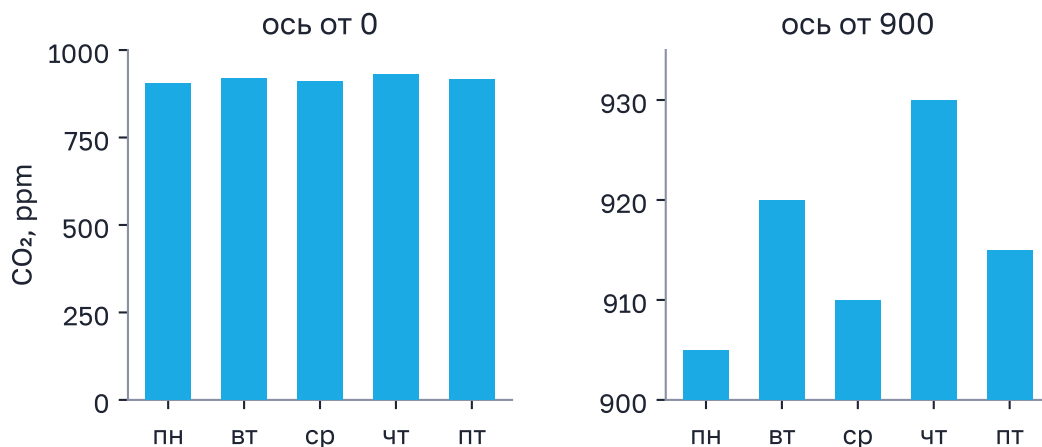


Рисунок 7.7 – Одни и те же пять чисел на двух столбчатых диаграммах. Слева ось начинается с нуля, справа – с 900

Главная ловушка графиков – **масштаб**. На правой диаграмме столбик четверга вшестеро выше понедельника, а по числам разница всего $930 - 905 = 25$ ppm, меньше 3 %. Прежде чем делать вывод по высоте столбиков или крутизне линии, посмотри, с какого числа начинается ось и сколько стоит клетка.

Запомни. Сначала оси: что отложено, в каких единицах, сколько стоит клетка и с какого числа начинается ось. Потом уже линия. Скорость изменения = изменение : время.

7.6. Сравниваем группы и выбираем верный вывод

НА ПАЛЬЦАХ

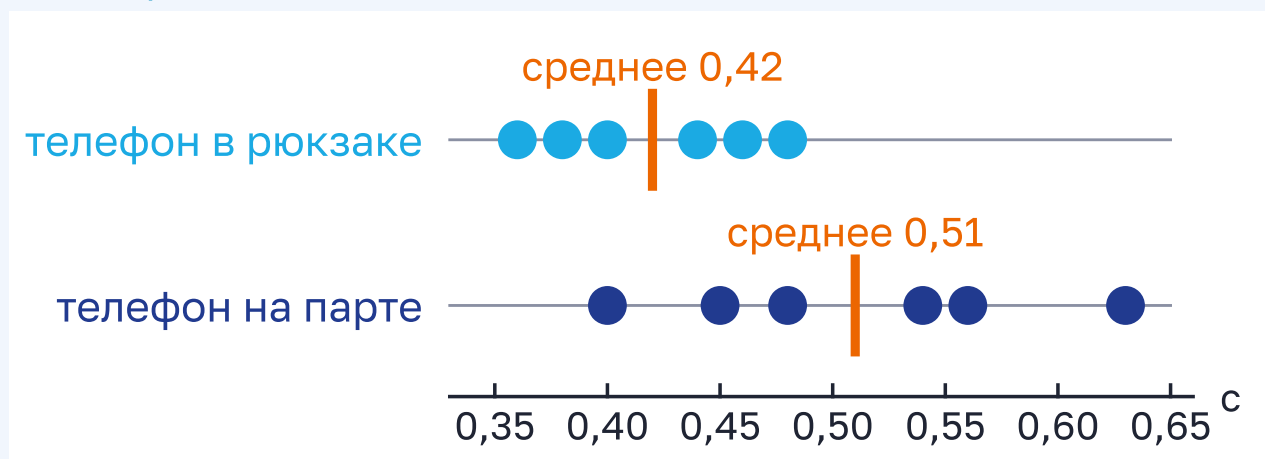


Рисунок 7.8 – Время реакции двух групп по шесть человек. Оранжевая черта – среднее группы

Средние разные, но точки перемешаны: самый быстрый из группы «на парте» (0,40 с) обогнал половину группы «в рюкзаке».

По одному человеку исследователи выводов не делают. Берут две группы – с уведомлениями на телефоне и без, до зарядки и после, класс с открытым окном и с закрытым – и смотрят на одни и те же числа.

- **Средние:** насколько одно больше другого в единицах ($0,51 - 0,42 = 0,09$ с) и в процентах. Проценты считают от той группы, **с которой** сравнивают: «во второй группе реакция дольше на $0,09 : 0,42 \cdot 100 = 21,4$ %».
- **Разброс:** минимум, максимум, размах. Если размах в группах большой, а разница средних маленькая, вывод «группы отличаются» получается шатким.
- **Размер групп:** вывод по трём людям весит меньше, чем по тридцати.

Когда группы разного размера, **среднее двух групп вместе не равно среднему двух средних**. Пример: в 6 «А» 25 человек, средний пульс 84; в 6 «Б» 20 человек, средний пульс 80. Вместе: сумма всех пульсов $25 \cdot 84 + 20 \cdot 80 = 2100 + 1600 = 3700$, людей 45, среднее $3700 : 45 = 82,2$ уд/мин. А $(84 + 80) : 2 = 82$ – неверно, потому что 6 «А» больше и тянет общее среднее к себе. Выручает сумма: среднее, умноженное на количество, даёт сумму всех чисел группы. Суммы групп складываем и делим на общее количество.

Тот же приём решает обратную задачу: каким должно быть недостающее число, чтобы среднее стало нужным. Нужно среднее умножаем на общее количество – получаем, какой должна быть сумма. Вычитаем то, что уже набрано, – остаток и есть недостающее число.

На отборе к таблице часто прикладывают утверждения: «выбери верные». Каждое проверяй отдельно, по числам, а не по ощущению. Неверные утверждения ломаются на одних и тех же местах:

- **«в среднем» превратили в «каждый»:** «в группе Б все реагировали медленнее» – а среднее этого не говорит;
- **проценты от другого числа:** разницу поделили не на то, с чем сравнивают;

- **вывод шире данных:** меряли реакцию, а в утверждении – «уведомления портят зрение»;
- **причина, которой не проверяли:** группы отличались ещё чем-то, кроме телефона.

Запомни. Общее среднее нескольких групп – через суммы: среднее · количество, суммы сложить, поделить на общее количество. Верный вывод опирается только на данные из таблицы и не говорит «каждый», если в таблице «в среднем».

Совет. В задаче «выбери верные» выпиши номера утверждений столбиком и против каждого поставь «да» или «нет» с коротким расчётом. Если в условии не сказано иначе, ответ пишут цифрами верных утверждений по возрастанию, без пробелов: 136.

7.7. Разбираем задачи

Задача 7.1 Датчик на уроке математики

разминка · электронная таблица, СРЗНАЧ, МАКС, МИН, размах

Датчик CO₂ записывал показания каждые 10 минут, и Миша перенёс их в таблицу: в ячейках B2:B6 стоят числа 620, 780, 930, 1060, 1150 (ppm). Что покажет ячейка B7 с формулой =СРЗНАЧ(B2:B6) ? А ячейка B8 с формулой =МАКС(B2:B6)-МИН(B2:B6) ?

Дано: B2 = 620, B3 = 780, B4 = 930, B5 = 1060, B6 = 1150.

Найти: значения формул в B7 и B8 – ?

Решение

Шаг 1. Что входит в диапазон. B2:B6 – ячейки B2, B3, B4, B5, B6. Их $6 - 2 + 1 = 5$, это все пять показаний.

Шаг 2. Сумма. $620 + 780 + 930 + 1060 + 1150 = 4540$ ppm.

Шаг 3. Среднее. $4540 : 5 = 908$ ppm – между самым маленьким показанием 620 и самым большим 1150.

Шаг 4. Формула в B8. Сначала таблица найдёт МАКС – 1150, потом МИН – 620, потом вычитет: $1150 - 620 = 530$ ppm. Это размах.

Среднее 908 попало в оранжевую зону (раздел 7.4), а максимум уже в красной: к концу урока класс стоило проветрить.

⚠ Типичная ошибка. Поделить сумму на 4, решив, что B2:B6 – это «от 2 до 6, четыре шага». Получится 1135 – почти максимум, и проверка «среднее между минимумом и максимумом» формально пройдёт. Поэтому считай ячейки, а не шаги.

Ответ: B7 – 908, B8 – 530.

Своя задача по программе сферы, модуль 6

Реши сам

7.1. Шестеро ребят из кружка измерили пульс в покое и записали в ячейки C2:C7: 78, 85, 92, 74, 88, 81 уд/мин. Что покажут формулы `=СРЗНАЧ(C2:C7)` и `=МАКС(C2:C7)-МИН(C2:C7)` ?

7.2. В ячейке C3 записана формула `=B3-B2`. Её протянули вниз до C8. Какая формула окажется в ячейке C6? Сколько ячеек в диапазоне `B2:B31` ?

Задача 7.2 Палец соскочил

среднее, выброс, медиана

Браслет каждые 10 секунд показывал пульс Ани, пока она сидела за партой: 78, 80, 0, 79, 83, 215, 81, 79 уд/мин. Ноль – палец соскочил с датчика, 215 – Аня дёрнула рукой. а) Каким получится среднее всех восьми чисел (до целых)? б) Каким будет среднее, если выбросить сбои – значения меньше 40 и больше 200 уд/мин? в) Найди медиану всех восьми чисел.

Дано: 78, 80, 0, 79, 83, 215, 81, 79; сбой – меньше 40 или больше 200.

Найти: среднее всех, среднее без сбоев, медиану – ?

Решение

Шаг 1. Сумма всех восьми. $78 + 80 + 0 + 79 + 83 + 215 + 81 + 79 = 695$.

Шаг 2. Среднее всех. $695 : 8 = 86,875 \approx 87$ уд/мин. Первая отброшенная цифра 8 – округляем вверх.

Шаг 3. Какие числа – сбои. Меньше 40 – только 0. Больше 200 – только 215. Остаются шесть чисел: 78, 80, 79, 83, 81, 79.

Шаг 4. Среднее без сбоев. Сумма $695 - 0 - 215 = 480$. Делим на **шесть**: $480 : 6 = 80$ уд/мин.

Шаг 5. Медиана всех восьми. По возрастанию: 0, 78, 79, **79, 80**, 81, 83, 215. Чисел восемь – в середине два, четвёртое и пятое: 79 и 80. Медиана $(79 + 80) : 2 = 79,5$ уд/мин.

Сравни три ответа. Сбои подняли среднее на 7 ударов, а медиана осталась в полушаге от честного значения 80. Для школьника за партой 80 ударов в минуту – обычный пульс, а 87 было бы завышено из-за одного рывка рукой.

⚠ Типичная ошибка. Выбросить сбои, а делить по-прежнему на 8: $480 : 8 = 60$. Получается пульс ниже всех настоящих показаний – верный признак, что поделили не на то количество. Вторая – забыть отсортировать числа перед поиском медианы и взять «средние по месту» 79 и 83.

Ответ: а) 87 уд/мин; б) 80 уд/мин; в) 79,5 уд/мин.

Своя задача по программе сферы, модуль 6

Реши сам

7.3. Датчик CO₂ выдал пять показаний: 910, 940, 0, 980, 1010 ppm. Ноль – сбой при переподключении провода. Найди среднее без сбоя и медиану всех пяти чисел.

7.4. Среднее пяти измерений пульса было 84 уд/мин. После шестого измерения среднее стало 86 уд/мин. Каким было шестое измерение?

Задача 7.3 Сколько времени было душно

порог, доля в процентах, СЧЁТЕСЛИ

Датчик CO₂ в кабинете информатики мерил воздух раз в 5 минут, 12 раз подряд: 820, 890, 960, 1030, 1090, 1140, 760, 690, 780, 870, 950, 1000 ppm. Красная лампочка горит, когда показание **больше** 1000 ppm. а) Какой процент измерений пришёлся на красную зону? б) На сколько процентов самое большое показание выше порога? Ответы округли до целых.

Дано: 12 показаний; красный – больше 1000 ppm.

Найти: доля красных измерений, %; превышение максимума над порогом, % – ?

Решение

Шаг 1. Какие показания больше 1000. Идём по списку: 1030 – да, 1090 – да, 1140 – да. Последнее, 1000, – **нет**: оно равно порогу, а не больше него. Красных показаний 3.

Шаг 2. Доля в процентах. $3:12 \cdot 100 = 25\%$. В таблице то же самое дала бы формула `=СЧЁТЕСЛИ(B2:B13;">1000")/12*100`.

Шаг 3. Максимум. Самое большое число в списке – 1140 ppm.

Шаг 4. На сколько процентов выше порога. Сравниваем с порогом, значит, делим на порог: $(1140 - 1000):1000 \cdot 100 = 140:1000 \cdot 100 = 14\%$.

После 1140 показание упало до 760 – на перемене открыли окно. Так что четверть времени в красной зоне – это как раз кусок урока до проветривания.

⚠ Типичная ошибка. Засчитать последнее показание 1000 в красную зону: получится 4 из 12, то есть 33 %. На отборе такая разница – неверный ответ. Во втором пункте ошибаются, деля на 1140 вместо 1000: $140:1140 \cdot 100 = 12\%$. Проценты превышения считают от порога.

Ответ: а) 25 %; б) 14 %.

Своя задача по программе сферы, модуль 6

Реши сам

7.5. В спортзале датчик сделал 10 измерений: 610, 680, 750, 820, 870, 800, 910, 640, 700, 790 ppm. Оранжевая лампочка загорается, когда показание **больше** 800 ppm. Какой процент измерений был больше 800?

7.6. В конце урока датчик показал 1350 ppm. Запиши это значение в процентах. Во сколько раз это больше, чем на улице (возьми 400 ppm), до десятых? На сколько процентов это выше порога 1000 ppm?

Задача 7.4 Когда открыли окно

чтение графика, скорость изменения, прогноз

На графике – показания датчика CO₂ за один урок. В какой-то момент учитель открыл окно, а потом закрыл. а) На какой минуте открыли окно и сколько показывал датчик в этот момент? б) С какой скоростью (ppm в минуту) росло CO₂ до этого? в) Сколько показал бы датчик в конце урока, на 45-й минуте, если бы окно не открывали и рост шёл с той же скоростью?

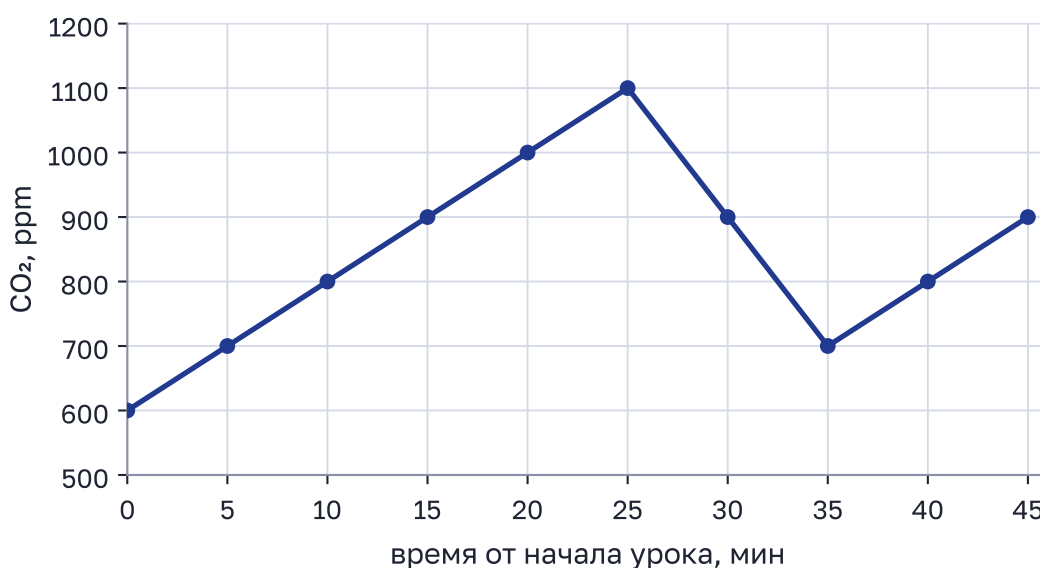


Рисунок 7.9 – Датчик CO₂ за урок. Точки – измерения через каждые 5 минут

Дано: график; измерения через 5 мин; урок 45 мин.

Найти: момент открытия окна и показание; скорость роста; прогноз на 45-ю минуту – ?

Решение

Шаг 1. Цена деления. По горизонтали подписи 0, 5, 10... – одна клетка 5 мин. По вертикали 500, 600, 700... – одна клетка 100 ppm.

Шаг 2. Когда открыли окно. Пока окно закрыто, CO₂ растёт, линия идёт вверх. Открыли – линия повернула вниз. Самая высокая точка стоит над отметкой 25 мин, на уровне 1100 ppm.

Шаг 3. Скорость роста. За первые 25 минут CO₂ выросло с 600 до 1100 ppm, то есть на $1100 - 600 = 500$ ppm. Скорость: $500 : 25 = 20$ ppm в минуту. Проверь по одной клетке: за 5 минут точки поднимаются на одну клетку, 100 ppm, и $100 : 5 = 20$.

Шаг 4. Прогноз. Без окна рост шёл бы все 45 минут: $20 \cdot 45 = 900$ ppm прибавки. Прибавляем к тому, что было в начале: $600 + 900 = 1500$ ppm.

После того как окно закрыли (35-я минута), линия снова поднимается на клетку за 5 минут – те же 20 ppm в минуту. Класс тот же, люди дышат так же, и скорость роста та же.

❗ **Типичная ошибка.** В прогнозе прибавить рост не к началу урока, а к последней точке графика: $900 + 900 = 1800$ ppm. Прогноз строят от момента, с которого шёл бы непрерывный рост, – от 600 ppm на нулевой минуте. Вторая – считать клетки по вертикали за 50 ppm, если не посмотрел на подписи.

Ответ: а) на 25-й минуте, 1100 ppm; б) 20 ppm в минуту; в) 1500 ppm.

Своя задача по программе сферы, модуль 6

Реши сам

7.7. По тому же графику из задачи 7.4: с какой скоростью (ppm в минуту) падало CO₂, пока окно было открыто? Сколько показывал датчик на 40-й минуте?

7.8. В спортзале в 10:00 датчик показал 500 ppm, в 10:20 – 900 ppm. CO₂ растёт равномерно. Во сколько показание дойдёт до 1100 ppm, если не открывать окна?

Задача 7.5 Игра на реакцию

десятичные дроби, мс и с, среднее, сравнение

На кружке собрали игру: загорается светодиод – жми кнопку, Arduino пишет, сколько прошло времени. Лиза играла на плате, которая выводит время в миллисекундах: 312, 287, 305, 296, 318. Костя – на плате, которая выводит секунды: 0,29; 0,315; 0,3; 0,28; 0,325. а) У кого лучшая (самая быстрая) попытка и сколько в ней секунд? б) У кого среднее время меньше и на сколько миллисекунд?

Дано: Лиза: 312, 287, 305, 296, 318 мс; Костя: 0,29; 0,315; 0,3; 0,28; 0,325 с.

Найти: лучшая попытка; разница средних, мс – ?

Решение

Шаг 1. Единицы. Сравнить можно только в одних единицах. Переведём Лизу в секунды, поделив на 1000: 0,312; 0,287; 0,305; 0,296; 0,318 с.

Шаг 2. Лучшая попытка. В игре на реакцию лучшая попытка – с **наименьшим** временем. У Лизы минимум 0,287 с. У Кости сравним, уравнивая знаки нулями: 0,290; 0,315; 0,300; 0,280; 0,325 – минимум 0,280. Теперь 0,280 и 0,287: сотые одинаковые (8), в тысячных 0 меньше 7. Быстрее Костя, 0,28 с.

Шаг 3. Среднее Лизы. $312 + 287 + 305 + 296 + 318 = 1518$ мс, $1518 : 5 = 303,6$ мс.

Шаг 4. Среднее Кости. $0,29 + 0,315 + 0,3 + 0,28 + 0,325 = 1,51$ с, $1,51 : 5 = 0,302$ с. В миллисекундах: $0,302 \cdot 1000 = 302$ мс.

Шаг 5. Разница. $303,6 - 302 = 1,6$ мс. Среднее меньше у Кости.

Разница меньше двух тысячных секунды – считай, ничья: так точно кнопку не нажимает никто. А сами результаты около 0,3 с – обычное время реакции человека на вспышку.

❗ **Типичная ошибка.** Посчитать лучшей попыткой самое большое число – 318 мс у Лизы и 0,325 с у Кости. В этой игре чем больше, тем хуже. Вторая ловушка – решить, что 0,3 меньше 0,29 («три меньше двадцати девяти»). Допиши ноль: 0,30 больше 0,29.

Ответ: а) у Кости, 0,28 с; б) у Кости, на 1,6 мс.

Своя задача по программе сферы, модуль 6

Реши сам

7.9. Пять попыток в игре на реакцию: 1) 0,4 с; 2) 0,35 с; 3) 0,305 с; 4) 0,53 с; 5) 0,3 с. Запиши номера попыток от самой быстрой к самой медленной, подряд без пробелов. Найди среднее время до сотых секунды.

7.10. В ячейках В2:В6 времена реакции: 0,34; 0,27; 0,31; 0,29; 0,33 с. В В7 формула =СРЗНАЧ(В2:В6) , и таблица показывает в ней 0,3 – так настроен вид ячейки. Какое число хранится в В7? Что покажет формула =ОКРУГЛ(В7;2) ?

Задача 7.6 Телефон на парте

сравнение групп, проценты, выбор верных выводов

Шестиклассники проверяли, мешают ли уведомления телефона вниманию. Два класса по 12 человек смотрели один и тот же мультфильм, в котором 25 раз на полсекунды появлялась звёздочка, – её нужно было заметить и нажать кнопку. После просмотра все отвечали на 10 вопросов по сюжету. В группе А телефоны лежали в рюкзаках, в группе Б – на партах, и на них приходили сообщения. Средние по группам:

Таблица 7.4

Группа	Заметили звёздочек (из 25)	Время реакции, с	Верных ответов (из 10)
А (телефон в рюкзаке)	21,5	0,42	8,2
Б (телефон на парте)	16	0,51	6,1

Какие утверждения верны? В ответ запиши их номера по возрастанию без пробелов.

1. В группе А в среднем замечали 86 % звёздочек.
2. В группе Б каждый заметил 16 звёздочек.
3. С уведомлениями время реакции в среднем больше на 90 мс.
4. В группе Б время реакции больше, чем в группе А, примерно на 21 %.
5. Опыт показал, что уведомления портят зрение.
6. В группе Б в среднем на 2,1 верного ответа меньше, чем в группе А.

Дано: таблица средних; 25 звёздочек; 10 вопросов.

Найти: номера верных утверждений – ?

Решение

Шаг 1. Утверждение 1. Доля от 25: $21,5 : 25 \cdot 100 = 86 \%$. Верно.

Шаг 2. Утверждение 2. В таблице **средние**. Среднее 16 получится и тогда, когда один заметил 10, а другой 22. Про каждого таблица ничего не говорит. Неверно.

Шаг 3. Утверждение 3. $0,51 - 0,42 = 0,09$ с. В миллисекундах: $0,09 \cdot 1000 = 90$ мс. Верно.

Шаг 4. Утверждение 4. Сравниваем Б с А, значит, делим на время группы А: $0,09 : 0,42 \cdot 100 = 21,4 \%$. «Примерно 21 %» – верно.

Шаг 5. Утверждение 5. Зрение в опыте не проверяли. Меряли, сколько звёздочек заметили и как быстро нажали, – это внимание, а не зрение. Вывод шире данных. Неверно.

Шаг 6. Утверждение 6. $8,2 - 6,1 = 2,1$. Верно.

Верные – 1, 3, 4, 6.

 **Типичная ошибка.** В утверждении 4 поделить разницу на время группы Б: $0,09 : 0,51 \cdot 100 = 17,6 \%$ – и вычеркнуть верное утверждение. «Больше, чем в группе А» – значит, группа А и есть то, с чем сравнивают. И ещё одна: поверить утверждению 5, потому что оно звучит убедительно. В таблице про зрение нет ни одного числа.

Ответ: 1346.

По мотивам задачи заключительного этапа НТО «Нейротехнологии и когнитивные науки» 2025/26, предметный тур по биологии, «Невидимый смартфон»

Реши сам

7.11. В конце третьего урока датчики в четырёх кабинетах показали: кабинет 12 – 1240 ppm, кабинет 15 – 780 ppm, кабинет 21 – 1010 ppm, спортзал – 950 ppm. Пороги как в разделе 7.4: красный – больше 1000 ppm; на улице считай 400 ppm. Какие утверждения верны? Запиши номера по возрастанию без пробелов. 1) Красный горит в двух помещениях. 2) В кабинете 15 пора проветривать. 3) В кабинете 21 показание выше порога на 1 %. 4) Среднее по четырём помещениям больше 1000 ppm. 5) В кабинете 12 углекислого газа больше, чем на улице, более чем в 3 раза.

7.12. На физкультуре пульс мерили после разминки. В 7 «А» 30 человек, средний пульс 88 уд/мин. В 7 «Б» 20 человек, средний пульс 80 уд/мин. Какие утверждения верны? Запиши номера по возрастанию без пробелов. 1) В 7 «А» у каждого пульс выше, чем у любого в 7 «Б». 2) Средний пульс 7 «А» на 10 % выше, чем у 7 «Б». 3) Средний пульс 7 «Б» на 10 % ниже, чем у 7 «А». 4) Средний пульс обоих классов вместе больше 84 уд/мин. 5) Сумма пульсов всех учеников 7 «А» – 2640.

Задача 7.7* *. Средний воздух за день

среднее по группам разного размера, обратная задача на среднее

Датчик в кабинете биологии мерит CO₂ раз в 5 минут и считает средние. За первый урок было 9 измерений, среднее – 880 ppm. За второй урок 9 измерений, среднее – 1030 ppm. На большой перемене окно открыли, 4 измерения, среднее – 680 ppm. а) Найди среднее по всем этим измерениям. б) Впереди третий урок, на нём будет ещё 9 измерений. Каким наибольшим может быть среднее за третий урок, чтобы среднее за весь день (все измерения) не превысило 950 ppm?

Дано: урок 1: 9 измерений, 880 ppm; урок 2: 9 измерений, 1030 ppm; перемена: 4 измерения, 680 ppm; урок 3: 9 измерений; среднее за день – не больше 950 ppm.

Найти: среднее по уроку 1, уроку 2 и перемене; наибольшее среднее за урок 3 – ?

Решение

Шаг 1. Почему нельзя взять среднее средних. Кусочки разные: на уроках по 9 измерений, на перемене 4. Каждое измерение должно весить одинаково, поэтому идём через суммы (раздел 7.6).

Шаг 2. Суммы по кусочкам. Среднее, умноженное на количество, – это сумма:

- урок 1: $880 \cdot 9 = 7920$;
- урок 2: $1030 \cdot 9 = 9270$;
- перемена: $680 \cdot 4 = 2720$.

Шаг 3. Общее среднее. Сумма $7920 + 9270 + 2720 = 19\,910$, измерений $9 + 9 + 4 = 22$. Среднее: $19\,910 : 22 = 905$ ppm. Это число между 680 и 1030 и ближе к урокам: там измерений больше.

Шаг 4. Сколько измерений будет за день. $22 + 9 = 31$.

Шаг 5. Какая сумма допустима. Среднее за день не больше 950, значит, сумма всех 31 измерений не больше $950 \cdot 31 = 29\,450$.

Шаг 6. Сколько остаётся на третий урок. $29\,450 - 19\,910 = 9540$ – столько в сумме могут дать 9 измерений урока 3.

Шаг 7. Наибольшее среднее урока 3. $9540 : 9 = 1060$ ppm.

Проверим обратно: $(19\,910 + 1060 \cdot 9) : 31 = (19\,910 + 9540) : 31 = 29\,450 : 31 = 950$. Выходит, весь третий урок может пройти в красной зоне, а среднее за день всё равно уложится в 950. Среднее прячет плохие куски – поэтому лампочка на датчике смотрит на каждое показание, а не на среднее за день.

⚠ Типичная ошибка. Среднее средних: $(880 + 1030 + 680) : 3 = 863,3$ ppm. Перемена с четырьмя измерениями тут весит столько же, сколько урок с девятью, и среднее выходит заниженным. Во второй части так же ошибаются: решают $(880 + 1030 + 680 + x) : 4 = 950$ и получают $x = 1210$ – на 150 ppm больше верного ответа.

Ответ: а) 905 ppm; б) 1060 ppm.

Своя задача по программе сферы, модуль 6

Реши сам

7.13.* В кружке 12 ребят, их средний пульс в покое – 82 уд/мин. Пришли трое новеньких, у них средний пульс 76 уд/мин. Каким стал средний пульс всего кружка (до десятых)?

7.14.* За урок датчик сделает 9 измерений CO_2 . Первые 8 уже сделаны, их среднее – 870 ppm. Каким может быть самое большое девятое показание, чтобы среднее за урок не превысило 900 ppm?

Свет, звук и колебания в медицине



Врач прикладывает к груди стетоскоп, медсестра водит по животу датчиком УЗИ, браслет на руке мигает зелёным огоньком – и никто из них не делает ни одного укола. Все трое работают на колебаниях: звуке, ультразвуке и свете. А в конце главы – рычаги: почему бицепс тянет с силой в сотни ньютонов, когда в руке лёгкая гантель, и как сервопривод сгибает палец бионической руки. На отборе такая задача – короткая история с прибором или с телом, а в ответ вводится одно число.

8.1. Колебания: период, частота, амплитуда

НА ПАЛЬЦАХ

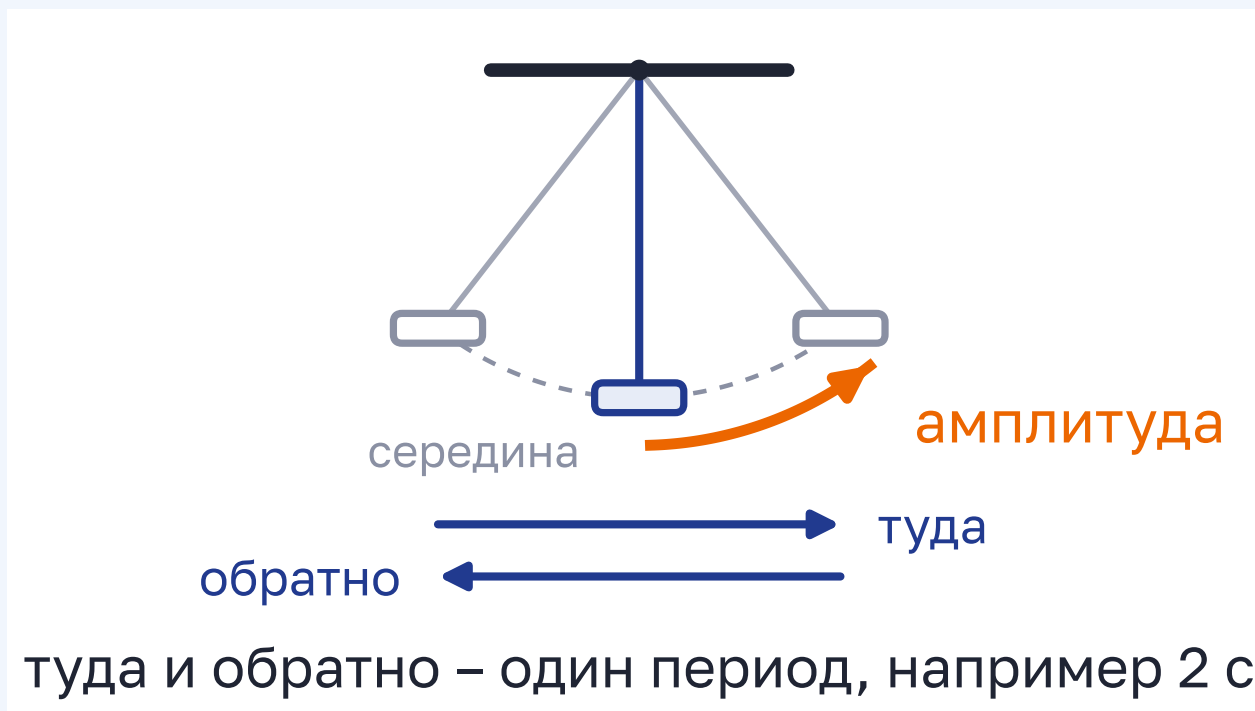


Рисунок 8.1 – Качели: взмах туда и обратно – один период, самое дальнее отклонение от середины – амплитуда

Качели раз за разом возвращаются в ту же точку за одно и то же время. Всё, что так качается, называют колебанием.

Колебание – движение, которое повторяется снова и снова: туда – обратно, вверх – вниз. Качели во дворе, маятник в часах, струна гитары, стенка артерии на запястье, крылышки комара. Звук и свет – тоже колебания, про них следующие разделы. У любого колебания меряют одни и те же числа.

- **Период T** – время одного полного колебания. Качели ушли вперёд, вернулись назад и оказались там же, откуда начали, – прошёл один период.
- **Частота f** – сколько полных колебаний укладывается в 1 секунду. Единица – **герц (Гц)**: 1 Гц – одно колебание в секунду.
- **Амплитуда A** – на сколько тело отходит от середины (от положения покоя) в самой дальней точке. У качелей её можно мерить в сантиметрах или в градусах, у сигнала на графике – в единицах этого графика.

Период и частоту мы уже встречали у пульса (глава 2) и у сигналов с датчиков (глава 4). Связь между ними такая: если одно колебание длится T секунд, то в секунду их уложится $1:T$.

$$f = \frac{1}{T}, \quad T = \frac{1}{f}.$$

Качели делают взмах туда-обратно за 2 с – частота $1:2 = 0,5$ Гц. Сердце в покое бьётся примерно раз в секунду – около 1 Гц.

В медицинских приборах колебания бывают куда быстрее: тысячи и миллионы в секунду. Чтобы не писать хвосты из нулей, к единицам добавляют приставки:

- **кило-** (к) – тысяча: $1 \text{ кГц} = 1000 \text{ Гц}$;
- **мега-** (М) – миллион: $1 \text{ МГц} = 1\,000\,000 \text{ Гц}$;
- **милли-** (м) – тысячная доля: $1 \text{ мс} = 0,001 \text{ с}$;
- **микро-** (мк) – миллионная доля: $1 \text{ мкс} = 0,000\,001 \text{ с}$. За одну микросекунду даже свет успевает пролететь всего 300 м.

Эти приставки ходят парами. Частота 4 кГц – это 4 колебания за каждую миллисекунду, значит, одно колебание длится $1 : 4 = 0,25 \text{ мс}$. Частота 5 МГц – 5 колебаний за каждую микросекунду, период $1 : 5 = 0,2 \text{ мкс}$. Получается удобное правило: частоту в **килогерцах** переворачиваешь – получаешь период в **миллисекундах**; частоту в **мегагерцах** переворачиваешь – получаешь период в **микросекундах**.

Таблица 8.1

Что колеблется	Частота	Период
Сердце в покое	около 1 Гц	около 1 с
Альфа-ритм мозга	10 Гц	0,1 с
Камертон невролога	128 Гц	около 7,8 мс
Писк прибора	2 кГц	0,5 мс
Ультразвуковой дальномер	40 кГц	25 мкс
Датчик УЗИ	5 МГц	0,2 мкс

Сколько колебаний пройдёт за время t ? Частота – это колебания за одну секунду, значит, за t секунд их будет в t раз больше: $N = f \cdot t$. Камертон на 128 Гц за 2 с качнётся $128 \cdot 2 = 256$ раз.

Если качели больше не подталкивать, они понемногу успокаиваются: размах всё меньше и меньше. Такое угасание называют **затуханием**. Меняется при этом амплитуда, а период остаётся почти прежним – поэтому затихающий камертон звучит всё тише, но на той же ноте.

Запомни. $f = 1 : T$, $T = 1 : f$, число колебаний $N = f \cdot t$. Килогерцы в паре с миллисекундами, мегагерцы – с микросекундами: 4 кГц – период 0,25 мс, 5 МГц – 0,2 мкс.

Совет. Ответ «0,000 000 2 с» верный, но в нём легко потерять нолик при вводе. Если в условии не сказано, в чём отвечать, переведи в микросекунды или миллисекунды – чем меньше нулей, тем меньше шансов промахнуться.

8.2. Звук: колебания, которые бегут

НА ПАЛЬЦАХ



Рисунок 8.2 – Мембрана динамика толкает воздух. Сгущения бегут к уху, а каждая частичка воздуха только качается на месте

Как волна на трибуне стадиона: болельщики только встают и садятся, а волна обегает весь стадион.

Стукни по столу – стол задрожит и толкнёт воздух рядом с собой. Этот воздух толкнёт соседний, тот – следующий, и толчки побегут во все стороны. Сами частички воздуха далеко не улетают: качнулись туда-сюда и остались где были. Бежит только толчок. Добежит он до уха – начнёт качаться **барабанная перепонка**, тонкая плёнка внутри уха, и ты услышишь стук. Колебание, которое бежит по воздуху, по воде или по телу, называется **волной**. Звук – волна из сгущений и разрежений воздуха.

Два свойства звука слышны сразу.

- **Высота** зависит от частоты. Толстая струна гитары качается медленно и гудит низко, тонкая – быстро и звенит высоко.
- **Громкость** зависит от амплитуды. Чем сильнее качается воздух, тем громче звук.

Человек слышит звуки примерно от 20 до 20 000 Гц, а с возрастом верхняя граница опускается. Колебания реже 20 Гц называют **инфразвуком**, чаще 20 000 Гц – **ультразвуком**. Ультразвук человеку не слышен. Летучие мыши им ориентируются в темноте: кричат и слушают эхо.

Скорость звука – сколько метров пробегает толчок за секунду. Она зависит от того, по чему бежит звук:

Таблица 8.2

По чему бежит звук	Скорость, примерно
Воздух	340 м/с
Вода	1500 м/с
Мягкие ткани тела	1540 м/с
Кость	3500–4000 м/с

«Мягкие ткани» – это мышцы, жир, печень, кровь. В жире звук бежит чуть медленнее, в мышцах чуть быстрее, но разница небольшая, и аппарат УЗИ считает для всех одну среднюю скорость – 1540 м/с. Поэтому глубина на его экране – очень точная прикидка, но не мерка линейкой.

Звук бежит с постоянной скоростью, значит, путь считается как у автобуса: **путь равен скорости, умноженной на время**, $s = v \cdot t$. Над Обским морем сверкнула молния, гром пришёл через 3 с. Свет вспышки долетает почти мгновенно, а звук шёл 3 с: $340 \cdot 3 = 1020$ м – гроза примерно в километре.

Длина волны λ («лямбда») – расстояние между двумя соседними сгущениями. Пока источник делает одно колебание (за один период T), первое сгущение успевает убежать на одну длину волны:

$$\lambda = v \cdot T = \frac{v}{f},$$

где v – скорость звука, f – частота. Звук 340 Гц в воздухе: $\lambda = 340 : 340 = 1$ м. Ультразвук дальномёра 40 кГц: $\lambda = 340 : 40\,000 = 0,0085$ м = 8,5 мм.

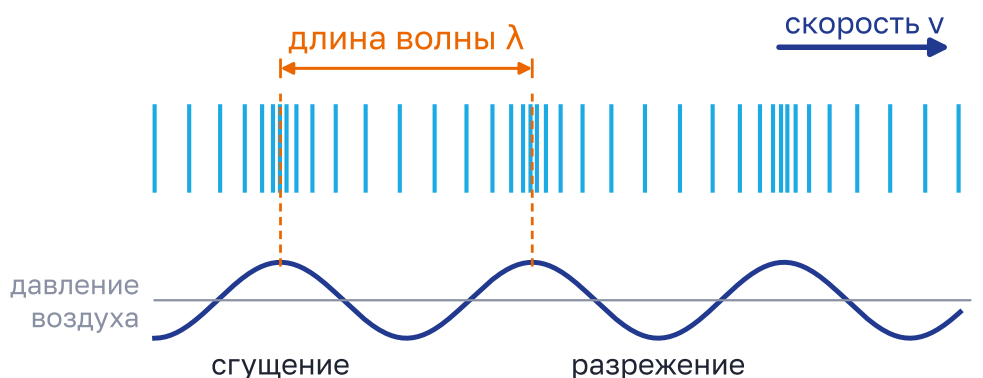


Рисунок 8.3 – Сгущения воздуха и тот же звук на графике: от гребня до гребня – длина волны

Стетоскоп – самый простой звуковой прибор врача. Головку прикладывают к груди, от неё к ушам идут трубки. «Тук-тук» сердца – это захлопываются его клапаны, и звуки эти тихие и низкие. Головка собирает колебания с кожи, а трубки ведут звук прямо в уши и не дают ему расплыться по комнате. У многих стетоскопов головка двусторонняя: плоская мембрана лучше передаёт высокие звуки, чашечка-«колокол» – низкие.

Запомни. Звук – бегущие сжатия воздуха, воды или тела. Высота – это частота, громкость – амплитуда. Слышим примерно от 20 до 20 000 Гц, выше – ультразвук. Скорость звука: в воздухе около 340 м/с, в мягких тканях около 1540 м/с. Длина волны $\lambda = v : f$.

Совет. В $\lambda = v : f$ скорость подставляй в м/с, а частоту – в герцах: мегагерцы сначала переведи в миллионы герц. Тогда длина волны выйдет в метрах, и останется только перевести её в миллиметры, умножив на 1000.

8.3. Эхо: как УЗИ заглядывает внутрь

НА ПАЛЬЦАХ

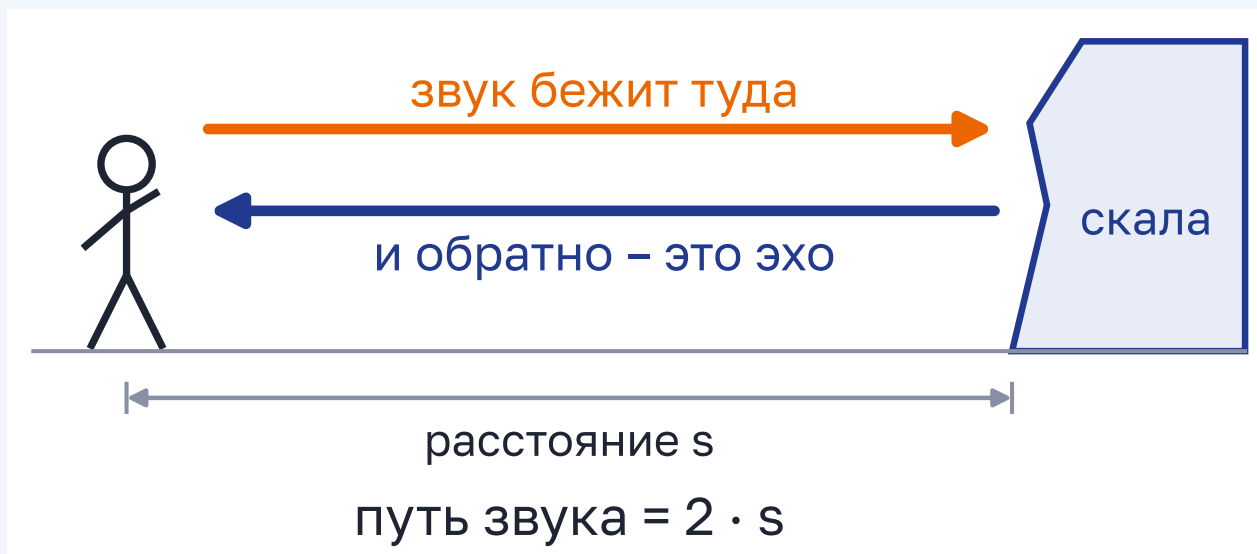


Рисунок 8.4 – Крикнул в сторону скалы – звук добежал до неё и вернулся. Эхо шло туда и обратно

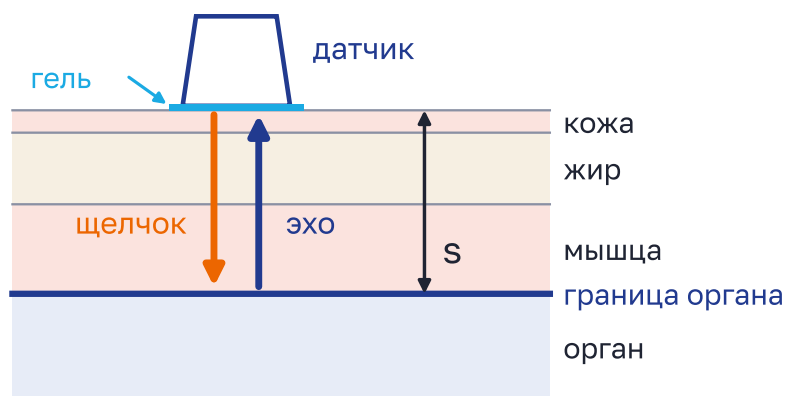
Звук пробежал два расстояния до скалы. Поэтому время эха делят пополам.

Эхо – звук, который отразился от препятствия и вернулся. Засеки, через сколько секунд пришло эхо, – и, зная скорость звука, узнаешь, далеко ли препятствие. Только помни, что звук пробежал путь дважды: до препятствия и обратно.

$$s = \frac{v \cdot t}{2},$$

где s – расстояние до препятствия, v – скорость звука, t – время от крика до эха. Крикнул в горах у скалы, эхо вернулось через 2 с: $s = 340 \cdot 2 : 2 = 340$ м.

УЗИ (ультразвуковое исследование) устроено так же. Датчик, которым врач водит по коже, – сразу и динамик, и микрофон. Он посылает очень короткий щелчок ультразвука, замолкает и слушает. Звук уходит в тело, и на каждой границе – между кожей и жиром, между мышцей и органом, между стенкой сосуда и кровью – часть его отражается и возвращается. Аппарат засекает, через сколько микросекунд пришло каждое эхо, и пересчитывает время в глубину по той же формуле, только со скоростью 1540 м/с. Щелчки повторяются тысячи раз в секунду и уходят веером в разные стороны, а из тысяч таких «глубин» аппарат складывает картинку на экране.



туда и обратно звук идёт за время t , глубина $s = v \cdot t : 2$

Рисунок 8.5 – Датчик УЗИ на коже. Щелчок ушёл вниз, отразился от границы на глубине s и вернулся через время t

Для УЗИ удобно помнить скорость в других единицах. 1540 м/с – это 1 540 000 мм за 1 000 000 мкс, то есть **1,54 мм за микросекунду**. Эхо пришло через 100 мкс: туда и обратно звук прошёл $1,54 \cdot 100 = 154$ мм, до границы – половина, 77 мм. А если перевернуть: на каждый сантиметр глубины эхо задерживается на $2 \cdot 10 : 1,54 \approx 13$ мкс.

Зачем на кожу мажут прозрачный гель? Если между датчиком и кожей останется хоть тонкая прослойка воздуха, она отразит почти весь ультразвук, и в тело ничего не попадёт. Гель выгоняет воздух.

А почему ультразвук, а не обычный звук? Дело в длине волны. Волна не замечает деталей намного мельче себя – так морская волна обходит торчащую из воды палку, будто её и нет. У ультразвука 5 МГц в теле длина волны около 0,3 мм, у звука 1 кГц – полтора метра. Чем выше частота, тем мельче детали видны. Но у высокой частоты есть цена: такой ультразвук сильнее гасится в тканях и не достаёт глубоко. Поэтому для глубоких органов (печень, почки) берут датчики пониже частотой, примерно 2–5 МГц, а для того, что близко к коже (сосуды шеи, суставы), – повыше, примерно 7–15 МГц.

Ультразвуковой дальномер. Такой модуль есть в наборе «Медтех Юниор», а распространённая модель с выводами Trig и Echo – в симуляторе Wokwi; программа ниже для неё. Дальномер работает в воздухе на частоте около 40 кГц. Arduino командует «щёлкни» по выводу Trig, модуль посылает ультразвук и держит сигнал на выводе Echo, пока не вернётся эхо. Команда `pulseIn` меряет, сколько микросекунд этот сигнал держался, – столько и шло эхо туда и обратно.

```

const int TRIG = 9;    // вывод «щёлки»
const int ECHO = 10;   // вывод «эхо»

void setup() {
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(TRIG, HIGH);    // короткая команда модулю: пошли ультразвук
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  long t = pulseIn(ECHO, HIGH); // время эха туда и обратно, мкс
  float d = t * 0.034 / 2;     // расстояние, см
  Serial.println(d);
  delay(200);
}

```

Откуда 0,034? Скорость звука 340 м/с = 34 000 см/с. Делим на миллион (в секунде миллион микросекунд) – выходит 0,034 см за микросекунду. Умножаем на время – получаем путь туда и обратно в сантиметрах, делим на 2 – расстояние. Эхо через 1000 мкс: $1000 \cdot 0,034 : 2 = 17$ см. На таких модулях делают бесконтактные дозаторы антисептика, ростомеры и трости для незрячих, которые пищат перед препятствием.

Запомни. Расстояние по эху $s = v \cdot t : 2$ – звук шёл туда и обратно. В теле 1,54 мм за микросекунду, и каждый сантиметр глубины даёт около 13 мкс задержки. В воздухе дальномер: сантиметры = микросекунды $\cdot 0,034 : 2$.

Совет. Прежде чем делить на 2, спроси себя: время в условии – это эхо (туда и обратно) или звук шёл в одну сторону? Эхо делим пополам всегда, а вот если микрофон стоит по другую сторону от источника, пополам делить не надо.

8.4. Свет: что видит оптический датчик

НА ПАЛЬЦАХ

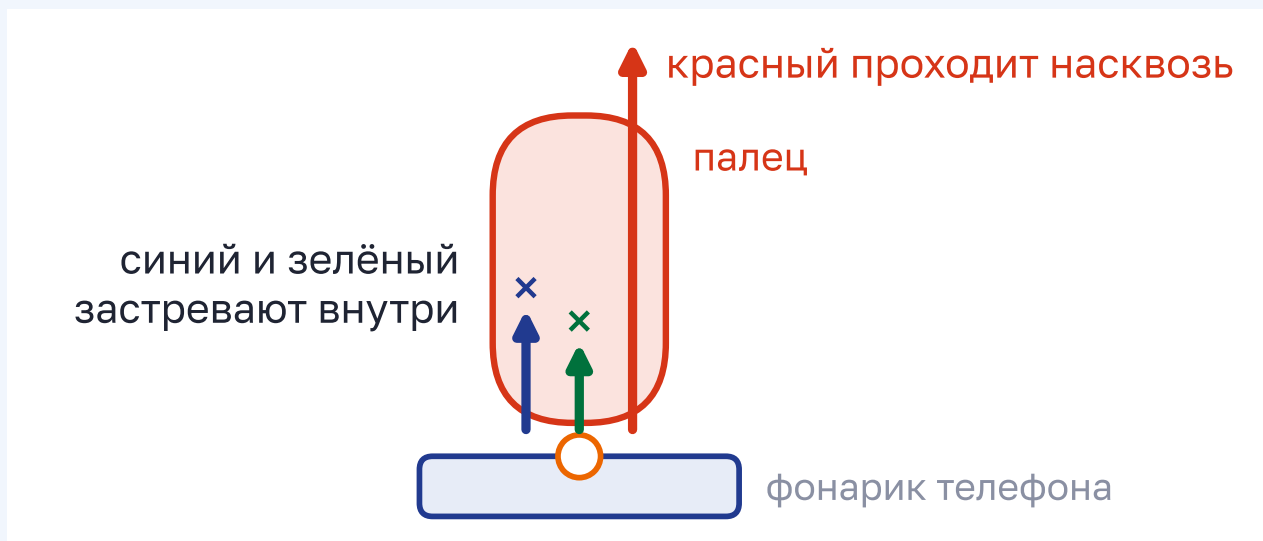


Рисунок 8.6 – Фонарик телефона прижат к пальцу. Синий и зелёный свет застревают внутри, красный проходит насквозь – палец светится красным

Проверь в тёмной комнате: прижми подушечку пальца к фонарику телефона, и палец засветится красным. Долго не держи – фонарик греется.

Свет – тоже волна, но особенная: ему не нужен воздух. От Солнца он летит к нам через пустой космос. Скорость света – около 300 000 км/с: за одну секунду он облетел бы Землю по экватору семь с половиной раз. И идёт свет по прямой – поэтому у предметов бывают тени, а из-за угла ничего не видно.

Цвет света – это его длина волны. Видимый свет занимает узкую полосу: примерно от 380 до 780 нанометров (границы включены; **нанометр**, нм, – миллионная доля миллиметра, 1 мкм = 1000 нм, так что это 0,38–0,78 мкм). Короче 380 нм – ультрафиолет. Самые короткие волны глаз видит фиолетовыми и синими, самые длинные – красными. Волны длиннее 780 нм – это **инфракрасный** свет (ИК). Глаз его не видит. Зато камера телефона часто видит: посмотри через неё на пульт от телевизора, когда жмёшь кнопку, – на кончике пульта замигает огонёк.

Свет, попавший на тело, ждёт три судьбы: часть **отражается**, часть **поглощается** (превращается в тепло), часть **проходит** дальше. Кровь сильно поглощает синий и зелёный свет, а красный и особенно ближний инфракрасный проходят сквозь ткани гораздо легче. Отсюда два вида датчиков пульса из главы 2 (раздел 2.5). Прищепка на палец светит красным и ИК **на просвет**, насквозь. Браслет светит зелёным **на отражение**: ему не нужно пробивать всю руку, свету хватает неглубокого слоя кожи, где зелёный заметнее всего меняется от прилива крови.

С каждым ударом сердца в пальце становится чуть больше крови – и до приёмника доходит чуть меньше света. Изменение крошечное, примерно на процент-другой, а то и меньше. Приёмнику нужно уловить эти проценты на фоне всего остального света, поэтому пульсовую волну так легко испортить шевелением руки.

Сколько света пройдёт сквозь слой. Возьми стопку одинаковых тёмных стёкол. Пусть первое пропускает половину света. Второе пропустит половину от того, что дошло до него, – четверть

от начального. Третье – восьмую часть. Каждый одинаковый слой отнимает не одинаковое количество света, а одинаковую **долю** того, что на него пришло. Толщину, которая пропускает половину, называют **слоем половинного ослабления**. Сквозь n таких слоёв пройдёт

$$\text{доля света} = \left(\frac{1}{2}\right)^n,$$

то есть половина, умноженная сама на себя n раз.

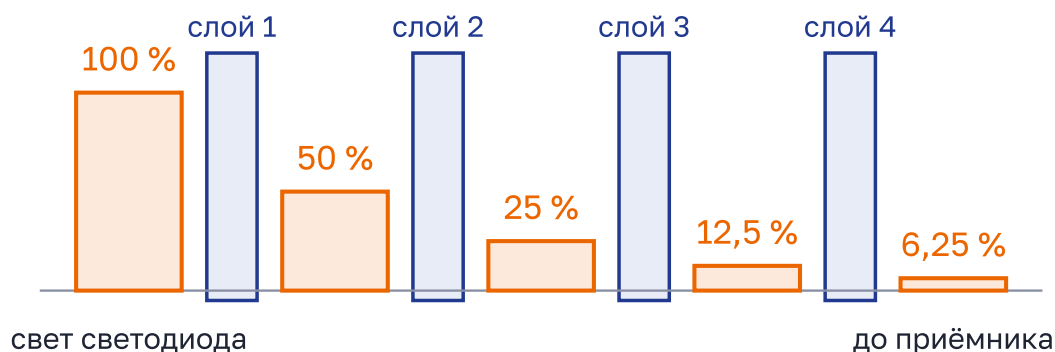


Рисунок 8.7 – Каждый слой половинного ослабления пропускает половину того, что на него упало

Таблица 8.3

Число слоёв n	Проходит света, %
0	100
1	50
2	25
3	12,5
4	6,25
5	3,125

У настоящего пальца толщина половинного слоя зависит от цвета света, от кожи, от того, сколько в пальце крови, так что в задачах её дают в условии. Но картина верная: свет в теле гаснет не «по линейке», а «пополам, ещё пополам».

Для любопытных: почему УЗИ не делают светом. Свет в теле быстро рассеивается и глубоко не проходит. И засечь его эхо очень трудно: от границы на глубине 6 см свет вернулся бы через $0,12 : 300\,000\,000 = 0,000\,000\,000\,4$ с. Звуку на тот же путь нужно около 78 мкс – примерно в двести тысяч раз дольше. Такое время простая электроника меряет легко, а свету понадобился бы очень дорогой и быстрый прибор.

Запомни. Свет – волна, которой не нужен воздух, скорость около 300 000 км/с. Цвет – это длина волны: красный длиннее синего, инфракрасный ещё длиннее и глазу не виден. Ткани пропускают красный и ИК лучше, чем зелёный и синий. Каждый слой половинного ослабления делит свет пополам.

8.5. Сила и рычаг: как устроена рука

НА ПАЛЬЦАХ

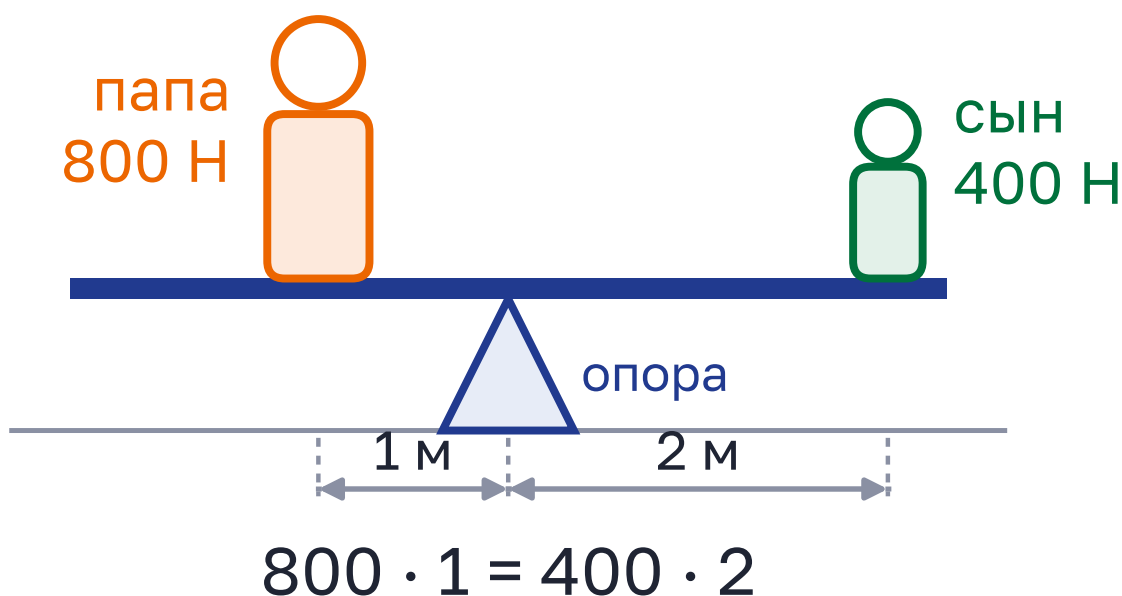


Рисунок 8.8 – Качели-балансир. Лёгкий сел далеко от опоры, тяжёлый – близко, и доска в равновесии

Кто сидит дальше от опоры, тому легче повернуть доску. Сила, умноженная на расстояние до опоры, у обоих одинаковая.

Сила – толчок или тяга. Меряют её в **ньютонaх** (Н). Чтобы почувствовать 1 Н, положи на ладонь небольшое яблоко граммов на сто: примерно с такой силой оно давит вниз. Предмет массой 1 кг давит на опору с силой около 10 Н (точнее, 9,8 Н). Эту силу называют **весом**. В задачах, если не сказано иначе, берём 10 Н на каждый килограмм: гантель 2 кг весит 20 Н.

Рычаг – твёрдая палка, которая может поворачиваться вокруг одной точки, **опоры**. Лом, ножницы, качели, дверь на петлях – всё это рычаги. **Плечо силы** – расстояние от опоры до точки, где приложена сила (в задачах сила тянет поперёк палки).

Рычаг не поворачивается, когда уравновешены две «силы поворота». Силу поворота физики называют **моментом силы**: это сила, умноженная на своё плечо. Правило рычага словами: **сила, умноженная на плечо, с одной стороны равна силе, умноженной на плечо, с другой**. Буквами:

$$F_1 \cdot l_1 = F_2 \cdot l_2,$$

где F_1 и F_2 – силы, l_1 и l_2 – их плечи. На картинке папа весит 800 Н и сидит в 1 м от опоры, сын весит 400 Н и сидит в 2 м: $800 \cdot 1 = 400 \cdot 2$, доска в равновесии. Если в одну сторону поворачивают несколько сил, их моменты складывают.

У рычага есть цена. Короткий конец движется меньше длинного во столько же раз, во сколько он сильнее. Хочешь поднять тяжёлое малой силой – приложи силу к длинному концу, но тогда и двигать его придётся далеко.

Рычаги в теле. Кости – палки, суставы – опоры, а мышцы тянут кости через **сухожилия**, прочные тяжи на концах мышц. Возьмём для примера предплечье. Опора здесь – локтевой сустав. **Бицепс** (двуглавая мышца плеча) крепится к предплечью совсем близко к локтю, примерно в 4–5 см. А ладонь с грузом – примерно в 30–35 см от локтя. Плечо груза длиннее плеча мышцы в 6–9 раз (смотря у кого какая рука), и во столько же раз бицепсу приходится тянуть сильнее, чем весит груз.

Пример. В ладони гантель 2 кг (20 Н) в 32 см от локтя, бицепс крепится в 4 см. Вес самого предплечья пока не учитываем. По правилу рычага $F \cdot 4 = 20 \cdot 32$, откуда $F = 640 : 4 = 160$ Н. Мышца тянет с силой, с которой давили бы 16 кг, хотя в руке всего два.

Зачем природе такой невыгодный рычаг? В силе рука проигрывает, зато выигрывает в размахе. Бицепс укорачивается на 1 см – ладонь за то же время поднимается на целых 8 см, то есть движется в 8 раз быстрее мышцы. Поэтому рукой можно резко бросить мяч или отдернуть её от горячего.

Бионическая рука устроена так же: пластиковые фаланги пальцев поворачиваются в шарнирах, а тянет их нить-«сухожилие» от сервопривода.

Запомни. 1 кг весит около 10 Н. Правило рычага: $F_1 \cdot l_1 = F_2 \cdot l_2$, плечо меряют от опоры. Несколько сил поворачивают в одну сторону – складываем их моменты. Мышцы крепятся близко к суставам: тянут сильнее груза, зато двигают руку быстро и далеко.

Совет. Начинай любую задачу на рычаг с того, что отметить опору. Все плечи меряют от неё, а не от края палки и не от соседней силы.

8.6. Сервопривод: угол, нить и импульсы

НА ПАЛЬЦАХ

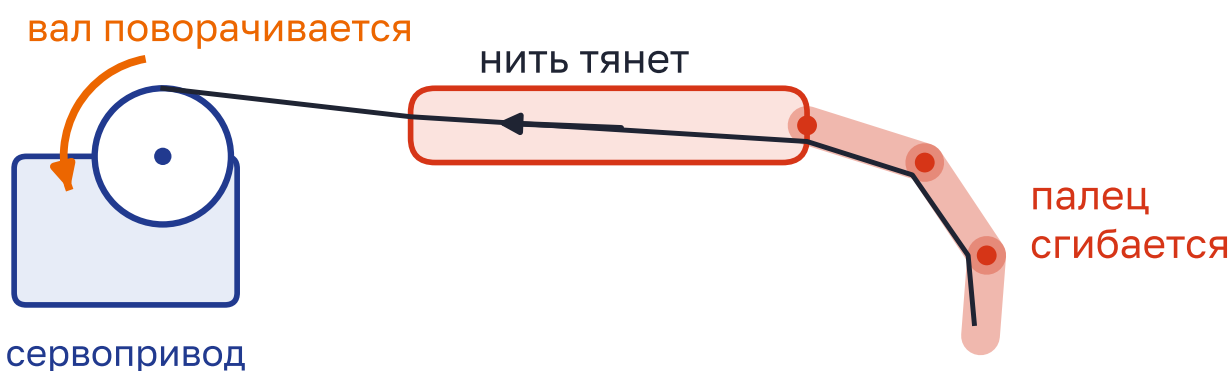


Рисунок 8.9 – Качалка сервопривода повернулась и намотала нить. Нить тянет палец – палец сгибается

Как ворот колодца: крутишь ручку – верёвка наматывается, ведро едет вверх.

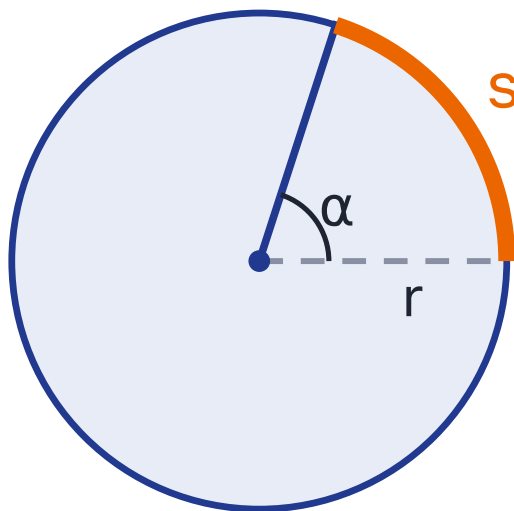
Сервопривод знаком нам по главе 3: мотор, который поворачивает свой вал на заданный угол от 0 до 180° и держит его. На вал надевают **качалку** – пластиковый диск или рычажок. Если

качалка круглая, нить наматывается на её край, как верёвка на ворот.

Сколько нити вытянет поворот. Край диска движется по окружности. **Длина окружности** $C = 2\pi r$, где r – радиус диска, а $\pi \approx 3,14$ («пи») – число, которое показывает, во сколько раз окружность длиннее своего диаметра. Полный оборот – 360° – наматывает всю окружность. Поворот на угол α («альфа») наматывает её часть $\alpha : 360$:

$$s = \frac{\alpha}{360} \cdot 2\pi r.$$

Пример. Радиус диска 15 мм, поворот на 90° – это четверть оборота. Длина окружности $2 \cdot 3,14159 \cdot 15 = 94,25$ мм, четверть от неё – 23,6 мм. Столько нити наматывается на диск.



$$\text{дуга } s = \alpha : 360 \cdot 2\pi r$$

Рисунок 8.10 – Качалка радиусом r повернулась на угол α . Нить вытянулась на длину дуги s

Какую силу даст сервопривод. В паспорте сервопривода пишут **момент** – например, «1,8 кг·см». Это значит: на плече 1 см вал удержит груз 1,8 кг, на плече 2 см – вдвое меньше, 0,9 кг. Знакомое правило рычага: сила, умноженная на плечо, не меняется. Поэтому большой диск тянет нить дальше, но слабее, а маленький – короче, но сильнее.

Как Arduino сообщает сервоприводу угол. По сигнальному проводу идут короткие импульсы – 50 раз в секунду, то есть с периодом $1 : 50 = 0,02 \text{ с} = 20 \text{ мс}$. **Импульс** – короткий промежуток, когда на проводе 5 В вместо 0. Угол задаёт **длительность импульса**: чем дольше он держится, тем больше угол. У многих моделей примерно 1 мс соответствует 0° , а 2 мс – 180° , и между ними угол растёт равномерно. Точные числа у разных сервоприводов свои, поэтому в задачах их дают в условии. В скетче считать это не нужно: команда `write(угол)` из библиотеки Servo сама подбирает длительность.

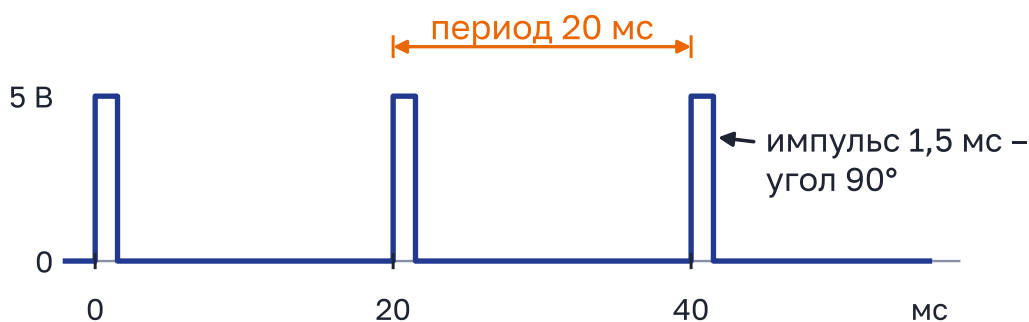


Рисунок 8.11 – Сигнал сервоприводу: импульс каждые 20 мс. Длина импульса 1,5 мс – это угол 90°

Угол по длительности импульса считается как пересчёт **map** из главы 3:

$$\alpha = \frac{\tau - \tau_0}{\tau_{180} - \tau_0} \cdot 180^\circ,$$

где τ («тау») – длительность импульса, τ_0 – длительность для 0°, τ_{180} – для 180°. Импульс 1,5 мс: $(1,5 - 1) : (2 - 1) \cdot 180 = 90^\circ$.

Поворачивается вал не мгновенно. В паспорте пишут скорость вроде «0,1 с на 60°»: 180° – это три раза по 60°, на них уйдёт $0,1 \cdot 3 = 0,3$ с. Если программа меняет угол чаще, вал не будет успевать доезжать.

Запомни. Нить на диске: $s = \alpha : 360 \cdot 2\pi r$. Сигнал сервоприводу – 50 импульсов в секунду (период 20 мс), угол задаёт длина импульса. Больше радиус качалки – длиннее ход нити, но меньше сила.

8.7. Разбираем задачи

Задача 8.1 Камертон невролога

разминка · частота и период, миллисекунды, число колебаний

Врач-невролог проверяет, чувствует ли человек вибрацию. Он ударяет камертоном – металлической вилкой, которая звенит одним тоном, – и ставит его ножку на косточку у запястья. Частота камертона 128 Гц. а) Сколько миллисекунд длится одно колебание? Округли до десятых. б) Сколько колебаний сделает камертон за 4 с? Считай, что частота всё это время не меняется.

Дано: $f = 128$ Гц; $t = 4$ с.

Найти: T (в мс) – ?, N – ?

Решение

Шаг 1. Период в секундах. Одно колебание длится $T = 1 : f = 1 : 128 = 0,0078125$ с.

Шаг 2. Переводим в миллисекунды. В секунде 1000 мс: $0,0078125 \cdot 1000 = 7,8125$ мс. До десятых – 7,8 мс.

Шаг 3. Число колебаний. За каждую секунду камертон качнётся 128 раз, за 4 с – в 4 раза больше: $N = f \cdot t = 128 \cdot 4 = 512$.

Проверим обратным ходом: 512 колебаний по 7,8125 мс – это $512 \cdot 7,8125 = 4000$ мс, те самые 4 с.

 **Типичная ошибка.** Записать «период 128 мс», просто переписав частоту. Но период – время одного колебания, и чем чаще камертон дрожит, тем период короче: сто с лишним колебаний в секунду никак не поместятся, если каждое тянется по 128 мс. Вторая ошибка – ввести 0,0078, хотя просили миллисекунды.

Ответ: а) 7,8 мс; б) 512.

Своя задача по программе сферы, модуль 3

Реши сам

8.1. Датчик аппарата УЗИ работает на частоте 4 МГц. а) Сколько микросекунд длится одно колебание? б) Сколько колебаний он сделает за 1 мс?

8.2. Пульсоксиметр у кровати больного на каждый удар сердца издаёт короткий писк: звук частотой 2 кГц длительностью 50 мс. а) Каков период этого звука в миллисекундах? б) Сколько колебаний укладывается в один писк?

Задача 8.2 Почему УЗИ делают ультразвуком

длина волны, мегагерцы, перевод метров в миллиметры

Сравним две волны. а) Низкий звук частотой 170 Гц, который врач слышит в стетоскоп, бежит по воздуху трубки со скоростью 340 м/с. б) Ультразвук датчика УЗИ частотой 5 МГц бежит в мягких тканях со скоростью 1540 м/с. Найди длину волны в обоих случаях: в пункте а) – в метрах, в пункте б) – в миллиметрах, округли до сотых.

Дано: а) $f = 170$ Гц, $v = 340$ м/с; б) $f = 5$ МГц, $v = 1540$ м/с.

Найти: λ в каждом случае – ?

Решение

Шаг 1. Формула. Пока источник делает одно колебание, волна убегает на одну длину волны. Колебаний в секунду f , значит, $\lambda = v : f$.

Шаг 2. Пункт а). $\lambda = 340 : 170 = 2$ м.

Шаг 3. Пункт б): переводим частоту. Мега – миллион: $5 \text{ МГц} = 5\,000\,000 \text{ Гц}$.

Шаг 4. Длина волны в метрах. $\lambda = 1540 : 5\,000\,000 = 0,000\,308$ м.

Шаг 5. В миллиметры. В метре 1000 мм: $0,000\,308 \cdot 1000 = 0,308$ мм. До сотых – 0,31 мм.

Разница огромная. Два метра – высота двери, а 0,31 мм – толщина трёх листов бумаги из тетради. Волна длиной с дверь проскочила бы мимо почки (а это всего сантиметров десять) и не

заметила бы в ней ничего, а ультразвук различает детали мельче миллиметра.

⚠ Типичная ошибка. Не перевести мегагерцы: $1540:5 = 308$, и в ответе «308 мм» – волна длиной с линейку. Бывает и наоборот: всё посчитано верно, но в ответ введено 0,000308 – а просили миллиметры.

Ответ: а) 2 м; б) 0,31 мм.

Своя задача по программе сферы, модуль 3

Реши сам

8.3. Для УЗИ печени взяли датчик частотой 3,5 МГц. Скорость звука в тканях 1540 м/с. Найди длину волны в миллиметрах, округли до сотых.

8.4. Врачу нужно, чтобы длина волны ультразвука в тканях была 0,22 мм. Скорость звука в тканях 1540 м/с. На какой частоте должен работать датчик? Ответ в мегагерцах.

Задача 8.3 На какой глубине граница

эхо, микросекунды, путь туда и обратно

Датчик УЗИ послал в тело короткий щелчок ультразвука. Эхо от границы органа вернулось через 78 мкс. Скорость звука в мягких тканях 1540 м/с. На какой глубине под датчиком эта граница? Ответ в миллиметрах, округли до целых.

Дано: $t = 78$ мкс; $v = 1540$ м/с.

Найти: глубину s (в мм) – ?

Решение

Шаг 1. Что прошёл звук. До границы и обратно – две глубины. Значит, путь звука мы найдём по времени, а глубину получим, поделив его пополам.

Шаг 2. Переводим время. В секунде миллион микросекунд: 78 мкс
 $= 78:1\,000\,000 = 0,000\,078$ с.

Шаг 3. Путь туда и обратно. $v \cdot t = 1540 \cdot 0,000\,078 = 0,120\,12$ м.

Шаг 4. Глубина. $0,120\,12:2 = 0,060\,06$ м = 60,06 мм. До целых – 60 мм.

Короче через «1,54 мм за микросекунду»: $1,54 \cdot 78 = 120,12$ мм туда и обратно, половина – 60,06 мм. И грубая прикидка по правилу «13 мкс на сантиметр»: $78:13 = 6$ см. Все три пути сошлись на шести сантиметрах – для органа в животе обычная глубина.

⚠ Типичная ошибка. Забыть поделить на 2 и получить 120 мм – вдвое глубже настоящей границы. Ещё одна – взять скорость звука в воздухе, 340 м/с: выйдет 13 мм, как будто орган лежит прямо под кожей. В теле звук бежит в четыре с половиной раза быстрее, чем в воздухе.

Ответ: 60 мм.

Своя задача по программе сферы, модуль 3

Реши сам

8.5. Граница между мышцей и костью лежит на глубине 4,5 см под датчиком УЗИ. Скорость звука в мягких тканях 1540 м/с. Через сколько микросекунд после щелчка вернётся эхо от неё? Округли до целых.

8.6. Датчик УЗИ приложили к бедру. Эхо от верхней границы мышцы пришло через 20 мкс, от нижней – через 85 мкс. Скорость звука в тканях 1540 м/с. Какова толщина мышцы в миллиметрах? Округли до целых.

Задача 8.4 Ростомер без линейки

ультразвуковой дальномер, `pulseIn`, эхо в воздухе

В медпункте лагеря сделали бесконтактный ростомер. Ультразвуковой дальномер закреплён на планке на высоте 210 см над полом и смотрит вниз. Ученик встаёт под него, и Arduino выполняет программу из раздела 8.3:

```
long t = pulseIn(ECHO, HIGH); // время эха туда и обратно, мкс
float d = t * 0.034 / 2;      // расстояние, см
```

Команда `pulseIn` вернула $t = 2700$. Какой рост у ученика? Ответ в сантиметрах, округли до целых.

Дано: высота датчика 210 см; $t = 2700$ мкс; скорость звука 340 м/с (0,034 см/мкс).

Найти: рост – ?

Решение

Шаг 1. Что измерила программа. `pulseIn` вернула время, за которое ультразвук долетел от датчика до макушки и вернулся обратно: 2700 мкс.

Шаг 2. Скорость в удобных единицах. 340 м/с = 34 000 см/с. В секунде миллион микросекунд: $34\,000 : 1\,000\,000 = 0,034$ см/мкс. Это число и стоит в программе.

Шаг 3. Путь туда и обратно. $2700 \cdot 0,034 = 91,8$ см.

Шаг 4. От датчика до макушки. $91,8 : 2 = 45,9$ см. Это число программа и напечатает.

Шаг 5. Рост. Датчик висит на 210 см над полом, от него до макушки 45,9 см:

$210 - 45,9 = 164,1$ см, до целых – 164 см.

У девятиклассника рост 164 см никого не удивит.

 **Типичная ошибка.** Забыть деление на 2: $210 - 91,8 = 118$ см – ростомер «укоротил» старшеклассника до первоклассника. Другой промах – ввести в ответ 45,9: программа печатает расстояние до макушки, а спрашивали рост.

Ответ: 164 см.

Своя задача по программе сферы, модуль 1

Реши сам

8.7. Ультразвуковой дальномер смотрит на стену. Команда `pulseIn` вернула 1470 мкс. Скорость звука 340 м/с. Каково расстояние до стены в сантиметрах? Округли до целых.

8.8. Бесконтактный дозатор антисептика должен включать насос, когда ладонь ближе 8 см к дальномеру. Скорость звука 340 м/с. Программа включает насос, если время эха не больше числа-границы. Какую наибольшую целую границу (в микросекундах) можно взять, чтобы на ладонь в 8 см и дальше насос не срабатывал? Прежде чем округлять, подумай, куда: вверх или вниз.

Задача 8.5 Сколько света пройдёт сквозь палец

свет в тканях, слой половинного ослабления, проценты

Датчик пульса на прищепке светит красным светодиодом сквозь палец, фотоприёмник стоит с другой стороны. В этой задаче считаем, что каждые 4 мм ткани пропускают половину красного света, который на них попал. Толщина пальца 16 мм. Какая часть света светодиода дойдёт до фотоприёмника? Ответ в процентах.

Дано: слой половинного ослабления 4 мм; толщина 16 мм.

Найти: долю прошедшего света в % – ?

Решение

Шаг 1. Сколько половинных слоёв в пальце. $16:4 = 4$ слоя.

Шаг 2. Делим пополам четыре раза. Перед пальцем 100 %. После первого слоя – 50 %, после второго – 25 %, после третьего – 12,5 %, после четвёртого – 6,25 %.

Шаг 3. То же одной строкой. $\left(\frac{1}{2}\right)^4 = \frac{1}{16} = 0,0625$, в процентах – $0,0625 \cdot 100 = 6,25$ %.

Немного, но приёмнику хватает. А теперь посвети зелёным. Пусть для него половинный слой в этой задаче – 1 мм. Тогда слоёв 16, и пройдёт $\frac{1}{65536}$ света, около 0,0015 %. Потому прищепки на палец и светят красным и инфракрасным, а не зелёным.

⚠ Типичная ошибка. Считать, что каждый слой отнимает половину **начального** света: четыре слоя по 50 % – минус 200 %, и света становится «меньше нуля». Каждый слой делит пополам то, что до него дошло. Другая ошибка – поделить 100 на 4 слоя и ответить 25 %: это всего два половинных слоя, а их четыре.

Ответ: 6,25 %.

Своя задача по программе сферы, модуль 2

Реши сам

8.9. Показание фотоприёмника пульсометра пропорционально свету, который до него дошёл. Между ударами `analogRead` выдаёт 640. В момент удара до приёмника доходит на 1,5 % света меньше. Какое показание будет в момент удара? Округли до целых.

8.10. Сквозь слой ткани толщиной 12 мм прошло 12,5 % света. Какова толщина слоя половинного ослабления для этого света? Ответ в миллиметрах.

Задача 8.6 Гантель и бицепс

правило рычага, сумма моментов, вес

Школьник держит в ладони гантель массой 2 кг. Предплечье горизонтально, а часть руки от плеча до локтя опущена вертикально (рисунок ниже). Опора – локтевой сустав. Бицепс тянет предплечье вверх в 4 см от локтя. Предплечье с кистью весит 16 Н, и этот вес можно считать приложенным в 15 см от локтя. Гантель лежит в ладони в 33 см от локтя. С какой силой тянет бицепс? Ответ в ньютонах, 1 кг весит 10 Н.

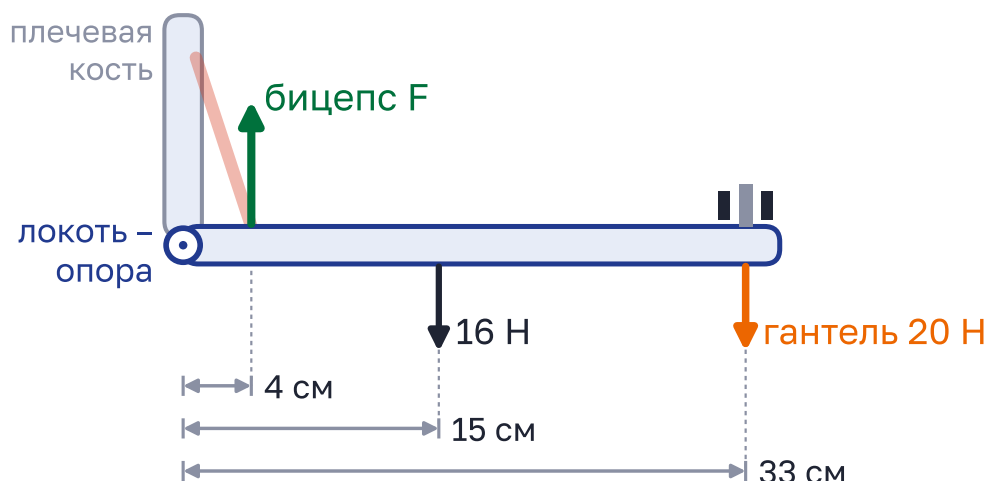


Рисунок 8.12 – Предплечье – рычаг с опорой в локте. Бицепс тянет вверх, вес предплечья и вес гантели – вниз

Дано: $m = 2$ кг; $l_б = 4$ см; $P_{пр} = 16$ Н, $l_{пр} = 15$ см; $l_г = 33$ см.

Найти: $F_б$ – ?

Решение

Шаг 1. Опора и силы. Опора – локоть, все плечи меряем от него. Вниз руку поворачивают две силы: вес предплечья и вес гантели. Вверх – одна, бицепс.

Шаг 2. Вес гантели. $2 \cdot 10 = 20$ Н.

Шаг 3. Моменты, которые тянут вниз. Предплечье: $16 \cdot 15 = 240$ Н·см. Гантель: $20 \cdot 33 = 660$ Н·см. Обе поворачивают в одну сторону, складываем: $240 + 660 = 900$ Н·см.

Шаг 4. Правило рычага. Момент бицепса должен быть таким же: $F_б \cdot 4 = 900$, откуда $F_б = 900 : 4 = 225$ Н.

225 Н – столько весит груз в 22,5 кг. Заметь: даже с пустой ладонью бицепс держит одно предплечье с силой $240:4 = 60$ Н.

❗ **Типичная ошибка.** Забыть про вес самого предплечья: $20 \cdot 33:4 = 165$ Н – ответ заметно меньше верного. А ещё плечо гантели иногда меряют от бицепса: $33 - 4 = 29$ см. Плечи всегда меряют от опоры, то есть от локтя.

Ответ: 225 Н.

По мотивам задачи первого отборочного этапа НТО «Нейротехнологии и когнитивные науки» 2025/26, предметный тур по информатике, «Вопросы про центр тяжести»

Реши сам

8.11. В ладони мяч массой 3 кг, он в 30 см от локтя. Бицепс крепится в 5 см от локтя, предплечье горизонтально. Вес предплечья не учитывай. С какой силой тянет бицепс? 1 кг весит 10 Н.

8.12. Палец бионической руки поворачивается в шарнире. Нить тянет его с силой 6 Н в 1,5 см от шарнира, а кончик пальца давит на кнопку с силой 1,2 Н. На каком расстоянии от шарнира кончик пальца? Ответ в сантиметрах.

Задача 8.7* Палец на сервоприводе

период импульсов, угол по длительности, длина дуги

Палец бионической руки сгибает нить, которая наматывается на круглую качалку сервопривода радиусом 12 мм. Arduino посылает сервоприводу импульсы с частотой 50 Гц. У этого сервопривода импульс длиной 1,0 мс означает угол 0° , 2,0 мс – угол 180° , а между ними угол растёт равномерно. Сначала вал стоял на 0° , потом программа стала посылать импульсы по 1,4 мс. а) Каков период импульсов в миллисекундах? б) На какой угол повернётся вал? в) На сколько миллиметров вытянется нить? Округли до десятых.

Дано: $r = 12$ мм; $f = 50$ Гц; 1,0 мс – 0° , 2,0 мс – 180° ; $t = 1,4$ мс.

Найти: T – ?, α – ?, s – ?

Решение

Шаг 1. Период. $T = 1:50 = 0,02$ с. В миллисекундах: $0,02 \cdot 1000 = 20$ мс.

Шаг 2. Какая часть диапазона. Весь диапазон импульсов – от 1,0 до 2,0 мс, это $2,0 - 1,0 = 1,0$ мс, и на него приходится 180° . Наш импульс длиннее начального на $1,4 - 1,0 = 0,4$ мс.

Шаг 3. Угол. $0,4:1,0 \cdot 180 = 72^\circ$.

Шаг 4. Длина окружности качалки. $C = 2 \cdot 3,14159 \cdot 12 = 75,40$ мм.

Шаг 5. Какая часть окружности намоталась. $72:360 = 0,2$ – одна пятая оборота.

Шаг 6. Длина нити. $s = 0,2 \cdot 75,40 = 15,08$ мм. До десятых – 15,1 мм.

Прикинем другим путём. Полоборота (180°) намотало бы половину окружности, $75,40:2 = 37,7$ мм. 72° – это $72:180 = 0,4$ от полуоборота, $0,4 \cdot 37,7 = 15,08$ мм – то же самое.

 **Типичная ошибка.** Посчитать угол «от нуля»: $1,4:2,0 \cdot 180 = 126^\circ$. Отсчёт идёт от импульса $1,0$ мс, а не от нуля – ловушка та же, что с **тар** в главе 3. Вторая ошибка – подставить в $2\pi r$ диаметр вместо радиуса и получить нить вдвое длиннее, $30,2$ мм.

Ответ: а) 20 мс; б) 72° ; в) 15,1 мм.

Своя задача по программе сферы, модуль 2

Реши сам

8.13. На круглую качалку радиусом 10 мм намотана нить. Вал повернулся с 30° на 120° . На сколько миллиметров вытянулась нить? Округли до десятых.

8.14.* Чтобы палец бионической руки согнулся полностью, нить нужно вытянуть на 20 мм. Качалка круглая, радиус 14 мм, сначала вал стоит на 0° . а) На какой угол повернуть вал? Ответ – целое число градусов, такое, чтобы палец точно согнулся полностью. б) Какой длины импульс для этого угла, если $1,0$ мс – это 0° , а $2,0$ мс – 180° ? Ответ в миллисекундах, до сотых. в) По паспорту сервопривод поворачивается на 60° за $0,1$ с. За сколько секунд вал повернётся от 0° до этого угла? Округли до сотых.

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. Тело человека как система сигналов

1.1. Ответ: 1 – В, 2 – Б, 3 – А, 4 – Г, 5 – Д.

1.2. Ответ: 1 – В, 2 – А, 3 – Б, 4 – Д, 5 – Г, 6 – Е (лишний – Ж).

Подсказка: мышца только тянет, поэтому из деталей ей подходит лебёдка с тросом, а вентилятор – это скорее лёгкие, которых в списке нет.

1.3. Ответ: 2346.

Подсказка: утверждение 1 ломает лёгочная артерия, утверждение 5 – само назначение клапанов.

1.4. Ответ: БГЕАДВ.

1.5. Ответ: 5,04 л.

1.6. Ответ: 3,2 л.

Подсказка: сначала объём воздуха за 10 минут ($0,5 \cdot 16 \cdot 10 = 80$ л), потом 4 % от него.

1.7. Ответ: 2.

1.8. Ответ: 23.

Подсказка: пальцы ног двигают мышцы голени и стопы, а электроды стоят на предплечье.

1.9. Ответ: 134.

1.10. Ответ: 1 – В, 2 – Г, 3 – А, 4 – Б.

1.11. Ответ: 25 мс.

1.12. Ответ: через 980 мс.

Подсказка: посчитай время каждого сигнала в миллисекундах (20 мс и 1000 мс) и найди разницу.

1.13. Ответ: а) в 1250 раз; б) в 30 раз.

1.14. Ответ: усиление в 1500 раз; 918 ступенек.

Подсказка: проверь каждое усиление: $3 \cdot 2000 = 6000$ мВ = 6 В – уже больше 5 В.

К главе 2. Пульс и сердце: считаем по графику

2.1. Ответ: 76 уд/мин.

2.2. Ответ: 30 ударов.

Подсказка: задача обратная: 90 ударов за 60 с, а 20 с – треть минуты.

2.3. Ответ: 125 уд/мин.

2.4. Ответ: 500 мс; 2 Гц.

2.5. Ответ: 75 уд/мин.

Подсказка: при 50 мм/с один миллиметр – это $1 : 50 = 0,02$ с, а не 0,04 с.

2.6. Ответ: 0,64 с; 94 уд/мин.

2.7. Ответ: средняя 60 уд/мин; самая маленькая 58 уд/мин.

Подсказка: самый медленный пульс – по самому длинному интервалу, 26 мм.

2.8. Ответ: средняя 75 уд/мин; наибольшая 80 уд/мин, наименьшая 71 уд/мин.

2.9. Ответ: на 67 %.

2.10. Ответ: 108 уд/мин; на 28 %.

Подсказка: 10 % считаются уже от 120, поэтому всего вышло не 30 %.

2.11. Ответ: от 100 до 133 мс.

Подсказка: 600 уд/мин даёт самый короткий интервал, 450 – самый длинный.

2.12. Ответ: 1 – В, 2 – А, 3 – Б.

2.13. Ответ: а) в 2,8 раза; б) 68 л.

Подсказка: для пункта б) минутный объём на тренировке (13,6 л) умножь на 5 минут.

2.14. Ответ: 84 мл и 120 мл; на 42,9 %.

Подсказка: цепочка обратная: ударный объём = минутный объём : ЧСС, литры сначала переведи в миллилитры.

К главе 3. Мышцы и мозг: ЭМГ и ЭЭГ

3.1. Ответ: 200.

3.2. Ответ: а) нет; б) 105.

Подсказка: при сжатии огибающая теперь опускается до 150 – это ниже старого порога 200.

3.3. Ответ: 4.

3.4. Ответ: 10 Гц, альфа (вариант 3).

Подсказка: сначала найди период: 25 отсчётов по 1/250 с каждый.

3.5. Ответ: а) 4; б) 1 с.

Подсказка: значение 250 порог не превышает, а 251 – уже превышает.

3.6. Ответ: а) 5 и 3; б) порог 350.

3.7. Ответ: а) 81°; б) 0°.

Подсказка: в пункте б) `map` даёт отрицательное число, и `constrain` поднимает его до 0.

3.8. Ответ: 380.

Подсказка: при 379 выходит 89,7°, и `map` оставляет 89.

3.9. Ответ: +2.

3.10. Ответ: 410.

Подсказка: флажок не двинется к Ане, если разность 450 – (огibaющая Бори) не больше 40.

3.11. Ответ: 13 м.

Подсказка: в первые две секунды ракета стоит на земле – ниже ей некуда.

3.12. Ответ: 14 с.

Подсказка: если бы все 20 секунд были «хорошими», ракета поднялась бы на 100 м. Каждая «плохая» секунда вместо «хорошей» отнимает $5 + 2 = 7$ м.

3.13. Ответ: а) 4; б) 1.

Подсказка: во втором сжатии огibaющая доходит только до 500, а это не больше 500.

3.14. Ответ: 3.

Подсказка: порог разжатия должен быть ниже всего, что бывает при сжатии (400), и выше всего, что бывает в покое (180).

К главе 4. Сигналы и обработка данных

4.1. Ответ: размах 164; амплитуда 82.

4.2. Ответ: наибольшее 620, наименьшее 380; размах 240.

Подсказка: задача обратная: от среднего уровня отложи амплитуду вверх и вниз.

4.3. Ответ: 4 мс; 3000 отсчётов.

4.4. Ответ: 40 Гц.

Подсказка: нужно больше $2 \cdot 13 = 26$ отсчётов в секунду, а 25 чуть-чуть не хватает.

4.5. Ответ: 0,55 с; 4 раза.

Подсказка: 0,81 после 0,93 – тоже попадание: взгляд вернулся на картинку справа.

4.6. Ответ: а) 4; б) 2; в) порог 600.

Подсказка: горбики (520 и 530) выше 450, но ниже 600 – низкий порог считает каждый удар дважды.

4.7. Ответ: 10 Гц, альфа-ритм.

4.8. Ответ: 32 отсчёта.

Подсказка: сначала интервал в секундах (60: 75), потом умножь его на число отсчётов в секунду.

4.9. Ответ: 330, 410, 430, 450, 420.

Подсказка: эти пять средних стоят под отсчётами со 2-го по 6-й: у первого и последнего чисел среднего нет.

4.10. Ответ: 123.

Подсказка: сравни каждую частоту с границей 20 Гц: проходит всё, что медленнее.

4.11. Ответ: 24 000 бит/с; 30 000 байт.

4.12. Ответ: 262 с.

Подсказка: переведи 16 Кбайт в биты ($16 \cdot 1024 \cdot 8 = 131\,072$) и раздели на поток 500 бит/с; дробную часть отбрось.

4.13. Ответ: а) 3; б) 2; в) 15 вдохов в минуту.

4.14. Ответ: по трём точкам – 7 чисел, наибольшее 805; по пяти точкам – 5 чисел, наибольшее 756.

Подсказка: по трём точкам средние стоят под отсчётами со 2-го по 8-й, по пяти – с 3-го по 7-й.

К главе 5. Электричество и датчики

5.1. Ответ: 3 мА.

5.2. Ответ: 1,5 кОм.

Подсказка: сначала найди общее сопротивление цепи по закону Ома, потом вычти известный резистор.

5.3. Ответ: 290 Ом; взять 330 Ом.

5.4. Ответ: а) 30 мА; б) нет; в) 220 Ом.

5.5. Ответ: 3,5 В; 716.

5.6. Ответ: 4 кОм.

Подсказка: подставь в формулу делителя каждый резистор из набора по очереди.

5.7. Ответ: 4,50 В.

5.8. Ответ: 180.

Подсказка: задача 5.4 наоборот: из температуры получи вольты, из вольт – число.

5.9. Ответ: 30 ч.

5.10. Ответ: а) 1080 мА·ч; б) 1200 мА·ч.

Подсказка: здесь округляют вверх: 1000 мА·ч на три дня не хватит.

5.11. Ответ: а) 4,0 В; б) 112.

Подсказка: в б) пропорция в обратную сторону: вольты умножь на 255 и раздели на 5.

5.12. Ответ: а) 153; б) 38.

5.13. Ответ: а) 38 мА; б) 131 ч.

5.14. Ответ: 3 мин.

Подсказка: найди, какой средний ток допустим (ёмкость делить на число часов в 5 сутках), а потом подбери минуты.

К главе 6. Программирование: Arduino и машины состояний

6.1. Ответ: 33.

6.2. Ответ: 11221.

Подсказка: иди от 49 назад: оно нечётное, значит, последней была команда «прибавь 5».

6.3. Ответ: Б, В, Г, Д, А.

6.4. Ответ: 3.

Подсказка: в варианте 4 `delay(0.5)` – это ноль миллисекунд, а не полсекунды.

6.5. Ответ: 68, 17, 12.

6.6. Ответ: в первой строке 4 числа, последнее – 62; во второй строке 4.

Подсказка: печать стоит внутри цикла после деления, поэтому 62 успевает напечататься раньше, чем условие станет ложным.

6.7. Ответ: 5 раз; ошибка в строке 8, нужно `level = 2`; без ошибки – 1 раз.

6.8. Ответ: напечатает 2; ошибка в строке 3, нужно `bpm ≥ 100`; должно получиться 5.

6.9. Ответ: 211233332312; 2 раза.

Подсказка: на 8-й секунде пульс 100, а это не меньше 100, тревога держится.

6.10. Ответ: 20 с.

Подсказка: измерения идут в 10, 15, 20, 30, 40, 45, 50 и 60 с: после тревоги следующее – через 5 с, а не через 10. Красный горит с 10-й по 20-ю и с 40-й по 50-ю секунду.

6.11. Ответ: а) 30000; б) выдох; в) 12 с.

6.12. Ответ: а) 50; б) 400 мс; в) 3 строки.

Подсказка: в строке номер k счётчик равен остатку от деления k на 100.

6.13. Ответ: 2 раза; через 300 мс.

6.14. Ответ: с 4 – «сжать», «разжать»; с 5 – ничего.

Подсказка: самая длинная серия подряд выше 500 в этих показаниях – четыре значения, с 4-го по 7-е.

К главе 7. Таблицы, графики и статистика

7.1. Ответ: среднее 83 уд/мин; размах 18 уд/мин.

7.2. Ответ: $=B6-B5$; 30 ячеек.

Подсказка: от C3 до C6 формула съехала на три строки вниз – оба адреса тоже.

7.3. Ответ: среднее без сбоя 960 ppm; медиана 940 ppm.

7.4. Ответ: 96 уд/мин.

Подсказка: задача обратная: сумма шести ($86 \cdot 6$) минус сумма пяти ($84 \cdot 5$).

7.5. Ответ: 30 %.

Подсказка: показание 800 не больше 800 – в оранжевую зону не идёт.

7.6. Ответ: 0,135 %; в 3,4 раза; на 35 %.

7.7. Ответ: 40 ppm в минуту; 800 ppm.

7.8. Ответ: в 10:30.

Подсказка: сначала скорость роста (20 ppm в минуту), потом сколько минут нужно на оставшиеся 200 ppm.

7.9. Ответ: 53214; 0,38 с.

7.10. Ответ: 0,308; 0,31.

Подсказка: вид ячейки не меняет число внутри – формулы работают с 0,308.

7.11. Ответ: 135.

7.12. Ответ: 245.

Подсказка: в третьем утверждении разницу 8 делят на 88 – выходит около 9 %; в четвёртом общее среднее считают через суммы: $(2640 + 1600) : 50 = 84,8$.

7.13. Ответ: 80,8 уд/мин.

7.14. Ответ: 1140 ppm.

К главе 8. Свет, звук и колебания в медицине

8.1. Ответ: а) 0,25 мкс; б) 4000.

8.2. Ответ: а) 0,5 мс; б) 100.

8.3. Ответ: 0,44 мм.

8.4. Ответ: 7 МГц.

Подсказка: задача обратная – $f = \nu : \lambda$, а длину волны сначала переведи в метры: 0,22 мм = 0,000 22 м.

8.5. Ответ: 58 мкс.

Подсказка: теперь ищем время: звук прошёл две глубины, $t = 2s : \nu$.

8.6. Ответ: 50 мм.

Подсказка: разница во времени двух эхо – это звук, прошедший толщину мышцы туда и обратно.

8.7. Ответ: 25 см.

8.8. Ответ: 470 мкс.

Подсказка: ладони ровно в 8 см соответствует эхо через 470,6 мкс; при границе 471 насос сработал бы и на 8,007 см, поэтому округляем вниз.

8.9. Ответ: 630.

Подсказка: уменьшить на 1,5 % – умножить на 0,985.

8.10. Ответ: 4 мм.

Подсказка: 12,5 % – это 100 %, поделённые пополам три раза.

8.11. Ответ: 180 Н.

8.12. Ответ: 7,5 см.

Подсказка: в правиле рычага неизвестно плечо, а не сила: $6 \cdot 1,5 = 1,2 \cdot l$.

8.13. Ответ: 15,7 мм.

Подсказка: нить вытягивает поворот вала: $120 - 30 = 90^\circ$.

8.14. Ответ: а) 82° ; б) 1,46 мс; в) 0,14 с.

Подсказка: из $s = \alpha : 360 \cdot 2\pi r$ выходит $81,85^\circ$; при 81° нить вытянется меньше 20 мм, поэтому округляем вверх.

Всё, что может понадобиться под рукой во время решения: числа и нормы, которые используются в книге, перевод единиц, формулы и правила из глав, мини-шпаргалки по Arduino, машинам состояний, симулятору, таблицам и Python, 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. Нормы для человека здесь примерные: они нужны для задач, а не для того, чтобы судить о чьём-то здоровье.

Таблица С.1

Величина	Значение
Пульс (ЧСС) в покое у взрослых	примерно 60–90 ударов в минуту
Пульс (ЧСС) в покое у школьников 11–14 лет	примерно 70–100 ударов в минуту
Частота дыхания в покое у взрослых	примерно 12–20 вдохов в минуту
Время реакции человека на вспышку света	примерно 0,2–0,3 с
Ритмы ЭЭГ: дельта / тета	примерно 0,5–4 Гц / 4–8 Гц
Ритмы ЭЭГ: альфа / бета	примерно 8–13 Гц / 13–30 Гц
Сила биосигналов на коже: ЭКГ / ЭМГ / ЭЭГ	около 1 мВ / до нескольких мВ / десятки мкВ
Углекислый газ CO ₂ в воздухе / в выдохе	около 0,04 % (400 ppm) / около 4 %
Arduino Uno: питание / значения <code>analogRead</code>	5 В / от 0 до 1023
Arduino Uno: значения <code>analogWrite</code> / угол сервопривода	от 0 до 255 / от 0 до 180°
Светодиод: рабочий ток / напряжение на нём	около 20 мА / около 2 В у красного, около 3 В у синего и белого
Скорость звука в воздухе / в мягких тканях тела	около 340 м/с / около 1540 м/с
Звук, который слышит человек	примерно от 20 до 20 000 Гц; выше – ультразвук
Скорость света	около 300 000 км/с
Видимый свет, длина волны	380–780 нм (границы включены); 1 мкм = 1000 нм, то есть 0,38–0,78 мкм

Границы ритмов ЭЭГ в разных учебниках немного разные: где-то альфа-ритм – 8–12 Гц, где-то 8–13 Гц, а тета – 4–7 или 4–8 Гц (в курсе встречаются оба варианта). В задаче всегда смотри, какие границы даны в условии. Пульс в покое в курсе называют «60–100 ударов в минуту».

Перевод единиц

Таблица С.2

Что	Как переводить
Время	1 мин = 60 с · 1 с = 1000 мс · 1 мс = 1000 мкс
Частота	1 Гц = 1 колебание в секунду · 1 кГц = 1000 Гц · 1 МГц = 1000 кГц
Напряжение	1 В = 1000 мВ · 1 мВ = 1000 мкВ
Ток	1 А = 1000 мА
Сопротивление	1 кОм = 1000 Ом · 1 МОм = 1000 кОм
Длина	1 м = 100 см = 1000 мм · 1 мм = 1000 мкм · 1 мкм = 1000 нм
Информация	1 байт = 8 бит · 1 Кбайт = 1024 байта
Доля газа в воздухе	1 % = 10 000 ppm (ppm – миллионная доля)

Совет. Приставки везде одни и те же: «кило» – в тысячу раз больше, «милли» – в тысячу раз меньше, «микро» – в миллион раз меньше. Запомнил один раз – переводить и секунды, и вольты, и метры.

Формулы и правила

Если в условии задачи правило записано по-своему, решай по условию.

Пульс и ЧСС

ЧСС (частота сердечных сокращений) – сколько раз сердце сокращается за минуту. Её считают в ударах в минуту (уд/мин, по-английски BPM).

Насчитал N ударов за t секунд – умножь число ударов на 60 и раздели на время:

$$\text{ЧСС} = \frac{N \cdot 60}{t}.$$

18 ударов за 15 секунд: $\frac{18 \cdot 60}{15} = 72$ уд/мин.

Знаешь время между двумя соседними ударами (**RR-интервал**) – раздели 60 секунд на этот интервал:

$$\text{ЧСС} = \frac{60}{T_{RR} \text{ (в секундах)}} = \frac{60\,000}{T_{RR} \text{ (в миллисекундах)}}.$$

Удары каждые 0,8 с: $60 : 0,8 = 75$ уд/мин. Удары каждые 750 мс: $60\,000 : 750 = 80$ уд/мин.

Лента ЭКГ. Если в условии сказано, что бумага движется со скоростью 25 мм/с, то одна маленькая клетка (1 мм) – это 0,04 с, а большая (5 мм) – 0,2 с. Расстояние между соседними пиками 20 мм – это $20 : 25 = 0,8$ с, то есть 75 уд/мин.

Совет. Между первым и последним из N пиков помещается $N - 1$ интервал, а не N . Пять пиков на графике – это четыре промежутка между ударами.

Период, частота, амплитуда

Период T – время одного полного колебания, в секундах. **Частота** f – сколько колебаний проходит за одну секунду, в герцах (Гц). Это две стороны одного и того же:

$$f = \frac{1}{T}, \quad T = \frac{1}{f}, \quad f = \frac{N}{t}.$$

Здесь N – число колебаний, t – время, за которое они прошли. Альфа-волна с периодом 0,1 с имеет частоту $1 : 0,1 = 10$ Гц. Частоте 50 Гц соответствует период $1 : 50 = 0,02$ с = 20 мс.

Размах сигнала – наибольшее значение минус наименьшее. **Амплитуда** – насколько сигнал отходит от своего среднего уровня (если сигнал качается около нуля, это то же, что расстояние от нуля до вершины). У ровного симметричного сигнала амплитуда равна половине размаха:

$$A = \frac{\max - \min}{2}.$$

Сигнал качается от 380 до 820: размах $820 - 380 = 440$, амплитуда $440 : 2 = 220$, средний уровень $(820 + 380) : 2 = 600$.

В задачах про мышцы под «амплитудой ЭМГ» иногда понимают размах или просто наибольшее значение. Как именно считать, всегда сказано в условии.

Оцифровка: отсчёты, АЦП, объём данных

Датчик выдаёт плавно меняющееся напряжение, а плата измеряет его через равные промежутки времени. Одно измерение называется **отсчётом**, число отсчётов за секунду – **частотой дискретизации** f_d .

Таблица С.3

Что найти	Формула	Пример
Промежуток между отсчётами	$\Delta t = \frac{1}{f_d}$	100 Гц: $1 : 100 = 0,01$ с = 10 мс
Число отсчётов за время t	$n = f_d \cdot t$	100 Гц за 30 с: 3000 отсчётов
Поток данных, бит в секунду	каналы · бит на отсчёт · f_d	1 канал, 10 бит, 100 Гц: 1000 бит/с
Объём записи	поток · время	1000 бит/с за минуту: 60 000 бит = 7500 байт

Правило частоты дискретизации (его называют теоремой Котельникова): чтобы не потерять колебания частотой f , измерять нужно больше чем $2f$ раз в секунду. Хочешь записать бета-ритм до 30 Гц – нужно больше 60 отсчётов в секунду.

АЦП (аналого-цифровой преобразователь) Arduino Uno превращает напряжение от 0 до 5 В в целое число от 0 до 1023 – всего 1024 ступеньки, по одной на каждые примерно 4,9 мВ. Обратный пересчёт:

$$U = N \cdot \frac{5}{1023}.$$

`analogRead` вернул 614: $614 \cdot 5 : 1023 \approx 3,0$ В. В некоторых задачах делят не на 1023, а на 1024 – разница крошечная, но ответ считай так, как сказано в условии.

Функция `map` пересчитывает число из одного диапазона в другой, например показание датчика в угол сервопривода:

$$\text{map}(x, a, b, c, d) = \frac{(x - a) \cdot (d - c)}{b - a} + c.$$

Arduino отбрасывает дробную часть, а не округляет. `map(512, 0, 1023, 0, 180)`: $512 \cdot 180 : 1023 = 90,09$, то есть 90. `map(300, 0, 1023, 0, 255)`: $300 \cdot 255 : 1023 = 74,8$, то есть 74, а не 75.

Обработка сигнала: порог, пики, сглаживание

Детектирование по уровню (амплитудное), или по порогу. Выбираем уровень-порог. Удар (пик, срабатывание) засчитываем, когда сигнал **пересекает порог снизу вверх**. Пока сигнал остаётся выше порога, новых ударов не считаем, иначе один широкий пик превратится в пять ударов.

- Порог слишком низкий – шум и мелкие горбики тоже считаются ударами, ЧСС получается больше настоящей.
- Порог слишком высокий – невысокие пики теряются, ЧСС получается меньше настоящей.

Детектирование по частоте (частотное), в программе кружка – «по периоду» – когда смотрят не на высоту сигнала, а на число импульсов за секунду или на время между ними. Равное порогу значение считается или нет – решает условие.

Скользящее среднее – простой фильтр от шума. Каждую точку заменяем средним её и соседних значений, например трёх подряд (сама точка, сосед слева и сосед справа), а результат пишем под средней точкой. Ряд 500, 530, 500, 710, 500 по трём точкам превращается в 510, 580, 570 (под второй, третьей и четвёртой точками): одиночный выброс 710 сгладился. Из n чисел по три получается $n - 2$ средних, у первой и последней точек среднего нет.

Фильтр низких частот пропускает медленные изменения и гасит быстрые. Скользящее среднее – самый простой такой фильтр: чем больше точек усредняем, тем сильнее гладится сигнал, но тем сильнее срезаются пики (а окно «по последним точкам» ещё и запаздывает).

Огибающая – плавная линия, которая следует за верхушками быстрых колебаний. Её получают так: сигнал «выпрямляют» (берут модуль) и сглаживают окном длиной в один период несущей, то есть быстрых колебаний. Поэтому огибающая идёт ниже верхушек. У ЭМГ её и смотрят: насколько сильно напряжена мышца.

Электричество

Напряжение U – в вольтах (В), ток I – в амперах (А), сопротивление R – в омах (Ом).

Таблица С.4

Правило	Формула	Пример
---------	---------	--------

Правило	Формула	Пример
Закон Ома	$I = \frac{U}{R}$	5 В на 1000 Ом: $5 : 1000 = 0,005$ A = 5 мА
Резистор для светодиода	$R = \frac{U_{\text{пит}} - U_{\text{св}}}{I}$	$(5 - 2) : 0,02 = 150$ Ом, берём ближайший больший из продающихся, например 220 Ом
Мощность	$P = U \cdot I$	5 В и 0,02 А: 0,1 Вт
Два резистора подряд (последовательно)	$R = R_1 + R_2$	3 кОм и 2 кОм: 5 кОм
Делитель напряжения	$U_{\text{вых}} = U \cdot \frac{R_2}{R_1 + R_2}$	5 В, 3 кОм и 2 кОм: $5 \cdot 2 : 5 = 2$ В

Ток в формулу подставляй в амперах: 20 мА = 0,02 А. Делитель напряжения стоит почти в каждом датчике на фоторезисторе: свет меняет сопротивление, а плата видит, как меняется напряжение.

Звук, свет, волны

Таблица С.5

Что найти	Формула	Пример
Путь при движении с постоянной скоростью	$s = v \cdot t$	звук за 0,01 с в воздухе: $340 \cdot 0,01 = 3,4$ м
Расстояние по эху (дальномер, УЗИ)	$s = \frac{v \cdot t}{2}$	эхо в воздухе через 2 мс: $340 \cdot 0,002 : 2 = 0,34$ м
Длина волны	$\lambda = \frac{v}{f} = v \cdot T$	ультразвук 40 кГц в воздухе: $340 : 40\,000 = 0,0085$ м = 8,5 мм

В формуле эха делим на два: звук идёт до препятствия и обратно, а нам нужно расстояние только в одну сторону. На УЗИ-аппарате так же, только скорость звука в теле около 1540 м/с: эхо через 0,0001 с означает глубину $1540 \cdot 0,0001 : 2 = 0,077$ м, почти 8 см.

Графики, среднее, проценты

Цена деления шкалы: возьми два соседних подписанных значения, вычти меньшее из большего и раздели на число делений между ними. Подписи 0 и 1 с, между ними 5 делений – одно деление $1 : 5 = 0,2$ с.

Таблица С.6

Что	Как найти
Среднее арифметическое	сложить все значения и разделить на их количество
Максимум и минимум	наибольшее и наименьшее значение

Что	Как найти
Размах	максимум минус минимум
Медиана	упорядочить по возрастанию и взять число посередине; если чисел чётное количество – среднее двух средних

Пульс за пять дней: 72, 75, 71, 90, 74. Среднее: $(72 + 75 + 71 + 90 + 74) : 5 = 382 : 5 = 76,4$ уд/мин. Медиана: 74. Размах: $90 - 71 = 19$.

Проценты. p процентов от числа A : $A \cdot \frac{p}{100}$. На сколько процентов выросло:

$$\frac{\text{новое} - \text{старое}}{\text{старое}} \cdot 100\%.$$

Пульс вырос с 70 до 84 уд/мин: $(84 - 70) : 70 \cdot 100\% = 20\%$. Проценты считают от того, что было **до** изменения.

Мини-шпаргалки по инструментам

Arduino и C++

Программа для Arduino (её называют скетчем) состоит из двух частей: `setup()` выполняется один раз при включении, `loop()` повторяется по кругу, пока есть питание. После `//` идёт комментарий: плата его не читает. Каждая команда кончается точкой с запятой.

```
const int LED = 13;           // светодиод - на выводе 13
const int SENSOR = A0;        // датчик - на аналоговом входе A0
int value = 0;                 // переменная для показаний

void setup() {                 // выполняется один раз при включении
  pinMode(LED, OUTPUT);        // вывод 13 работает как выход
  Serial.begin(9600);          // включаем связь с компьютером
}

void loop() {                  // повторяется по кругу, пока есть питание
  value = analogRead(SENSOR);   // число от 0 до 1023
  Serial.println(value);        // в монитор порта и на график
  if (value > 600) {            // выше порога?
    digitalWrite(LED, HIGH);    // да - включить светодиод
  } else {
    digitalWrite(LED, LOW);     // нет - выключить
  }
  delay(20);                   // пауза 20 мс
}
```

Таблица С.7

Команда	Что делает
<code>pinMode(13, OUTPUT)</code>	вывод 13 будет выходом (INPUT – входом)
<code>digitalWrite(13, HIGH)</code>	подать на вывод 5 В (LOW – 0 В)

Команда	Что делает
<code>digitalRead(2)</code>	прочитать кнопку: HIGH или LOW
<code>analogRead(A0)</code>	измерить напряжение на входе A0: число от 0 до 1023
<code>analogWrite(9, 128)</code>	«полсилы» на выводе 9: число от 0 до 255
<code>delay(500)</code>	подождать 500 мс, ничего не делая
<code>millis()</code>	сколько миллисекунд прошло с включения платы
<code>map(v, 0, 1023, 0, 180)</code>	пересчитать число из одного диапазона в другой
<code>Serial.println(v)</code>	отправить число на компьютер и перейти на новую строку

Массив и цикл. Массив – несколько значений под одним именем, номера начинаются с нуля.

```
int leds[] = {2, 3, 4, 5};      // массив: номера четырёх выводов
for (int i = 0; i < 4; i++) {   // i = 0, 1, 2, 3
    pinMode(leds[i], OUTPUT);
    digitalWrite(leds[i], HIGH); // leds[0] - это вывод 2
}
```

Пульсометр: ЧСС по времени между ударами. Удар засчитываем, когда сигнал пересёк порог снизу вверх, а время берём из `millis()`.

```
const int THRESHOLD = 600;      // порог: выше него - удар
unsigned long lastBeat = 0;     // когда был прошлый удар, мс
bool above = false;            // сигнал сейчас выше порога?

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

void loop() {
    int v = analogRead(A0);
    if (v > THRESHOLD && !above) { // пересёк порог снизу вверх
        above = true;
        unsigned long now = millis(); // сколько мс прошло с включения
        if (lastBeat > 0) {           // первый удар только запоминаем
            unsigned long rr = now - lastBeat; // мс между ударами
            Serial.println(60000 / rr);      // ЧСС, уд/мин
        }
        lastBeat = now;
    }
    if (v < THRESHOLD) {             // опустился - ждём следующий удар
        above = false;
    }
    delay(10);
}
```

Сервопривод. Библиотека `Servo` поворачивает вал на угол от 0 до 180°. Так двигают палец бионической руки.

```
#include <Servo.h>
Servo hand;           // сервопривод по имени hand

void setup() {
  hand.attach(9);      // сигнальный провод - на вывод 9
}

void loop() {
  int v = analogRead(A0); // 0...1023
  int angle = map(v, 0, 1023, 0, 180); // пересчёт в угол 0...180°
  hand.write(angle);      // повернуть
  delay(15);
}
```

Монитор порта и плоттер. `Serial.println` отправляет числа на компьютер. В мониторе порта они бегут столбиком, а плоттер рисует из них график – так сигнал видно глазами. Чтобы на графике было две линии, выводи два числа в одной строке через запятую: `Serial.print(a); Serial.print(","); Serial.println(b);` (в Arduino IDE сойдёт и пробел, а в Tinkercad курс пока-зывает запятую). Если в симуляторе линии не разделились, посмотри пример вывода не-скольких сигналов в курсе сферы.

Совет. Ошибки, на которых спотыкаются чаще всего: `=` кладёт значение в переменную, `a =` сравнивает; забыта точка с запятой в конце команды; забыт `Serial.begin(9600)` в `setup()` – и монитор порта молчит; в `for (int i = 0; i < 4; i++)` последнее значение `i` – 3, а не 4; `60000 / rr` в целых числах отбрасывает дробную часть.

Машина состояний и ПРИМС

ПРИМС – «программирование расширенных иерархических машин состояний». Это визуаль-ный язык: программу не пишут строчками, а рисуют схемой из состояний и стрелок. Рисуют её в Кибериаде IDE, а она умеет сделать из схемы код на C++ для Arduino.

Таблица С.8

Элемент	Что это	Пример
Состояние	режим работы устройства	«спокойствие», «тревога», «пауза»
Событие	то, что случилось и заставило реагировать	«нажата кнопка», «прошло 2 с», «пришло показание»
Условие	фильтр: переход сработает, только если оно верно	[ЧСС > 100]
Действие	что устройство делает	«включить красный светодиод»
Переход	смена состояния, на схеме – стрелка	«спокойствие» → «тревога»

Подпись на стрелке в нашей книге всегда одна и та же: **событие [условие] / действие**. Например: «пришло показание [ЧСС > 100] / включить красный».

- В каждый момент устройство находится **в одном** состоянии, не в двух сразу.
- Действие можно выполнить на входе в состояние (один раз, когда попали в него), на выходе (когда уходим) или на переходе.
- Если условие на стрелке ложно, устройство остаётся в том же состоянии.
- Имя состояния отвечает на вопрос «что сейчас происходит?». «Состояние 1» – плохое имя, «ждём удар» – хорошее.

Как проследить работу машины: заведи таблицу «шаг – текущее состояние – событие – условие верно? – новое состояние – действие» и заполняй строку за строкой. Не прыгай через шаги.

Симулятор и макетная плата

Так было в сентябре 2026 года. Интерфейс симуляторов может обновиться – смысл останется.

- **Tinkercad**. Курс сферы даёт ссылку на класс и логин – входи по ней. В редакторе схем две главные кнопки: «Код» (писать программу) и «Начать моделирование» (запустить). В панели компонентов выбери показ всех компонентов, иначе видна только часть. Монитор порта и график – внизу панели кода. В общем классе схемы не сохраняются между заходами, поэтому текст программы сохраняй себе в файл.
- **Wokwi** (wokwi.com) работает без установки. Выбираешь плату Arduino Uno, добавляешь детали на схему, пишешь код и запускаешь моделирование.
- **Макетная плата**. Длинные ряды по краям – шины питания: все отверстия ряда соединены между собой. В середине отверстия соединены короткими столбиками по пять. Две ножки в одном столбике соединены, в разных – нет.
- **Светодиод** пропускает ток в одну сторону. Длинная ножка (анод) – к плюсу, короткая – к минусу (GND). Без резистора светодиод сгорит; симулятор в этом случае выдаёт предупреждение о слишком большом токе.
- **Выводы Arduino Uno**. Цифровые 0–13, на 13-м есть встроенный светодиод. Аналоговые входы A0–A5 – для датчиков. «Полсилы» через `analogWrite` умеют только выводы, помеченные волной ~: 3, 5, 6, 9, 10, 11.
- **Потенциометр** – ручка с тремя ножками: крайние к 5 В и GND, средняя – к аналоговому входу. В симуляторе им удобно заменять датчик: крутишь ручку – «сигнал» растёт.

Табличный редактор

Подойдёт любой: LibreOffice Calc, Excel, Google Таблицы. Формула начинается со знака `=`. В русских версиях аргументы разделяют точкой с запятой.

Таблица С.9

Формула	Что считает
<code>=СРЗНАЧ(В2:В31)</code>	среднее по ячейкам от В2 до В31 (в английской версии <code>AVERAGE</code>)
<code>=МАКС(В2:В31) , =МИН(В2:В31)</code>	наибольшее и наименьшее (<code>MAX</code> , <code>MIN</code>)

Формула	Что считает
=СУММ(B2:B31)	сумму (SUM)
=МЕДИАНА(B2:B31)	медиану (MEDIAN)
=СЧЁТЕСЛИ(B2:B31;">100")	сколько значений больше 100 (COUNTIF)
=ОКРУГЛ(C2;1)	округлить до одного знака после запятой (ROUND)
=(C2-B2)/B2*100	на сколько процентов C2 больше B2

Формулу пишут один раз, а потом тянут за правый нижний уголок ячейки вниз – она сама пересчитается для каждой строки. График: выдели столбцы с данными и вставь диаграмму «линия» – для сигнала во времени, «гистограмма» – для сравнения нескольких значений.

Python

Python по желанию, но им удобно проверять расчёты. Всё ниже работает в любом онлайн-интерпретаторе Python 3. После # – комментарий.

```
pulse = [72, 75, 71, 90, 74]      # список: пульс за пять дней
print(len(pulse))                # 5 – сколько чисел
print(max(pulse), min(pulse))    # 90 71
print(sum(pulse) / len(pulse))   # 76.4 – среднее
print(round(76.4))               # 76 – до целых
print(round(3.14159, 2))         # 3.14 – до сотых
```

Счёт ударов по порогу – тот же алгоритм, что и в скетче пульсометра:

```
signal = [300, 450, 720, 650, 380, 310, 690, 760, 420, 330]
threshold = 600                 # порог
beats = 0                       # счётчик ударов
above = False                   # сейчас выше порога?
for v in signal:
    if v > threshold and not above: # пересёк порог снизу вверх
        beats = beats + 1
        above = True
    if v < threshold:
        above = False
print(beats)                    # 2
```

Скользящее среднее по трём точкам:

```
a = [500, 530, 500, 710, 500]
smooth = []
for i in range(len(a) - 2):
    smooth.append((a[i] + a[i + 1] + a[i + 2]) / 3)
print(smooth)                  # [510.0, 580.0, 570.0]
```

Словарик

Таблица С.10

Слово	Что значит
Актуатор	исполнительное устройство: мотор, сервопривод, светодиод, пищалка. Датчик чувствует, актуатор делает
Альфа-ритм	волны ЭЭГ частотой примерно 8–13 Гц. Хорошо видны, когда человек спокойно сидит с закрытыми глазами
Амплитуда	насколько сигнал отходит от своего среднего уровня
Аналоговый сигнал	меняется плавно и может принимать любые значения: напряжение от датчика
АЦП	аналого-цифровой преобразователь: превращает напряжение в число. У Arduino Uno – от 0 до 1023
Бета-ритм	волны ЭЭГ частотой примерно 13–30 Гц. Сильнее, когда человек занят делом и сосредоточен
Биосигнал	сигнал, который даёт живой организм: пульс, дыхание, электрическая работа мышц, сердца, мозга
БОС-тренинг	тренировка с биологической обратной связью: прибор показывает человеку его сигнал, и тот учится им управлять
Датчик (сенсор)	прибор, который превращает то, что происходит вокруг (свет, давление, напряжение мышцы), в электрический сигнал
Детектирование	поиск в сигнале нужного события: удара сердца, сжатия мышцы
Дискретизация	измерение сигнала через равные промежутки времени
Калибровка	подстройка прибора под конкретного человека и условия: какой сигнал считать покоем, а какой – сжатием
Кибериада IDE	программа, в которой рисуют машины состояний на языке ПРИМС
Макетная плата	плата с отверстиями, чтобы собирать схему без пайки
Машина состояний	способ описать поведение устройства через режимы (состояния) и переходы между ними

Слово	Что значит
Микроконтроллер	маленький компьютер на одной микросхеме. Сердце платы Arduino
Модуляция	способ «нагрузить» сообщение на сигнал-переносчик (несущую), меняя его размах или частоту
Нейроинтерфейс	устройство, которое связывает сигналы мозга или нервов с компьютером или прибором
Нейрон	нервная клетка. Передаёт сообщения электрическими импульсами
Огибающая	плавная линия вдоль вершушек быстрых колебаний: сигнал выпрямляют (модуль) и сглаживают окном в период несущей, поэтому она ниже вершушек
Отсчёт	одно измерение сигнала
Период	время одного полного колебания
Плоттер	окно, которое рисует график из чисел, пришедших с платы
Порог	уровень сигнала: выше него считаем, что событие случилось
Потенциометр	резистор с ручкой: крутишь – меняется напряжение на средней ножке
ПРИМС	программирование расширенных иерархических машин состояний: визуальный язык, на котором рисуют программы
Пульсовая волна	толчок крови по артериям после каждого сокращения сердца
Резистор	деталь, которая ограничивает ток
Сервопривод	мотор, который поворачивается на заданный угол от 0 до 180°
Скетч	программа для Arduino
Скользящее среднее	сглаживание: каждую точку заменяют средним её и соседних значений (окно с центром в точке)
Спектр	из каких частот складывается сигнал и какой вклад у каждой
Фильтр	обработка, которая убирает из сигнала лишнее: шум, слишком быстрые или слишком медленные колебания
Фоторезистор	резистор, у которого сопротивление зависит от освещения

Слово	Что значит
Частота	сколько колебаний проходит за одну секунду, в герцах
ЧСС	частота сердечных сокращений, в ударах в минуту
Шум	случайные помехи, которые примешиваются к полезному сигналу
ШИМ	быстрое включение и выключение, которое делает «полсилы»: так analogWrite притушивает светодиод
ЭКГ	электрокардиограмма: запись электрической работы сердца
Электрод	контакт, который прикладывают к коже, чтобы снять биосигнал
ЭМГ	электромиограмма: запись электрической работы мышц. Чем сильнее напряжена мышца, тем больше размах
ЭЭГ	электроэнцефалограмма: запись электрической работы мозга. Самый слабый из этих сигналов
ppm	миллионная доля: так датчики CO ₂ показывают, сколько углекислого газа в воздухе
RR-интервал	время между двумя соседними ударами сердца

Шпаргалка: всё «Запомни» из глав

Глава 1. Тело человека как система сигналов

Запомни. Орган делает свою работу, система органов – общее дело. Инженерная «карта» тела: органы чувств – датчики, нервы – провода, мозг – контроллер, мышцы – моторы, сердце с сосудами – насос с трубами, лёгкие – вентиляция.

Запомни. Артерия – от сердца, вена – к сердцу, и сколько в крови кислорода, здесь неважно. Правая половина сердца – кровь бедная кислородом, левая – богатая. Объём за минуту = ударный объём · ЧСС.

Запомни. Вдох делают мышцы, главная из них – диафрагма, спокойный выдох идёт сам. В альвеолах кислород уходит в кровь, углекислый газ – в воздух. Углекислого газа в выдохе примерно в 100 раз больше, чем во вдохе.

Запомни. Дендриты принимают, аксон передаёт, по нейрону сигнал идёт в одну сторону. Нерв – пучок аксонов. Безусловный рефлекс есть с рождения, условный вырабатывается. Миелин ускоряет сигнал.

Запомни. Мышца только тянет, поэтому мышцы работают парами. Бицепс сгибает руку в локте, трицепс разгибает. Сильнее напряжение – больше амплитуда ЭМГ, расслабленная мышца – почти ровная линия.

Запомни. Рецептор превращает свет, звук, запах или касание в нервные импульсы, а ощущение складывается в мозге. Колбочки – цвет, палочки – сумерки. В ухе два датчика: улитка – звук, вестибулярный аппарат – равновесие.

Запомни. Внутри клетки в покое около -70 мВ; нервный импульс – всплеск примерно до $+30$ мВ длиной около 1 мс. На коже: ЭКГ – около 1 мВ, ЭМГ – от десятков микровольт до нескольких милливольт, ЭЭГ – десятки микровольт. $1 \text{ мВ} = 1000 \text{ мкВ}$.

Запомни. ЭКГ – сердце, ЭМГ – мышцы, ЭЭГ – мозг: это электричество самого тела, его ловят электроды. Пульс на пальце – свет, дыхание – растяжение груди или CO_2 у носа: здесь датчик сам делает электрический сигнал из неэлектрического.

Глава 2. Пульс и сердце: считаем по графику

Запомни. ЧСС – удары за минуту. Считал 15 секунд – умножь на 4, считал 30 – умножь на 2, в общем случае $\text{ЧСС} = N \cdot 60 / t$. Спокойный пульс школьника – примерно 70–100 уд/мин.

Запомни. $\text{ЧСС} = 60 : T$, где T – в секундах, или $60\,000 : T$, где T – в миллисекундах. Короче интервал – чаще пульс. Уд/мин в герцы – раздели на 60.

Запомни. N пиков – $N - 1$ промежутков. Средний интервал = (время последнего пика – время первого) : $(N - 1)$. На ЭКГ считаем зубцы R, на пульсовой волне – только высокие пики, без вторых горбиков.

Запомни. «На сколько» – вычитаем, «во сколько раз» – делим, «на сколько процентов» – прибавку делим на значение «до». Пульс падает после нагрузки – «до» равно самому высокому значению.

Запомни. Минутный объём = ударный объём · ЧСС. Ударный объём в миллилитрах даёт ответ в миллилитрах, для литров раздели на 1000.

Запомни. Оптический датчик – светодиод плюс фотоприёмник. Удар – в сосудах больше крови – света до приёмника доходит меньше. Пики на графике – удары, время между пиками даёт пульс.

Глава 3. Мышцы и мозг: ЭМГ и ЭЭГ

Запомни. ЭМГ ловит сразу сотни волокон, поэтому она «лохматая» и без чётких пиков. Сильнее напряг мышцу – шире полоса. Огибающая рисует ширину полосы одной плавной линией: по ней и работают все программы этой главы.

Запомни. Порог ставят между покоем и сжатием: выше самого большого значения в покое и ниже самого маленького при сжатии, лучше посередине. Срабатывания считают по переходам через порог снизу вверх. Сигнал дрожит у порога – бери два порога: включать по верхнему, выключать по нижнему.

Запомни. Пропорциональное управление:
 $\text{угол} = (\text{огibaющая} - \text{покой}) \cdot 180 : (\text{сжатие} - \text{покой})$. Arduino-команда `map` отбрасывает дробную часть, а `constrain` не пускает угол за 0 и 180°. В мио-армрестлинге смотрят на разность двух сигналов, а маленькую разность гасит мёртвая зона.

Запомни. Альфа – примерно 8–13 Гц, покой с закрытыми глазами. Бета – примерно 13–30 Гц, внимание и работа. Частота: $f = N : t$ (волны за время) или $f = 1 : T$ (через период в секундах). Открыл глаза или задумался – альфа слабеет.

Запомни. БОС – человек видит свой биосигнал и учится им управлять. Петля: датчик → обработка → экран → человек → датчик. В задачах про тренажёр иди по секундам по порядку и сверяйся с условием: «меньше» там или «не меньше», может ли ракета уйти ниже земли.

Глава 4. Сигналы и обработка данных

Запомни. Сигнал – то, что меняется и этим несёт сообщение. Шум – лишнее, что к нему примешалось: движение, моргание, наводка от сети 50 Гц, плохой контакт. Датчик пишет сумму того и другого.

Запомни. Размах = максимум – минимум. Амплитуда – половина размаха, средний уровень – середина между максимумом и минимумом. Период – секунды на одно колебание, частота – колебания за секунду: $f = 1:T$.

Запомни. Отсчёт – одно измерение. Частота дискретизации – отсчётов за секунду. Шаг $\Delta t = 1:f_d$, отсчётов за время t : $n = f_d \cdot t$. Измерять надо чаще, чем $2f$, где f – самые быстрые колебания, которые нужно сохранить.

Запомни. Срабатывание – сигнал был не выше порога и стал выше. Один пик – одно срабатывание, сколько бы отсчётов он ни занимал. Время выше порога = число отсчётов выше порога · шаг. Равное порогу значение считается или нет – решает условие («выше» – не считается, «не меньше» – считается). Частота по номерам отсчётов: разница номеров делится на f_d – получаются секунды.

Запомни. Скользящее среднее: окошко из k чисел едет по ряду, каждое положение даёт одно среднее, и его пишут под средней точкой окошка (для окна 3 – сама точка и соседи слева и справа). Из n чисел получается $n - k + 1$, на краях средних нет. Шире окошко – глаже линия, ниже пики; окно «по последним» ещё и запаздывает.

Запомни. Спектр – из каких частот сложен сигнал и сколько каждой. Фильтр низких частот пропускает медленное и гасит быстрое, высоких – наоборот, полосовой оставляет полосу. Скользящее среднее – простейший фильтр низких частот.

Запомни. Бит – одна двоичная цифра, b бит дают 2^b уровней, 10 бит – 1024 (от 0 до 1023). Поток = каналы · бит на отсчёт · отсчётов в секунду. 1 байт = 8 бит, 1 Кбайт = 1024 байта.

Глава 5. Электричество и датчики

Запомни. Ток идёт только по замкнутой цепи. Напряжение U – «напор», в вольтах (В). Сила тока I – «сколько течёт», в амперах (А) и миллиамперах: $1 \text{ А} = 1000 \text{ мА}$.

Запомни. Закон Ома: $I = U : R$, $U = I \cdot R$, $R = U : I$. Подставляй вольты, амперы и омы. Последовательно стоящие сопротивления складываются.

Запомни. Светодиод – только с резистором. $R = (U_{\text{пит}} - U_{\text{св}}) : I$, ток – в амперах. Из набора бери ближайший **больший** номинал.

Запомни. Датчик переводит величину (свет, тепло, нажатие, биосигнал) в напряжение. Если датчик меняет сопротивление, к нему добавляют резистор. Биосигналы крошечные – перед платой их усиливают: $U_{\text{вых}} = K \cdot U_{\text{вх}}$.

Запомни. $U_{\text{вых}} = U \cdot R_2 : (R_1 + R_2)$, где R_2 – резистор между выходом и GND. Одинаковые резисторы делят напряжение пополам.

Запомни. `analogRead` : 0 В → 0, 5 В → 1023, ступенька около 4,9 мВ. Из числа в вольты: $U = N \cdot 5 : 1023$. Из вольт в число: $N = U \cdot 1023 : 5$.

Запомни. $t = C : I$: ёмкость в мА·ч делим на ток в мА – получаем часы. Если ток меняется, сначала найди средний ток.

Глава 6. Программирование: Arduino и машины состояний

Запомни. Алгоритм – шаги по порядку, исполнитель – тот, кто их выполняет и умеет только свои команды. Программу для учебного исполнителя проверяй прогоном: выпиши число после каждой команды. Ищешь программу – иди от конца к началу.

Запомни. `=` кладёт в переменную, старое значение пропадает. Целое делим на целое – получаем целое, дробь отбрасывается: $82 / 4$ даёт 20. `%` – остаток: $82 \% 4$ даёт 2.

Запомни. `if (вопрос) { ... } else { ... }` – развилка на две дороги. `=` сравнивает, `=` кладёт. В цепочке `if ... else if ... else` срабатывает только первая подходящая ветка. Внимательно смотри на границу: > 100 и ≥ 100 по-разному поступают с числом 100.

Запомни. `for` – повторить заданное число раз, `while` – повторять, пока условие верно. Условие проверяется перед каждым кругом: когда оно стало ложным, цикл заканчивается сразу. Чтобы узнать, что выведет цикл, води трассировку.

Запомни. `setup()` – один раз, `loop()` – по кругу. `digitalWrite` – только HIGH (5 В) или LOW (0 В), `analogRead` – число от 0 до 1023. Время у Arduino в миллисекундах. Длительность круга `loop()` – сумма всех `delay` в нём.

Запомни. «При запуске» = `setup()` , «повторять всегда» = `loop()` , скоба «если – иначе» = `if ... else` . Что стоит внутри скобы, в тексте стоит внутри фигурных скобок. В тексте время – в миллисекундах.

Запомни. Машина состояний: в каждый момент одно состояние, переходы по событиям, условие в квадратных скобках решает, сработает ли стрелка. В коде – переменная `state` и цепочка `if ... else if` в `loop()` , время – через `millis()` - `start` , а не через `delay` .

Глава 7. Таблицы, графики и статистика

Запомни. Адрес – латинская буква столбца и номер строки. Формула начинается с `=` . `B2:B6` – ячейки с B2 по B6, обе включены, всего 5. Протянул формулу вниз – адреса в ней съехали на столько же строк.

Запомни. Среднее = сумма : количество, и оно всегда между минимумом и максимумом. Размах = максимум – минимум. Медиана – середина упорядоченного ряда, её сбой почти не сдвигает. Убрал число из ряда – дели на новое количество.

Запомни. Сравнивая дроби, уравнивай число знаков нулями: $0,30 > 0,29$. Миллисекунды → секунды: делим на 1000. Округляют по первой отброшенной цифре (5 и больше – вверх) и только в самом конце.

Запомни. $1\% = 10\,000\text{ ppm}$. На улице чуть больше 400 ppm, в задачах берут 400. Доля измерений выше порога = (сколько выше : сколько всего) · 100. «На сколько процентов больше» считают от того, с чем сравнивают: от порога, от начального значения.

Запомни. Сначала оси: что отложено, в каких единицах, сколько стоит клетка и с какого числа начинается ось. Потом уже линия. Скорость изменения = изменение : время.

Запомни. Общее среднее нескольких групп – через суммы: среднее · количество, суммы сложить, поделить на общее количество. Верный вывод опирается только на данные из таблицы и не говорит «каждый», если в таблице «в среднем».

Глава 8. Свет, звук и колебания в медицине

Запомни. $f = 1 : T$, $T = 1 : f$, число колебаний $N = f \cdot t$. Килогерцы в паре с миллисекундами, мегагерцы – с микросекундами: 4 кГц – период 0,25 мс, 5 МГц – 0,2 мкс.

Запомни. Звук – бегущие сжатия воздуха, воды или тела. Высота – это частота, громкость – амплитуда. Слышим примерно от 20 до 20 000 Гц, выше – ультразвук. Скорость звука: в воздухе около 340 м/с, в мягких тканях около 1540 м/с. Длина волны $\lambda = v : f$.

Запомни. Расстояние по эху $s = v \cdot t : 2$ – звук шёл туда и обратно. В теле 1,54 мм за микросекунду, и каждый сантиметр глубины даёт около 13 мкс задержки. В воздухе дальнометр: сантиметры = микросекунды · 0,034 : 2.

Запомни. Свет – волна, которой не нужен воздух, скорость около 300 000 км/с. Цвет – это длина волны: красный длиннее синего, инфракрасный ещё длиннее и глазу не виден. Ткани пропускают красный и ИК лучше, чем зелёный и синий. Каждый слой половинного ослабления делит свет пополам.

Запомни. 1 кг весит около 10 Н. Правило рычага: $F_1 \cdot l_1 = F_2 \cdot l_2$, плечо меряют от опоры. Несколько сил поворачивают в одну сторону – складываем их моменты. Мышцы крепятся близко к суставам: тянут сильнее груза, зато двигают руку быстро и далеко.

Запомни. Нить на диске: $s = \alpha : 360 \cdot 2\pi r$. Сигнал сервоприводу – 50 импульсов в секунду (период 20 мс), угол задаёт длина импульса. Больше радиус качалки – длиннее ход нити, но меньше сила.

Где что найти

Таблица С.11

Что	Где
Правила олимпиады, сферы, сроки	junior.ntcontest.ru
Личный кабинет участника	my.ntcontest.ru
Курс сферы «Медтех» (за тест курса +5 баллов к отбору)	academy.junior.online/event/medtech_junior , вход через «Талант»
Платформа отбора	orbita.education (заходить кнопкой из личного кабинета)
Памятки для Новосибирской области	nto.kvantorium54.ru
Симулятор Tinkercad	по ссылке на класс из курса сферы
Симулятор Wokwi	wokwi.com
Arduino IDE	по желанию, для настоящей платы; ставит педагог кружка
Кибериада IDE	ссылка и инструкция – в курсе сферы
Python	python.org или любой онлайн-интерпретатор
Анатомия и физиология человека (видеокурс из списка литературы программы сферы)	teach-in.ru/course/one-hundred-hours-of-school-biology

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.