[Глава I. Цель и задачи исследования 1](#_Toc12103)

[1 Современные технологии в бадминтоне 1](#_Toc23457)

[2 Постановка проблемы 3](#_Toc8081)

[3 Анализ статистических данных 4](#_Toc25526)

[4 Выбор решения проблемы 13](#_Toc26263)

[4.1 Лучевые и проводные системы 13](#_Toc12923)

[4.2 Системы компьютерного зрения 15](#_Toc18923)

[4.3 Система на велостате 15](#_Toc26677)

[5 Анализ применимости материала Velostat 16](#_Toc7609)

[5.1 Анализ чувствительности 17](#_Toc20853)

[5.2 Анализ быстродействия 20](#_Toc26906)

[5.2 Анализ точности 20](#_Toc14996)

**Глава I. Цель и задачи исследования**

**1 Современные технологии в бадминтоне**

Множество современных устройств не только упрощают нашу повседневную жизнь, но и помогают во многих сферах деятельности — медицине, инженерии, спорте, финансах и прочих. В спорте в том числе. Уже сложно представить просмотр теннисного матча без вызовов систем автоматического контроля линий (automatic line-calling system, ALC system). Hawk-eye (Хок-ай, Ястребиный глаз) помогает судьям принять решение при спорных ситуациях, моделируя полет игрового снаряда.

Используются технологии и в тренировочном процессе, не только упрощая его, но и делая более интересным для спортсменов. Особой популярностью пользуются умные часы — они позволяют измерить пульс, засечь время, а также собрать различную статистику, к примеру, сколько километров человек пробежал за тренировку. Таким образом, тренировочный процесс не превращается в однообразную рутину и приобретает соревновательный характер — полученные результаты можно сравнивать с результатами товарищей.

В бадминтоне на тренировках также иногда можно увидеть специализированные пушки для подачи воланов, подобно той, что представлена на рисунке 1. Они не только позволяют имитировать удары, но и позволяют спортсменам проводить «многоволанки» в одиночку, не привлекая второго игрока, набрасывающего волан. Однако стоимость большей части моделей подобного устройства превышает 100 тыс. рублей, что делает его трудно доступной для частных лиц [Topshin].



Рисунок 1 — Бадминтонная пушка SIBOASI модель B2000

Находят свое применение и различные новинки инженерии. Так, технологии дополненной реальности могут применяться в тренировочном процессе для отработки тактик [Коновалов]. Тренер прямо во время игрового процесса подсказывает спортсмену, куда лучше переместиться или ударить, позволяя наработать навык планирования игры, который крайне важен в таких высокоскоростных видах спорта, как бадминтон.

Как уже было видно выше, в спорте могут использоваться не только специализированные технологии, но и те изобретения, которые активно применяются в других сферах. Так, одним из известных примеров областей, где можно встретить окулографию, является UI/UX тестирование. Используется айтрекинг и в бадминтоне. С помощью него можно отследить, куда направлен взгляд спортсмена, и в случае необходимости тренировками расширить поле зрения [The Tech Race].



Рисунок 2 — Использование окулографии в тренировках

**2 Постановка проблемы**

Хорошего спортсмена отличает не только отличная физическая подготовка, но и умение продемонстрировать все свои навыки на соревнованиях. Однако часто сложно выложиться на свой максимум, и причины этому бывают самые разные, начиная с недосыпа из-за волнения и заканчивая болезнями.

Важно не только подготовить себя перед игрой, но хорошо выступить в процессе. Но возникает проблема, свойственная не только начинающим, но и многим опытным спортсменам — волнение. Именно оно зачастую становится преградой на пути к победе, из-за нервов человек просто не может сосредоточиться на игре.

С волнением все справляются по-разному — кто-то пьет успокоительные, кто-то больше времени уделяет размике. Редких счастливчиков эта проблема обходит стороной, а наиболее опытные научились справляться с эмоциями и направлять их в нужной русло.

Однако, помимо волнения, существует и другая проблема — потеря концентрации. Особенно она прослеживается в высокоскоростных видах спорта, которые требуют мгновенного принятия решений. Спортсмену приходится не только следить за своей техникой, но и за действиями соперника, на ходу меняя тактику, подстраиваясь под игру. Неудивительно, что после самых напряженных моментов человек устает не только физически, но и морально. После этого в партии часто могут наступить «провалы», когда спортсмен теряет много очков часто из-за самых простых ошибок. И хорошо, если он сможет прийти в себя и вернуться в игру. Но многие партии были проиграны именно из-за такой нестабильности.

Тренировочный процесс никогда не сможет сымитировать соревновательный, несмотря на возможность проведения игровых тренировок — подсознательно человек будет понимать, что на кону не стоит первенство, что это лишь очередная отработка навыков. Однако на тренировке также можно встретить проблему потери концентрации. Спортсмен может отвлечься на какие-то свои мысли и, не уделив достаточного внимания занятию, неправильно поставить себе какой-либо удар. А это в потенциале может привести к большим проблемам — переучить человека куда сложнее, чем обучить его чему-то новому.

В видах спорта, подобных теннису и бадминтону, правильность выполнения того или иного удара может определяться попаданием его в игровую зону. И, если волан или мяч летят в сетку, то это легко определить, но в залах со старой разметкой или в ходе продолжительной тренировки (когда лежащие на полу снаряды могут заслонять обзор) вопрос попадания в поле или аут может стоять наиболее остро. А во время статической отработки ударов (без перемещения человека по площадке) отвлечься на что-либо другое бывает очень легко.

Учитывая имеющиеся проблемы, легко поставить задачу разработки устройства, которое:

1. Сможет отслеживать попадание игрового снаряда в поле или в аут;
2. Сможет оповестить спортсмена при большом количестве аутов подряд.

3 Анализ статистических данных

Описанные выше проблемы возможно подкрепить собранными данными. Было проведено два опроса. В первом из них определялось общее отношение людей к современным технологиям в спорте и бадминтоне в частности. В опросе приняли участие 30 человек, их распределение по полу и возрасту представлено на рисунках 3-4. Большая часть респондентов — представители женского пола, в опросе принимали участие люди от 18 до 39 лет.

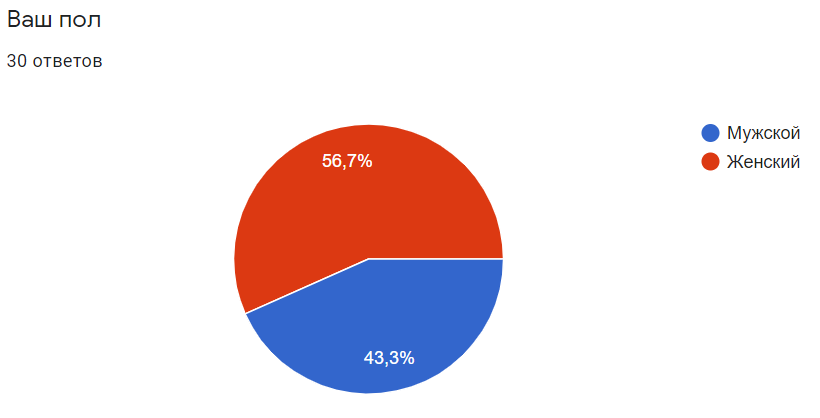


Рисунок 3 — Распределение респондентов по полу

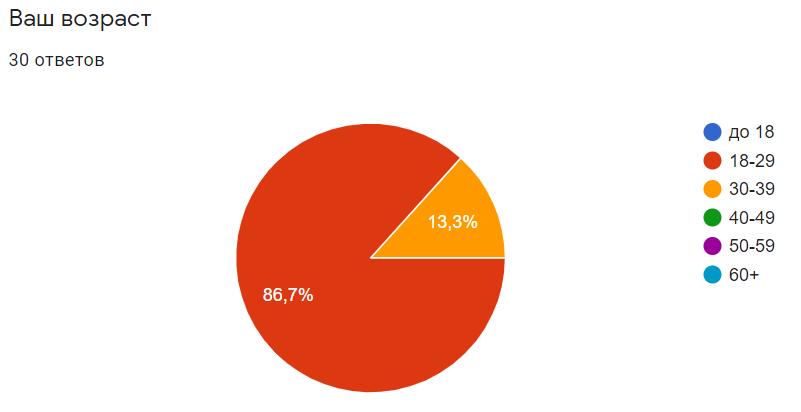


Рисунок 4 — Распределение респондентов по возрасту

Из опрошенных большая часть (26 человек из 30) занималась или занимается бадминтоном, что представлено на рисунке 5.



Рисунок 5 — Распределение респондентов по вовлеченности в бадминтон

При этом меньшая часть опрошенных считает уровень использования технологий в спорте достаточным, распределение представлено на рисунке 6.



Рисунок 6 — Распределение респондентов по вопросу, считают ли они уровень используемых технологий в спорте достаточным

Далее респондентам был предложен список из наиболее популярных тренировочных технологий:

* Автоматические системы контроля линий компьютерного зрения (Hawk-eye, Foxtenn).
* Лучевые автоматические системы контроля линий (Циклоп).
* Проводные автоматические системы контроля линий (TEL).
* Системы видео повторов.
* Окулография (отслеживание движения глаз).
* Дополненная реальность (AR).
* Виртуальная реальность (VR).
* Спортивные тренажеры (беговые дорожки, велотренажеры и проч.).

Из указанных выше технологий респондентам предлагалось выбрать те, о которых они слышали или использование которых видели. Результаты представлены на рисунках 7-8.



Рисунок 7 — Ответы респондентов на вопрос знания спортивных технологий

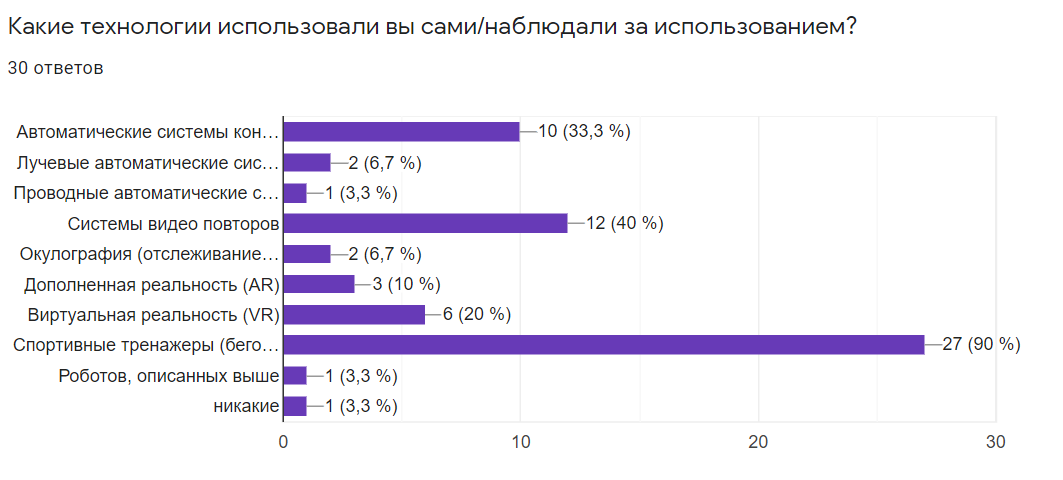


Рисунок 8 — Ответы респондентов на вопрос применения спортивных технологий

Повсеместная доступность тренажеров (не только в спортивных залах, но и на специально оборудованных площадках) определяет их лидерство. Известность видео повторов и автоматических систем контроля линий по типу Hawk-eye подтверждает то, что современные спортивные матчи, транслирующиеся по телевидению или сети интернет, уже невозможно представить без использования этих технологий.

Однако результаты опроса, представленного на рисунке 9, показывают что, несмотря на активное использование технологий на соревнованиях, в тренировочном процессе они не используются в достаточном количестве.



Рисунок 9 — Распределение респондентов по вопросу, считают ли они активность использования технологий в тренировочном процессе достаточной

Из устройств, наиболее часто используемых самими респондентами, абсолютным лидером стали «умные» часы (упомянуты в 11 письменных ответах из 24). Доступная цена, широкий спектр использования, включающий себя измерение пульса, отслеживание пройденного за день расстояния, а также удобство являются неоспоримыми преимуществами данного устройства, которые обуславливают его популярность.

Абсолютное большинство респондентов, как представлено на рисунке 10, считают, что технологии помогают разнообразить тренировки и сделать их более интересными.



Рисунок 10 — Распределение респондентов по вопросу, считают ли они , что применение технологий способно разнообразить тренировки

Результаты данного опроса показывают, что разработка новых устройств для спортивных тренировок является перспективным направлением деятельности.

Второй опрос был признан выявить, действительно ли проблема потери концентрации является большой проблемой для бадминтонистов.

В опросе приняло участие 24 человека в возрасте от 18 до 40 лет. Распределение опрошенных по полу и возрасту представлены на рисунках 11 и 12.

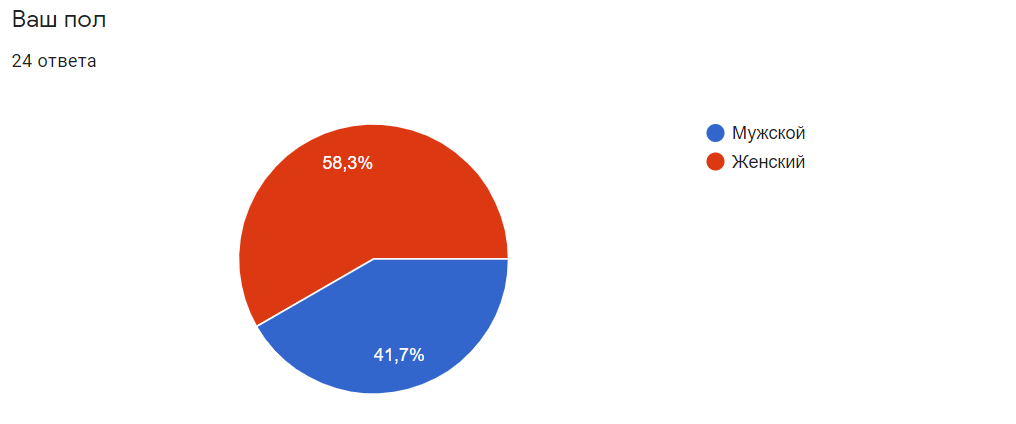


Рисунок 11 — Распределение респондентов-бадминтонистов по полу

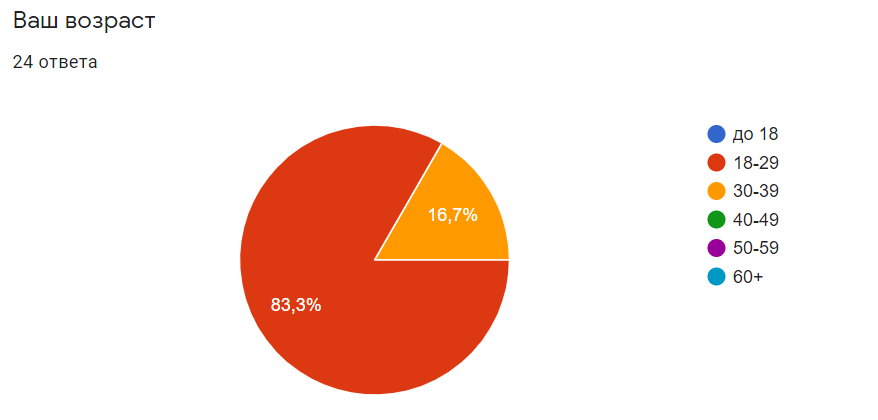


Рисунок 12 — Распределение респондентов-бадминтонистов по возрасту

Из них половина занимается данным видом спорта от 8 до 15 лет, еще 41,7% — больше 15 лет, что представлено на рисунке 13.

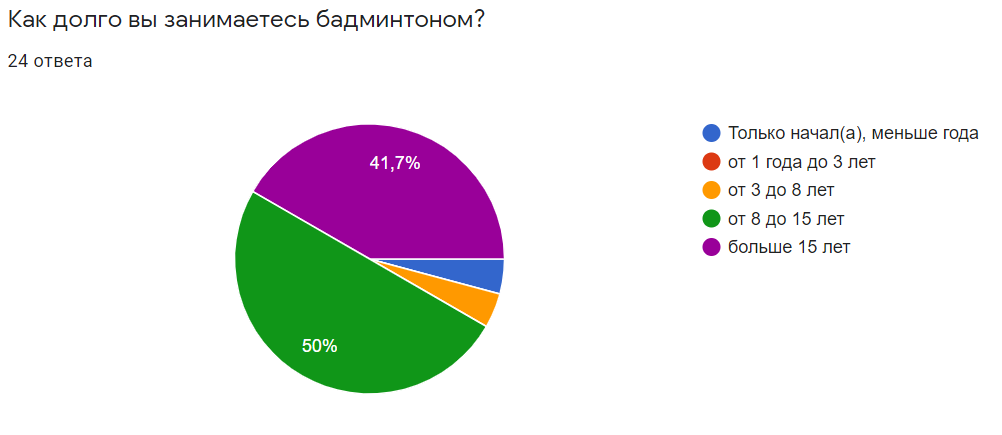


Рисунок 13 — Распределение респондентов-бадминтонистов по опыту игры

Большая часть из опрошенных когда-то занималась профессионально (принимало активное участие в соревнованиях), примерно треть занималась профессионально на момент опроса, распределение представлено на рисунке 14..

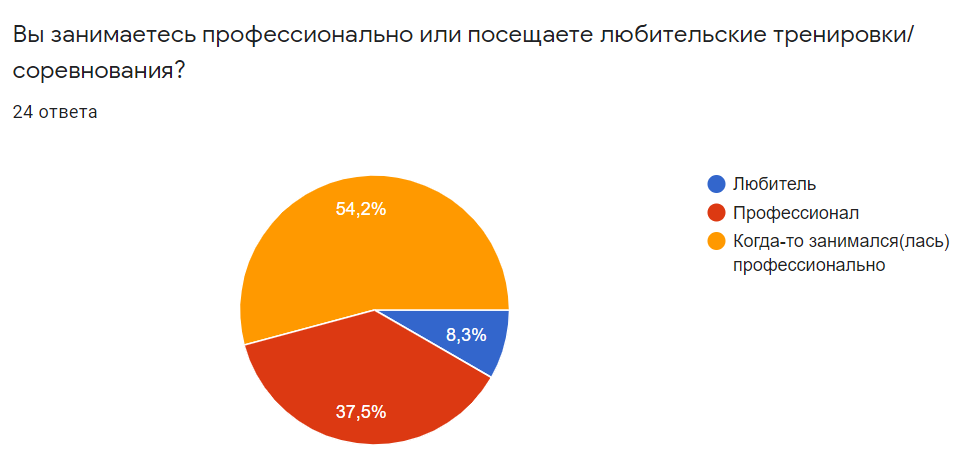


Рисунок 14 — Распределение респондентов-бадминтонистов по степени вовлеченности в данный вид спорта

Из опрошенных только один человек указал, что не сталкивался с проблемой потери концентрации, что представлено на рисунке 15.



Рисунок 15 — Распределение респондентов-бадминтонистов по частоте столкновения с проблемой потери концентрации

Как видно из рисунка 16, абсолютное большинство при этом сталкивается с данной проблемой не только во время соревнований, но и во время тренировок.



Рисунок 16 — Распределение респондентов-бадминтонистов по месту столкновения с проблемой потери концентрации

Все опрошенные отметили, что, потеря концентрации негативно сказывалась на уровне их игры. При этом большая часть отмечает причиной потери концентрации не столько физическую, сколько ментальную усталость, а также отвлечение мыслей от происходящего на игровой площадке, что представлено на рисунке 17.



Рисунок 17 — Ответы респондентов-бадминтонистов на вопрос о причинах потери концентрации

Наконец, как представлено на рисунке 18, большая часть респондентов пытается справится с проблемой своими силами. Части опрошенных вернуться к игре помогают замечания или сигналы тренера, напарника.



Рисунок 18 — Ответы респондентов-бадминтонистов на вопрос о способах решения проблемы потери концентрации

Таким образом, решение проблемы потери концентрации внимания действительно актуально и может помочь спортсменам при подготовке.

**4 Выбор решения проблемы**

Когда возникает необходимость в определении, попал игровой снаряд в поле или в аут, люди обычно обращаются к автоматическим системам контроля линий или ALC системам.

Большая часть ALC систем используется или использовалась в теннисе, как в более популярном (и более финансируемом) виде спорта, однако принцип работы для бадминтона не будет отличаться принципиально.

На конференции, созванной ITF в 2003 году, было представлено несколько различных типов ALC систем. Шрихарш Келкар (Shreeharsh Kelkar), автор «Сила зрения: Hawk-eye в теннисе» («The Power of Sight: The Case of Hawk-Eye in Tennis»), разделяет их на три вида (пункты 4.1 и 4.2 ниже)[].

4.1 Лучевые и проводные системы

Лучевые системы, такие, как Циклоп, являются самыми первыми ALC системами. Принцип их работы заключается в том, что электромагнитные лучи перебрасываются с одного конца корта на другой. В Циклопе один луч направляют вдоль линии подачи, еще четыре — в зоне аута. Как только один из лучей был задет, все остальные выключаются. Если мяч вышел за пределы корта, Циклоп издает звуковой сигнал. Расположение лучей Циклопа представлено на рисунке 19.

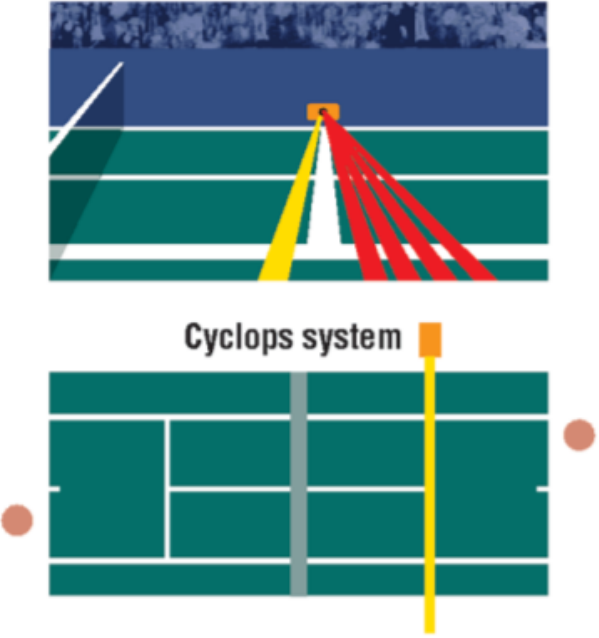


Рисунок 19 — Принцип работы системы Циклоп

У данной системы есть ряд существенных недостатков. Она может использоваться только при подаче, так как не различает, какой предмет возник на пути луча — мяч или что-либо другое. По этой же причине судья вынужден выключать Циклоп после подачи и включать перед началом следующего розыгрыша. Циклоп активно применялся начиная примерно с 1980 года (впервые применен на турнире в Уимблдоне) и окончательно вышел из использования с приходом в теннис Hawk-eye.

Принцип работы проводных систем также полностью отражает их название — такие системы предусматривают наличие сенсоров, расположенных по линиям игрового поля. Иногда модифицируют и сами мячи.

Одной из самых первых таких систем является TEL. Линии поля могут «чувствовать», когда мяч находится близко к ним, а затем алгоритм рассчитывает отпечаток мяча. Этот алгоритм пытается учитывать деформацию снаряда и его скольжение. TEL был испытан в 1992 на US Open. В 9% спорных моментов судья на вышке и система пришли к различным результатам, что является довольно неплохим результатом. Также TEL один год успешно использовался на Кубке Хопмана. При этом присутствовал только судья на вышке, судьи на линии отсутствовали. Однако были выявлены следующие проблемы: на случай технической ошибки все равно требовалось наличие судей на линии, а также было необходимо перекапывать корт для установки.

**4.2 Системы компьютерного зрения**

Эпоха расцвета для ALC систем пришла с появлением систем компьютерного зрения. Предшественником системы Hawk-eye была система AutoRef.

Первым турниром высокого уровня, на котором была применена система Hawk-eye, стал Кубок Хопмана, проведенный в январе 2006 года в Австралии. Игрокам было разрешено оспаривать решения, принятые судьями. В таких случаях также рассматривались результаты, полученные с помощью системы Hawk-eye. Дебютным рейтинговым турниром стал мартовский Miami Masters того же года. Первым же турниром Большого Шлема, на котором использовался Hawk-eye, стал US Open 2006 года. Каждый игрок мог оспорить 2 мяча за сет. К марту 2008 года три руководящих органа в мире тенниса, ITF, ATP и WTA, создали стандарт правил по использованию системы Hawk-eye.

**4.3 Система на материале Velostat**

Несмотря на большую популярность системы Ястребиного глаза, нельзя отрицать, что стоимость ее установки делает её крайне недоступной для тренировочных целей. Лучевые и проводные системы являются более подходящими аналогами, однако у них есть существенные недостатки.

Лучевые системы будут некорректно работать в случае «многоволанок», когда игровые снаряды остаются на площадке. Таким образом, они будут перекрывать друг друга и система не сможет отследить попадание новых воланов.

Проводные системы предполагают внедрение изменений в сами линии корта и игровые снаряды. И если в случае тенниса подобное подошло бы для тренировок, то в случае бадминтона это маловероятно — масса волана не должна превышать 5,5 грамм [Badminton: How], что делает его модификацию затруднительной (для сравнения, вес мяча для большого тенниса должен быть примерно в 10 раз больше).

В данном случае имеет смысл создание системы, которая не предполагает изменение игрового снаряда, но делает возможным добавление каких-либо элементов на корт. К примеру, мата, который способен определять, к каким его точкам приложено давление. Если определить на таком мате линию корта, но при достаточной точности системы не составит труда определить, попал ли игровой снаряд в поле или в аут.

Velostat — токопроводящая пленка, изготовленная из непрозрачного, пропитанного углеродом, полиолефинового материала, обладающего объемной проводимостью [Портал магистров]. На проводящие свойства пленки не оказывает влияние влажность и дата изготовления.

Обычно она используется для обеспечения физической и электростатической защиты, однако, благодаря своему свойству менять сопротивление в зависимости от прикладываемого давления, Velostat приобрел большую популярность в проектах, где требуются датчики, фиксирующие силу нажатия — примером здесь являются устройства для диагностики вестибулярного аппарата, «умные стельки». Способствовала этому и цена материала — ~24.30$ (~1784,47 руб. на 20.09.2021 с ebay.com) за лист 80 см x 80 см. Единственной проблемой для российского рынка может быть то, что данный материал сложно достать в стране — он мало где продается, а потому возникает необходимость заказывать его из-за рубежа, к примеру, из Франции.

Среди действующих тренеров по бадминтону были проведены опросы, поможет ли, на их взгляд, подобный мат решить проблему потери концентрации среди спортсменов. Описание разработки было встречено положительно, большинство выразило готовность помочь в разработке путем использования макетов во время тренировок, предоставлением обратной связи, а также рассмотрением возможности применения готового продукта в индивидуальных программах подготовки спортсменов.

**5 Анализ применимости материала Velostat**

Для исследования применимости необходимо было собрать макетный образец устройства. Он состоит из двух частей — самого мата и микроконтроллера, определяющего его поведение.

Подробное описание разработки макетного образца и описание принципов его работы представлено в части 2.

**5.1 Анализ чувствительности**

В первую очередь необходимо было исследовать, сможет ли материал корректно реагировать на воздействие волана.

Для этого были проведены эксперименты — волан с различной высоты бросался на макет. Считывая значение с аналогового входа, можно было определить, в какой момент происходило соприкосновение игрового снаряда с поверхностью мата.

Преобразование значения аналогового сигнала в напряжение производится по формуле 1:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

где — значение напряжения на аналоговом входе, — значение, полученное с аналогового входа (от 0 до 1023), — напряжение питания, 5 В.

Из экспериментов было выявлено, что материал обладает высокой чувствительностью — вне зависимости от изменений высоты, в момент перед отскоком значение с аналогового входа находилось в районе 600-900. От высоты зависела сила, с которой волан давил на мат в момент после отскока, во время второго приземления, а также само это время отскока.

Для базового случая, когда волан опускается с небольшой высоты, были получены графики зависимости напряжения на аналоговом входе микроконтроллера при давлении падающего волана на Velostat от времени. Совместное представление части из этих графиков можно увидеть на рисунке 20.

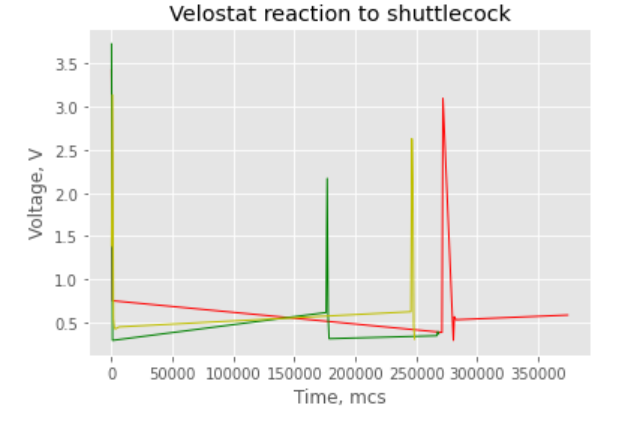


Рисунок 20 — Реакция материала на падение волана

Из рисунка видно, что графики обладают общей моделью поведения. В начальный момент наблюдения с аналогового входа получается максимальное значение — волан с максимальной кинетической энергией соприкасается с поверхностью мата. Затем происходит отскок, во время которого материал возвращается в состояние, близкое к начальному. После этого происходит второе касание, которое оказывает на мат меньшее давление.

Отдельно графики представлены на рисунках 21-22.

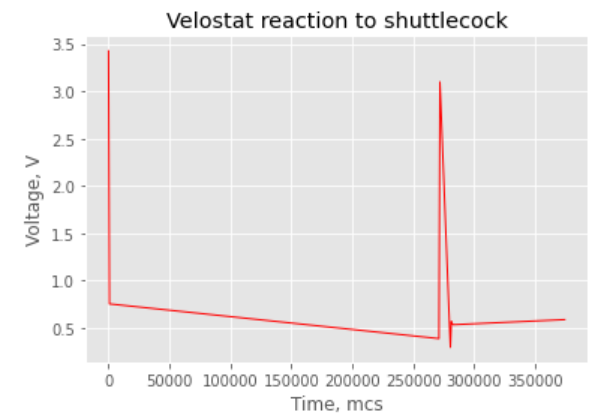


Рисунок 21 — Анализ чувствительности. Эксперимент 1

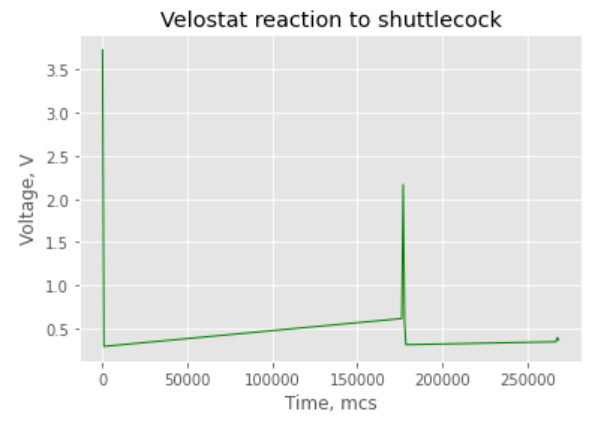


Рисунок 22 — Анализ чувствительности. Эксперимент 2

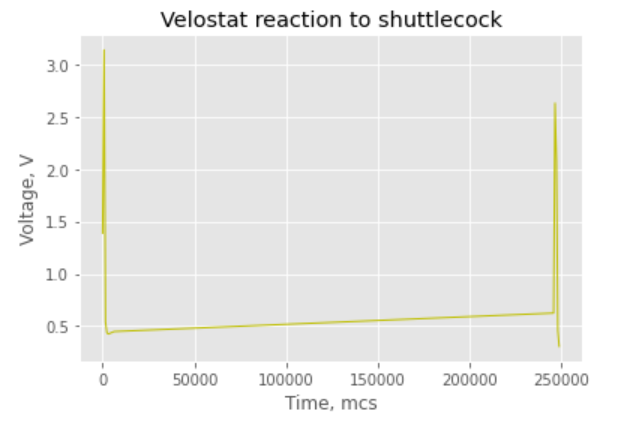


Рисунок 23 — Анализ чувствительности. Эксперимент 3

Как видно из графиков, среднее время возврата материала в исходное состояние не превышало 350 мс.

**5.2 Анализ быстродействия**

Для того, чтобы определить быстродействие схемы, с помощью внутреннего таймера микроконтроллера было посчитано время, затрачиваемое на:

1. Заполнение матрицы значений.
2. Передачу матрицы значений.

Для того, чтобы схема имела максимальное быстродействие, некоторые ее вычисления можно перенести на иное устройство, которое будет использоваться конечным пользователем — на телефон или компьютер. Уже там можно производить анализ полученной матрицы, то есть сравнивать ее элементы с минимальным значением с аналогового вывода в отсутствие давления. Это также будет наиболее оптимальным выходом, поскольку появляется возможность задействовать многопоточность для таких вычислений.

В данном эксперименте передача значений производилась через COM-порт. Было произведено несколько замеров, после чего было высчитано их среднее. Результаты измерений представлены в таблице 1.

Таблица 1 — Временные измерения

|  |  |
| --- | --- |
| Вид измерения | Время, мкс |
| Заполнение матрицы значений | 44216 |
| Передача матрицы значений | 390181 |

Таким образом, один полный проход занимает примерно 434 мс. При этом заполнение матрицы значений занимает меньшую часть этого времени, 44 мс.

Учитывая данные пункта 5.1, можно сделать вывод, что вероятность пропуска волана мала — частота попаданий волана не будет превышать допустимую, чтобы попасть в окно отправки данных на внешнее устройство .

**5.3 Анализ точности**

В основе мата находится сетка 16х16 из меди, которая позволяет считывать значения из 256 точек. Ширина линии меди составляет 40 мм — равна ширине линии корта. Расстояние между линиями составляет 2 мм. Ширина головки игрового снаряда составляет от 25 до 28 мм, ширина участка, которым производится касание с поверхностью, меньше — она составляет от 7 до 10 мм.

В случае полного попадания в зону линии медной ленты можно безошибочно определить результат. Проблему здесь представляют граничные касания — влияет ли близкое нахождение волана к одной линии на результаты считывания напряжения с другой линии?

Для проведения эксперимента был собран дополнительный макет, в котором использовались линии медной ленты меньшего размера, шириной 10 мм. При этом собранный макетный образец безошибочно определял результат при нажатии воланом на разные линии. Так как при обработке результатов приоритет отдавался линиям игрового «поля», а не «аута», с обработкой спорных ситуаций (снаряд оказался между двумя линиями) не возникло проблем.

**Глава II. Программная и аппаратная части системы**

**1 Аппаратная часть**

**1.1 Принцип работы аппаратной части**

В аппаратной части системы контроля попадания волана в заданную зону можно выделить две основные части — устройство, состоящее из микроконтроллера и других микросхем, и датчик давления, созданный на основе материала Velostat и меди.

В данном устройстве в качестве проводящих слоёв была использована медная лента. Сам мат представляет из себя следующее: лист материала Velostat располагается между рядами лент из меди. К ним припаиваются провода. К одному слою поступает питание, провода со второго слоя поступают на аналоговые входы собранной схемы.

Общая схема мата выглядит примерно так, как показано на рисунке 2.1.

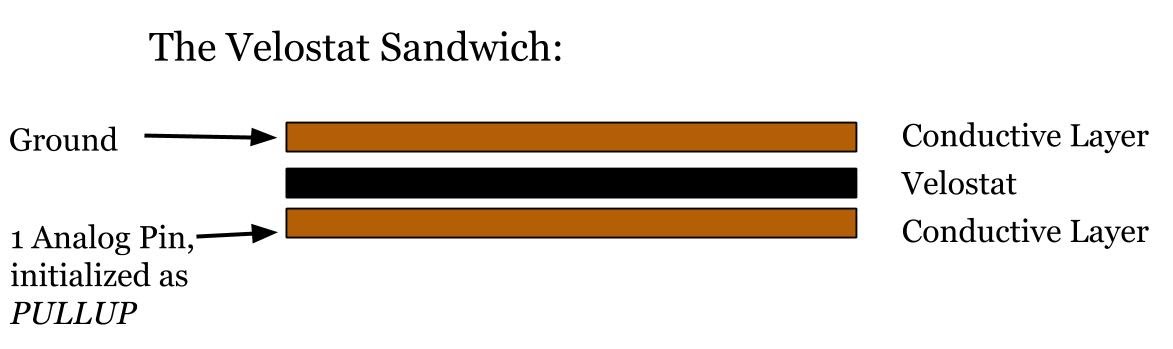
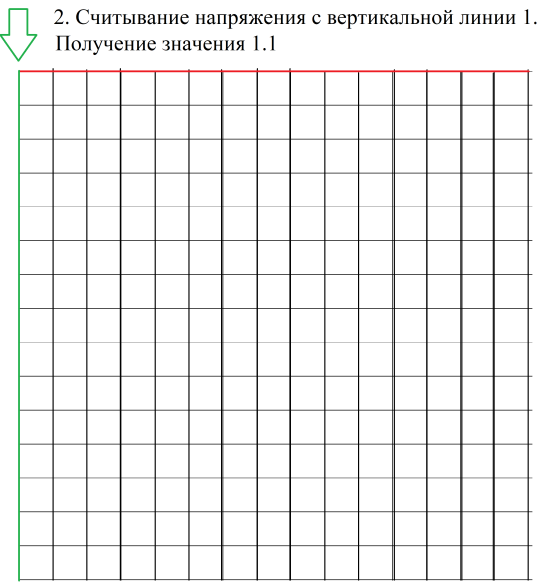
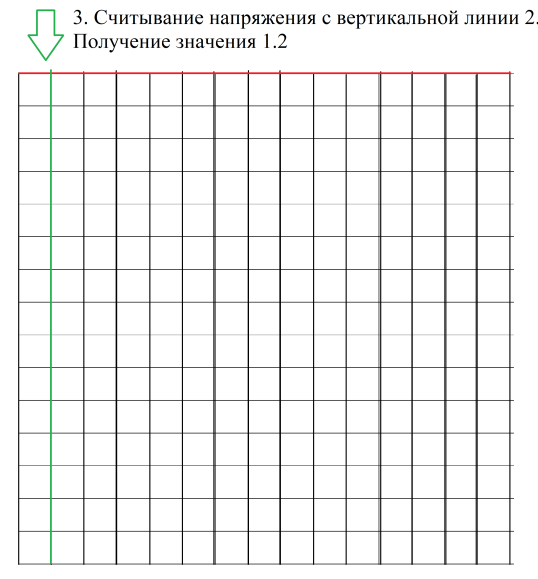
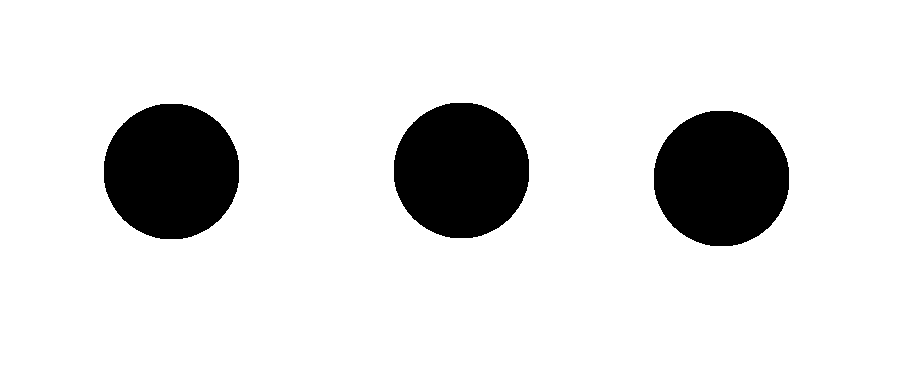


Рисунок 2.1 — Основа мата

Принцип работы заключается в следующем. С платы микроконтроллера с помощью подачи напряжения на цифровые выходы производится выборка текущей горизонтальной линии, которая будет подключена к питанию. Затем производится последовательная выборка 16-ти вертикальных линий. Если к мату приложено давление, в одной из этих точек напряжение будет ненулевым.

Данная выборка производится для всех горизонтальных линий, на которые последовательно подается напряжение. Если изображать схематически, весь процесс выглядит так, как представлено на рисунке 2.2.





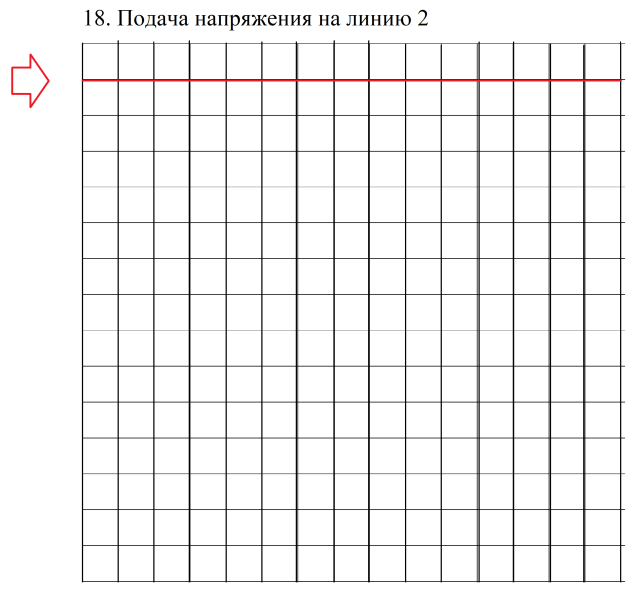
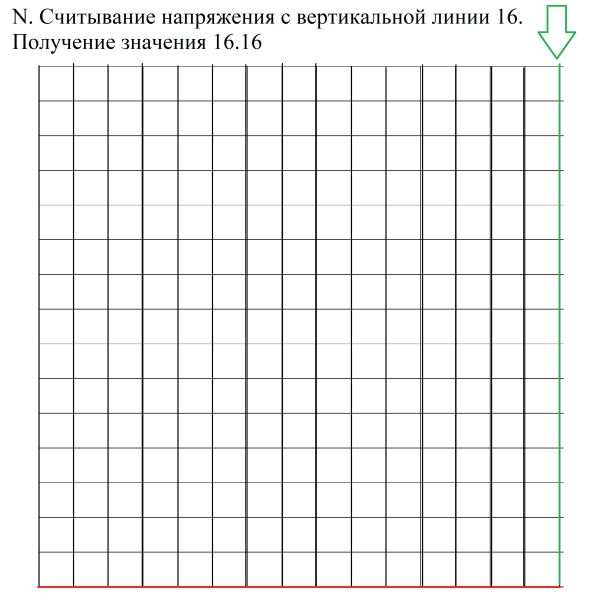
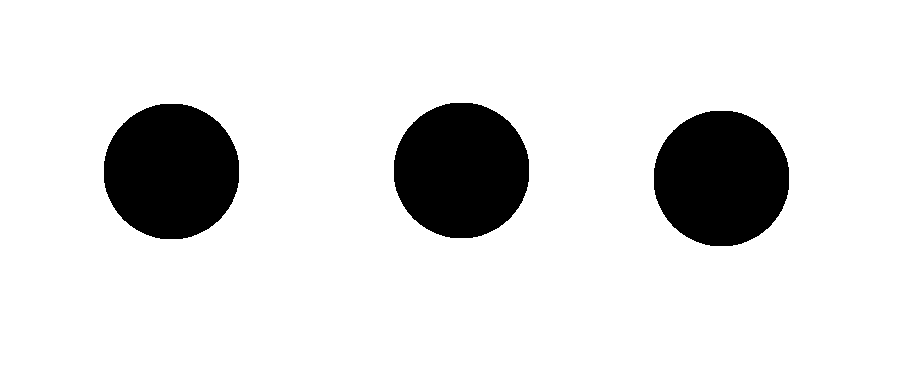


Рисунок 2.2 — Выборка точек

**1.2 Реализация взаимодействия с программной частью**

В настоящее время устройства, использующие технологию Bluetooth для связи с другими устройствами, используются повсеместно. Люди слушают музыку через беспроводные наушники. Можно избавиться от лишних проводов на столе, подключив клавиатуру по Bluetooth к компьютеру. Эта универсальная технология позволила обеспечить надежную беспроводную передачу данных, что и обусловило ее популярность.

Подключение Bluetooth модуля к микропроцессорной системе может обеспечить возможность управления ей с другого устройства. Наличие Bluetooth в любом современном смартфоне определяет актуальность управления подобными системами с помощью мобильного приложения.

Bluetooth — производственная спецификация беспроводных [персональных сетей](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C" \o "Персональная сеть) (Wireless personal area network, WPAN). В настоящее время Bluetooth существует в двух спецификациях — Bluetooth Classic и Bluetooth Low Energy.

Bluetooth Classic также называют Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR). Большей частью используется для беспроводной передачи аудио данных — в беспроводных громкоговорителях, наушниках, встроенных автомобильных развлекательных системах.

Bluetooth Low Energy (BLE) — технология с низким энергопотреблением. Благодаря этой особенности, активно применяется в IoT. Появилась в версии стандарта Bluetooth 4.0.

Эти два типа Bluetooth несовместимы друг с другом. Устройства, которые используют Bluetooth Classic, не могут напрямую связываться с устройствами, использующими BLE, и наоборот. Однако есть устройства, которые поддерживают оба типа соединения. Тогда связь между устройствами может происходить так, как на рисунке 2.3.

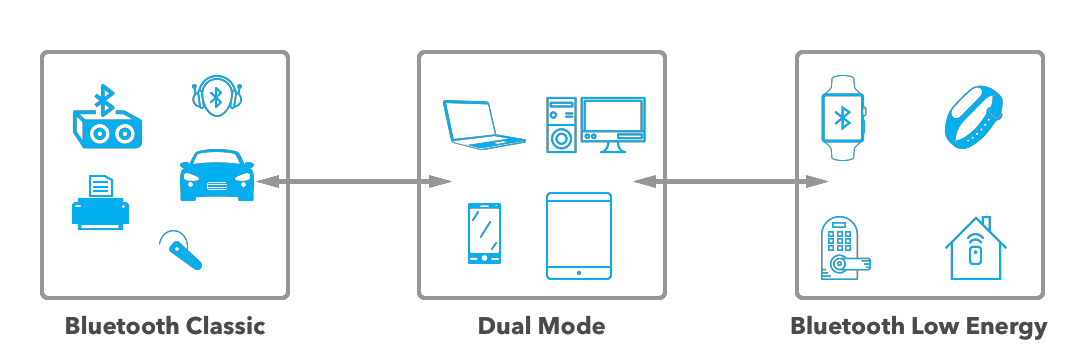


Рисунок 2.3 — Типы Bluetooth-устройств

Bluetooth Classic обычно используется для приложений, в которых требуется передавать большое количество информации. Несмотря на то, что классический Bluetooth не оптимизирован для низкого энергопотребления, он поддерживает большую скорость передачи (максимум 3 МБит/с против 2 МБит/c BLE). Bluetooth Classic поддерживает один тип топологии — Точка-Точка (P2P). Описанные выше характеристики являются преимуществами для данной работы, для использования была выбрана именно эта технология.

**1.3 Разработка электрической функциональной схемы устройства**

Чтобы определять линии, на которые подается питание в определенный момент времени, было принято решение использовать демультиплексор (для подачи питания на выбранную линию) и мультиплексор (для считывания напряжения с выбранной линии). Также использование этих двух устройств позволило использовать более компактные (и более дешевые) модели программируемых плат. Обычно эти модели обладают меньшим количеством цифровых и аналоговых входов и выходов и расширяются при помощи дополнительных модулей, в том числе мультиплексоров и демультиплексоров.

Программирование микроконтроллера производилось через преобразователь USB-UART, макетный образец подключался к компьютеру/ноутбуку. После загрузки программы питание осуществлялось от батарейки.

На основании приведенных выше данных были определены следующие блоки:

1. Источник питания — подача напряжения на всю схему через общую линию питания микроконтроллера микроконтроллера.
2. Микроконтроллер — управляющее устройство.
3. Мультиплексор и демультиплексор — блоки выборки линий мата.
4. Чувствительный к давлению мат — собранный из материала Velostat, линий медной ленты и непроводящего защищающего материала (в самой простой версии используется бумага) датчик.
5. Светодиоды — блок вывода результатов для пользователя, используется для отладки работы микропроцессорной системы.
6. Bluetooth-модуль — блок сообщения с программной частью.

Полученная функциональная схема устройства представлена на рисунке 2.4.

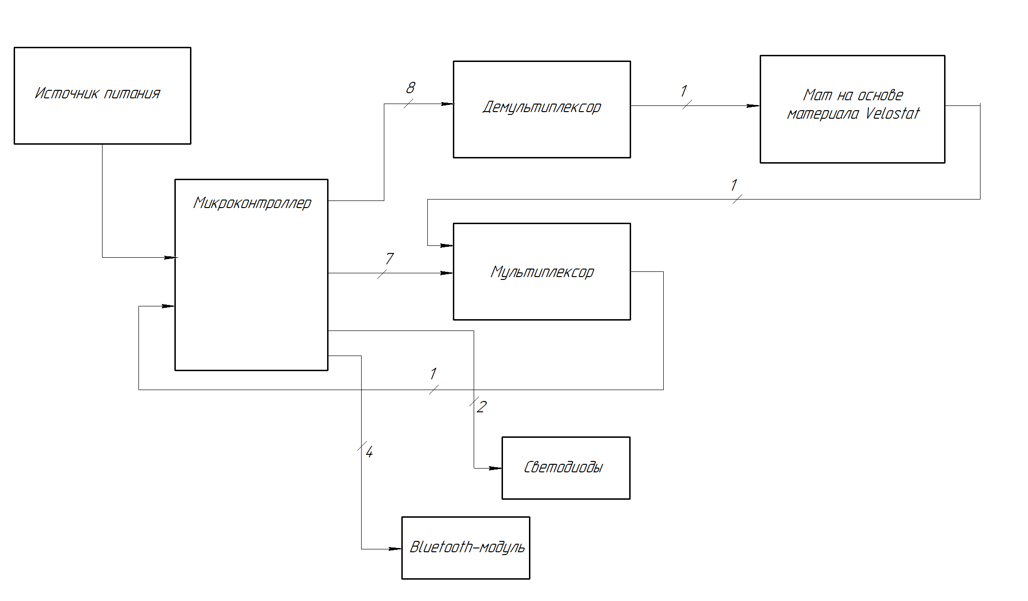


Рисунок 2.4 — Функциональная схема

**1.4 Выбор элементной базы**

Выбирая элементную базу, учли потребляемую мощность микросхем, а также их стоимость.

Для платы с микроконтроллером выбор производился среди линейки Arduino — это одна из наиболее доступных фирм для макетирования систем в домашних условиях.

Линейка Arduino представлена широким спектром моделей, начиная с Arduino Mega 2560, у которой 54 цифровых вывода, заканчивая Arduino Nano, у которой всего 14 цифровых пинов. В классической линейке устройств в основном применяются микроконтроллеры семейства Atmel AVR — ATmega2560, ATmega32U4, ATmega328, ATtiny85, ATmega168.

Для данной микропроцессорной системы было принято решение использовать модель Arduino Pro Mini с микроконтроллером ATmega328. Среди остальной линейки серии эта плата отличается маленьким размером. Также Arduino Pro Mini обладает удовлетворительным соотношением цены к техническим требованиям для разработанного мата.

Данная модель обладает характеристиками, представленными в таблице 2.1.

Таблица 2.1 — Характеристики платформы Arduino Pro Mini

|  |  |
| --- | --- |
| Характеристика | Значение |
| Рабочее напряжение | 5 В |
| Входное напряжение | 5-12 В |
| Цифровые входы/выходы | 14 (6 из которых могут использоваться как выходы ШИМ) |
| Аналоговые входы | 8 |
| Постоянный ток через вход/выход | 40 мА |
| Флеш-память | 32 Кб (2 используются для загрузчика) |
| ОЗУ | 1 Кб |
| EEPROM | 512 байт |
| Тактовая частота | 16 МГц |
| Рабочая температура | -40..85°C |

Структурная схема микроконтроллера ATMega328 представлена на рисунке 2.5.

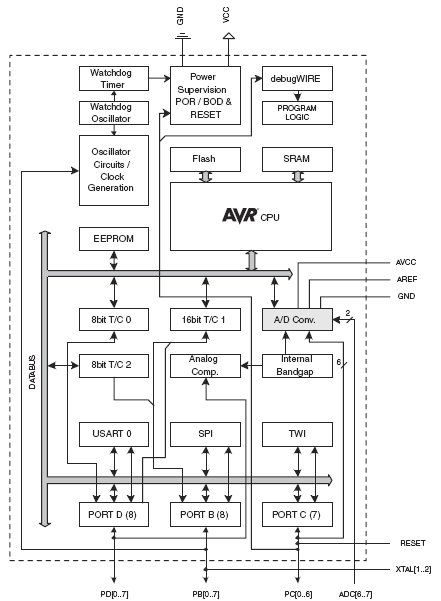


Рисунок 2.5 — Структурная схема микроконтроллера ATMega328

Микроконтроллер обладает следующими периферийными устройствами:

1. Двумя 8-битными таймерами/счетчиками с модулям сравнения и делителями частоты.
2. 16-битным таймером/счетчиком с модулем сравнения и делителем частоты, а также с режимом записи.
3. Счетчиком реального времени с отдельным генератором.
4. Шестью каналами PWM (аналог ЦАП).
5. 6-канальным ЦАП со встроенным датчиком температуры.
6. Программируемым последовательным портом USART.
7. Последовательным интерфейсом SPI.
8. Интерфейсом I2C.
9. Программируемым сторожевым таймером с отдельным внутренним генератором.
10. Внутренней схемой сравнения напряжений.
11. Блоком обработки прерываний и пробуждения при изменении напряжений на выводах микроконтроллера.

Также микроконтроллер обладает специальными функциями:

1. Сбросом при включении питания и программным распознаванием.
2. Внутренним калибруемым генератором тактовых импульсов.
3. Обработкой внутренних и внешних прерываний.
4. 6 режимами сна (пониженным энергопотреблением и снижением шумов для более точного преобразования АЦП).

Плату Arduino Pro Mini можно питать несколькими способам — подавать стабилизированное напряжение 5 В (через USB и преобразователь USB-UART или через вывод VCC) или подавать нестабилизированное напряжение от источника питания — батарейки, зарядного устройства.

Внешний вид платы представлен на рисунке 2.6.

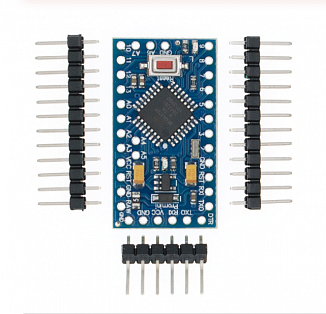


Рисунок 2.6 — Внешний вид паты Arduino Pro Mini

Каждый из 14 цифровых выводов может работать как в режиме входа, так и в режиме выхода. Максимальный ток через один вывод составляет 40 мА. Все выводы сопряжены с внутренними подтягивающими резисторами номиналом 20-50 кОм. Также некоторые выводы Arduino могут выполнять дополнительные функции:

1. Последовательный интерфейс: выводы 0 (RX) и 1 (TX). Используются для получения (RX) и передачи (TX) данных по последовательному интерфейсу. Эти выводы соединены с линиями TX-0 и RX-1 шестиконтактного разъема.
2. Внешние прерывания: выводы 2 и 3. Данные выводы могут служить источниками прерываний, возникающих при различных условиях: при низком уровне сигнала, по фронту, по спаду или при изменении сигнала на этих выводах.
3. ШИМ: выводы 3, 5, 6, 9, 10 и 11.
4. Интерфейс SPI: выводы 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Данные выводы позволяют осуществлять связь по интерфейсу SPI.
5. Светодиод: 13. Встроенный светодиод, подсоединенный к цифровому выводу 13. При отправке значения HIGH светодиод включается, при отправке LOW - выключается.

В Arduino Pro Mini есть 6 аналоговых входов, каждый из которых может представить аналоговое напряжение в виде 10-битного числа (1024 различных значения). Четыре из них выведены на контакты возле края платы; два остальных (входы 4 и 5) выведены на распаечные площадки внутри платы. Измерение напряжения осуществляется относительно диапазона от 0 до VCC. Помимо этого, некоторые из аналоговых входов имеют дополнительные функции:

1. I2C: вывод A4 (SDA) и вывод A5 (SCL). С использованием [библиотеки Wire](https://doc.arduino.ua/ru/prog/Wire) позволяют осуществлять связь по интерфейсу I2С.
2. Reset. Формирование низкого уровня (LOW) на этом выводе приведет к перезагрузке микроконтроллера.

Как было указано выше, Arduino Pro Mini обладает 6 аналоговыми выводами. В случае разработанного макета происходит подача питания и считывание информации с 32 линий (по 16 на слой). Даже у самой большой платы линейки Arduino, Arduino Mega, есть только 16 аналоговых входов. Для увеличения их количества было принято решение использовать мультиплексор и демультиплексор.

Наиболее часто используемой в системах Arduino микросхемой является 16-канальный мультиплексор/демультиплексор CD74HC4067. Он также поставляется на плате, что облегчает его подключение к микроконтроллеру Arduino. Внешний вид CD74HC4067 представлен на рисунке 2.7.

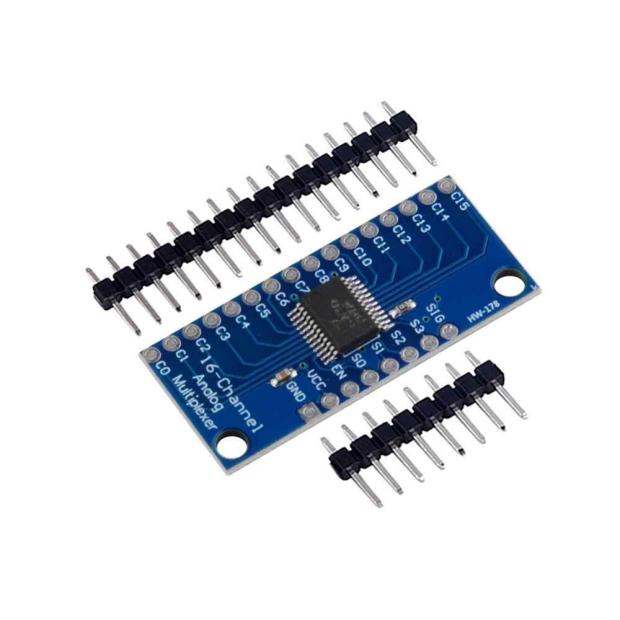


Рисунок 2.7 — 16-канальный мультиплексор/демультиплексор CD74HC4067

Технические параметры устройства представлены в таблице 2.2.

Таблица 2.2 — Характеристики микросхемы CD74HC4067

|  |  |
| --- | --- |
| Характеристика | Значение |
| Напряжение питания | 2-6 В |
| Внутреннее сопротивление ключей | 70 Ом при напряжении 4.5 В |
| Скорость срабатывания | 6 нс при напряжении 4.5 В |
| Диапазон рабочих температур | -55..125 °C |

Технические параметры устройства подходят под условия работы разработанного макетного образца.

В схеме были использованы светодиоды GNL-3012HD с рабочей температурой -40..80 °C с постоянным прямым током 20 мА и постоянным напряжением 2.1 В (сопротивление 105 Ом). Эта часть цепи питается от 5 В. Для понижения силы тока по формуле 2.1 рассчитано сопротивление.

|  |  |  |
| --- | --- | --- |
|  |  | (2.1) |

В данном случае было допустимо использовать резисторы с сопротивлением 150 Ом. Для макета были выбраны резисторы металлопленочные MF-25 (С2-23) с рабочей температурой -55…155 °C и точностью 1%.

В отсутствие приложенного давления с АЦП получаются значения приблизительно в 800 единиц. Тогда можно рассчитать примерное напряжение на входе по формуле 2.2.

|  |  |  |
| --- | --- | --- |
|  |  | (2.2) |

ADC — значение с АЦП, — напряжение на выводе, 5 В. Тогда = 3.85 В.

Помимо того, чтобы просто определять присутствие приложенного давления, полезно было бы уметь отличать силу более сильного нажатия от более слабого. Однако в данном диапазоне (от 800 до 1023) это не представлялось возможным.

Для того, чтобы убрать из цепи часть напряжения, было принято добавить резисторы в 1 кОм на выводах демультиплексора. Были использованы резисторы углеродистые CF-25 (С1-4) с сопротивлением 1 кОм и погрешностью 5%, с рабочей температурой -55…155 °C. Таким образом, в состоянии покоя получились значения, близкие к нулевым — 10-20 единиц, то есть около 68 мВ. При приложенном к Velostat давлении были получены значения, близкие к 800 единицам, то есть около 3.85 В.

Для подключения к программной системе был выбран модуль Bluetooth HC-05. Модуль поддерживает стек  Bluetooth 2.0 + EDR. Подключение модуля осуществляется с помощью разъема XH-2-6P, выводы которого имеют следующее назначение:

STATE — к цифровому выводу контроллера, который предназначен для передачи информации о режимеработы модуля;

RX — к выводу TX последовательного порта контроллера для передачи данных;

TX — к выводу RX последовательного порта контроллера для передачи данных;

VCC — к “+” источника питания контроллера;

GND — «земля» (общий провод).

Модуль представляет из себя печатную плату, на которой смонтированы компоненты, отвечающие за установление связи с контроллером по последовательному интерфейсу UART и организацию передачи данных по Bluetooth каналу при сопряжении с каким либо другим электронным устройством.

Особенностью модуля является возможность его работы в нескольких режимах:

1. Режим поиска устройств и сопряжение с ними, после чего свободно выполняется передача данных между устройствами;

2. Режим передачи данных между модулем и сопряженным устройством. В этом случае возможны два варианта взаимодействия модуля с подключаемыми устройствами:

2.1. Slave — является подчиненным по отношению к другим устройствам. В этом режиме модуль способен только принимать и передавать данные управляющему устройству, с которым он сопряжен. Например, данный режим используется при подключении модуля к смартфону. Режим Slave установлен в настройках модуля по умолчанию.

2.2. Master — модуль является управляющим по отношению к другим устройствам. В этом режиме Bluetooth модуль способен осуществлять поиск Bluetooth-устройств, выполнять сопряжение с ними и, в дальнейшем, управлять их работой, передавая и принимая данные от подчиненного (Slave) устройства.

Данный режим может использоваться для обеспечения связи между двумя контроллерами. В режим Master можно перевести модуль через режим программирования.

1. Режим программирования. Данный режим позволяет изменять всевозможные настройки модуля: запрашивать и менять его имя, адрес, скорость передачи данных; менять режимы со Slave на Master и обратно, запрашивать статусы и классы, выполнять изменение кода доступа к модулю, а также менять другие параметры работы.

В данной работе модуль работает в режиме Slave.

Для определения того, в каком режиме работает модуль, существует система индикации с помощью индикаторного светодиода, находящегося на поверхности платы:

1. Частое мигание светодиода — модуль не подключен к устройствам, поиск устройств для подключения;

2. Светодиод мигает с частотой 1 Гц — модуль в режиме программирования;

3. Светодиод кратковременно мигает один раз в две секунды — модуль сопряжен с другим Bluetooth-устройством.

Модуль обладает техническими характеристиками, представленными в таблице 2.3.

Таблица 2.3 — Характеристики модуля HC-05

|  |  |
| --- | --- |
| Характеристика | Значение |
| Диапазон частот радиосвязи | 2,4–2,48 ГГц |
| Мощность передачи | 0,25–2,5 мВт |
| Чувствительность | –80 dBм |
| Потребляемый ток | 50 мА |
| Радиус действия | до 10 метров |
| Температура хранения | –40…85 °C |
| Рабочий диапазон температур | –25…75 °C |

**1.5 Разработка электрической принципиальной схемы устройства**

Часть принципиальной схемы с микроконтроллером, его обвязкой и блоком светодиодов представлена на рисунке 2.8.

<work in progress>

На данном рисунке можно увидеть элементы, которые принадлежат самой плате Arduino Pro Mini и которые легко можно на ней найти, к примеру, кнопку сброса SW1, светодиод HL1.

На рисунке 2.9 представлена часть схемы, содержащая стабилизатор напряжения.

<work in progress>

Рисунок 2.9 — Стабилизатор напряжения

Разъем XP2 используется для подключения нестабилизированного напряжения 9 В, к примеру, батарейки «Крона». C помощью MIC5205 производится понижение напряжения до 5 В, светодиод HL4 присутствует для индикации наличия питания на плате. Обвязка из конденсаторов позволяет погасить лишние шумы.

Блок выборки с демультиплексором представлен на рисунке 2.10.

<work in progress>

Рисунок 2.10 — CD74HC4067 в режиме демультиплексора

Элемент CD74HC4067 может работать как в режиме демультиплексора, так и в режиме мультиплексора. К демультиплексору с микроконтроллера поступают сигналы D10, D11, D12, D13, с помощью которых производится выборка нужного канала (0...15), а также A0, который передается на выбранный канал. Сигнал D4 запрещает или разрешает работу элемента.

Часть схемы с датчиком давления представлена на рисунке 2.11.

<work in progress>

Рисунок 2.11 — Чувствительный к давлению мат

Было принято решение обозначить материал Velostat как матрицу тензорезисторов, так как его сопротивление меняется в зависимости от приложенного давления.

Так, для того, чтобы получить значение напряжения с резистора R2.19, необходимо подать питание на вывод 1 (сигнал 8) демультипелексора и считать сигнал 7.

Блок выборки с демультиплексором представлен на рисунке 2.12.

<work in progress>

Рисунок 2.12 — CD74HC4067 в режиме мультиплексора

К мультиплексору с микроконтроллера поступают сигналы D6, D7, D8, D9, с помощью которых производится выборка нужного канала (0...15) Сигнал D5 запрещает или разрешает работу элемента. Сигнал с выбранного канала передается на выход A1 и поступает на аналоговый вход микроконтроллера. Резисторы R8…R23 используются в качестве нагрузочных.

Блок с модулем Bluetooth HC-05 представлен на рисунке 2.13.

<work in progress>

Рисунок 2.13 — Модуль HC-05

<описание модуля>

**1.****6 Расчет потребляемой мощности**

Для расчета напряжений резисторов был использован вывод из закона Ома. Была получена формула 2.3, данные по резисторам внесены в таблицу 2.4.

|  |  |  |
| --- | --- | --- |
|  |  | (2.3) |

Таблица 2.4 — Мощности резисторов

|  |  |  |  |
| --- | --- | --- | --- |
| Элемент | Количество | Значение мощности, Вт | Σ, Вт |
| MF-25 | 2 | 0.167 | 0.334 |
| CF-25 | 16 | 0.025 | 0.4 |

Потребляемая мощность схемы рассчитывается по формуле 2.4.

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |

Статическую мощность можно рассчитать по формуле 2.5, динамическую — по формуле 2.6.

|  |  |  |
| --- | --- | --- |
|  |  | (2.5) |
|  |  | (2.6) |

Значения для CD74HC4067 вынесем в таблицу 2.5.

Таблица 2.5 — Подсчет мощности микросхемы CD74HC4067

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| , В | , μА | , мВт | , пФ | , МГц | , мВт | N | , Вт |
| 5 | 160 | 0.8 | 29 | 16 | 11.6 | 2 | 0.025 |

Arduino Pro Mini при питании от 9 В потребляет 0.162 Вт, при питании от 5 В — 0.085 Вт. Среди линейки Arduino данная плата обладает наименьшим энергопотреблением.

Мощности для светодиодов были рассчитаны из вывода формулы 2.3.

Мощность для модуля Bluetooth была рассчитана из формулы 2.5.

Общая сводка мощностей собрана в таблице 2.7.

Таблица 2.7 — Подсчет мощности электрической схемы

|  |  |
| --- | --- |
| Элемент | Мощность, Вт |
| Arduino Pro Mini | 0.162 |
| CD74HC4067 | 0.025 |
| MF-25 | 0.334 |
| CF-25 | 0.4 |
| GNL-3012HD | 0.084 |
| HC-05 | 0.165 |
|  | 1.17 |

**1.7 Расчет надежности**

Надежность — свойство объекта выполнять заданные функции, сохраняя во времени значения установленных эксплуатационных показателей в заданных пределах.  Надежность ОБ, как системы, характеризуется потоком отказов ʌ, численно равное сумме интенсивности отказов отдельных устройств, как представлено в формуле 2.7.

|  |  |  |
| --- | --- | --- |
|  |  | (2.7) |

В таблице 2.8 приведен расчет суммы интенсивности отказов.

Таблица 2.8 — Интенсивность отказов системы

|  |  |  |  |
| --- | --- | --- | --- |
| Элемент | , \*10-5, 1/ч | i |  |
| Резисторы постоянные | 0.03 | 18 |  |
| Транзисторы германиевые | 0.3 |  |  |
| Диод кремниевый | 0.2 |  |  |
| Соединения паяные | 0.004 |  |  |
| Конденсаторы электролитические | 0.035 |  |  |
| Провода соединительные | 0.015 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**2.1.8 Программирование аппаратной части**

Язык программирования Arduino называется Arduino C и представляет собой язык [C++](https://ru.wikipedia.org/wiki/C++" \o "C++) с [фреймворком](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA" \o "Фреймворк) [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)" \o "en:Wiring (development platform)), Он имеет некоторые отличия по части написания кода, который компилируется и собирается с помощью [avr-gcc](https://ru.wikipedia.org/wiki/GNU_Compiler_Collection" \o "GNU Compiler Collection), с особенностями, облегчающими написание работающей программы — имеется набор библиотек, включающий в себя функции и объекты.

Программирование микроконтроллера ATmega328 было произведено с помощью среды разработки Arduino IDE.

Все микроконтроллеры от Arduino имеют специальный загрузчик, созданный на основе Atmel AVR Application Note AN109. Этот загрузчик может работать через интерфейс USB, с его помощью производилось программирование.

Работу программы можно представить в виде конечного автомата из рисунка 2.14.

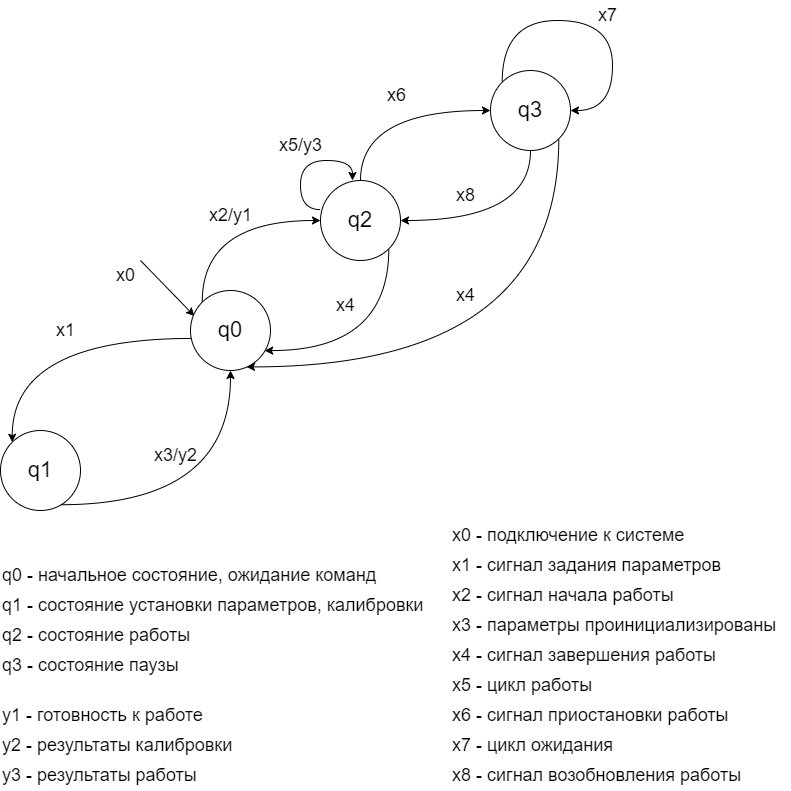


Рисунок 2.14 — Конечный автомат

Процессом «калибровки» назван процесс обхода по матрице мата, при котором считывается максимальное значение снимаемого напряжения в отсутствие приложенного к мату давления. Это значение названо «шумом».

Для общения с программной частью установлен следующий формат получаемых и принимаемых сообщений — «{номер команды/номер состояния} {аргумент 1} {аргумент 2} … {аргумент n}». Таким образом, часть переходных сигналов, инициализируемая программной частью, моет быть определена таким образом:

1. x1 — «{0} {номер строки} {номер столбца}».
2. x2 — «{1}».
3. x4 — «{4}».
4. x6 — «{2}».
5. x8 — «{3}».

Выходные же сигналы определены следующим образом:

1. y1 — «{0}».
2. y2 — «{1} {результаты калибровки}».
3. y3 — «{2} {результат попадания} {номер строки} {номер столбца}».

В алгоритме работы можно выделить несколько блоков.

Первый блок — основной цикл, состояние q0. В данном состоянии происходит ожидание получения сигналов. В первую очередь система должна получить сообщение об установке параметров, x1, и перейти в состояние q2.

Второй блок — состояние, в котором происходит установка параметров и процесс калибровки. Здесь же, в случае успешного выхода из состояния, необходимо установить флаг, который позволит системе, получив из состояния q0 сигнал x2, перейти в состояние x3.

Третий блок — состояние работы. В цикле для каждой линии матрицы мата необходимо было совершить следующие действия:

1. Подать напряжение на линию питания.
2. Считать 16 значений напряжения с 16 точек, заполнить логическую матрицу — матрицу из 256 значений, в которой «1» означает значение выше некоторого минимального значения напряжения, «0» же означает значение ниже этого.

Дополнительно для отладки сохраняется матрица значений напряжения, которые мат считывает с аналогового входа — при подключении через USB-UART и связи через COM-порт эти значения передавались на компьютер.

Даже производится оценка логической матрицы. В состоянии q1 задаются зоны «поля» и «аута»; точки, при нажатии на которые макетный образец должен подавать напряжение на зеленый светодиод, на остальные же — на красный. Если в матрице присутствуют единицы, система формирует сообщение y3 и отправляет его программной системе.

Четвертый блок — состояние ожидания. Выйти из него система может выйти только по сигналам x8 и x4.

Схемы алгоритмов работы аппаратной части представлены на рисунках 2.16-2.22.

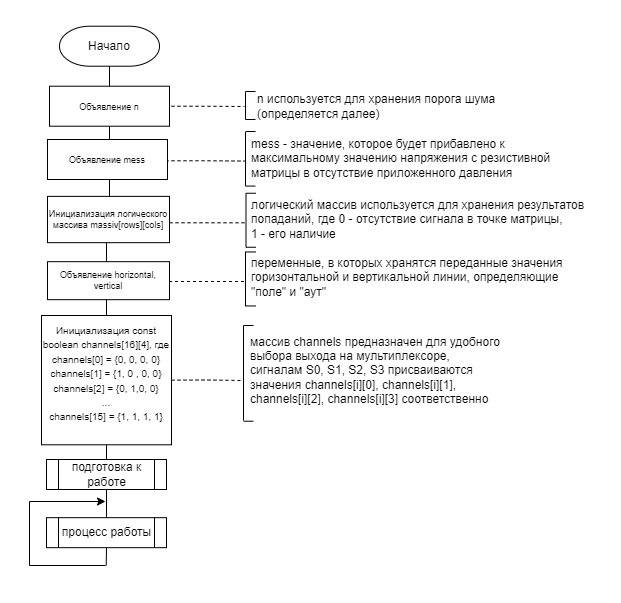


Рисунок 2.16 — Основной цикл работы

Основной цикл работы составляют объявление глобальных переменных, а также процедуры подготовки к работе и сам процесс работы микропроцессорной системы.

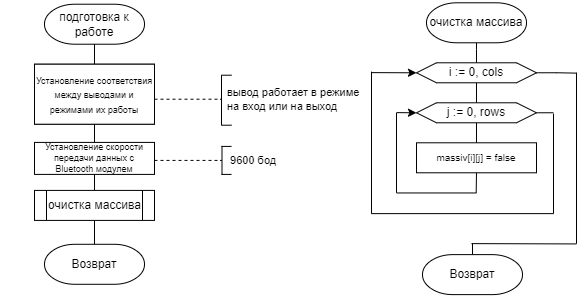


Рисунок 2.17 — Подготовка к работе и очистка массива

Данные по попаданиям в «поле» или «аут» хранятся в логическом массиве. Перед началом работы необходимо обнулить его, задав значения «0», а также инициализировать выводы микроконтроллера на работу в необходимом режиме (вход или выход), установить скорость передачи данных с выбранным Bluetooth модулем (9600 бод для HC-05).

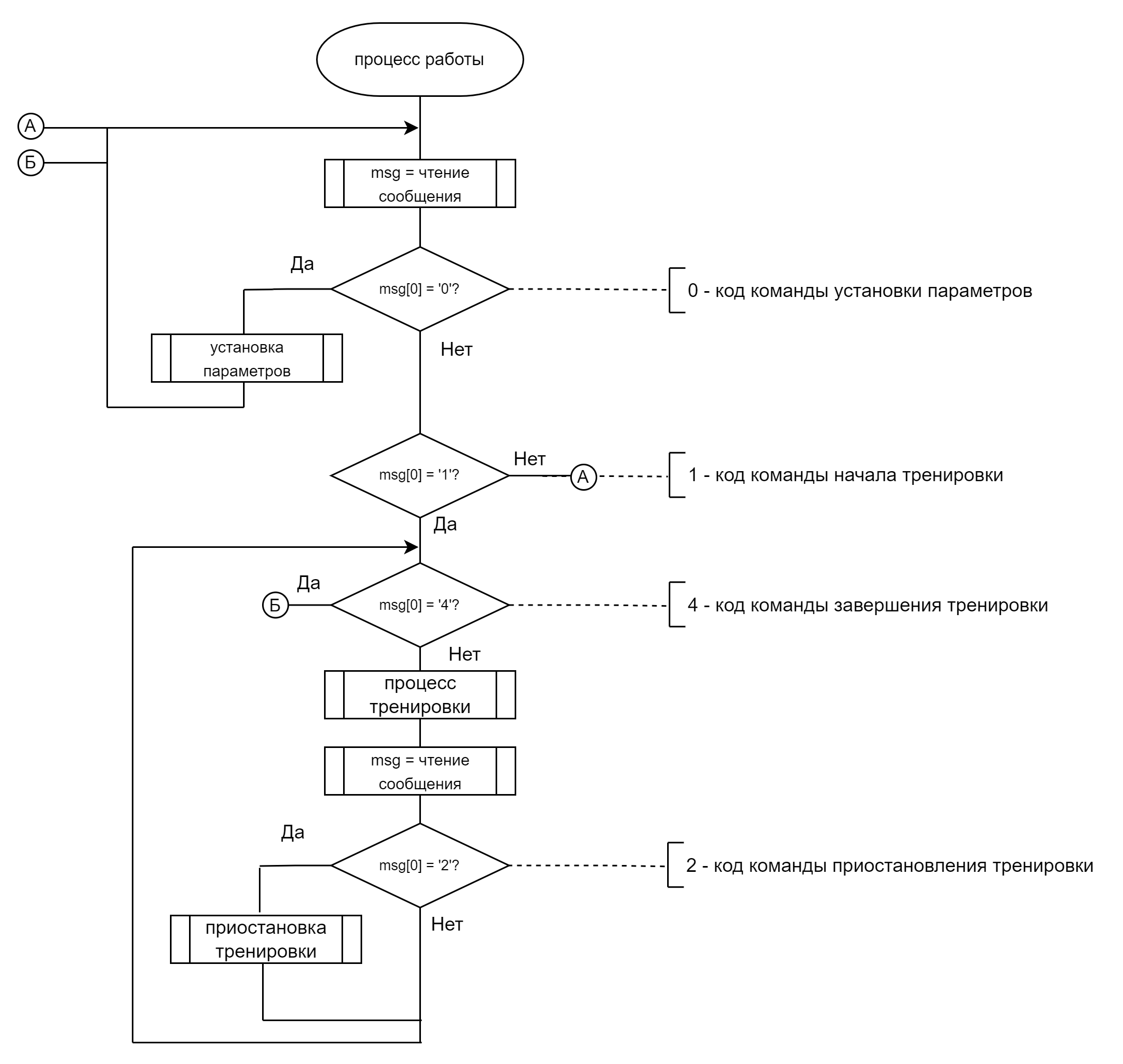


Рисунок 2.18 — Процесс работы

Указанный на рисунке 2.18 процесс повторяется каждый цикл работы микропроцессора. Через получение сообщение реализовано взаимодействие с мобильным приложением. Именно на рисунке 2.18 можно увидеть состоянии показанные на графе конечного автомата на рисунке 2.14.

По сообщению «0» происходит установка параметров и возврат в начало цикла — чтение сообщения.

====схемы алгоритмов и их описания===

**2.1.9 Подключение аппаратной части**

Схема подключения к мату на основе материала Velostat представлена на рисунке 2.23.

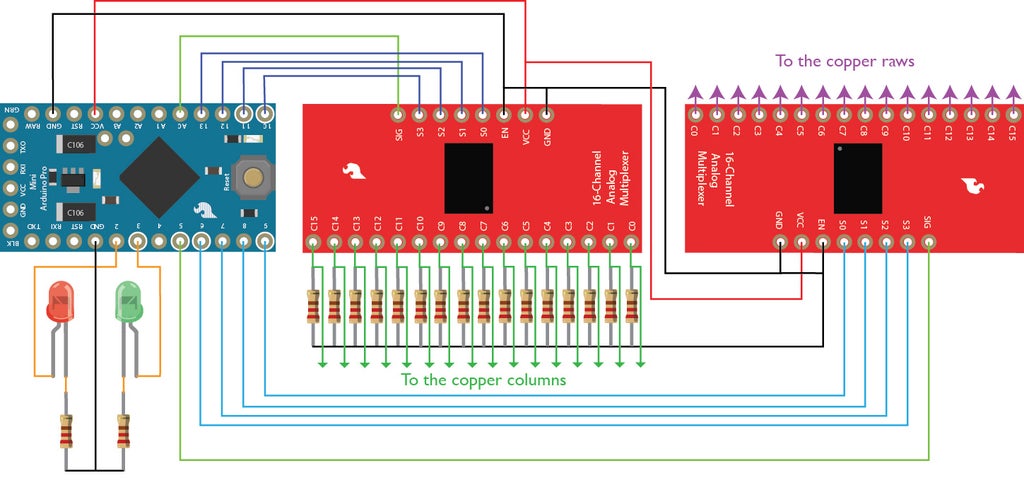


Рисунок 2.23 — Схема подключения

Внешний вид макетного образца представлен на рисунке 2.24.

<work in progress>

Рисунок 2.24 — Макетный образец

На рисунке 2.25 показано строение самого мата.



Рисунок 2.25 — Внутреннее содержимое мата — линии медной ленты и Velostat

**2 Программная часть**

**2.1 Анализ задания и выбор технологии, языка и среды разработки**

Программная система была разработана с использованием объектно-ориентированного подхода — возможность четко выразить классы предметной области и особенности разработки под мобильное приложение доказали его оптимальность.

Приложение было создано под ОС Android. Исходя из данных рисунка 2.26, это самая популярная из используемых мобильных операционных систем[7].

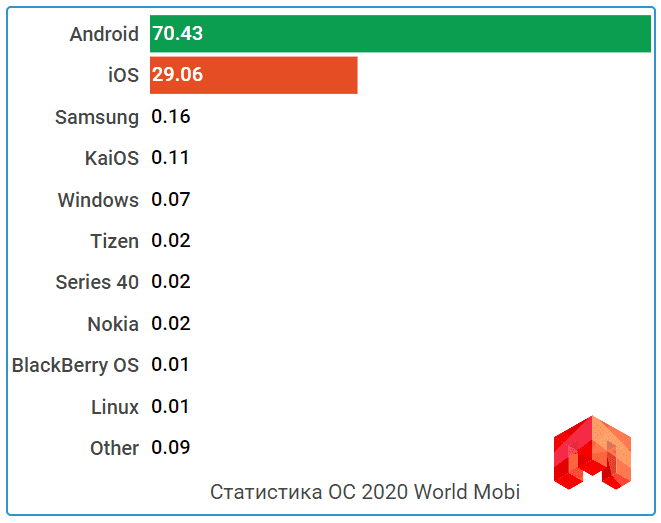


Рисунок 2.26 — Процентное соотношение используемых мобильных операционных систем

Особенностью ОС Android является то, что в один момент времени существует множество устройств, на которых установлены разные версии операционной системы. Так, еще до выхода Android 11 предыдущая версия Android 10 использовалась примерно 8,2% устройств[9]. Исходя из данных, представленных на рисунке 2.27, а также используемых элементов пользовательского интерфейса, было принято решение провести разработку под Android версии 5.0.

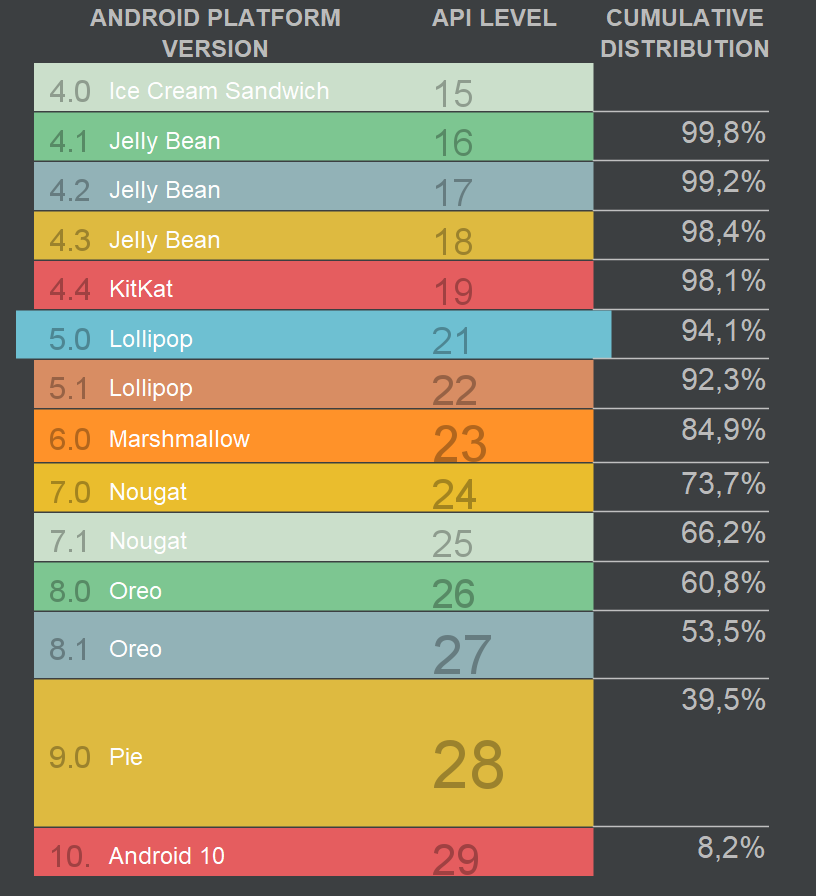


Рисунок 2.27 — Распределение версий ОС по охвату устройств

Одной из лучших IDE для работы с платформой Android является Android Studio[как выбрать язык]. Она предлагается в виде пакета вместе с Android SDK. Разработчик получает все необходимое для того, чтобы создать приложение под данную операционную систему — от визуального конструктора для верстки страниц до эмуляторов различных устройств с ОС Android.

Android Studio поддерживает два официальных языка разработки — Java и Kotlin[android developers]. Также он способен поддерживать и другие языки программирования (к примеру, C/C++, C#).

Для данного проекта было принято решение использовать язык программирования Kotlin. Он является более современным, а также обладает следующими ключевыми характеристиками[k vs jv]:

1. В Kotlin невозможно присвоить null объекту или переменной по умолчанию. Любые такие присваивания определяются на этапе компиляции. Это позволяет сделать код более безопасным.
2. Kotlin является более компактным языком, нежели Java, что значительно ускоряет разработку.
3. Kotlin поддерживает корутины — легковесные потоки, облегчающие работу с многопоточностью, что очень важно при разработке мобильного приложения.

**2.2 Разработка диаграммы вариантов использования**

Разработка программной части была начата с построения диаграммы вариантов использования. Она позволила наглядно продемонстрировать, какие функции должно выполнять приложение.

В диаграмме представлено одно действующее лицо — пользователь. Получившаяся в ходе анализа предметной области диаграмма представлена на рисунке 2.28.

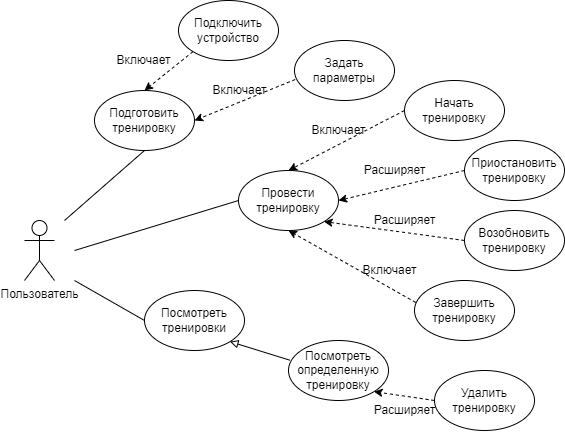


Рисунок 2.28 — Диаграмма вариантов использования

Диаграмма вариантов использования отвечает требованиям, представленным в техническом задании.

Пользователь может начать новую тренировку. Для этого ему необходимо сначала подключить устройство по Bluetooth, затем — задать параметры тренировки и после этого уже приступить к её началу. Пользователь должен обладать способностью стартовать, приостановить, возобновить и завершить тренировку.

Также пользователь должен иметь возможность просмотреть завершенные тренировки, а также, при желании, удалить их.

2.3 Хранение данных

Завершенные тренировки хранятся на локальном устройстве в базе данных SQLite в одной таблице — Trainings. Ее структура представлена на рисунке 2.29.

<work in progress>

Рисунок 2.29 — Структура таблицы Trainings

Для взаимодействия с базой данных была выбрана библиотека Room[android developers]. Room имеет три основных компонента: Entity, Dao и Database. Аннотация [Entity](https://developer.android.com/reference/android/arch/persistence/room/Entity.html" \t "https://startandroid.ru/ru/courses/architecture-components/27-course/architecture-components/_blank) используется для того, чтобы пометить объект, который мы будет хранится в базе данных. Для объекта Training это будет выглядеть так, как представлено в листинге 2.1.

Листинг 2.1 — Entity Training

|  |
| --- |
| @Entity class TrainingEntity(  var date: Long,  var duration: Long,  var hitsIn: Int,  var hitsOut: Int,  var concentrationFaults: Int,  var x: Int,  var y: Int,  var maxErrors: Int,  var hits: String ) {  @PrimaryKey(autoGenerate = true)  var id: Long = 0 } |

В объекте Dao описываются методы для работы с базой данных, они представлены в листинге 2.2.

Листинг 2.2 — Dao TrainingDao

|  |
| --- |
| @Dao interface TrainingDao {  @Query("SELECT \* FROM TrainingEntity")  fun getAll(): List<TrainingEntity>?   @Query("SELECT \* FROM TrainingEntity WHERE id = :id")  fun getById(id: Long): TrainingEntity?   @Insert  fun insert(training: TrainingEntity?)   @Query("DELETE FROM TrainingEntity WHERE id = :id")  fun delete(id: Long) } |

В данном случае необходимо иметь методы создания, удаления, а также получения всего списка тренировок и получения определенной тренировки по первичному ключу — id.

Аннотацией [Database](https://developer.android.com/reference/android/arch/persistence/room/Database.html" \t "https://startandroid.ru/ru/courses/architecture-components/27-course/architecture-components/_blank) помечаtncz основной класс по работе с базой данных. Этот класс должен быть абстрактным и наследовать RoomDatabase, что представлено в листинге 2.3

Листинг 2.3 — AppDatabase

|  |
| --- |
| @Database(entities = [TrainingEntity::class], version = 1) abstract class AppDatabase : RoomDatabase() {  abstract val trainDao: TrainingDao?   companion object {  private var instance: AppDatabase? = null  @Synchronized  fun getInstance(context: Context?): AppDatabase? {  if (instance == null) {  instance = create(context)  }  return instance  }   private fun create(context: Context?): AppDatabase {  return Room.databaseBuilder(  context!!,  AppDatabase::class.java,  "training"  ).allowMainThreadQueries()  .build()  }  } } |

С помощью этих сущностей представляется возможным работа с небольшими объемами данных, которая необходима в данной программной системе.

**2.3 Разработка интерфейсов пользователя**

В разработанном приложении применен объектно-ориентированный подход к разработке интерфейсов — пользователь взаимодействует с объектами напрямую, инициируя выполнение операций.

**2.3.1 Построение диаграммы состояний интерфейса**

Диаграмма состояний интерфейса представлена на рисунке 2.30.

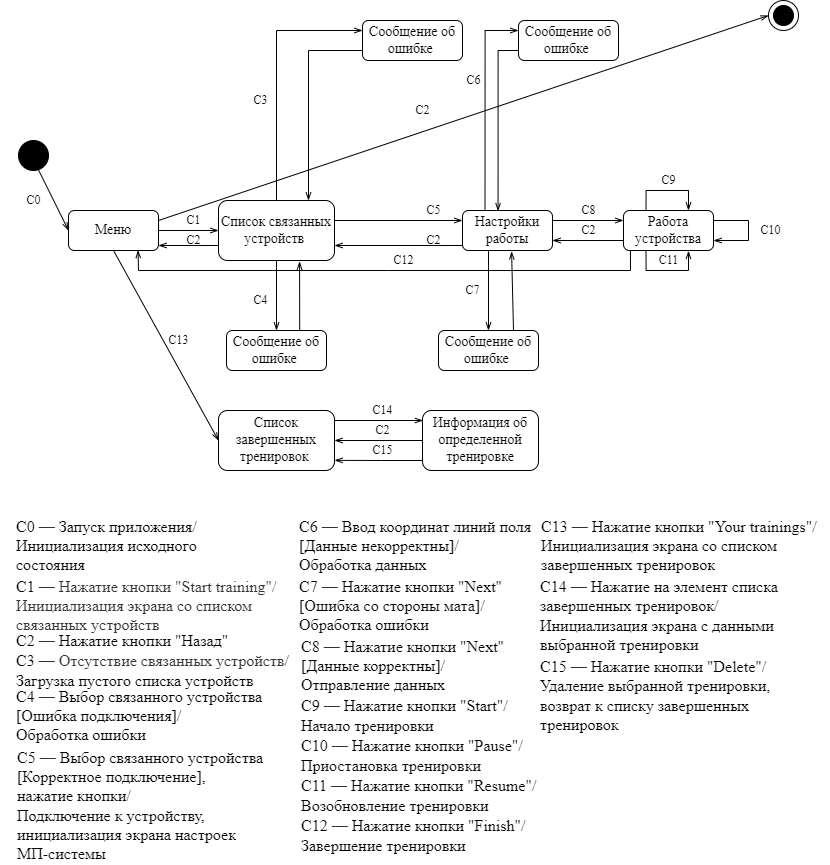


Рисунок 2.30 — Диаграмма состояний интерфейса

Под состоянием тренировки в событиях С9-С12 понимается такое состояние программно-аппаратной системы, когда:

1. Аппаратная часть системы реагирует на падения игрового снаряда, то есть на его попадания в зону, обозначенную как «поле», включает зеленый светодиод, а в зону, обозначенную как «аут» — красный светодиод. Также аппаратная часть передает данные о попаданиях программной части.
2. Полученные от аппаратной части данные программная часть обрабатывает следующим образом:

* Отрисовывает падения на графике, где по оси абсцисс указывается время, пройденное с начала тренировки, а по оси ординат — «1» при попадании снаряд в поле и «-1» при попадании снаряда в аут.
* Подсчитывает сумму попаданий в поле и в аут, общее количество попаданий.
* Подсчитывает последовательное количество попаданий в аут. При превышении заданного количества система издает звуковой сигнал.

**2.3.2 Разработка форм интерфейса**

На основе диаграммы состояний интерфейса было разработано 6 форм для взаимодействия с пользователем. Данные формы представлены на рисунках 2.31-2.36.

<work in progress>

Рисунок 2.31 — Форма стартового экрана

<описания work in progress>

<work in progress>

Рисунок 2.32 — Форма списка связанных устройств

<описания work in progress>

<work in progress>

Рисунок 2.33 — Форма задания настроек

<описания work in progress>

<work in progress>

Рисунок 2.34 — Форма процесса тренировки

<описания work in progress>

<work in progress>

Рисунок 2.35 — Форма списка завершенных тренировок

<описания work in progress>

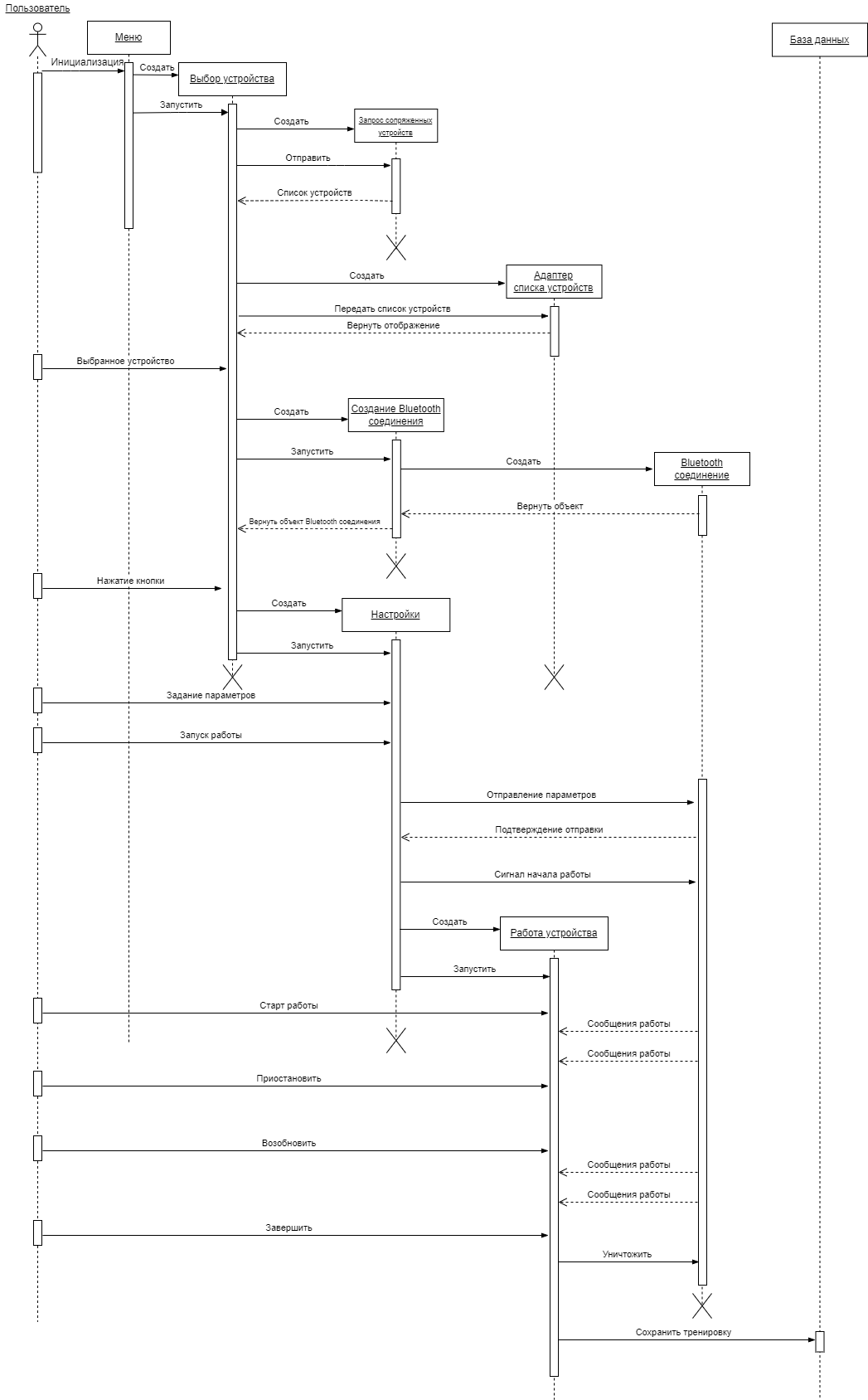
<work in progress>

Рисунок 2.36 — Форма выбранной тренировки

<описания work in progress>

**2.4 Разработка диаграммы последовательности действий**

Взаимодействие компонентов системы рассмотрим на примере диаграммы последовательности действий при стандартном пользовательском сценарии проведения тренировки на рисунке 2.37.

Рисунок 2.37 — Диаграмма последовательности действий

**2.5 Выбор архитектуры**

Одними из основных компонентов приложения в Android являются Activity и Fragment.

Activity отвечает за визуальную часть приложения, за взаимодействие с пользователем. Обычно один Activity отвечает за один экран.

Fragment — модульные, переиспользуемые части пользовательского интерфейса. Они не самостоятельны, а зависят от Activity. Жизненные циклы Activity и Fragment представлены на рисунке 2.38.

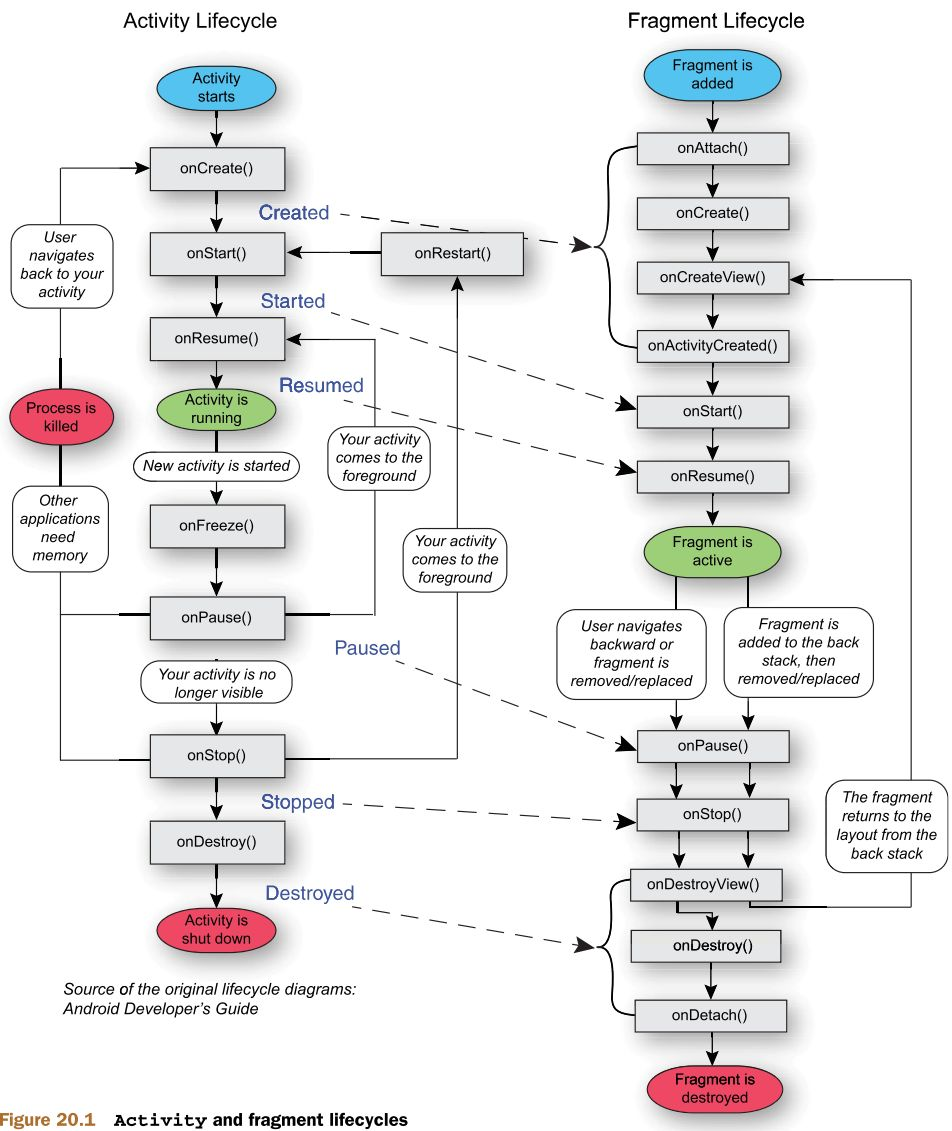


Рисунок 2.38 — Жизненные циклы Activity и Fragment

У фрагмента жизненный цикл немного длиннее, но его методы все равно соотносятся с методами Activity.

В данном приложении было принято решение использовать архитектуру Single Activity. В ней присутствует только одна Activity и несколько Fragment. Основным преимуществом в данной архитектуре является то, что Fragment легче, чем Activity — присутствует выигрыш в производительности. Также в данном приложении необходимо реализовать объект сокета, соединяющего смартфон и плату Arduino по Bluetooth. Для того, чтобы иметь возможность работать с установленным соединением из любой формы и каждый раз не пересоздавать его, сокет можно инициализировать в MainActivity. Тогда соединение будет существовать во всех экранах вплоть до выхода из приложения.

**Глава III. Разработка технологии тестирования**

**3.1 Тестирование методом «структурный контроль»**

Программа анализируется по списку вопросов для выявления исторически сложившихся общих ошибок программирования[тестирование по иванова]. Результаты проверки представлены в таблице 1.

Таблица 1 — Результаты структурного контроля

|  |  |
| --- | --- |
| Вопрос | Результат |
| Все ли переменные инициализированы? | Все переменные инициализированы. |
| Не превышены ли максимальные (или реальные) размеры массивов и строк? | Размеры массивов и строк не превышены. Там, где при вводе пользователь может превысить допустимые значения, присутствуют проверки. |
| Присутствуют ли переменные со сходными именами? | Переменные со сходными именами присутствуют, однако они расположены в разных частях программы и несут схожий смысл, что помогает ориентироваться в коде программы. |
| Правильно ли записаны выражения (порядок следования операторов)? | Выражения записаны корректно |
| Возможно ли переполнение разрядной сетки или ситуация машинного нуля? | Переполнение разрядной сетки и ситуация машинного нуля невозможны |
| Будут ли корректно завершены циклы? | Циклы будут завершены корректно — условия выходов из циклов достигаются всегда. |
| Будет ли завершена программа? | Программа будет завершена при получении сигнала от пользователя — нажатию кнопки «назад» на стартовом экране. |
| Существуют ли циклы, которые не будут выполняться из-за нарушения условия входа? Корректно ли продолжатся вычисления? | Подобных циклов не присутствует |
| Не изменяет ли подпрограмма аргументов, которые не должны изменяться? | Не изменяет |
| Не происходит ли нарушения области действия глобальных и локальных переменных с одинаковыми именами? | Не происходит |

**3.2 Тестирование по принципу «черный ящик»**

**3.3 Smoke-тестирование**

**3.4 Оценочное тестирование**