

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра физики открытых систем

**Разработка системы для считывания движений нижних конечностей
человека с использованием акселерометров**

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

Студента 2 курса 2241 группы
направления 09.04.02 Информационные системы и технологии

Института физики

Тюлюкина Валерия Андреевича

Научный руководитель

Доцент кафедры ТФиСА,
к.ф.-м.н.

Р.Н. Фадеев

Заведующий кафедрой

физики открытых систем,

д.ф.-м.н., профессор

А.А. Короновский

Саратов 2025 год

Введение

Технологии захвата движений (англ. “motion-capture”), представляют собой один из самых значительных прорывов в области анимации, киноиндустрии, видеоигр и даже виртуальной реальности. С момента своего появления эти технологии претерпели множество изменений, адаптируясь к новым требованиям и стандартам, и сегодня они занимают важнейшую нишу в создании контента, который требует высокой степени реализма и эмоциональной выразительности. Использование захвата движений основано на возможности записывать и анализировать движения человека с помощью различных сенсоров, камер и программного обеспечения. Этот процесс заключается в том, что специальные маркеры или датчики размещаются на теле актёра, что позволяет точно отслеживать их движения в пространстве. Записанные данные затем переводятся в цифровой формат, который может быть использован для анимации трёхмерных моделей. Это позволяет создавать персонажей, которые двигаются и ведут себя так же, как и реальные люди, что особенно важно для создания выразительных и реалистичных сцен в кино и видеоиграх. Помимо киноиндустрии, захват движений находит применение в видеоиграх, где интерактивность и пользовательский опыт играют ключевую роль. Игры становятся всё более сложными и многослойными, и использование технологий захвата движений позволяет создавать персонажей с богатой анимацией, что делает игровой процесс более захватывающим и живым. Игроки могут видеть, как персонажи реагируют на их действия в реальном времени, что создаёт более глубокую связь между игроком и игрой, позволяя окунуться в мир, созданный разработчиками. Технологии захвата движений также начали использоваться в области спорта и медиа, где они позволяют анализировать движения спортсменов для улучшения их техники. Это может включать в себя как захват движений для тренировки, так и создание виртуальных моделей спортсменов для анализа их способности к движению.

Одной из ключевых задач является разработка математических моделей, имитирующих поведение сложных систем, в том числе человеческого организма. В связи с этим, разработка математической модели ноги человека на основе экспериментальных данных и создание виртуальной модели представляют собой актуальную и перспективную задачу в области биомеханики, медицины и спорта.

Целью данного исследования является разработка системы, способной считывать и интерпретировать данные о движениях нижних конечностей с использованием датчиков ускорения.

Для достижения этой цели необходимо выполнить следующие задачи:

1. Теоретическое исследование, включающее в себя сбор информации о существующих методах получения данных о человеческих движениях
2. Разработка измерительного оборудования и программного обеспечения для него, необходимого для получения данных о реальном движении человека
3. Проведение экспериментов, анализ полученных в результате данных
4. Разработка программного кода для визуализации движений ноги человека;

В последние годы активно развивается использование виртуальных моделей для имитации движения человеческой ноги в различных средах и условиях. Создание виртуальных моделей позволяет проводить компьютерное моделирование биомеханических процессов, оптимизировать проектирование протезов и реабилитационных устройств, а также улучшать тренировочные программы и методики реабилитации. В данном исследовании описывается процесс разработки системы, данные от которой позволят построить виртуальную модель ноги человека для дальнейших экспериментов и исследований.

Данная работа имеет четыре основных раздела:

1. Описание методов получения и первичной обработки данных;
2. Структура разрабатываемой системы;
3. Описание используемого в системе программного кода
4. Описание результатов работы системы

Основная часть

В данной работе основным методом захвата движений выбран инерциальный ввиду простоты его технической реализации, максимальной автономности используемого оборудования, а методом описания является численное моделирование, в частности - метод опорных точек . Опорные точки — это точки, которые играют ключевую роль при движении конечности и могут быть использованы для анализа движения. Данный метод использует следующие точки:

1. Первый и пятый пальцы стопы: Эти точки также имеют определенное значение при распределении нагрузки во время ходьбы и при выполнении различных движений ног. Данные, полученные с этих точек, позволят описать наклон стопы вокруг продольной оси.

2. Пятка (Calcaneus): точка, расположенная на задней части стопы, которая является основной точкой опоры при стоянии и ходьбе. Пятка играет важную роль при передвижении тела вперед. Данные, полученные с данной точки, вместе с данными о положении первого и пятого пальца позволят определить наклон стопы вдоль поперечной оси.

3. Колено (Femur и Tibia): Колено является соединительным звеном между бедром (Femur) и голенью (Tibia) и играет важную роль при сгибании и разгибании ноги. По одной точке размещено выше и ниже каждого колена, что позволит определить и сгиб колена, и вращательное движение

4. Таз (Pelvis): Таз - это опорный элемент для бедра и также влияет на движения ноги при ходьбе и других активностях. Эти точки (по одной на

каждый тазобедренный сустав) могут служить осью отсчета для обработки данных с остальных опорных точек.

В каждой опорной точке располагается устройство, осуществляющее измерение ускорения и угловой скорости по каждой из трех осей. Пример расположения датчиков на опорных точках приведен на рисунке 1.

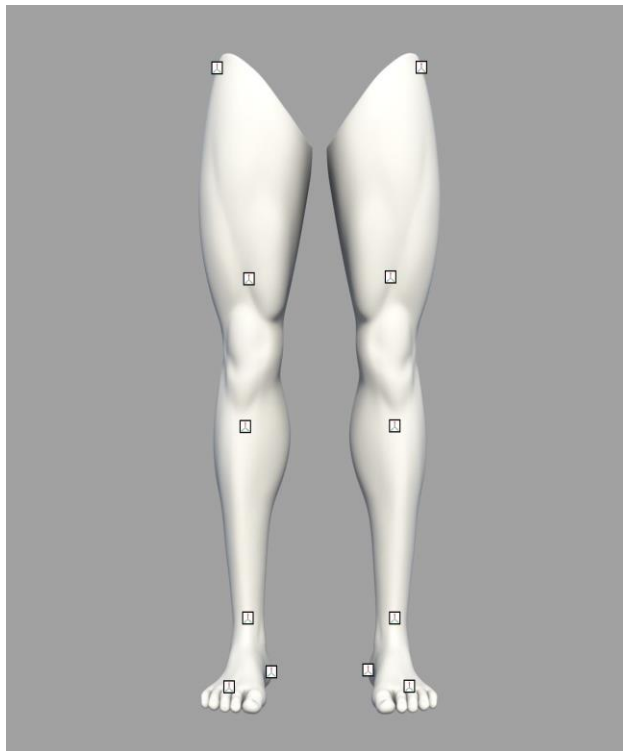


Рисунок 1. Пример расположения опорных точек.

Человеческая нога, очевидно, движется в трёхмерном пространстве. Следовательно, для описания движения ноги необходимо для каждой опорной точки получить три временных ряда, содержащих данные о координате по соответствующей оси в момент времени t , каждый из которых должен иметь начальное значение - показания измерительных приборов в начальный момент времени $t(0)$.

Простейшим методом получения таких данных является использование данных от акселерометра - временных рядов, описывающих ускорение опорной точки в момент времени i . В таком случае, с учётом начальных условий вычисление координаты точки по оси x в момент i -ого измерения выглядит следующим образом:

$$v_{i+1} = v_i + a_i \Delta t, \quad (1)$$

где $v(i)$ - скорость опорной точки в момент времени i , $a(i)$ - ускорение опорной точки в момент времени i , Δt - промежуток времени между измерениями.

Используя значение скорости, полученное на предыдущем шаге, можно вычислить координату $x(i + 1)$

$$x_{i+1} = x_i + v_i * \Delta t + 1/2 * a_i * (\Delta t)^2 \quad (2)$$

Приведённые выше вычисления нужно повторить для каждой координаты.

Дополнительно стоит отметить, что в разрабатываемой системе есть три фактора, вносящих погрешность в получаемый результат. Во-первых, движения испытуемого происходят не в идеальных условиях, но под действием внешних сил, таких как сопротивление воздуха, частично сковывающая движения система измерительного оборудования, малые колебания опоры под испытуемым и отклонения в положении датчиков на теле. Для борьбы с погрешностями в работе используется фильтр Калмана.

Фильтр Калмана, разработанный Рудольфом Калманом в 1960 году, представляет собой алгоритм, используемый для оценки состояния динамических систем с шумными и неполными данными. Он широко применяется в таких областях, как навигация, автоматическое управление, компьютерное зрение и анализ временных рядов. Фильтр Калмана является оптимальным в смысле минимизации среднеквадратичной ошибки при условии, что шумы в системе имеют нормальное распределение. Фильтр Калмана основан на одном из способов отображения динамических систем через линейные дискретные уравнения. Данный фильтр работает в два этапа. На первом этапе, этапе прогнозирования, фильтр Калмана экстраполирует значения переменных состояния, а втором этапе по данным измерения (полученного с некоторой погрешностью) результат прогнозирования уточняется. Так как данный алгоритм является пошаговым, с его помощью возможно отслеживание состояния системы в реальном времени на основе

имеющихся данных о системе и предыдущем состоянии. Данный фильтр можно описать следующей формулой:

$$E(e_{(k+1)}^2) = \frac{\sigma_n^2(Ee_k^2 + \sigma_\varepsilon^2)}{Ee_k^2 + \sigma_\varepsilon^2 + \sigma_n^2}$$

Коэффициент Калмана K с нарастанием номера отсчёта k стремится к стабилизации около значения $K_{(stab)}$, зависящему от отношения среднеквадратичных ошибок (рисунок 2). Это свойство коэффициента Калмана позволяет применять его в большем количестве случаев, чем предполагается теоретически, итеративно подбирая коэффициент K до тех пор, пока он не станет достаточно близок к $K_{(stab)}$.

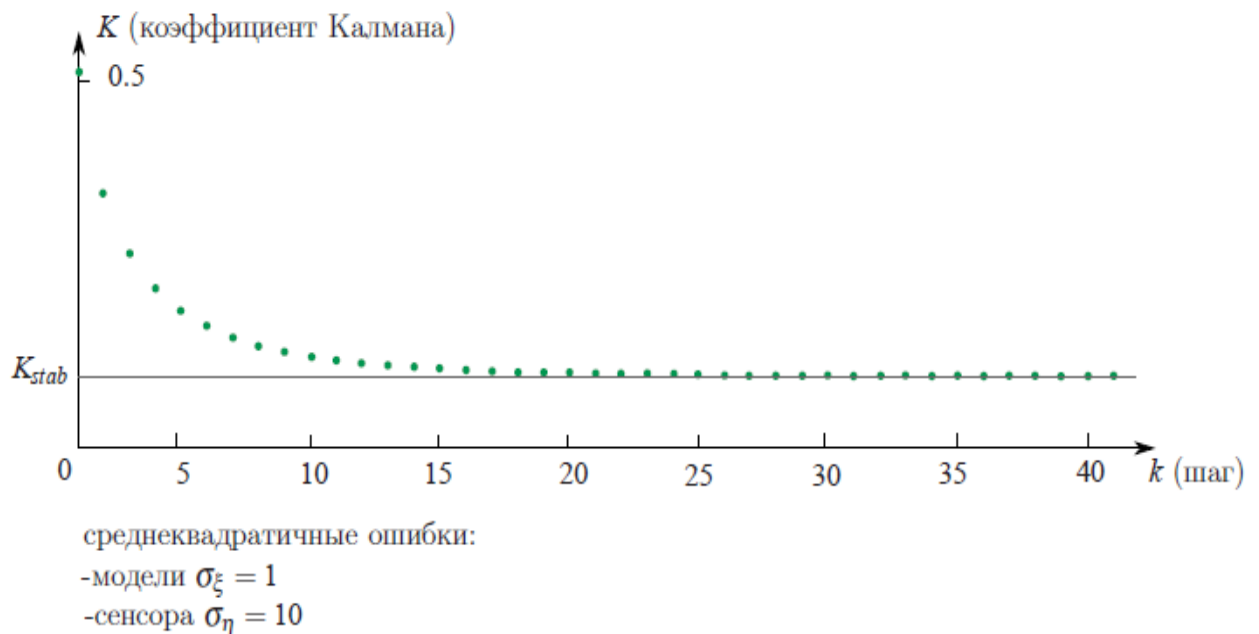


Рисунок 2 – Процесс стабилизации коэффициента Калмана K около значения $K_{(stab)}$

В таком случае, рабочая формула приобретает вид

$$x_{k+1}^o = K_{(stab)} * z_{(k+1)} + (1 - K_{(stab)}) * (x_k^o) \quad (8)$$

Разрабатываемую систему можно условно разделить на три части: подсистема считывания данных, подсистема передачи данных и подсистема визуализации данных. Схема системы с разделением на подсистемы приведена на рисунке 3.



Рисунок 3 – Структурная схема разрабатываемой системы с разделением на подсистемы

Подсистема считывания данных представляет из себя пять трёхосевых датчиков ускорения (акселерометров), подключенных к расширителю I²C - соединения. Каждый из этих датчиков расположен на одной из контрольных точек ноги испытуемого (см. рисунок 1). Нумерация выполнена по нисходящему принципу, бедро, верхняя и нижняя точки колена, пятка и носок имеют номера от одного до пяти соответственно.

В качестве используемых акселерометров выбраны GY-521 (рисунок 4) на основе чипа MPU-6050 от TDK Corporation. Производитель чипа MPU-6050 заявляет следующие характеристики:

1. Напряжение питания: 2.750 до 3.46 вольт;
2. Потребляемый ток: до 4 миллиампер;
3. Потребляемый акселерометром ток: до 500 наноампер;
4. Диапазоны работы акселерометра: +/- 2g, +/- 4g, +/- 8g, +/- 16g;
5. Размеры: 4x4x0.9 миллиметров.
6. Высокоточный аналого-цифровой преобразователь

7. С целью энергосбережения акселерометр может быть переведен в иные режимы работы с пониженной частотой измерений.

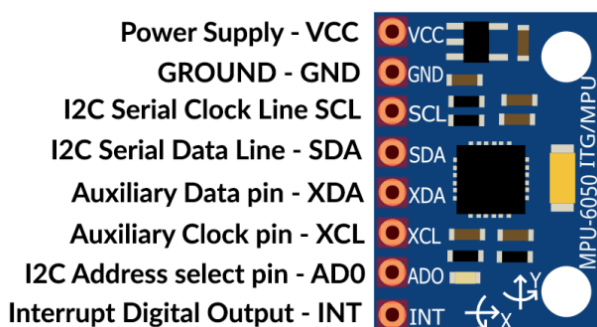


Рисунок 4 – схема модуля GY-521 с маркерами выходных пинов

Подсистема передачи данных состоит из двух модулей беспроводной связи ESP-01, один из них выступает в роли устройства управления акселерометрами, тогда как второй – в роли устройства приема данных для передачи их в подсистему визуализации, и расширителя I^2C -шины. Присутствие в системе расширителя I^2C -шины обусловлено тем фактом, что модули GY-521, в частности микросхемы MPU-6050, имеют один серьезный недостаток: все микросхемы имеют одинаковый адрес в шине I^2C , что не позволяет применять больше одной такой микросхемы в шине. При использовании расширителя микроконтроллеру, управляющему акселерометрами, для обращения к любому из ведомых устройств достаточно обратиться по единому адресу расширителя, указав ему номер выходного порта, к которому нужно обратиться. В качестве модуля расширения расширителя I^2C -шины используется модуль CJMCU-9548 на базе микросхемы TCA9548A от Texas Instruments.

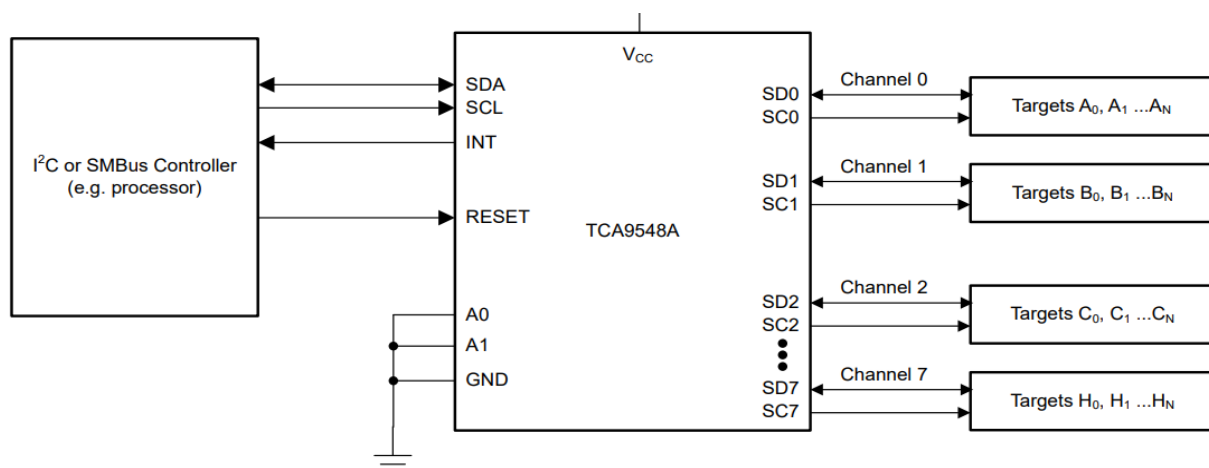


Рисунок 5 – Упрощенная диаграмма примера использования CJMCU-9548

Отдельно следует рассмотреть описание модулей ESP-01, используемых для беспроводной передачи данных от подсистемы считывания данных к подсистеме визуализации. Данный модуль был разработан командой AI-thinker Team на основе микроконтроллера ESP8266 на ядре Tensilica L106. Данный модуль предназначен для взаимодействия с сетями Wi-Fi и способен создавать сети, выступая в роли точки доступа и, при необходимости, сервера, равно как и взаимодействовать с существующими Wi-Fi сетями в роли клиента.

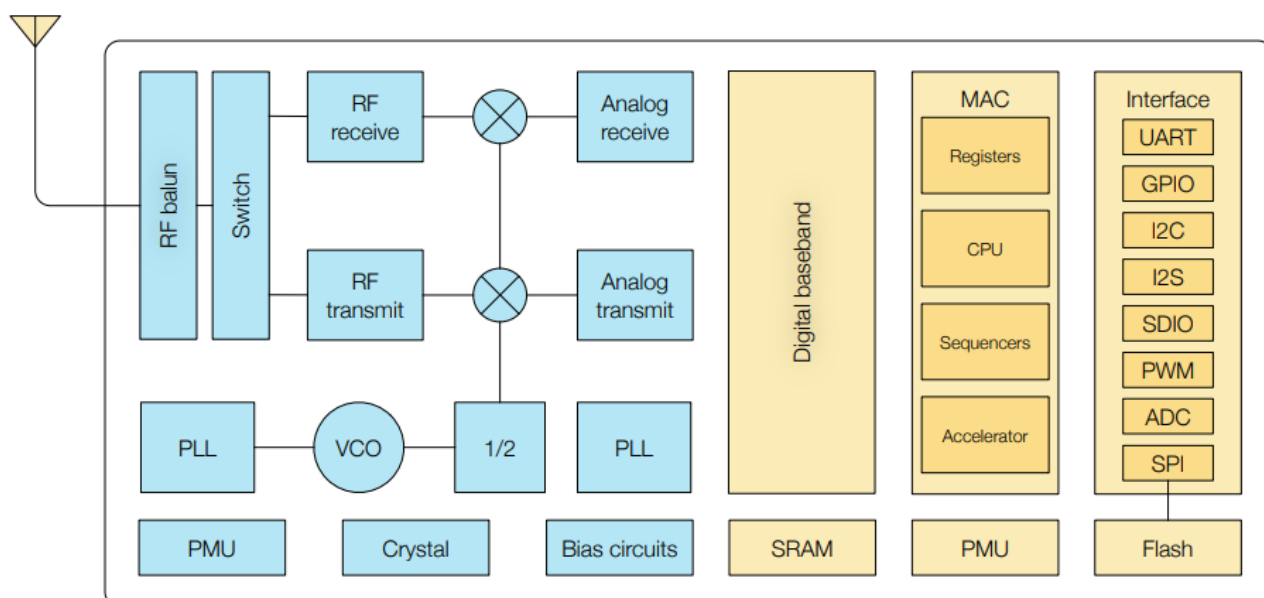


Рисунок 6 – функциональная диаграмма микроконтроллера ESP8266

Подсистема визуализации представляет собой два приложения, выполняемые на персональном компьютере, условно называемых «обработчик сырых данных» и «отрисовщик».

Обработчик сырых данных это приложение, которое принимает данные от UART интерфейса и выполняет описанные в предыдущей главе математические операции над набором данных. Отрисовщик представляет собой приложение, которое на основе данных о координатах опорных точек, закрепленных на ноге испытуемого, строит визуализацию движения. В качестве отрисовщика используется приложение, использующее графический движок Unity. Unity (от англ. «единство») — это кроссплатформенная среда для разработки компьютерных игр, созданная американской компанией Unity Technologies.

Рассмотрим более подробно то, как персональный компьютер под управлением операционной системы Windows получает данные по UART-интерфейсу от модуля ESP-01, выступающего в роли сервера в подсистеме передачи данных. В разрабатываемой системе в роли UART-COM преобразователя использован модуль на основе микросхемы CH340.

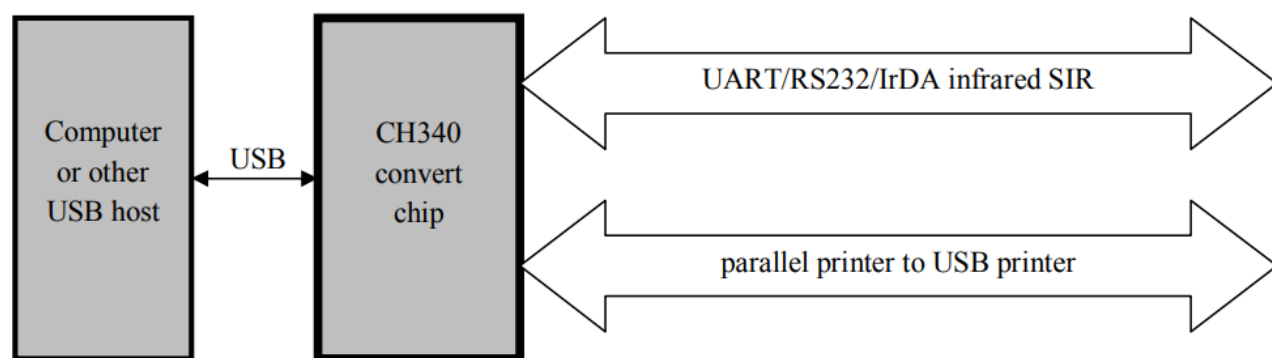


Рисунок 7 – схема примера применения микросхемы CH340

Также в работе приведены схемы разработанной системы в разных нотациях: диаграмма состояний, DFD и IDEF0.

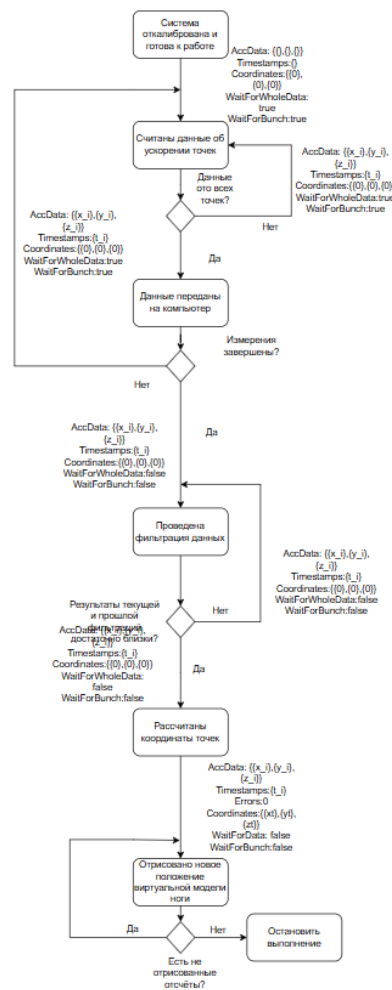


Рисунок 8 – Диаграмма состояний разрабатываемой системы

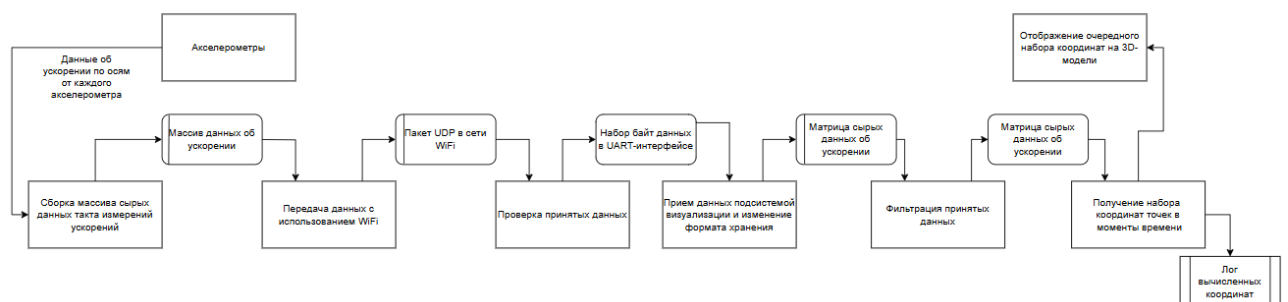


Рисунок 9 – DFD диаграмма разрабатываемой системы

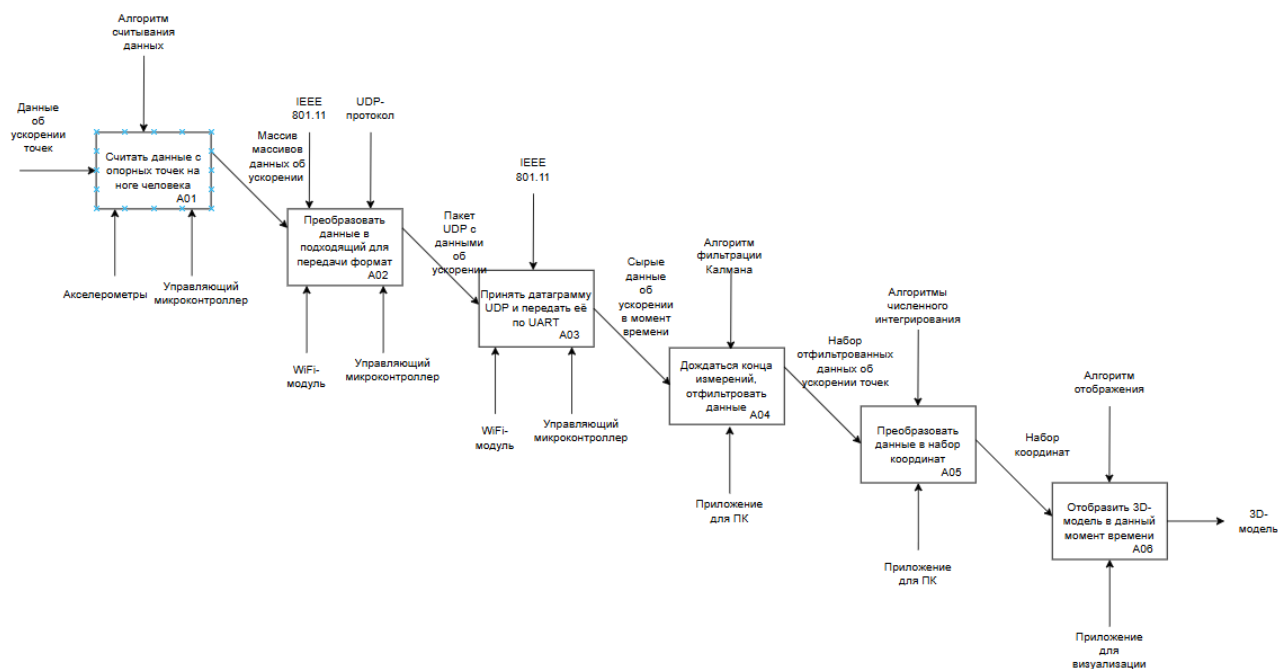


Рисунок 10 – Диаграмма разрабатываемой системы в нотации IDEF0, уровень A0-A6

В разработанной системе присутствует четыре программных модуля: два являются интегрируемым программным обеспечением (прошивками) для микроконтроллеров ESP8266, входящих в используемые модули ESP-01, другие два – модуль обработки сырых данных и модуль отрисовки подсистемы визуализации. Встраиваемое программное обеспечение написано на языке C, программное обеспечение для обработки сырых данных и отрисовки на языке C#.

Завершающей частью работы является раздел с описанием результатов испытаний системы. Начнем рассмотрение результатов работы с содержимого файла, данные в котором являются основой для построения визуализации движений контрольных точек на ноге испытуемого. Данные в этом файле хранятся в строках, имеющих вид $(x_0, y_0, z_0, x_1, y_1, z_1, \dots, x_4, y_4, z_4)$, то есть представляют последовательно записанные координаты каждой контрольной точки. На рисунке 11 представлена случайно выбранная вырезка данных из одной из записей движения ноги испытуемого.

0.008965	0.729328	0.100338	0.007932	0.523773	0.109110	0.008059	0.383027	0.076842	0.009323	0.088612	-0.015586	0.009175	0.088569	0.160520
0.009467	0.729389	0.100003	0.007826	0.523666	0.109000	0.009134	0.382750	0.077188	0.008346	0.087489	-0.015749	0.007770	0.087060	0.161137
0.007543	0.730588	0.100151	0.008997	0.523576	0.108248	0.008353	0.382745	0.077055	0.009431	0.087775	-0.016071	0.008426	0.088773	0.160120
0.008746	0.729608	0.100744	0.008545	0.524352	0.109129	0.008136	0.383677	0.076765	0.008478	0.088479	-0.017141	0.007985	0.088460	0.159721
0.008782	0.729189	0.101108	0.007965	0.523664	0.108225	0.009469	0.383627	0.076928	0.007864	0.087945	-0.016211	0.009511	0.087171	0.160090
0.009185	0.730089	0.100309	0.008839	0.524259	0.109317	0.009093	0.382628	0.076972	0.008619	0.088368	-0.016642	0.009358	0.086804	0.158397
0.007784	0.730092	0.100044	0.008591	0.523788	0.109582	0.008718	0.382708	0.077399	0.008926	0.089277	-0.016900	0.008560	0.087376	0.159767
0.008693	0.730336	0.100392	0.008500	0.524592	0.109388	0.008472	0.382916	0.076935	0.009056	0.088319	-0.017588	0.009418	0.086703	0.158284
0.007656	0.729657	0.099676	0.008070	0.523133	0.108256	0.008486	0.383320	0.077164	0.008062	0.088873	-0.017621	0.008388	0.088279	0.158707
0.008772	0.730669	0.100089	0.007517	0.523299	0.108855	0.009362	0.382302	0.076980	0.009377	0.088385	-0.018804	0.009203	0.087340	0.158351
0.008594	0.729346	0.101329	0.008964	0.523470	0.108787	0.007739	0.382925	0.077908	0.008772	0.089317	-0.019834	0.008798	0.086249	0.157482
0.007600	0.729233	0.100099	0.009257	0.523702	0.108889	0.009328	0.381742	0.076545	0.008701	0.089495	-0.018786	0.009170	0.087128	0.157854
0.008133	0.729356	0.100241	0.009416	0.524112	0.108241	0.009106	0.381981	0.077876	0.009066	0.088870	-0.018534	0.009518	0.087040	0.156294
0.008544	0.729400	0.101352	0.007991	0.522892	0.109855	0.009007	0.382090	0.077398	0.008961	0.089053	-0.019683	0.008981	0.085983	0.157317
0.007933	0.730640	0.100108	0.007539	0.522946	0.107962	0.008393	0.382992	0.076934	0.007485	0.088639	-0.019790	0.008518	0.086722	0.155623
0.008395	0.729358	0.101308	0.008658	0.523923	0.109791	0.007761	0.382898	0.077632	0.008573	0.090286	-0.019313	0.009113	0.087645	0.156117
0.009307	0.729408	0.101568	0.008076	0.524556	0.109729	0.009178	0.382175	0.076686	0.008853	0.089419	-0.020707	0.008958	0.087579	0.156291
0.007814	0.729527	0.100912	0.008883	0.524747	0.109432	0.008597	0.381129	0.076173	0.007580	0.090117	-0.021341	0.008690	0.085551	0.154891
0.008836	0.729204	0.101060	0.007538	0.524703	0.109523	0.008335	0.382696	0.077012	0.008733	0.089878	-0.021191	0.008756	0.087185	0.155081
0.008622	0.730099	0.101855	0.009084	0.524186	0.109978	0.008766	0.381735	0.076843	0.008438	0.089447	-0.022044	0.008946	0.086457	0.155937
0.008974	0.729122	0.101238	0.007853	0.524699	0.109220	0.007990	0.380987	0.077152	0.007957	0.090225	-0.021134	0.007934	0.087140	0.155316
0.009046	0.729506	0.100784	0.007721	0.524368	0.108692	0.007912	0.382391	0.076370	0.008023	0.090009	-0.021313	0.007914	0.086636	0.154911
0.009307	0.730739	0.101348	0.008684	0.522927	0.108928	0.008163	0.381170	0.077477	0.008054	0.090418	-0.021729	0.008971	0.086134	0.154769
0.007748	0.729173	0.100801	0.008535	0.524399	0.109853	0.007870	0.381113	0.077789	0.008437	0.089592	-0.021927	0.008264	0.086392	0.154790
0.008199	0.729257	0.101421	0.008566	0.523384	0.108601	0.009026	0.380933	0.077336	0.007421	0.091224	-0.022825	0.008031	0.085920	0.154130
0.007735	0.730195	0.101308	0.007523	0.524009	0.109516	0.009278	0.381289	0.077030	0.008865	0.089638	-0.022760	0.009174	0.085635	0.154141
0.008465	0.729833	0.102572	0.008031	0.523131	0.109178	0.009149	0.381172	0.076986	0.007669	0.090061	-0.022731	0.009339	0.085541	0.152072
0.009024	0.729200	0.101427	0.008579	0.522744	0.109731	0.007579	0.382044	0.077907	0.008766	0.090965	-0.023881	0.008380	0.086466	0.152837
0.008465	0.729440	0.102302	0.008774	0.523318	0.108202	0.007729	0.381914	0.078061	0.009064	0.091432	-0.023096	0.008056	0.086121	0.152095
0.009460	0.730833	0.102497	0.008003	0.523775	0.108357	0.008549	0.382369	0.076510	0.007422	0.090493	-0.023949	0.009170	0.085250	0.152664
0.007643	0.729813	0.102211	0.009061	0.523128	0.108337	0.009183	0.381662	0.077290	0.008091	0.091134	-0.023690	0.008379	0.085289	0.151932
0.009133	0.730293	0.102551	0.009018	0.523891	0.109627	0.009207	0.380543	0.077325	0.007638	0.090946	-0.024890	0.008405	0.085111	0.152321
0.009285	0.729927	0.102849	0.007545	0.522714	0.108308	0.008804	0.380362	0.076463	0.008673	0.092020	-0.025487	0.008220	0.084434	0.151348
0.009143	0.729564	0.101334	0.007615	0.523558	0.109753	0.008288	0.380520	0.077288	0.007879	0.092257	-0.024514	0.008892	0.084435	0.151472
0.009571	0.730152	0.102720	0.007850	0.523623	0.109434	0.009042	0.381318	0.077102	0.009193	0.091986	-0.024739	0.007680	0.084380	0.150346
0.008881	0.730997	0.102786	0.008810	0.523760	0.108244	0.009163	0.380302	0.077980	0.008285	0.092435	-0.025426	0.009223	0.085933	0.149232
0.008485	0.729426	0.101629	0.007705	0.523671	0.109557	0.009343	0.381130	0.076254	0.009059	0.092083	-0.025473	0.007730	0.084303	0.149224
0.008217	0.729078	0.102355	0.009111	0.523743	0.109800	0.008481	0.381947	0.077546	0.008288	0.090923	-0.025598	0.008553	0.085772	0.149622
0.009618	0.730776	0.102319	0.008363	0.522577	0.110149	0.008270	0.380323	0.077267	0.008083	0.092674	-0.027724	0.009207	0.084417	0.150059
0.009538	0.730505	0.102948	0.008525	0.523302	0.109166	0.008116	0.381649	0.076956	0.008804	0.092359	-0.026368	0.008114	0.084540	0.148519
0.009219	0.730108	0.102385	0.009327	0.523806	0.109523	0.008844	0.380226	0.077798	0.009151	0.092394	-0.027894	0.008245	0.084596	0.149268
0.008951	0.730484	0.103588	0.008191	0.523347	0.109199	0.007979	0.380229	0.076327	0.008662	0.092023	-0.027237	0.009143	0.083785	0.149100
0.008555	0.729846	0.103542	0.009155	0.523411	0.110045	0.007701	0.380564	0.077065	0.008797	0.091570	-0.029017	0.008578	0.083510	0.148579
0.008944	0.730575	0.103517	0.009204	0.523410	0.108441	0.007548	0.379786	0.077073	0.007230	0.093003	-0.028573	0.008799	0.085052	0.147570

Рисунок 11 – Случайная вырезка из набора координат, описывающего движение конечности.

На основе данных, представленных в таком формате, отрисовщик строит трехмерную визуализацию движения используя код из листинга 6. На рисунках 12–16 приведены несколько положений в условиях одного движения.

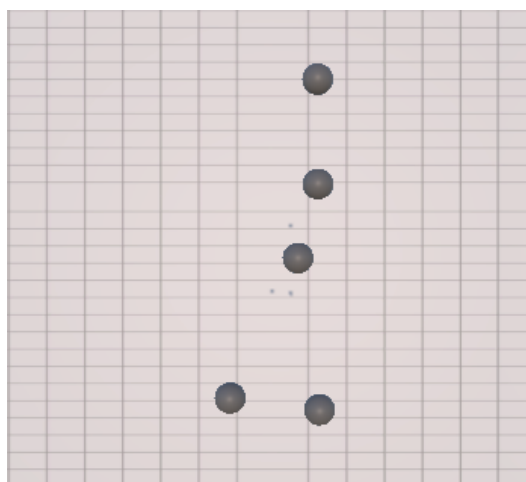


Рисунок 12 – виртуальное отображение движения ноги, 1 из 5

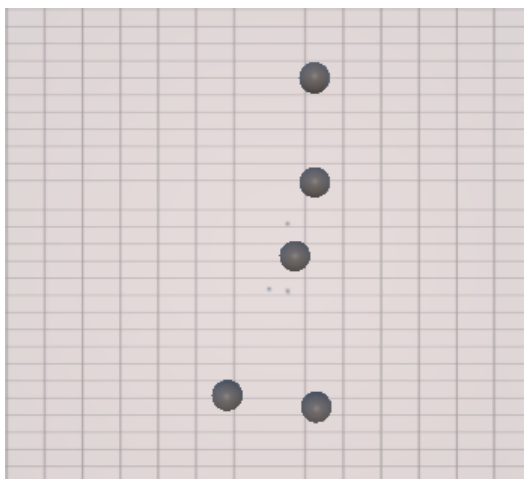


Рисунок 13– виртуальное отображение движения ноги, 2 из 5

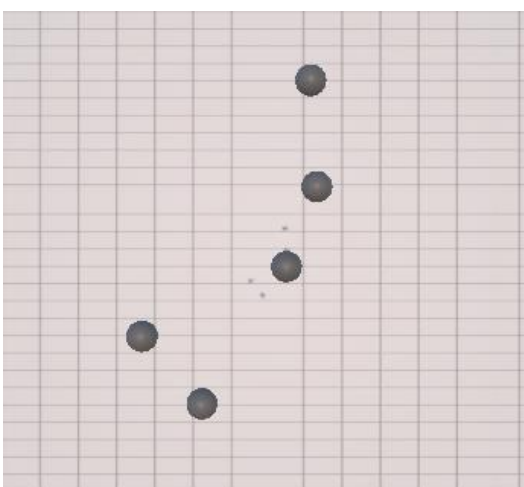


Рисунок 14 – виртуальное отображение движения ноги, 3 из 5

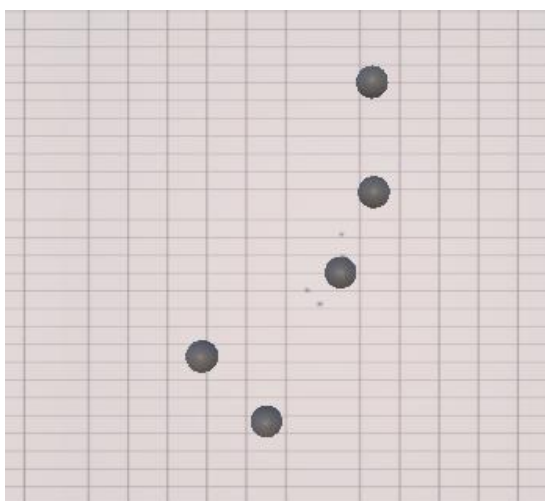


Рисунок 15 – виртуальное отображение движения ноги, 4 из 5

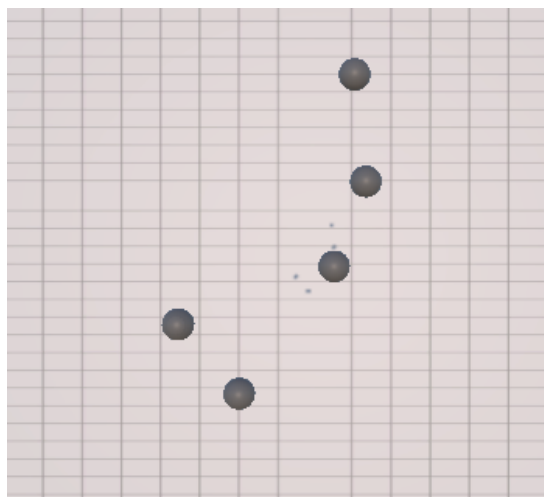


Рисунок 16– виртуальное отображение движения ноги, 5 из 5

На рисунке 17 изображено наложение фотографии реального положения ноги испытуемого, носящего систему для считывания движений и виртуальной визуализации. Реальным положением датчиков являются центры темных кругов.



Рисунок 17 – Наложение фотографии ноги испытуемого и виртуального отображения

Заключение

В данной работе описано устройство, математические основы, программные решения и основные принципы работы системы считывания движений нижней конечности человека.

Результатом теоретического исследования стал выбор инерциального метода считывания данных о движении, реализуемый с использованием акселерометров, расположенных в контрольных точках на исследуемой конечности.

В тексте работы приведены:

1. Математическая модель вычисления координат из сведений об ускорении;
2. Математическая модель фильтра Калмана, используемого для уменьшения влияния погрешностей измерения;
3. Структурная схема системы;
4. DFD и IDEF0 диаграммы;
5. Диаграмма состояний системы;
6. Описаны основные физические компоненты системы, обоснован выбор тех или иных устройств для считывания и передачи данных.

Также в данной работе подробно описаны принципы работы программного кода и используемые программные инструменты.

Для получения и демонстрации результатов проведены испытания системы, проведен анализ данных с использованием программного кода системы и визуализированы движения, выполненные в процессе испытаний.