

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

_____ О. В. Непомнящий

подпись инициалы, фамилия

« _____ » _____ 2020 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Программно-аппаратный комплекс контроля выполнения лечебно-
восстановительных упражнений

Тема

09.04.01 «Информатика и вычислительная техника»

код и наименование направления

09.04.01.06 «Микропроцессорные системы»

код и наименование магистерской программы

Научный
руководитель

подпись, дата

профессор, зав кафедры, к.т.н
должность, ученая степень

О.В. Непомнящий
инициалы, фамилия

Выпускник

подпись, дата

А.В. Осипов
инициалы, фамилия

Рецензент

подпись, дата

К.Т.Н., доцент
должность

Э.Ф. Цокур
инициалы, фамилия

Нормоконтролер

подпись, дата

О.В. Непомнящий
инициалы, фамилия

Красноярск 2020

Федеральное государственное автономное
образовательное учреждение
высшего профессионального образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О. В. Непомнящий

подпись инициалы, фамилия

« ____ » _____ 2019 г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме _____ магистерской диссертации

Студенту _____ Осипову Антону Владимировичу _____

фамилия, имя, отчество

Группа КИ18-01-6М Направление (специальность) _____ 09.04.01
номер код

«Информатика и вычислительная техника»

наименование

Тема выпускной квалификационной работы «Программно-аппаратный комплекс контроля выполнения лечебно-восстановительных упражнений»

Утверждена приказом по университету № 19249/с от 21.12.2018

Руководитель ВКР Непомнящий О.В. профессор, зав. кафедры ВТ ИКИТ СФУ,
к.т.н. _____

инициалы, фамилия, должность, ученое звание

Исходные данные для ВКР разработка программно-аппаратного комплекса контроля выполнения лечебно-восстановительных упражнений.

Перечень разделов ВКР Реферат, Введение, Анализ задания на магистерскую диссертацию, Разработка лабораторного стенда, Разработка комплекса, Инструкция для пользователя, Заключение, Список используемых источников.

Перечень графического материала Презентация, демонстрационное видео.

Руководитель ВКР

подпись

О.В. Непомнящий

инициалы, фамилия

Задание принял к исполнению

подпись

А.В. Осипов

инициалы, фамилия

« ____ » _____ 2019

РЕФЕРАТ

Выпускная квалификационная работа по теме «Программно-аппаратный комплекс контроля выполнения лечебно-восстановительных упражнений» содержит в себе 47 страниц текстового документа, 12 использованных источников, 17 иллюстраций, 8 таблиц, 4 приложения.

ВИДЕОПОТОК, СЕНСОР, СУСТАВЫ ТЕХНИЧЕСКОЕ ЗРЕНИЕ, РАСПОЗНАВАНИЕ, ИЗОБРАЖЕНИЕ.

– Целью проводимых в рамках магистерской диссертации исследований является разработка программно-аппаратного комплекса контроля выполнения лечебно-восстановительных упражнений. Для достижения поставленной цели определены следующие основные задачи ВКР:

- Выбор элементов и узлов комплекса.
- Выбор параметров и характеристик, которым должен удовлетворять моделируемый комплекс на основе анализа имеющихся решений.
- Разработка и моделирование метода.
- Разработка демонстрационного макета
- Проверка работоспособности разработанной системы.

Во введении раскрывается актуальность работы.

В первой главе обозначены цели и задачи приведен анализ существующих систем.

Во второй главе описан процесс разработки комплекса, выбор аппаратной части и операционной системы.

В третьей главе описаны программные модули комплекса.

В четвертой главе составлена инструкция для пользователя.

В итоге выполнения ВКР разработан программно-аппаратный комплекс контроля выполнения лечебно-восстановительных упражнений. Разработана инструкция для пользователя

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Анализ задания на магистерскую диссертацию	5
1.1 Цель и задачи	5
1.2 Обзор существующих аналоговых систем	5
1.2.1 Обзор существующего аппаратного обеспечения	5
1.2.2 BP YouGrabber Jintronix	7
1.3 Требования к программно-аппаратному комплексу	9
1.4 Выводы по главе	11
2 Разработка лабораторного стенда	12
2.1 Блок обработки данных	12
2.1.1 Выбор одноплатного компьютера	14
2.1.2 Выбор операционной системы	16
2.2 Выбор устройства технического зрения	20
2.2.1 Microsoft Kinect	20
2.2.2 Asus Xtion	22
2.2.3 Z-cam	21
2.3 Выводы по главе	23
3 Разработка комплекса	25
3.1 Описание программных модулей	25
3.1.1 Описание модуля первичной обработки	26
3.1.2 Описание модуля вторичной обработки	28
3.1.3 Описание программы редактора упражнений	30
3.1.3.1 Описание выходного файла программы редактор	33
3.1.4 Описание программы котроллера	35
3.2 Алгоритм работы программы котроллера	37
3.3 Выводы по главе	38

4 Инструкция для пользователя	39
4.1 Инструкция для создания файла упражнений доктором	39
4.2 Инструкция по сбору комплекса и запуску системы	41
4.3 Инструкция по работе с модулями обработки программой редактор.....	42
4.4 Выводы к главе	44
ЗАКЛЮЧЕНИЕ	45
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	46

ВВЕДЕНИЕ

В современном мире технологии занимают центральное место в жизни человека. Благодаря технологиям темпы развития очень быстро растут в различных сферах: образование, культура, наука, спорт, строительство медицина.

Быстрый темп жизни и неблагоприятные экологические изменения, частые стрессовые ситуации - все это приводит к возникновению серьезных заболеваний у человека. На сегодняшний день скорость развития медицины и медицинского приборостроения очень высока. Однако стоимость лечения и медицинского оборудования так же достаточно велика. Очень многие люди, которым необходимо медицинское оборудование не могут себе позволить столь дорогостоящее лечение, что значительно ухудшает процесс реабилитации [1].

Одним из методов восстановления после определенных травм или болезней, таких как инсульт является лечебная физкультура или ЛФК. ЛФК-это специально разработанный комплекс физических упражнений, выполняющихся под присмотром специалиста. Они помогают устранить дегенеративные изменения в тканях и органах, позволяют справиться с атрофией. ЛФК способствует нормализации обмена веществ, улучшает работу сердечно-сосудистой системы, укрепляет мышцы и способствует расслаблению нервной системы [2].

По статистике ежегодно около полумиллиона человек на всей территории Российской федерации переносят заболевания, связанные с сердечно-сосудистой системой, такие как инсульт, инфаркт и т.д. Из них полностью восстанавливаются только 13-15%, остальная часть пациентов либо становятся инвалидами, либо болезнь заканчивается летальным исходом. [1] Однако тот небольшой процент перенесших инсульт и восстановившихся полностью, после не могут чувствовать себя в полной безопасности, ведь у 50% из них в ближайшие пять лет происходит повторный инсульт [2].

По данным, предоставленным Минздравом, 31% пациентов, перенесших инсульт, необходим специальный уход, 20% не могут самостоятельно передвигаться, и только 8% восстанавливаются и могут жить полноценной жизнью. Так же спортивный травматизм на сегодня составляет от 5 до 7% травматизма в целом [2].

А если вспомнить о все возрастающей статистике ДТП на наших дорогах и о случайных повреждениях и переломах, особенно в зимний период, становится ясно, что травмы может получить любой человек, даже не увлекающийся профессиональным спортом.

Всем этим людям необходим курс реабилитации для выздоровления и возможности жить полноценно в дальнейшем, однако не у всех есть возможность каждый день несколько часов присутствовать в медицинском учреждении для занятия ЛФК под присмотром специалиста. Актуальной становится задача разработки программно-аппаратного комплекса, позволяющего формировать индивидуальный список упражнений и систему тренировок, наиболее подходящей для пациента с учетом его физиологических особенностей и специфике травмы или заболевания, и контролировать прогресс пациента.

1 Анализ задания на магистерскую диссертацию

1.1 Цель и задачи

Целью выпускной магистерской диссертации является разработка программно-аппаратного комплекса для контроля выполнения лечебно-восстановительных упражнений.

Для реализации данной цели определены следующие задачи:

1. Выбор элементов и узлов комплекса
2. Выбор параметров и характеристик, которым должен удовлетворять моделируемый комплекс на основе анализа имеющихся решений
3. Разработка и моделирование метода
4. Разработка демонстрационного макета
5. Проверка работоспособности разработанной системы.

1.2 Обзор существующих систем

1.2.1 Reflexion

Reflexion Health применяется для медицинских целей, чтобы сделать физическую терапию и лечебную профилактическую физкультуру, более эффективной для пациентов, - и удобнее на практике для работников медицины. Он сочетает технологию датчиков Kinect для Windows и комплекта программного обеспечения с его программным обеспечением для предоставления интерактивного решения, которое может помочь пациентам и докторам улучшить и упростить процесс физической терапии и оздоровительной физкультуры. Данная технология была разработана в 2011 году командой Reflection совместно с доктором Эриком Хофмайстером, директором кафедры ортопедической хирургии в Военно-морском ортопедическом центре в Сан Диего,

Калифорния. Данная технология впервые прошла клинические испытания в октябре 2012 года [3].

Достоинства данного программного продукта:

1. Возможность создания планов работы для каждого пациента:
2. Используя Kinect для Windows, данный продукт позволяет специалистам-медикам планировать процедуры и график работы с пациентами, и в конечном итоге предлагают возможность удаленного мониторинга пациентов;
3. Помощь пациентам и контроль за правильным выполнением упражнений:
4. При помощи Reflection health становится возможным удаленно контролировать пациента и выполнение им определенных упражнений;
5. Снижение затрат на лечение.

С помощью Kinect и решения Reflection health пациент может проходить курс восстановительной терапии без значительных затрат денежных средств, что делает данное решение доступным практически для каждого [3].

Программное и аппаратное обеспечение Reflection health:

Аппаратное обеспечение - Kinect for Windows Sensor, Компьютер с операционной системой Windows 7 и выше.

Программное обеспечение: Microsoft Kinect for Windows, Microsoft .NET Framework, Microsoft Visual Studio, Microsoft DirectX, Microsoft Remote Desktop Protocol.

Графическое представление Reflection health представлено на рисунке 1.



Рисунок 1 - Пример работы решения Reflection health

1.2.2 BP YouGrabber Jintronix

Система BP YouGrabber Jintronix (YouRehab Ltd., Швейцария) представляет собой перчатки с датчиками, в которую заложена программа с различными играми для обучения, как это показано на рисунке 2.

Jintronix была первоначально разработана для реабилитации постинсультных больных, что очень удобно и обеспечивает выбор игр/упражнений для тренировки сидя. Пять игр: Fish Frenzy, Ball Maze, Garden Grab, Bike Barrier, Kitchen Clean-up. Сложность игры контролируется таким образом, что, если пациент может выполнить игру с легкостью, уровень сложности игры увеличивается, требуя больше скорости, расстояния и точности, постепенно развивая его моторику. Система Jintronix дает обратную связь, что позволяет наблюдать как пациент выполнил то или иное задание в игре [4]. Исследования показали, что использование данного продукта, в комплексе с традиционными методами реабилитации, позволяет получить больше преимуществ в восстановлении двигательной функции. Jintronix обеспечивает выполнение различных видов двигательных функций пациента, проведение реальных

манипуляций объектами, взаимодействуя с виртуальным сценарием. Движения контролируются с помощью системы захвата движений [4]. Например, простое движение, такое как поставить стакан на полку, представлено виртуальным стаканом и полкой. Физиотерапевт держал в руке реальный стакан с датчиками, расположенными на объекте и совершал акт размещения стакана на виртуальной полке. Виртуальный сценарий отображал правильный путь движения стекла по направлению к полке. Пациенту требовалось повторить правильное движение, выполненное заранее терапевтом. Правильная траектория отображалась в фоновом режиме виртуальной сцены, чтобы облегчить восприятие и корректировать движения пациента. Принимая во внимание ошибки пациента, терапевт выбирал характеристики и сложность двигательных задач путем изменения положения или ориентации виртуального объекта. Задачи усложняются добавлением объектов в виртуальном сценарии, благодаря чему пациенты были вынуждены активировать различные группы мышц рук, чтобы выполнить более сложные задачи. Роль терапевта заключается в управлении виртуальной средой, чтобы адаптировать её физическому состоянию пациента и давать пациенту инструкции в случае возникновения трудностей во время выполнения интерактивных упражнений. После выполнения упражнений терапевт обсуждает с пациентами результаты, полученные в ходе сеанса терапии [4].

Jintronix. ВР состоит из программного обеспечения (Jintronix, Монреаль, PQ) и трехмерного изображения (Kinect v2, MicrosoftCanadaCo., Mississauga, ON). А камера захватывает движения участника и позволяет пациенту контролировать аватар, который взаимодействует с игрой [4].

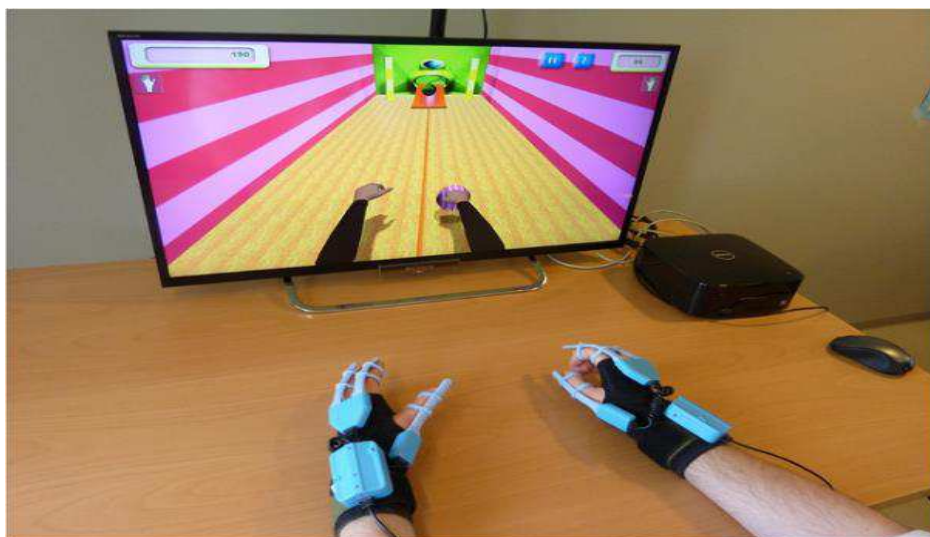


Рисунок 2 – Пример работы системы Jintronix

1.3 Требования к программно-аппаратному комплексу

Рассмотрев существующее программно-аппаратные обеспечения, было принято решение создать собственный программно-аппаратный продукт для реабилитации пациентов, основанный на методе обработки изображения.

Данный комплекс должен обладать следующими свойствами:

Требования к аппаратной части комплекса:

1. Обязательной частью комплекса должен являться орган технического зрения, способный передавать видеопоток в режиме реального времени;
2. Комплекс должен обладать мобильностью;
3. Для управления процессом необходимо устройство дистанционного управления пользовательским ПО;

Требования к программной части комплекса:

1. Для реализации данного комплекса необходим программный модуль непрерывной передачи видеопотока в режиме реального времени;
2. Необходим модуль обработки видеопотока для получения возможности отслеживать движения пользователя в кадре;

3. Для полноценного функционирования комплекса необходимо два программных модуля: один для создания и редактирования упражнений, другой для непосредственного контроля правильности выполнения упражнений

Для полноценного функционирования комплекса было принято решение разместить программную часть комплекса на одноплатном компьютере или микроконтроллере для мобильности, создать программные модули передачи и обработки видеопотока в режиме реального времени на основе встроенных библиотек Open NI, и Nite 2. Для корректной работы комплекса по отслеживанию выполнения упражнений при выполнении их пациентом, а также по созданию и редактированию упражнений доктором на протяжении курса реабилитации было принято решение разработать программу «Редактор» для создания и редактирования упражнений и программы «Контроллер» для отслеживания правильности выполнения упражнений.

Требования к программе «Редактор»:

1. Наличие визуального редактора–аватара для создания и редактирования упражнений;
2. Возможность выбирать только суставы, координаты которых необходимы при создании упражнения;
3. Выходные данные программы – файл описания упражнения: координаты положения суставов в момент конечной фазы выполнения упражнения;
4. Функционал для анализа файлов, которые будут созданы программным модулем контроля после выполнения упражнения пользователем.

Требования для программы «Контроллер»:

1. Функционал приема данных, в частности видеопотока и координат суставов пользователя;
2. Возможность выбора упражнений;
3. Индикатор выполнения упражнения;

4. Результаты выполнения упражнения должны сохраняться в файл, содержащий статистическую информацию о проделанных упражнениях, для дальнейшей передачи врачу.

1.4 Выводы по главе

На основе рассмотренных методов реабилитации пациентов, нуждающихся в восстановлении опорно-двигательного аппарата, было принято решение разработать собственный метод реабилитации пациентов с дисфункцией опорно-двигательного аппарата, основанный на принципе определения пространственного положения пациента с помощью органов технического зрения, позволяющий сократить сроки реабилитации и автоматизировать процесс. Для реализации данного метода было принято решение разработать программно-аппаратный комплекс, состоящий из органа технического зрения, одноплатного компьютера пациента с программным обеспечением для контролируемого выполнения упражнений, периферийного устройства вывода информации в виде монитора, устройства управления пользовательским программным обеспечением, а также электронного носителя с файлами упражнений, которые пациенту необходимо выполнить. Для врача необходимо разработать программный модуль для создания описательных файлов упражнений, а также для дальнейшего анализа выходных файлов, полученных от пациента после выполнения. Такой инструмент в руках опытного специалиста позволит редактировать курс реабилитации таким образом, чтобы процесс реабилитации был более рациональным и продуктивным.

2 Разработка лабораторного стенда

Исходя из требований к программно-аппаратному комплексу, сформулированных ранее было принято решение создать лабораторный стенд программно-аппаратного комплекса. Структурная схема лабораторного стенда представлена на рисунке 3. Центральной частью комплекса будет выступать мобильный блок обработки данных. На блоке обработки будут запускаться программные модули первичной и вторичной обработки посредством устройства управления для синхронизации с сенсором Kinect и непрерывной передачи видеопотока, и координат точек скелета человека. Для запуска программы контроллер необходим файл с упражнениями, созданный ранее доктором в программе редактор, этот файл на электронном носителе. Для получения обратной связи необходимо устройство вывода – монитор.



Рисунок 3 – Структурная схема лабораторного комплекса

2.1 Блок обработки данных

Основной задачей магистерской диссертации является усовершенствование готового комплекса, посредством модернизации, а именно, переноса вычислительных процессов на более мобильное устройство. В рамках данного проекта устройство для обработки данных можно разбить на 2 группы:

платы на микроконтроллере (типичным представителем является Arduino Uno) и одноплатные компьютеры (типичный представитель – Raspberry Pi). Микроконтроллеры могут одновременно исполнять всего одну задачу и отлично с этим справляются. А одноплатные компьютеры исполняют программы в рамках операционной системы, обладают большей производительностью и широкими мультимедийными возможностями. Для сравнения устройств была составлена таблица, включающая основные характеристики, необходимые для сравнения в рамках данного проекта [5].

Таблица 1 – Сравнительная таблица микроконтроллеров и одноплатных компьютеров

Характеристика	Микроконтроллер	Одноплатный компьютер
Производительность	1 ядро, десятки-сотни МГц, десятки КБ оперативной и постоянной памяти	1 или более ядер, сотни-тысячи МГц, сотни МБ оперативной памяти, гигабайты постоянной памяти
Многозадачность	Нет, но можно эмулировать	Да, управляется ОС
Работы с интернетом	Обычно нужны дополнительные модули и глубокое знание протоколов	Встроенный сетевой модуль
Языки программирования	Ограниченный. Чаще C/C++	Python, JavaScript, Bash и десятки других: любые доступные в ОС
Работа с видео, компьютерное зрение	Не хватит мощности	OpenCV, аппаратные видеокодеки, HDMI-выход

В рамках выпускной квалификационной работы для блока обработки информации необходимо выбрать устройство для работы с органом технического зрения – сенсором Kinect, поэтому было принято решение за основу блока обработки информации взять одноплатный компьютер.

2.1.1 Выбор одноплатного компьютера

Одноплатный компьютер - компьютер, собранный на одной печатной плате, на которой установлены микропроцессор, оперативная память, системы ввода-вывода и другие модули, необходимые для функционирования компьютера. Одним из первых доступных одноплатных компьютеров стал Raspberry Pi, который на данный момент является самым популярным (продано более 5 миллионов устройств) вышедший в 2012 году. Затем стали появляться его клоны и аналоги [6].

На сегодняшний день существует большое множество моделей одноплатных компьютеров разного размера, функциональности, производительности и стоимости. Некоторые из них являются клонами Raspberry Pi, а некоторые имеют совсем другое назначение и архитектуру. Для выбора одноплатного компьютера была составлена таблица сравнения одноплатного компьютера Raspberry Pi и еще нескольких наиболее популярных на рынке одноплатных компьютеров по критериям, наиболее важным в рамках данного проекта [7].

Таблица 2 – Сравнительная таблица одноплатных компьютеров

Характеристика	Raspberry Pi 3 model B	ODROID-C2	BeagleBone Black Rev. C	A20-OLinuXino-MICRO-4GB
Ориентировочная цена	\$35	\$46	\$45	\$89
Поддержка Windows	Windows 10 IoT Core	-	-	-
Поддержка Linux	есть	есть	есть	есть
Bluetooth	Bluetooth BLE 4.2	-	-	-
USB host	4	4	1	2
Объем ОЗУ	1 Гбайт	2 Гбайт	512 Мбайт	1 Гбайт
Внешняя память	MicroSD	8Гбайт MicroSD,64 Гбайт eMMC	MicroSD	MicroSD

Основой блока обработки данных был выбран Raspberry Pi 3 model B – одноплатный компьютер с поддержкой беспроводной локальной сети и Bluetooth (Рисунок 4) [8].

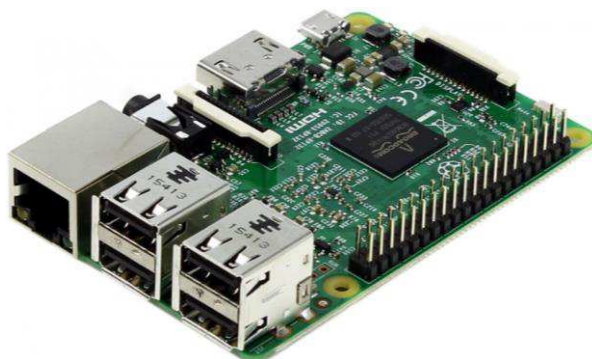


Рисунок 4 – Одноплатный компьютер Raspberry Pi 3 model B

Данный одноплатный компьютер имеет следующие характеристики:

- 4-хъядерный процессор 1,2 ГГц Broadcom BCM2837 64-битный процессор;
- 1 ГБ ОЗУ;
- Беспроводная локальная сеть BCM43438 и Bluetooth Low Energy (BLE);
- 100 базовых сетей Ethernet;
- 40-контактный удлиненный GPIO;
- 4 порта USB 2;
- 4-полюсный стереовыход и композитный видеопорт;
- Полноразмерный HDMI;
- Порт камеры CSI для подключения камеры Raspberry Pi;
- Порт дисплея DSI для подключения сенсорного дисплея Raspberry Pi;
- Порт Micro SD для загрузки операционной системы и хранения данных;
- Источник питания Micro USB до 2,5 А

2.1.2 Выбор операционной системы

Для полноценного функционирования комплекса по мимо выбора аппаратной части необходимо выбрать программное обеспечение – операционную систему, необходимую для функционирования блока обработки данных. Так как основой блока обработки данных был выбран Raspberry Pi 3 model B необходимо выбрать операционную систему для данного одноплатного компьютера. Raspberry Pi является популярным и универсальным устройством, для устройства было разработано несколько операционных систем. Это дистрибутивы Linux, последняя Windows 10 различные версии систем, такие как:

1. Raspbian - основанная на Debian (операционная система, состоящая из свободного ПО с открытым исходным кодом) операционная система для Raspberry Pi. Существует несколько версий Raspbian, в том числе Raspbian Stretch и Raspbian Jessie. С 2015 года Raspbian официально представлена Raspberry Pi Foundation в качестве основной операционной системы для одноплатных компьютеров Raspberry Pi. Raspbian была создана Майком Томпсоном и Питером Грином в качестве независимого проекта. Первоначальная сборка была выполнена в июне 2012 года. Raspbian оптимизирована для низкопроизводительных процессоров ARM, используемых в линейке компьютеров Raspberry Pi. Система разработана специально для этого устройства и имеет все необходимое программное обеспечение. Raspbian содержит такие программы по умолчанию - офисный пакет LibreOffice, веб-браузер, почтовый клиент - Claws Mail, легкое окружение рабочего стола, а также некоторые инструменты для обучения программированию. Интерфейс данной операционной системы представлен на рисунке 5.

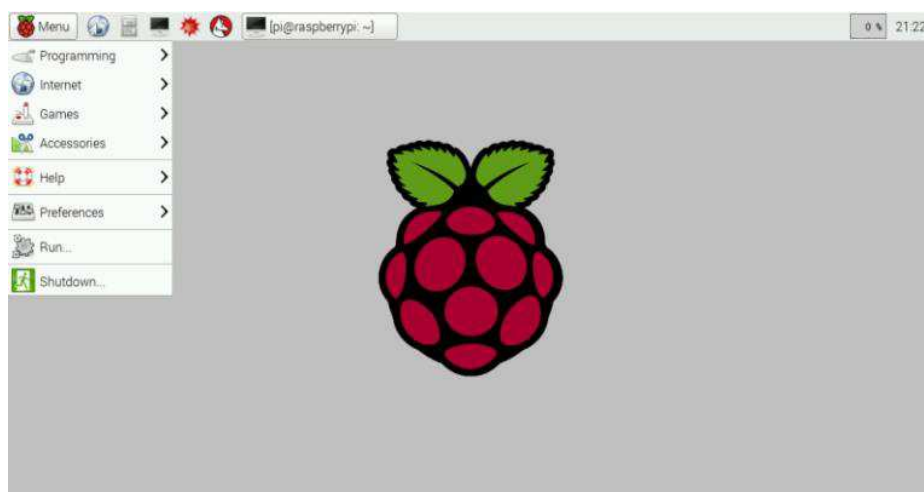


Рисунок 5 – Интерфейс операционной системы Raspbian

2. Windows 10 IoT – операционная система, предлагаемая Microsoft для одноплатных компьютеров и подобных устройств. Windows 10 IoT, с точки зрения совместимости, не отличается от Windows 10 Professional и Windows 10

Enterprise в обычном "настольном" варианте. Версия не является ни "урезанной" ни "компонентной". Отличия только в отсутствии Microsoft Store и связанных с ним приложений (Edge, Почта, Погода, новости и т.п.). Windows 10 IoT является операционной системой семейства Windows Embedded, предназначенной для создания компьютерных устройств фиксированного назначения, таких как: кассовые и POS-системы, системы видеонаблюдения, информационные и платежные киоски, рекламные цифровые панели (Digital Signage), тонкие клиенты, SCADA, системы контроля и управления доступом (СКУД), медицинские устройства, системы электронной очереди, промышленные контроллеры, шлюзы сбора данных в решениях Интернета Вещей и тому подобных устройств. Важным преимуществом является то, что это система Windows, являющаяся наиболее популярной что в рамках данной магистерской диссертации будет плюсом для целевой аудитории. Интерфейс данной операционной системы представлен на рисунке 6.

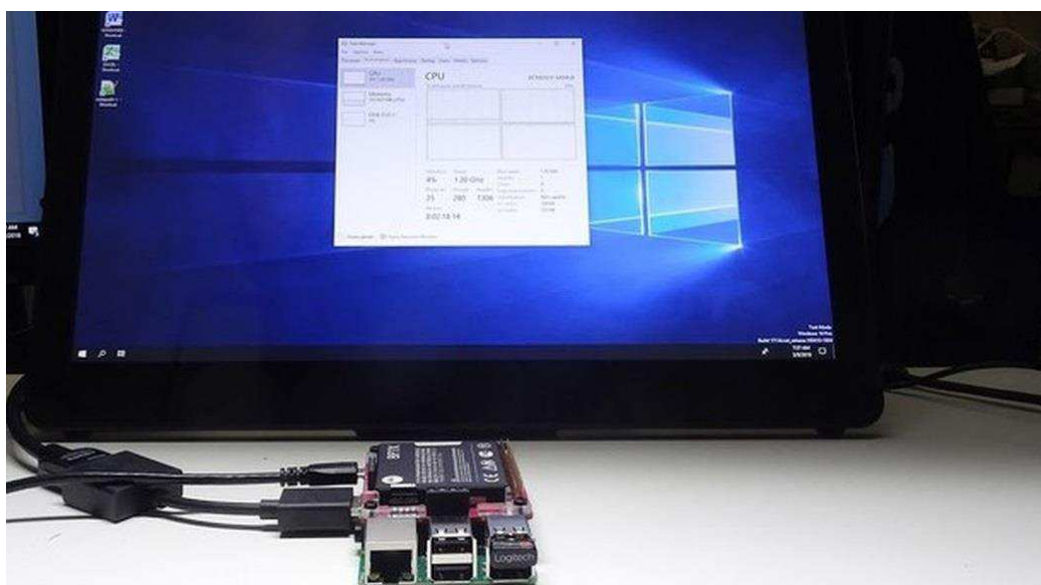


Рисунок 6 – Интерфейс операционной системы Windows 10 IoT

3. ArchLinux - версия ArchLinux для Raspberry Pi основана на проекте ArchLinux ARM. Он почти ничем не отличается от системы для настольного компьютера. Операционная система Raspberry Pi 3 содержит только базовую

систему, которую можно приспособить под себя. В дистрибутиве нет графического окружения, но есть возможность его установить. Метод обновления данной системы - плавающие релизы (роллинг-релизы). Этот метод предусматривает, что операционная система постоянно получает обновления и всегда находится в состоянии самой последней версии. А также может быть установлена с образа компакт-диска, USB или через FTP-сервер. Дистрибутив очень часто обновляется и из-за необходимости работы в терминале не подходит для новичков. Интерфейс данной системы представлен на рисунке 7.



Рисунок 7 – Интерфейс операционной системы ArchLinux

Операционной системой для блока обработки данных была выбрана система Windows 10 IoT, так как в рамках данной работы является самой оптимальной: дружелюбный и знакомый интерфейс для целевой аудитории, полная совместимость с приложениями и устройствами Windows, готовый дистрибутив, не требующий доработок.

2.2 Выбор устройства технического зрения

В этом контексте может быть рассмотрена задача технического зрения в целом, и задача распознавания человека и взаимодействия с ним [9]. Однако, дополняя возможности устройств оптического ввода информации, возможно использование сенсоров глубины, представляющих собой инфракрасный проектор, объединенный с монохромной КМОП-матрицей. Это позволяет осуществлять полное 3-х мерное распознавание движений тела, мимики лица и позы тела. [10]

Продукты, совмещающие в себе данные функции, представлены на рынке в небольшом количестве, среди них:

- Microsoft Kinect;
- Microsoft Kinect 2;
- Asus Xtion v1, v2;
- Некоторые модели камер Z-Cam.

2.2.1 Microsoft Kinect

Kinect - бесконтактный сенсорный контроллер, преимущественно предназначен для игровой индустрии, разработанный и представленный для консоли 4 ноября 2010 года в США и 10 ноября в Европе. Поставки версии для Windows были начаты 1 февраля 2012 года. Первоначально был представлен для консоли Xbox 360, и значительно позднее для Xbox One и персональных компьютеров под управлением ОС Windows. Разработан компанией Microsoft. Основанный на добавлении периферийного устройства к игровой приставке Xbox 360, Kinect позволяет пользователю взаимодействовать с ней без помощи контактного игрового контроллера через устные команды, позы тела и показываемые объекты или рисунки. Microsoft Kinect Состоит из двух сенсоров

глубины, цветной видеокамеры и микрофонной решетки. Изображение Microsoft Kinect v1 представлено на рисунке 8 [10].



Рисунок 8 – изображение устройства Microsoft Kinect V1

2.2.2 Asus Xtion

Контроллер ASUS Xtion Pro нельзя назвать полностью собственной разработкой ASUS. За основу этого устройства была взята та же технология от Израильской компании PrimeSense, которая является мировым лидером по работе с датчиками глубины. Разработки данной компании была использована Microsoft в контроллере Kinect. Для регистрации движений пользователя в режиме реального времени в контроллере Xtion Pro используется инфракрасный датчик и технология адаптивного отслеживания глубины. В 2017 году выпущена вторая версия данного устройства Asus Xtion v2. с разрешением глубины 640x480 точек (30 кадров в секунду) и разрешением изображения 2592 x 1944 точек. ASUS Xtion 2 обладает низким энергопотреблением и использует в качестве питания интерфейс USB 3.0. Совместимость с графическим стандартом OpenNI и поддержка операционных систем Windows и Linux открывают широкие возможности для разработчиков. Изображение устройства Asus xtion V2 представлена на рисунке 9.



Рисунок 9 – Изображение устройства Asus Xtion V2

2.2.3. Z-cam

Веб-камеры ZCam, выпущенной в 2000 г. фирмой 3DV Systems. ZCam является разновидностью TOF-камеры, позволяющей получать трёхмерную видео информацию. FPS 60 кадр/сек достигается при съёмке видео с Z CAM S1 в качестве 4k, в разрешении 6k максимальная частота кадров составит 30 кадр/сек., данная модель - уникальное решение для VR-продакшн студий. Камера принадлежит к профессиональному сегменту. Съёмка Z CAM S1 построена на синхронизированной работе 4 линз, управление предполагает только ручной режим контроля записи посредством нажатия соответствующих кнопок (REC/STOP и др.) на камере, беспроводные модули Wi-Fi или Bluetooth отсутствуют. Изображение устройства Z-cam представлено на рисунке 10.



Рисунок 10 – Изображение устройства типа Z-cam

Среди всех описанных аналогов наиболее подходящим устройством для реализации поставленной задачи в рамках магистерской квалификационной работы является Kinect v1 в связи с тем, что является наиболее доступным.

2.3 Выводы по главе

В данной главе была разработана структурная схема модернизированного программно-аппаратного комплекса. На основе анализа доступных источников был сделан выбор подходящего устройства для производства вычислительных операций при работе комплекса. Была выбрана операционная система для данного устройства, исходя из требований к работе был выбран одноплатный компьютер Raspberry Pi 3 model B и система windows 10 IoT. Для полноценного функционирования комплекса необходим орган технического зрения. Был проведен сравнительный анализ устройств, сочетающих в себе необходимые характеристики для полноценного функционирования комплекса. Было принято решение выбрать Kinect v1 в связи с доступностью устройства и сочетанием всех необходимых функций.

3 Разработка комплекса

3.1 Описание программных модулей

Для решения поставленной задачи было принято решение разработать собственный программно-аппаратный комплекс контроля выполнения лечебно-восстановительных упражнений. Графическая модель комплекса представлена на рисунке 11. Комплекс состоит из органа технического зрения (Kinect v1), Модуля первичной обработки, разработанного для подключения к органу технического зрения и непрерывного получения цветного кадра и кадра глубины, а также для дальнейшей передачи последних модулю вторичной обработки. Модуль вторичной обработки разработан для приема и анализа цветного кадра и кадра глубины, а именно для выделения человека в кадре и получения координат суставов скелета человека. Стоит отметить, что данный комплекс так же предусмотрен для модернизации и добавления нескольких блоков вторичной обработки для возможности модернизации программной части и комплекса в целом. Программа редактор разработана для создания и редактирования комплексов упражнений и передачи файлов упражнений программе контроллеру. Программа контроллер разработана для отслеживания правильности выполнения упражнений.



Рисунок 11 – Структурная схема комплекса

3.1.1 Описание модуля первичной обработки

При первичной обработке происходит получение цветного кадра и глубинного изображения из видеопотока, передаваемого Kinect. Основная задача данного модуля – получение видеопотока и передача цветного кадра программному обеспечению. Для решения данной подзадачи был разработан программный продукт на языке C++ с использованием кроссплатформенного Фреймворка QT, по причине того, что Qt позволяет запускать написанное с его помощью программное обеспечение в большинстве современных операционных систем путём простой компиляции программы для каждой системы без изменения исходного кода. При разработке данного модуля была использована библиотека Open NI. Работа данного блока обработки представлена на рисунке 12 [11].

[illegible]

Рисунок 12 – Получение видеопотока с сенсора

Одним из основных классов в данном модуле является класс «Server», выполняющий функцию передачи данных с сенсора абоненту, т.е. самой

программе. Описание некоторых методов класса «Server» представлено в таблице 3.

Таблица 3 – Описание методов класса «Server»

Метод класса	Описание метода
void exchangeData();	Метод осуществляет обмен информацией с клиентом, определяет тип запрашиваемых кадров.
int getTransmittedSize();	Метод получения количества переданных данных.
int getClientNumber();	Метод получения количества клиентов подключенных к серверу
void clearTransmittedSize();	Метод обнуления счетчика переданных данных.
void setColorFrame();	Метод сохранения цветного кадра во временное хранилище.
void setDepthFrame();	Метод сохранения кадра глубины во временное хранилище.
void setRawDepthFrame();	Метод сохранения сырого кадра во временное хранилище.
bool run();	Метод запуска сервера на всех доступных IP-адресах.
void socketDisconnected();	Метод, реагирующий на отключение клиента от сервера.

Класс «Start» предназначен для управления программой и установления соединения с сенсором. Описание некоторых методов класса «Start» Представлено в таблице 4

Таблица 4 – Описание методов класса «Start»

Метод класса	Описание метода
bool detectDevices();	Метод определения подключенных устройств
void showDevices();	Метод определения подключенных устройств
void setColorVideoMode();	Метод выбора цветного режима видеопотока
void setDepthVideoMode();	Метод выбора глубинного режима видеопотока
void setDevice();	Метод выбора устройства
void setPlaybackFile();	Метод создания файла воспроизведения
void setRecordFile();	Метод создания файла записи
void recordActivate()	Метод запуска записи данных
void run();	Метод запуска сервера передачи данных с сенсора
void showTransmittedSize();	Метод транслирования размера передаваемого кадра
void prepareData();	Метод обработки данных

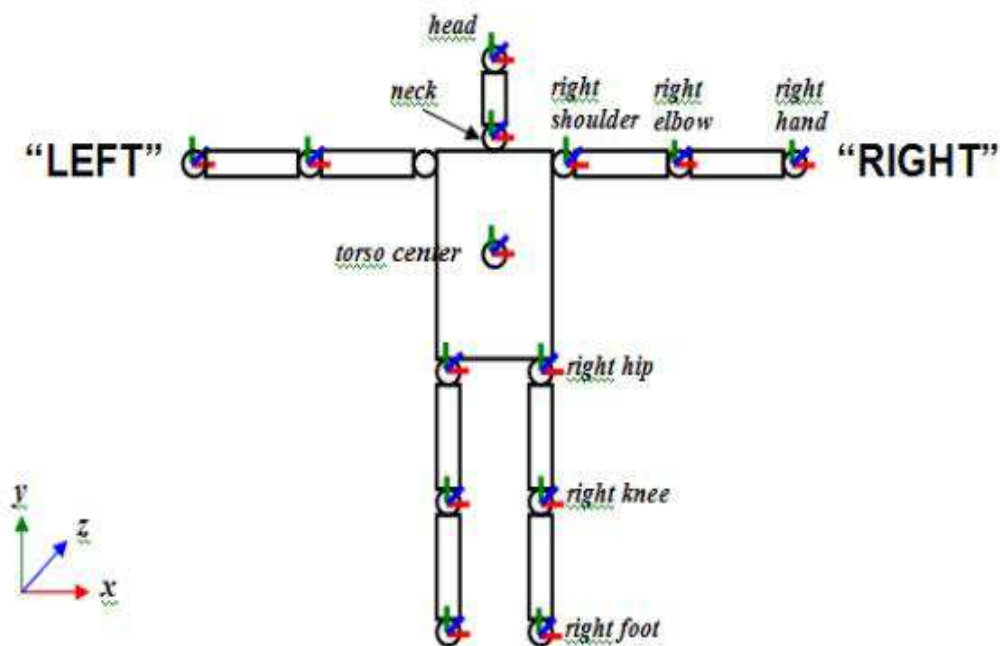
3.1.2 Описание модуля вторичной обработки

При вторичной обработке происходит получение глубины изображения видеопотока, от модуля первичной обработки. Основная задача данного модуля – получение видеопотока и передача координаты каждого сустава модели человека программному обеспечению. Для решения данной подзадачи был создан программный продукт на языке C++ с использованием кроссплатформенного Фреймворка QT, по причине того, что Qt позволяет запускать написанное с его помощью программное обеспечение в большинстве современных операционных систем путём простой компиляции программы для каждой системы без изменения исходного кода. При разработке данного модуля была использована библиотека Nite 2 [12].

Была использована лишь малая часть данной библиотеки, позволяющая решить задачи вторичной обработки, а именно функции:

- 1) `userTrackerFrame.getUsers();` - распознает количество пользователей и возвращает структуры с информацией о них;
- 2) `user.getSkeleton().getJoint((nite::JoinType));` - позволяет получить координаты конечностей(суставов);
- 3) `Joints[].getPosition().x();` - позволяет получить координату X в трехмерном пространстве определенного сустава;
- 4) `Joints[].getPosition().y();` - позволяет получить координату Y в трехмерном пространстве определенного сустава;
- 5) `Joints[].getPosition().z();` - позволяет получить координату Z в трехмерном пространстве определенного сустава.

Результат работы данного модуля представлен на рисунке 13 [12].



NOTE 1: Skeleton's front side is seen in this figure
NOTE 2: Upper arm is twisted such that if elbow is flexed the lower arm will bend forwards towards sensor.

Рисунок 13 – Графическое представление результата работы модуля вторичной обработки

Класс «frameListener» - класс-слушатель для получения кадров цвета и кадров глубины, осуществляет передачу данных от блока первичной обработки. Методы класса «frameListener» и описание основных методов представлено в таблице 5.

Таблица 5 – Описание основных методов класса «frameListener»

Метод класса	Описание метода
void onNewFrame();	Метод получает последний кадр из устройства и передает его серверу для хранения
setColorFrame();	Метод получения цветного кадра от модуля первичной обработки
setDepthFrame();	Метод получения кадра глубины от модуля первичной обработки

Класс «coordinateDetector» разработан для отслеживания пользователя в кадре и передачи координат суставов скелета человека программному обеспечению. Описание основных методов класса «coordinateDetector» представлено в таблице 6.

Таблица 6 – Описание основных методов класса «coordinateDetector»

Метод класса	Описание метода
startTracking();	Метод отслеживания пользователя в кадре
getJoints();	Функция получения координат суставов человека

3.1.3 Описание программы редактора упражнений

Программа редактор написана для создания упражнений, добавления их в список и создания выходного файла. Программная форма состоит из рабочей

области, на которой располагается аватар-скелет, с которым пользователь будет проводить манипуляции чтобы создать упражнение. Программная форма представлена на рисунке 14, которая содержит рабочую область с расположенным в ней скелетом-аватаром, набором суставов, их всего 15, и управляющими элементами.

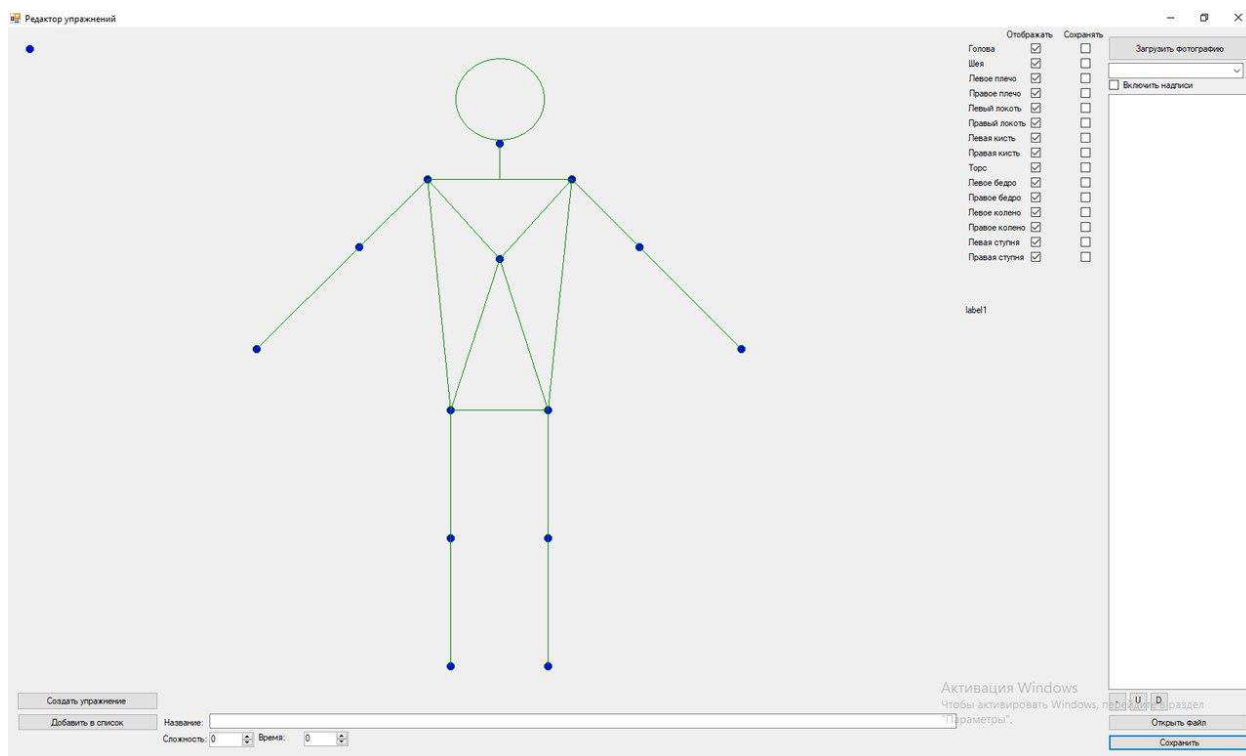


Рисунок 14 - Программная форма приложения редактор

Для удобства можно загрузить фотографию с конечным положением тела при выполнении упражнения, для этого в правом верхнем углу есть кнопка «Загрузить фотографию». Под данной кнопкой расположен выпадающий список с режимами отображения фотографии. Под выпадающим списком расположено поле, где будет расположен список упражнений, созданных в редакторе. Так же в правом верхнем углу представлен Checkbox «Включить надписи». При установлении флажка в окно на рабочей панели все суставы скелета будут подписаны. Под объектом Checkbox расположено окно, в котором будет отображаться список упражнений. Слева от кнопки «Загрузить фотографию»

расположено пятнадцать парных объектов Checkbox с названием «Отображать» и «Сохранять» соответственно. Объекты в столбце «Отображать» необходимы для отображения суставов скелета, пользователь может убрать галочку из объекта Checkbox и соответствующий сустав станет невидимым. Объекты Checkbox «Сохранить» необходимы, когда пользователь создает упражнения, при этом он должен отметить ключевые суставы в данном упражнении, что нужно для написания выходного файла с упражнениями и, впоследствии, для пользовательской программы, отслеживающей выполнение упражнений.

В нижнем левом углу окна находится кнопка «Создать упражнение», при нажатии на которую необходимо установить аватар-скелет в конечное положение при выполнении упражнения. Под предыдущей кнопкой расположена кнопка «Добавить в список», а рядом с ней поля для написания названия упражнения, для задания сложности по десятибалльной шкале и для задания времени в секундах. Стоит отметить, что такие параметры как сложность и время необходимы для программы, отслеживающей выполнение упражнений пользователем. В зависимости от заданной сложности будет корректироваться поле, в которое пользователь должен поместить определенный сустав, чем сложность меньше, тем данное поле больше. И при истечении времени и невозможности выполнения упражнения в заданный срок, и если пользователь не сможет поместить сустав в определенное поле за определенное время, это поле так же будет увеличено, тем самым уменьшая сложность выполнения упражнения.

Кнопки в правом нижнем углу «Открыть файл» и «Сохранить» необходимы для открытия готового файла с комплексом упражнений и сохранения нового комплекса в файл соответственно. Над кнопками, описанными ранее расположены кнопки, необходимые для редактирования списка упражнений в комплексе. Кнопка «-» предназначена для удаления упражнения из списка. Кнопки «U» и «D» необходимы для перемещения упражнения в списке на одну позицию выше и ниже соответственно.

3.1.3.1 Описание выходного файла программы редактор

При создании упражнения создается XML-файл с жёсткой структурой. Шаблон структуры выглядит следующим образом:

```
struct Exercise  
{public int frameWidth;  
public int frameHeight;  
public List<Point> joints;  
int difficult;  
int time;  
public String name;};
```

В поле **Width** и **Height** записываются размеры кадра, на котором было создано упражнение, т.е. ширина и высота соответственно. В поле **joints** описываются координаты каждого сустава, задействованного в упражнении. Каждый сустав, представленный в программной форме, имеет свое наименование в выходном файле и представлен в таблице 7.

Таблица 7 – Наименование суставов

Наименование в программе редактор	Наименование в выходном файле
Голова	Head
Шея	Neck
Левое плечо	leftShould
Правое плечо	rightShould
Левый локоть	leftElbow
Правый локоть	rightElbow
Торс	Torso
Левая кисть	leftHand
Правая кисть	rightHand
Левый таз	leftHip
Правый таз	rightHip

Окончание таблицы 7

Наименование в программе редактор	Наименование в выходном файле
Левое колено	leftKnee
Правое колено	rightKnee
Левая нога	leftFoot
Правая нога	rightFoot

В поле **difficult** описана сложность выполнения упражнений от одного до десяти соответственно. В поле **time** записано время, отведенное на выполнение упражнения в секундах, а в поле **name** записано название упражнения.

Ниже приведен пример выходного файла при создании упражнения «Подъем руки»:

```

<Exercise1> //Начало описания упражнения
  <Width>1264</Width> // Ширина кадра, на котором создано упражнение
  <Height>815</Height> // Высота кадра, на котором создано упражнение
  <JointPositions> // Начало описания координат суставов
    <Torso> //Начало описания координат суставов точки Торс
      <X>710</X> //Координата X
      <Y>150</Y> //Координата Y
      <Z>0</Z> //Координата Z
    </Torso> //Конец описания координат точки Торс
    <RightElbow> //Начало описания координат сустава Правый
локоть
      <X>877</X> //Координата X
      <Y>178</Y> //Координата Y
      <Z>0</Z> //Координата Z
    </RightElbow> //Конец описания координат точки Правый
локоть
    <RightHand> //Начало описания координат точки Правая рука
      <X>1026</X> //Координата X

```

```

<Y>194</Y> //Координата Y
<Z>0</Z> //Координата Z
</RightHand> //Конец описания координат точки Правая рука
</JointPositions> //Конец описания координат суставов
<Difficult>3</Difficult> //Поле Сложность
<Time>30</Time> //Поле Время
<Name>Подъем руки</Name> //Название упражнения
</Exercise1> //Конец описания упражнения

```

3.1.4 Описание программы контроллера

Программа контроллер написана для отслеживания упражнений, выполняемых пациентом, создания отчетного csv-файла для отслеживания и анализа прогресса пациента при выполнении упражнений. Программная форма состоит из рабочей области, на которой происходит трансляция в режиме онлайн, та часть программной формы, где пользователь будет видеть себя. Данная программная форма представлена на рисунке 15 и содержит в себе рабочую область, таймер, список упражнений и кнопку запуска выполнения упражнений.

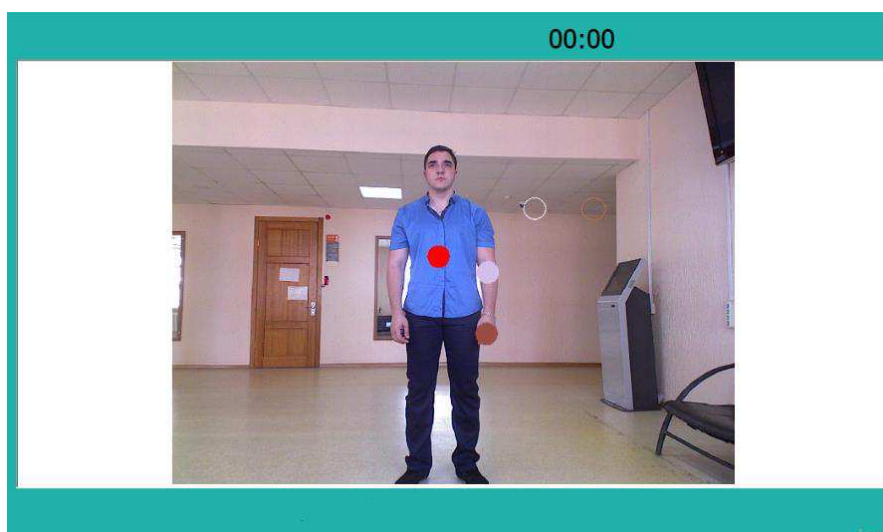


Рисунок 15 – Программная форма программы контроллер

Класс «MainWindow» является основной программы контроллер, некоторые методы класса «MainWindow» представлены в таблице 8

Таблица 8 – Описание основных методов класса «MainWindow»

Метод класса	Описание метода
void setFrame();	Метод получения видеопотока от модуля вторичной обработки.
getExercisesList();	Метод получения списка упражнений для возможности выбора и запуска
on_listWidget_itemSelectionChanged();	Метод выбора упражнения из списка упражнений
getFramePosition();	Метод получения координат идеального положения суставов при выполнении упражнения из файла упражнений
showExercisePosition();	Метод отрисовки идеального положения суставов в момент завершения упражнения
showSceleton();	Метод отрисовки суставов пользователя, необходимых для выполнения упражнения
void showTime();	Метод транслирования времени выполнения упражнения
int detectHits();	Метод отслеживания попадания

3.2 Алгоритм работы программы контроллера

Алгоритм работы программы контроллер и программно-аппаратного комплекса в целом состоит из шагов, представленных далее.

1 Инициализация системы

Инициализация начинается с запуска блока первичной обработки информации, полученной с сенсора по требованию программы. Информация представляет собой непрерывный видеопоток.

2 Запуск блока вторичной обработки

На данном этапе происходит получение информации (подключение к блоку первичной обработки и ожидание пользователя в кадре).

3 Получение координат конечностей суставов

Во время работы блока вторичной обработки происходит непрерывная передача координат точки каждого сустава программа контроллер их принимает (В дальнейшем $X1$, $Y1$, $Z1$).

4 Запуск программы контроллера

После запуска программы происходит ожидание пользователя в кадре.

5 Непрерывное получение видеопотока

Получение видеопотока от блока первичной обработки программой контроллером.

6 Получение координат

Получение координат конечностей пользователя от блока вторичной обработки для дальнейшего анализа попадания суставов в нужные сферы.

7 Открытие файла упражнений

Открытие файла, в котором описаны координаты точки суставов, участвующих в выполнении упражнения в конечной фазе выполнения, созданного в программе редактор.

8 Отрисовка сфер, в которые пользователю необходимо поместить суставы, участвующие в выполнении упражнения

На основании файла упражнений и точек каждого сустава участвующего в выполнении (В дальнейшем X_0, Y_0, Z_0) ,а так же в зависимости от поля «Сложность», на основании которого задается радиус сферы происходит отрисовка сфер, в которые пользователь должен поместить суставы в режиме реального времени по уравнению:

$$(X - X_0)^2 + (Y - Y_0)^2 + (Z - Z_0)^2 = R_1^2 \quad (1)$$

10 Непрерывное отслеживание выполнения упражнения

Для координат каждого сустава должно выполняться условие, представленное в следующем уравнении:

$$(X_0 - X_1)^2 + (Y_0 - Y_1)^2 + (Z_0 - Z_1)^2 \leq R_1^2 \quad (2)$$

3.3 Выводы по главе

Представлена структурная схема программного обеспечения комплекса. Разработан программный модуль для подключения органа технического зрения и получения необходимых данных от сенсора и передачи их. Представлен программный модуль для отслеживания пользователя в кадре и получения координат суставов скелета пользователя. Разработана программа для создания и редактирования упражнений. Разработана программа для отслеживания правильности выполнения упражнений. Ведется работа по тестированию программных модулей, доработка, а также создание опытной модели.

4 Инструкция для пользователя

Для корректного функционирования комплекса необходима правильная настройка и правильный порядок действий. Комплекс состоит из сенсора Kinect v1 и usb провод для подключения к блоку обработки данных, блока обработки данных, представленный одноплатным компьютером Raspberry Pi 3 model B, hdmi кабель для подключения монитора или телевизора к блоку обработки данных, устройство управления комплексом, этим устройством является компьютерная мышь, usb накопитель с файлом упражнений. В данной главе предоставлена инструкции для пользователей для полноценного корректного функционирования комплекса.

4.1 Инструкция для создания файла упражнений доктором

Работа комплекса начинается с создания индивидуального файла упражнений для каждого пациента, инструкция по корректному созданию файла представлена далее:

1) Для создания файла упражнений с корректными координатами необходимо сфотографировать пациента так, чтобы суставы, участвующие в упражнении, находились в приблизительно идеальном положении, в котором должны быть в момент завершения упражнения, при возникновении возможных трудностей необходимо помочь пациенту;

2) После необходимо загрузить фотографию на компьютер доктора, на котором запущена программа редактор;

3) При успешной загрузке фотографии необходимо запустить программу редактор;

4) Необходимо загрузить фотографию в программу редактор, для этого есть кнопка в правом верхнем углу программной формы «Загрузить фотографию»;

5) В выпадающем списке, располагающемся под кнопкой загрузить необходимо выбрать режим отображения фотографии, чтобы было корректное отображение фото на программной форме;

6) Нажать кнопку «Создать упражнение» в левом нижнем углу программной формы;

7) Курсор мыши навести на центральную точку туловища и переместить скелет-аватар так, чтобы центральная точка туловища скелета-аватара совпала с центральной точкой туловища человека на фотографии;

8) Совместить суставы скелета и человека на фотографии, которые задействованы в упражнении; Результат изменения режима отображения фото и сопоставления суставов представлен на рисунке 16;

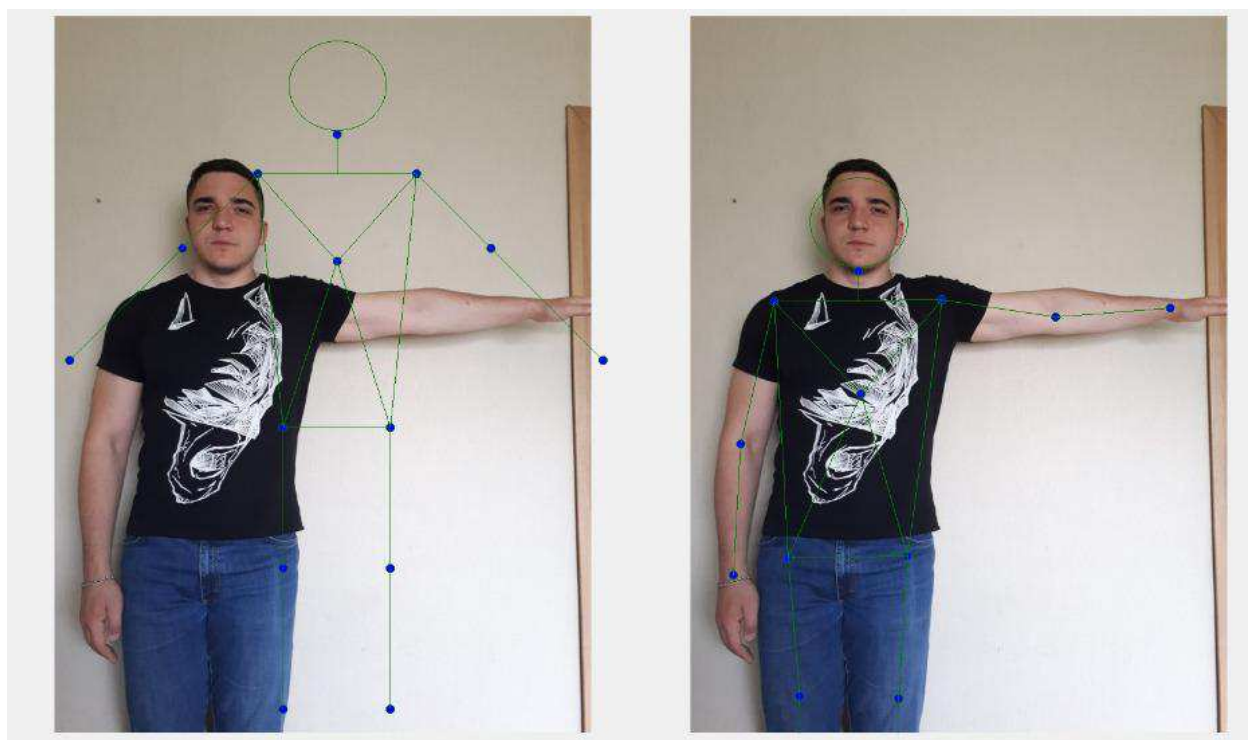


Рисунок 16 – Результат изменения режима отображения фото и сопоставления суставов

9) В столбце «Сохранить», расположенном в правом верхнем углу программной формы выбрать суставы, задействованные при выполнении данного упражнения и отметить их флажком

10) Задать сложность в диапазоне от одного до десяти в окне «Сложность», расположенном в левом нижнем углу программной формы;

11) Задать время выполнения упражнения в секундах в окне «Время», расположенном в левом нижнем углу программной формы;

12) Указать название упражнения в окне «Название», расположенном в левом нижнем углу;

13) Нажать кнопку «Добавить в список», расположенную в левом нижнем углу программной формы после чего упражнение появится в списке упражнений в правой части программной формы;

14) Для редактирования положения упражнения в списке необходимо воспользоваться кнопками, расположенными в поле под списком упражнений;

15) Для создания еще одного упражнения необходимо вернуться к пункту 1, представленной инструкции;

16) Для сохранения комплекса упражнений в отдельный файл необходимо нажать кнопку «Сохранить», расположенной в правом нижнем углу программной формы;

17) После создания комплекса упражнений необходимо перенести файл с упражнениями на usb накопитель и передать пациенту.

4.2 Инструкция по сбору комплекса и запуску системы

Для запуска программно-аппаратного комплекса необходимо для начала собрать аппаратную часть, инструкция по сбору аппаратной части комплекса представлена ниже:

1) Для подключения монитора или телевизора необходимо подключить кабель hdmi к монитору/телевизору и в разъем блока обработки данных;

2) Подключить сенсор Kinect v1 кабелем usb к блоку обработки данных;

- 3) Подключить компьютерную мышь в разъем usb блока обработки данных;
- 4) Подключить usb накопитель с файлом упражнений к блоку обработки данных;
- 5) Подключить кабель micro usb в разъем питания блока обработки данных, при подключении питания запустится система;

Корректное подключение необходимых устройств к блоку обработки данных представлено на рисунке 16.

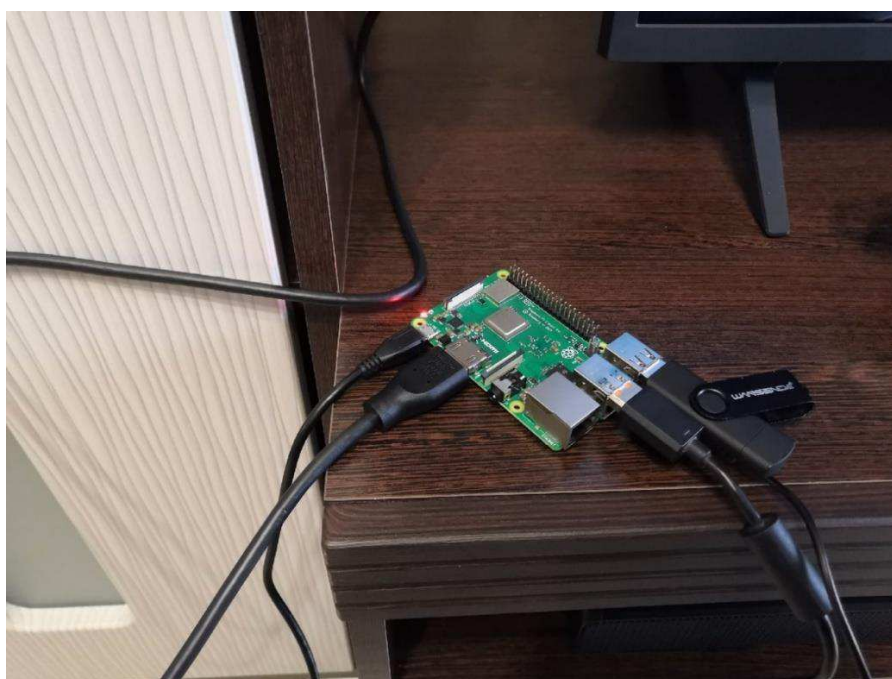


Рисунок 16 – Корректное подключение устройств к блоку обработки данных

4.3 Инструкция по работе с модулями обработки программой редактор

Запуск модулей первичной обработки, вторичной обработки и программы контроллер происходит автоматически при запуске системы, однако если этого не произошло, необходимо выполнить ряд действий, представленных далее:

- 1) При запуске системы необходимо запустить файл «Primary.exe», расположенном на рабочем столе;
- 2) После запуска файла «Primary» откроется командная строка со справочной информацией, необходимо ввести команду «-l» и клавишу «enter»;
- 3) Далее необходимо выбрать сенсор Kinect и нажать «enter», модуль первичной обработки будет запущен и сенсор начнет передавать видеопоток блоку обработки данных;
- 4) Следующим этапом является запуск модуля вторичной обработки, необходимо запустить файл на рабочем столе «Secondary.exe»;
- 5) Пользователю (пациенту) необходимо начать двигаться перед сенсором, чтобы программа распознала пользователя в кадре;
- 6) При запущенных модулях первичной и вторичной обработки необходимо запустить программу контроллер, для этого нужно запустить файл «Controller.exe», расположенном на рабочем столе;
- 7) После запуска программы контроллер необходимо нажать кнопку «Начать» и начать выполнять комплекс упражнений. После выполнения комплекса упражнений созданных доктором столько раз, сколько рекомендовано доктором, необходимо передать usb накопитель доктору для дальнейшего анализа файла результата выполнения упражнений «Results.csv», который создается при работе программы контроллер. результат работы комплекса представлен на рисунке 17.

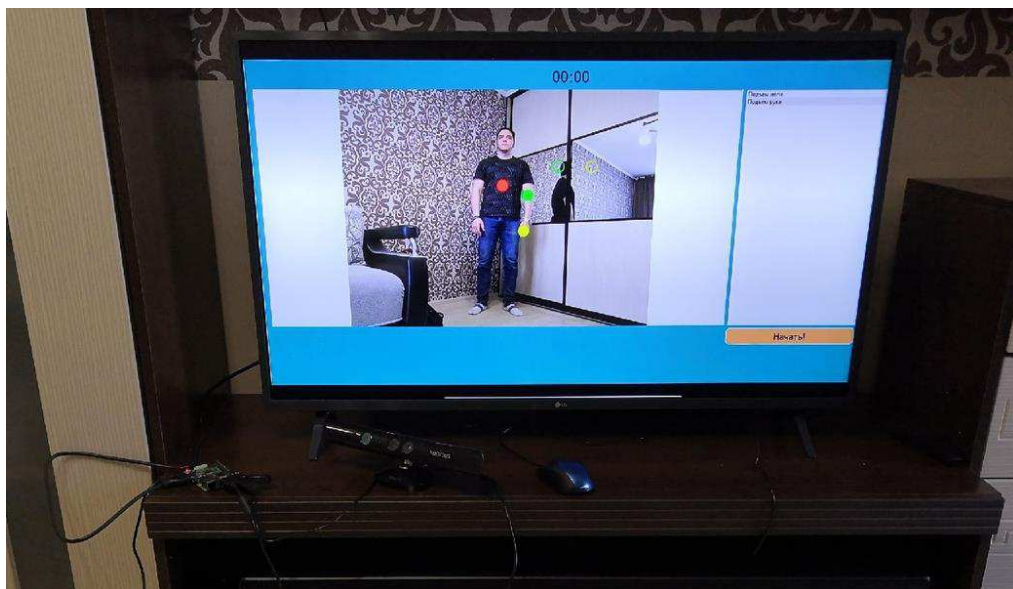


Рисунок 17 – Результат работы комплекса

4.4 Выводы по главе

В данной главе представлена инструкция для пользователей на каждом этапе разработки комплекса упражнений. Была разработана инструкция для доктора и корректного создания комплекса упражнений для конкретного пациента, описан процесс сбора комплекса, подключения устройств. Создана инструкция для пользователя по запуску системы и программных модулей, необходимых для корректной работы комплекса, а также по запуску программы редактор и корректного выполнения упражнений.

ЗАКЛЮЧЕНИЕ

В рамках магистерской диссертации был разработан метод контроля выполнения лечебно-восстановительных упражнений. Произведен анализ существующих систем, выполняющих похожие функции. Выдвинуты требования к собственному, разрабатываемому комплексу. Разработана структурная схема комплекса, выбрана аппаратная часть, выбрано устройство технического зрения, выбрана операционная система комплекса. Написана и опубликована статья «Программно-аппаратный комплекс контроля выполнения лечебно-восстановительных упражнений» в V международной научной конференции «Приоритетные направления инновационной деятельности в промышленности» Написана и опубликована статья «Интеллектуальная система управления реабилитационным манипулятором» в международной научно-практической конференции «Science and education: problem and innovations». Разработан модуль первичной обработки, модуль вторичной обработки. Реализована программа редактор для создания упражнений доктором и программа контроллер для отслеживания выполнения упражнений. Разработан программно-аппаратный комплекс контроля выполнения лечебно-восстановительных упражнений. Написана пошаговая инструкция пользователя для работы с комплексом. Недостатком является недоработка инструмента для анализа выходного файла программы редактор доктором. На данном этапе доктор анализирует выходные файлы при помощи Microsoft Excel.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Буковский, В. А. Физиологические основы здоровья и долголетия : учебное пособие / В.А. Буковский. – Москва : Академия, 2014. – 213 с.
2. В.В. Ковальчук / Современные аспекты реабилитации больных, перенесших инсульт / Ковальчук В.В., Богатырева М.Д, Минуллин Т.И. // Журнал неврологии и психиатрии им. С.С. Корсакова – Москва, 2014, с. 101-105.
3. Reflexion Health Kinect CS : Описательная документация работы продукта Reflection health, Reflection Health Inn. – San Diego, California, 2013. – 4с.
4. Jintronix [Электронный ресурс] : официальный сайт проекта Jintronix. - Швейцария, [2016] – Режим доступа: <http://www.jintronix.com/>.
5. Амперка [Электронный ресурс] : Arduino или Raspberry Pi: как выбрать контроллер для проекта. – Режим доступа: <https://amperka.ru/page/development-board-guide>
6. ТЕРРА Электроника [Электронный ресурс] : Обзор четырех популярных одноплатных компьютеров. – Режим доступа: <https://krs.terraelectronica.ru/news/5826>
7. LOSST [Электронный ресурс] : Обзор четырех популярных одноплатных компьютеров. – Режим доступа: <https://losst.ru/luchshie-operatsionnye-sistemy-raspberry-pi-3>
8. Raspberrypi.org [Электронный ресурс] : Raspberry Pi 3 Model B. – Режим доступа: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
9. Масабаев, Е. С. Применение IT-технологий в реабилитации постинсультных больных (литературный обзор) / Е.С. Масабаев, Ф.Ж. Шокутбаева, А.М. Кабдешев // Молодой ученый. — 2016. — №30. — С. 111-115.
10. Розалиев В.Л. / АВТОМАТИЗАЦИЯ РАСПОЗНАВАНИЯ КИСТЕЙ РУК ЧЕЛОВЕКА С ПОМОЩЬЮ KINECT ДЛЯ ПЕРЕВОДА ЖЕСТОВОГО ЯЗЫКА/ Розалиев В.Л. // ИЗВЕСТИЯ ВОЛГОГРАДСКОГО

ГОСУДАРСТВЕННОГО ТЕХНИЧЕСКОГО УНИВЕРСИТЕТА - Волгоград,
2015.

11. OpenNI – Prime Sense [Электронный ресурс] : OpenNI. – Режим доступа:
<https://en.wikipedia.org/wiki/OpenNI>.

12. Prime sense [Электронный ресурс] : Prime Sense NITE Algorithms 1.5. –
Режим доступа: <http://www.primesense.com/>

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий


институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

 О. В. Непомнящий

подпись инициалы, фамилия

« ____ » ____ 2020 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Программно-аппаратный комплекс контроля выполнения лечебно-
восстановительных упражнений

Тема

09.04.01 «Информатика и вычислительная техника»

код и наименование направления

09.04.01.06 «Микропроцессорные системы»

код и наименование магистерской программы

Научный
руководитель


подпись, дата


профессор, зав кафедры, к.т.н

должность, ученая степень

О.В. Непомнящий

инициалы, фамилия

Выпускник

 26.06.2020
подпись, дата

А.В. Осипов

инициалы, фамилия

Рецензент


подпись, дата

к.т.н., доцент

должность

Э.Ф. Цокур

инициалы, фамилия

Нормоконтролер

подпись, дата

 О.В. Непомнящий

инициалы, фамилия

Красноярск 2020