

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

РАЗРАБОТКА ПОДСИСТЕМЫ УПРАВЛЕНИЯ КАМЕРОЙ
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 – Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Тяпаева Рината Олеговича

Научный руководитель
к. т. н., доцент _____ Д. Ю. Петров
Заведующий кафедрой
к. ф.-м. н., доцент _____ А. С. Иванов

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Описание работы	4
1.1 Язык UML	4
1.2 Rational Rhapsody Developer	4
1.3 Платформа Arduino	5
1.4 Библиотека OpenCV	5
1.5 Фреймворк Flask	6
2 Создание проекта	7
2.1 Техническое задание	7
2.2 Разработка модели в среде IBM Rational Rhapsody	7
2.2.1 Диаграмма вариантов использования	7
2.2.2 Структурная диаграмма	8
2.2.3 Диаграмма последовательности	9
2.2.4 Диаграмма деятельности	10
2.2.5 Диаграмма состояний	11
2.2.6 Диаграмма панели	12
2.3 Создание аппаратного прототипа	12
2.4 Разработка веб-приложения	13
2.5 Запуск приложения	13
ЗАКЛЮЧЕНИЕ	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	16

ВВЕДЕНИЕ

Интернет вещей [1] представляет собой систему взаимосвязанных вычислительных устройств, механических и цифровых машин, объектов, которые снабжены уникальными идентификаторами и способностью обмениваться данными по сети.

Интернет вещей помогает людям жить и работать умнее, а также получить полный контроль над своей жизнью. В дополнение к предложению интеллектуальных устройств для автоматизации домов, интернет вещей имеет важное значение для бизнеса. Интернет вещей предоставляет компаниям возможность наблюдения за работой их системы в режиме реального времени, обеспечивая мониторинг многих вещей: от производительности машин до цепочки поставок и логистических операций.

Таким образом, интернет вещей является одной из наиболее важных технологий повседневной жизни, и она будет продолжать набирать обороты, поскольку все больше компаний осознают потенциал подключенных устройств для поддержания их конкурентоспособности.

В ходе работы была поставлена задача создать проект автоматизированной подсистемы управления камерой, состоящий из:

- абстрактной модели подсистемы, реализованной в среде разработки IBM Rational Rhapsody
- физического прототипа на базе микроконтроллера Arduino
- веб-приложения для удаленного управления прототипом на языке Python

Бакалаврская работа состоит из введения, двух глав, заключения, списка использованных источников и одного приложения. Объем работы составляет 50 страниц. Список литературы включает 26 наименований.

В первой главе приводятся сведения о технологиях и средствах, используемых в дальнейших разделах работы: среде разработки Rational Rhapsody Developer, платформе Arduino, библиотеке OpenCV и веб-фреймворке Flask.

Вторая глава детально описывает разработку проекта: определение требований к системе, её объектов, их взаимодействия и поведения, описания функционала системы, затем сборку прототипа и разработку веб-приложения для управления прототипом.

1 Описание работы

1.1 Язык UML

UML [2] – это унифицированный язык моделирования, основным инструментом которого является набор диаграмм для разработки программного обеспечения с целью определения, визуализации, построения и документирования особенностей программных систем, а так же для бизнес-моделирования. UML – важнейшая часть разработки объектно-ориентированного программного обеспечения. Данный язык использует в основном графические нотации для описания дизайнера или программного проекта. Использование UML [3] помогает упростить разработку потенциального дизайна и архитектуры проектируемой системы. Использование моделей позволяет разработчику работать на более высоком уровне абстракции. Модель может скрыть некоторые детали, отображая общую картину или фокусируясь на различных аспектах прототипа. В UML можно уменьшить масштаб рассмотрения от детального представления приложения в среду, в которой оно выполняется, визуализировать взаимодействие с другими приложениями или еще более подробно рассмотреть модель. Кроме того, можно сфокусироваться на различных аспектах приложения, таких как автоматизирующийся бизнес-процесс или представление бизнес-правил. Новая возможность группировать элементы модели, добавленные в UML, напрямую поддерживает эту концепцию. Когда речь заходит о UML, то первое, о чем вспоминают люди – это классы и объекты, но возможности этого языка не ограничиваются только этим. UML обладает гораздо более широкими и мощными возможностями. UML также включает описание примитивных элементов поведения – действий и деятельностей, а также средства для спецификации допустимой последовательности их выполнения с использованием диаграмм деятельности и диаграмм состояний. UML предоставляет средства для описания примеров взаимодействия, преимущественно с использованием диаграмм последовательности. И, наконец, варианты использования позволяют сгруппировать требования в удобные для использования, внутренне согласованные функциональные модули.

1.2 Rational Rhapsody Developer

Rational Rhapsody Developer [4–6] – это совместная среда разработки для системных инженеров и разработчиков программного обеспечения, пред-

назначенная для создания, тестирования и документирования систем реального времени и встроенных систем и их программного обеспечения. Данная среда помогает сотрудничать различным командам, чтобы анализировать требования, оптимизировать проектные решения и проверять функциональность на раннем этапе жизненного цикла разработки, выполнять обзоры дизайна и автоматизировать доставку инновационных высококачественных продуктов. Линейка продуктов интегрируется с общим жизненным циклом разработки продукта — от спецификации и разработки до тестирования и доставки.

1.3 Платформа Arduino

Arduino [7,8] — это электронная платформа с открытым исходным кодом, основанная на простом в использовании аппаратном и программном обеспечении. Платы Arduino могут считывать входные данные — свет на сенсоре, палец на кнопке или сообщение в соцсетях — и превращать его в выходной сигнал — активировать мотор, включать светодиод и публиковать что-то в Интернете. Вы можете установить поведение системы, отправив набор инструкций микроконтроллеру на плате. Для этого вы используете язык программирования Arduino (на основе Wiring) и программное обеспечение Arduino (IDE) на основе Processing.

1.4 Библиотека OpenCV

OpenCV (Open Source Computer Vision Library) [9] — это библиотека программного обеспечения для компьютерного зрения и машинного обучения с открытым исходным кодом. OpenCV был создан для обеспечения общей инфраструктуры для приложений компьютерного зрения и для ускорения использования машинного восприятия в коммерческих продуктах. Являясь BSD-лицензированным продуктом, OpenCV позволяет предприятиям легко использовать и модифицировать код.

Наряду с известными компаниями, такими как Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, которые используют библиотеку, существует много стартапов, таких как Applied Minds, VideoSurf и Zeitera, которые широко используют OpenCV. Развёрнутые области применения OpenCV охватывают диапазон: от шивания изображений улиц до обнаружения вторжений в видеонаблюдение в Израиле, наблюдения за шахтным оборудованием в Китае, помощи роботам в навигации и подборе объектов в Willow Garage, обна-

ружения аварий утопления в плавательных бассейнах в Европе, выполнения интерактивного искусства в Испания и Нью-Йорк проверяют взлетно-посадочные полосы на наличие мусора в Турции, проверяют этикетки на продуктах на заводах по всему миру на предмет быстрого обнаружения лиц в Японии.

1.5 Фреймворк Flask

Flask [10] — это фреймворк для создания веб приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2. Flask является микрофреймворком, но это не означает, что все веб-приложение должно уместиться в один файл Python (хотя, безусловно, может), и конечно же, не означает, что Flask не хватает функциональности. «Микро» означает, что Flask стремится сделать ядро простым и при этом расширяемым.

По умолчанию Flask не включает уровень абстракции базы данных, проверку формы и другие опции, для которых существуют разные библиотеки. Вместо этого Flask поддерживает расширения для добавления различных функций в приложение. Многочисленные расширения обеспечивают интеграцию базы данных, проверку формы, обработку загрузки, различные технологии аутентификации и многое другое. Данный микрофреймворк прекрасно подходит для широкого спектра веб-приложений.

Flask использует преимущества уже проделанной работы в Werkzeug по реализации интерфейса WSGI (что иногда может быть сложной задачей). Благодаря последним разработкам в инфраструктуре пакетов Python, пакеты с зависимостями больше не являются проблемой, и для того, чтобы не иметь библиотек, которые зависят от других, остается достаточно мало причин.

2 Создание проекта

2.1 Техническое задание

Было дано техническое задание, заключающееся в разработке автоматизированной подсистемы управления камерой. Требования к подсистеме:

- при запуске системы изображение с камеры транслируется на сервер
- пользователь может изменять положение камеры: вращать камеру по горизонтали и вертикали.
- пользователь может перевести камеру в автономный режим, в котором камера будет распознавать двигающиеся объекты и наводиться в их сторону

Проектная работа должна состоять из программы с визуализацией процессов в среде разработки IBM Rational Rhapsody, физического прототипа и веб-приложения для удалённого управления прототипом на языке Python.

2.2 Разработка модели в среде IBM Rational Rhapsody

При определении инструментов разработки была возможность выбора Matlab Simulink для описания требований и поведения подсистемы. После сравнительного анализа IBM Rational Rhapsody и Matlab Simulink выбор был сделан в пользу первой среды разработки, так как более богатый набор возможностей этой среды позволяет вести разработку с нуля и полностью, начиная от определения требований и заканчивая тестированием, в то время, как средства Simulink позволяют лишь провести имитационное моделирование.

Разработка абстрактной модели подсистемы с реализацией процесса MDD заключается в построении следующих UML-диаграмм:

- диаграмма вариантов использования
- структурная диаграмма
- диаграмма последовательности
- диаграмма деятельности
- диаграмма состояний
- панельная диаграмма

2.2.1 Диаграмма вариантов использования

Диаграмма вариантов использования описывает взаимодействие объектов внутри системы (рис. 1). Изображение с камеры транслируется на сервер (Streaming).

Пользователь может осуществлять управления камерой (Camera control), которое включает в себя изменение положения камеры (Camera rotation, Camera depression and elevation), осуществляемое приводами (Horizontal drive, Vertical drive) под управлением контроллера Arduino (Arduino controller). Также пользователь может перевести систему в автономный режим, в котором будет происходить распознавание двигающихся объектов (Morion recognition) и наведение на них (Motion lock).

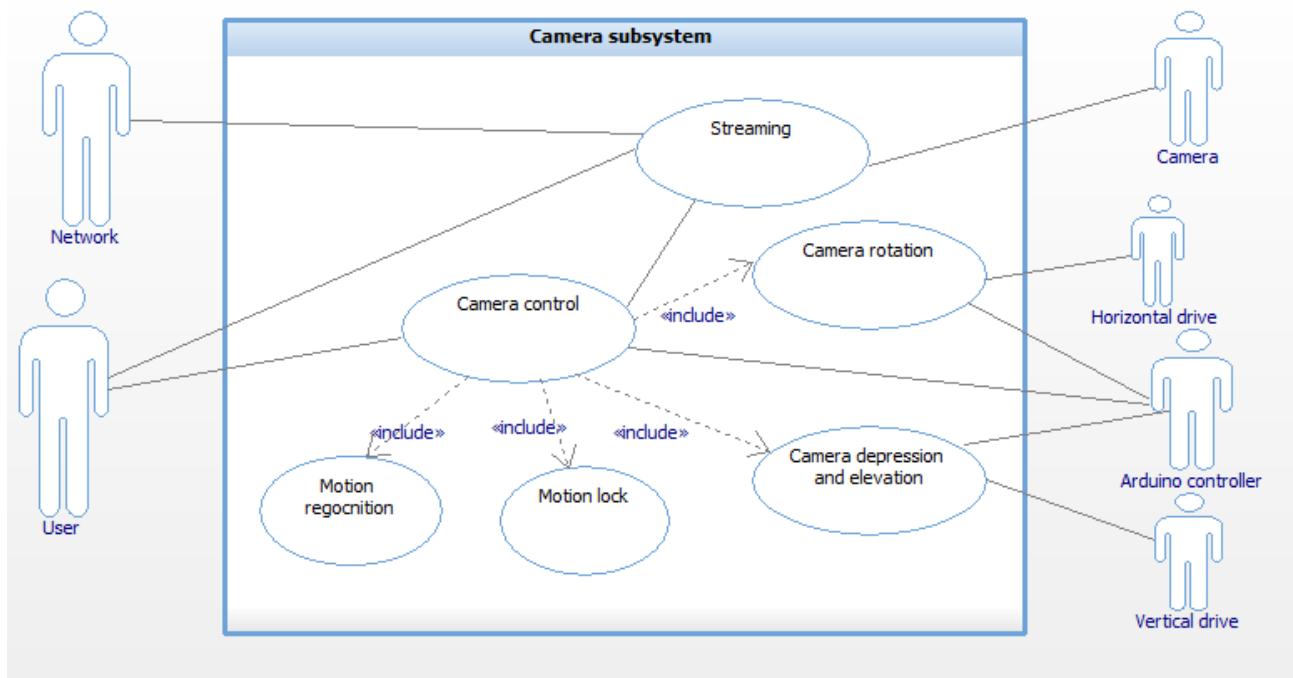


Рисунок 1 – Диаграмма вариантов использования

2.2.2 Структурная диаграмма

Данная диаграмма описывает артефакты системы как взаимосвязанные объекты классов с атрибутами и функциями (рис. 2). На диаграмме приведены следующие объекты:

- устройство пользователя для удаленного управления (UserDevice)
- сервер (Server) с программным обеспечением (CameraControlProgramm)
- микроконтроллер Arduino (ArduinoController)
- вертикальный (VerticalDrive) и горизонтальный (HorizontalDrive) приводы
- камера (CameraModule)

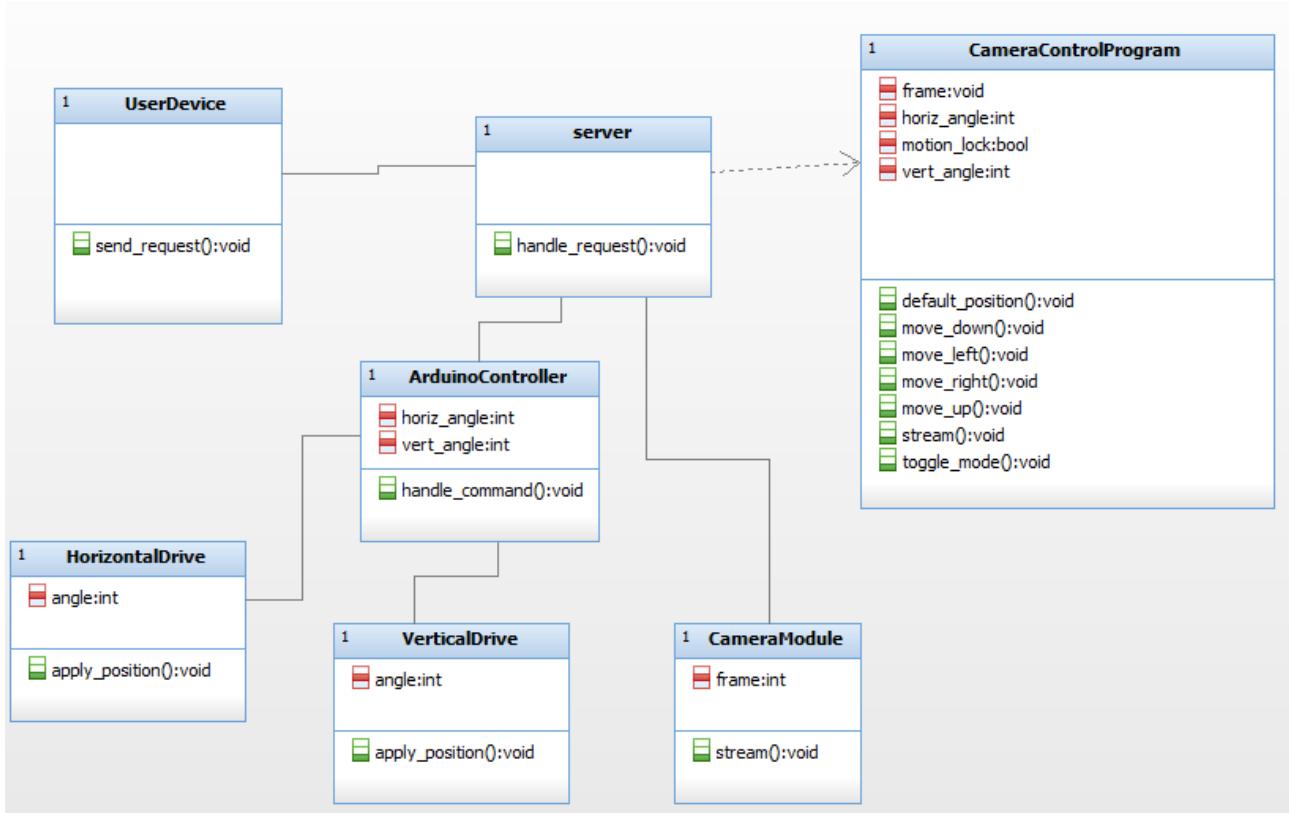


Рисунок 2 – Структурная диаграмма

2.2.3 Диаграмма последовательности

Диаграмма последовательности описывает жизненный цикл объектов системы и их взаимодействие в определенных условиях (рис. 3).

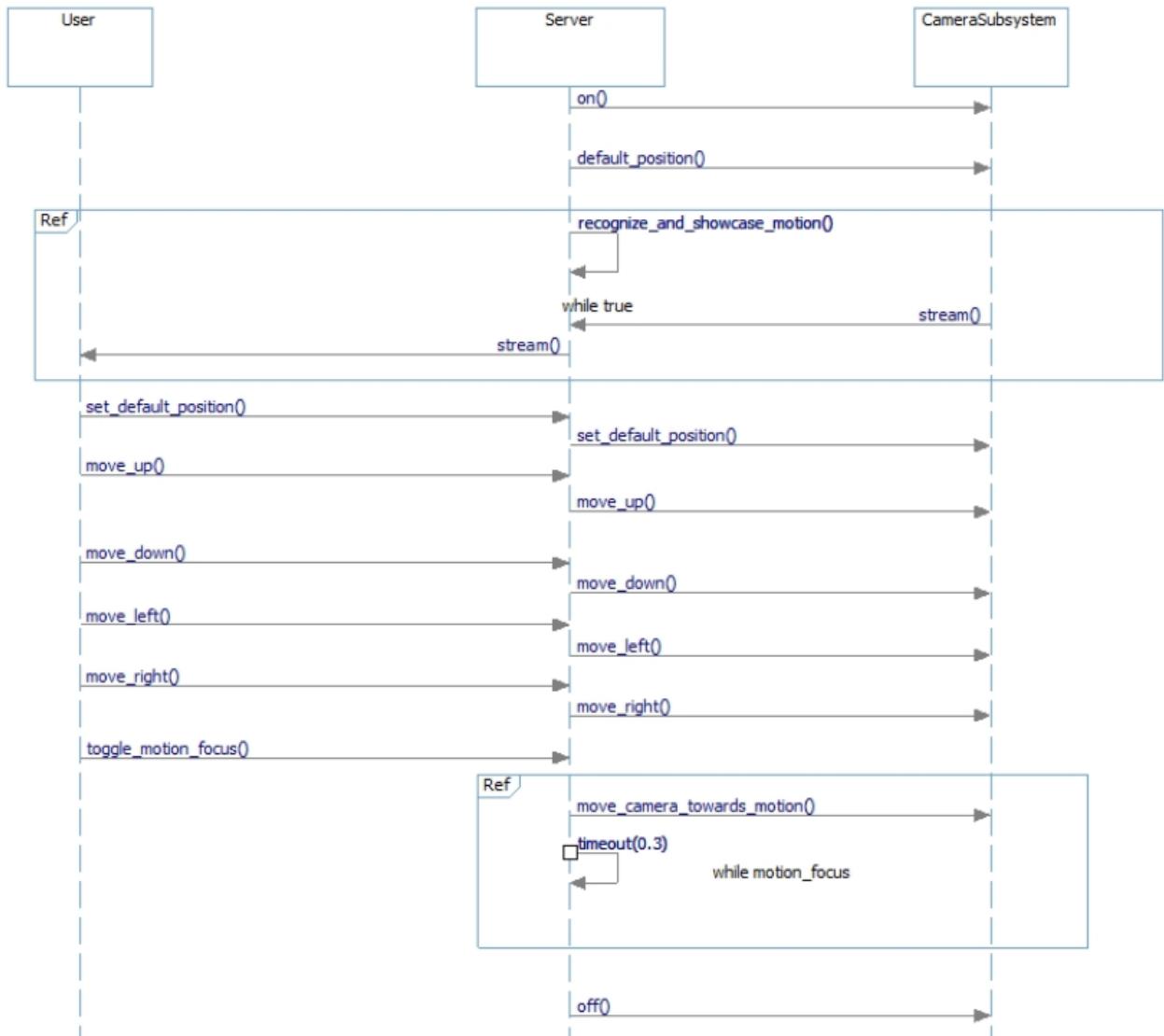


Рисунок 3 – Диаграмма последовательности

2.2.4 Диаграмма деятельности

Данная диаграмма описывает последовательность действий системы (рис. 4).

При инициализации камера переводится в стандартное положение. Затем пользователь может давать команды движения, перевода в автономный режим, в котором будет происходить наведение на движущиеся объекты, или перевода в ручной режим.

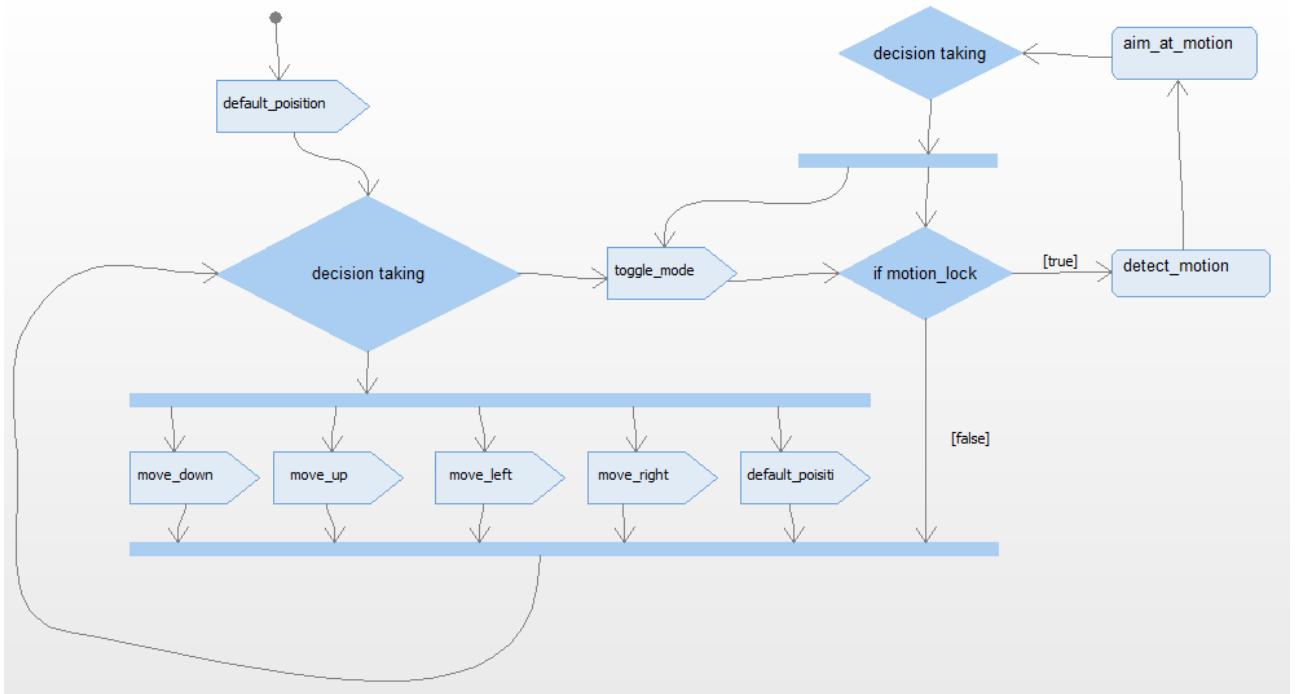


Рисунок 4 – Диаграмма деятельности

2.2.5 Диаграмма состояний

Диаграмма состояний описывает систему как конечный автомат (рис. state).

Система инициализируется в режиме ручного управления (`ManualMode`), в котором могут подаваться команды изменения положения или перевода в автономный режим (`MotionLockMode`), в котором происходит распознавание движения. По команде пользователя система может быть переведена в режим ручного управления соответственно.

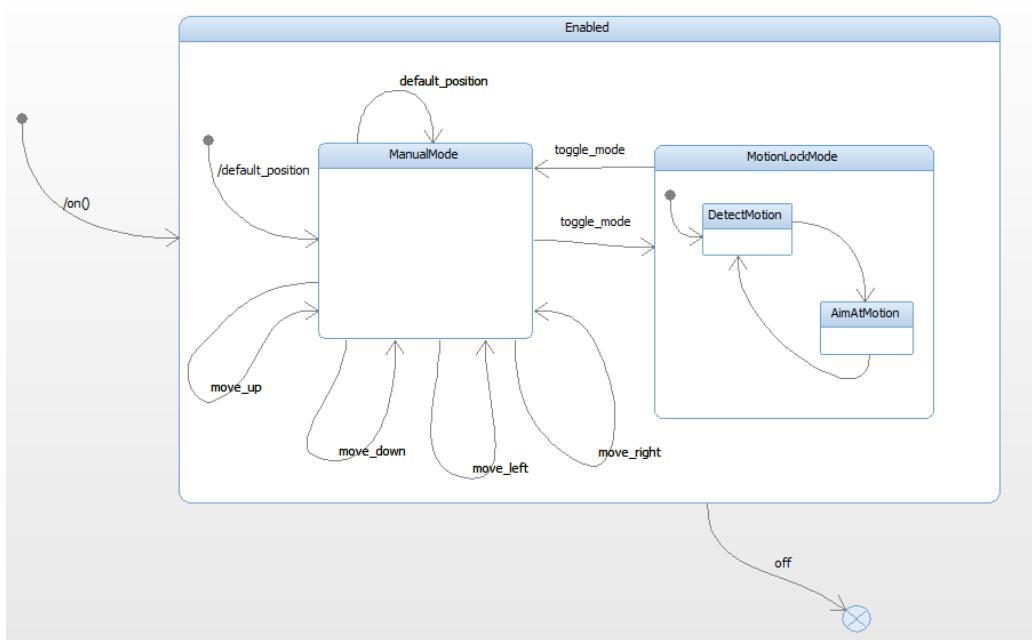


Рисунок 5 – Диаграмма состояний

2.2.6 Диаграмма панели

В данной диаграмме представлена форма пользовательского интерфейса для управления системой (рис. 6). На ней представлены кнопки поворота и наклона, сброса положения камеры и изменения режима, а также поля для отображения положения камеры и текущего режима.

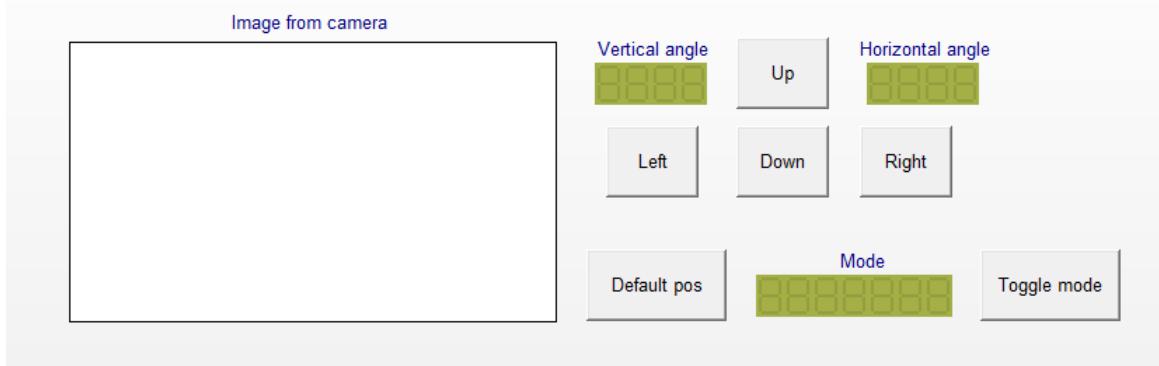


Рисунок 6 – Диаграмма панели

2.3 Создание аппаратного прототипа

Для создания прототипа было использовано следующее оборудование:

- Микроконтроллер Arduino Uno Rev3
- Сервопривод SG-90 x2
- Камера Ritmix RVC-025M
- Соединительные провода

Камера была закреплена на горизонтальном сервоприводе, который в свою очередь был закреплен на вертикальном сервоприводе. Приводы были подключены к микроконтроллеру Arduino по 3-ему и 5-ому цифровым портам. Микроконтроллер был подключен к компьютеру, выполняющему роль сервера.

2.4 Разработка веб-приложения

Веб-приложение для управления прототипом создавалось на языке Python с помощь фреймворка Flask. Были реализованы функции для трансляции изображения на страницу, обработки нажатия кнопок и вычисления углов поворота приводов для наведения в область с распознанным движением.

Для управления функциями прототипа с помощью контроллера Arduino была использована библиотека Pyfirmata языка Python, с помощью которой были реализованы функции изменения положения приводов и сброса позиций в положения по умолчанию.

Обработка изображения и распознавание движения были реализованы с помощью средств библиотеки OpenCV. Движение распознавалось как абсолютная разность текущего кадра и бэкграунд-модели на основе предыдущих кадров.

2.5 Запуск приложения

На веб-страницу (рис. 7) выводится изображение с камеры, текущее положение приводов (Vertical angle, Horizontal angle) и режим работы (Manual mode по умолчанию), также присутствуют кнопки поворота (Left, Right) и наклона (Up, Down), перевода приводов в изначальное положение (Default) и переключения режим работы (Toggle mode).

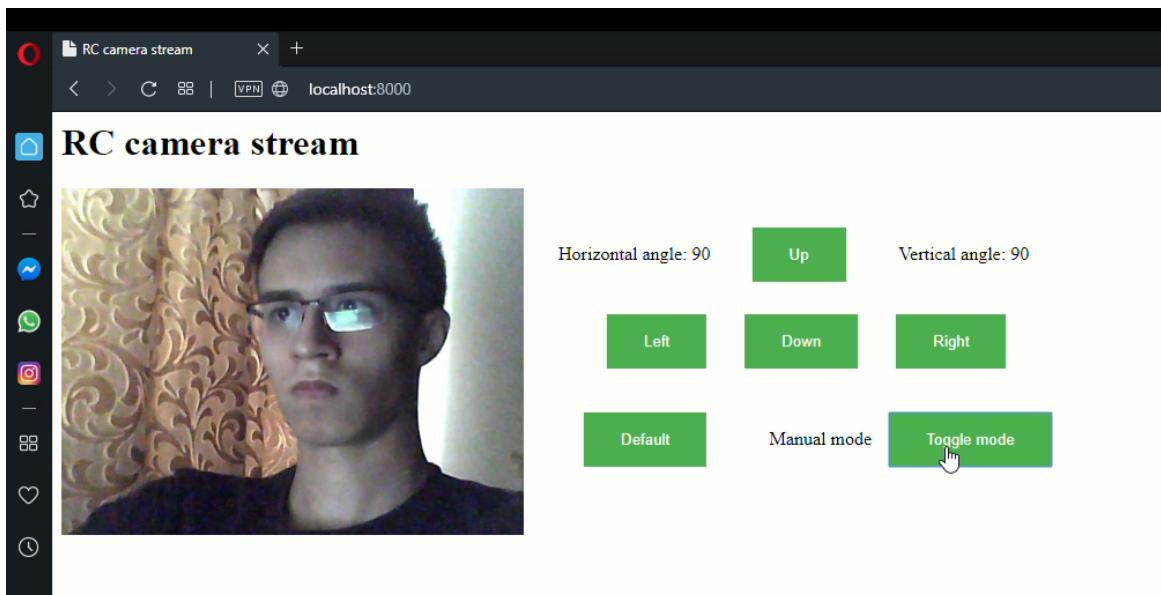


Рисунок 7 – Веб-страница приложения

Также систему можно перевести в автономный режим, в котором происходит распознавание движения: зона, в которой было распознано движение, выделяется прямоугольником, затем происходит наведение в зону, где было распознано движение.

ЗАКЛЮЧЕНИЕ

В данной дипломной работе был рассмотрен процесс создания автоматизированной подсистемы, состоящий из проектирования абстрактной модели, сборки прототипа и разработки веб-приложения для управления прототипом.

В среде разработки Rational Rhapsody была создана абстрактная модель системы, были определены её компоненты и их взаимодействие, а также состояния и поведение системы в этих состояниях.

На базе микроконтроллера Arduino был собран физический прототип подсистемы.

На языке Python с использованием фреймворка Flask было разработано веб-приложение для управления прототипом.

Данная разработка может быть взята за основу для создания системы видеонаблюдения и автоматического контроля доступа, интегрируемой в проект «умный дом».

При выполнении данного технического задания были получены навыки разработки встраиваемых систем, работы с электронными компонентами, в частности микроконтроллером Arduino и его программированием, а также разработки REST API приложения на языке Python.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Internet of things (IoT) [Электронный ресурс]. — URL: <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT> (Дата обращения 19.04.2020). Загл. с экрана. Яз. англ.
- 2 Introduction to Unified Modelling Language [Электронный ресурс]. — URL: <http://www.uml.org/what-is-uml.htm> (Дата обращения 19.04.2020). Загл. с экрана. Яз. англ.
- 3 Douglas, B. Real-time UML Workshop for Embedded Systems / B. Douglas. — Oxford: Newnes, 2007.
- 4 IBM Engineering Systems Design Rhapsody [Электронный ресурс]. — URL: <https://www.ibm.com/products/systems-design-rhapsody/details> (Дата обращения 19.04.2020). Загл. с экрана. Яз. англ.
- 5 IBM Engineering Systems Design Rhapsody - Developer [Электронный ресурс]. — URL: <https://www.ibm.com/ru-ru/marketplace/uml-tools> (Дата обращения 19.04.2020). Загл. с экрана. Яз. рус.
- 6 Overview of Rational Rhapsody [Электронный ресурс]. — URL: https://www.ibm.com/support/knowledgecenter/SSQQC5_6.0.0/com.ibm.rational.sse.doc/topics/c_overview_rhap.html (Дата обращения 19.04.2020). Загл. с экрана. Яз. англ.
- 7 What is Arduino? [Электронный ресурс]. — URL: <https://www.arduino.cc/en/Guide/Introduction> (Дата обращения 19.04.2020). Загл. с экрана. Яз. англ.
- 8 Arduino Uno [Электронный ресурс]. — URL: <http://arduino.ru/Hardware/ArduinoBoardUno> (Дата обращения 19.04.2020). Загл. с экрана. Яз. рус.
- 9 About OpenCV [Электронный ресурс]. — URL: <https://opencv.org/about/> (Дата обращения 19.04.2020). Загл. с экрана. Яз. англ.
- 10 Flask framework: foreword [Электронный ресурс]. — URL: <https://flask.palletsprojects.com/en/1.1.x/foreword/> (Дата обращения 19.04.2020). Загл. с экрана. Яз. англ.