

Некоммерческое акционерное общество
«Алматинский университет энергетики и связи им. Гумарбека Даукеева»
Кафедра “Электроника и робототехника”

«ДОПУЩЕН К ЗАЩИТЕ»

Зав. кафедрой к.т.н., доцент Чигамбаев Т.О.

_____ «___» _____ 2020 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: “Разработка системы технического зрения манипулятора для погрузочных работ”

Специальность 5B071600 - Приборостроение

Выполнил Дөненбетов Алмас Ерланұлы Группа ПС(КИП)-16-4

Научный руководитель д.PhD, доцент Балбаев Г.К.

Консультанты:

по экономической части: доцент Боканова Г.Ш.

_____ «___» _____ 2020 г.
(подпись)

по безопасности жизнедеятельности: д.х.н. Приходько Н.Г.

_____ «___» _____ 2020 г.
(подпись)

Нормоконтролер: Фазылова А.Р.

_____ «___» _____ 2020 г.
(подпись)

Рецензент: доцент. Ожикенов К.А.

_____ «___» _____ 2020 г.
(подпись)

Алматы, 2020

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт “Космическая инженерия и телекоммуникации”
Кафедра “Электроника и робототехника”
Специальность “5В071600 - Приборостроение”

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Дөненбетову А.Е.

Тема проекта: «Разработка системы технического зрения манипулятора для погрузочных работ»

Утверждена приказом ректора № ____ от «__» _____ 2020 г.

Срок сдачи, законченной работы «__» _____ 2020г.

Исходные данные к проекту (требуемые параметры результатов проектирования) и исходные данные:

- 1) Arduino UNO, веб-камера Logitech, сервоприводы SG90;
- 2) Программные компоненты: Processing для написания программного кода и симуляции движений манипулятора, САПР FreeCAD для проектирования и дальнейшей сборки самого манипулятора, среда разработки Arduino для подачи сигналов серводвигателям.

Перечень подлежащих разработке в дипломном проекте вопросов или краткое содержание дипломного проекта:

1. Анализ видов распознавания объектов и существующих систем технического зрения.
2. Выбор подходящей технологии для манипулятора, занимающегося погрузочными работами.
3. Конструирование манипулятора.
4. Написание программного кода.
5. Реализация проекта.
6. Экономическая часть (расчет современных показателей экономической эффективности проекта).
7. Безопасность и жизнедеятельность (расчет освещения офисного помещения, расчёт зануления).

Перечень графического материала (с точным указанием обязательных чертежей): в данной работе содержится 45 рисунков и 11 таблиц.

Основная рекомендуемая литература:

1 Подкалицкий Л. Архитектура микроконтроллеров AVR: Пер. с англ. - 2019.

2 Г.Ш. Боканова. Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В071900 – Радиотехника, электроника и телекоммуникаций. – Алматы: АУЭС, 2020

3 Методические указания к выполнению раздела «Электробезопасность в электроустановках» в выпускных работах для специальности 050718 - Электроэнергетика. Бакалавриат - Алматы: АИЭС, 2009.

Консультанты по проекту с указанием относящихся к ним разделов работы:

Раздел	Консультант	Сроки	Подпись
Составление плана	Балбаев Г.К.	04.04.20	
Обзор существующих систем технического зрения	Балбаев Г.К.	06.05.20	
Конструирование манипулятора	Балбаев Г.К.	20.05.20	
Разработка программы системы технического зрения	Балбаев Г.К.	03.06.20	
Экономическая часть	Боканова Г.Ш.	18.05.20	
Безопасность жизнедеятельности	Приходько Н.Г.	19.05.20	

Г РАФИК
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечани е
1	2	3	4
1	Анализ видов распознавания объектов и существующих систем технического зрения.	04.04.20	
2	Выбор подходящей технологии для манипулятора, занимающегося погрузочными работами.	06.05.20	
3	Конструирование манипулятора.	20.05.20	
4	Написание программного кода.	23.05.20	
5	Реализация проекта.	06.06.20	
6	Экономическая часть	28.05.20	
7	Безопасность жизнедеятельности	29.05.20	

Дата выдачи задания «___» _____ 2020 г.

Заведующий кафедрой _____ (Чигамбаев Т.О.)
(подпись)

Руководитель _____ (Балбаев Г.К.)
(подпись)

Задание принял к
исполнению студент _____ (Дөненбетов А.Е.)
(подпись)

Аннотация

В данной дипломной работе рассмотрены вопросы о технологиях распознавания объектов с помощью технического зрения, используемого на погрузочных работах. Была создана система распознавания предметов по цветам пикселей и была разработана 3D модель манипулятора.

Также рассмотрены вопросы безопасности жизнедеятельности, рассчитаны экономические показатели эффективности проекта.

Андатпа

Бұл тезисте тиеу жұмыстарында қолданылатын техникалық көріністі қолдана отырып нысандарды тану технологиялары туралы сұрақтар қарастырылады. Нысандарды пиксель түстерімен тану жүйесі құрылды және манипулятордың 3D моделі жасалды.

Тіршілік қауіпсіздігі мәселелері де қаралды, жоба тиімділігінің экономикалық көрсеткіштері есептелді.

Annotation

In this thesis, questions are considered about technologies for recognizing objects using the technical vision used in loading operations. A system for recognizing objects by pixel colors was created and a 3D model of the manipulator was developed.

The issues of life safety were also considered, economic indicators of project efficiency were calculated.

Содержание

Введение	7
1 Анализ состояния проблемы, описание объекта автоматизации, требования к системе в целом и постановка задачи	8
1.1 Анализ состояния проблемы	8
1.2 Описание объекта автоматизации	8
1.3 Постановка задачи	11
1.4 Требования к использованным устройствам	11
1.5 Требования к возможностям системы	14
2 Математическая модель манипулятора, расчёт положения звеньев и сборка корпуса манипулятора в программе FreeCAD	15
2.1 Расчёт положения звеньев манипулятора	15
2.2 3D моделирование и сборка манипулятора	19
3 Разработка системы технического зрения манипулятора	22
3.1 Структурная схема	22
3.2 Функциональная схема	22
3.3 Выбор цветовой модели представления изображения для распознавания	23
3.4 Написание программы для распознавания изображения	28
4 Техничко-экономическое обоснование проекта	31
4.1 Расчёт капитальных вложений	31
4.2 Расчёт экономической эффективности проекта	38
4.3 Вывод	42
5 Безопасность жизнедеятельности	43
5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал	43
5.1.1 Микроклимат	43
5.1.2 Естественная освещенность	43
5.1.3 Уровень шума	45
5.1.4 Уровень электромагнитных излучений	45
5.1.5 Электробезопасность	46
5.2.1 Расчёт искусственного освещения офисного помещения	48
5.2.2 Расчёт защитного заземления	50
Заключение	52
Список сокращений	53
Список литературы	54
Приложение	55

Введение

Сегодня при активном развитии автоматизации производства приоритет отдается использованию промышленных манипуляторов. Это система, которая содержит много элементов для различных действий. Робототехника существенно отличается от обычных элементов автоматизации своей многофункциональностью. Производство манипуляторов позволяет человеку не работать в неблагоприятных для него условиях. Они способны выполнять физические и умственные действия, как человек. Для этого они наделены специальными датчиками, сенсорами, которые дают им возможность слышать, видеть, трогать и так далее. [1].

Манипуляторы первого поколения имеют программное управление, потому что они работают в соответствии с простой неизменной программой. Они были созданы для автоматизации совместных операций. Но их система управления очень примитивна и не позволяет им выполнять много действий. Этот тип манипулятора не способен адаптироваться к изменяющейся среде, он требует вмешательства человека и создания программы, которая позволит ему взаимодействовать с другими условиями.

Манипуляторы второго поколения - сенсорные манипуляторы, предназначенные для работы с объектами любой формы. Они лучше, чем манипуляторы первого поколения практически во всех отношениях, а именно: самая совершенная система управления (СУ), значительное количество различных внутренних и внешних датчиков.

Они могут проводить исследования, сборку и установку, анализировать окружающий их мир благодаря своей способности различать ситуацию вокруг них и быстро адаптироваться к меняющимся условиям.

Манипуляторы третьего поколения - это интеллектуальные манипуляторы, которые, помимо механических функций, обладают интеллектуальным потенциалом, иными словами, искусственным интеллектом (ИИ), они способны самообучаться на постоянной основе, тем самым автоматизируя не только физиологические и моторные функции, но и также умственные. Они превосходят манипуляторы предыдущих поколений во всех отношениях, его система управления способна адаптироваться во внешней среде путем анализа и реконструкции модели мира [1].

Целью дипломной работы будет создание технического видения для манипулятора второго поколения.

1 Анализ состояния проблемы, описание объекта автоматизации, требования к системе в целом и постановка задачи

1.1 Анализ состояния проблемы

Высокое качество, бесперебойность работы, возможность вовлечения их в области, вредные для человека, робототехника является одним из важнейших технологических этапов современной промышленности и, как правило, является залогом качества продукции. Единственная проблема автоматизации системы управления - самостоятельное позиционирование в свободном пространстве.

Управление программами - самый простой вид управления, оно предназначено для роботов без датчиков, поэтому для них пишется простая программа, которая постоянно повторяется.

Приборы с адаптивной системой управления оснащены сенсорной частью. Благодаря способности чувствовать и хотя бы каким-то образом взаимодействовать с окружающей средой, они могут анализировать и принимать решения об операциях.

Система технического зрения является одним из вариантов адаптивного управления, который предоставляет системе данные для идентификации объекта.

1.2 Описание объекта автоматизации

Производственный робот-манипулятор представляет собой комплекс, в состав которого входят манипулятор и автоматизированная организация управления, которая обеспечивает «интеллектуальную» работу устройства и позволяет его перепрограммирование в довольно широких пределах. Как правило, роботы считаются частью механического производства. Это экономически выгодно, потому что это значительно повышает эффективность [2].

Чаще всего под определением «промышленный робот» они принимают манипулятора - это непосредственно главный элемент системы. Следует подчеркнуть, что роботы различаются в первую очередь по своему назначению и вариантам конструкции манипулятора. В зависимости от проблемы, которую должен решить робот, используются разные типы проектов.

Основные конструкции манипуляторов:

Декартовые манипуляторы промышленных роботов. Эта система основана на концепции координат, исследованной философом Р. Декартом. Он состоит из нескольких линейных осей движения, расположенных точно вертикально друг к другу. Структурная и кинематическая схемы декартового манипулятора показаны на рисунке 1.1.

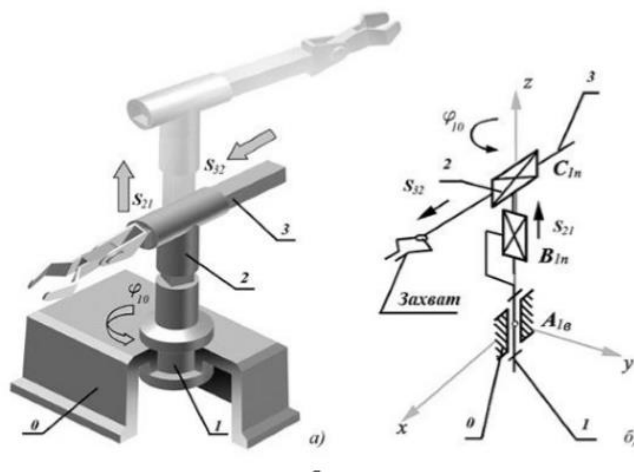


Рисунок 1.1 – Схемы декартового манипулятора

Декартовой дизайн был преимуществом, но производители теперь предоставляют готовое программное обеспечение для управления и обработки информации, поэтому это преимущество утратило свое значение, но при этом сам дизайн имеет низкая стоимость за счет упрощенного дизайна.

К недостаткам этого дизайна можно отнести только его размеры, поскольку существует множество сайтов с ограниченной доступностью.

Недостатки декартовых манипуляторов заставили инженеров разрабатывать более гибкие системы. А во время опроса 1981 года компания получила совершенно новую систему манипуляторов, показанную на рисунке 1.2. И они назвали это устройство (Выборочное соответствие сборки роботизированной руки), или вкратце - SCARA [3].



Рисунок 1.2 – Манипуляторы

Преимущества этой конструкции включают возможность свободно перемещаться, не теряя своей жесткости благодаря параллельному соединению. Поскольку структура состоит из двух звеньев, она может растягиваться и изгибаться в зависимости от ситуации.

В производстве одной из наиболее частых операций является перемещение детали из одной зоны в другую, и для этой операции были созданы роботы с параллельными стержнями. Суть работы такова: 3 «плеча»

прикреплены к установке, соединены параллельно, создавая кинематическую диаграмму параллелограмма. Благодаря этому механизму можно легко изменять скорость и усилия исходящей ссылки [3] в координате. Манипулятор легкий, поэтому он может работать на сверхскоростных скоростях. Манипулятор показан на рисунке 1.3.



Рисунок 1.3 – Параллельные роботы манипуляторы

В настоящее время популярны устройства под названием «дельта-роботы». Они были разработаны для ускорения процесса обработки легких предметов. Первый дельта-робот был представлен миру в 1980-х годах.

Суть работы дельта-робота: 3 манипулятора, которые приводятся в движение карданным приводом, который расположен в верхней части манипулятора, а расположенные ниже конструкции соединены небольшой треугольной платформой, а четвертый рычаг предлагает другая степень свободы - вращательная [3].

Шарнирные промышленные роботизированные руки. Этот тип манипулятора напоминает человеческую руку. Конструкция имеет как минимум 3 вращающихся соединения, благодаря которым формируется полярная система координат. Суть работы: основные шарниры обеспечивают вращение манипулятора, наклон в плечевом суставе и сгибание в локте; Другие 3 вращающиеся оси и 1 призматическое соединение гарантируют дополнительные степени свободы, благодаря чему рука выбирает все типы направлений и свободно проходит по радиусу. Главной особенностью этой конструкции является гибкость, которая позволяет манипулятору избегать препятствий. Эта конструкция находит применение в различных секторах, например, «рука» со сварочной головкой обращается с ней так же, как человек [3]. Кроме того, этот робот очень практичен, и его пространство для обслуживания намного больше, чем он занимает. На рисунке 1.4 показана шарнирная рука робота.

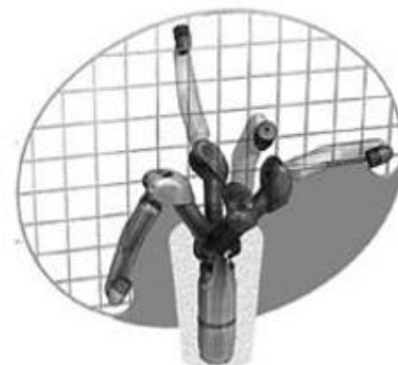
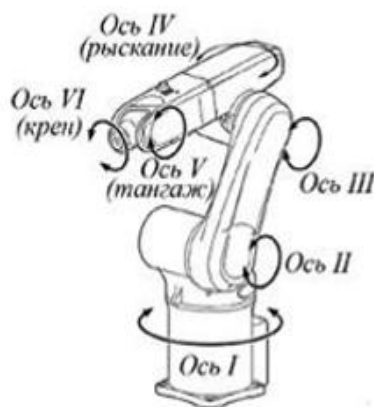


Рисунок 1.4 – Шарнирные роботы манипуляторы

1.3 Постановка задачи

В данном дипломной работе необходимо разработать ПО системы технического зрения для погрузочных работ, это будет проделяваться посредством ПК и веб-камеры. Технические требования всех используемых устройств, сборочных материалов рассмотрены ниже, также необходимо провести анализ части безопасности-жизнедеятельности и провести расчёт всех технико-экономических показателей данного проекта.

1.4 Требования к использованным устройствам

Система технического зрения будет состоять из веб-камеры, подключённой к ПК, в качестве этой камеры используется Logitech C-100 (рисунок 1.5). А также установленной на ПК среде разработки Processing, Arduino и программы для создания манипулятора FreeCAD. Технические характеристики камеры приведены в таблице 1.1.

Таблица 1.1 – Характеристики веб-камеры Logitech C 100 [4]

Наименование параметров	Значения
Разрешение	1.3 млн пикселей
Подключение	USB
Фокусировка	Ручная
Совместимые операционные системы	Windows XP/Vista/7/10



Рисунок 1.5 – веб-камера Logitech Webcam C 100

Также система управления (СУ) будет состоять из сервоприводов (SG-90), платы Arduino (рисунок 1.6) и ПК с программой Processing [5], получающей координаты цели и рассчитывающей требуемые углы поворота библиотека Servo.

Таблица 1.2 – Характеристики микроконтроллера Arduino UNO [4]

Наименование параметров	Значения
Микроконтроллер	ATmega328
Рабочее напряжение	5 В
Память	32 Кб
ОЗУ	2 Кб
EEPROM	1 Кб
Тактовая частота	16 МГц
Входное напряжение	7 – 12 В
Входное напряжение	6 – 20 В
Цифровые Входы/Выходы	20

Продолжение таблицы 1.2

Аналоговые каналы	12
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3.3 В	50 мА

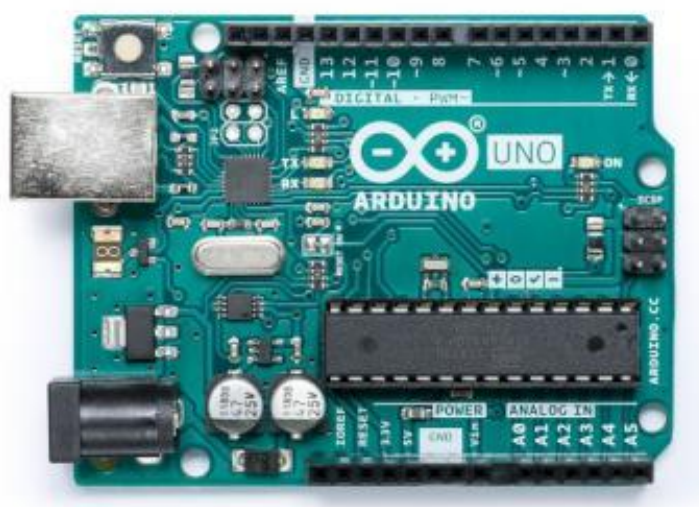


Рисунок 1.6 – плата Arduino

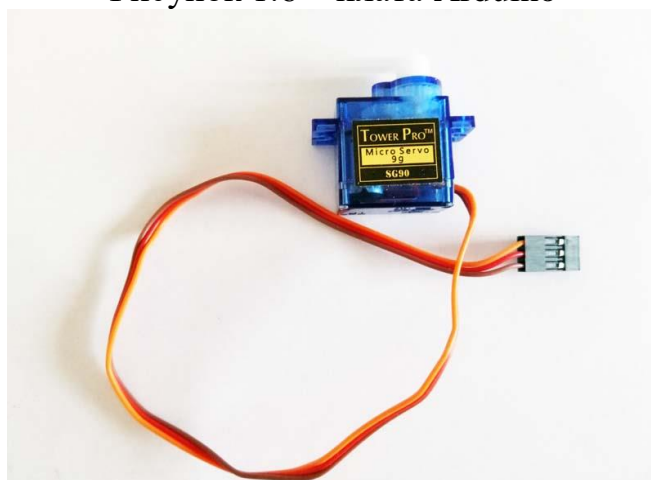


Рисунок 1.7 – сервопривод SG-90

Маленький сервопривод SG90. Размеры: 22.6 x 21.8 x 11.4 мм. Имеет три вывода: Плюс, Минус и Сигнал. Вращение ротора осуществляется на 180 градусов. Обеспечиваемый момент вращения при 4,8 В около 1,98 кг*см.

Скорость вращения — 60 градусов за 0,14 сек. Рабочее напряжение от 3,5 до 7,2 Вольт [4].



Рисунок 1.8 – подключение сервопривода SG-90

Чтобы рассчитать все без исключения нужные углы и координаты цели применяется Processing. Это программа, позволяющая создавать непростые анимации и интерактивные инсталляции при помощи языка программирования Processing, основанного на Java. Представляет из себя гибкий инструмент разработки с открытым исходным кодом, который отлично подойдет как для начинающих, так и для профессионалов. Первоначально предполагалось, что программа станет чем-то на подобии цифрового скетчбука, по этой причине она может похвастаться максимально простым, минималистическим дружеским интерфейсом [5].

В главном окне программы размещаются все требуемые элементы: панели инструментов, текстовый редактор, вкладки проекта и консоль. Все главные функции и инструменты всегда будут у вас под рукой, что даст возможность совершить работу над проектами, которые в языке Processing называются скетчами, еще комфортнее. А закончив работу над программами, вы сумеете привести в действие исполнение кода и приступить к взаимодействию со своим творением нажатием всего одной кнопки.

1.5 Требования к возможностям системы

Система технического зрения должна обеспечивать поиск и распознавание предметов, вычисление координат их местоположения относительно робота и также передачу преобразованных координат системе для расчёта углов поворота серводвигателей.

Требования от ПО:

- организация протокола связи с устройством;
- кроссплатформенность;

2 Математическая модель манипулятора, расчёт положения звеньев и сборка корпуса манипулятора в программе FreeCAD

2.1 Расчёт положения звеньев манипулятора

Задача системы управления (СУ) – устанавливать ошибку среди требуемым и текущим положениями схвата робота, рассчитывать передвижения отдельных звеньев и предоставлять управляющий сигнал на серводвигатели. СУ будет состоять из сервоприводов (SG-90), платы Arduino и ПК с Processing, получающей координаты цели и рассчитывающей необходимые углы поворота серводвигателей.

Общий алгоритм управления:

1) Определить требуемое состояние схвата: схват обязан прийти в точку цели. Цель будет подвергаться обработке с камеры, висящей над рабочей областью манипулятора, выделив на нем предмет необходимого цвета или формы.

2) Рассчитать необходимые положения звеньев, т.е. углы поворота двигателей. В программе на Processing, установив координаты цели, сразу же рассчитаем и все углы, высвобождая этим ресурсы Arduino.

3) Передача рассчитанных необходимых углов в Arduino. А дальше библиотека Servo и команда Servo:write даст возможность привести в движение сервомотор. Но попытка просто использовать эту функцию может послужить причиной тому, что в случае требования повернуться на большой угол, двигатель постарается моментально прийти в указанное положение. Манипулятор имеет некоторую массу и обладает инерцией, а это означает, что он будет сопротивляться изменению собственной скорости. Силы инерции могут сорвать зубья шестеренок серводвигателей. Следует попробовать сгладить переход между положениями, растянув его во времени.

Поэтому необходимо разбить весь переход на несколько временных шагов и в каждом передвигаться на часть, соразмерную ошибке.

Итак, микроконтроллер, получив значение необходимого угла поворота, рассчитает ошибку положения и за определенное количество тактов размеренно переместит манипулятор. От коэффициента пропорциональности зависит плавность.

Манипулятор имеет 3 степени свободы (3 цилиндрических шарнира – 3 оси вращения), что дает возможность ему передвигаться по трем координатам x , y , z , однако обойдемся плоскостью x и y .

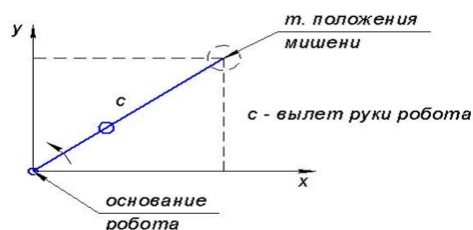


Рисунок 2.1 – Схема манипулятора в плоскости x , y .

На рисунке 2.1 видно схему манипулятора и точку положения мишени с координатами x, y , которые уже известны. Зная координаты, рассчитываем общий вылет манипулятора – C . Угол поворота основания рассчитывается как дуга, соответствующая $\sin(y/c)$.

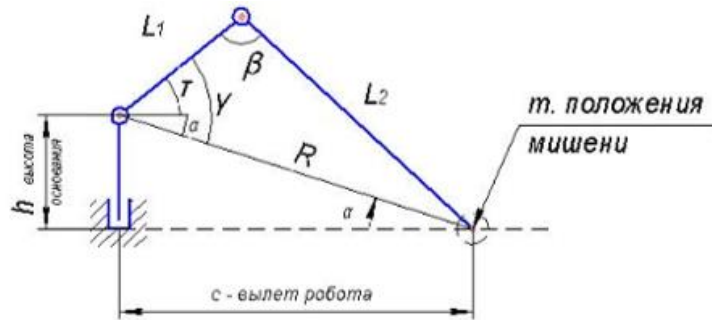


Рисунок 2.2 – Профильный вид манипулятора

Угол поворота основания робота:

$$\eta = \arcsin\left(\frac{y}{c}\right) \quad (2.1)$$

На профильном рисунке 2.2, соединив точки центров серв и хвата, получим треугольник со сторонами L_1, L_2, R . Длина стороны R рассчитывается из прямоугольного треугольника ChR (где h - высота основания робота).

$$R = \sqrt{h^2 + c^2} \quad (2.2)$$

$$\alpha = \arcsin\left(\frac{h}{R}\right) \quad (2.3)$$

$$\gamma = \arccos\left(\frac{L_1^2 + R^2 - L_2^2}{2L_1R}\right) \quad (2.4)$$

Угол поворота плеча (L_1):

$$\tau = \gamma - \alpha \quad (2.5)$$

Угол поворота предплечья (L_2):

$$\beta = \arccos\left(\frac{L_1^2 + L_2^2 - R^2}{2L_1L_2}\right) \quad (2.6)$$

Теперь напишем модель манипулятора в среде разработки, для этого создаем глобальные переменные для хранения размеров звеньев робота и текущих углов поворота этих звеньев, а также окно 800x600 (Рисунок 2.3):


```

1  /* Модель манипулятора
2  для отработки алгоритмов управления
3  */
4
5  /* Переменные с размерами */
6  float c = 0.0; //вылет робота
7  float h = 80;  //высота основания
8  float L1 = h*3; //плечо
9  float L2 = h*4; //предплечье
10
11 /* Переменные с углами манипулятора*/
12 float alfa;
13 float Beta,Beta2;
14 float gamma;
15 float tau,tau2;
16
17 void setup(){
18   size(800,600,P3D);
19 }
20
21 void draw(){
22
23   //Очищаем окно
24   background(#463F3F);
25
26   //Выводим значения углов
27   textSize(14);
28   fill(#26B41F);

```

Рисунок 2.3 – Написание программы симуляции работы звеньев манипулятора

В функции draw() в любом такте перерисовки экрана :

- отображаем цель по координатам мыши
- переносим систему координат в комфортное место с целью формирования первого звена (основания)
- рассчитываем положения звеньев манипулятора
- рисуем звено основания, плеча и предплечья (с последовательным переносом систем координат)

Функция расчета положений звеньев – computeRobotpos() (Рисунок 2.4):

```

59 void computeRobotpos(){
60
61   //Вылет
62   float c = (mouseX-width/4);
63   float R = sqrt(h*h + c*c);
64   println(R);
65   alfa = asin(h/R);
66
67   Beta = acos( (L1*L1 + L2*L2 - R*R) / (2*L1*L2));
68
69   gamma = acos( (L1*L1 + R*R - L2*L2) / (2*L1*R));
70
71   tau = gamma - alfa;
72
73   tau2 = 90 + degrees(alfa - gamma);
74   Beta2 = degrees(Beta) - 90;
75 }

```

Рисунок 2.4 – Расчёт положений звеньев манипулятора

Функции создания звеньев переносят систему координат (СК) в точку начала следующего звена, поворачивают её на рассчитанный угол и командой `line()` чертят линию вдоль оси X, по длине совпадающую с длиной данного звена (Рисунок 2.5):

```

93 void drawBase(){
94     //Точка центра
95     strokeWeight(3);
96     stroke(255,255,0);
97     point(0,0);
98
99     strokeWeight(3);
100    stroke(255,255,0);
101    line(0,0,0,h);
102    strokeWeight(15);
103    point(0,h);
104    drawlocalCS();
105 }
106
107 void drawArm(){
108     //переносим систему координат
109     //в точку начала звена
110     translate(0,h);
111     rotateZ((tau));
112
113     strokeWeight(3);
114     stroke(255,255,0);
115     line(0,0,L1,0);
116     strokeWeight(15);
117     point(L1,0);
118     drawlocalCS();
119 }
120
121 void drawForeArm(){
122     //переносим систему координат
123     //в точку начала звена
124     translate(L1,0);
125     rotateZ(-(PI-Beta));
126
127     strokeWeight(3);

```

Рисунок 2.5 – Функции создания звеньев

В результате получаем симуляцию работы манипулятора (Рисунок 2.6):

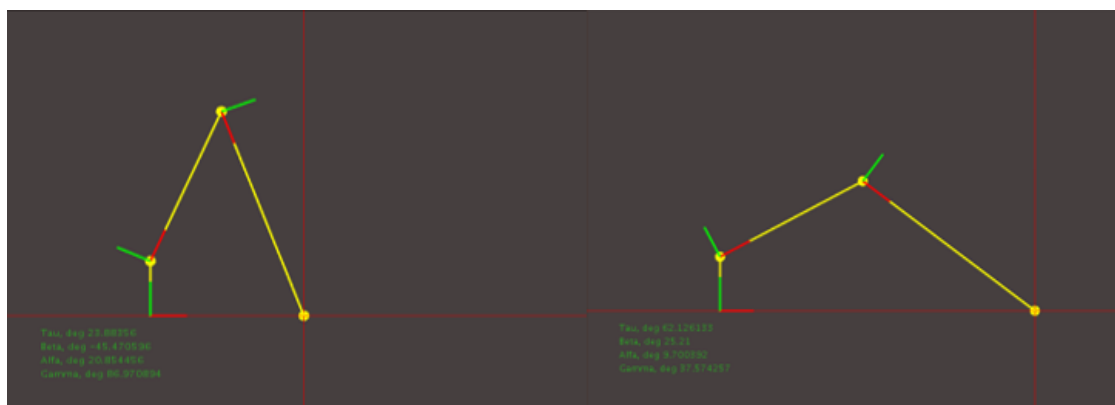


Рисунок 2.6 – Модель манипулятора

После создания каждой детали по отдельности, потребуется создать новый проект, в который мы будем добавлять все детали сборки с помощью команды «Объединить проект» (Рисунок 2.8).

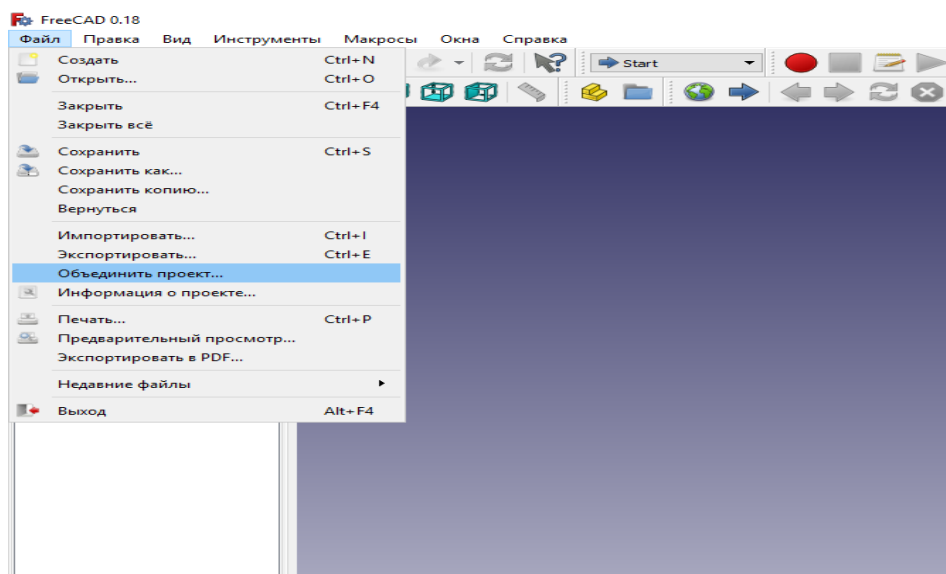


Рисунок 2.8 – Начало сборки манипулятора

Таким образом добавляем все детали и настраиваем сопряжения. Для сборки используется верстак – Part Design. На рисунках 2.9, 2.10, 2.11 можно наблюдать процесс сборки и готовую 3D модель манипулятора [8].

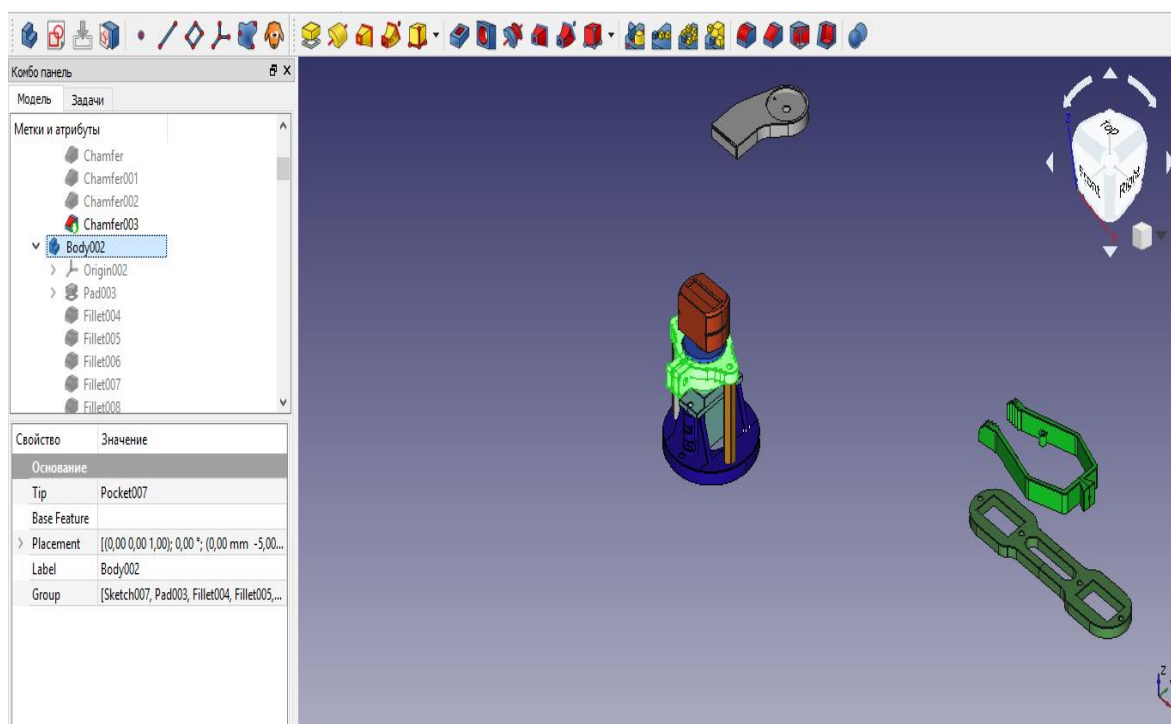


Рисунок 2.9 – Сборка манипулятора

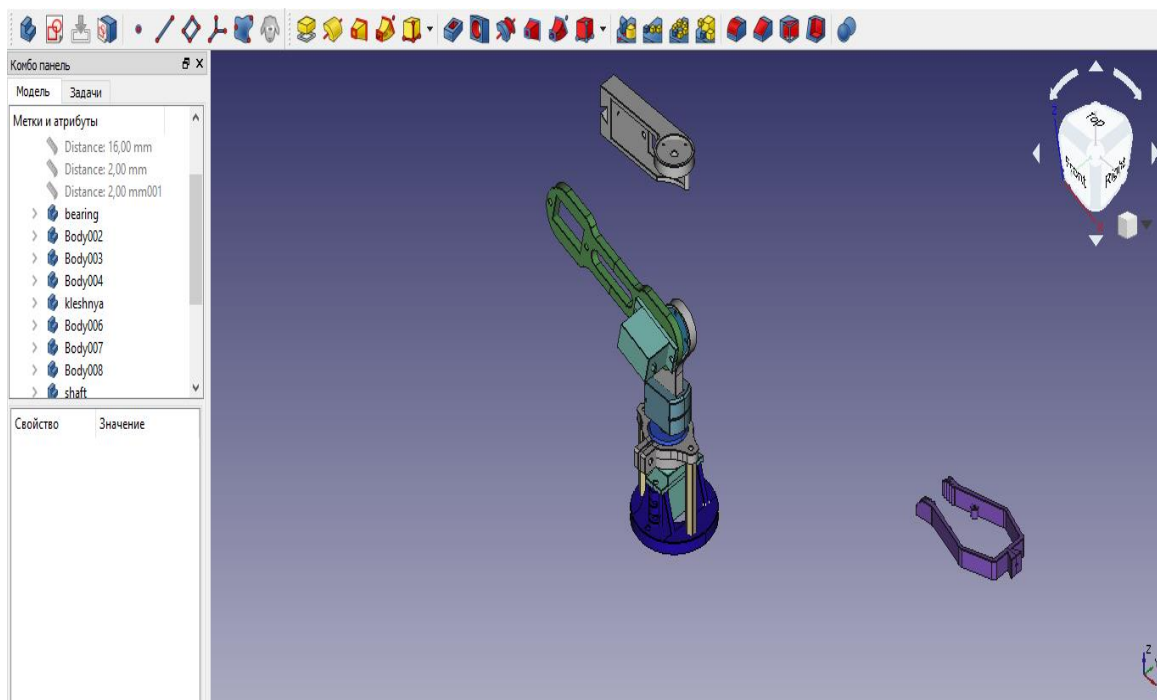


Рисунок 2.10 – Сборка манипулятора

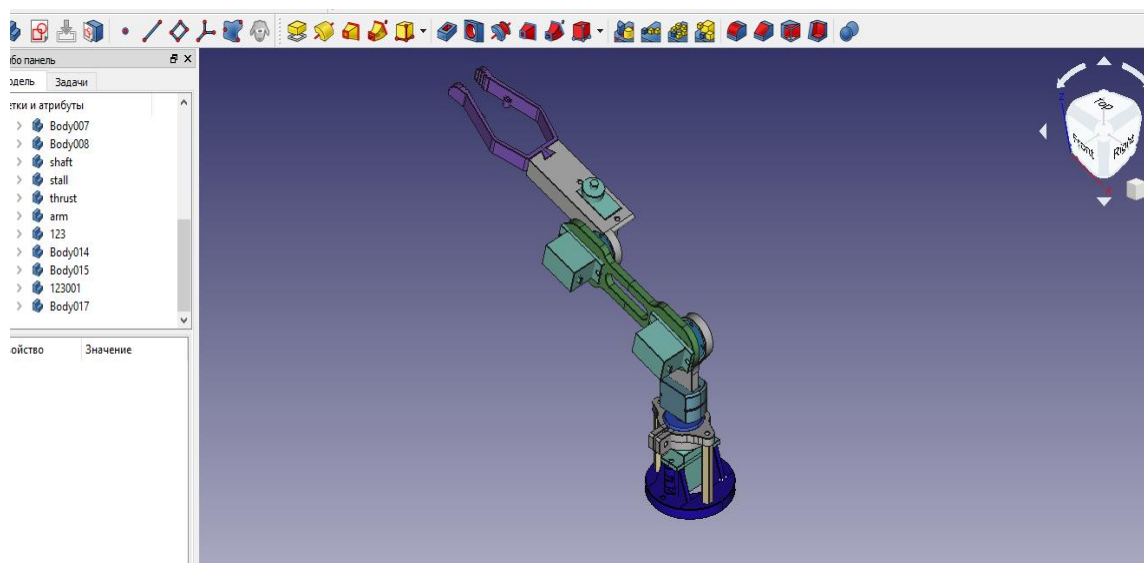


Рисунок 2.11 – 3D модель манипулятора

3 Разработка системы технического зрения манипулятора

3.1 Структурная схема

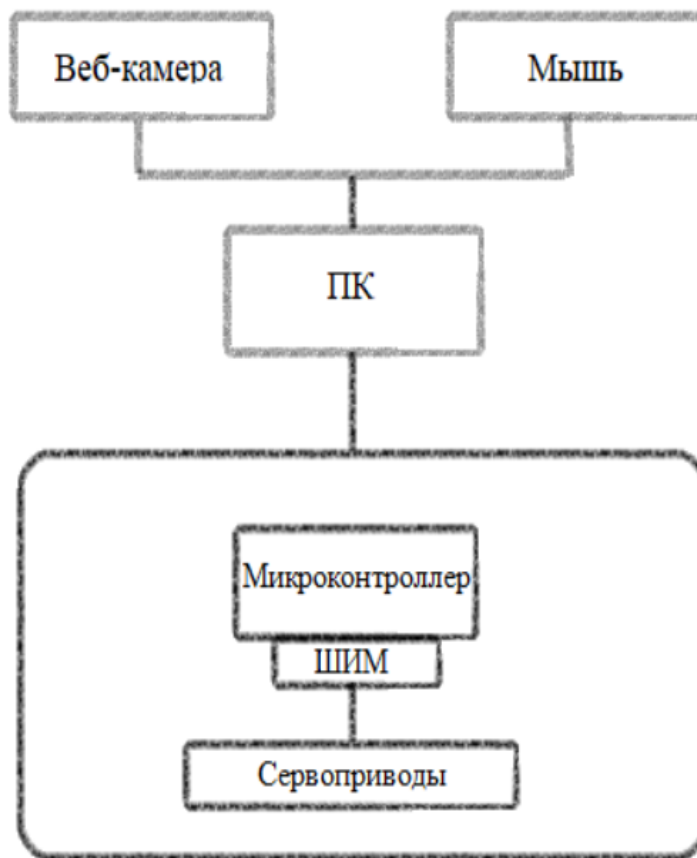


Рисунок 3.1 - Структурная схема

На рисунке 3.1 расположена структурная схема работы системы технического зрения манипулятора для погрузочных работ. Веб-камера расположена сверху относительно манипулятора и передает координаты изображения на ПК. Далее идёт обработка картинки и расчёт углов смещения манипулятора, а также дальнейшая передача их на микроконтроллер, который в последствии передаёт сигнал сервоприводам, приводящие захват манипулятора в точку назначения. Также для управления манипулятором можно использовать компьютерную мышь.

3.2 Функциональная схема

Функциональная схема устройства управления манипулятором представлена на рисунке 3.2 [11].

В состав устройства управления манипулятором входят:

- микроконтроллер АТМega328;
- интегральная схема МАХ232, преобразующая сигналы последовательного порта RS-232 в сигналы, пригодные для использования в цифровых схемах на базе ТТЛ;

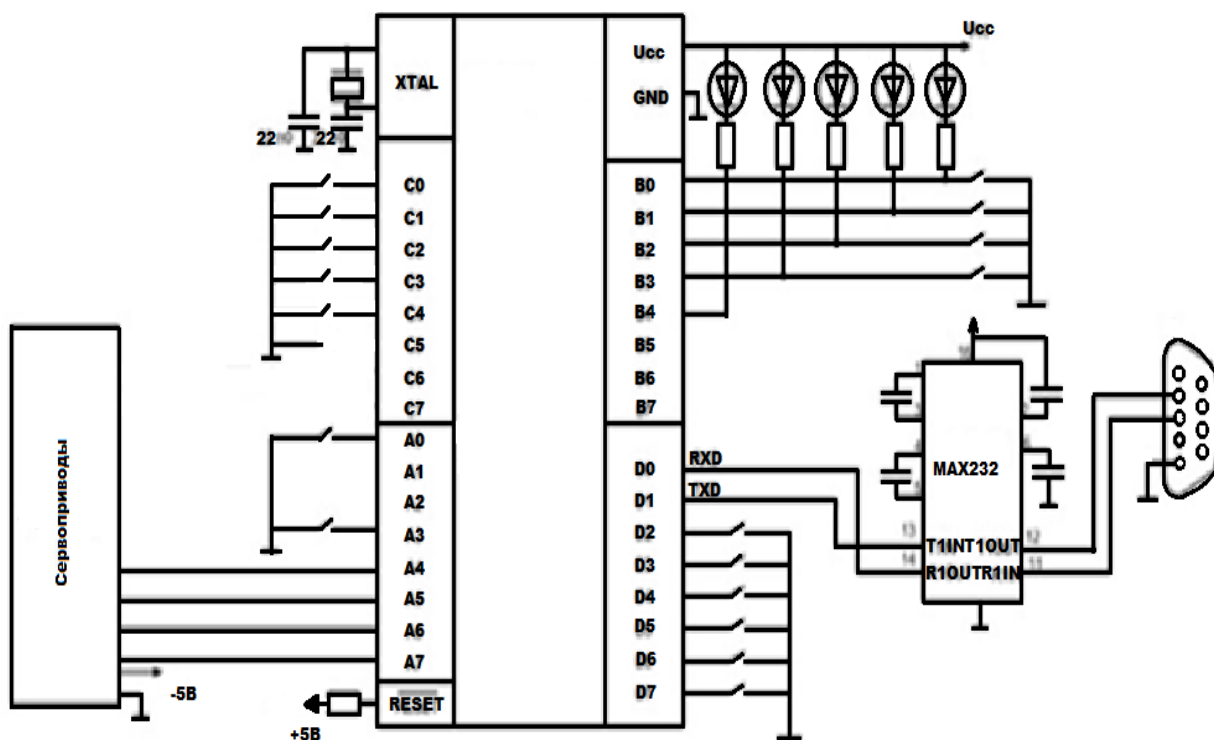


Рисунок 3.2 – Функциональная схема

3.3 Выбор цветовой модели представления изображения для распознавания

Изображение с камеры будет отражено в модели RGB (красный, зеленый, синий). Чтобы найти контуры с использованием модели RGB, используется следующий алгоритм действий: изображение отображается серым цветом, а пиксели с оттенком серого окрашиваются в белый цвет, а все остальные - в черный [11].

Конечно, эта модель имеет свои существенные недостатки, то есть сильное влияние теней и слабого освещения сильно влияет на эффективность определения. Маркеры еще сложнее различить, поскольку в условиях низкой освещенности зеленый оттенок не будет подходящим для условий выбора, и в этом случае весьма вероятно, что пиксель не будет идентифицирован.

И еще одним недостатком является то, что есть только три координаты для установки оттенка цвета. Установить желаемый цвет невероятно сложно, так, как только три координаты задают тон.

Учитывая эти недостатки, было решено проанализировать все модели, чтобы выбрать наиболее оптимальную для этой работы.

Восприятие тона зависит от физических качеств света, то есть от электромагнитной энергии, ее взаимодействия с физическими элементами, а также от их интерпретации зрительной системой человека. Зрительная система человека воспринимает электромагнитную энергию с длинами волн от 400 до 700 нм в виде видимого света ($1 \text{ нм} = 10^{-9} \text{ м}$), как показано на рисунке 3.1








	Цвет	Длина Волны
Фиолетовый		380-430
Синий		470-500
Голубой		430-470
Зеленый		500-560
Желтый		560-590
Оранжевый		590-620
Красный		620-760

Рисунок 3.1 – Видимый свет

Свет воспринимается от источника света с помощью рефракции или отражения. Источник или объект считается бесцветным, если наблюдаемый свет охватывает все видимые длины волн, приблизительно равные [12].

Яркость или яркость объекта зависит от относительной чувствительности глаза к разным длинам волн. При дневном свете чувствительность глаза максимальна на длине волны около 550 нм, а по краям видимого спектра быстро уменьшается.

Это мера энергии света или насыщенности, учитывающая свойства глаза. Если воспринимаемый свет содержит длины волн в случайных, неравных количествах, это называется хроматическим.

Если длины волн находятся в верхней части диапазона, свет станет красным; если длины волн находятся в нижней части диапазона, свет будет синим, но электромагнитная энергия бесцветна. Цвет объекта зависит от распределения длин волн источника света и физических качеств объекта. Объект выглядит цветным, если он отражает или пропускает свет только в диапазоне ограниченных длин волн и поглощает все остальные. Взаимодействие падающих цветов и отражений приводит к непредсказуемым результатам.

Обычно это не чистые монохроматические тона, а смеси. Трехкомпонентная теория света основана на предположении, что в центральной части сетчатки есть три типа чувствительных к цвету колбочек. Первый принимает длины волн, которые находятся в середине видимого спектра, то есть зеленый; вторая длина волны в верхней части видимой полосы, вторая длина волны на верхнем крае видимой полосы, то есть красный; Третьи - короткие волны от нижней части диапазона, то есть синие. Относительная чувствительность глаза, как показано на рисунке 3.3, максимальна для зеленого и минимальна для синего. Если на все три типа колбочек влияет одна и та же яркость энергии (энергия в единицу времени), свет будет белым.

Модель RGB также называется аддитивной, поскольку цвет становится светлее по мере увеличения яркости определенных цветов. На рисунке 3.2 показано, как смешиваются цвета.

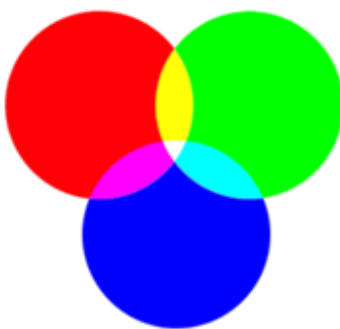


Рисунок 3.2 – Смесь тонов в RGB

Поскольку эта цветовая модель позволяет отображать на экране почти все 16 миллионов цветов, эта модель является наиболее подходящей, но не все созданные цвета можно распечатать.

Другой моделью, используемой в компьютерной графике, является модель HSV. Растровые форматы не используют модель HSV для сохранения изображений, так как она включает в себя только 3 миллиона цветов.

В модели HSV цвет раскладывается на 3 компонента:

- оттенок - цветовой тон, каждое значение соответствует одному цвету, например, 0 - это красный, 45 - это оттенок оранжевого и 55 - это оттенок желтого. Иногда снижается до диапазона 0-100 или 0-1;

- насыщенность - насыщенность. Это соотношение основного тона и его равных по яркости бесцветному серому. Наиболее насыщенный цвет вообще не содержит серого. Чем ниже насыщенность цвета, тем он более нейтральный, тем сложнее однозначно охарактеризовать его. Чем больше этот параметр, тем «чище» цвет, поэтому этот параметр иногда называют чистотой цвета.

- значение цвета - общая яркость цвета. Минимальное значение этого параметра превращает любой цвет в черный.

Графическое представление этой цветовой модели показано на рисунке 3.3.

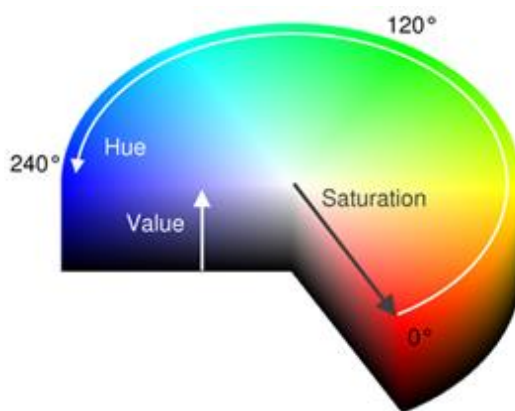


Рисунок 3.3 – HSV

Преимущество модели HSV по сравнению с RGB заключается в том, что цвет задается одним координатным оттенком, что значительно упрощает выбор необходимых тонов на изображении. Кроме того, используя насыщенность и значение координат, можно полностью уменьшить или полностью устранить эффект слабого освещения, затемнения или теней. Цветовая модель HSL, также известная как HLS или HSI, представляет собой следующее:

- HUE - тип цвета. – Изменяется от 0 to 360°, каждое значение соответствует одному цвету, например, 0 – это красный цвет, 45 – оттенок оранжевого и 55 – оттенок желтого.
- SATURATION – насыщенность, вариации цвета в зависимости от яркости. Изменяется от 0 до 100% (от центра оси L черный-белый).
- LIGHTNESS (а также Luminance, или Luminosity, или Intensity) – светлота. Изменяется от 0 до 100% (от черного к белому).

Этой модели соответствуют следующие графические интерпретации, приведённые на рисунке 3.4 [13].

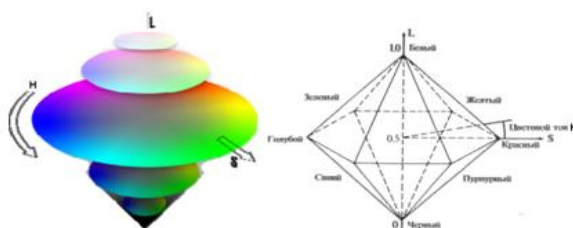


Рисунок 3.4 – HSL

Цветовая модель HLS (оттенок, яркость, насыщенность) в типе с двойным шестиугольным конусом рассматривается как расширение конуса HSV. Поскольку модель HLS используется для самосветящихся объектов, яркость здесь означает яркость. В модели HLS цветовой куб RGB проецируется таким образом, чтобы получить двойной шестиугольный конус с яркостью вдоль оси от 0 (черный) до вершины 1 (белый) во второй (рис. 3.7). Как и в модели HSV, насыщенность определяется радиальным пространством от главной оси. Полностью насыщенные основные тона и их дополнения получаются при $S = 1$.

Цилиндрическое представление также используется в цветовой модели Mansell, которая основана на серии образцов цвета. В этой модели цвет определяется оттенком, насыщенностью (чистотой) и яркостью (освещенностью). На центральной оси цилиндра находятся значения насыщенности от черного внизу до белого сверху. Интенсивность или чистота цвета увеличивается с увеличением радиального расстояния от оси. Оттенок определяется углом, как показано на рисунке 3.5.



Рисунок 3.5 – Концептуальное представление Mansell

Наиболее важным преимуществом модели Mansell является то, что подобные приращения насыщенности, тона и интенсивности активируют ощущения аналогичных изменений в восприятии. Существует трансформация субъективной концепции цвета в модели Mansell в фундаментальные тона системы CIE. Можно использовать цвета Mansell при выводе на экран. В то же время основные тона МСО предназначались в качестве промежуточного обычного цветового пространства: сначала координаты Mansell реорганизуются в основные цвета системы CIE XYZ, но затем они изменяются на координаты RGB для цветного монитора. Используя этот метод, были получены определенные цвета, которые ранее были известны только как экстраполяция существующих стандартов. Отсюда практическое значение обычного цветового пространства CIE (CIE).

Модель RGB, в которой изображение излучается камерой, хотя она может представлять все 16 миллионов цветов, которые может отображать дисплей, считается неудобной в смысле цветового кодирования, поскольку все 3 координаты используются для активности цветового тона, и это также создаст проблемы при настройке цветового спектра для экранирования. Кроме того, довольно сложно уменьшить влияние слабого освещения или теней в рабочем пространстве камеры, изменив все 3 цветовых элемента.

Есть еще три цветовые модели, которые похожи по набору элементов, кодирующих тон, - это модели HSV, HSL и Mansell. В них для определения цветового тона используется только одна координата, что облегчает выбор только необходимых оттенков изображения. Кроме того, благодаря поддержке координат насыщенности и значения можно уменьшить или полностью устранить эффект слабого освещения, затемнения или теней в рабочей области камеры. Хотя модель HSL имеет более широкий спектр, чем модели HSV и Mansell, было решено использовать модель HSV, поскольку 3 миллиона цветов,

которые можно представить с ее помощью, достаточно для нахождения контуров и минимизации эффектов слабого освещения и теней.

3.4 Написание программы для распознавания изображения

Ранне была спроектирована кинематика движения, систему звеньев и алгоритм движения.

Система управления робота, исполняющая наш алгоритм, рассчитает углы поворота звеньев в случае если мы обеспечим ей координаты точки, в которую следует привести захват робота.

Робот станет охотиться за мишенями соответствующего тона, помещенными в его рабочую область.

Повесим веб-камеру над рабочей областью, в Processing захватим видеоматериал с данной камеры.

Обработка в Processing будет представлять чтение отображения с камеры несколько раз в одну секунду, а потом выделения пикселей необходимого тона. Отыскав все оптимальные пиксели, зная их координаты, установим “центр масс” всей фигуры – это и есть точка, целеуказание для робота (рисунок 3.14).

```
import processing.video.*;

Capture cam;

float huesel = 0.0; //Оттенок цвета
float satsel = 0.0;
float brsel = 0.0;

//Переменные для расчета центра фигуры
int ii=0;
float sumX=0.0;
float sumY=0.0;

int w=320;
int h=240;
```

Рисунок 3.14 – Создание переменных

Сама обработка взятой картинки (функция draw) состоит в обходе абсолютно всех пикселей в двойном цикле for. Внешний цикл по столбцам по координате X экрана – перебираем целиком по счетчику i от нулевого до ширины экрана width.

Внутренний цикл по координате Y перебирает все пиксели этого столбика (выбранного во внешнем цикле) передвигаясь по счетчику j от нулевой строки до height.

Получить тон пикселя можно по его координатам на экране функцией get(i,j)

Сравнивая тон текущего пикселя с установленным цветом, в блоке if решаем, принадлежит данный пиксель к избранной мишени или нет.

В конце нужно рассчитать смещение координат. В Processing координаты отсчитываются от верхнего левого угла и координаты мишени для робота необходимо пересчитать, так как ноль координат в системе робота был в центре его основания, показано на рисунке 3.15.

```
void draw() {
    //Захватываем изображение с камеры
    if (cam.available() == true) {
        cam.read();
    }

    //Выводим изображение в окно
    image(cam, 0, 0, width, height);

    ii=0; sumX=0.0; sumY=0.0;
    for (int i=0;i<w;i+=1){
        for (int j=0; j<h;j+=1){
            //получаем цвет текущего пикселя
            color cc = get(i,j);
            //и переводим его в HSB-пространство
            float h = hue(cc);
            float s = saturation(cc);
            float br = brightness(cc);
            //Сравниваем тек. пиксель с выбранным с учетом
            //некоторого допуска на отличие в цвете
            if (h<=(huesel+15) && h>=(huesel-15) &&
                s<=(satsel+15) && s>=(satsel-15) &&
                br<=(brsel+150) && br>=(brsel-150))
            {
                sumX+=i;
                sumY+=j;
                ii++;

                color cch = color(255,0,0);
                set(i,j,cch); //set(i,j-1,cch);set(i-1,j-1,cch);
            }
        }
    }
}
```

Рисунок 3.15 – Обработка изображения

Функция mouseclicked() необходима для указания необходимого тона.

Применяемая шифровка цвета – HSB. В данной кодировке непосредственно оттенок цвета установлен первоначальным значением, по которому комфортно сравнивать пиксели.

Центр фигуры определяется как отношение суммы координат точек к их количеству (рисунок 3.16).

```

    }
}

//вычисляем центр фигуры из найденных пикселей
float cX = sumX/ii;
float cY = sumY/ii;

//перекрестие
strokeWeight(5);
stroke(255,0,0);
point(cX,cY);
strokeWeight(1);
line(cX,0,cX,height);
line(0,cY,width,cY);

//перевод координат в мм
float cXmm = cX*167/320;
float cYmm = cY*125/240;
fill(230,0,0);
text("X" " " " " +cXmm,width-50,20);
text("Y" " " " " +cYmm,width-50,40);

}

void mouseClicked(){
    color col = get(mouseX,mouseY);
    huesel = hue(col);
    satsel = saturation(col);
    brsel = brightness(col);
}

```

Рисунок 3.16 – Определение центра фигуры и метод mouseClicked()

4 Технико-экономическое обоснование проекта

4.1 Расчет капитальных вложений

Чтобы рассчитать капитальные вложения нужно воспользоваться формулой [14]:

$$K = K_{\text{пр}} + K_{\text{об}} + K_{\text{сбор}} + K_{\text{тр}} + K_{\text{м}} + C_{\text{п}}, \quad (4.1)$$

где $K_{\text{пр}}$ – капитальные инвестиции в исследование НИР, тг;
 $K_{\text{об}}$ – расходы на спецоборудование, тг;
 $K_{\text{сбор}}$ – капитальные инвестиции в производство манипулятора, тг;
 $K_{\text{тр}}$ – капитальные инвестиции в автотранспортные сервисы, тг;
 $K_{\text{м}}$ – капитальные инвестиции в приобретение использованных материалов, тг;

$C_{\text{п}}$ – капитальные инвестиции в разработку ПО, тг.

Расходы на исследование НИР согласно формуле:

$$K_{\text{пр}} = Z_{\text{фот}} + Z_{\text{с}} + Z_{\text{об}} + Z_{\text{н}}, \quad (4.2)$$

где $K_{\text{пр}}$ – расходы на НИР, тг;

$Z_{\text{фот}}$ – единый фонд оплаты работы разработчиков, тг;

$Z_{\text{с}}$ – отчисления согласно социальному налогу, тг;

$Z_{\text{об}}$ – особое спецоборудование, тг;

$Z_{\text{н}}$ – накладные затраты, тг.

Единый ФОТ разработчиков ($Z_{\text{фот}}$) рассчитаем, как совокупность основной и добавочной зарплаты согласно формуле:

$$Z_{\text{фот}} = Z_{\text{о.}} + Z_{\text{д}}, \quad (4.3)$$

где $Z_{\text{о}}$ – основная зарплата, тг;

$Z_{\text{д}}$ – добавочная зарплата, тг.

Таблица 4.1 – Основная зарплата на исследования системы технического зрения манипулятора [14]

№	Наименование этапов	Исполнители (по категориям)	Трудоемкость (чел./дни)	Должностной оклад (тг)	Стоимость 1 чел.дн. (тг)	Заработная плата (тг)
1	Постановка целей и задач исследования	Инженер	2	126000	6000	12000

Продолжение таблицы 4.1

2	Разработка и утверждение технического задания (ТЗ)	Инженер	1	126000	6000	6000
3	Изучение основ проектирования и программирования систем технического зрения	Инженер	1	126000	6000	6000
4	Подбор используемого оборудования	Инженер	1	105000	5000	5000
5	Разработка рабочей документации	Руководитель	2	157500	7500	15000
		Инженер	1	126000	6000	6000
6	Составление алгоритма работы системы управления	Инженер	1	126000	6000	6000
7	Проверка результатов	Руководитель	1	157500	7500	7500
	Итого		10			63500

По таблице 4.1 общая длительность выполнения работ составит 10 рабочих дней.

Добавочная зарплата сотрудников, которая предусматривает заболевания, отпуска и прочие внезапные обстоятельства (как правило, 10% от зарплаты):

$$З_{д} = З_{о} \times H_{д}/100, \quad (4.4)$$

где $H_{д}$ – коэффициент добавочной зарплаты программистов.

$$З_{д} = 0,1 \times 63500 = 6350 \text{ (тг)}.$$

По формуле 4.3 ФОТ:

$$З_{\text{фот}} = 6350 + 63500 = 69\,850 \text{ (тг)}.$$

Отчисления в единый общественный налог рассчитываются, основываясь на заработной плате работника и составляет 9,5% [14]:

$$З_c = (З_{\text{фот}} - \text{ПО}) \times 9,5\%, \quad (4.5)$$

где ПО – взнос в Пенсионный фонд в РК:

$$\text{ПО} = З_{\text{фот}} * 10\% , \quad (4.6)$$

Итак, $\text{ПО} = 69850 * 0,1 = 6985$ (тг).

$$З_c = (69850 - 6985) \times 0.095 = 5972.18 \text{ (тг)}.$$

Расходы на оборудование составляют 50500 тг (таблица 4.2).

Таблица 4.2 - Затраты на спецоборудование [15]

№	Название материалов	Единица измерения	Количество	Цена, тг	Сумма, тг
1.	Компьютер	шт.	1	50000	50000
2.	FreeCAD	шт.	1	10000	10000
3.	Microsoft Windows 10	шт.	1	3500	3500
	Итого				63500

Накладные затраты составят 30% от зарплаты:

$$З_n = 0,3 \times 69850 = 20955 \text{ (тг)}.$$

Таким образом, согласно формуле 2 расходы на создание НИР будут равны:

$$K_{\text{пр}} = 69850 + 5972.18 + 63500 + 20955 = 160277.18 \text{ (тг)}.$$

Капитальные инвестиции на спецоборудование составляют 37900 тг. Список нужного спецоборудования изображен в таблице 4.3.

Таблица 4.3 – Список нужного спецоборудования для реализации проекта [14]

Название оборудования	Количество, ед.	Цена за единицу, тг	Стоимость с учетом НДС, тг	Стоимость без НДС, тг
Микроконтроллер Arduino	1	5000	5600	5000
Веб-камера	1	7000	7840	7000
Составляющие манипулятора	1	15000	16800	15000
Wi-Fi Роутер	1	10000	11200	10000
Сервопривод SG90	3	300	1008	900
Итого			42448	37900

Для расчета капитальных инвестиций на производство воспользуемся формулой:

$$K_{\text{сбор}} = K_{\text{об}} \times 10\%, \quad (4.7)$$

Подобным образом, $K_{\text{сбор}} = 37900 \times 10\% = 3790$ (тг).

Капитальные инвестиции в автотранспортные обслуживание рассчитываются согласно формуле:

$$K_{\text{тр}} = K_{\text{об}} \times 5\%, \quad (4.8)$$

Таким образом, $K_{\text{тр}} = 37900 \times 5\% = 1895$ (тг).

Капитальные инвестиции на использованные материалы, требуемые для осуществления проекта представлены в таблице 4.4

Таблица 4.4 – Затраты на материалы [15]

№ п/п	Название материалов	Единица измерения	Количество	Цена, тг	Сумма, тг
1.	Бумага для принтера, упак. 500 листов	шт	1	700	700
2.	ВН1750	шт	2	100	200
3.	Соединительные болты	шт	100	4	400
4.	АТmega328	шт	1	2000	2000
5.	Провода	шт	40	8.75	350

Продолжение таблицы 4.4

6.	LF328Z	шт	1	950	950
7.	XMS18	шт	1	900	900
	Итого				5500

Расходы (C_{Π}) на разработку ПО [13]:

$$C_{\Pi} = Z_{\text{фот}} + Z_{\text{с}} + P_{\text{м}} + P_{\text{э}} + P_{\text{н}}, \quad (4.9)$$

где $Z_{\text{фот}}$ – единый фонд оплаты работы, тг;

$Z_{\text{с}}$ – социальные отчисления, тг;

$P_{\text{м}}$ – расходы на амортизацию техники, тг;

$P_{\text{э}}$ – расходы на электроэнергию, тг;

$P_{\text{н}}$ – накладные, тг.

Расчёт зарплаты:

$$Z_{\text{o}} = t_{\text{р}} \times Z_{\text{по}}, \quad (4.10)$$

где $t_{\text{р}}$ – трудоемкость разработки, чел-дни;

$Z_{\text{по}}$ – зарплата разработчика, (тг).

Итоговые расходы работы рассчитываются как совокупность составных расходов труда согласно формуле, чел/час:

$$t = t_{\text{o}} + t_{\text{а}} + t_{\text{н}} + t_{\text{отл}} + t_{\text{д}}. \quad (4.11)$$

где t_{o} – время затраченное на описание задачи, чел/час;

$t_{\text{а}}$ – составление алгоритма программы, чел/час ;

$t_{\text{н}}$ – время затраченное на написание программы, чел/час;

$t_{\text{отл}}$ – время, отведенное на отладку ПО на ПК, чел/час;

$t_{\text{д}}$ – время затраченное на разработку документации, чел/час.

Условное число операторов в программе задачи определяется по формуле:

$$Q = q \times c, \quad (4.12)$$

где Q – условное число операторов;

q – предполагаемое число операторов в зависимости от типа задачи;

c – коэффициент, учитывающий новизну и сложность программы.

Таким образом, $Q = 1500 \times 1.0 = 1500$ (команд).

После этого необходимо определить время, затраченное на создание ПО на каждом этапе:

1) t_0 берется по факту: $t_0 = 24$ (чел / час).

2) t_a найдем по формуле:

$$t_a = Q / (50 \times K), \quad (4.13)$$

где K – коэффициент, учитывающий квалификацию программиста (таблица 4.7).

Таким образом, $t_a = 1500 / (50 \times 1.1) = 27.27$ (чел/час).

3) t_H найдем по формуле:

$$t_H = Q \times 1,5 / (50 \times K), \quad (4.14)$$

Таким образом, согласно формуле 6.14:

$$t_H = 1500 \times 1,5 / (50 \times 1.1) = 40.9 \text{ (чел/час)}.$$

4) $t_{отл}$ определяется по формуле:

$$t_{отл} = Q \times 4,2/50 \times K, \quad (4.15)$$

Согласно формуле 4.15:

$$t_{отл} = 1500 \times 4,2/50 \times 1.1 = 114.54 \text{ (чел/час)}.$$

5) t_d , берется по факту и составляет (от 3-х до 5-ти дней по 8 часов):

$$t_d = 24 \text{ (чел / час)}.$$

Исходя из формулы 4.11:

$$t = 24 + 27.27 + 40.9 + 114.54 + 24 = 235.71 \text{ (час)}.$$

Подобным способом, время, потраченное на создание ПО составляет 235.71 часов либо 29.4 чел./дня.

Таблица 4.5 – Сводные результаты расчета затрат основной заработной платы

Наименование содержания работ	Исполнитель	Трудоёмкость норма-час	Заработная плата за час работы	Сумма заработной платы
1	1	8	750	6000

Основная зарплата:

$$З_o = t_p \times З_{по}, \quad (4.16)$$

где t_p – трудозатратность разработки программного продукта, чел-дни;

$З_{по}$ – дневная зарплата разработчика, тг (таблица 4.8) ;

Поэтому, $З_o = 29.4 \times 6000 = 176\,400$ (тг).

Добавочная зарплата ($З_d$):

$$З_d = 176\,400 \times 0.1 = 17\,640 \text{ (тг).}$$

Подобным образом, ФОТ:

$$З_{ФОТ} = 176\,400 + 17\,640 = 194\,040 \text{ (тг).}$$

Социальный налог рассчитывается согласно формуле 4.5.

Где ПО:

$$\text{Так, } ПО = 194\,040 \times 10\% = 19\,404 \text{ (тг).}$$

$$\text{Подобным способом, } З_c = (194\,040 - 19\,404) \times 9,5\% = 16\,590.42 \text{ (тг).}$$

Расходы на амортизацию:

$$P_m = C_{обор} * N_a * N / 100 * 12 * t, \quad (4.17)$$

где N_a – норма амортизации (25%);

$C_{обор}$ – начальная цена оборудования (тг).;

N – время применения ПК (дней);

t – число рабочих дней в месяце.

$$P_m = \frac{60000 \times 25\% \times 29.4}{100 \times 12 \times 22} = 175 \text{ (тг).}$$

Расходы на электрическую энергию:

$$P_э = M \times k \times T \times C_{кВт-ч}, \quad (4.18)$$

где M – мощность ЭВМ, кВт;

k – коэффициент загрузки (0.8);

$C_{кВт-ч}$ – цена 1 кВт-час электроэнергии, тенге/ кВт-час;

T – время работы, час.

$$\text{Отсюда следует, } P_э = 0,08 \times 0,8 \times 235.71 \times 16,65 = 251,17 \text{ (тг).}$$

Накладные расходы (P_n) составляют 20% от основной заработной платы:

$$P_n = 176400 \times 20\% = 35280 \text{ (тг)}.$$

Следовательно, расходы на создание ПО:

$$C_{\Pi} = 194040 + 16590.42 + 175 + 251.17 + 35280 = 246336.59 \text{ (тг)}.$$

Подобным образом, общая совокупность капитальных инвестиций на реализацию проекта без учета НДС:

$$K = 160277.18 + 37900 + 3790 + 1895 + 5500 + 246336.59 = 455698.77 \text{ (тг)}.$$

4.2 Расчет экономической эффективности проекта

Целесообразность введения манипуляторов и ПР обуславливается соответствующими условиями:

- необходимость смены работников на физически тяжелых, вредоносных, также небезопасных для здоровья человека, однообразных (изнурительных) технологических операциях; недостатком рабочей силы для исполнения конкретных операций (работ);

- экономической эффективностью при смене работников промышленными роботами;

- потребностью механизации производственных действий (сменой ручного труда) при создании автоматизированных систем управления производственными "процессами; социальным фактором - ликвидацией отличия между интеллектуальным и физическим трудом [16].

Анализ финансовой производительности проекта:

$$\mathcal{E} = \mathcal{E}_r - K \times E, \quad (4.19)$$

где \mathcal{E} – ожидаемый годовой экономический эффект, тг;

\mathcal{E}_r – условная экономия от внедрения проекта, тг,

K – капитальные вложения, тг;

E – нормативный коэффициент экономической эффективности капитальных вложений, который определяется по формуле:

$$E = \frac{1}{T_n}, \quad (4.20)$$

где T_n – нормативный срок окупаемости капитальных вложений, который составляет 5 лет.

Таким образом, согласно формуле 4.20:

$$E = \frac{1}{5} = 2$$

Условная экономия от внедрения проекта находится по формуле:

$$\mathcal{E}_r = \mathcal{Z}_6 - \mathcal{Z}_в \quad (4.21)$$

где \mathcal{Z}_6 – годовые расходы на зарплаты работникам.

$\mathcal{Z}_в$ – годовые расходы на обслуживание манипуляторов.

Путем автоматизации производства будет освобождено место работника сборочного цеха. Средняя зарплата работника механообрабатывающего и механосборочных цехов составляет 47000 (тг) в месяц либо 564000 (тг) в год.

Расходы на использование (содержание) манипулятора с системой технического зрения при односменной работе составляет 180000 (тг) в год.

Подобным способом, согласно формуле 4.21:

$$\mathcal{E}_r = 564000 - 180000 = 384000 \text{ (тг)}.$$

Опираясь на формулу 4.19 ожидаемый, экономический эффект (\mathcal{E}) составит 384000 (тг).

Коэффициент экономической эффективности капитальных затрат рассчитывается по формуле:

$$E_p = \frac{\mathcal{E}_e}{K}, \quad (4.22)$$

где E_p – коэффициент экономической эффективности капитальных затрат;

\mathcal{E}_e – ожидаемая условно-годовая экономия, тг;

K – капитальные вложения на создание системы, тг.

$$E_p = 384000 / 455698,77 * 100\% = 84.3\%$$

Срок окупаемости капитальных вложений рассчитывается по формуле:

$$T_p = \frac{1}{E_p}, \text{ (год)}. \quad (4.23)$$

И составляет:

$$T_p = \frac{1}{0.843} = 1.2, \text{ (год)}.$$

Сводные результаты расчета экономической эффективности проекта представлены в таблице 4.7

Таблица 4.6 – Показатели экономической эффективности внедрения системы технического зрения манипулятора

Показатели	
Капитальные инвестиции, тг	455698.77
Условная экономия, тг	384000
Экономическая эффективность, %	84.3%
Срок окупаемости, год	1.2

Рассчитаем современные показатели экономической эффективности в таблице 4.7:

Таблица 4.7 – Современные показатели экономической эффективности

Показатели	0	1 год	2 год
1. Исходная инвестиция	455698,77		
2. Входной денежный поток		384000	384000
3. Коэффициент дисконтирования	1	0,91	0,83
4. Чистая текущая стоимость (PV)	-455698,77	349090,91	317355,37
5. Чистый дисконтированный доход (NPV)	-455698,77	-106607,86	210747,51

Чистый поток денежных средств (CF) = прибыль – затраты на содержание = 564000 - 180000 = 384000 (тенге).

Коэффициент дисконтирования рассчитывается по годам:

Коэффициент дисконтирования = $1/(1 + \text{норма дисконта})^1 = 0,9$; с каждым последующим годом степень увеличивается на 1, то есть за второй год вторая степень, за третий – третья.

Чистая текущая стоимость (PV) = чистый поток денежных средств * коэффициент дисконтирования. Например, для 1-го года:

$$PV_1 = 384000 * 0,9 = 349090,9 \text{ (тенге).}$$

Чистый дисконтированный доход (NPV) = сумма дисконтированных потоков – инвестиционные затраты. По таблице видим, что $NPV > 0$ и он равняется 210747,51.

Находим индекс рентабельности (PI):

$$PI = 455698,77 + 210747,51 / 455698,77 = 1,46$$

Далее, находим дисконтированный период окупаемости (DPP). Для этого смотрим какой год соответствует последнему отрицательному значению, в моем случае это - 1 год, это означает 1 полный год. Для того, чтобы найти количество месяцев нам необходимо взять невозмещенную стоимость начала периода окупаемости по модулю это 106607 и разделить на денежный поток периода окупаемости 317355,37 и чтобы перевести все это в месяцы умножаем на 12:

$$DPP = 106607 / 317355,37 * 12 = 4.03, \text{ значит } 1 \text{ год } 4 \text{ месяца}$$

Находим внутреннюю норму доходности (IRR), она определяется как процентная ставка, при которой $NPV = 0$, поэтому с помощью функций в MS Excel переходим в раздел Данные - Анализ “что если” - Подбор параметров и устанавливаем значение 0. В итоге получаем 0 значение NPV при $IRR = 43\%$.

Вывод

Исследование экономических показателей подтверждает, что основная прибыль выходит за счет экономии живого труда, т.е. заработной платы работников, освобожденных в ходе роботизации.

Введение системы технического зрения манипуляторов в литейном, кузнечнопрессовом, сварочном также металлургическом производствах еще более рационально и экономически эффективнее, нежели в машиностроительном производстве, таким образом, как заработная плата работников в горячих цехах больше, нежели в механообрабатывающих и механосборочных. Помимо этого, станут более весомы социальные и прочие факторы, оказывающие воздействие на целесообразность введения технического зрения.

В следствии, роботизация становится основным инновационным процессом нынешней экономики, повышая не только лишь объемы изготовления цивилизованных государств, но также снижая расходы при производстве продукции [19].

Из расчетов становится очевидно, что использование роботов в производстве – это весьма экономически эффективное решение, которое исключает вероятность риска и погрешности.

5 Безопасность жизнедеятельности

5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал

С каждым годом интенсивность использования компьютерных технологий в сферах жизни человека возрастает. Работая на ПК, организм человека может столкнуться с некоторыми опасными факторами производства: излучением, шумом и вибрацией, статическим электричеством и т.д. При работе с компьютером человек испытывает сильное умственное и глазное напряжение.

5.1.1 Микроклимат

Комфортные условия труда являются одним из важнейших факторов качественной работы. Если на рабочем месте нарушается микроклимат, человеческий организм подвергается риску многих заболеваний и травм.

Таблица 5.1 – Параметры микроклимата рабочего места

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Категория 1а	23-25	40-60	0,1
Теплый	Категория 1а	20-22	40-60	0,1

В зимнее время в помещении предусмотрена система отопления, которая поддерживает оптимальную рабочую температуру. В соответствии с характеристикой помещения определен расход свежего воздуха согласно [17] и приведен в Таблице 5.2.

Таблица 5.2 – Нормы подачи свежего воздуха в помещения, где расположены компьютеры

Характеристика помещения	Объемный расход подаваемого в помещение свежего воздуха
Объем до 20 м ³ на человека 20...40 м ³ на человека	Не менее 30 Не менее 20

5.1.2 Естественная освещенность

При плохом освещении рабочего помещения способность к работе снижается, организм быстро устает, что приводит к большому количеству недостатков и ошибок.

Рабочая область или рабочее место разработчика освещается таким образом, что вы можете четко видеть рабочий процесс, не напрягая глаза, а

также прямое воздействие источника света на глаза. Кроме того, уровень необходимого освещения зависит от степени точности зрительной работы. Наименьший размер объекта дискриминации составляет 0,5 - 1 мм. В комнате нет естественного света.

Таблица 5.3 – Нормирование освещённости для работы за ПК [2]

1	2	3	4	5	6	7	8	9	10	11
Высокой точности	От 0,3	Б	1	Более 70	300	100	40	15	3,0	1,0
	От 0,5		2	Менее 70	200	75	60	20	2,5	0,7

Где:

- 1 - характеристика зрительных работ;
 - 2 - наименьший или эквивалентный размер объекта различения, мм;
 - 3 - разряд зрительной работы;
 - 4 - подразряд зрительной работы;
 - 5 - относительная продолжительность зрительной работы, %;
 - 6 - освещенность на рабочей поверхности от системы общего искусственного освещения, лк;
 - 7 - цилиндрическая освещенность, лк;
 - 8 - показатель дискомфорта;
 - 9 - коэффициент пульсации освещенности, %;
 - 10 - КЕО при верхнем освещении, %;
 - 11 - КЕО при боковом освещении, %
- Требования к освещению на рабочих местах, оборудованных ПК, представлены в таблице 5.4. [19]

Таблица 5.4 – Требования к освещению на рабочих местах, оборудованных ПК [19]

Освещенность на рабочем столе	300-500 лк
Освещенность на экране ПК	не выше 300 лк
Блики на экране	не выше 40 кд/м ²
Прямая блескость источника света	200 кд/м ²
Показатель ослеплённости	не более 20
Показатель дискомфорта	не более 15
Отношение яркости:	

Продолжение таблицы 5.4

– между рабочими поверхностями	3:1–5:1
– между поверхностями стен и оборудования	10:1
Коэффициент пульсации:	не более 5%

Освещаемая площадь помещения составляет 160 м², количество ламп 24, каждая 2500 люмен. Тогда $E=60000/160=375$ лк. Это является достаточным для выполнения условий, приведенных в таблице 5.4.

5.1.3 Уровень шума

Работающие в условиях длительного шумового воздействия испытывают стресс, мигрень, сонливость, неспособность к запоминанию, отсутствие аппетита и т. д. Под воздействием шума снижается концентрация внимания, нарушаются физиологические функции, появляется усталость в связи с повышенными затратами энергии и нервно- психическим напряжением, ухудшается речевая коммутация. Длительное воздействие интенсивного шума (больше 80 дБ) на слух человека приводит к его частичной или полной потере.

Допустимые уровни звука на рабочем месте за ПК, приведены в [20].

Для устранения шума необходимо прочистить вентилятор от пыли или заменить полностью.

5.1.4 Уровень электромагнитных излучений

Энергетический эффект электромагнитного излучения может варьироваться по силе и интенсивности. Не воспринимается человеком до ощущения тепла с высокоэффективным излучением, но подобный эффект можно увидеть, только если превышен допустимый уровень.

В дополнение к электромагнитному излучению во время работы компьютера создается электростатическое поле, которое может деионизировать окружающий воздух. Компьютер долго нагревается, что не только слабо ионизирует воздух, но и делает его сухим. Этот воздух очень вреден, трудно дышать и в благоприятных условиях способствует развитию многих аллергических заболеваний и, соответственно, респираторных заболеваний.

Энергетическая экспозиция за рабочий день не должна превышать значений [21], указанных в таблице 5.5.

Таблица 5.5 – Предельно допустимые значения энергетической экспозиции

	Предельно допустимая энергетическая экспозиция		
Диапазоны частот	По электрической составляющей,	По магнитной составляющей,	По плотности потока энергии

Продолжение таблицы 5.5

30 кГц - 3 МГц	20000,0	200,0	х
3 - 30 МГц	7000,0	Не разработаны	х
30 - 50 МГц	800,0	0,72	х
50 - 300 МГц	800,0	Не разработаны	х
300 МГц – 300 ГГц	х	х	200,0

Для обеспечения меньшего электромагнитного излучения используется жидкокристаллический монитор. Необходимо учитывать расстояние до монитора, поскольку при большем расстоянии от человека влияние оказывается меньше. Из-за того, что электромагнитное излучение от стенок монитора намного больше, необходимо ограничить его стенками, т.е. поставить в угол.

5.1.5 Электробезопасность

Даже самая надежная изоляция теряет свои качества при длительном использовании в сложных производственных условиях [22]. Риск поражения электрическим током значительно возрастает, если человек находится в комнате с железным полом или вступает в контакт с оборудованием, расположенным рядом с заземленными металлическими предметами. Согласно правилам устройства электроустановок, производственные цеха, коммерческие и жилые здания, как правило, делятся на 3 отдельных класса, которые определяют уровень угрозы. Давайте рассмотрим каждый из них в отдельности.

Первый класс - "помещения без повышенной опасности". К этой группе относятся здания с низкой влажностью, которые оснащены системой вентиляции и отопления при необходимости.

Кроме того, полы в таких помещениях не должны быть токопроводящими. Под концепцию токопроводящих полов предлагаем металл, железобетон, земля и т.д. Чтобы классифицировать производственный цех как безвредный, его пол должен быть покрыт диэлектрическим материалом. Очевидными примерами безвредных объектов являются простые жилые и офисные здания, помещения для хранения инструментов, лаборатории и производственные цеха производителей инструментов, конструкция которых изначально предусматривала изолированный пол, прочные воздушные фильтры для удаления пыли и систему контроля температуры воздуха.

Второй класс - "помещения с повышенной опасностью". Действующая классификация помещений на предмет опасности поражения электрическим

током устанавливает структуру группы риска при наличии одного из следующих условий:

- влажность;
- проводящая пыль;
- здания с токопроводящими полами;
- высокая температура;
- условия, при которых человек может одновременно касаться металлических корпусов электрооборудования и заземленных металлических конструкций зданий.

Третий класс - "особо опасные помещения". К самой опасной категории относятся среды с высоким уровнем влажности, концентрированная взвесь химически активных веществ в воздухе, а также как минимум два других фактора по сравнению с категорией помещений с наибольшей опасностью.

- помещения с «особой влажностью»;
- помещения, в которых наблюдается химическая активность и органическая среда;
- два или более условия из пункта 2;

Электробезопасность - система организационно-технических мероприятий и средств защиты людей от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества.

При работе с компьютером существует опасность электропоражения:

- при прикосновении к нетоковедущим частям, оказавшимся под напряжением;
- при прикосновении с полом, стенами, оказавшимися под напряжением;
- имеется опасность короткого замыкания в высоковольтных блоках: блоке питания и блоке дисплейной развертки.

В зависимости от условий в помещении риск поражения электрическим током увеличивается или уменьшается. По [22] помещение, в котором находится рабочее место, относится к категории помещений без повышенной опасности. Он может быть описан как сухой, без пыли, с непроводящими полами и нормальной температурой воздуха. Температура, влажность, химическая среда не способствуют разрушению изоляции электрооборудования.

Безопасность при работе с электроустановками обеспечивается применением различных технических и организационных мер. Основные технические средства защиты от поражения электрическим током:

- изоляция токопроводящих частей и ее непрерывный контроль;
- установка оградительных устройств;
- отдельные помещения для манипуляторов;
- предупредительная сигнализация и блокировки;
- защитное заземление;
- зануление;
- защитное отключение по [23].

5.2 Расчёт искусственного освещения офисного помещения

Исходные данные:

- Помещение $a = 9$ м, $b = 6$ м, $h = 3,2$ м,
- Светильник растровый встраиваемый на 4 люминесцентные лампы 18 Вт тип ARS/R 4x18 W,
- Лампы люминесцентные 18 Вт, в одном встраиваемом растровом светильнике 4 лампы $\Phi = 1150$ лм (для люминесцентной лампы производства Philips TLD 18/54,
- Нормы освещенности $E = 300$ лк на уровне 0,8 м от пола (рабочая поверхность стола),
- Коэффициент запаса $K_z = 1,25$,
- Коэффициент отражения потолка - 50, стен - 30, пол - 10.

Расчётные формулы:

- Определение площади помещения:

$$S = a * b \quad (5.1)$$

- Определение индекса помещения:

-

$$i = S / ((h_1 - h_2) * (a + b)) \quad (5.2)$$

- Определение нужного количества светильников:

$$N = E * S * 100 * K_z / U * n * \Phi_{\text{л}}, \quad (5.3)$$

Где:

E – требуемая освещенность горизонтальной плоскости, лк;

S – площадь помещения, м.кв;

K_z – коэффициент запаса;

U – коэффициент использования осветительной установки;

$\Phi_{\text{л}}$ – световой поток одной лампы, лм;

n – число ламп в одном светильнике.

На рисунке 5.1 изображён офис искусственным освещением, которое нужно рассчитать.



Рисунок 5.1 – Офис с искусственным освещением

1. Для начала нужно определить площадь помещения:

$$S = 9 \cdot 6 = 54 \text{ м. кв.},$$

2. Затем определим индекс помещения:

$$i = \frac{5}{((3,2-0,8) \cdot 15)} = 1,5$$

3. Также определим коэффициент использования, исходя из значений коэффициентов отражения и индекса помещения по рисунку 5.3 $U = 51$.

4. Расчёт требуемого количества светильников:

$$N = \frac{(300 \cdot 54 \cdot 100 \cdot 1,25)}{(51 \cdot 4 \cdot 1150)} = 8,63 \sim 9 \text{ светильников}$$

5.3 Расчёт защитного заземления

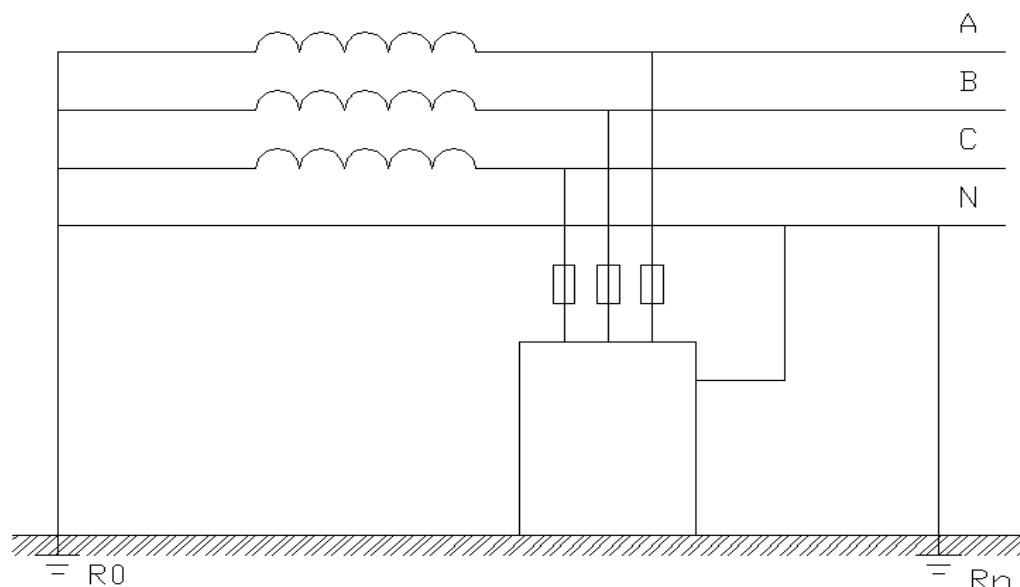


Рисунок 5.4 – Схема защитного заземления

Заземление обеспечивается использованием нулевого защитного проводника с полной проводимостью 50% от полной проводимости фазового провода, что позволяет получить ток короткого замыкания, который по крайней мере в три раза больше номинального.

Нулевой защитный проводник на входе в здание и через каждые последующие 100-150 м заземляется.

Полное сопротивление контура нулевой фазы (состоящего из собственного сопротивления контура и сопротивления одной из фаз трансформатора питания) должно обеспечивать равный или больший ток короткого замыкания в аварийной ситуации. в три раза больше номинального тока ($I_{кз} \geq K I_{ном}$).

Рассчитаем сопротивление фазы нулевой петли.

Сопротивление контура нулевой фазы имеет решающее значение для выбора теплового предохранителя или автоматического выключателя для тока короткого замыкания.

Полное сопротивление петли нулевой фазовой петли рассчитывается по формуле:

$$Z_{фн} = Z_{тр} + \sqrt{(R_{ф} + R_{н})^2 + X_{п}^2}, \quad (5.4)$$

где $X_{п}$ - индуктивное сопротивление петли “фаза-нуль”, принимаем равным 0.6 Ом/м;

$Z_{тр}$ – сопротивление входного трансформатора, равное 3.1 Ом;

R_{ϕ}, R_n - активное сопротивление фазного и нулевого провода

$$R_{\phi}; R_n = \rho_{\pi} * l_{\pi} / S_{\pi}, \quad (5.5)$$

где ρ_{π} - удельное сопротивление проводника, равное 0,017

l_{π} - длина проводника, равная 50 м;

S_{π} - сечение проводника, равное 2.5 мм².

$$R_{\phi}; R_n = \rho_{\pi} \frac{l_{\pi}}{S_{\pi}} = 17.5 * 10^{-3} * \frac{50}{2.5} = 0.35 \text{ Ом}$$

$$Z_{\phi n} = Z_{\text{тр}} + \sqrt{(R_{\phi} + R_n)^2 + X_{\Pi}^2} = x + \sqrt{(0.02 + 0.02)^2 + 0.6^2} = 4.02 \text{ Ом}$$

Произведем выбор номинального значения тока срабатывания предохранителя. Для того, чтобы заземление выполняло свои функции, необходимо

$$I_{\text{кз}} \geq K I_{\text{ном}}, \quad (5.6)$$

K - коэффициент кратности, принимается для тепловых автоматов и плавких вставок равным 3;

$I_{\text{кз}}$ определяем из отношения:

$$I_{\text{кз}} = U_{\phi} / Z_{\phi n}, \quad (5.7)$$

Где U_{ϕ} - фазное напряжение, равное В;

$Z_{\phi n}$ - сопротивление петли “фаза-нуль”;

$$I_{\text{кз}} = 220 / 4.02 = 54 \text{ А}$$

Так как максимальный ток нагрузки не превышает 10А, в качестве защиты используем плавкий предохранитель с номинальным током 15А.

Заключение

В результате выполнения дипломной работы был приобретён опыт анализа устройства манипуляторов, их виды и специализация. Также были получены навыки разработки программного обеспечения, создания симуляции кинематических движений звеньев робота в среде разработки Processing и проектирования составляющих манипулятора в параметрической САПР программе FreeCAD. В результате выполнения проекта были разработаны структурная, принципиальная электрическая схема. Был проведён расчёт экономических затрат, экономической эффективности данной разработки и разработку программного обеспечения. В результате технико-экономического расчёта можно сделать вывод, что время окупаемости вложений в систему технического зрения составит 1,2 года, а это означает, что использование данной системы не только практично, но и экономически выгодно. Путем автоматизации разных видов производств высвобождаются некоторые рабочие роли, но в то же время создаются новые рабочие места в большем количестве, что непременно положительно скажется и на экономике страны. Также были изучены некоторые экологически важные факторы при работе с установкой в помещении, а именно: микроклимат, важность освещения, пожаробезопасность и другие. Был произведён расчет искусственного освещения в офисном помещении и защитного заземления.

Список сокращений

USB –Universal Serial Bus (универсальная последовательная шина)

ПО – программное обеспечение

RGB - аббревиатура английских слов Red, Green, Blue — красный, зелёный, синий.

СМΥК - Cyan, Magenta, Yellow, Key color - субтрактивная (subtract, англ. - вычитать) схема формирования цвета.

Lab - значение светлоты отделено от значения хроматической составляющей цвета.

АСУ – автоматизированная система управления.

ШИМ – широтно-импульсная модуляция.

ПК – персональный компьютер.

HSB - модель, которая в принципе является аналогом RGB, она основана на её цветах, но отличается системой координат.

НИР – научно-исследовательская работа

ФОТ – фонд оплаты труда

НДС – налог на добавленную стоимость

Список литературы

1. <https://kurs.kz/> (дата обращения – 13.01.2020)
2. <https://volmax.kz> (дата обращения 17.01.2020)
3. <http://ba3ar.kz> (дата обращения – 06.04.2020)
4. <https://smartshop.kz/catalog> (дата обращения – 06.04.2020)
5. <https://office-expert.kz> (дата обращения – 06.04.2020)
6. <https://deltachip.kz/> (дата обращения – 08.05.2020)
7. <https://habr.com/ru/post/418339/> (дата обращения – 10.01.2020)
8. <https://ru.wikipedia.org> (дата обращения – 28.05.2020)
9. <https://habr.com/ru/post/422019/> (дата обращения – 11.01.2020)
10. <https://www.processing.org/> (дата обращения – 20.01.2020)
11. Э.А. Киреева, С.Н. Шерстне; Под общ. ред. С.Н. Шерстнева Полный справочник по электрооборудованию и электротехнике: КНОРУС, 2013 с 21-24.
12. Свешников А.М., Кириллов П.П. Проектирование ШИМ контроллера// Радио - 2012, № 12, с. 43
13. <https://www.freecadweb.org> (дата обращения – 23.03.2020)
14. Г.Ш. Боканова. Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В071900 – Радиотехника, электроника и телекоммуникаций. – Алматы: АУЭС, 2020 – 26 с. (дата обращения – 20.05.2020)
15. Алматы ЭнергоСбыт // [<https://esalmaty.kz/ru/?region=almaty>] / «Тарифы для юридических лиц» (дата обращения – 6.02.2020)
16. <http://ciframagazine.com/> (дата обращения – 06.02.2020)
17. СанПиН «Гигиенические требования к микроклимату производственных помещений», утвержденные МЗ РК 14.07.2005 г. № 355.
18. СНиП РК 2.04-05-2002 Естественное и искусственное освещение.
19. СН 2.2.4/ 2.1.8.562-96 Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки.
20. СанПиН 2.2.1/2.1.1.1278-03 Гигиенические требования к естественному, искусственному и совмещённому освещению жилых и общественных зданий.
21. РК ГОСТ р МЭК 60950-2005 Безопасность оборудования информационной технологии, включая электрическое контрольное оборудование.
22. СанПиН № 1.01.004.01 Санитарно-эпидемиологические правила и нормативы «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.
23. «Правил пожарной безопасности в Республики Казахстан» (ППБ РК № 1682-2011).

Приложение

```
float c = 0.0; //вылет
float h = 80; //высота основания
float L1 = h*3; //плечо
float L2 = h*4; //предплечье

float alfa;
float Beta,Beta2;
float gamma;
float tau,tau2;

void setup(){
    size(800,600,P3D);
}

void draw(){

    //Очищаем окно
    background(#463F3F);

    //Выводим значения углов
    textSize(14);
    fill(#26B41F);
    text("Tau, deg"+" "+(tau2),50,0.8*height);
    text("Beta, deg"+" "+(Beta2),50,0.8*height+20);
    text("Alfa, deg"+" "+degrees(alfa),50,0.8*height+40);
    text("Gamma, deg"+" "+degrees(gamma),50,0.8*height+60);

    drawTarget();

    //Перенос системы координат
    //и поворот

    translate(width/4,height/2+height/4);
    rotateX(radians(180));

    /* Рассчитываем положение - углы*/
    computeRobotpos();

    // Основание
    drawBase();

    // Плечо
    drawArm();

    // Предплечье
    drawForeArm();
}

void computeRobotpos(){

    //Вылет
    float c = (mouseX-width/4);
    float R = sqrt(h*h + c*c);
    println(R);
    alfa = asin(h/R);

    Beta = acos( (L1*L1 + L2*L2 - R*R) / (2*L1*L2));

    gamma = acos( (L1*L1 + R*R - L2*L2) / (2*L1*R));

    tau = gamma - alfa;

    tau2 = 90 + degrees(alfa - gamma);
    Beta2 = degrees(Beta) - 90;
}

void drawlocalCS(){
```

```

        stroke(255,255,0);
        line(0,0,0,h);
        strokeWeight(15);
        point(0,h);
        drawlocalCS();
    }

    void drawArm(){
        //переносим систему координат
        //в точку начала звена
        translate(0,h);
        rotateZ((tau));

        strokeWeight(3);
        stroke(255,255,0);
        line(0,0,L1,0);
        strokeWeight(15);
        point(L1,0);
        drawlocalCS();
    }

    void drawForeArm(){
        //переносим систему координат
        //в точку начала звена
        translate(L1,0);
        rotateZ(-(PI-Beta));

        strokeWeight(3);
        stroke(255,255,0);
        line(0,0,L2,0);
        strokeWeight(15);
        point(L2,0);
        drawlocalCS();
    }
}

```



```

void setup() {

    size(320, 240);
    //size(640, 480);
    String[] cameras = Capture.list();

    if (cameras == null) {
        println("Failed to retrieve the list of available cameras, will try the default...");
        cam = new Capture(this, w, hh);
    } if (cameras.length == 0) {
        println("There are no cameras available for capture.");
        exit();
    } else {
        println("Available cameras:");
        printArray(cameras);

        //Интерфейс камеры с нужным разрешением(320x240)
        // cam = new Capture(this, cameras[9]);
        cam = new Capture(this, cameras[12]);
        cam.start();
    }

    port = new Serial(this, "COM21", 9600);
    port.bufferUntil('\n');
}

void draw() {
    //Захватываем изображение с камеры
    if (cam.available() == true) {
        cam.read();
    }

    //Выводим изображение в окно
    image(cam, 0, 0, width, height);

    image(cam, 0, 0, width, height);

    //Обрабатываем изображение и ищем нужные пиксели
    imageProcess();

    //Получение ЦМ фигуры
    getCenter();

    //Перенос СК из вернего правого угла к центру робота
    Xt = cXmm-80;
    Yt = -cYmm+170;
    fill(230,0,0);
    text("Xt"+" "+Xt,width-50,60);
    text("Yt"+" "+Yt,width-50,80);

    //Вылет робота – размер рабочей зоны от 90 до 165 мм*/
    c = sqrt(Xt*Xt+Yt*Yt);

    //Расчет угла в плане
    if(c<=160){
        planeAng = degrees( asin( Yt/sqrt(c*c) ) );
        if (Xt<0) planeAng = 90 + degrees( asin(-Xt/(sqrt(c*c)))));
    }
    planeAng = constrain(planeAng,25,160);
    println(planeAng+" "+c);

    if ( (c<160) && (c>90)){
        computeRobotpos();
        sendPacket();
    }

    //Запись кадров
    //saveFrame("frames/####.tiff");
}

```

```

//Обработчик изображения
void imageProcess(){
    //Пробегаем по всем пикселям изображения
    ii=0; sumX=0.0; sumY=0.0;
    for (int i=0;i<w;i+=1){
        for(int j=0; j<hh;j+=1){
            //получаем цвет текущего пикселя
            color cc = get(i,j);
            //и переводим его в HSB-пространство
            float hue = hue(cc);
            float s = saturation(cc);
            float br = brightness(cc);
            //Сравниваем тек. пиксель с выбранным с учетом
            //некоторого допуска на отличие в цвете
            if (hue<=(huesel+15) && hue>=(huesel-15) &&
                s<=(satsel+15) && s>=(satsel-15) &&
                br<=(brsel+150) && br>=(brsel-150))
            {
                //добавляем координату к общей сумме
                sumX+=i;
                sumY+=j;
                ii++;
                //делаем найденный пиксель красным
                color cch = color(255,0,0);
                set(i,j,cch); //set(i,j-1,cch);set(i-1,j-1,cch);
            }
        }
    }

    //Получение ЦМ фигуры
    void getCenter(){
        //вычисляем центр фигуры из найденных пикселей

```

```

void sendPacket(){
    /* Посылка пакета с углами поворота приводов*/
    byte a = 111;
    byte[] data = {(byte) a, (byte) map(planeAng,0,180,0,128),
                  (byte) map(tau2,0,180,0,128),
                  (byte) map(Beta2,0,180,0,128)};
    port.write(data);
    //println(data);
}

void computeRobotpos(){
    //Вылет Робота
    float R = sqrt(h*h + (c+r)*(c+r));
    //println(R);
    alfa = asin(h/R);

    Beta = acos( (L1*L1 + L2*L2 - R*R) / (2*L1*L2));

    gamma = acos( (L1*L1 + R*R - L2*L2) / (2*L1*R));

    tau = gamma - alfa;

    tau2 = 90 + degrees(alfa - gamma);
    Beta2 = degrees(Beta) - 90;

    tau2 = constrain(tau2,0,180);
    Beta2 = constrain(Beta2,0,180);
}

```

```

#include <Servo.h>

Servo servo1;
Servo servo2;
Servo servo3;

int servo1Pos;
int servo2Pos;
int servo3Pos;

/* Углы поворота Серв, взятые из пришедшего пакета от ПК */
byte angA=90;
byte angB=90;
byte angC=90;

/* Переменные для обработки и хранения данных пришедшего пакета*/
byte flag=0; //Состояние 1 - дописываем в пакет //Состояние 2 - ищем заголовок
byte packetSize=4; //Кол-во байт в принимаемом пакете
byte c = 0; //счетчик принятых байтов в пакете
byte packet[8]; //пакет с принимаемыми байтами
boolean packetComplete = false; //

void setup()
{
  Serial.begin(9600);
  servo1.attach(4); // attaches the servo on pin 9 to the servo object
  servo2.attach(5);
}

void loop()
{
  /*Обрабатываем входящий байт и формируем пакет*/
  serialEvent();

  if (packetComplete) {

    //Забираем из пакета угол для 1-го Сервы
    angA = map(packet[1],0,128,0,180);
    angA = constrain(angA,0,180);
    //Забираем из пакета угол для 2-го Сервы
    angB = map(packet[2],0,128,0,180);
    angB = constrain(angB,0,180);
    //Забираем из пакета угол для 3-го Сервы
    angC = map(packet[3],0,128,0,180);
    angC = constrain(angC,0,180);

    /* Поворачиваем Сервы на соответствующие углы */
    rot( angA, servo1);
    rot( angB, servo2);
    rot( angC, servo3);

    packetComplete = false;
  }
  printPacket();
}

```

```

void printPacket() {
    Serial.print(packet[0]);Serial.print(',');
    Serial.print(packet[1]);Serial.print(',');
    Serial.print(packet[2]);Serial.print(',');
    Serial.println(packet[3]);
}

void serialEvent() {
    while (Serial.available()) {
        // get the new byte:
        byte inByte = (byte)Serial.read();

        //1 - Продолжаем получать пакет
        if (flag==1){
            packet[c] = inByte;
            c++;

            if (c == packetSize){
                packetComplete = true;
                flag=0;
            }

            //2 - Поиск заголовка
        } else if (inByte == 111){
            c=0;
            flag=1;
            packet[c] = inByte;
            c++;
        }
    }
}

```

```

        rot( 90, servo1);

rot( 90, servo2);
rot( 45, servo2);
rot( 90, servo2);

rot( 90, servo3);
rot( 45, servo3);
rot( 90, servo3);
}

void rot(int posTo, Servo servo){

    int posFrom = servo.read();

    if (posTo > posFrom){
        while(posTo >= servo.read() ){
            servo.write(posFrom++);
            delay(10);
            Serial.println(servo.read());
        }
    }

    if (posTo < posFrom){
        while(posTo <= servo.read() ){
            servo.write(posFrom--);
            delay(10);
            Serial.println(servo.read());

```
