

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Систем автоматики, автоматизированного управления и
проектирования»

УТВЕРЖДАЮ
Заведующий кафедрой

_____ С.В. Ченцов

« ____ » _____ 2021 г.

БАКАЛАВРСКАЯ РАБОТА

15.03.04 Автоматизация технологических процессов и производств

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ
ТРАНСПОРТИРОВКОЙ ГОРНОЙ МАССЫ**

Руководитель	_____	___06. 2021 г.	доцент, канд. техн. наук А. В. Чубарь
Выпускник	_____	___06. 2021 г.	В. А. Сергеев
Нормоконтролер	_____	___06. 2021 г.	Т.А. Грудинова

Красноярск 2021

РЕФЕРАТ

Выпускная квалификационная работа по теме «Автоматизированная система управления транспортировкой горной массы» содержит 83 страниц текстового документа, 10 приложений, 20 использованных источников.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА, СРЕДСТВА
ТРАНСПОРТИРОВКИ, ГОРНАЯ МАССА, СИСТЕМА УПРАВЛЕНИЯ,
SIMINTECH, МОДЕЛЬ.

Объектом разработки является имитационная модель средств транспортировки руды или породы с горизонта загрузки на склад руды или породы.

Цель работы: создание имитационной модели объекта автоматизации и программы управления и контроля параметров для работы с ней.

Задачи, решаемые в ходе выполнения данной бакалаврской работы:

- 1) Изучение объекта разработки;
- 2) Описание используемых средств автоматизации;
- 3) Создание имитационной модели;
- 4) Разработка программ управления и контроля параметров.

В ходе выпускной квалификационной работы был изучен технологический процесс транспортировки горной массы из шахты на поверхность, описаны используемые средства транспортировки, разработана имитационная модель автоматизированной системы управления транспортировкой горной массы, алгоритмы управления и человеко-машинный интерфейс в среде динамического моделирования *SimInTech*.

СОДЕРЖАНИЕ

Введение.....	4
1 Автоматизированная система управления транспортировкой горной массы ...	6
1.1 Описание объекта разработки.....	6
1.2 Описание средств транспортировки	9
1.3 Оборудование АСУТП	16
2 Автоматизация объекта	20
2.1 Перечень сигналов	20
2.2 Структурно-функциональная анализ	22
2.3 Алгоритмы управления	24
3 Создание имитационной модели	28
3.1 Выбор и обоснование среды разработки	28
3.2 Требование к модели	29
3.3 Реализация модели	30
3.4 Составление базы данных	32
3.5 Моделирование схем общего вида	35
Заключение	48
Список использованных источников	49
Приложение А Скрипт для управления загрузкой	52
Приложение Б Скрипт для управления подъемной установкой	54
Приложение В Скрипт для управления разгрузкой	58
Приложение Г Скрипт для управления подъемной установкой вручную	60
Приложение Д Скрипт окна анимации «Диспетчер»	61
Приложение Е Скрипт окна анимации «ПУ загрузки»	66
Приложение Ж Скрипт окна анимации «ПУ ПМ»	71
Приложение И Скрипт окна анимации «ПУ разгрузки»	76
Приложение К Скрипт окна анимации «Графики»	82
Приложение Л Функциональная схема	83

ВВЕДЕНИЕ

В настоящее время важным этапом проектирования технологических процессов и их автоматизации является компьютерное моделирование, сущность которого заключается в замене исходного объекта его имитационной моделью и дальнейшем изучении модели с помощью реализуемых на компьютерах вычислительно-логических алгоритмов. Работа с моделью, с одной стороны, дает возможность без существенных материальных и временных затрат исследовать свойства технологического процесса, его характеристики в заданных условиях, реализовывать наборы алгоритмов, отображающих ситуации, возникающие в состоянии моделируемого объекта и изменяющиеся по определенным сценариям [1]. С другой стороны, компьютерная модель может использоваться как средство накопления и хранения знаний об объекте исследования и выполнять обучающую функцию.

АСУТП – автоматизированная система управления технологическими процессами, которая представляет собой совокупность технических и программных средств, позволяющая в автоматическом режиме управлять оборудованием на предприятиях промышленной сферы.

Она может быть составляющим элементом общей системы, обеспечивающей автоматизацию управления предприятием.

Как правило, данная система – это эффективное решение, автоматизирующее операции, как в целом всего производства, так и отдельного участка, который обеспечивает выпуск конкретного продукта.

«Автоматизированное» управление следует отличать от «автоматического», потому что для выполнения конкретных действий необходимо, чтобы участвовал человек, в частности, контролировал ход процесса, а также по той причине, что для некоторых операций крайне сложно и нецелесообразно автоматизировать.

На сегодняшний день автоматизированные системы управления используются практически во всех направлениях промышленной сферы. Сегодня можно разработать систему, которая автоматизирует разные процессы, в частности, выгрузку, прием, хранение, фильтрацию, переработку, измерение, дозировку и др [2].

В данной бакалаврской работе в качестве автоматизированной системы управления технологическими процессами была выбрана автоматизированная система управления транспортировкой горной массы.

Актуальность работы состоит в облегчении разработки, изменения и отладки программ управления и контроля параметров для объекта автоматизации, уменьшении расходов на этапе внедрения и обучении персонала для работы с объектом.

Целью работы является создание имитационной модели объекта автоматизации и программы управления и контроля параметров для работы с ней.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- 1) Изучение объекта автоматизации;
- 2) Описание используемых средств автоматизации;
- 3) Создание имитационной модели;
- 4) Разработка программ управления и контроля параметров.

1 Автоматизированная система управления транспортировкой горной массы

1.1 Описание объекта разработки

Объект моделирования – средства транспортировка руды с горизонта загрузки на склад руды.

Поточно-транспортная система для транспортировки руды (породы) включает систему загрузки скипов, скиповой подъем, систему разгрузки скипов, приемный бункер, узел перегрузки руды и конвейерную галерею.

Система загрузки скипов находится в подземной части рудника. Схему можно увидеть на рисунке 1.1. В состав оборудования входят:

- бункеры, подающие руду на питатели 1 и 2;
- питатели 1 и 2 подающие руду на ленточный конвейер длиной 45 м на горизонте загрузки
- подземные загрузочные устройства – дозаторы 1 и 2, через которые осуществляется загрузка скипов 1 и 2 с ленточного конвейера;

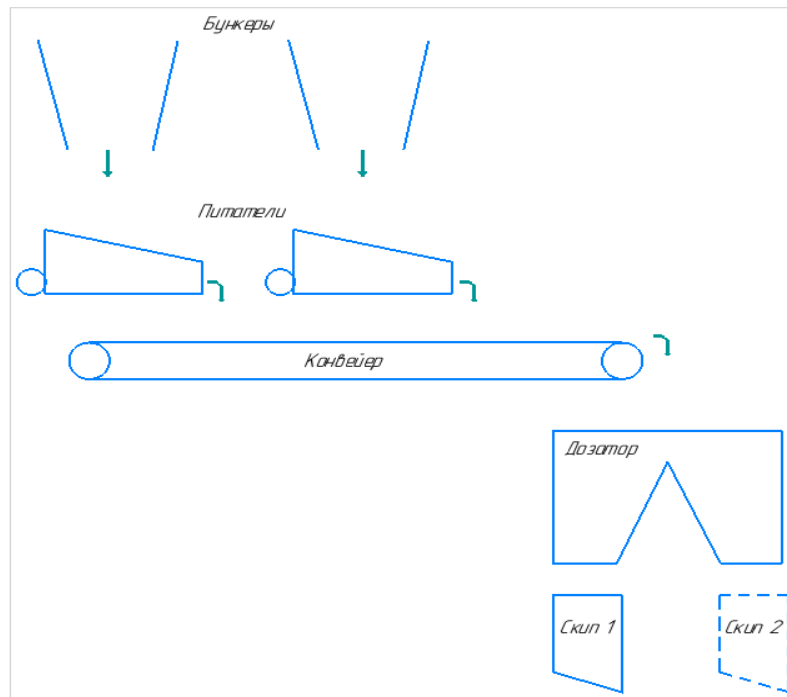


Рисунок 1.1 – Схема подсистемы загрузки

Скиповый подъем состоит из двухскиповой одноканатной подъемной установки [3]. Система управления и автоматизации подъемной машины построена на основе микропроцессорных программируемых контроллеров.

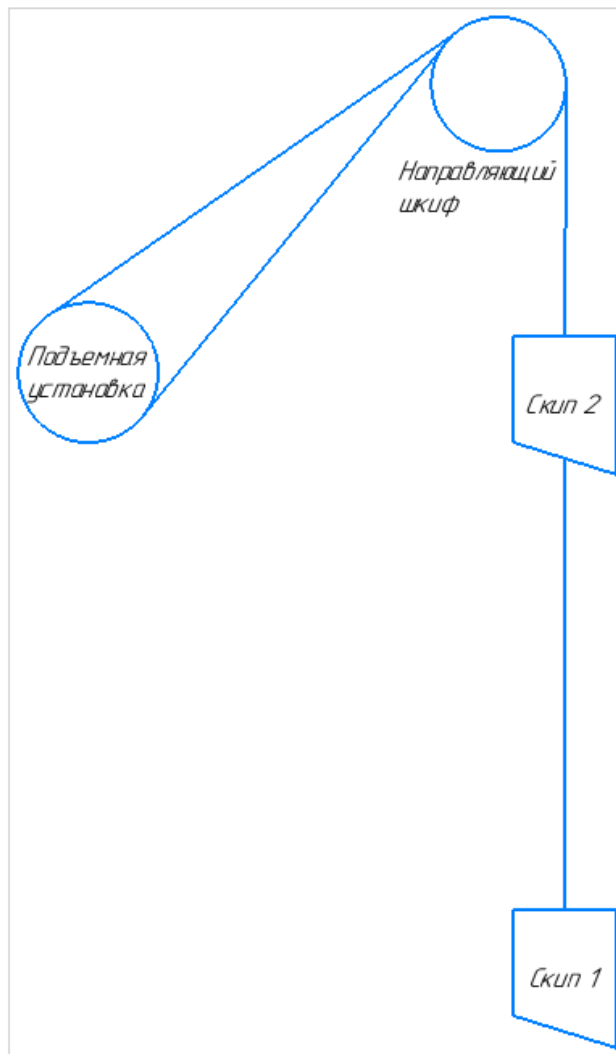


Рисунок 1.2 – Шахтная двухскиповая подъемная установка

Управление технологической автоматикой выполняет контроллер подъемной установки, контроль движения сосудов в стволе – монитор подъемной установки. Управление подъемной установкой выполняет оператор ПМ с центрального пульта управления. Информация к оператору поступает в основном от контроллера на панель пульта и экран монитора персонального компьютера АРМ СУ ПМ

Комплекс разгрузки скипов, расположенный в надшахтном сооружении (копер ВКС) и состоит из:

- узел разгрузки скипов.

- приемный бункер руды.
- питатель копра ВКС, подающий руду (породу) из приемного бункера;
- ленточный конвейер, транспортирующий руду (породу) по транспортной галерее на склад.

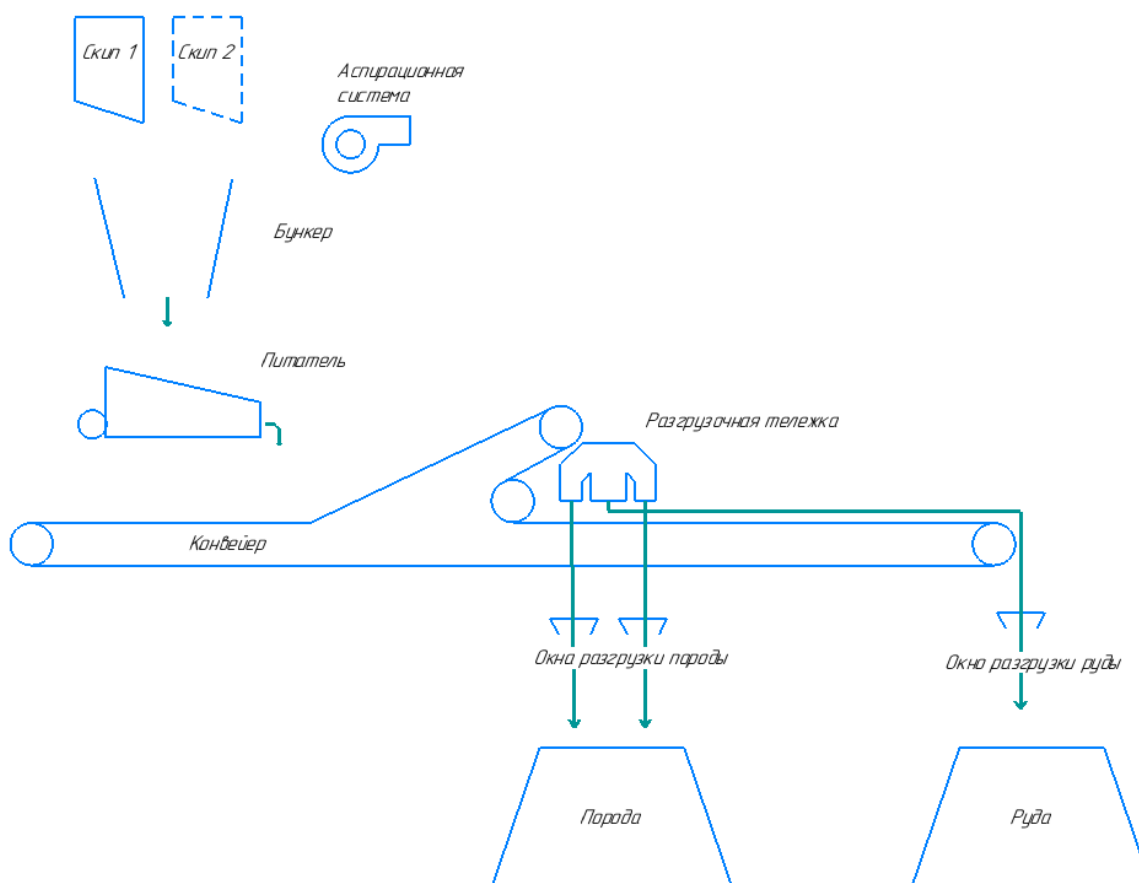


Рисунок 1.3 – Схема подсистемы разгрузки

1.2 Описание средств транспортировки

Создание поточной технологии на всех или на отдельных транспортных звеньях горного предприятия невозможно без применения конвейерного транспорта, обеспечивающего непрерывность грузопотока, ритмичность работы и повышение производительности труда при наиболее низкой энергоемкости и трудоемкости процессов доставки и транспортирования руды [4].

Основными преимуществами конвейерного транспорта являются: непрерывность потока и высокая производительность; большая длина отдельных конвейеров или конвейерных линий и возможность транспортирования под углами 18° ; небольшие поперечные размеры става конвейера, допускающие его установку в горных выработках небольшого сечения; возможность монтажа конвейеров в выработках, пройденных по почве беспокойно залегающих пластов; возможность полной автоматизации конвейерных установок с централизованным управлением; повышение безопасности и улучшение условий труда.

Основные недостатки конвейерного транспорта: сложность доставки абразивных крупнокусковых грузов и необходимость предварительного дробления крепких руд; интенсивный износ гибких грузонесущих органов; высокая стоимость оборудования и его монтажа; необходимость дополнительной транспортной системы для доставки вспомогательных грузов.

В ленточном конвейере транспортирование горной массы осуществляется на конвейерной ленте, выполняющей функции тягового и несущего органа. Замкнутая бесконечная лента 1, представленная на рис. 1.4, а, огибает головной приводной 2 и хвостовой натяжной 3 барабаны. Лента поддерживается по длине конвейера стационарными роlikоопорами 4 и 5, причем расстояние между роlikоопорами для верхней грузовой ветви в 2—2,5 раза меньше, чем для нижней порожней ветви. Загрузка возможна практически в любой точке по длине конвейера. Обычно ленточные конвейеры загружаются в хвостовой части через загрузочную воронку 6, а разгружаются при сходе ленты с головного барабана. Возможна разгрузка ленточного конвейера в промежуточных пунктах с помощью плужковых сбрасывателей или разгрузочных тележек. В зависимости от назначения и условий эксплуатации ленточные конвейеры оснащают дополнительными устройствами для очистки ленты и барабанов, и улавливания ленты в случае ее обрыва (на наклонном конвейере). Для контроля за работой и

автоматизации конвейеров устанавливают различные датчики и приспособления.

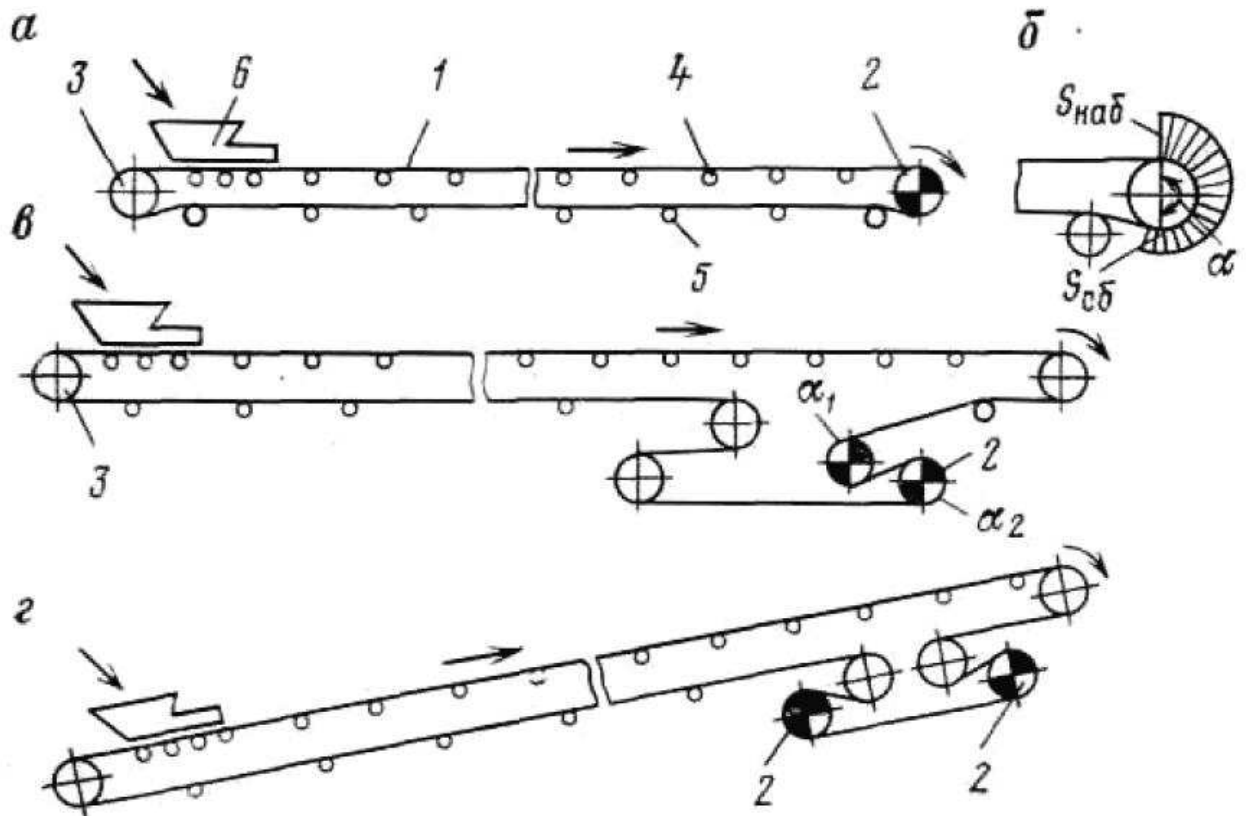


Рисунок 1.4 – Схема ленточных конвейеров и их приводов

Преимуществами ленточных конвейеров являются: высокая производительность, большая длина как в одном ставе, так и всей конвейерной линии; относительная простота конструкции; значительно меньшие масса и удельная энергоёмкость по сравнению со скребковыми конвейерами; высокая надёжность, безопасность и возможность полной автоматизации работы.

Недостатки: ограничение по крупности транспортируемой горной массы (до 500 мм), необходимость прямолинейной установки конвейера в плане, ограниченный угол наклона (с гладкой лентой при транспортировании вверх — до 18° , вниз — до 16°), высокая стоимость и относительно небольшой срок службы конвейерной ленты [5].

Питатели, устройства предназначенные, для равномерной разгрузки промежуточных и накопительных бункеров от кусковых и сыпучих материалов, подачи их в дозированных объёмах.

Питатели качающиеся применяются для выгрузки из бункеров мелко кусковых материалов и равномерной подачи их на переработку и отгрузку потребителю. Принцип действия питателя качающийся основан на возвратно поступательном движении лотка и сброса материала, лежащего на нём. Размер кусков составляет от 15 до 20 мм.

Питатель качающийся представляет собой агрегат, состоящий из: опорной сварной рамы, установленного на ней приёмного блока материала, в нижней части которого установлен лоток (качающийся стол), который перемещается на опорных роликах, и привода лотка, в виде механизма качания, редуктора и электродвигателя. Иногда, над приёмным блоком устанавливают бункер с колосниковой решёткой, для отсеивания крупных кусков.

Принцип работы питателя качающегося основан на бросковом перемещении материала, лежащего на лотке, который перемещается на опорных роликах, и сброса материала, на грохочение или, для последующей подачи на транспортёр передачи в накопительный бункер. Производительность ПК регулируется частотой качаний лотка, что реализуется регулировкой числа оборотов электродвигателя привода [6].

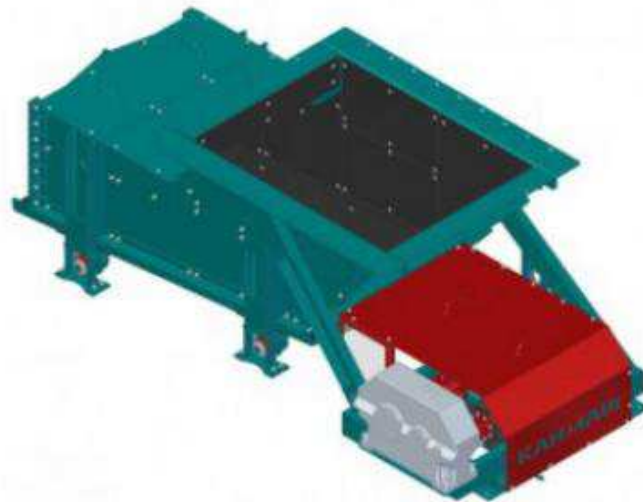


Рисунок 1.5 – Питатель качающийся

Промежуточная разгрузка конвейеров, транспортирующих грузы повышенной абразивности, осуществляется с помощью барабанных разгрузателей (разгрузочных тележек). Их конструкция (рис. 1.6) состоит из оборотных барабанов 1 и 2, установленных на тележке 4 и огибаемых грузонесущей лентой конвейера. Тележка перемещается по рельсам, установленным на раме горизонтального участка става конвейера, с помощью индивидуального электропривода. Разгрузка конвейера производится через барабан 2 в разгрузочную воронку 3, которая снабжается рукавами для односторонней или двусторонней разгрузки [7].

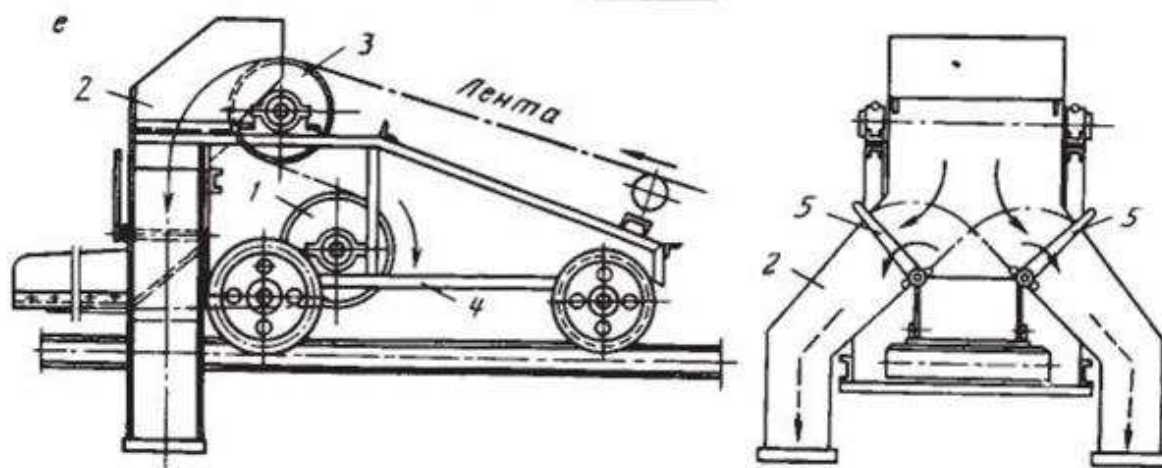


Рисунок 1.6 – Конструкция разгрузочной тележки

Барабанные разгрузатели используются для загрузки ячеек аккумулярующих бункеров или открытых складов. Они совершают реверсивное движение вдоль всего фронта разгрузки, длина которого достигает 100 м и более. Управление передвижением тележки автоматическое, осуществляемое с центрального пульта шахты. Недостатками барабанных разгрузателей являются сложность конструкции, большие габариты и вес, двукратный перегиб ленты на барабанах, снижающий срок ее службы. Их использование целесообразно в тех случаях, когда невозможно применить более простые плужковые разгрузатели [8].

Рис. 1.7 дает представление об общем устройстве шахтной вертикальной двухскиповой подъемной установки [3], служащей для подъема из шахты на поверхность сыпучих материалов, каковыми являются полезное ископаемое и сопутствующие горные породы. Доставленную из забоев в вагонетках горную массу при помощи вагоноопрокидывателя 1 технологического комплекса околоствольного двора 2 разгружают в подземный бункер 3 и затем при помощи дозирующего устройства 4 ее перегружают в подъемный сосуд-скип 5, подвешенный посредством соответствующего прицепного устройства к концу головного (тягового) каната 6 в шахтном стволе 7. Головной канат огибает направляющий (копровой) шкив 8, располагаемый на копре 9. Второй конец

каната закреплен у поверхности канатоведущего органа-барабана 10 подъемной машины, размещаемой вместе с электроприводом в специальном здании 11 на поверхности вблизи шахтного ствола. При вращении барабана один из канатов навивается на его поверхность, обеспечивая подъем груженого сосуда 5, а в это же время второй канат свивается с поверхности барабана, предоставляя возможность порожнему скипу 12 опускаться вниз под действием собственного веса. Подъемная установка настроена таким образом, чтобы загрузка одного скипа в шахте совмещалась с разгрузкой другого скипа на поверхности. Разгрузка осуществляется в бункер 13, откуда горная масса перегружается в соответствующие средства транспорта 14 и направляется к конечному пункту ее доставки на поверхности.

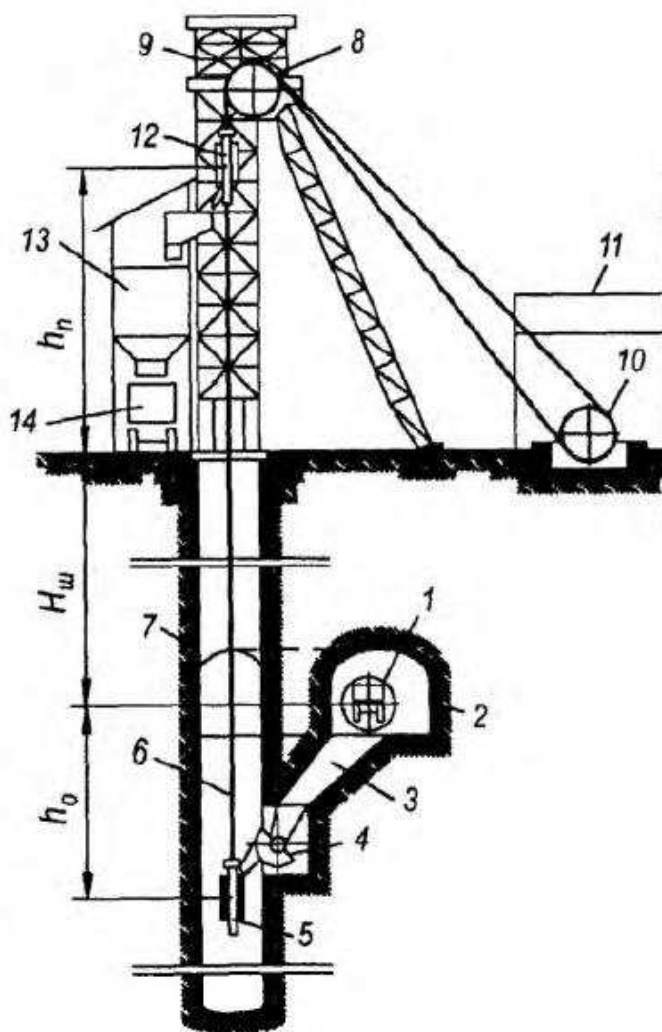


Рисунок 1.7 – Шахтная двухскиповая подъемная установка

1.3 Оборудование АСУТП

Промышленное оборудование АСУТП широко используется во всех отраслях промышленности от нефтяной и газовой сфер до машиностроения, металлообработки и сферы ЖКХ.

В зависимости от типа применяемого оборудования АСУТП, можно создать систему автоматического управления технологическими процессами без участия человека. Или распределенную систему по контролю и сбору данных с большого числа модулей ввода-вывода. Человеку остается только наблюдать за технологическими параметрами на автоматизированное рабочее место (АРМ) оператора [9].

К оборудованию для автоматизации АСУТП относятся контроллеры, модули ввода-вывода, панели оператора, различные датчики и измерители, нормализаторы сигналов, источники питания.

К программному обеспечению можно отнести SCADA системы, инструментальные системы для программирования контроллеров и различные утилиты, упрощающие работу с оборудованием.

Промышленные контроллеры — управляющие устройства, используемые для автоматизации выполнения задач по контролю, управлению и сбору данных с различных устройств без участия человека. В настоящее время широкое распространение получили PC-совместимые контроллеры, отличающиеся от классических ПЛК (программируемый логический контроллер или PLC) тем, что в них большинство функций, которые у ПЛК решаются на аппаратном уровне, выполняются с помощью программного обеспечения. Управление PC-совместимым контроллером осуществляется по средствам SCADA системы или программы написанной с помощью инструментальной системы. Программирование ПЛК обычно осуществляется на стандартных языках МЭК 61131-3, либо на языке СИ, возможно использование и других языков. PC-

контроллеры вобрали в себя все лучшее от компьютеров, в частности, высокую производительность при обработке данных, а от классических ПЛК модульность и большое количество каналов ввода-вывода. В зависимости от задачи такие контроллеры могут управлять от нескольких до десятков тысяч каналов ввода-вывода.

Средства удаленного ввода-вывода представляют собой модули устройства связи с объектом (УСО), которые позволяют собрать показания датчиков, состояния каналов ввода-вывода, а затем передать эти данные на верхний уровень контроллерам или SCADA-системам. Модули удаленного ввода-вывода можно разделить на группы:

- по типу интерфейса: RS-485, Ethernet, Wi-Fi, USB, CAN;
- по типу используемого протокола: DCON, Modbus, Profibus, Profinet, EtherNetIP, BACnet, EtherCAT, CAN, CANopen, DeviceNET;
- по типу каналов ввода-вывода: аналоговый ввод-вывод, дискретный ввод-вывод, релейный выход.

Самое главное преимущество использования модулей удаленного ввода-вывода – это возможность удаленной установки модуля ввода-вывода от управляющего контроллера или компьютера. Вся информация с датчиков передается в цифровом виде по каналам связи и не искажается в отличие от передачи по аналоговым линиям связи.

Платы ввода-вывода — основной элемент для построения автоматизированных систем сбора и обработки данных на основе компьютера. Большой выбор плат ввода-вывода позволяет решать широкий спектр задач цифровой обработки аналоговых, дискретных и импульсных сигналов самых разнообразных диапазонов, формировать сигналы заданной амплитуды, формы и последовательности. Многофункциональные адаптеры ввода-вывода представляют из себя законченную подсистему для решения задач сбора данных и управления. Такие платы служат основой для построения контроллеров,

сложных лабораторных систем обработки сигналов, "умных" измерительных приборов на основе РС-совместимых компьютеров.

Панели оператора и терминалы (HMI) представляют собой компактные специализированные устройства, позволяющие решить задачу организации операторского интерфейса для вывода данных и управления. Панель оператора и терминалы особенно актуальны для локальных систем управления, в которых часто возникает задача ввода и корректировки параметров технологического процесса в самых разнообразных точках системы. По внешнему виду панели оператора похожи на панельные компьютеры, т.к. имеют сенсорный экран, но отличаются от них меньшими вычислительными возможностями и обычно управляются специальным программным обеспечением.

Измерители — устройства, предназначенные для измерения каких-либо физических величин. От датчиков отличаются тем, что они могут работать автономно без подключения дополнительных устройств. Измерители помогают контролировать процессы АСУТП и своевременно реагировать на изменения в окружающей среде. С помощью измерителей можно получить данные о значениях тока, напряжения и мощности в сети или значения температуры и влажности окружающей среды.

Нормализаторы преобразуют сигналы источников постоянного и переменного напряжения и тока, токовой петли, термопар, термометров сопротивления, тензодатчиков, резисторов и потенциометров и т.д. к унифицированным сигналам стандартных диапазонов. Для аналоговых сигналов такими диапазонами обычно являются, 0...5 В, 0...10 В, 0...20 мА, 4...20 мА, либо токовая петля. Для дискретных - сигналы TTL-уровня с диапазоном 0...5 В. Преобразователи и нормализаторы сигналов являются необходимым связующим звеном между огромным миром датчиков и исполнительных устройств и системами сбора данных и управления на основе компьютеров или контроллеров. Являясь неотъемлемой частью измерительного канала, они обеспечивают информационную совместимость всей системы. Кроме функции

преобразования, нормализаторы дополнительно обеспечивают полную гальваническую развязку, фильтрацию, буферизацию, усиление или ослабление сигнала, линейризацию, детектирование обрыва датчика.

Программное обеспечение позволяет пользователю взаимодействовать с оборудованием: настраивать, программировать работу устройств, получать данные и так далее. ПО для программирования контроллеров называется инструментальной системой. На контроллер же устанавливается ПО типа “исполнительная система”, которое управляет контроллером по алгоритму, заложенному пользователем. SCADA-системы позволяют:

- программировать логику работы устройств на языках МЭК-61131-3;
- получать данные от различных устройств по стандартным протоколам;
- автоматически принимать решения в соответствии с заложенным алгоритмом;
- вести мониторинг и архивирование данных.

2 Автоматизация объекта

2.1 Перечень сигналов

Подсистема загрузки:

- 1) Управляемые сигналы
 - a) Питатель 1 включить/выключить
 - b) Питатель 2 включить/выключить
 - c) Шибер бункера 1 открыть/закрыть
 - d) Шибер бункера 2 открыть/закрыть
 - e) Систему орошения 1 включить/выключить
 - f) Система орошения 2 включить/выключить
 - g) Конвейер включить/выключить
 - h) Затвор дозатора скипа 1 открыть/закрыть
 - i) Затвор дозатора скипы 2 открыть/закрыть
- 2) Регулируемые сигналы
 - a) Скорость ленты конвейера
 - b) Уровень в бункере 1
 - c) Уровень в бункере 2
 - d) Уровень в дозаторе 1
 - e) Уровень в дозаторе 2
- 3) Контролируемые сигналы
 - a) Положение шибера бункера 1 (Открыт/Закрыт)
 - b) Положение шибера бункера 2
 - c) Состояние питателя 1 (Включен/Выключен)
 - d) Состояние питателя 2
 - e) Состояние конвейера (Включен/Выключен)
 - f) Положение затвора дозатора скипы 1 (Открыт/Закрыт)
 - g) Положение затвора дозатора скипы 2
 - h) Скип 1 на горизонте

- i) Скип 2 на горизонте
- j) Подъем руды
- k) Подъем породы

Подсистема разгрузки:

1) Управляемые сигналы:

- a) Питатель включить/выключить
- b) Конвейер включить/выключить
- c) Установить тележку в положение «Порода» / «Руда»
- d) Шибер «порода» открыть/закрыть
- e) Шибер «руда» открыть/закрыть
- f) Систему аспирации включить/выключить
- g) Задвижку скипа 1 открыть/закрыть
- h) Задвижку скипа 2 открыть/закрыть
- i) Затвор пневмораспределителя скипа 1 открыть/закрыть
- j) Затвор пневмораспределителя скипа 2 открыть/закрыть

2) Регулируемые сигналы:

- a) Уровень в бункере
- b) Давление в пневмоустройстве скипа 1
- c) Давление в пневмоустройстве скипа 2

3) Контролируемые сигналы:

- a) Состояние питателя (включен/выключен)
- b) Состояние конвейера
- c) Положение натяжной станции конвейера (Мин/Макс)
- d) Положение тележки («Порода» / «Руда»)
- e) Положение шибера «порода» (Открыт/Закрыт)
- f) Положение шибера «руда»
- g) Состояние системы аспирации (включен/выключен)
- h) Положение задвижки скипа 1
- i) Положение задвижки скипа 2

- ж) Положение затвора пневмораспределителя скипа 1
- к) Положение затвора пневмораспределителя скипа 2
- л) Скип 1 на разгрузке
- м) Скип 2 на разгрузке

2.2 Структурно-функциональная анализ

Функциональная схема представлена на рисунке 2.1.

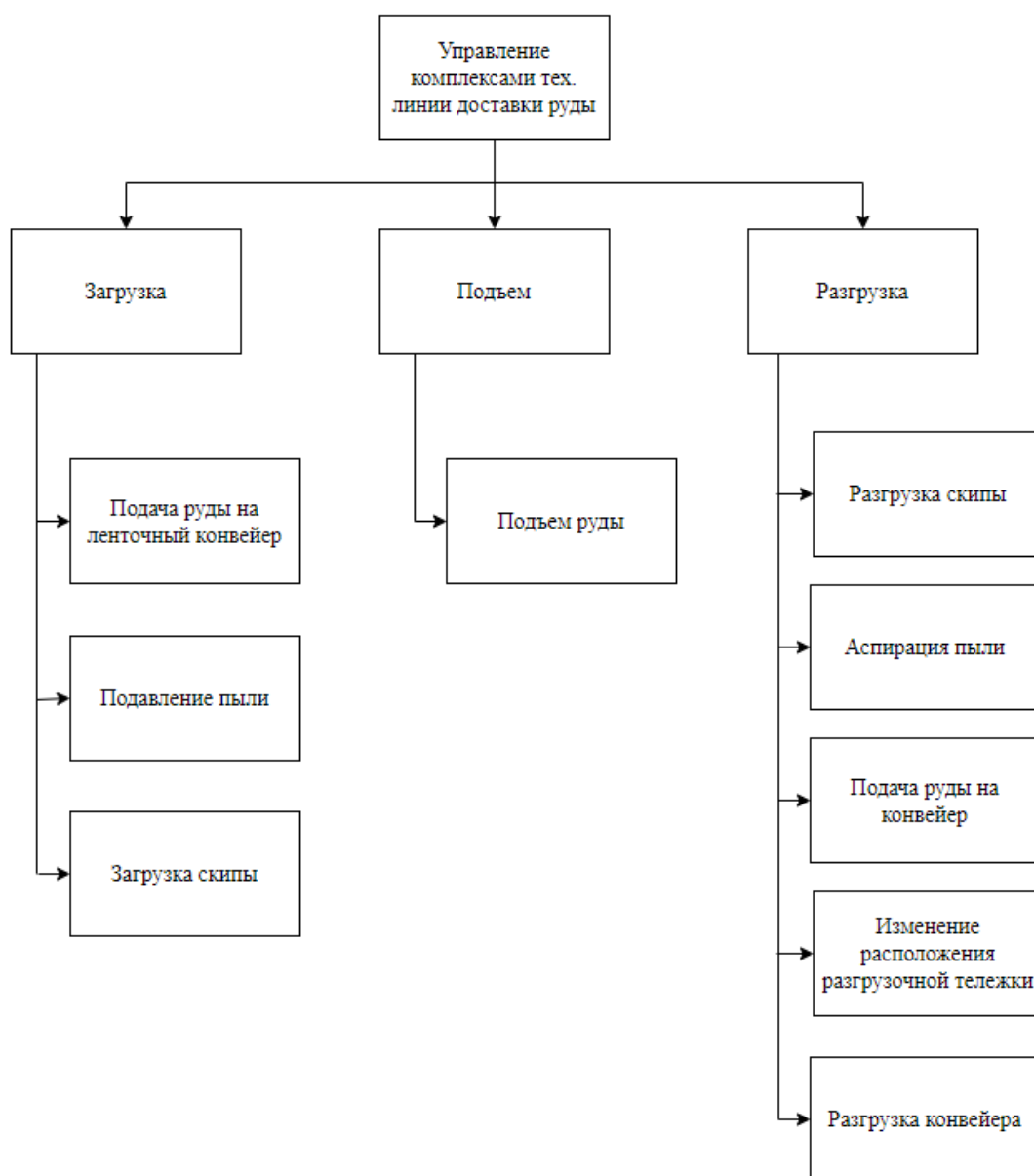


Рисунок 2.1 – Функциональная иерархическая схема

Структурная схема представлена на рисунке 2.2.

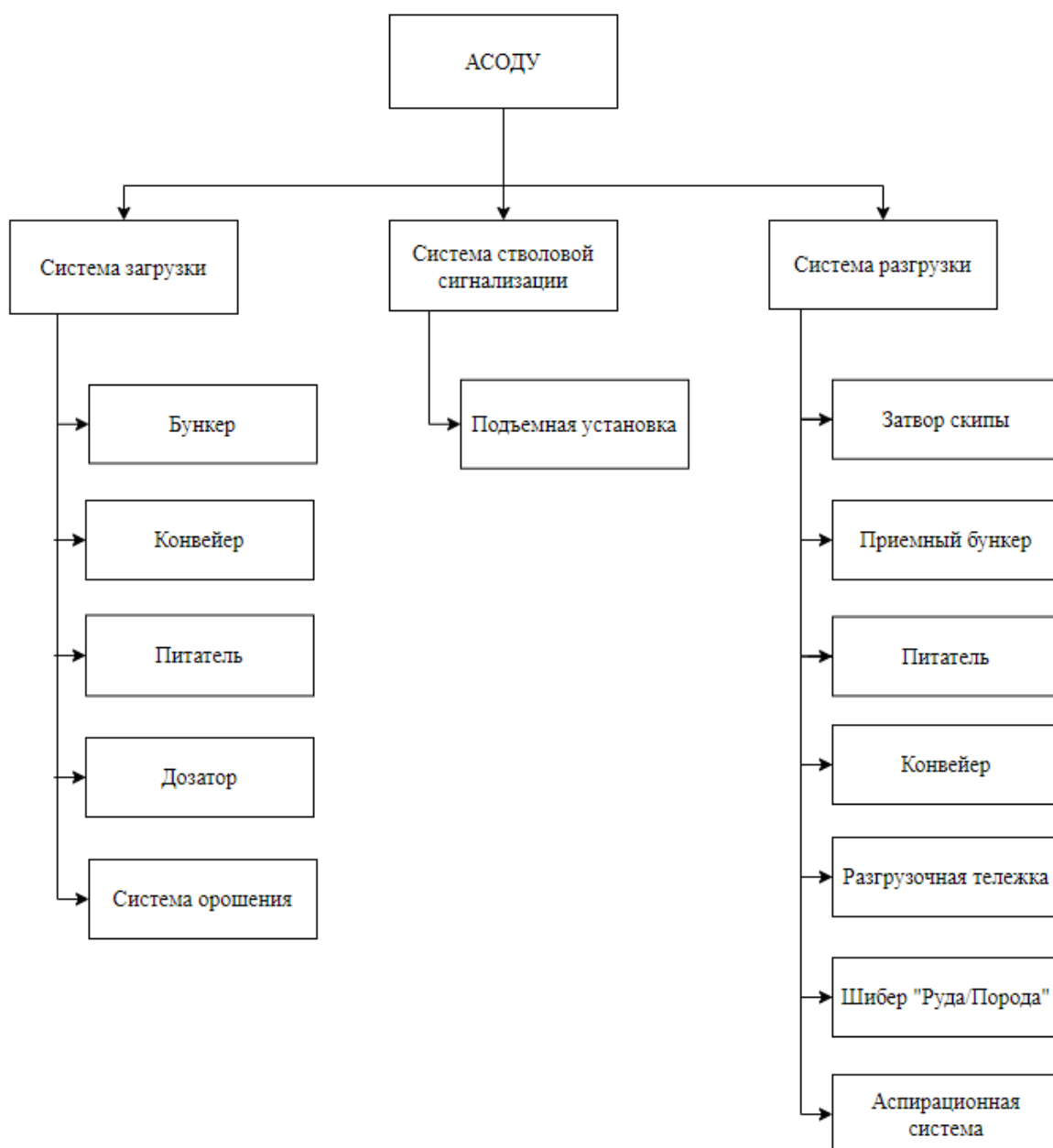


Рисунок 2.2 –Структурная иерархическая схема

Объединим полученные иерархии в матрицу проекций, столбцы которой соответствуют функциям автоматизированной системы управления, а строки – элементам структурной организации АСУ.

	Бункер	Система орошения	Питатель	Конвейер	Дозатор	Подъемная установка	Затвор скипы	Аспирационная система	Приемный бункер	Питатель	Конвейер	Разгрузочная тележка	Шибера "Руда/Порода"
Подача руды	×												
Подавление пыли		×											
Загрузка скипы			×	×	×								
Подъем руды						×							
Разгрузка скипы							×						
Аспирация пыли								×					
Подача руды									×	×	×		
Изменение расположения разгрузочной тележки												×	
Разгрузка конвейера													×

Рисунок 2.3 – Функционально-структурная организация

На рисунке 2.3 представлена функционально-структурная организация, по которой можно сделать вывод, что каждой функции соответствуют средства транспортировки.

Функциональная схема представлена в приложении Л.

2.3 Алгоритмы управления

Алгоритм автоматического управления систем загрузки представлен на рисунке 2.4. Как только дозатор заполняется до верхнего уровня, отключаются средства транспортировки системы загрузки, закрываются шибера бункеров 1 и 2. Далее происходит ожидание скипы для опорожнения бункера дозатора. Когда бункер дозатора станет полностью пустой, затвор закроется и включаться средства транспортировки, откроются шибера бункеров.

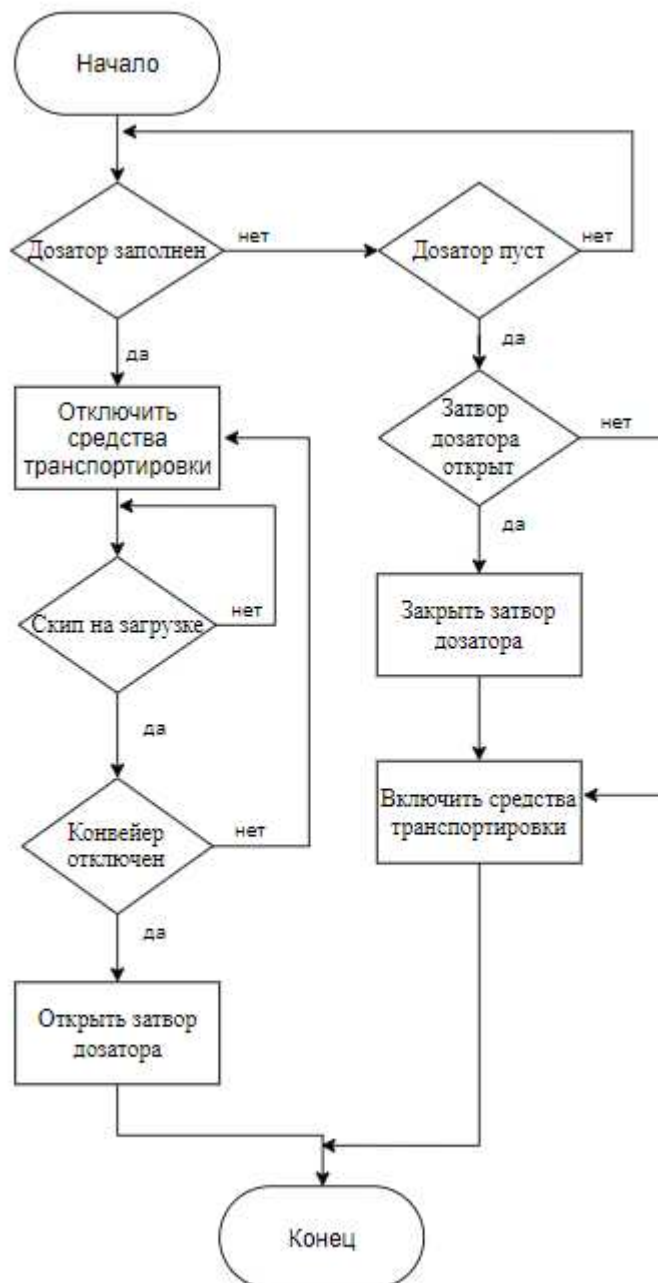


Рисунок 2.4 – Схема алгоритма управления системой загрузки

Алгоритм автоматического управления подъемной установкой представлен в виде схемы [10] на рисунке 2.5. Если дозатора заполнен, то к нему движется соответствующий скип, и как только скип дойдет до горизонта загрузки, он останавливается и ждет пока не опорожнится бункер дозатора. Второй скип на горизонте разгрузки открывает затвор, и весь материал

пересыпается в приемный бункер. Далее тот скип, который был на загрузке, поднимается, другой – спускается.

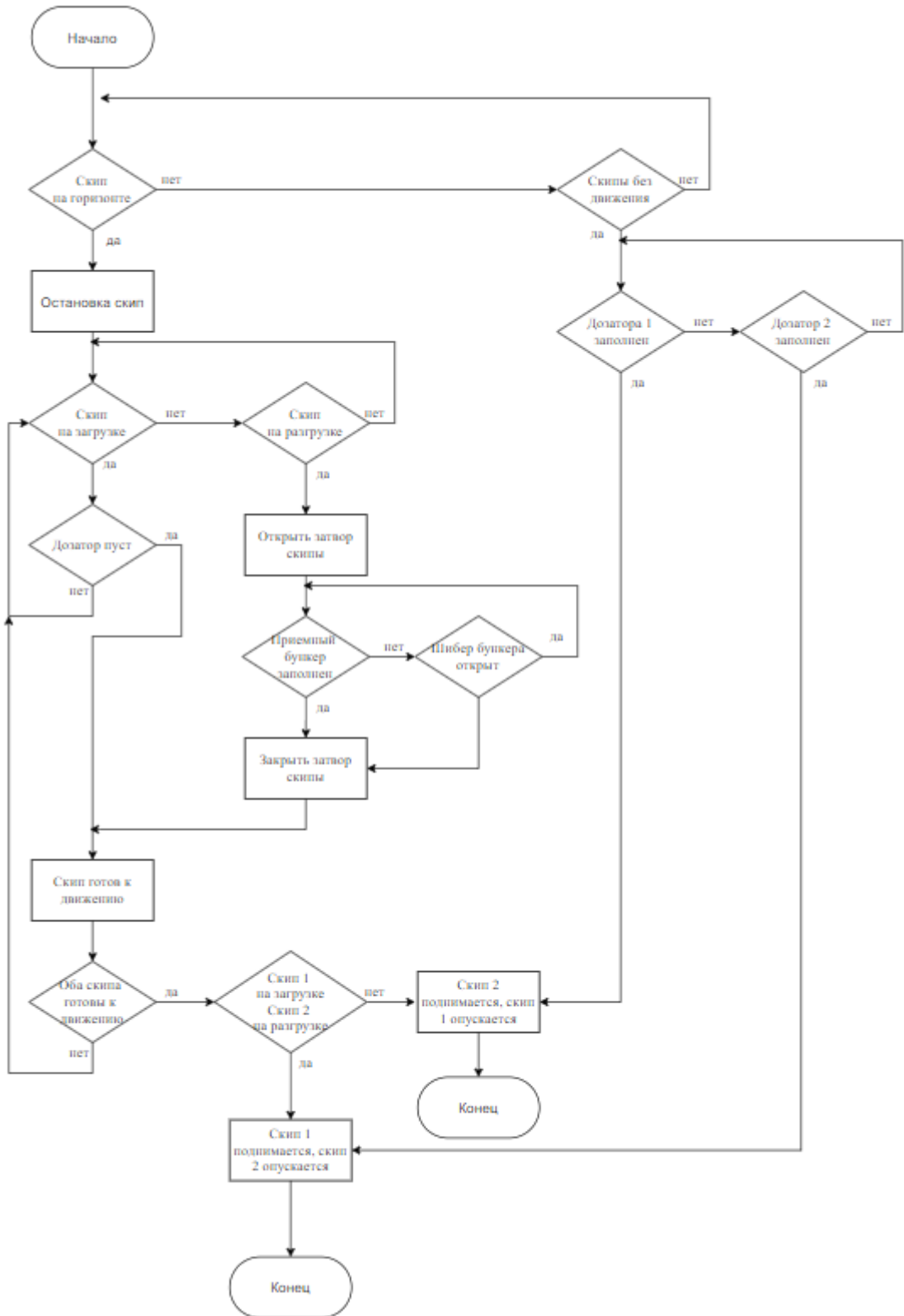


Рисунок 2.5 – Схема алгоритма управления подъемной установкой

Как только приемный бункер заполнится, включаются средства транспортировки, открываются шиберы, и если тип перевозимого материала «руда», то разгрузочную тележку перемещают к окну разгрузки руды и наоборот.

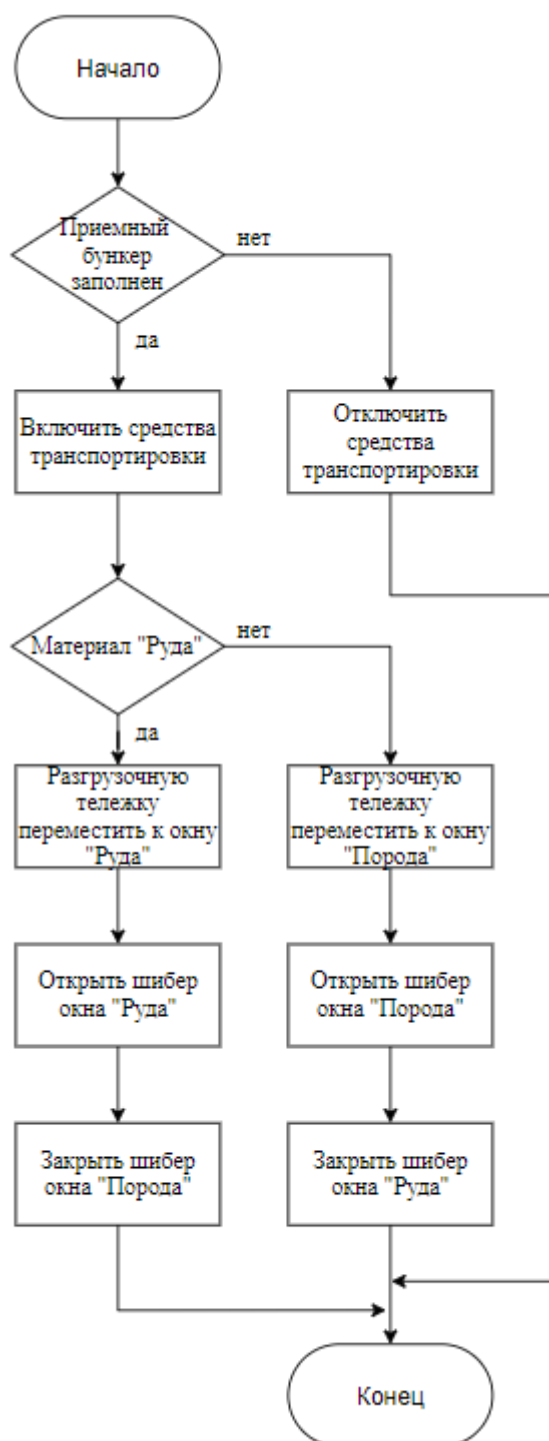


Рисунок 2.6 – Схема алгоритма управления системой разгрузки

3 Создание имитационной модели

3.1 Выбор и обоснование среды разработки

В качестве среды разработки во время прохождения научно-исследовательской работы была выбрана среда динамического моделирования технических систем *SimInTech*.

SimInTech – российская система модельно-ориентированного проектирования систем автоматического управления [11]. Данное программное обеспечение является средой создания математических моделей, алгоритмов управления, интерфейсов управления и автоматической генерации кода для программируемых контроллеров и графических дисплеев. Является аналогом систем *Matlab Simulink*, *Dymola (Dassault)*, *AMESim (Siemens)*, *SCADE (Esterel)* и других.

Главное преимущество *SimInTech* в том, что это Российская разработка, т.е. можно быстро получить обратную связь. Сам интерфейс очень прост и лаконичен, исходные коды ПО доступны для сертификации по требованиям. Технически это самое быстрое математическое ядро, автоматическое оформление проектов по требованиям организации, оптимальная генерация кода, высокая гибкость. Индивидуальный подход – команда разработчиков находится в Москве, что сокращает сроки реакции на запросы в части технической поддержки, также возможна адаптация ПО под задачи конкретного заказчика.

Одной из основных особенностей использования *SimInTech* для создания комплексных моделей является идеология использования «Базы данных сигналов» – структурированного списка переменных, обеспечивающих обмен расчетными значениями между расчетными схемами в единой модели [12]. База данных сигналов *SimInTech* является объектной и обеспечивает пользователю удобное решение следующих задач:

- объединение нескольких расчетных схем в единую модель;
- обеспечение возможности векторной обработки сигналов для типовых алгоритмов управления;
- объектно-ориентированное проектирование модели технических систем;
- автоматизация создания и обработки переменных в комплексных моделях.

Для создания сложной математической модели происходит объединение моделей основных подсистем, реализованных в виде отдельных проектов, в единый пакет [13]. Преимуществом пакетного способа реализации является разбиение модели на отдельные проекты, которые можно редактировать вне зависимости от других проектов и которые взаимосвязаны между собой с помощью базы данных сигналов проекта.

3.2 Требование к модели

Автоматизированная система транспортировки горной массы с горизонта загрузки на склад руды реализуется в среде динамического моделирования технических систем *SimInTech* как пакет проектов, взаимосвязанных через общую базу сигналов.

Состав модели:

- 1) База данных сигналов;
- 2) Проект подъемной установки;
- 3) Проект подсистемы загрузки;
- 4) Проект подсистемы разгрузки;
- 5) Модуль управления;
- 6) Блок диспетчера;
- 7) Блок оператора подсистемы загрузки;
- 8) Блок оператора подсистемы разгрузки.

Основные используемые средства *SimInTech*:

- База Данных Сигналов;
- Типовые динамические звенья;
- Операторы;
- Ключи;
- Субструктуры;
- Блок программирования;
- Блок Анимации;
- Панель Прimitives;
- Блок Язык программирования;
- Вывод данных;

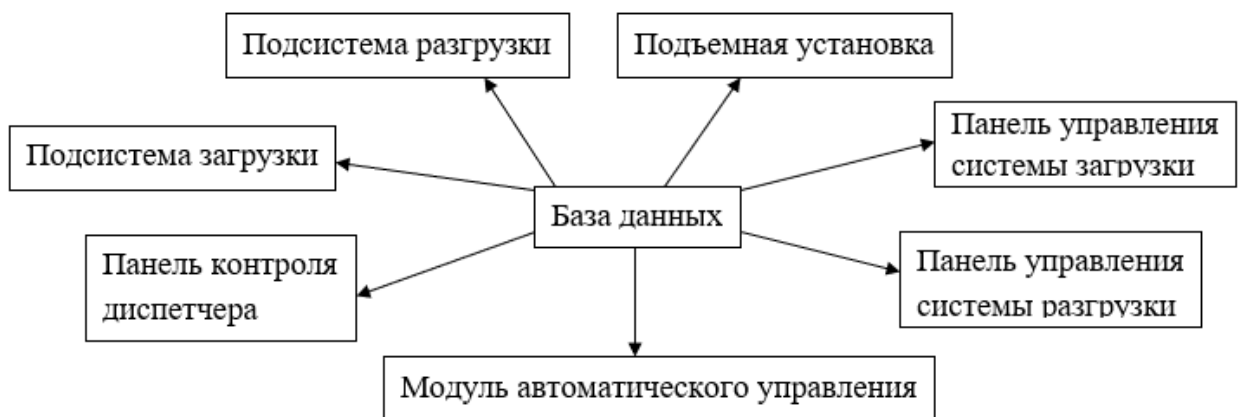


Рисунок 3.1 – Структура модели

Структура модели, представленная на рисунке 3.1, демонстрирует объединение нескольких проектов в единую модель.

3.3 Реализация модели

Перечень проектов:

- Проект подъемной установки (lift.prt);

- Проект подсистемы загрузки (charge.prt);
- Проект подсистемы разгрузки (discharge.prt);
- Проекта системы управления (controller.prt);
- Проект визуализации (3Dproject);

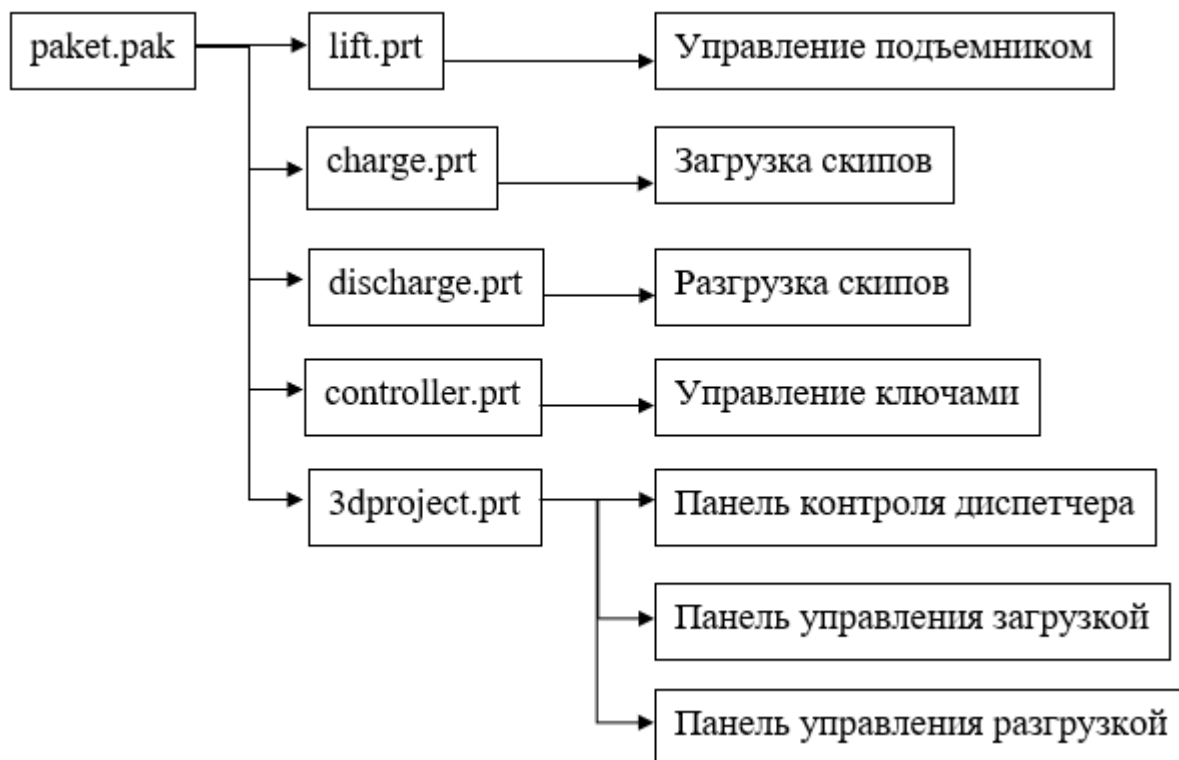


Рисунок 3.2 – Реализации модели в *SimInTech*

На рисунке 3.2 представлено распределение функций модели по отдельным проектам.

3.4 Составление базы данных

Для работы проекта необходима связь между схемами общего вида, для этого была создана общая база данных.

Таблица 3.1 – База данных сигналов

Имя	Название	Тип данных
Группа сигналов <i>charge_com</i>		
<i>bunker1</i>	Загрузка бункера 1	Двоичное
<i>bunker2</i>	Загрузка бункера 2	Двоичное
<i>bunker1_gate</i>	Открытие/Закрытие шибера бункера 1	Двоичное
<i>bunker2_gate</i>	Открытие/Закрытие шибера бункера 2	Двоичное
<i>feeder1</i>	Включение/Выключение питателя 1	Двоичное
<i>feeder2</i>	Включение/Выключение питателя 2	Двоичное
<i>conveyor</i>	Включение/Выключение конвейера	Двоичное
<i>dozator1_gate</i>	Открыть/Закрыть шибер дозатора 1	Двоичное
<i>dozator2_gate</i>	Открыть/Закрыть шибер дозатора 1	Двоичное
<i>watering</i>	Система орошения	Двоичное
<i>auto</i>	Включение режима автомат	Двоичное
Группа сигналов <i>charge_state</i>		
<i>bunker1_lvl</i>	Уровень в бункере 1	Вещественное
<i>bunker2_lvl</i>	Уровень в бункере 2	Вещественное
<i>bunker1</i>	Загрузка бункера 1	Двоичное
<i>bunker2</i>	Загрузка бункера 2	Двоичное
<i>bunker1_gate</i>	Шибер бункера 1 открыт/закрыт	Двоичное
<i>bunker2_gate</i>	Шибер бункера 2 открыт/закрыт	Двоичное
<i>feeder1</i>	Питатель 1 включен/выключен	Двоичное
<i>feeder2</i>	Питатель 2 включен/выключен	Двоичное
<i>conveyor</i>	Конвейер включен/выключен	Двоичное
<i>dozator1_lvl</i>	Уровень в дозаторе 1	Вещественное
<i>dozator2_lvl</i>	Уровень в дозаторе 2	Вещественное
<i>dozator1_gate</i>	Шибер дозатора 1 открыт/закрыт	Двоичное
<i>dozator2_gate</i>	Шибер дозатора 1 открыт/закрыт	Двоичное
<i>watering</i>	Система орошения	Двоичное
<i>auto</i>	Автоматический режим управления	Двоичное
Группа сигналов <i>discharge_com</i>		
<i>bunker_gate</i>	Открыть/Закрыть шибер приемного бункера	Двоичное
<i>feeder</i>	Включить/Выключить питатель	Двоичное
<i>conveyor</i>	Включить/Выключить конвейер	Двоичное
<i>car_pos_solid</i>	Изменить положение тележки на "порода"	Двоичное

Продолжение таблицы 3.1

Имя	Название	Тип данных
<i>car_pos_ore</i>	Изменить положение тележки на "руда"	Двоичное
<i>solid_gate</i>	Открыть/Закрыть шибер окна породы	Двоичное
<i>ore_gate</i>	Открыть/Закрыть шибер окна руды	Двоичное
<i>aspiration</i>	Аспирационная система	Двоичное
<i>auto</i>	Включение режима автомат	Двоичное
Группа сигналов <i>discharge_state</i>		
<i>bunker_lvl</i>	Уровень в приемном бункере	Вещественное
<i>bunker_gate</i>	Шибер приемного бункера Открыт/Закрыт	Двоичное
<i>feeder</i>	Питатель Включен/Выключен	Двоичное
<i>conveyor</i>	Конвейер Включен/Выключен	Двоичное
<i>car_pos_solid</i>	Положение тележки "порода"	Двоичное
<i>car_pos_ore</i>	Положение тележки "руда"	Двоичное
<i>solid_gate</i>	Шибер окна породы Открыт/Закрыт	Двоичное
<i>ore_gate</i>	Шибер окна руды Открыт/Закрыт	Двоичное
<i>discharge</i>	Процесс разгрузки	Двоичное
<i>aspiration</i>	Аспирационная система	Двоичное
<i>auto</i>	Автоматический режим управления	Двоичное
Группа сигналов <i>lift_com</i>		
<i>course</i>	Направление движения	Вещественное
<i>stop</i>	Остановка	Вещественное
<i>skip1_gate</i>	Открыть/Закрыть задвижку скипы 1	Двоичное
<i>skip2_gate</i>	Открыть/Закрыть задвижку скипы 2	Двоичное
<i>skip1_pneum</i>	Открыть/Закрыть пневмораспределитель скипы 1	Двоичное
<i>skip2_pneum</i>	Открыть/Закрыть пневмораспределитель скипы 2	Двоичное
<i>skip1_ready</i>	Разрешение на движение скипы 1	Двоичное
<i>skip2_ready</i>	Разрешение на движение скипы 2	Двоичное
<i>auto</i>	Автоматический режим управления	Двоичное
Группа сигналов <i>lift_state</i>		
<i>course</i>	Направление движения	Двоичное
<i>stop</i>	Остановка	Двоичное
<i>skip1_gate</i>	Задвижка скипы 1 открыта/закрыта	Двоичное
<i>skip2_gate</i>	Задвижка скипы 2 открыта/закрыта	Двоичное
<i>skip1_lvl</i>	Уровень в скипе 1	Вещественное
<i>skip2_lvl</i>	Уровень в скипе 2	Вещественное
<i>skip1_hgt</i>	Высота скипы 1	Вещественное
<i>skip2_hgt</i>	Высота скипы 2	Вещественное

Окончание таблицы 3.1

Имя	Название	Тип данных
<i>skip1_charge</i>	Скип 1 на горизонте загрузки	Двоичное
<i>skip2_charge</i>	Скип 2 на горизонте загрузки	Двоичное
<i>skip1_discharge</i>	Скип 1 на горизонте разгрузки	Двоичное
<i>skip2_discharge</i>	Скип 2 на горизонте разгрузки	Двоичное
<i>skip1_pneum</i>	Положение пневмораспределителя скипы 1	Двоичное
<i>skip2_pneum</i>	Положение пневмораспределителя скипы 2	Двоичное
<i>auto</i>	Автоматический режим управления	Двоичное
Группа сигналов <i>charge_py</i>		
<i>bunker1</i>	Загрузка бункера 1	Двоичное
<i>bunker2</i>	Загрузка бункера 2	Двоичное
<i>bunker1_gate</i>	Открытие/Закрытие шибер бункера 1	Двоичное
<i>bunker2_gate</i>	Открытие/Закрытие шибер бункера 2	Двоичное
<i>feeder1</i>	Включение/Выключение питателя 1	Двоичное
<i>feeder2</i>	Включение/Выключение питателя 2	Двоичное
<i>conveyor</i>	Включение/Выключение конвейера	Двоичное
<i>dozator1_gate</i>	Открыть/Закрыть шибер дозатора 1	Двоичное
<i>dozator2_gate</i>	Открыть/Закрыть шибер дозатора 1	Двоичное
<i>solid</i>	Тип загружаемого материала "порода"	Двоичное
<i>ore</i>	Тип загружаемого материала "руда"	Двоичное
<i>auto</i>	Включение режима автомат	Двоичное
Группа сигналов <i>discharge_py</i>		
<i>bunker_gate</i>	Открыть/Закрыть шибер приемного бункера	Двоичное
<i>feeder</i>	Включить/Выключить питатель	Двоичное
<i>conveyor</i>	Включить/Выключить конвейер	Двоичное
<i>car_pos_solid</i>	Изменить положение тележки на "порода"	Двоичное
<i>car_pos_ore</i>	Изменить положение тележки на "руда"	Двоичное
<i>solid_gate</i>	Открыть/Закрыть шибер окна породы	Двоичное
<i>ore_gate</i>	Открыть/Закрыть шибер окна руды	Двоичное
<i>auto</i>	Включение режима автомат	Двоичное
Группа сигналов <i>lift_py</i>		
<i>skip1_gate</i>	Затвор скипы 1	Двоичное
<i>skip2_gate</i>	Затвор скипы 2	Двоичное
<i>skip1_up</i>	Направление скипы 1 вверх	Двоичное
<i>skip1_down</i>	Направление скипы 1 вниз	Двоичное
<i>skip2_up</i>	Направление скипы 2 вверх	Двоичное
<i>skip2_down</i>	Направление скипы 2 вниз	Двоичное
<i>stop</i>	Пуск/Остановка	Двоичное

3.5 Моделирование схем общего вида

На рисунке 3.3 представлен проект подъемной установки, имеющий 4 субструктуры, 2 демультиплексора и список выходных сигналов. Текущий проект отвечает за изменение высоты скип 1 и 2, регулирования их скорости и изменение уровня заполненности.

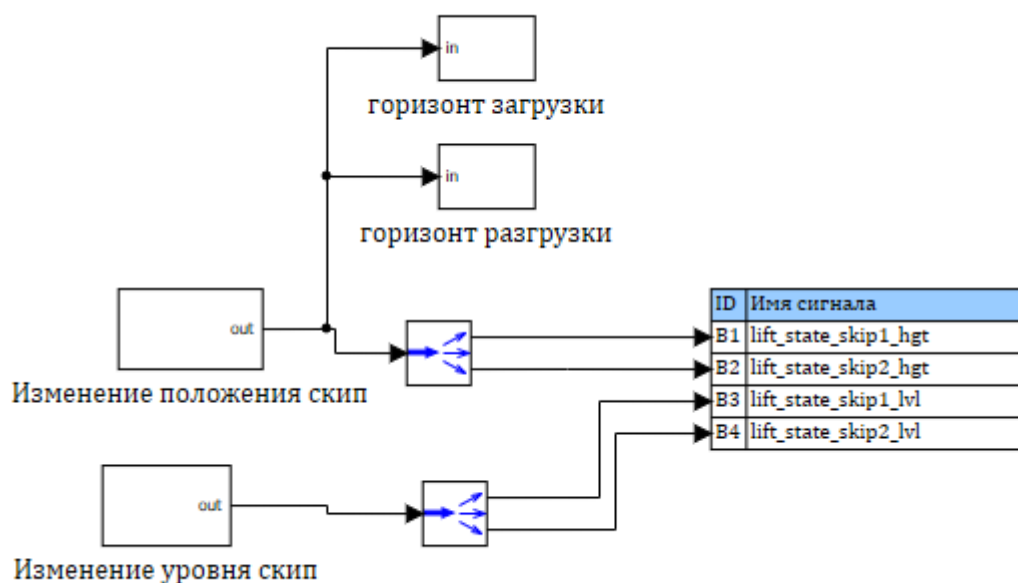


Рисунок 3.3 – Проект подъемной установки

Субструктура «Изменение уровня скип», показанный на рисунке 3.4 имеет список входных сигналов, 6 ключей, 2 блока сравнения, 2 сумматора, 2 интегратора с ограничением и мультиплексор. Это субструктура имитирует изменение уровня наполнения скип.

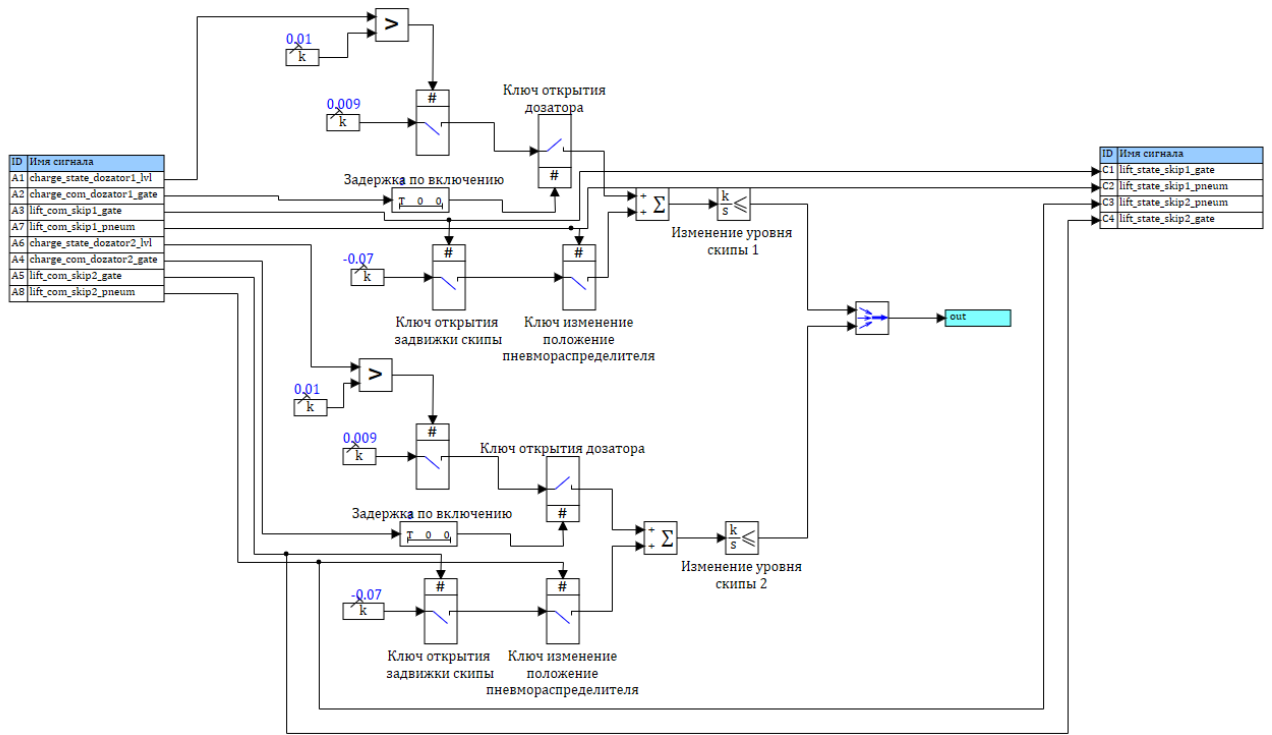


Рисунок 3.4 – Субструктура «Изменение уровня скип»

На рисунке 3.5 субструктура «Изменение положения скип» имеет 2 передаточные функции, 3 ключа, 2 интегратора с ограничением, списки входных и выходных сигналов и мультиплексор. Эта субструктура имитирует перемещение скипы вдоль ствола и замедление скорости при приближении к горизонтам загрузки и разгрузки.

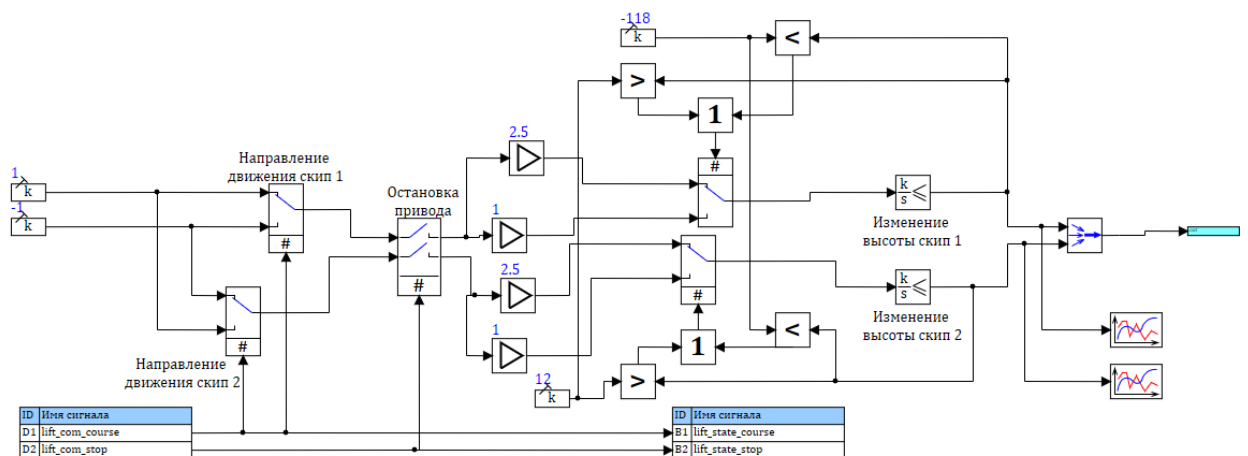


Рисунок 3.5 – Субструктура «Изменение положения скип»

Субструктуры «Горизонт загрузки» (рис. 3.6) и «Горизонт разгрузки» (рис.2.7) имеют по 2 блока сравнения, демультиплексоры и списки выходных сигналов, и нужны для обозначения, что скипы добрались до необходимой высоты загрузки или разгрузки.

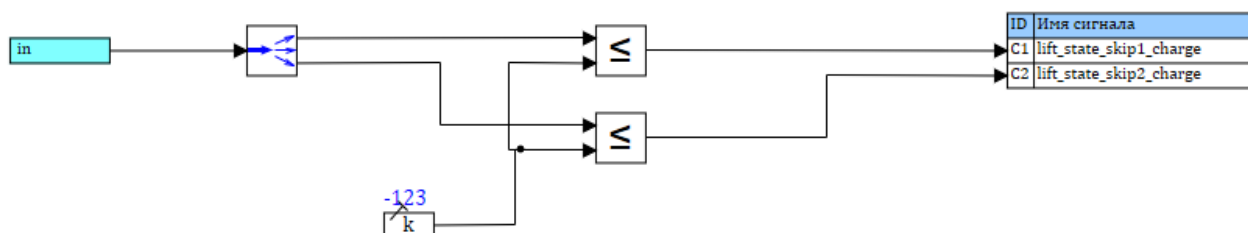


Рисунок 3.6 – Субструктура «Горизонт загрузки»

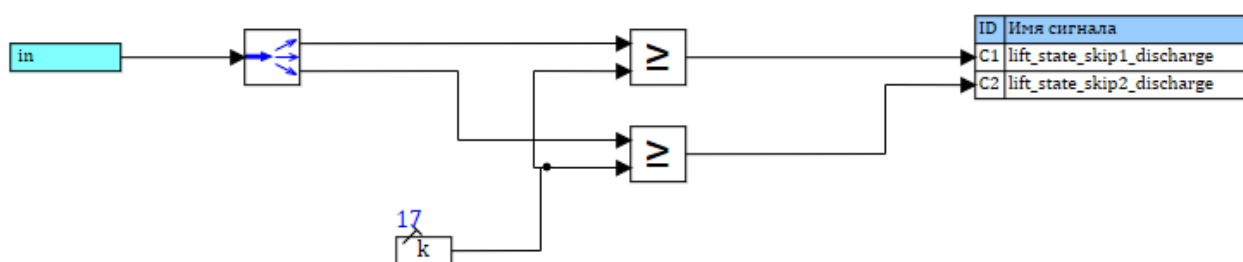


Рисунок 3.7 – Субструктура «Горизонт разгрузки»

Проект загрузки, представленный на рис. 3.8, имеет 2 субструктуры, мультиплексор, 2 демультиплексора, входной сигнал, блок задержки и список выходных сигналов. Этот проект отвечает за изменение уровня наполненности бункеров 1 и 2 системы загрузки, изменения уровня наполненности дозаторов 1 и 2 и включении или отключении системы орошения для подавления пыли.

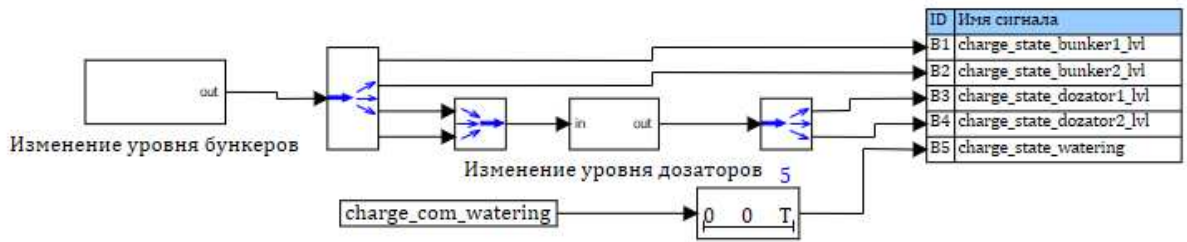


Рисунок 3.8 – Проект загрузки

Субструктура «Изменение уровня бункеров», видимый на рисунке 3.9, имеет списки входных и выходных сигналов, 4 блока случайных величин, 4 усилителя, 6 ключей, 2 сумматора, 2 интегратора с ограничением, 2 сравнивающих блока и мультиплексор. Эта субструктура имитирует изменение уровня наполненности бункера: опустошение и наполнение. Также тут присутствует условие опустошения, при выполнении которого бункер больше не сможет опустошаться.

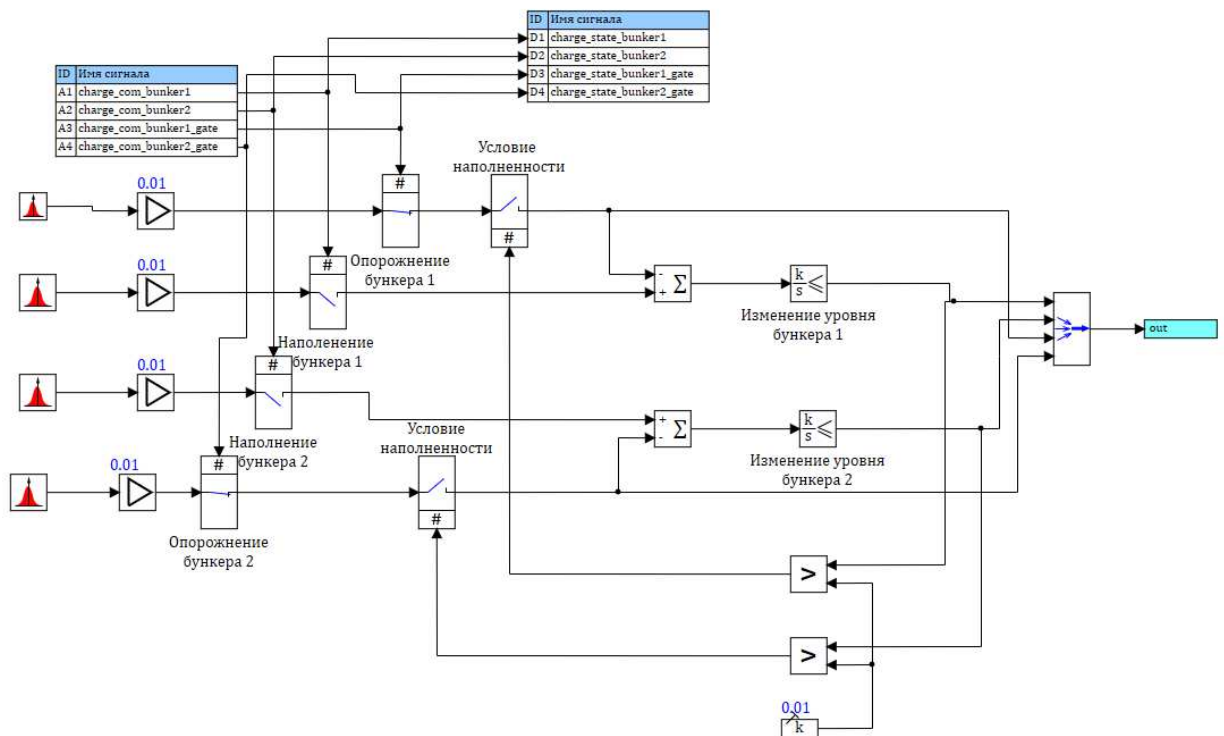


Рисунок 3.9 – Субструктура «Изменение уровня бункеров»

Субструктура «Изменение уровня дозаторов», показанный на рисунке 3.10 имеет списки входных и выходных сигналов, блок случайных величин, 10 ключей, 2 усилителя, 2 сумматора, 2 интегратора с ограничением, 2 блока сравнения, демультиплексор и мультиплексор. Эта субструктура имитирует изменение уровня наполненности бункеров дозаторов 1 и 2

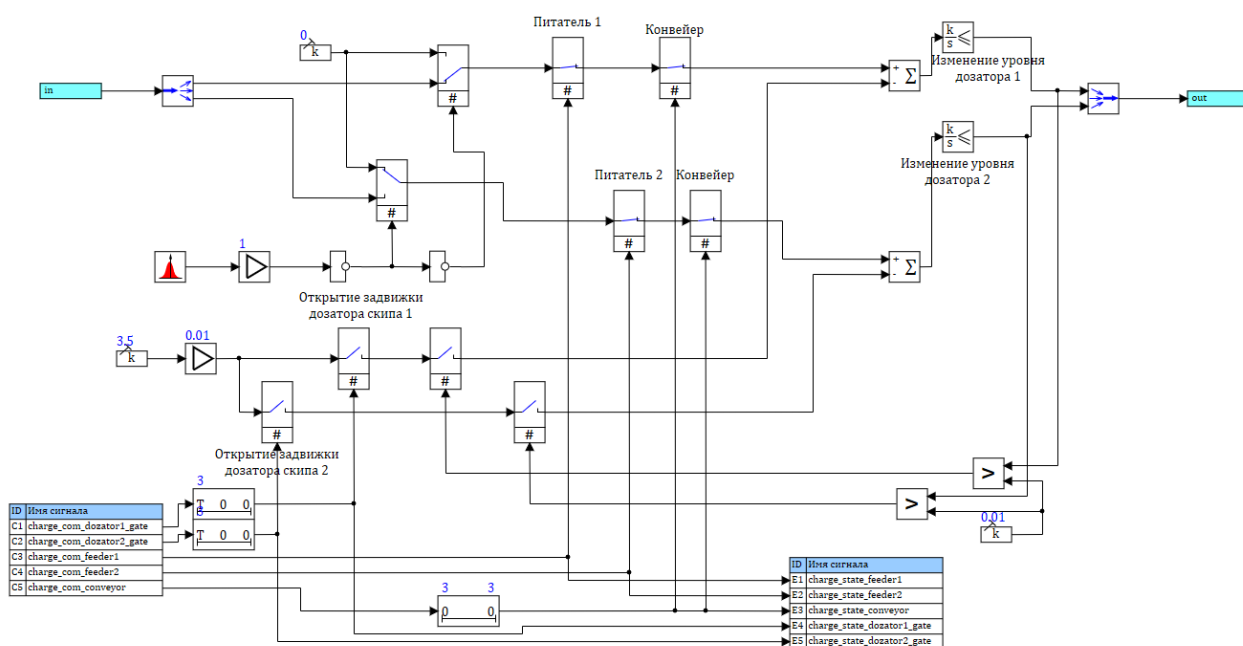


Рисунок 3.10 – Субструктура «Изменение уровня дозаторов»

На рисунке 3.11 представлен проект разгрузки, который имеет 2 субструктуры, демультиплексор и список выходных сигналов. Проект имитирует изменение уровня наполненности приемного бункера системы разгрузки и сам процесс разгрузки, считая количество разгруженного материала.

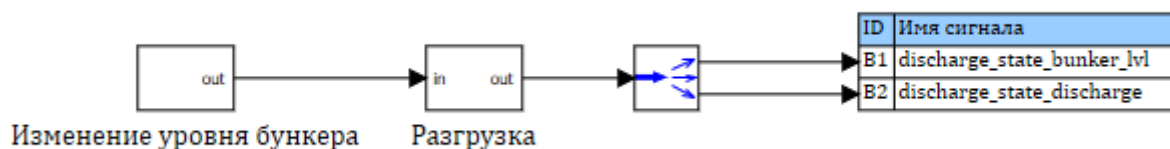


Рисунок 3.11 – Проект разгрузки

Субструктура «Изменение уровня бункера» представлена на рисунке 3.12 и имеет список входных и выходных сигналов, 3 блока сравнения, 2 блока конъюнкции, блока дизъюнкции, 3 ключа, блок случайных величин, усилитель, сумматор, интегратор с ограничением и мультиплексор. Эта субструктура имитирует изменение уровня наполненности приемного бункера системы разгрузки.

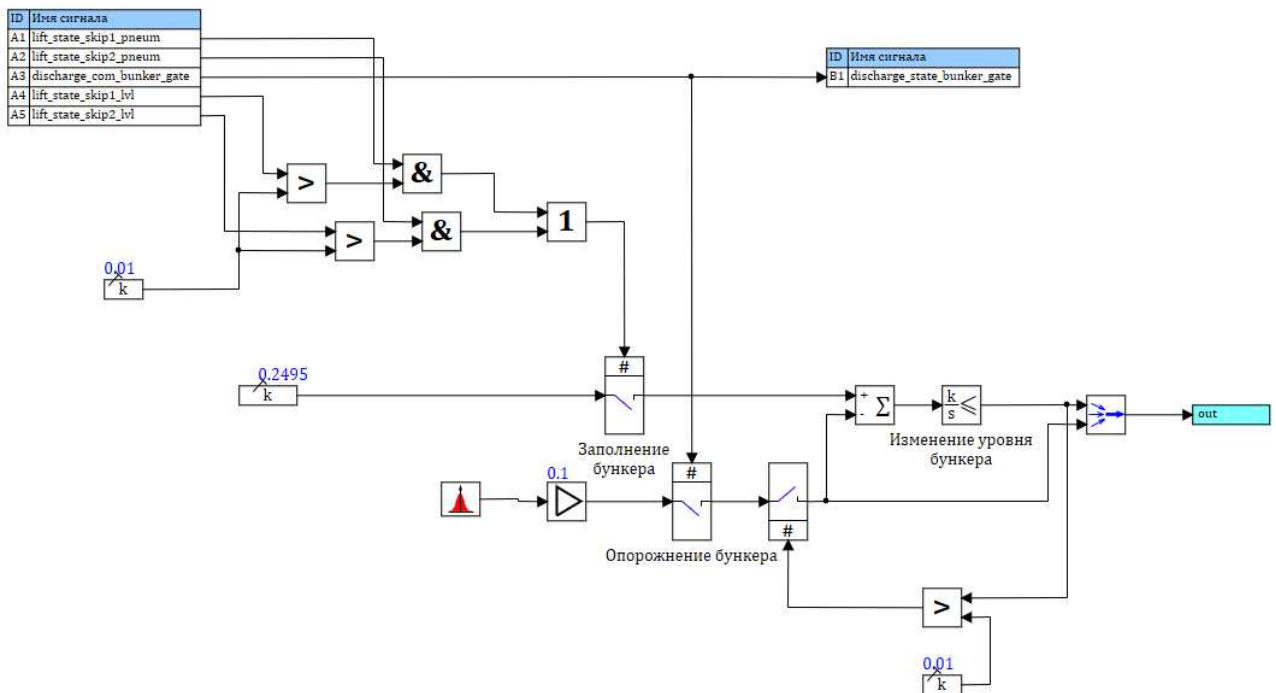


Рисунок 3.12 – Субструктура «Изменение уровня бункера»

На рисунке 3.13 субструктура «Разгрузка» имеет списки входных и выходных сигналов, 7 ключей, блок сравнения, сумматор, интегратор с ограничением, демультимплексор и мультиплексор. Эта субструктура подсчитывает успешно разгруженный материал.

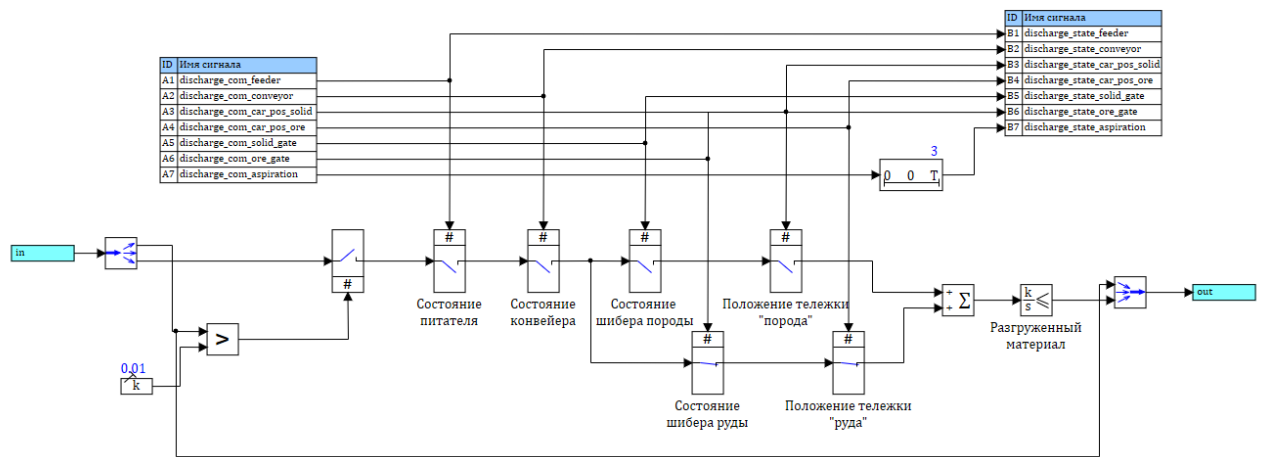


Рисунок 3.13 – Субструктура «Разгрузка»

Модуль управления системой загрузки, представленный на рисунке 3.14, состоит из списков входных и выходных сигналов, 7 ключей, блоков задержки и блока программирования и блока дизъюнкции. Проект отвечает за работу всех ключей проекта системы загрузки в автоматическом режиме работы и в ручном режиме работы. Блок программирования отвечает за автоматический режим работы средств транспортировки системы загрузки.

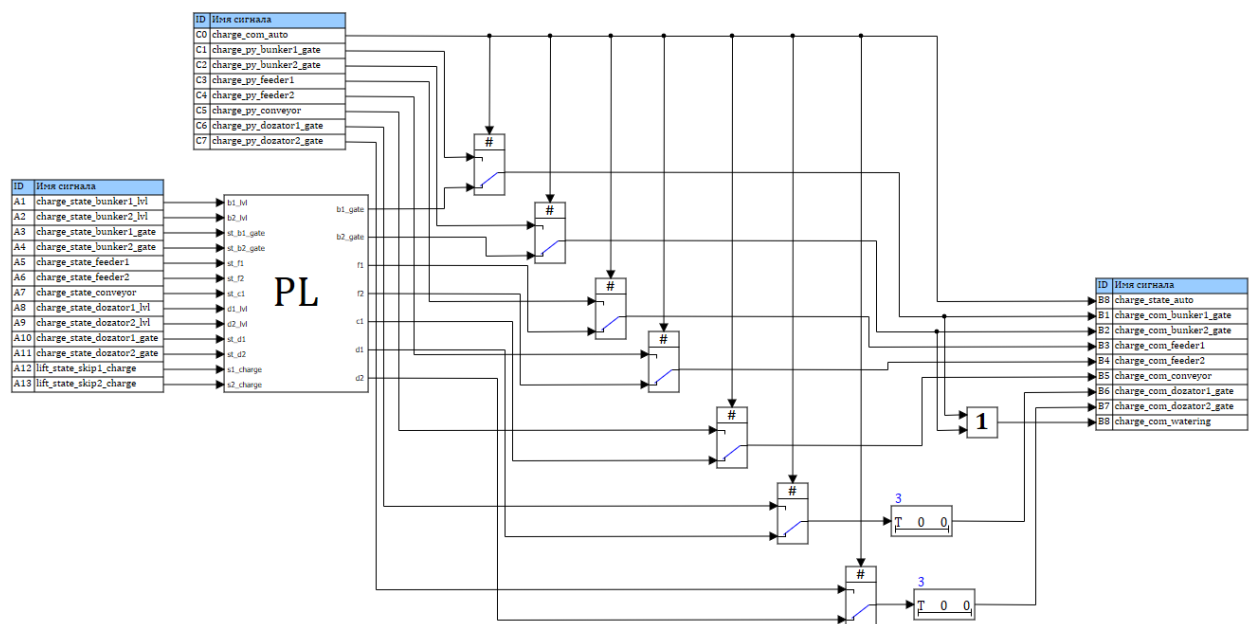


Рисунок 3.14 – Модуль управления системой загрузки

Модуль управления подъемной установкой, показанный на рисунке 3.15, состоит из списков входных и выходных сигналов, 2 блоков программирования, 6 ключей и 4 блока задержки. Проект отвечает за работу всех ключей проекта подъемной установки в автоматическом режиме работы и в ручном режиме работы. Нижний блок программирования отвечает за автоматический режим работы средств транспортировки подъемной установки, а верхний – за ручной режим работы.

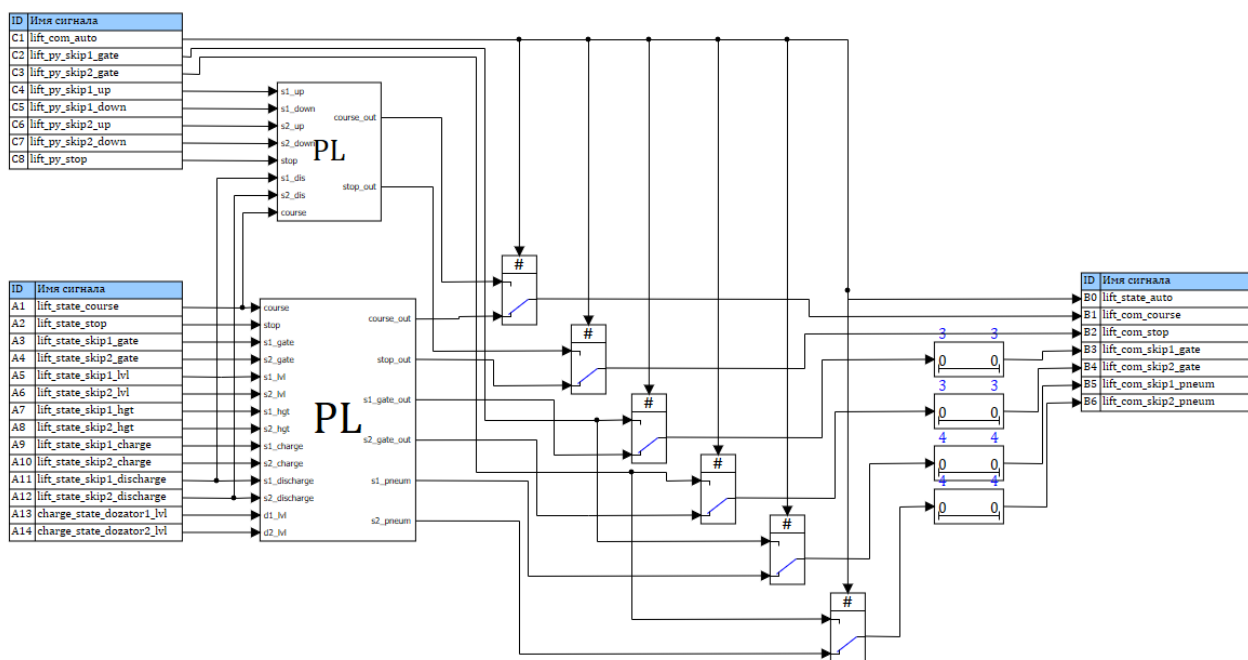


Рисунок 3.15 – Модуль управления подъемной установки

Модуль управления системой разгрузки представлен на рисунке 3.16 и состоит из списков входных и выходных сигналов, 7 ключей и блока программирования. Проект отвечает за работу всех ключей проекта системы разгрузки в автоматическом и в ручном режиме работы. Блок программирования отвечает за автоматический режим работы средств транспортировки системы разгрузки.

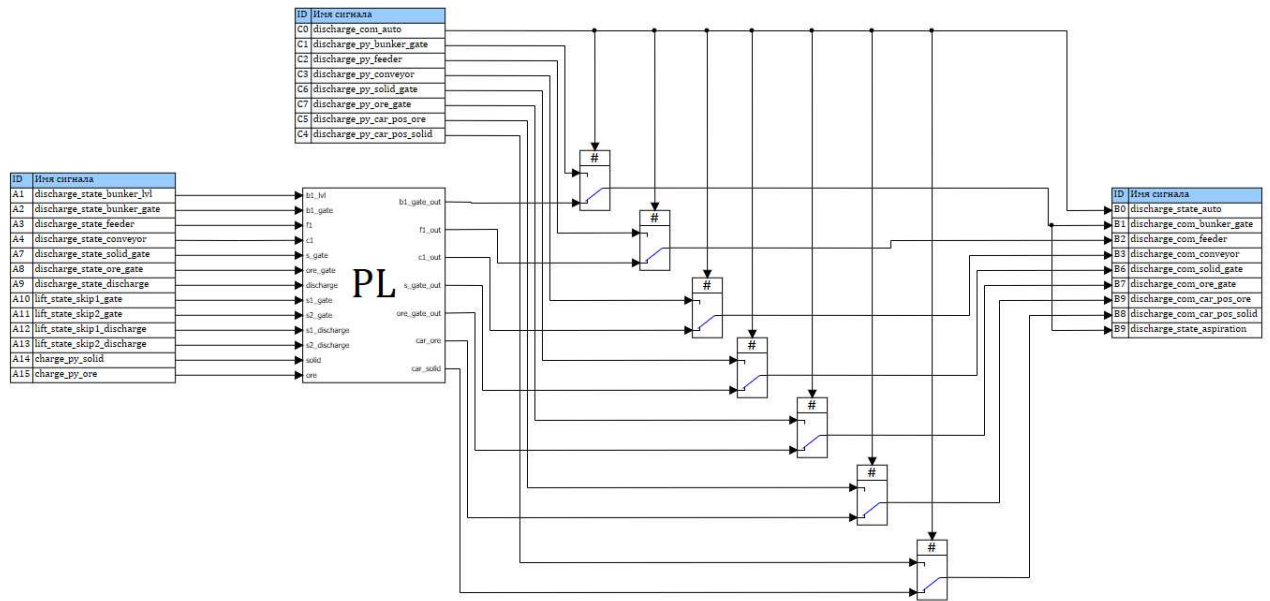


Рисунок 3.16 – Модуль управления системой разгрузки

Пульс контроля диспетчера, изображенный на рисунке 3.17 создан для обеспечения контроля системы транспортировки горной массы с горизонта загрузки на склад. На нем отображены положения средств транспортировки, положение шиберов и затворов, состояние работы оборудования, текущий уровень заполненности бункеров, дозаторов, скип 1 и 2.

В зависимости от состояния изменяется цвет соответствующего оборудования. Желтый цвет означает отключенное состояние или закрытое положение. Зеленый цвет – включенное состояние или открытое положение. Отображение режима работы, уровень наполненности и высота скип изменяет текстовую информацию.

Кнопка «Графики» открывает окно графиков (рис. 3.17), на котором показаны графики заполнения дозаторов, приемного бункера и прибытие скип на горизонт загрузки или разгрузки.

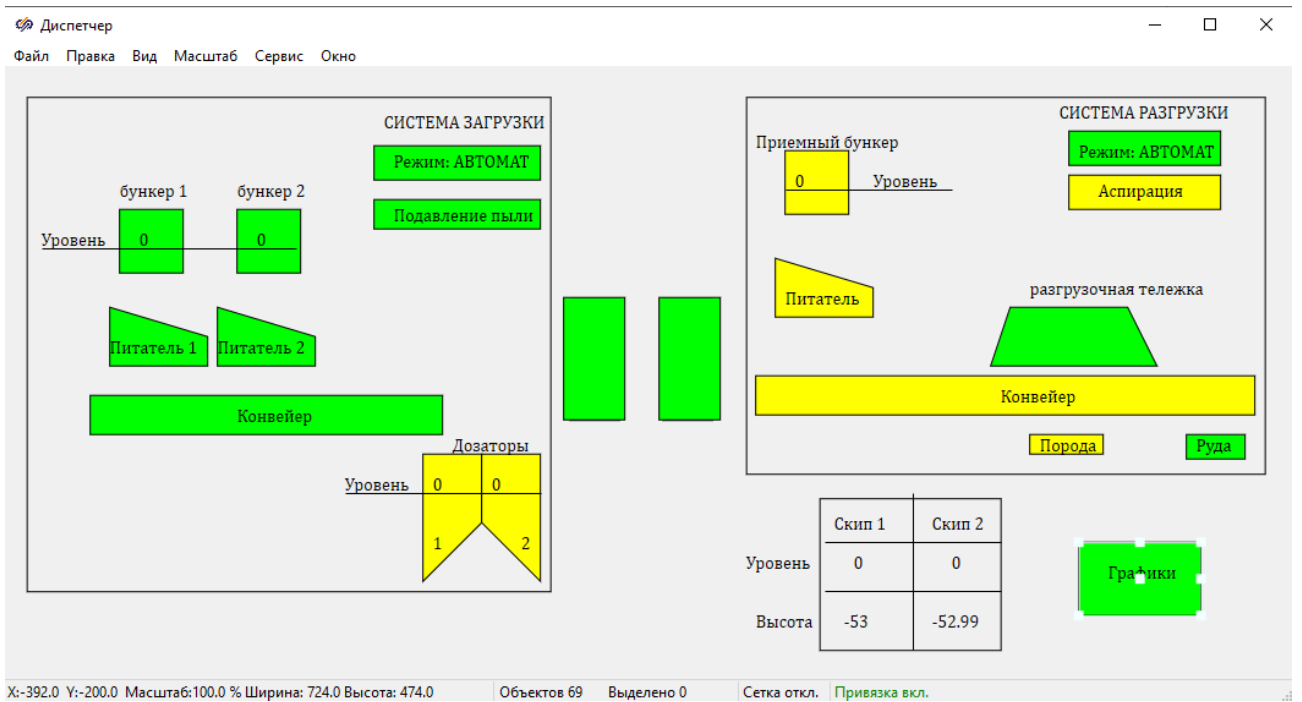


Рисунок 3.17 – Блок диспетчера

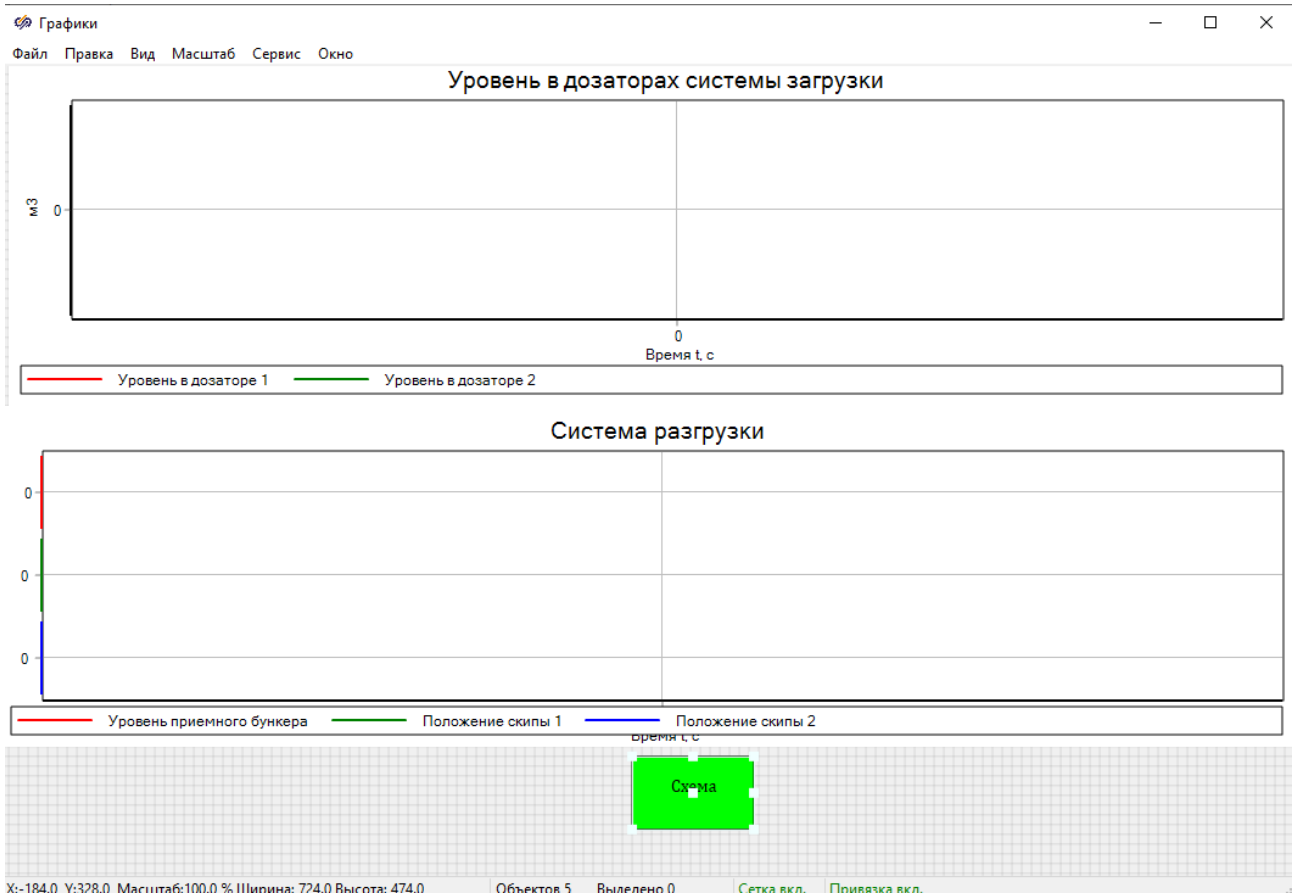


Рисунок 3.18 – Окно графиков

Блок оператора системы загрузки создан для обеспечения контроля и управления средствами транспортировки системы загрузки. Если режим управления стоит в положение «АВТОМАТ», то управления механизмами с помощью кнопок заблокировано, управления средствами транспортировки происходит в автоматическом режиме. Круговые индикаторы отображают текущее состояние оборудования. Желтый цвет круговых индикаторов означает отключенное состояние или закрытое положение. Загружаемый материал изменяется переключением кнопки. Круговые индикаторы скипов на загрузке белым цветом показывают отсутствие соответствующих скип, зеленым – присутствие.

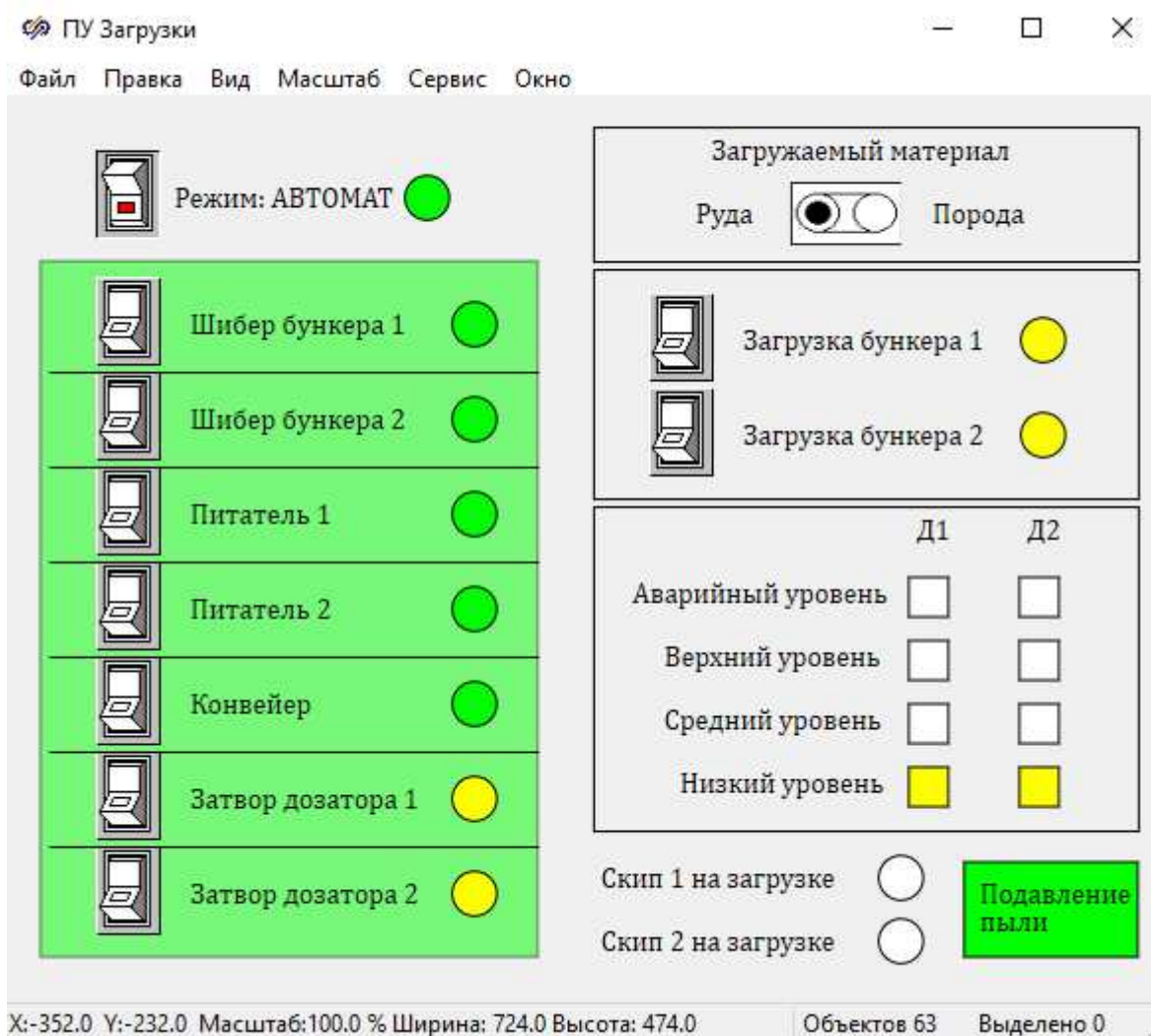


Рисунок 3.19 – Пуль управления системой загрузки

Пульт управления подъемной установкой разработан для обеспечения контроля и управления положением скип 1 и 2. Если режим управления стоит в положение «АВТОМАТ», то управления механизмами с помощью кнопок заблокировано, управления скипами происходит в автоматическом режиме.

В зависимости от уровня заполненности скипы 1 и 2 на пульте управления меняют цвет. Когда скип пустой, то он окрашивается в белый цвет. При среднем уровне заполненности – желтый, при верхнем – зеленый, при аварийном – красный. Стрелки являются индикаторами и переключателями направления одновременно. Если стрелка окрашена в зеленый цвет, значит скипа движется в направлении указывающих стрелкой. Если стрелка обведена сплошной линией, значит выбрано направление, указывающее стрелкой. Состояние затворов, задвижек, положение и высота скип указывается в окне состояния под изображением скип.

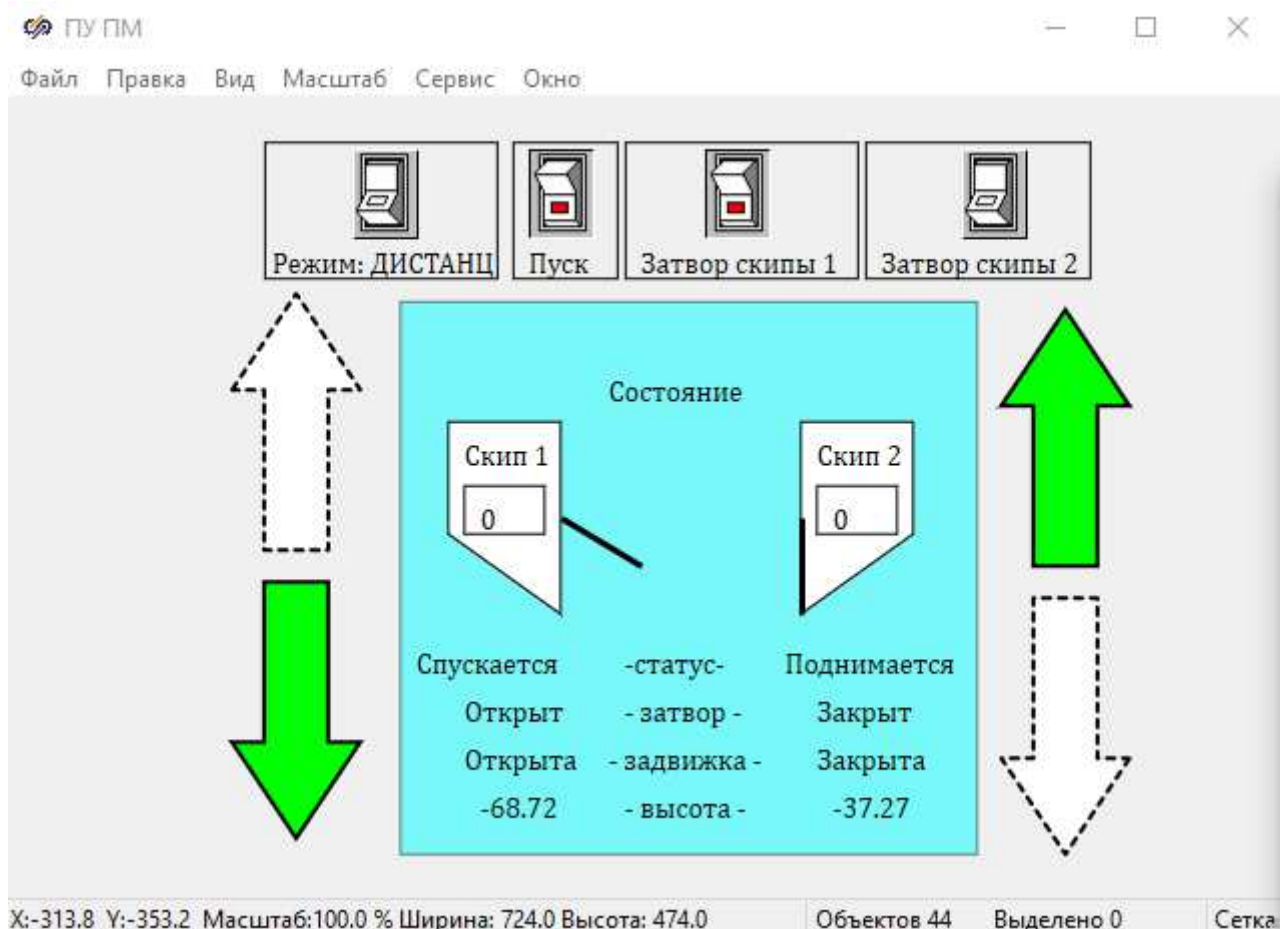


Рисунок 3.20 – Пульт управления подъемной установкой

Пульт управления системой разгрузки разработан для обеспечения контроля и управления средствами транспортировки системы разгрузки. Если режим управления стоит в положение «АВТОМАТ», то управления механизмами с помощью кнопок заблокировано, управления скипами происходит в автоматическом режиме.

На пульте изображена мнемосхема системы разгрузки. Круговые индикаторы отображают текущее состояние оборудования. Желтый цвет круговых индикаторов означает отключенное состояние или закрытое положение. Загружаемый материал изменяется переключением кнопки. В зависимости от состояния изменяется цвет соответствующего оборудования. Желтый цвет означает отключенное состояние или закрытое положение. Зеленый цвет – включенное состояние или открытое положение. Отображение режима работы, статус присутствия скип 1 и 2 изменяет текстовую информацию. Белый цвет скип обозначает отсутствие скип на горизонте разгрузки, зеленый – присутствие.

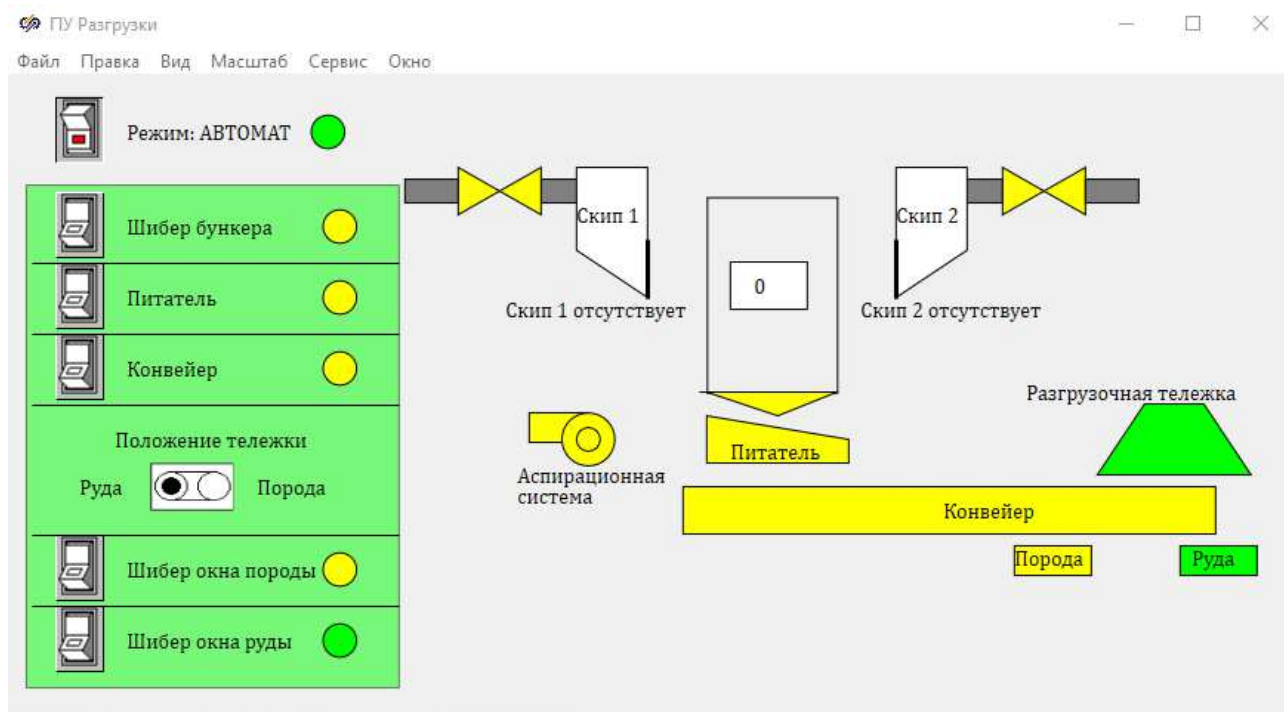


Рисунок 3.21 – Пульт управления системой разгрузки

ЗАКЛЮЧЕНИЕ

В ходе выпускной квалификационной работы был изучен технологический процесс транспортировки горной массы из шахты на поверхность. Были изучены используемые средства транспортировки горной массы из шахты на поверхность и описаны в полном объеме. Разработана имитационная модель автоматизированной системы управления транспортировкой горной массы со средствами транспортировки, визуализирующая работу механизмов и имитирующая их принципы работы. Тестирование показало работоспособность модели. Разработаны алгоритмы автоматического управления и удобный человеко-машинный интерфейс в среде динамического моделирования *SimInTech*. Алгоритмы достаточно надежны, что гарантирует безотказную и эффективную работу автоматизированного процесса.

Задачи, определенные в процессе выпускной квалификационной работы, выполнены в объеме. Значит поставленная цель достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Компьютерное моделирование технологических процессов / Тамбовский государственный технический университет [Электронный ресурс]. – Режим доступа: <http://tstuisman.tstu.ru/pdf/lecture3.pdf>

2 АСУТП / Альянс Автоматика [Электронный ресурс]. – Режим доступа: <https://a-automation.ru/asu-tp/>

3 Стационарные машины – Рудничные подъемные установки: учебник для вузов / А. П. Гришко. – Москва: Издательство МГГУ, 2006. – т.1. – С. 10-11

4 Рудничный транспорт: учебник / Ю. С. Пухов. – Москва: Недра, 1991. – 463 с.

5 Технические характеристики вибротранспортных машин на сайте refwin.ru [Электронный ресурс]. – Режим доступа: <http://refwin.ru/739857774.html>.

6 Питатели. Устройство и работа // Журнал спецтехники [Электронный ресурс]. – Режим доступа: <https://spec-machine.ru/stroitelnaya/pitateli-ustrojstvo-i-rabota/>

7 Транспортные машины и оборудование шахт и рудников // Портал студенческих и научных материалов Ozlib.com [Электронный ресурс]. – Режим доступа: https://ozlib.com/862024/tehnika/transportnye_mashiny_i_oborudovanie_shaht_i_rudnikov

8 Васильев, К. А. Транспортные машины и грузоподъемное оборудование обогатительных фабрик: учеб. пособие для студентов вузов, обучающихся по специальности "Обогащение полез. ископаемых" направления подгот. дипломир. специалистов "Горное дело" / К. А. Васильев, А. К. Николаев, К. Г. Сазонов. - Санкт-Петербург: Наука, 2006. – 358 с.

9 Оборудование для АСУТП // Сайт группы компаний «Industrial PC to You» [Электронный ресурс]. – Режим доступа: <https://ipc2u.ru/catalog/promyshlennaya-avtomatizaciya/>

10 ГОСТ 19.701-90 Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. – введ. 01.01.92 – Москва: Стандартинформ, 2010. – 24 с.

11 Справочная система SimInTech [Электронный ресурс]. – Режим доступа: <http://help.simintech.ru>

12 Среда динамического моделирования технических систем *SimInTech* / Б. А. Карштанов, О. С. Козлов, Е. А. Шабаев, А. М. Щекатуров. – Москва: ДМК Пресс, 2017. – 424 с.

13 Мызникова, В. А. Построение нечетких регуляторов для систем управления автономных объектов в среде *SimInTech* / В. А. Мызникова, В. В. Устименко, А. В. Чубарь // Космические аппараты и технологии. – 2019. – №1 – С.6

14 Пожаркова, И.Н. Моделирование технологического процесса в среде визуального моделирования *SimInTech* / И.Н. Пожаркова, А.В. Чубарь, И.А. Грищенко, Е.Ю. Трояк // Сибирский пожарно-спасательный вестник. – 2018. – №2 (9). – С.29-37.

15 СТО 4.2 07 2014. Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Взамен СТО 4.2 07 2012; дата введ. 09.01.2014. – Красноярск, 2014. – 60 с.

16 ГОСТ 2.721-74. Единая система конструкторской документации. Обозначения условные графические в схемах. Обозначения общего применения. – Взамен ГОСТ 2.721-68, ГОСТ 2.783-69, ГОСТ 2.750-68, ГОСТ 2.751-73; введ. 01.07.1975 – Москва: Стандартинформ, 2008. – 34 с.

17 ГОСТ 2.794-79. Единая система конструкторской документации. Обозначения условные графические. Устройства питающие и дозирующие; введ. 01.01.1981 – Москва: Стандартинформ, 2008. – 4 с.

18 ГОСТ 2.782-96. Единая система конструкторской документации. Обозначения условные графические. Машины гидравлические и пневматические. – Взамен ГОСТ 2.782-68; введ. 01.01.1998 – Москва: Стандартинформ, 2012. – 16 с.

19 ГОСТ 21.208-2013. Система проектной документации для строительства. Автоматизация технологических процессов. Обозначения условные приборов и средств автоматизации в схемах. – Взамен ГОСТ 21.404-85; введ. – 01.11.2014 – Москва: Стандартинформ, 2020. – 34 с.

20 ГОСТ 21.408-93. Система проектной документации для строительства. Автоматизация технологических процессов. Правила выполнения рабочей документации автоматизации технологических процессов. – ВЗАМЕН ГОСТ 21.40485; введ. – 01.11.2014 – Москва: Стандартинформ, 2015. – 30 с.

ПРИЛОЖЕНИЕ А

Скрипт для управления модулем загрузки

```
input b1_lvl, b2_lvl, st_b1_gate, st_b2_gate, st_f1,
      st_f2, st_c1, d1_lvl, d2_lvl, st_d1, st_d2,
s1_charge, s2_charge;
output b1_gate, b2_gate, f1, f2, c1, d1, d2;

var MaxLvl, MinLvl: double;
MaxLvl = 1.8;
MinLvl = 0.01

//-----загрузка дозаторов-----
-----

if (d1_lvl >= MaxLvl) or (d2_lvl >= MaxLvl) then //если верхний
уровень дозаторов
begin
    f1 = false; f2 = false; c1 = false; b1_gate = false; b2_gate =
false;//отключаем питатели и конвейер
end;

if (d1_lvl <= MaxLvl) and (d2_lvl <= MaxLvl)//если уровень дозаторов
не полный
    and (s1_charge = false) and (s2_charge = false) then
begin
    f1 = true; f2 = true; c1 = true; b1_gate = true; b2_gate =
true//включаем питатели и конвейер
end;

//-----загрузка скип-----
-----

if (s1_charge = true) and (d1_lvl >= MaxLvl) and (st_c1 = false)
then
```

```

//если скип1 на загрузке, верхний уровень дозатора и конвейер
отключен
begin
    d1 = true;    //открываем затвор дозатора 1
end;
if (s2_charge = true) and (d2_lvl >= MaxLvl) and (st_c1 = false)
then
//если скип2 на загрузке, верхний уровень дозатора и конвейер
отключен
begin
    d2 = true;    //открываем затвор дозатора 2
end;
if (d1_lvl <= MinLvl) then //если уровень дозатора на нуле
begin
    d1 = false; //закрываем затвор дозатор 1
end;
if (d2_lvl <= MinLvl) then //если уровень дозатора на нуле
begin
    d2 = false; //закрываем затвор дозатор 1
end;

```

ПРИЛОЖЕНИЕ Б

Скрипт для управления модулем ПМ

```
input course, stop, s1_gate, s2_gate, s1_lvl, s2_lvl, s1_hgt,
s2_hgt,
           s1_charge, s2_charge, s1_discharge, s2_discharge,
d1_lvl, d2_lvl;
output course_out, stop_out, s1_gate_out, s2_gate_out, s1_pneum,
s2_pneum;

var s1_ready, s2_ready: boolean;

initialization
s1_ready = true; s2_ready = true;
end;

var MaxLvl, MinLvl: double;
MaxLvl = 1.8; MinLvl = 0.01;

//-----скипы на загрузке-----
-----

if (s1_lvl >= MaxLvl) and (s1_charge) and (d1_lvl <= MinLvl) then
s1_ready = true; //если скип1 заполнен на загрузке, то он готов
if (s2_lvl >= MaxLvl) and (s2_charge) and (d2_lvl <= MinLvl) then
s2_ready = true; //если скип2 заполнен на загрузке, то он готов

if (s1_lvl <= MaxLvl) and (s1_charge) then //если скип1 пуст
begin
    s1_ready = false; //то он не готов к движению
end;
if (s2_lvl <= MaxLvl) and (s2_charge) then
begin
    s2_ready = false; //если скип 2 пуст на загрузке, то он стоит
```

```

end;

//-----скипы на разгрузке-----
//-----

if (s1_lvl <= MinLvl) and (s1_discharge) then //если скип 1 пуст
на разгрузке
begin
    s1_pneum = false; //закреть пневмораспределитель 1
    s1_gate_out = false; // закрыть затвор скипы 1
    s1_ready = true; //готов к движению
end;

if (s2_lvl <= MinLvl) and (s2_discharge) then //если скип 2 пуст
на разгрузке
begin
    s2_pneum = false; //закреть пневмораспределитель 2
    s2_gate_out = false; // закрыть затвор скипы 1
    s2_ready = true; //готов к движению
end;

if (s1_lvl >= MinLvl) and (s1_discharge) then //если скип 1 не
пуст на разгрузке
begin;
    s1_pneum = true; //открыть пневмораспределитель 1
    s1_ready = false; //не готов к движению
    s1_gate_out = true; // открыть затвор скипы 1
end;

if (s2_lvl >= MinLvl) and (s2_discharge) then //если скип 2 не
пуст на разгрузке
begin;
    s2_pneum = true; //открыть пневмораспределитель 2
    s2_ready = false; //не готов к движению
    s2_gate_out = true; // открыть затвор скипы 2
end;

```

```

//-----условия остановки-----
-----

if (s1_charge) and (s2_discharge) then stop_out = true;//если
скип1 на загрузке, скип2 на разгрузке, то стоп

if (s1_discharge) and (s2_charge) then stop_out = true;//если
скип2 на загрузке, скип1 на разгрузке, то стоп

if (d1_lvl <= MinLvl) and (d2_lvl <= MinLvl)//если дозаторы пустые
and (s1_lvl <= MinLvl) and (s2_lvl <= MinLvl) //и скипы
пустые

then stop_out = true; //стоп

//-----условия движения-----
-----

if (s1_ready) and (s2_ready) then//если скип1 и скип2 готовы к
движению

begin

if (s1_charge) and (s2_discharge) then //если скип1 на
загрузке, скип2 на разгрузке

begin

stop_out = false; //начать движение
course_out = false; //скип1 поднимается

end;

if (s2_charge) and (s1_discharge) then //если скип2 на
загрузке, скип1 на разгрузке

begin

stop_out = false; //начать движение
course_out = true; //скип2 поднимается

end;

end;

if ((d1_lvl >= MaxLvl) or (d2_lvl >= MaxLvl))
and not((s1_charge) or (s2_charge)) and (stop) then

```



```
begin
    if(d1_lvl >= MaxLvl) then course_out = true;//скип1 спускается
    if(d2_lvl >= MaxLvl) then course_out = false;//скип2
    спускается
    stop_out = false;//начать движение
end;
```

ПРИЛОЖЕНИЕ В

Скрипт для управления модулем разгрузки

```
input b1_lvl, b1_gate, f1, c1, s_gate, ore_gate, discharge, s1_gate,
s2_gate,
           s1_discharge, s2_discharge, solid, ore;
output  b1_gate_out,  f1_out,  c1_out,  s_gate_out,  ore_gate_out,
car_ore, car_solid;

if (b1_lvl >= 0.01) and (b1_gate = false) then
begin
    b1_gate_out = true;
    f1_out = true;
    c1_out = true;
end;

if(b1_lvl <= 0.01) and (b1_gate = true) then
begin
    b1_gate_out = false;
    f1_out = false;
    c1_out = false;
end;

if (solid) then
begin
    car_ore = false;
    car_solid = true;
    ore_gate_out = false;
    s_gate_out = true;
end;
```

```
if (ore) then
begin
    car_solid = false;
    car_ore = true;
    s_gate_out = false;
    ore_gate_out = true;
end;
```

ПРИЛОЖЕНИЕ Г

Скрипт для управления ПМ вручную

```
input s1_up: boolean, s1_down: boolean, s2_up: boolean, s2_down:
boolean,
           stop: boolean, s1_dis: boolean, s2_dis: boolean,
course: boolean;
output course_out: boolean, stop_out: boolean;

if (s1_up) and (s2_down) then course_out = false;
if (s2_up) and (s1_down) then course_out = true;

if not(stop) then
begin
    if not(s1_dis) and not(s2_dis) then stop_out = false;

    if (s1_dis) and not(course) then stop_out = true;
    if (s2_dis) and (course) then stop_out = true;

    if (s1_dis) and (course) then stop_out = false;
    if (s2_dis) and not(course) then stop_out = false;
end;
if (stop) then stop_out = true;
```

ПРИЛОЖЕНИЕ Д

Скрипт окна анимации «Диспетчер»

```
var yellow, green, white, red, MaxLvl, EmergencyLvl;
yellow = 65535; green = 65280; white = 16777215; red = 255;
MaxLvl = 1.8; EmergencyLvl = 2;

//-----БУНКЕРЫ-----
-----

if (charge_state_bunker1_gate = 1) then BUNKER1.Color = green;
if (charge_state_bunker1_gate = 0) then BUNKER1.Color = yellow;

if (charge_state_bunker2_gate = 1) then BUNKER2.Color = green;
if (charge_state_bunker2_gate = 0) then BUNKER2.Color = yellow;

if (discharge_state_bunker_gate = 1) then BUNKER3.Color = green;
if (discharge_state_bunker_gate = 0) then BUNKER3.Color = yellow;

BUNKER1_LVL.Text = FloatToStr(roundstep(charge_state_bunker1_lvl,
0.01));
BUNKER2_LVL.Text = FloatToStr(roundstep(charge_state_bunker2_lvl,
0.01));
BUNKER3_LVL.Text = FloatToStr(roundstep(discharge_state_bunker_lvl,
0.01));

//-----ПИТАТЕЛИ-----
-----

if (charge_state_feeder1 = 1) then FEEDER1.Color = green;
if (charge_state_feeder1 = 0) then FEEDER1.Color = yellow;

if (charge_state_feeder2 = 1) then FEEDER2.Color = green;
if (charge_state_feeder2 = 0) then FEEDER2.Color = yellow;
```

```

if (discharge_state_feeder = 1) then FEEDER3.Color = green;
if (discharge_state_feeder = 0) then FEEDER3.Color = yellow;
//-----СКИПЫ-----
-----

SKIP1.Height = lift_state_skip1_lvl*50;

if paintstep then SKIP1.Points = [(100 , -23 -
lift_state_skip1_lvl*25 - (lift_state_skip1_hgt)*2), (125 ,
350), (100 , 300)]; ///[(170 , 350), (195 , 350), (170 , 300)]

Rectangle.Points = [(100 , -75 - lift_state_skip1_hgt*2), (125 ,
350), (100 , 300)];

SKIP2.Height = lift_state_skip2_lvl*50;

if paintstep then SKIP2.Points = [(170 , -23 -
lift_state_skip2_lvl*25 - (lift_state_skip2_hgt)*2), (195 ,
350), (170 , 300)];

Rectangle1.Points = [(170 , -75 - lift_state_skip2_hgt*2), (195 ,
350), (170 , 300)];

SKIP1_LVL.Text = FloatToStr(roundstep(lift_state_skip1_lvl, 0.01));
SKIP2_LVL.Text = FloatToStr(roundstep(lift_state_skip2_lvl, 0.01));

SKIP1_HGT.Text = FloatToStr(roundstep(lift_state_skip1_hgt, 0.01));
SKIP2_HGT.Text = FloatToStr(roundstep(lift_state_skip2_hgt, 0.01));
//-----КОНВЕЙЕР-----
-----

if (charge_state_conveyor = 1) then CONVEYOR1.Color = green;
if (charge_state_conveyor = 0) then CONVEYOR1.Color = yellow;

if (discharge_state_conveyor = 1) then CONVEYOR2.Color = green;
if (discharge_state_conveyor = 0) then CONVEYOR2.Color = yellow;
//-----ДОЗАТОРЫ-----
-----

if (charge_state_dozator1_gate = 1) then DOZATOR1.Color = green;
if (charge_state_dozator1_gate = 0) then DOZATOR1.Color = yellow;

```

```

if (charge_state_doзатор2_gate = 1) then DOZATOR2.Color = green;
if (charge_state_doзатор2_gate = 0) then DOZATOR2.Color = yellow;

DOZATOR1_LVL.Text = FloatToStr(roundstep(charge_state_doзатор1_lvl,
0.01));

DOZATOR2_LVL.Text = FloatToStr(roundstep(charge_state_doзатор2_lvl,
0.01));

//-----разгрузочная тележка-----
-----

if (discharge_state_car_pos_ore) then
begin
    Polygon.Points = [(504 , 24),(640 , 24),(616 , -24),(520 , -
24),(520 , -24)];
end;

if (discharge_state_car_pos_solid) then
begin
    Polygon.Points = [(424 , 24),(560 , 24),(536 , -24),(440 , -
24),(440 , -24)];
end;

if (discharge_state_ore_gate) then GATE_ORE.Color = green;
if not(discharge_state_ore_gate) then GATE_ORE.Color = yellow;
if (discharge_state_solid_gate) then GATE_SOLID.Color = green;
if not(discharge_state_solid_gate) then GATE_SOLID.Color = yellow;

//-----СИГНАЛЫ-----
-----

{if (charge_state_doзатор1_lvl >= MaxLvl) then DOZATOR1_UPLVL.Color
= yellow;

if (charge_state_doзатор1_lvl < MaxLvl) then DOZATOR1_UPLVL.Color =
white;

if (charge_state_doзатор2_lvl >= MaxLvl) then DOZATOR2_UPLVL.Color
= yellow;

```

```

if (charge_state_dozator2_lvl < MaxLvl) then DOZATOR2_UPLVL.Color =
white;

if      (charge_state_dozator1_lvl      >=      EmergencyLvl)      then
DOZATOR1_ALARM.Color = red;

if      (charge_state_dozator2_lvl      >=      EmergencyLvl)      then
DOZATOR2_ALARM.Color = red;}

if (charge_state_auto) then
begin
    FillRect.Color = green; TextLabel.Text = "Режим: АВТОМАТ";
end;
if not(charge_state_auto) then
begin
    FillRect.Color = yellow; TextLabel.Text = "Режим: ДИСТАНЦ";
end;
if (discharge_state_auto) then
begin
    FillRect1.Color = green; TextLabel27.Text = "Режим: АВТОМАТ";
end;
if not(discharge_state_auto) then
begin
    FillRect1.Color = yellow; TextLabel27.Text = "Режим: ДИСТАНЦ";
end;

if (SENSOR_PLOT.LeftButton) then btn_plot = true;

if (btn_plot) then minimizeform;
if not(btn_plot) then
begin
    normalizeform;
    bringtofrontform;
end;

```



```
//подавление пыли
if (charge_state_watering) then FillRect3.Color = green;
if not(charge_state_watering) then FillRect3.Color = yellow;

if (discharge_state_aspiration) then FillRect4.Color = green;
if not(discharge_state_aspiration) then FillRect4.Color = yellow;
```

ПРИЛОЖЕНИЕ Е

Скрипт окна анимации «ПУ загрузки»

```
var yellow, green, white, red, MaxLvl, UpLvl, AveLvl, LowLvl;
yellow = 65535; green = 65280; white = 16777215; red = 255;
MaxLvl = 2; UpLvl = 1.8; AveLvl = 1; LowLvl = 0.1;
initialization
BTN_BUNKER1.Down = 0; BTN_BUNKER1.Value = 0;
BTN_BUNKER2.Down = 0; BTN_BUNKER2.Value = 0;
BTN_GATE_BUNKER1.Down = 0; BTN_GATE_BUNKER1.Value = 0;
BTN_GATE_BUNKER2.Down = 0; BTN_GATE_BUNKER2.Value = 0;
BTN_FEEDER1.Down = 0; BTN_FEEDER1.Value = 0;
BTN_FEEDER2.Down = 0; BTN_FEEDER2.Value = 0;
BTN_CONVEYOR.Down = 0; BTN_CONVEYOR.Value = 0;
BTN_DOZATOR1.Down = 0; BTN_DOZATOR1.Value = 0;
BTN_DOZATOR2.Down = 0; BTN_DOZATOR2.Value = 0;
BTN_AUTO.Down = 1;    BTN_AUTO.Value = 1;
end;

//бункеры
if (charge_state_bunker1) then CHARGE_BUNKER1.Color = green;
if not(charge_state_bunker1) then CHARGE_BUNKER1.Color = yellow;

if (charge_state_bunker2) then CHARGE_BUNKER2.Color = green;
if not(charge_state_bunker2) then CHARGE_BUNKER2.Color = yellow;

if (charge_state_bunker1_gate) then GATE_BUNKER1.Color = green;
if not(charge_state_bunker1_gate) then GATE_BUNKER1.Color = yellow;

if (charge_state_bunker2_gate) then GATE_BUNKER2.Color = green;
```

```

if not(charge_state_bunker2_gate) then GATE_BUNKER2.Color = yellow;

charge_com_bunker1 = BTN_BUNKER1.Value;
charge_com_bunker2 = BTN_BUNKER2.Value;

//питатели
if (charge_state_feeder1) then FEEDER1.Color = green;
if not(charge_state_feeder1) then FEEDER1.Color = yellow;

if (charge_state_feeder2) then FEEDER2.Color = green;
if not(charge_state_feeder2) then FEEDER2.Color = yellow;

//конвейер
if (charge_state_conveyor) then CONVEYOR.Color = green;
if not(charge_state_conveyor) then CONVEYOR.Color = yellow;

//дозаторы
if (charge_state_dozator1_gate) then GATE_DOZATOR1.Color = green;
if not(charge_state_dozator1_gate) then GATE_DOZATOR1.Color =
yellow;

if (charge_state_dozator2_gate) then GATE_DOZATOR2.Color = green;
if not(charge_state_dozator2_gate) then GATE_DOZATOR2.Color =
yellow;

if (charge_state_dozator1_lvl >= MaxLvl) then
begin
    MAXLVL_D1.Color = red; UPLVL_D1.Color = white; AVELVL_D1.Color
= white; LOWLVL_D1.Color = white;
end;

if (charge_state_dozator1_lvl >= UpLvl) and
(charge_state_dozator1_lvl < MaxLvl) then
begin

```

```

        MAXLVL_D1.Color    =    white;    UPLVL_D1.Color    =    yellow;
    AVELVL_D1.Color = white; LOWLVL_D1.Color = white;

end;

if      (charge_state_dozator1_lvl      >=      AveLvl)      and
(charge_state_dozator1_lvl < UpLvl) then

begin

        MAXLVL_D1.Color    =    white;    UPLVL_D1.Color    =    white;
    AVELVL_D1.Color = yellow; LOWLVL_D1.Color = white;

end;

if (charge_state_dozator1_lvl <= LowLvl) then

begin

        MAXLVL_D1.Color    =    white;    UPLVL_D1.Color    =    white;
    AVELVL_D1.Color = white; LOWLVL_D1.Color = yellow;

end;

if (charge_state_dozator2_lvl >= MaxLvl) then

begin

        MAXLVL_D2.Color = red; UPLVL_D2.Color = white; AVELVL_D2.Color
    = white; LOWLVL_D2.Color = white;

end;

if      (charge_state_dozator2_lvl      >=      UpLvl)      and
(charge_state_dozator2_lvl < MaxLvl) then

begin

        MAXLVL_D2.Color    =    white;    UPLVL_D2.Color    =    yellow;
    AVELVL_D2.Color = white; LOWLVL_D2.Color = white;

end;

if      (charge_state_dozator2_lvl      >=      AveLvl)      and
(charge_state_dozator2_lvl < UpLvl) then

begin

        MAXLVL_D2.Color    =    white;    UPLVL_D2.Color    =    white;
    AVELVL_D2.Color = yellow; LOWLVL_D2.Color = white;

end;

if (charge_state_dozator2_lvl <= LowLvl) then

begin

```

```

    MAXLVL_D2.Color = white;    UPLVL_D2.Color = white;
    AVELVL_D2.Color = white;    LOWLVL_D2.Color = yellow;
end;

//руда-порода
if (Button.Value = 0) then
begin
    charge_py_ore = true;
    charge_py_solid = false;
end;
if (Button.Value = 1) then
begin
    charge_py_ore = false;
    charge_py_solid = true;
end;

//СКИПЫ
if (lift_state_skip1_charge) then SKIP1.Color = green;
if not(lift_state_skip1_charge) then SKIP1.Color = white;

if (lift_state_skip2_charge) then SKIP2.Color = green;
if not(lift_state_skip2_charge) then SKIP2.Color = white;

//подавление пыли
if (charge_state_watering) then FillRect1.Color = green;
if not(charge_state_watering) then FillRect1.Color = yellow;

//режим управления
if (charge_state_auto) then
begin
    AUTO.Color = green; FillRect.Color = green;
    TextLabel5.Text = "Режим: АВТОМАТ";
end;

```

```

    BTN_GATE_BUNKER1.Down = 0; BTN_GATE_BUNKER1.Value = 0;
    BTN_GATE_BUNKER2.Down = 0; BTN_GATE_BUNKER2.Value = 0;
    BTN_FEEDER1.Down = 0; BTN_FEEDER1.Value = 0;
    BTN_FEEDER2.Down = 0; BTN_FEEDER2.Value = 0;
    BTN_CONVEYOR.Down = 0; BTN_CONVEYOR.Value = 0;
    BTN_DOZATOR1.Down = 0; BTN_DOZATOR1.Value = 0;
    BTN_DOZATOR2.Down = 0; BTN_DOZATOR2.Value = 0;
end;

if not(charge_state_auto) then
begin
    AUTO.Color = yellow; FillRect.Color = yellow;
    TextLabel5.Text = "Режим: ДИСТАНЦ";
    charge_py_bunker1_gate = BTN_GATE_BUNKER1.Value;
    charge_py_bunker2_gate = BTN_GATE_BUNKER2.Value;
    charge_py_feeder1 = BTN_FEEDER1.Value;
    charge_py_feeder2 = BTN_FEEDER2.Value;
    charge_py_conveyor = BTN_CONVEYOR.Value;
    charge_py_dozator1_gate = BTN_DOZATOR1.Value;
    charge_py_dozator2_gate = BTN_DOZATOR2.Value;
end;

charge_com_auto = BTN_AUTO.Value;

```

ПРИЛОЖЕНИЕ Ж

Скрипт окна анимации «ПУ ПМ»

```
var yellow, green, white, red, MaxLvl, UpLvl, AveLvl, LowLvl;
yellow = 65535; green = 65280; white = 16777215; red = 255;

initialization
BTN_GATE_SKIP1.Down = 0; BTN_GATE_SKIP1.Value = 0;
BTN_GATE_SKIP2.Down = 0; BTN_GATE_SKIP2.Value = 0;
BTN_AUTO.Down = 1; BTN_AUTO.Value = 1;
BTN_START.Down = 0; BTN_START.Value = 1;
end;

//состояние
if (lift_state_course) and not(lift_state_stop) then
begin
    TextLabel3.Text = "Спускается"; TextLabel4.Text =
"Поднимается";
    SKIP1_DOWN.Color = green; SKIP1_UP.Color = white;
    SKIP2_DOWN.Color = white; SKIP2_UP.Color = green;
end;
if not(lift_state_course) and not(lift_state_stop) then
begin
    TextLabel3.Text = "Поднимается"; TextLabel4.Text =
"Спускается";
    SKIP1_DOWN.Color = white; SKIP1_UP.Color = green;
    SKIP2_DOWN.Color = green; SKIP2_UP.Color = white;
end;

//границы стрелок
if (lift_state_course) then
```

```

begin
    SKIP1_DOWN.BorderPenStyle = 0; SKIP1_UP.BorderPenStyle = 1;
    SKIP2_DOWN.BorderPenStyle = 1; SKIP2_UP.BorderPenStyle = 0;
end;

if not(lift_state_course) then
begin
    SKIP1_DOWN.BorderPenStyle = 1; SKIP1_UP.BorderPenStyle = 0;
    SKIP2_DOWN.BorderPenStyle = 0; SKIP2_UP.BorderPenStyle = 1;
end;

//положение
if (lift_state_stop) then
begin
    SKIP1_DOWN.Color = white; SKIP1_UP.Color = white;
    SKIP2_DOWN.Color = white; SKIP2_UP.Color = white;
    TextLabel3.Text = "Нет движения"; TextLabel4.Text = "Нет
движения";
end;

//затвор
if (lift_state_skip1_gate) then TextLabel12.Text = "Открыт";
if not(lift_state_skip1_gate) then TextLabel12.Text = "Закрыт";
if (lift_state_skip2_gate) then TextLabel14.Text = "Открыт";
if not(lift_state_skip2_gate) then TextLabel14.Text = "Закрыт";

//задвижка
if (lift_state_skip1_pneum) then
begin
    Line.Points = [(-40 , -120),(-80 , -144)]; TextLabel13.Text =
"Открыта";
end;

```



```

if not(lift_state_skip1_pneum) then
begin
    Line.Points = [(-80 , -96),(-80 , -144)]; TextLabel13.Text =
"Закрыта";
end;
if (lift_state_skip2_pneum) then
begin
    Line1.Points = [(0 , -128),(40 , -144)]; TextLabel15.Text =
"Открыта";
end;
if not(lift_state_skip2_pneum) then
begin
    Line1.Points = [(40 , -96),(40 , -144)]; TextLabel15.Text =
"Закрыта";
end;

//положение
if (lift_state_stop) and (lift_state_skip1_charge) then
TextLabel3.Text = "На загрузке";
if (lift_state_stop) and (lift_state_skip1_discharge) then
TextLabel3.Text = "На разгрузке";
if (lift_state_stop) and (lift_state_skip2_charge) then
TextLabel4.Text = "На загрузке";
if (lift_state_stop) and (lift_state_skip2_discharge) then
TextLabel4.Text = "На разгрузке";

//скипы
TextLabel16.Text = FloatToStr(roundstep(lift_state_skip1_lvl,
0.01));
TextLabel17.Text = FloatToStr(roundstep(lift_state_skip2_lvl,
0.01));
if (lift_state_skip1_lvl < 0.01) then SKIP1.Color = white;
if (lift_state_skip1_lvl < 1) and (lift_state_skip1_lvl > 0.1) then
SKIP1.Color = yellow;

```

```

if (lift_state_skip1_lvl >= 1.8) and (lift_state_skip1_lvl < 2) then
SKIP1.Color = green;

if (lift_state_skip1_lvl >= 2) then SKIP1.Color = red;

if (lift_state_skip2_lvl < 0.01) then SKIP2.Color = white;

if (lift_state_skip2_lvl < 1) and (lift_state_skip2_lvl > 0.1) then
SKIP2.Color = yellow;

if (lift_state_skip2_lvl >= 1.8) and (lift_state_skip2_lvl <= 2)
then SKIP2.Color = green;

if (lift_state_skip2_lvl >= 2) then SKIP2.Color = red;

//состояние высота
TextLabel19.Text      =      FloatToStr(roundstep(lift_state_skip1_hgt,
0.01));

TextLabel20.Text      =      FloatToStr(roundstep(lift_state_skip2_hgt,
0.01));

//кнопки
lift_com_auto = BTN_AUTO.Value;
if (lift_state_auto) then
begin
    TextLabel21.Text = "Режим: АВТОМАТ";
    BTN_GATE_SKIP1.Down = 0; BTN_GATE_SKIP1.Value = 0;
    BTN_GATE_SKIP2.Down = 0; BTN_GATE_SKIP2.Value = 0;
    BTN_START.Down = 0; BTN_START.Value = 1;
end;
if not(lift_state_auto) then
begin
    TextLabel21.Text = "Режим: ДИСТАНЦ";
    lift_py_stop = BTN_START.Value;
    lift_py_skip1_gate = BTN_GATE_SKIP1.Value;
    lift_py_skip2_gate = BTN_GATE_SKIP2.Value;
    if (SENSOR_SKIP1_UP.LeftButton) then
begin

```

```

        lift_py_skip1_up = 1; lift_py_skip1_down = 0;
        lift_py_skip2_up = 0; lift_py_skip2_down = 1;
end;
if (SENSOR_SKIP1_DOWN.LeftButton) then
begin
    lift_py_skip1_up = 0; lift_py_skip1_down = 1;
    lift_py_skip2_up = 1; lift_py_skip2_down = 0;
end;
if (SENSOR_SKIP2_UP.LeftButton) then
begin
    lift_py_skip1_up = 0; lift_py_skip1_down = 1;
    lift_py_skip2_up = 1; lift_py_skip2_down = 0;
end;
if (SENSOR_SKIP2_DOWN.LeftButton) then
begin
    lift_py_skip1_up = 1; lift_py_skip1_down = 0;
    lift_py_skip2_up = 0; lift_py_skip2_down = 1;
end;
end;
end;

```

ПРИЛОЖЕНИЕ И

Скрипт окна анимации «ПУ разгрузки»

```
var yellow, green, white, red;
yellow = 65535; green = 65280; white = 16777215; red = 255;

initialization
BTN_GATE_BUNKER.Down = 0; BTN_GATE_BUNKER.Value = 0;
BTN_FEEDER.Down = 0; BTN_FEEDER.Value = 0;
BTN_CONVEYOR.Down = 0; BTN_CONVEYOR.Value = 0;
BTN_AUTO.Down = 1;    BTN_AUTO.Value = 1;
BTN_GATE_SOLID.Down = 0; BTN_GATE_SOLID.Value = 0;
BTN_GATE_ORE.Down = 0; BTN_GATE_ORE.Value = 0;
end;

//скипы
if (lift_state_skip1_gate) then SKIP1_GATE.Color = green;
if not(lift_state_skip1_gate) then SKIP1_GATE.Color = yellow;
if (lift_state_skip2_gate) then SKIP2_GATE.Color = green;
if not(lift_state_skip2_gate) then SKIP2_GATE.Color = yellow;

if(lift_state_skip1_discharge) then
begin
    SKIP1.Color = green;
    TextLabel14.Text = "Скип 1 на разгрузке";
end;

if not(lift_state_skip1_discharge) then
begin
    SKIP1.Color = white;
    TextLabel14.Text = "Скип 1 отсутствует";
```

```

end;

if(lift_state_skip2_discharge) then
begin
    SKIP2.Color = green;
    TextLabel15.Text = "Скип 2 на разгрузке";
end;
if not(lift_state_skip2_discharge) then
begin
    SKIP2.Color = white;
    TextLabel15.Text = "Скип 2 отсутствует";
end;

if (lift_state_skip1_pneum) then Line5.Points = [(104 , -96),(72 ,
-120)];
if not(lift_state_skip1_pneum) then Line5.Points = [(72 , -80),(72
, -120)];
if (lift_state_skip2_pneum) then Line6.Points = [(208 , -96),(240 ,
-120)];
if not(lift_state_skip2_pneum) then Line6.Points = [(240 , -80),(240
, -120)];

//приемный бункер
if (discharge_state_bunker_gate) then
begin
    GATE_BUNKER.Color = green
    BUNKER_GATE.Color = green;
end;
if not(discharge_state_bunker_gate) then
begin
    GATE_BUNKER.Color = yellow
    BUNKER_GATE.Color = yellow;
end;

```

```

TextLabel9.Text = FloatToStr(roundstep(discharge_state_bunker_lvl,
0.01));

BUNKER.Height = discharge_state_bunker_lvl * 13.2;
BUNKER.Points = [(156 , -16 - discharge_state_bunker_lvl * 6.8), (200
, -82), (156 , -148)];

//питатель
if (discharge_state_feeder) then
begin
FEEDER.Color = green;
FEEDER1.Color = green;
end;
if not(discharge_state_feeder) then
begin
FEEDER.Color = yellow;
FEEDER1.Color = yellow;
end;

//конвейер
if (discharge_state_conveyor) then
begin
    CONVEYOR.Color = green;
    CONVEYOR1.Color = green;
end;
if not(discharge_state_conveyor) then
begin
    CONVEYOR.Color = yellow;
    CONVEYOR1.Color = yellow;
end;

//разгрузочная тележка

```

```

if (discharge_state_car_pos_solid) then CAR.Points = [(288 ,
40),(392 , 40),(360 , -8),(320 , -8),(320 , -8)];

if (discharge_state_car_pos_ore) then CAR.Points = [(376 , 40),(480
, 40),(448 , -8),(408 , -8),(408 , -8)];

//шибер окна породы
if (discharge_state_solid_gate) then
begin
    GATE_SOLID.Color = green;
    GATE_S.Color = green;
end;
if not(discharge_state_solid_gate) then
begin
    GATE_SOLID.Color = yellow;
    GATE_S.Color = yellow;
end;

//шибер окна руды
if (discharge_state_ore_gate) then
begin
    GATE_ORE.Color = green;
    GATE_O.Color = green;
end;
if not(discharge_state_ore_gate) then
begin
    GATE_ORE.Color = yellow;
    GATE_O.Color = yellow;
end;

//аспирационная система
if (discharge_state_aspiration) then
begin

```

```

        FillRect7.Color = green;
        FillCircle.Color = green;
end;
if not(discharge_state_aspiration) then
begin
    FillRect7.Color = yellow;
    FillCircle.Color = yellow;
end;

//режим управления
discharge_com_auto = BTN_AUTO.Value;
if (discharge_state_auto) then //автоматический режим управления
begin
    AUTO.Color = green; FillRect.Color = green;
    TextLabel.Text = "Режим: АВТОМАТ";
    BTN_GATE_BUNKER.Down = 0; BTN_GATE_BUNKER.Value = 0;
    BTN_FEEDER.Down = 0; BTN_FEEDER.Value = 0;
    BTN_CONVEYOR.Down = 0; BTN_CONVEYOR.Value = 0;
    BTN_GATE_SOLID.Down = 0; BTN_GATE_SOLID.Value = 0;
    BTN_GATE_ORE.Down = 0; BTN_GATE_ORE.Value = 0;
end;
if not(discharge_state_auto) then //дистанционный
begin
    AUTO.Color = yellow; FillRect.Color = yellow;
    TextLabel.Text = "Режим: ДИСТАНЦ";
    discharge_py_bunker_gate = BTN_GATE_BUNKER.Value;
    discharge_py_feeder = BTN_FEEDER.Value;
    discharge_py_conveyor = BTN_CONVEYOR.Value;
    if (BTN_CAR_POS.Value = 0) then
begin
        discharge_py_car_pos_solid = false;
        discharge_py_car_pos_ore = true;

```



```
end;  
if (BTN_CAR_POS.Value = 1) then  
begin  
    discharge_py_car_pos_solid = true;  
    discharge_py_car_pos_ore = false;  
end;  
discharge_py_solid_gate = BTN_GATE_SOLID.Value;  
discharge_py_ore_gate = BTN_GATE_ORE.Value;  
end;
```

ПРИЛОЖЕНИЕ К

Скрипт окна анимации «Графики»

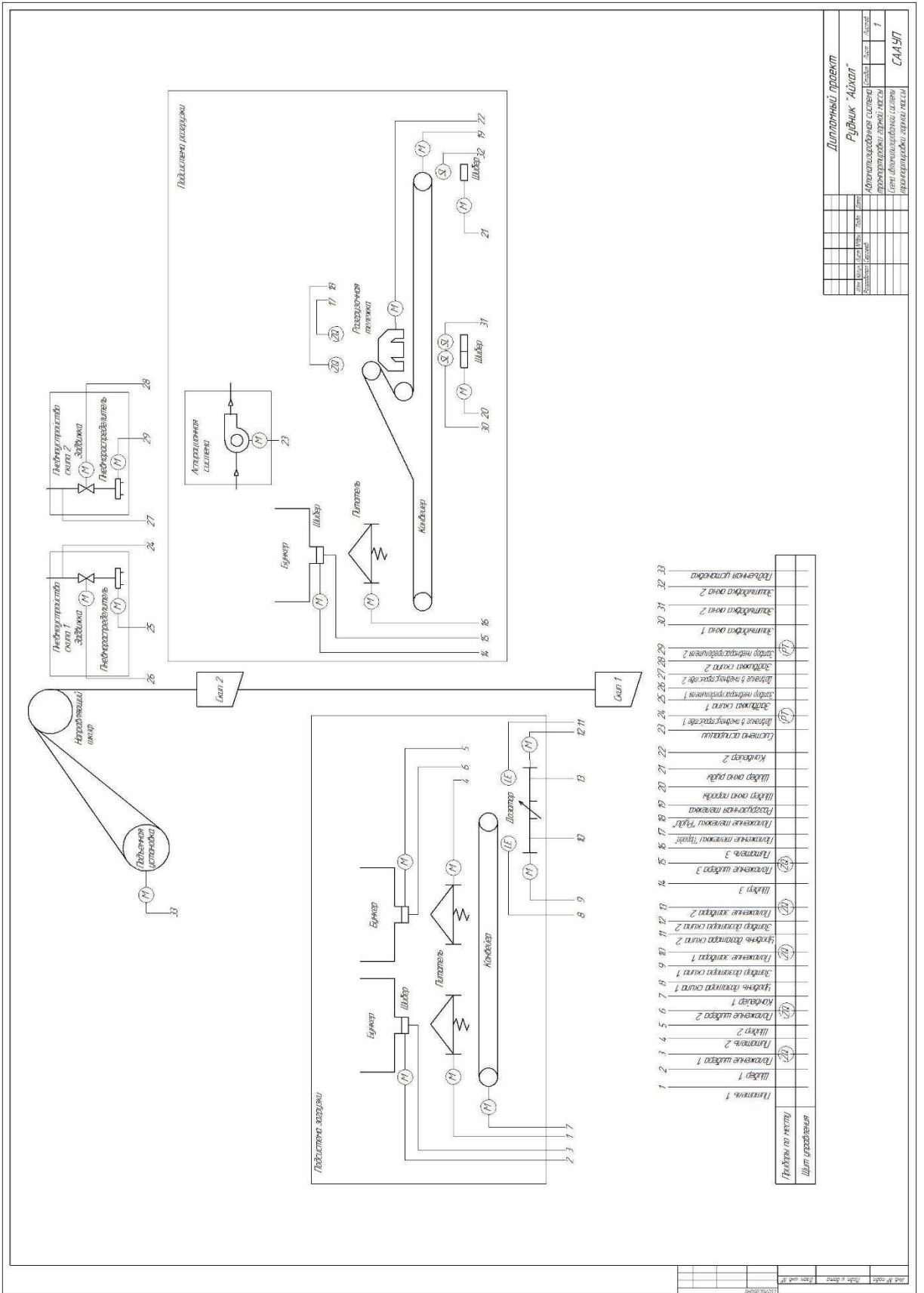
```
initialization
Plot.YValue = [0, 0];
Plot.XValue = [Time, Time];
end;
//график
Plot.YValue = [charge_state_dozator1_lvl,
charge_state_dozator2_lvl];
Plot.XValue = [Time, Time];

Plot1.YValue = [discharge_state_bunker_lvl,
lift_state_skip1_discharge - lift_state_skip1_charge,
lift_state_skip2_discharge - lift_state_skip2_charge];
Plot1.XValue = [time, time, time];

if (SENSOR.LeftButton) then btn_plot = false;
if (btn_plot) then
begin
    normalizeform;
    bringtofrontform;
end;
if not(btn_plot) then
begin
    minimizeform;
end;
```

ПРИЛОЖЕНИЕ Л

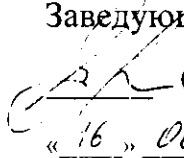
Функциональная схема



Дипломный проект	
Рубчик "Алхал"	
Автоматизация системы	Сектор 1
Автоматизация системы	Сектор 2
Автоматизация системы	Сектор 3
Автоматизация системы	Сектор 4
Автоматизация системы	Сектор 5
Автоматизация системы	Сектор 6
Автоматизация системы	Сектор 7
Автоматизация системы	Сектор 8
Автоматизация системы	Сектор 9
Автоматизация системы	Сектор 10
Автоматизация системы	Сектор 11
Автоматизация системы	Сектор 12
Автоматизация системы	Сектор 13
Автоматизация системы	Сектор 14
Автоматизация системы	Сектор 15
Автоматизация системы	Сектор 16
Автоматизация системы	Сектор 17
Автоматизация системы	Сектор 18
Автоматизация системы	Сектор 19
Автоматизация системы	Сектор 20
Автоматизация системы	Сектор 21
Автоматизация системы	Сектор 22
Автоматизация системы	Сектор 23
Автоматизация системы	Сектор 24
Автоматизация системы	Сектор 25
Автоматизация системы	Сектор 26
Автоматизация системы	Сектор 27
Автоматизация системы	Сектор 28
Автоматизация системы	Сектор 29
Автоматизация системы	Сектор 30
Автоматизация системы	Сектор 31
Автоматизация системы	Сектор 32
Автоматизация системы	Сектор 33

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Систем автоматики, автоматизированного управления и
проектирования»

УТВЕРЖДАЮ
Заведующий кафедрой

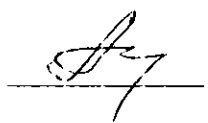

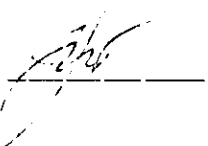
 С.В. Ченцов

«16» 06 2021 г.

БАКАЛАВРСКАЯ РАБОТА

15.03.04 Автоматизация технологических процессов и производств

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ
ТРАНСПОРТИРОВКОЙ ГОРНОЙ МАССЫ**

Руководитель		16.06.2021 г.	доцент, канд. техн. наук А. В. Чубарь
Выпускник		16.06.2021 г.	В. А. Сергеев
Нормоконтролер		16.06.2021 г.	Т.А. Грудинова

Красноярск 2021