

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего  
образования  
«Тольяттинский государственный университет»  
Институт машиностроения  
\_\_\_\_\_  
(наименование института полностью)

Кафедра «Промышленная электроника»  
(наименование)

11.04.04 Электроника и нанoeлектроника  
\_\_\_\_\_  
(код и наименование направления подготовки)

Электронные приборы и устройства  
\_\_\_\_\_  
(направленность (профиль))

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)**

на тему «Разработка мехатронного учебного стенда»

Студент

А.В. Владимиров

(И.О. Фамилия)

\_\_\_\_\_  
(личная подпись)

Научный

руководитель

к.т.н., доцент, А.А. Шевцов

(И.О. Фамилия)

Тольятти 2020

## Содержание

1	Формирование целей и задач работы.....	5
1.1	Постановка вопроса .....	5
1.2	Цели и задачи работы .....	7
1.3	Обзор существующих технических решений .....	8
1.4	Объект исследования .....	10
1.5	Актуальность .....	12
2	Алгоритм работы стенда .....	13
3	Архитектура АСУ ТП.....	15
4	Структура стенда.....	18
4.1	Нижний уровень.....	23
4.2	Средний уровень .....	37
4.3	Верхний уровень .....	45
5	Программная часть.....	48
5.1	Средний уровень .....	48
5.2	Верхний уровень .....	82
6	Интерфейс АРМ .....	93
	Заключение.....	104
	Список используемой литературы и список используемых источников .....	105
	Приложение А Датчик цвета TCS 230.....	108
	Приложение Б Драйвер управления электродвигателем.....	111
	Приложение В Переменные OPC – сервера.....	115

## Введение

Мехатроника — это область науки и техники, основанная на синергетическом объединении узлов точной механики с электронными, электротехническими и компьютерными компонентами, обеспечивающими проектирование и производство качественно новых механизмов, машин и систем с интеллектуальным управлением их функциональными движениями.

Термин мехатроника относится к весьма новым научным направлениям. Первое упоминание данного термина следует из 1969 года, его ввела японская компания Yaskawa Electric.

Благодаря данной науке человечество обрело качественно новые возможности создания и внедрения новых прорывных технологий «завтрашнего дня». Это стало возможным благодаря мировой индустриализации и научно-техническому прогрессу.

Из определения термина можно подчеркнуть следующее, мехатроника объединяет в себе ключевые знания таких научных направлений как «механика», «электротехника» и «электроника». Данные направления появились гораздо раньше, но благодаря научно-техническому прогрессу, их рост развития и глубина изучения достигли уровня который требовал от инженеров комбинированных знаний. В результате произошло синергетическое объединение вышеуказанных направлений. Все эти факторы послужили появлению мехатроники как отдельной науки.

Примерами мехатронных систем могут служить различные устройства, начиная от простых ленточных конвейеров до сложных систем автоматического управления производством.

Современные промышленные предприятия уже не могут обойтись без средств автоматизации производства. Данное утверждение следует из условий спроса, которые диктует мировой рынок. Требования которые предъявляют к продуктам весьма разнообразны, но есть основное касающееся всех без исключения продуктов – получение наивысшего качества при минимально возможной цене.

Сегодня выполнение данного требования невозможно представить без применения новых технологий промышленной автоматизации. Благодаря данным реалиям, человечество получает качественно новые продукты, которые отвечают всем современным требованиям. Можно утверждать что без развития науки и техники человечество не получило бы таких результатов, поэтому мехатроника является новой прогрессивной наукой двадцать первого века.

В данной диссертации рассматриваются основные аспекты создания мехатронного стенда, на котором реализована автоматическая система управления технологическим процессом склада. Мехатронный стенд позволит учащимся изучать различные технологические процессы, познакомиться с современным оборудованием, а так же на основе данного стенда можно создать множество вариаций различных систем.

# 1 Формирование целей и задач работы

## 1.1 Постановка вопроса

В современном мире человечество идет по пути полномасштабной автоматизации. Все существующие предприятия сейчас используют, в той или иной степени, системы автоматического управления (САУ), с различным уровнем интеграции в технологический процесс.

В классическом понимании процесс автоматизации предполагает переход человеческого труда к машинному. Т.е. происходит следующее, функции которые выполнял человек передаются устройствам, приборам, технологическим установкам, и т.д.

Основной причиной внедрения автоматизации для производства является положительный экономический результат, который достигается за счет следующих гарантированных преимуществ:

- Повышение качества продукции;
- Увеличение производительности производства;
- Обеспечение достойного уровня безопасности;
- Повышение рентабельности производства.

Автоматизация производства происходит путем внедрения в технологический процесс комплекса программных и аппаратных средств, которые образуют автоматическую систему управления технологическим процессом (АСУ ТП). Многие компании внедрились данную технологию в производство и ощутили положительный эффект. Но также есть компании, которые не просто внедрились АСУ ТП, но и сделали уверенный шаг вперед объединив различные АСУ ТП в одну, получив автоматическую систему управления производством (АСУП). Данные системы служат универсальным инструментом, который позволяет не только получать в оперативном режиме ключевые показатели производства, но эффективно отслеживать и предотвращать возможные негативные ситуации, которые ведут к

снижению основных показателей производства. При использовании АСУП удастся сократить количество технических и административных ошибок к минимальному уровню, а в некоторых случаях и вовсе их исключить. Таким образом данные системы позволяют более точно проводить системный анализ показателей предприятия и как следствие упростить процедуры связанные с управлением производства. Как правило, результатом внедрения АСУП является снижение себестоимости выпускаемого продукта, за счет увеличения производительности, но при этом повышается и качество выпускаемого продукта. Данный факт безусловно является плюсом как для производителя, так и для потребителя.

Но при всех вышеперечисленных положительных эффектах автоматизации перед любым предприятием существует ряд сложных задач, которые возникают при внедрении АСУ ТП.

Первая задача, которую необходимо решить при внедрении АСУ ТП заключается в грамотном проектировании системы. При проектировании АСУ ТП необходимо учесть всевозможные нюансы, связанные с производством конечного продукта. При условии, если техническое задание (ТЗ) на проектирование АСУ ТП, будет учитывать все требования (предъявляемые к конечному продукту), тем выше будет конечный результат, а соответственно качество готовой продукции будет соответствовать всем необходимым стандартам. Также на этапе проектирования необходимо выполнить экономический расчет эффективности данной системы автоматизации. Ключевые требования к АСУ ТП формируются на основании требуемых технических характеристик к готовому продукту и на основе экономического расчета эффективности АСУ ТП. При проектировании необходимо учитывать полноту, корректность и глубину проработки документации готового проекта. Это позволит не тратить время на согласование спорных вопросов при проведении шеф монтажных работ (ШМР) и пуско – наладочных работ (ПНР).

Вторая немаловажная задача заключается в корректности сборки оборудования, которое входит в состав АСУ ТП. Необходимо вести тщательный авторский надзор при проведении ШМР и ПНР. Это позволит устранить все скрытые ошибки допущенные на этапе проектирования.

Третья задача. Необходимо предусмотреть необходимый запас резервного оборудования для бесперебойной работы АСУ ТП. Т.к. простой предприятия приводит в итоге к финансовым потерям. Зачастую на практике формирование эффективного запаса резервного оборудования происходит либо с необоснованно большими финансовыми потерями и большой технической прочностью (ситуация когда закуплено резервное оборудование в размере сопоставимом с размером самой системы), либо напротив с большой финансовой выгодой и малым запасом технической прочности. Таким образом составляя данный перечень необходимо руководствоваться накопленным опытом работы с оборудованием, а эффективность сформированного запаса покажет только практика использования оборудования.

Но самой значимой задачей для предприятия является кадровый вопрос. Для решения вышеперечисленных задач требования к кадровому составу весьма высоки, причем для эффективного решения этих задач необходим не просто один человек, а целая команда уникальных специалистов с существенным запасом знаний в таких областях науки и техники как механика, электротехника, электроника, программирование.

На сегодняшний день требования к инженерам АСУ ТП достаточно высоки. Инженер АСУ ТП должен обладать глубокими теоретическими и практическими знаниями в области мехатроники. (Мехатроника – это область науки и техники, основанная на синергетическом объединении узлов точной механики с электронными, электротехническими и компьютерными компонентами, обеспечивающими проектирование и производство качественно новых механизмов, машин и систем с интеллектуальным управлением их функциональными движениями.).

## 1.2 Цели и задачи работы

Цель работы: снижение стоимости учебного оборудования с сохранением его дидактической ценности и возможностью использовать как совместно с ПЛК, так и совместно с микроконтроллерными системами управления.

Задачи работы:

- 1) Установление ключевых особенностей имеющегося на рынке учебного оборудования.
- 2) Выработка основных алгоритмов работы разрабатываемого учебного оборудования.
- 3) Разработка физической структуры основных элементов учебного стенда.
- 4) Разработка основных типизированных программных решений для выполнения с помощью разрабатываемого оборудования учебных действий.

На основе данной системы, студенты смогут познакомиться с современным оборудованием, а так же смогут изучить основные принципы и приемы построения систем автоматического управления.

### 1.3 Обзор существующих технических решений

На сегодняшний день крупные промышленные интеграторы, такие как Camozzi и Festo, ведут работу не только по интеграции своих технических решений, но и занимаются вопросом квалификации кадров. Компании проводят семинары, представляя на них свои новинки продукции, а так же проводят обучение персонала на базе своих технических решений.

Компании Camozzi и Festo принимают активное участие в процессе обучения специалистов в области мехатроники. Доказательством данного факта является наличие в их ассортименте такой позиции как учебные стенды. Данные учебные стенды полностью состоят из оборудования указанных брендов.

Для примера рассмотрим учебно-лабораторный комплекс «Мехатронная ячейка» DID-MECH, от компании Camozzi. Мехатронная ячейка DID-MECH представлена на рисунке 1.

Мехатронная ячейка DID-MECH – многофункциональная производственная линия, позволяющая моделировать реальные технологические и производственные процессы посредством синергетического объединения отдельных модулей.



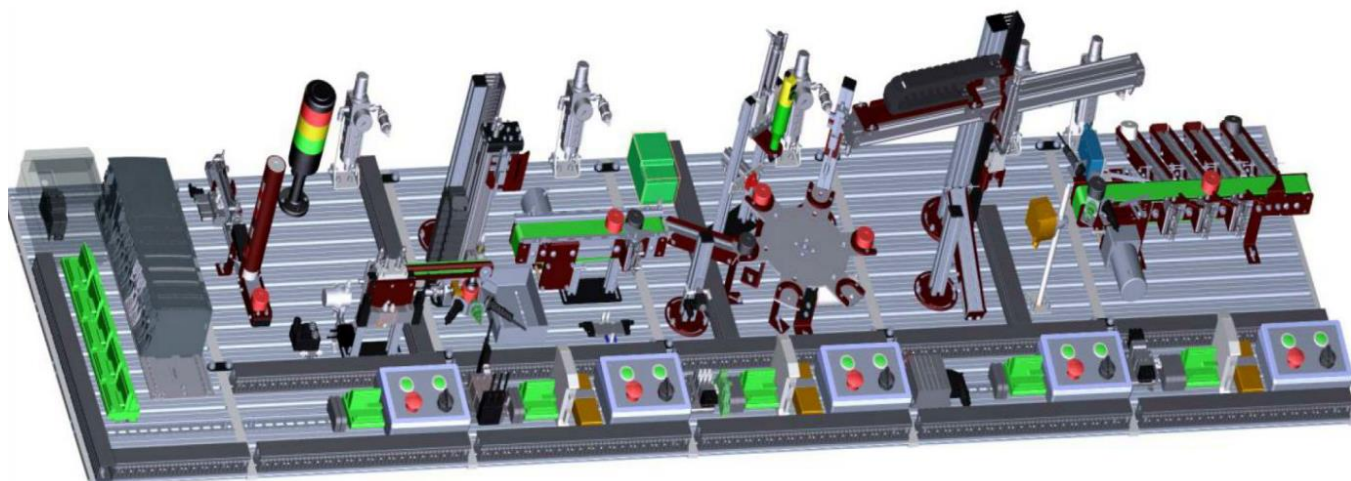


Рисунок 1 – Мехатронная ячейка DID-MECH.

В состав станда входят различные исполнительные механизмы, датчики, светосигнальные индикаторы, кнопки и селектора. Управление оборудованием осуществляется при помощи программируемого логического контроллера (ПЛК). Мехатронная ячейка оснащена ПЛК от компании SIMENS, Simatic S7-1500. Данный ПЛК имеет модульную архитектуру, которая позволяет увеличивать или уменьшать число входов/выходов, путем добавления/отсоединения необходимых модулей. В стандартном исполнении в ПЛК задействуются более 70 входов и более 45 выходов, включая аналоговые входы и выходы.

Так же в качестве примера можно рассмотреть предложение от компании Festo. Компания предлагает целый ассортимент модульных решений учебного оборудования. Модульная производственная система (MPS) компании Festo представлена на рисунке 2.

Данная система содержит различные исполнительные механизмы, датчики, светосигнальные индикаторы, кнопки и селектора, а управление системой осуществляется при помощи ПЛК. Данные станды в стандартном исполнении комплектуются ПЛК от компании SIMENS, Simatic S7-300, но можно также выбрать другие ПЛК от компаний Festo, Allen-Bradley и Mitsubishi.

Данные учебные станды весьма популярны, поскольку география использования продукции данных компаний охватывает весь мир. Продукция компаний Camozzi и Festo уверено зарекомендовала себя как надежный, долговечный и качественный продукт. В данных учебных стандах используются те

же компоненты, которые применяются на производстве, что дает обучающимся наглядное представление о работе используемых устройств и элементов.



Рисунок 2 – MPS система компании Festo.

Данные стенды имеют так же и недостатки. Первым существенным недостатком является их цена. Данный минус обуславливается тем, что компоненты которые используются в стендах, изначально рассчитаны на промышленные нагрузки и отвечают всем стандартам качества. При использовании ПЛК компании SIMENS, необходимо дополнительно приобретать лицензии на программное обеспечение, что так же существенно увеличивает их стоимость. Вторым недостатком данных систем является то, что они ориентированы на свою продукцию, которая является уникальной, и при выходе из строя какой либо действующей единицы, ее невозможно заменить аналогом от другого производителя.

#### 1.4 Объект исследования

На основании вышесказанного было принято решение о разработке мехатронного учебного стенда в котором были бы отражены все современные

технологии построения систем АСУ ТП. Данный стенд будет выполнять роль опытного образца, на основе которого студенты смогут изучить основные принципы построения современных систем АСУ ТП, изучить их архитектуру, приобрести практические навыки работы, модернизировать предложенную систему.

В предложенном мехатронном стенде реализована миниатюрная модель склада. Данное направление было выбрано исходя из существующих вызовов рынка автоматизации. Вопрос автоматизации склада на данный момент весьма интересен поскольку любое производство имеет в своем функциональном цикле такую действующую единицу как склад. На любом предприятии вопрос автоматизации склада выносится на высокий уровень, но он весьма гибкий и составляется из соображений назначения склада и хранимой продукции в нем. Исходя из этого утверждения было принято решение создать на основе мехатронного стенда унифицированную модель АСУ ТП склада.

На данный момент существуют великое множество различных моделей АСУ ТП склада. Это утверждение основывается на специфике хранимого продукта, но главная роль всех без исключения таких АСУ ТП, заключается в непосредственном технологическом процессе склада. Т.е. есть три основные технологические функции склада, это прием, хранение и отгрузка продукта.

При приеме продукции часто возникает потребность в ее сортировке, ведь различная продукция может требовать различные условия хранения. Для примера возьмем продуктовый склад, который осуществляет хранение различных продуктов питания. Допустим на склад поступил груз, состоящий из овощей: картофель, капуста и морковь. Данная продукция поступила для длительного хранения. Для реализации данной задачи, необходимо создать определенные условия хранения данной продукции. Капуста и морковь будут сохранять свои потребительские свойства, если будут храниться при температуре от нуля градусов, до плюс одного градуса цельсия. Картофель для долгого хранения необходимо поместить в среду с температурным диапазоном от плюс трех, до плюс десяти градусов цельсия. Поэтому при поступлении на склад такого груза его необходимо отсортировать и определить места для его правильного хранения с соблюдением температурных

режимов, а так же других санитарно-технических норм. При таком подходе реализация данной задачи, длительного хранения на складе, становится выполнима.

Так же для примера можно рассмотреть склад готовых деталей. Допустим на склад поступил груз шестеренок. Они выполнены из различных материалов (метал, пластик) и имеют различный размер. Такой груз тоже необходимо отсортировать, учитывая данные признаки.

Функция автоматической сортировки – это весьма сложная технологическая задача, потребность в реализации которой возникает на многих предприятиях. Сегодня функция автоматической сортировки успешно реализуется в таких отраслях промышленности как автомобилестроение, почта, сельское хозяйство и др.

В качестве примера в диссертации построим систему автоматического управления складом. Технологическая схема устройства мехатронного станда приведена на рисунке 3.

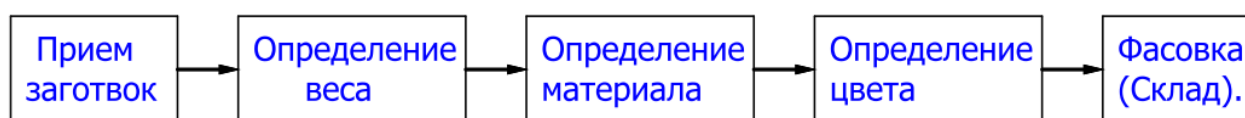


Рисунок 3 – Технологическая схема мехатронного станда.

## 1.5 Актуальность

Исходя из вышесказанного мы можем с уверенностью утверждать то что современный мир нуждается в специалистах высокого уровня области АСУ ТП. Данный факт послужил для принятия решения о создании наглядной системы, в которой будут реализованы современные технологические приемы, построенные на современном, высокотехнологичном оборудовании.

Таким образом, актуальность работы подтверждается необходимостью использования сочетания как теоритической, так и практической части при подготовке кадров, уровня бакалавриата и магистратуры, с использованием актуальных программно-аппаратных дидактических решений универсального характера, находящихся в среднем и ниже среднего ценовых диапазонах.

## 2 Алгоритм работы стенда

Для начала работы, необходимо подготовить оборудование входящее в состав мехатронного стенда. В первую очередь, на стенд необходимо подать питание 380В, о чем будет сигнализировать сигнальная белая лампа (HL1). После необходимо взвести в рабочее положение автоматические выключатели QF1 и QF2. Далее необходимо запустить АРМ (персональный компьютер на котором реализована система управления мехатронным стендом). Вместе с запуском АРМа нужно запустить коммутатор. Необходимо убедиться в завершении загрузки системы, в отсутствии ошибок, наличии связи с ПЛК и наличии связи с модулями, работающими по каналу связи RS-485. Следующим шагом подготовки к работе стенда является запуск компрессора, который можно запустить как с кнопки «Запуск компрессор» (SB2), расположенной на стенде, так и с АРМа. Лампа «Компрессор ПУСК» (HL2) сигнализирует подачу питающего напряжения на схему автоматического управления компрессором. После того как компрессор закачает необходимое давление в ресивер, нужно убедиться в том, что показания давления на манометрах МН1 и МН2 соответствует 5 Бар (в случае если данное значение отличается от указанного, необходимо выставить на указанных манометрах заданное значение давления путем регулировки редуцированных клапанов КР1 и КР2 соответственно). Далее необходимо убедиться в наличии заготовок в приемном устройстве.

Для запуска работы мехатронного стенда, необходимо нажать на кнопку «Запуск работы» (SB3) расположенную на стенде, либо нажать на аналогичную кнопку на АРМ. После однократного нажатия на одну из кнопок «Запуск работы», происходит запуск приемного конвейера. Далее пневмоцилиндр выдает заготовку на конвейер и возвращается в исходное положение. Заготовка помещается на приемный конвейер и перемещается по нему в зону переноса.

После того как заготовку переместили в зону переноса, происходит захват объекта при помощи механизма переноса – механическая рука, которая оснащена специальной присоской.

При помощи присоски происходит захват объекта и далее объект переносится в зону «Определения характеристик заготовок». Первой считываемой характеристикой является вес. Заготовка помещается на весы. После взвешивания заготовки, данная информация передается в ПЛК и на АРМ.

ПЛК обрабатывает данную информацию и далее дает команду на захват объекта механизму переноса. Механизм переноса перемещает объект на технологический стол, на котором определяются следующие характеристики, цвет заготовки и материал, из которого изготовлена заготовка. Датчики, расположенные в данной зоне, производят замеры и передают информацию в ПЛК для ее обработки и передачи данных на АРМ.

Далее, после обработки информации ПЛК дает команду на перенос объекта в зону «Фасовки». После получения данной информации ПЛК принимает решение в какую ячейку на складе переместить объект и передает информацию на АРМ. После чего запускается конвейер, зоны фасовки, и заготовка останавливается перед ячейкой с соответствующими параметрами, а пневмоцилиндр установленный на против ячейки перемещает заготовку в ячейку определенную системой управления.

По окончании вышеуказанных действий, механизм переноса принимает исходное положение и весь цикл запускается сначала.

Так же мехатронный стенд оснащен средствами защиты при необходимости аварийной остановки. Аварийная остановка работы мехатронного стенда осуществляется путем нажатия на кнопку «Стоп» (SB4). После однократного нажатия на кнопку «Стоп» все исполнительные механизмы останавливаются. При двух кратном нажатии данной кнопки (в течении 5с), формируется команда «Reset», которая возвращает все исполнительные механизмы в исходное положение. Последующий запуск работы мехатронного стенда будет осуществлен с самого начала.

Таким образом, реализованный алгоритм работы стенда полностью соответствует технологическому процессу, а так же учитывает выполнение корректной работы оборудования при аварийных ситуациях, обеспечивая безопасность при работе со стендом.

### 3 Архитектура АСУ ТП

Архитектура автоматической системы управления – это совокупность частей (технических средств, коллектива исполнителей и математических методов), образующая организационное комплексное единое целое и обеспечивающая решение требуемого набора задач автоматизации с заданной точностью в пределах ограничения во времени и стоимости.

На сегодняшний день существует огромное множество систем автоматического управления, они осуществляют различные технологические процессы, но их объединяет общая классическая архитектура построения систем АСУ ТП. В классической архитектуре любая система автоматического управления представлена в виде иерархии уровней представляющие собой физическую связь устройств входящих в систему. Обычно выделяют три основных уровня.

- Нижний уровень.
- Средний уровень.
- Верхний уровень.

В нижний уровень входят различные датчики и исполнительные механизмы.

В средний уровень входят различные устройства управления исполнительными механизмами (ЧРП – частотно регулируемые приводы, драйверы двигателей, и др.), модули ввода/вывода, ПЛК и др.

В верхний уровень входят устройства коммуникации и связи с устройствами входящими в средний уровень, автоматические рабочие места АРМ (ПК), различные сервера выполняющие функцию хранения и архивирования информации, и др.

Схема классической архитектуры системы автоматического управления приведена на рисунке 4.

На рисунке 4 представлена схема, отражающая физическую составляющую типовой системы автоматического управления, но помимо физической составляющей присутствует и программная часть. Программная часть, в технически сложных системах, присутствует на всех уровнях иерархии архитектуры АСУ.



Рисунок 4 – Классическая архитектура АСУ

Это утверждение основывается на том, что в современном мире существуют датчики с программной обработкой сигнала, т.е. датчик является технически законченным решением, в котором реализованы функции обработки сигнала в унифицированную величину. Например, компания «Электронстандарт-прибор» имеет в своем ассортименте продукции прибор ИПЭС – ИК/УК – это извещатель пламени. Данный прибор производит измерения в инфракрасном и ультрафиолетовом диапазонах, выдавая в систему сигнал о пожаре. Данный датчик производит самодиагностику и имеет возможность передавать свое состояние в систему, он оснащен двумя дискретными выходами (первый сигнализирует о состоянии датчика (исправен/не исправен), а второй о сигнале пожар), так же датчик оснащен унифицированным токовым выходом (сигнал 4...20мА). Безусловным преимуществом данного прибора является то, что он оснащен каналом связи RS-485 и поддерживает протокол связи Modbus RTU, по которому он может передавать все измеренные параметры. Для корректной работы данного датчика необходимо произвести его настройку при помощи программы поставляемой вместе с прибором. Таким образом, если рассмотреть данный датчик как единичный элемент системы управления, можно сделать вывод о том, что в нем реализована своя технически сложная система контроля и управления. Данное утверждение позволяет говорить о



том, что датчик является технически законченным решением, реализованным под управлением микроконтроллера с соответствующей программой.

Программная часть среднего уровня реализует обработку сигналов нижнего уровня, а также управляет исполнительными механизмами. Но помимо данных функций, на программе среднего уровня лежит функция информационного моста от верхнего уровня к нижнему. Программа среднего уровня зачастую управляет таким устройством как ПЛК. Именно на данном уровне реализуется функция управления технологическим процессом, что является ответственной задачей, при решении которой необходимо учесть все необходимые параметры, особенного внимания требует вопрос быстродействия системы.

Программа верхнего уровня реализует связь между оператором и системой, позволяя оператору оперативно вмешиваться в технологический процесс, она осуществляет функцию архивирования данных. Так же на программу верхнего уровня могут ложиться некоторые вычислительные функции в зависимости от типа и сложности системы. Также существуют системы, в которых вычислительные мощности и логика работы автоматической системой управления полностью лежат на программе верхнего уровня. В таких системах применяются в качестве моста от нижнего к верхнему уровню такие устройства как модули ввода вывода. Данные модули оснащаются чаще всего интерфейсами RS-485 и/или Ethernet.

Архитектура разработанного стенда в полной мере отражает классическое представление АСУ ТП. Архитектура стенда отражена на структурной схеме, приложенной к данной диссертации. Архитектура стенда разделена на три уровня: нижний, средний и верхний.

Таким образом, архитектура системы управления учебного мехатронного стенда полностью соответствует классическому представлению автоматических систем управления. Данная архитектура, отражает в себе представление самых распространенных систем АСУ ТП, которые применяются на производстве, это позволит студентам изучить на практике архитектурные принципы построения систем АСУ ТП.

#### 4 Структура станда

Структура станда соответствует классическому представлению систем автоматизации и имеет определенную иерархию. Структура станда разделена на три уровня: нижний, средний и верхний.

К нижнему уровню относятся исполнительные устройства и датчики, которые входят в состав станда. Как видно на структурной схеме нижний уровень разбит на четыре технологические зоны: прием заготовок, определение характеристик заготовок, фасовка, переноса. Так же к нижнему уровню относится компрессор, он обеспечивает систему сжатым воздухом.

Технологическая зона приема заготовок состоит из устройства выдачи заготовок и первого конвейера. Устройство выдачи заготовок представляет собой цилиндрическую емкость с технологическими отверстиями в своем основании, через которые пневмоцилиндр Ц1 выталкивает заготовки на конвейер 1. Устройство выдачи заготовок приведено на рисунке 5.

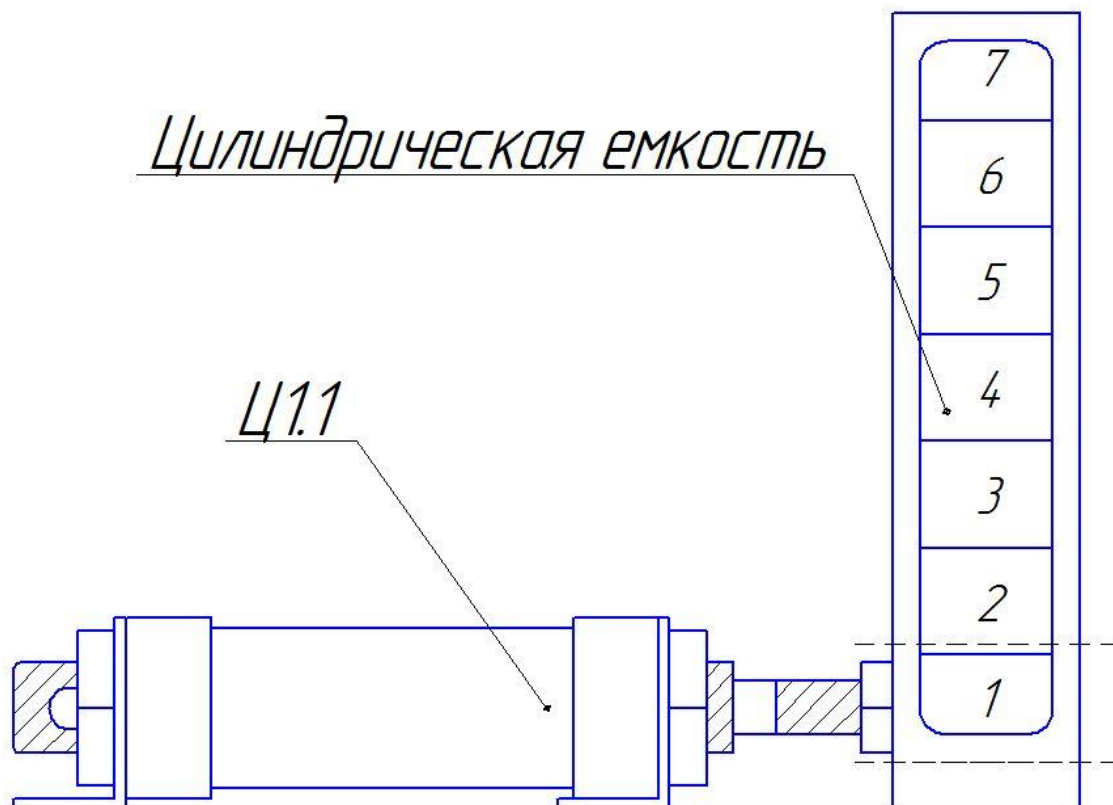


Рисунок 5 – Устройство выдачи заготовок.

Конвейер 1 служит для транспортировки заготовок в зону переноса. Конвейер 1 приводится в движение двигателем постоянного тока М2, управление которого осуществляется при помощи драйвера. Драйвер U2, выполняет функции запуска и остановки конвейера, данные команды управления поступают от ПЛК.

Как видно на структурной схеме, конвейер оснащен оптическим датчиком DK1.1, данный датчик выполняет функции концевого (когда заготовка переместится к краю конвейера 1, датчик DK1.1 подаст сигнал о том, что заготовка находится в зоне переноса и конвейер остановится).

Технологическая зона определения характеристик заготовок состоит из весов и стола. В качестве весов используется тензометрический датчик, показания которого измеряет аналоговый модуль ввода сигналов U1. Аналоговый модуль ввода сигналов U1 передает измеренные данные в ПЛК.

Стол является технологической зоной для измерений. В данной зоне установлены датчики D2 и D3. Датчик D2 определяет цвет заготовки и передает считанные данные в ПЛК. Датчик D3 – это индуктивный датчик. Он определяет наличие металла в заготовке.

В технологическую зону фасовки входит конвейер 2 и пять пневмоцилиндров. Конструкция данной технологической зоны представляет собой конвейер, с одной стороны которого установлены пневмоцилиндры и концевые оптические датчики, а с противоположной, располагаются зоны хранения заготовок. Вид сверху на технологическую зону фасовки, изображен на рисунке 6.

Управление конвейером 2 осуществляется аналогично конвейеру 1. Конвейер 2, приводится в движение двигателем постоянного тока М3, управление которого осуществляется при помощи драйвера. Драйвер U3, выполняет функции запуска и остановки конвейера, данные команды управления поступают от ПЛК. Конвейер оснащен пятью концевыми оптическими датчиками DK2.1 ... DK2.5.

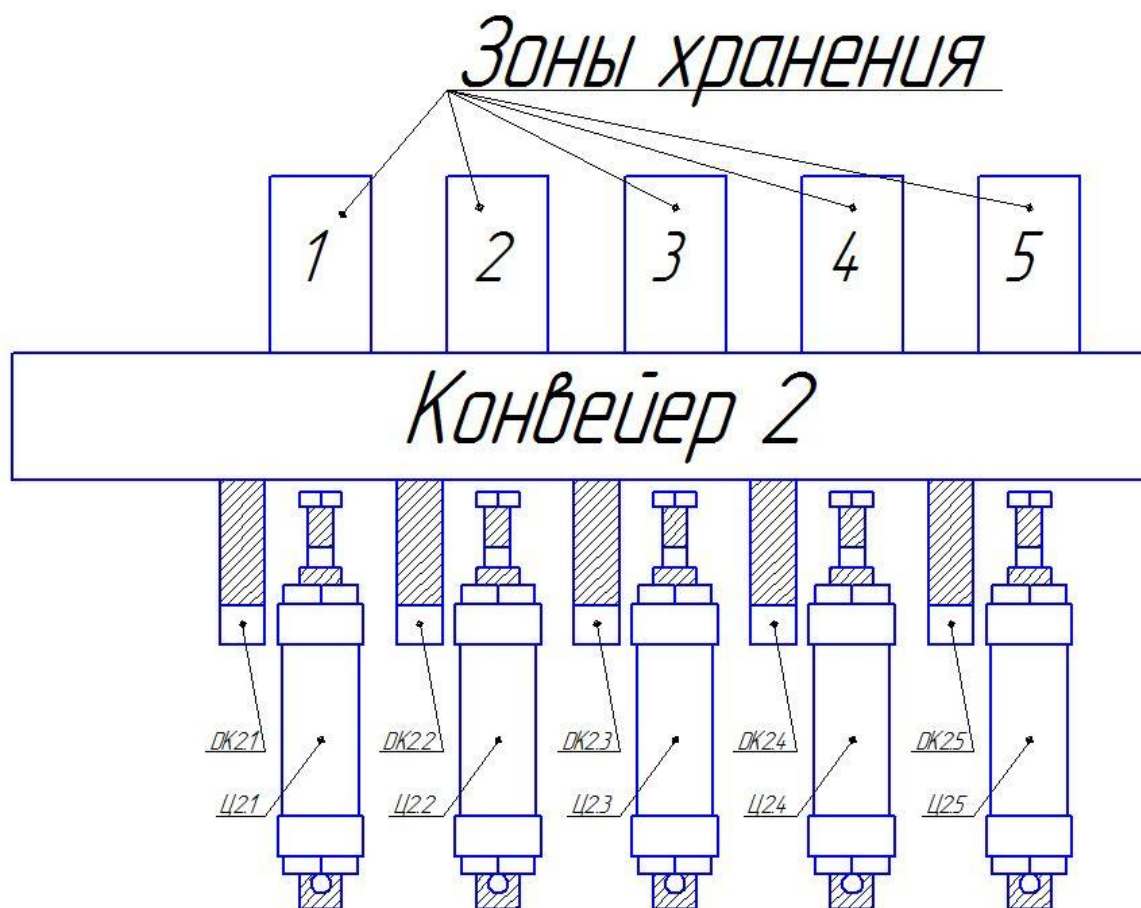


Рисунок 6 – Зона фасовки.

Заготовки перемещаются по конвейеру с лева на право, относительно рисунка 6. Концевые оптические датчики располагаются перед пневмоцилиндрами по ходу перемещения заготовок, а пневмоцилиндры установлены напротив зон хранения заготовок. Система управления мехатронным стандом после получения всех данных о характеристиках заготовки принимает решение в какую зону хранения поместить ту или иную заготовку. Когда заготовка, при перемещении по конвейеру, оказывается перед необходимой зоной хранения (о чем оповещает соответствующий оптический концевой датчик), конвейер останавливается и соответствующий пневмоцилиндр сталкивает заготовку в нужную зону хранения.

Технологическая зона переноса выполняет функции последовательного перемещения заготовок из зоны приема, через зону определения характеристик, в зону фасовки. Функции данной зоны выполняет механическая рука. Механическая рука – это устройство, состоящее из линейного актуатора, пневмоцилиндра и присоски. Механическая рука изображена на рисунке 7.

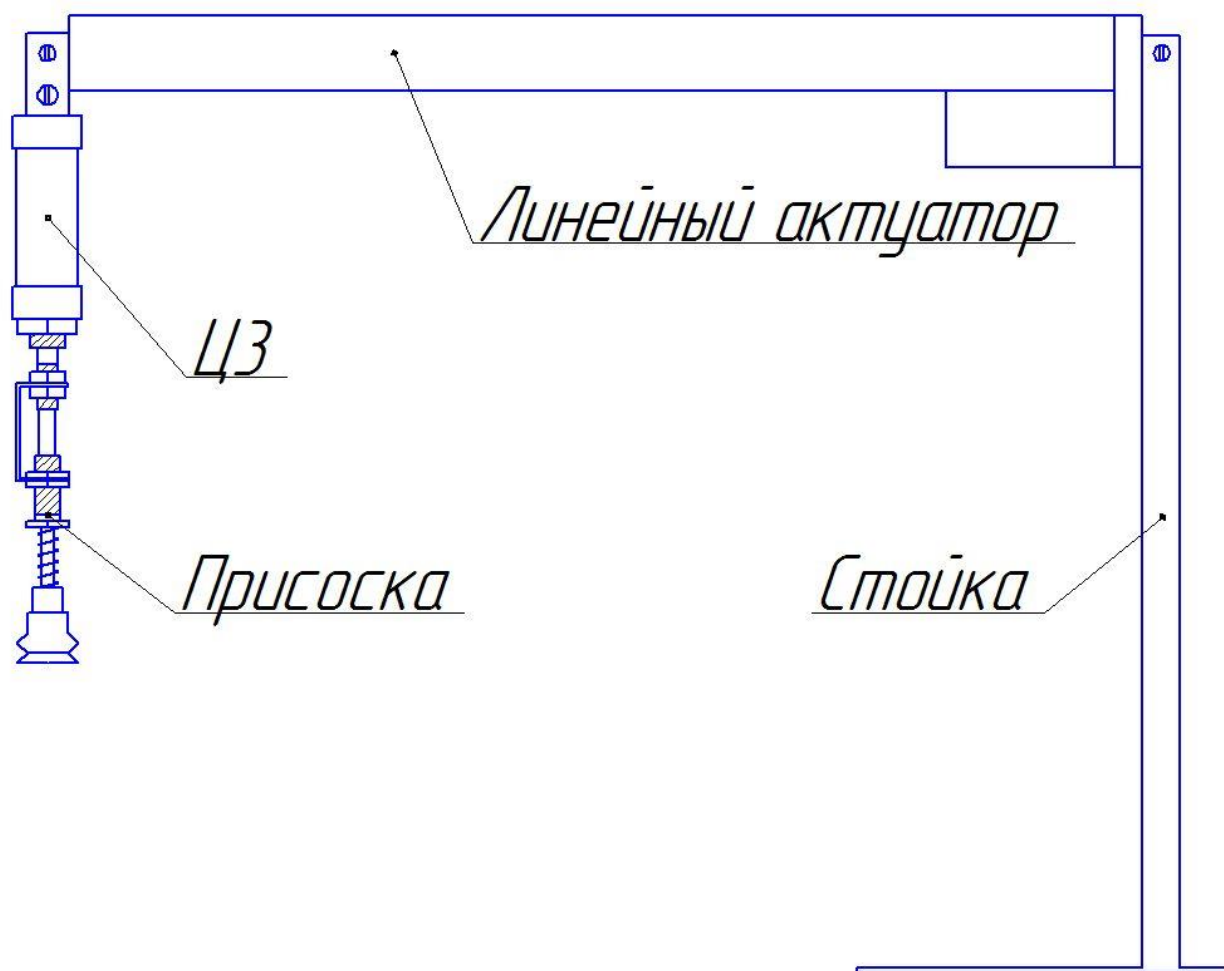


Рисунок 7 – Механическая рука.

На рисунке 7 изображена стойка, она служит для установки механической руки на основание стенда. Для понимания работы механической руки, ее можно условно разбить на два механизма, механизм захвата и механизм переноса. В качестве захвата используется пневмоцилиндр и присоска. Пневмоцилиндр опускает присоску к заготовке, после чего происходит ее захват. Далее пневмоцилиндр поднимает заготовку и линейный актуатор перемещает ее в другую технологическую зону, пневмоцилиндр опускает заготовку после присоска освобождает заготовку. Далее пневмоцилиндр поднимает присоску для корректной работы других устройств находящихся в определенных технологических зонах.

Средний уровень автоматической системы управления мехатронным стендом состоит из программируемого логического контроллера (ПЛК 160М), модуля ввода сигнала (U1), драйверов двигателей (U2, U3) и релейной схемы управления (релейная схема управления стендом приведена на электрической принципиальной

схеме, которая изображена на чертеже приложенному к данной диссертации). Как видно на структурной схеме, в данной системе управления используются устройства которые имеют интерфейс связи RS – 485. Все эти устройства объединены единой шиной по которой контроллер не только передает управляющие команды исполнительным механизмам, но и принимает сигналы от датчиков входящих в систему.

ПЛК 160М является главным устройством, которое задает логику работы всего станда, так же ПЛК 160М является связующим устройством между нижним и верхним уровнем. Все датчики и исполнительные механизмы системы управления мехатронным стандом подключены к ПЛК 160М в соответствии с схемой электрической принципиальной. ПЛК 160М имеет мощный вычислительный комплекс, что позволяет ему обеспечивать бесперебойную и без аварийную работу системы управления.

Верхний уровень автоматической системы управления мехатронного станда состоит из коммутатора и АРМа. В качестве автоматизированного рабочего места выступает персональный компьютер, на котором реализована SCADA система управления мехатронным стандом. Коммутатор выступает в качестве устройства, которое обеспечит систему управления свойством гибкости в случае ее модернизации.

## 4.1 Нижний уровень

Нижний уровень автоматической системы управления мехатронным стендом состоит из электрической и пневматической части. В электрическую часть входят электронные датчики и приводы (двигатели постоянного тока). В пневматическую часть входят пневмоцилиндры, устройство захвата (пневмоприсоска) и компрессор.

### 4.1.1 Электрическая часть

Система автоматического управления имеет в своем составе оптические датчики, тензометрический датчик, датчик индуктивности и датчик определения цвета. Датчики, входящие в состав системы автоматического управления просты в своем исполнении и именно поэтому имеют высокую надежность.

Оптические датчики используются в качестве концевых выключателей. Они сигнализируют о положении заготовок, когда те находятся на определенных местах. В системе применяются датчики типа ВБЗ С.18М.65.ТР.100.5.1.К – это диффузные оптические датчики, они работают на отражение от объекта. Данные датчики выполнены в едином корпусе цилиндрической формы диаметром 18мм и длиной 65мм. Они оснащены стеклянной оптикой и имеют диапазон чувствительности 10см, т.е. данные датчики выдают сигнал, когда объект (в данном случае заготовка), будет находиться перед датчиком на расстоянии, не превышающем данное значение. Данное требование необходимо учитывать при монтаже данных датчиков. Изображение датчика ВБЗ С.18М.65.ТР.100.5.1.К приведено на рисунке 8.



Рисунок 8 – Датчик ВБЗ С.18М.65.ТР.100.5.1.К.

Датчики питаются от постоянного напряжения, и как заявляет производитель, диапазон питающего напряжения лежит в пределах от 10В до 30В. Датчики ВБЗ С.18М.65.ТR.100.5.1.К оснащены транзисторным выходом типа ррр. Схема подключения данного датчика приведена на рисунке 7.

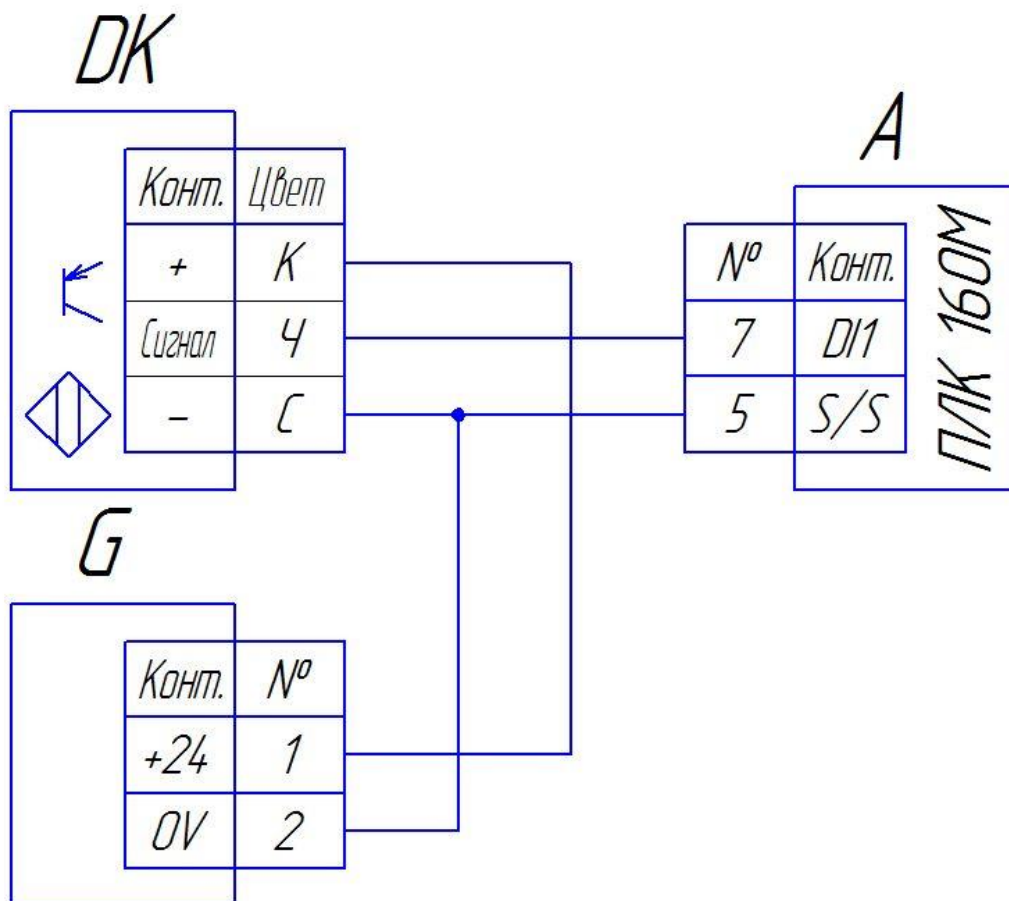


Рисунок 9 – Схема подключения оптического датчика ВБЗ С.18М.65.ТR.100.5.1.К.

Конструкция оптического датчика весьма надежна. Корпус датчика выполнен из металлического сплава (латунь) и прочного АБС пластика, а чувствительные элементы находятся под закаленным стеклом. Данный датчик имеет степень защиты IP67, что позволяет устанавливать его в агрессивной среде с температурным диапазоном от – 20 до +70 градусов цельсия.

Завод изготовитель оснащает данный датчик двухметровым, трех жильным кабелем для подключения. Распиновка жил кабеля приведена на схеме, изображенной на рисунке 9. При монтаже датчика необходимо строго соблюдать



его схему подключения. При нарушении данного требования, датчик выйдет из строя.

Тензометрический датчик используется в системе автоматического управления для определения веса заготовок. В системе применяется датчик YZC-1B. Работа датчика основана на преобразовании деформации тела в унифицированный электрический сигнал – напряжение. В датчике используются четыре пленочных тензорезисторов включенных по мостовой схеме Уитсона, что позволяет производить измерения с большей точностью. Изображение тензометрического датчика YZC-1B приведено на рисунке 10.



Рисунок 10 – Тензометрический датчик YZC-1B.

Датчик выполнен из алюминия и имеет форму бруска размером 130x25x22мм. На датчике имеются сквозные отверстия с резьбой под болт М6, они расположены по краям датчика и служат для его фиксации.

При монтаже датчика необходимо учитывать его конструктивные особенности. Датчик необходимо закрепить с одной стороны на крепление к рабочей поверхности стенда, а с другой стороны необходимо закрепить платформу, на которую будут помещаться заготовки. На датчике имеется изображение стрелки, которое указывает направления воздействия, т.е. указано направление давления груза. Данная рабочая сторона предназначена для крепления платформы, обратную сторону датчика необходимо закрепить на крепление к стенду. Для подключения

датчик имеет кабель длиной 35см, поэтому необходимо учесть данную конструктивную особенность при проектировании стенда (заложить в закупочную спецификацию экранированный кабель 4x0,35мм<sup>2</sup> для подключения датчика).

Данный датчик весьма надежен, и как заявляет производитель, может работать в агрессивной среде с температурным диапазоном от -35 до +80 градусов цельсия. Датчик может измерять грузы не более чем 3 килограмма. Поскольку в датчике применена схема моста Уитсона, состоящая из четырех тензорезисторов, датчик обладает высокой разрешающей способностью, но изменения напряжения на нем составляет два милливольт, что предполагает использование специализированного измерительного устройства, способного измерить настолько малое изменение напряжения. В системе автоматического управления мехатронным стендом применяется для решения данной задачи отдельный прибор, который передает измеренные показания датчика в контроллер. Показания тензометрического датчика измеряет прибор МВ 110-224.1ТД. Схема подключения тензометрического датчика YZC-1В приведена на рисунке 11. При подключении датчика необходимо строго соблюдать схему подключения, неверное подключение датчика приведет к его некорректной работе, что в последствии может привести к потере работоспособности датчика.

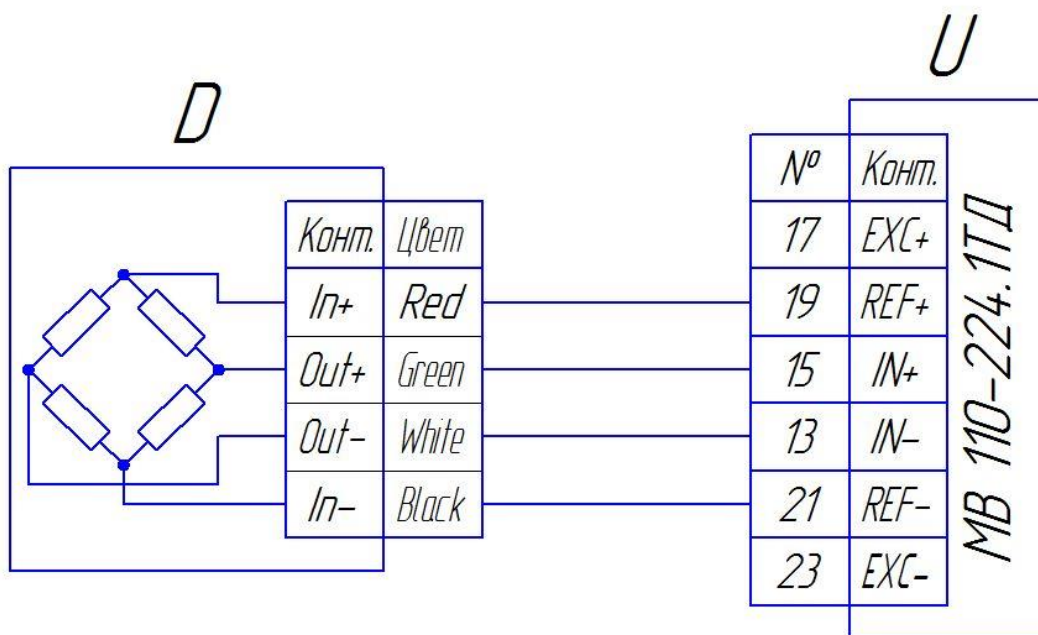


Рисунок 11 – Схема подключения тензометрического датчика YZC-1В.

Датчик индуктивности применяется в системе автоматического управления мехатронным станком для определения наличия металла в теле заготовки. В системе применяется датчик ВБ2.32.ХХ.15.1.1.К. Работа датчика основывается на измерении магнитного поля создаваемого катушкой индуктивности. При попадании в зону действия металлического предмета, магнитное поле, создаваемое катушкой датчика, изменяет параметры поля и датчик выдает унифицированный дискретный сигнал, о наличии металла. Изображение датчика индуктивности ВБ2.32.ХХ.15.1.1.К приведено на рисунке 12.

Датчик изготовлен из высокопрочного полиамида и имеет размер 50x30x15мм. Датчик в своем основании имеет два сквозных отверстия, служащих для его надежной фиксации. В данной системе управления данный датчик расположен в технологической зоне определения характеристик заготовки, конструктивно он располагается на столе таким образом, что при переносе, заготовка оказывается на чувствительном элементе датчика индуктивности. Датчик имеет зону чувствительности 15мм, что соответствует высоте заготовки, поэтому датчик индуктивности может точно определить наличие металла в теле заготовки.

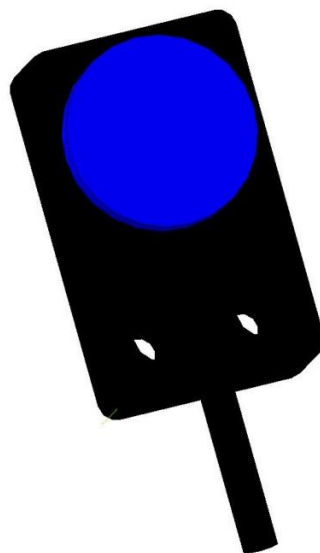


Рисунок 12 – Датчик индуктивности ВБ2.32.ХХ.15.1.1.К.

Питание датчика индуктивности должно осуществляться от сети постоянного тока в диапазоне напряжений от 10В до 30В. Датчик ВБ2.32.ХХ.15.1.1.К оснащен транзисторным выходом типа рnp. Для подключения датчика индуктивности

используется кабель (3x0,35мм<sup>2</sup>). Данный индуктивный датчик производитель оснащает кабелем длиной два метра. Схема подключения и распиновка кабеля датчика индуктивности ВБ2.32.ХХ.15.1.1.К приведена на рисунке 13.

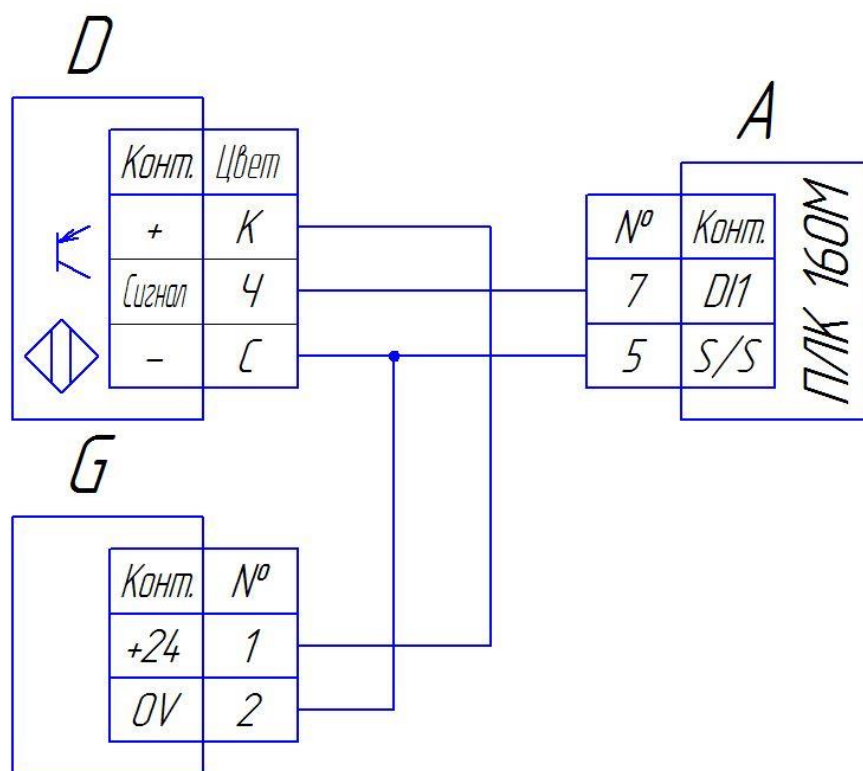


Рисунок 13 – Схема подключения датчика индуктивности ВБ2.32.ХХ.15.1.1.К.

Данный датчик индуктивности как заявляет производитель имеет степень защиты IP67, что позволяет использовать данный датчик в агрессивных средах с температурой окружающей среды от -25 до +75 градусов цельсия. При монтаже датчика необходимо строго соблюдать его схему подключения. Производитель заявляет что данный датчик имеет защиты от короткого замыкания, перегрузки по току и от переполюсовки, но как показывает практика, при нарушении данного требования, датчик может выйти из строя.

Датчик цвета применяется в автоматической системе управления мехатронным станком для определения цвета заготовок. В основе датчика лежит микросхема TCS3200. Работа данного датчика основывается на отраженном свете от объекта. Микросхема TCS3200 оснащена шестьюдесятью четырьмя фотодиодами, которые расположены в центре микросхемы. Изображение микросхемы TCS3200 приведено на рисунке 14.



Рисунок 14 – Микросхема TCS3200.

Для лучшей чувствительности датчика, над фотодиодами располагаются светофильтры (красный, зеленый, синий) в определенной последовательности. Светофильтры служат для отсеивания волн другой длины, таким образом, светофильтры выполняют функцию усиления отраженного света в своем диапазоне. Светофильтры формируют чувствительные элементы датчика, которые образуют матрицу размером четыре на четыре. Схема расположения светофильтров одного чувствительного элемента приведена на рисунке 15.

<i>Красный</i>	<i>Зеленый</i>
<i>Синий</i>	<i>Без фильтра</i>

Рисунок 15 – Схема расположение светофильтров.

Для получения усиленного отражения световой волны используется плата НУ-067. На плате НУ-067 микросхема TCS3200 расположена в центре, а по углам платы установлены яркие светодиоды, которые усиливают световой поток, в следствии чего повышается чувствительность датчика. Изображение платы НУ-067 приведено на рисунке 16.



Рисунок 16 – Датчик определения цвета TCS2300.

Данный датчик состоит из платы питания, преобразователя интерфейсов и микроконтроллера ATmega328. На микроконтроллере лежат две функции, считывание показаний микросхемы TCS3200 и передачи измеренных данных в ПЛК используя протокол Modbus RTU. Датчик цвета в сети Modbus RTU является подчиненным устройством. Датчик оснащен физическим каналом связи RS – 485. Для согласования уровней напряжения в линии RS – 485 и микроконтроллера, используется микросхема MAX3485. Датчик цвета питается от сети постоянного тока, с диапазоном напряжения от 5В до 32В. Описание датчика цвета приведено в приложении А.

В качестве приводов конвейеров используются электродвигатели постоянного тока JGY-370. Рабочее напряжение питания электродвигателей равняется 24В. Данный электродвигатель поставляется в сборе с редуктором рассчитанным на максимальную скорость вращения равную 210 оборотам в минуту. Электродвигатель JGY-370 в сборе с редуктором изображен на рисунке 17.



Рисунок 17 – Электродвигатель JGY-370 в сборе с редуктором.

Линейный актуатор, входящий в состав механической руки, приводится в движение при помощи электродвигателя постоянного тока. Рабочее напряжение электродвигателя линейного актуатора 24В. В мехатронном стенде применена модель линейного актуатора GK81. Данный линейный актуатор имеет длину 40см, а скорость перемещения штока равна 0,5 см/с. Данный линейный актуатор имеет момент на штоке равный 180 Н, а потребляемая мощность равна 30Вт. Изображение линейного актуатора GK81 приведено на рисунке 18.



Рисунок 18 – Линейный актуатор GK81.

#### 4.1.2 Пневматическая часть

Система автоматического управления имеет в своем составе пневмоцилиндры, распределители, вакуумный эжектор, присоску и компрессор. Для подключения пневмоустройств, в системе используется гибкая труба высокого давления, размером 8x5мм. Максимальное давление, которое может выдержать гибкая труба равняется 100 Бар.

В составе автоматической системе управления мехатронным стендом используются пневмоцилиндры двухстороннего действия MAL-25-100-CA от производителя Makerele. Пневмоцилиндр — это элемент пневмосистемы, который использует сжатый воздух для преобразования его энергии в линейное

механическое перемещение. Данные пневмоцилиндры имеют длину хода штока 10см. Диапазон рабочего давления пневмоцилиндров лежит в пределах от 0,1 Бар до 8 Бар. Изображение пневмоцилиндра MAL-25-100-CA приведено на рисунке 19. Подключение пневмоцилиндра к воздушной линии осуществляется при помощи быстросъёмных разъёмов.



Рисунок 19 – Пневмоцилиндр MAL-25-100-CA.

Для управления пневмоцилиндров MAL-25-100-CA применяются электромагнитные пневматические клапаны типа 4V210-08. Электромагнитный пневматический клапан — это электромеханическое устройство, являющееся элементом пневмосистемы, выполняющее функцию распределения потоков воздуха или других газов. Пневмораспределитель состоит из корпуса, поршня переключения направления потока и электромагнита. Поршень является сердечником электромагнита. При подаче управляющего напряжения на катушку электромагнита, сердечник (поршень) притягивается к нему и происходит переключение потока воздуха.

Величина управляющего напряжения для пневмораспределителя 4V210-08 равна 24В. Диапазон рабочего давления пневмораспределителей лежит в диапазоне от 0,15 Бар до 8 Бар. Подключение пневмораспределителей к линии сжатого воздуха, осуществляется при помощи быстросъёмных разъёмов. Изображение пневмораспределителя 4V210-08 приведено на рисунке 20.





Рисунок 20 – Пневмораспределитель 4V210-08.

В системе автоматического управления мехатронным стендом применяется пневмораспределитель 3V210-08. Пневмораспределитель служит для управления присоской. Величина управляющего напряжения катушки электромагнита для пневмораспределителя 3V210-08 равна 24В. Диапазон рабочего давления пневмораспределителей лежит в диапазоне от 0,15 Бар до 8 Бар. Подключение пневмораспределителей к линии сжатого воздуха, осуществляется при помощи быстросъёмных разъёмов. Изображение пневмораспределителя 3V210-08 приведено на рисунке 21.



Рисунок 21 – Пневмораспределитель 3V210-08.

Пневматический вакуумный эжектор – это устройство, которое служит для создания вакуума путем передачи кинетической энергии сжатого воздуха окружающей среде. Пневматический вакуумный эжектор работает по закону Бернулли. Сжатый воздух создает в специальном сопле с сужающимся сечением пониженное давление окружающей среды, вызывая подсос в другой поток. В системе автоматического управления используется пневматический вакуумный эжектор CV-20HS, производителя AIRAURRI. Данный пневматический вакуумный эжектор работая при давлении 5 Бар, создает вакуумное давление величиной 0,99 Бар и при данном показателе имеет расход воздуха 180 л/мин. Подключение пневматического вакуумного эжектора к линии сжатого воздуха, осуществляется при помощи быстросъёмного разъёма. Изображение пневматического вакуумного эжектор CV-20HS приведено на рисунке 22.

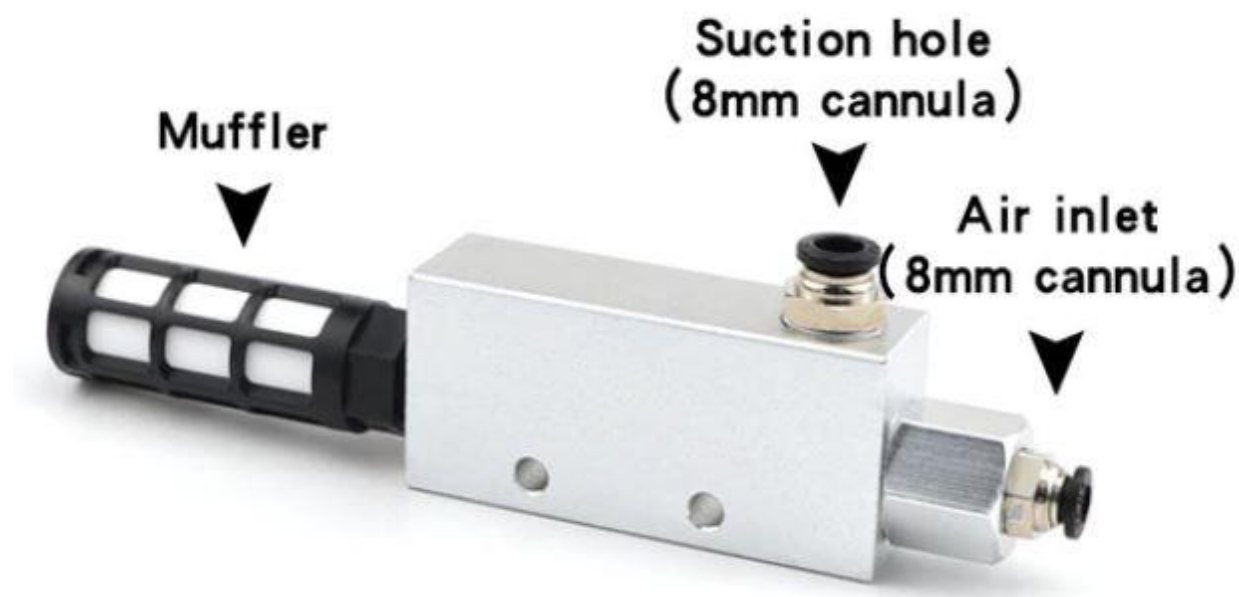


Рисунок 22 – Пневматический вакуумный эжектор CV-20HS.

Для захвата заготовок механическая рука оснащена присоской. Присоска изображена на рисунке 23. Как видно из рисунка 21 в конструкцию присоски входит не только сильфонная присоска, но и механизм компенсации прижима. Ход механизма компенсации прижима равен 30 мм. Диаметр захвата присоски равен 30 мм.



Рисунок 23 – Присоска.

Пневматическая система мехатронного стенда состоит из различных пневмоустройств, которые нуждаются в определенном количестве и качестве сжатого воздуха, что в свою очередь предъявляет высокие требования к устройству, которое будет обогащать пневмосистему сжатым воздухом. В качестве такого устройства выступает компрессор С-50 LB 40, производителя REMEZA. Компрессор С-50 LB 40 – это компрессор поршневого типа. Поршни данного компрессора приводятся в движение при помощи асинхронного электродвигателя через ременную передачу. Компрессор запрещено использовать во взрывопожароопасных помещениях, а так же под воздействием атмосферных осадков. Изображение компрессора С-50 LB 40 приведено на рисунке 24.

Данный компрессор питается от трех фазной сети переменного тока с напряжением 380В. Перед подключением компрессора к питающей сети, необходимо заземлить компрессор проводом с сечением не менее  $4\text{мм}^2$ , и убедиться в том, что заземление устройства выполнено надежно. Мощность потребляемая электродвигателем равна 3 кВт.

Компрессор оснащен ресивером объемом 50л. Производительность данного компрессора равна 530 л/мин.



Рисунок 24 – Компрессор С-50 LB 40.

Компрессор С-50 LB 40 осуществляет автоматическое регулирование производительности путем периодического пуска и остановки компрессора при помощи телепрессостата. Компрессор снабжен манометром служащим в качестве визуального контролирующего прибора, который показывает давление сжатого воздуха в ресивере. Так же конструкцией компрессора предусмотрен разгрузочный клапан, который отсекает сжатый воздух от поршневого блока после остановки электродвигателя. В компрессоре предусмотрен предохранительный клапан, который защищает компрессор от превышения максимально допустимого давления. Для реализации защиты электродвигателя, в компрессоре реализована защита от обрыва одной из фаз питающей сети и от короткого замыкания.

## 4.2 Средний уровень

Средний уровень автоматической системы управления мехатронным стендом состоит из промышленного логического контроллера, релейной схемы, модуля ввода и драйверов управления электродвигателями.

### 4.2.1 ПЛК 160М

ПЛК 160М – это промышленный логический контроллер от компании ОВЕН. ПЛК 160М является главным узлом автоматической системы управления мехатронным стендом. В состав стенда входит ПЛК 160М модификации ПЛК 160 – 220.А.М. Данная модификация является продолжением линейки ПЛК 160 и является улучшенной версией. Изображение ПЛК 160 – 220.А.М приведено на рисунке 25.



Рисунок 25 – ПЛК 160 – 220.А.М.

Промышленный контроллер был выбран исходя из своих характеристик и популярности использования данной линейки в различных системах автоматического управления. Основные характеристики контроллера ПЛК 160 – 220А.М приведены в таблицах 1 – 7.

Таблица 1 – Питание ПЛК 160М.

Параметр	Значение
Напряжение питания, В:	от 90В до 264В
Потребляемая мощность, ВА	45
Пусковой ток, А, не более	54
Длительность переходного процесса, мс, не более	2
Параметры встроенного источника питания	Выходное напряжение 24±3 В, ток потребления не более 400 мА
Аккумулятор резервного питания	LIR2477

Таблица 2 – Ресурсы ПЛК 160М.

Параметр	Значение
Центральный процессор	RISC-процессор на базе ядра ARM-9, 32 разряда, 180 МГц
Объем оперативной памяти (тип памяти)	8 Мб (SDRAM), из них 1 Мб для кода пользовательской программы, 128 кб для переменных пользовательской программы
Объем энергонезависимой памяти	6 Мб для хранения файлов
Размер Retain-памяти	16 Кб
Время выполнения одного цикла программы	– Минимальное (нестабилизируемое) – 250 мкс; – Установленное по умолчанию (стабилизированное) – 1 мс
Дополнительное оборудование	– Часы реального времени с автономным аккумуляторным питанием – Встроенный источник выдачи звукового сигнала;

Таблица 3 – Дискретные (Цифровые) входы ПЛК 160М.

Параметр	Значение
Количество входов из них быстродействующих	16 4 (DI1-DI4)
Напряжение питания дискретных входов, В	24 ± 3
Максимальный входной ток дискретного входа	не более 7 мА при питании 24 В, не более 8,5 мА при питании 27 В
Сигнал «логической единицы», соответствующий состоянию «Включено», дискретных входов для постоянного напряжения, В (ток в цепи)	от 15 до 30 (ток от 3 до 15 мА)
Сигнал «логического нуля», соответствующий состоянию «Выключено», дискретных входов для постоянного напряжения, В (ток в цепи)	от минус 3 до 5 (ток до 15 мА)
Минимальная длительность импульса, мс	
– для обычных входов	1
– для быстродействующих	0,02
Гальваническая развязка	Групповая
Электрическая прочность изоляции между группой дискретных входов и:	
– цепями питания, В	1780
– группами цепей дискретных выходов, В	1780
– группами остальных цепей, В	560

Таблица 4 – Дискретные (Цифровые) выходы ПЛК 160.

Параметр	Значение
Количество релейных выходов	12
Гальваническая развязка	Индивидуальная (для DO1-DO8), групповая (DO9-DO10;DO11-DO12)
Электрическая прочность изоляции, В	1780
Максимальный ток, коммутируемый контактами реле, А, не более	3
Время переключения контактов реле из состояния «лог. 0» в «лог. 1» и обратно, мс, не более	50 (выходы DO1-DO12)
Механический ресурс реле	– не менее 300 000 циклов переключений при максимальной коммутируемой нагрузке;

Таблица 5 - Аналоговые входы ПЛК 160.

Параметр	Значение
Количество аналоговых входов	8
Тип поддерживаемых унифицированных сигналов	Ток от 0 (4) до 20 мА Ток от 0 до 5 мА Напряжение от 0 до 10 В
Разрядность АЦП, бит	14
Входное сопротивление: В режиме измерения тока В режиме измерения напряжения	Не более 170 Ом Не менее 200 кОм
Период опроса аналоговых входов, мс	10
Предел основной приведенной погрешности преобразования	±0,25 %



Таблица 6 – Аналоговые выходы ПЛК 160.

Параметр	Значение
Количество аналоговых выходов	4
Тип выходного сигнала, Разрядность ЦАП (бит)	Универсальный, ток от 4 до 20 мА, напряжение от 0 до 10 В, 12
Предел основной приведенной погрешности ЦАП	$\pm 0,5 \%$
Минимальный период обновления значений выходов	100 мс
Питание аналоговых выходов	Внешнее (24 $\pm$ 3 В)
Гальваническая изоляция аналоговых выходов	Индивидуальная
Электрическая прочность изоляции между группами аналоговых выходов и группами остальных цепей, В	1780

Таблица 7 – Интерфейсы связи ПЛК 160

Интерфейс	Количество
RS-485	1
Ethernet 100 Base-T	1
RS-232	1
RS-232-Debug	1

#### 4.2.2 МВ 110 – 224.1ТР

Модуль аналогового ввода МВ 110 – 224.1ТР в системе автоматического управления мехатронным стендом выполняет функции обработки показаний тензометрического датчика, перевода показаний датчика в физическую величину и

передачи показаний в канал связи RS – 485. Изображение модуля аналогового ввода MB 110 – 224.1TP приведено на рисунке 26.



Рисунок 26 – Модуль аналогового ввода MB 110 – 224.1TP.

Модуль MB 110 – 224.1TP оснащен одним измерительным каналом для работы с тензометрическим датчиком чувствительным элементом которого являются четыре пленочных тензорезистора включенных по мостовой схеме Уитсона. Питание данного модуля в системе автоматического управления мехатронным стендом осуществляется от сети переменного тока, напряжением 220В. Модуль MB 110 – 224.1TP оснащен универсальным входом питания и как заявляет производитель, питание может осуществляться как от сети переменного тока с величиной напряжения сети от 90В до 264В, так и от сети постоянного тока с величиной напряжения от 20В до 60В. Мощность модуля MB 110 – 224.1TP не превышает 5Вт. Аналогово - цифровой преобразователь модуля MB 110 – 224.1TP

имеет разрядность 24 бит, что позволяет модулю производить измерения с высокой точностью. Разрешающая способность прибора равна 1,5мкВ. Сопротивление подключаемых тензометрических датчиков должно находиться в пределах от 87Ом до 1кОм. Данный модуль имеет высокую скорость опроса аналогового входа, МВ 110 – 224.1ТР производит измерения показаний датчика со скоростью, одно измерение в 25мс.

Модуль МВ 110 – 224.1ТР оснащен интерфейсом RS -485 и может производить обмен информацией на скоростях от 9600 бит/с до 115200 бит/с. Модуль МВ 110 – 224.1ТР работает с протоколом связи Modbus RTU и выступает в сети RS – 485 подчиненным устройством, т.е. модуль должен опрашивать мастера сети.

#### 4.2.3 Драйверы управления электродвигателями

Драйверы управления электродвигателями постоянного тока применяются в автоматической системе управления мехатронным стендом для управления технологическими конвейерами, расположенными в зонах выдачи заготовок и фасовки. Приводным элементом драйвера служит микросхема L298N. Микросхема L298N изображена на рисунке 27.

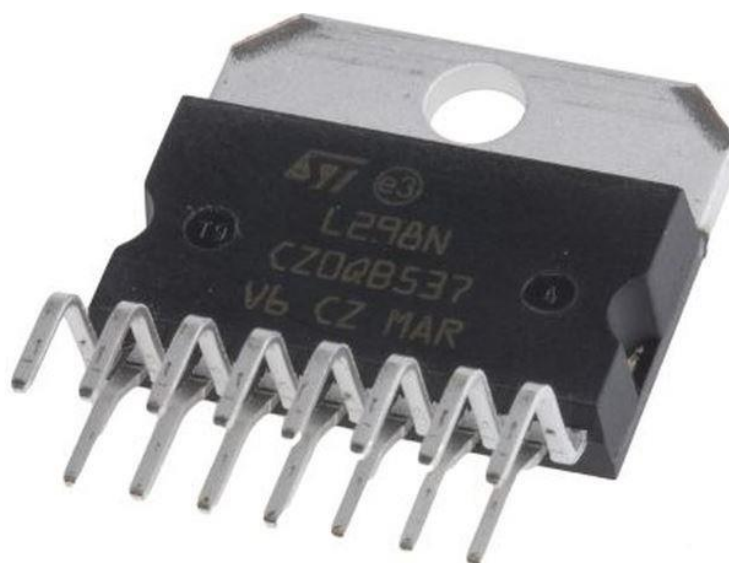


Рисунок 27 – Микросхема L298N.

Микросхема L298N – это двухмостовой драйвер управления электродвигателями. Максимальный рабочий выходной ток микросхемы равен 2А. Напряжение питания нагрузки лежит в пределах от 4,8В до 46В.

В системе автоматического управления мехатронным стендом используются электродвигатели JGY-370 мощность которых составляет 30Вт. Система управления питает двигатели напряжением 24В, таким образом микросхема должна выдерживать рабочий ток электродвигателя величиной равной 1,25А. Как видно из паспортных данных микросхемы, максимальный рабочий ток равен величине 2А, что превышает требуемое значение почти на 40%. Исходя из алгоритма работы стенда, можно с уверенностью сказать, что данная микросхема будет работать надежно и долговечно.

Управление микросхемой осуществляет микроконтроллер ATmega328P. Микроконтроллер принимает управляющие команды (пуск/стоп) от ПЛК 160М, обрабатывает полученные команды и выдает управляющий сигнал на электродвигатель постоянного тока. Драйверы оснащены физическим каналом связи RS – 485, по которому поступают команды от ПЛК 160М по протоколу связи Modbus RTU. Для согласования уровней напряжения в линии RS – 485 и микроконтроллера, применяется микросхема MAX3485. В сети Modbus RTU драйверы управления электродвигателями постоянного тока являются подчиненными устройствами. Питание драйверов осуществляется от сети постоянного тока, с напряжением 24В. Описание драйверов управления двигателями постоянного тока приведено в приложении Б.

## 4.3 Верхний уровень

Верхний уровень автоматической системы управления мехатронным стендом состоит из коммутатора КСН210 и автоматического рабочего места – АРМа.

### 5.3.1 Автоматическое рабочее место – АРМ

В качестве автоматического рабочего места в системе автоматического управления мехатронным стендом выступает персональный компьютер (ПК). В качестве системного блока ПК был выбран настольный компьютер Dell Optiplex 7071 Tower. На данном ПК установлена операционная система Windows 10 версии Pro. Основные характеристики ПК приведены в таблице 8.

Таблица 8 – Основные характеристики Dell Optiplex 7071 Tower.

№	Параметр	Значение
1	Модель процессора	Intel Core i5 9500
2	Количество ядер процессора	6
3	Частота процессора	3000 МГц
4	Тип видеокарты	встроенная
5	Модель интегрированной видеокарты	Intel UHD Graphics 630
6	Объем видеопамати	выделяется из оперативной
7	Тип оперативной памяти	DDR4
8	Объем оперативной памяти	8 Гб
9	Тип накопителя	SSD (твердотельный)
10	Объем накопителя	256 Гб
11	Скорость сетевого адаптера	1 Гбит

Характеристики Dell Optiplex 7071 Tower позволяют с уверенностью утверждать, что все программное обеспечение, которое задействованное в данном проекте будет работать с высоким быстродействием. Особое внимание уделяется

сетевой карте. В Dell Optiplex 7071 Tower установлена гигабитная сетевая карта, которая позволяет гарантированно передавать большой объем информации с большой скоростью.

Выбор монитора остановился на модели PhotoVue SW240 производителя BenQ. Данный монитор имеет размер экрана 24.1". Основные характеристики монитора приведены в таблице 9.

Таблица 9 – Основные характеристики BenQ PhotoVue SW240.

№	Параметр	Значение
1	Максимальное разрешение	1920x1200
2	Тип подсветки матрицы	LED
3	Технология изготовления матрицы	IPS
4	Размер видимой области экрана	519x324 мм
5	Яркость	250 Кд/м <sup>2</sup>
6	Угол обзора	178°
7	Потребляемая мощность	55 Вт

### 5.3.2 Коммутатор KCH210

В качестве коммутатора в системе автоматического управления мехатронным стендом используется KCH210, производителя компании ОВЕН. Данный коммутатор выполняет функции шлюза между ПЛК и АРМ. KCH210 неуправляемый коммутатор, он работает в промышленных сетях стандарта IEEE 802.3. Коммутатор оснащен четырьмя портами типа Fast Ethernet (100 Мбит/с) и одним комбо-портом. Комбо-порт работает по интерфейсу Gigabit Ethernet (1000 Мбит/с) или по оптоволокну через SFP-модуль.

В системе автоматического управления мехатронным стендом коммутатор располагается в непосредственной близости от ПЛК 160М. Подключение ПЛК 160М к коммутатору необходимо произвести при помощи SFTP патч-корда с категорией кабеля не менее 5е. При использовании такого кабеля исключаются возможные

электромагнитные помехи, которые могут наводиться на проводнике создавая различные помехи.

Подключение АРМа необходимо произвести через комбо-порт. Подключение необходимо осуществить при помощи SFTP патч-корда с категорией кабеля 6. Данное подключение позволит располагать АРМ на значительном удалении от стенда, не более 100м. При необходимости увеличения расстояния между АРМ и стендом необходимо использовать оптоволокно, для обеспечения стабильной скорости передачи данных.

Основные характеристики приведены в таблице 10.

Таблица 10 – Основные характеристики КСН210.

№	Параметр	Значение
1	Напряжение питания	10...48 (24) В постоянного тока
2	Максимальная мощность потребления	4 Вт
3	Ethernet порты и поддерживаемые интерфейсы	<ul style="list-style-type: none"> <li>• 4 × 10/100BASE-T/Te;</li> <li>• 1 × 1000BASE-T</li> </ul>
4	Тип разъема	8P8C (RJ45)
5	Электрическая прочность изоляции интерфейсов Ethernet, не менее	1000 В
6	SFP порт (поддерживаемые интерфейсы)	<ul style="list-style-type: none"> <li>• 100BASE-FX;</li> <li>• 1000BASE-X (1000BASE-CX, 1000BASE-LX, 1000BASE-SX)</li> </ul>
7	Степень защиты корпуса согласно ГОСТ IEC 61131-2	IP20

В данном разделе подробно описана физическая архитектура стенда, реализующая технологический процесс сортировки. Исполнительные механизмы и датчики входящие в состав устройств стенда в полной мере способны выполнять задачу технологического процесса. Устройства, входящие в состав стенда имеют достаточную надежность и обладают большим набором функций, таким образом, на базе данных устройств можно реализовать различные технологические процессы.

## 5 Программная часть

### 5.1 Средний уровень

#### 5.1.1 Конфигуратор М 110

Настройка работы модуля МВ 110-224.1ТР производится в программе Конфигуратор М 110. Конфигуратор М 110 – это программа служащая для настройки различных модулей ввода/вывода линейки 110, от производителя компании Овен. Данная программа, инструкция по ее установке и инструкция по работе с прибором МВ 110-224.1ТР поставляется на диске вместе с прибором либо данные файлы можно скачать на официальном сайте производителя. Подключение МВ 110-224.1ТР для его настройки производится по каналу связи RS – 485.

Работа в программе «Конфигуратор М 110» начинается с установки связи с прибором. При запуске программы «Конфигуратор М 110» открывается окно «Установка связи с прибором», рисунок 28.

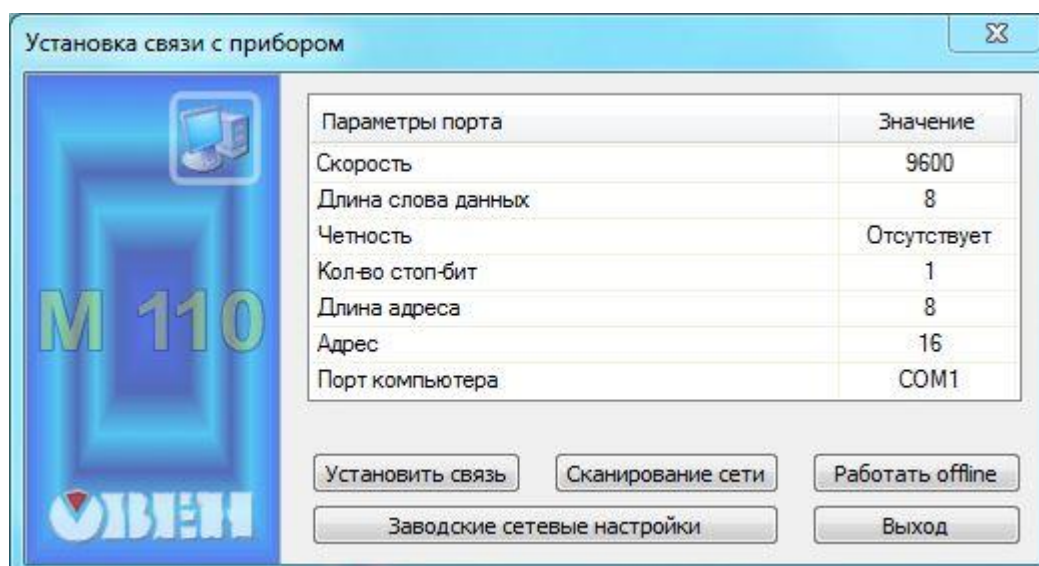


Рисунок 28 – Окно «Установка связи с прибором».

В данном окне необходимо выставить параметры сети, если они известны. Если узнать сетевые параметры прибора МВ 110-224.1ТР не представляется возможным, то прибор можно восстановить к заводским настройкам. Для восстановления заводских настроек необходимо установить проводную перемычку



на пятую и седьмую клемму и подать питание на прибор, подождать минуту, отключить питание и восстановить подключение к ПК. Заводские сетевые настройки модуля приведены в таблице 11.

Таблица 11 – Заводские сетевые настройки МВ 110-224.1ТР.

№	Параметры порта	Значение
1	Скорость	9600
2	Длинна слова данных	8
3	Четность	Отсутствует
4	Кол-во стоп-бит	1
5	Длина адреса	8
6	Адрес	16

Параметр «Порт компьютера» устанавливается в зависимости от того, к какому порту подключен прибор МВ 110-224.1ТР.

После успешного подключения к модулю, программа «Конфигуратор М 110» выдаст окно с сообщением об успешном подключении к прибору и откроет окно конфигурации, рисунок 29. Необходимо создать новую конфигурацию (Файл, Новый) и сохранить (Файл, Сохранить).

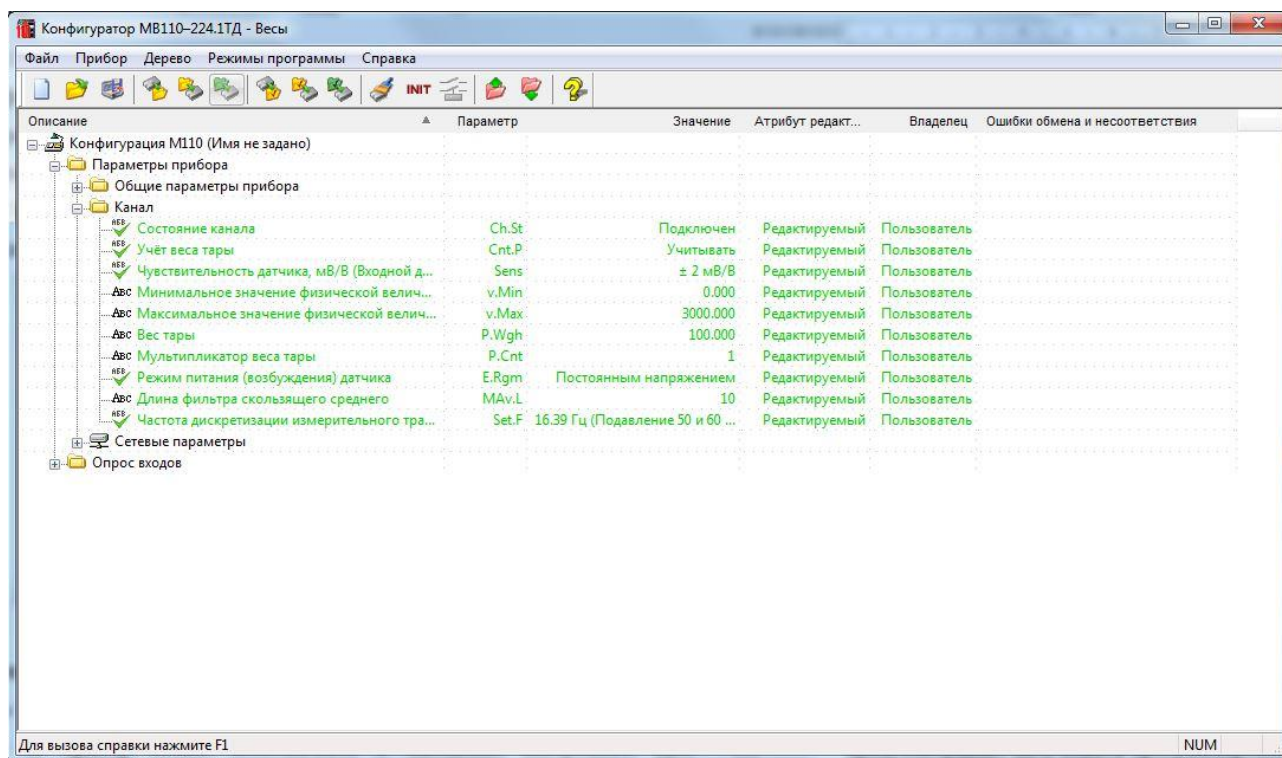


Рисунок 29 – Окно конфигурации прибора МВ 110-224.1ТР.

В окне «Конфигуратор МВ 110-224.1ТР» необходимо открыть параметры прибора, относящиеся к измерительному каналу как показано на рисунке 71.

Параметры измерительного канала заполняются с учетом используемого тензометрического датчика (Параметры и устройство датчика подробно описано в «Электрической части» раздела «Нижний уровень»). Параметры модуля МВ 110-224.1ТР измерительного канала, используемые в данном проекте приведены в таблице 12.

Таблица 12 – Параметры модуля МВ 110-224.1ТР измерительного канала.

№	Описание	Параметр	Значение
1	Состояние канала	Ch.St	Подключен
2	Учет веса тары	Ch.P	Учитывать
3	Чувствительность датчика	Sens	±2мВ/В
4	Минимальное значение физической величины	v.Min	0.000
5	Максимальное значение физической величины	v.Max	3000.000
6	Вес тары	P.Wgh	100.000
7	Мультипликатор веса тары	P/Cnt	1
8	Режим питания (возбуждения) датчика	E.Rgm	Постоянным напряжением
9	Длина фильтра скользящего среднего	Mav.L	10
10	Частота дискретизации измерительного тракта	Set.F	16.39 (Подавление 50 и 60 Гц)

## 5.1.2 CoDeSys

Программирование ПЛК 160М производится в программе Codesys V2. Codesys – это среда предназначенная для разработки программного обеспечения управления программируемых логических контроллеров. В codesys реализована возможность полного использования языков стандарта ГОСТ Р МЭК 61131-3-2016, что делает среду универсальным инструментом для написания программного обеспечения под различные платформы ПЛК. Стандарт ГОСТ Р МЭК 61131-3-2016 предусматривает пять языков программирования:

- IL (Instruction List) — Язык инструкций. По синтаксису написания программ имеет схожесть с таким языком программирования как ассемблер.
- ST (Structured Text) — Язык структурированного текста. Синтаксис данного языка схож с языком Pascal.
- LD (Ladder Diagram) — Язык диаграмм релейной логики. В данном языке строятся схемы на основе релейных компонентов.
- FBD (Function Block Diagram) — Язык диаграмм функциональных блоков. Язык предусматривает графическое построение схем на основе подпрограмм (блоков) имеющих различное функциональное назначение.
- SFC (Sequential Function Chart) — Язык диаграмм последовательных функций. Язык SFC представляет собой последовательность определенных ходов и перемещений по программе описывающую ее структуру.

Codesys поддерживает еще один язык который основан на языке FBD, CFC (Continuous Function Chart) – язык непрерывных функциональных схем. Данный язык, как и FBD, предусматривает графическое построение функциональных блоков, но в нем имеется возможность произвольной расстановки порядка их выполнения, что дает программисту дополнительные возможности и как следствие более удобную форму написания. В данной системе, в качестве языка для написания программы, был выбран язык CFC.

Для написания программы для ПЛК 160М необходимо использовать версию codesys V2. Данная программа поставляется вместе с промышленным логическим

контроллером на диске, а также данную версию пакета можно скачать с официального сайта компании «Овен», во вкладке на продукт ПЛК 160М. Помимо среды разработки, потребуется target файл. Target файл содержит подробное описание целевой платформы, т.е. подробное описание ресурсов ПЛК. Актуальная версия target файла поставляется на том же диске что и среда разработки, либо ее так же можно скачать на официальном сайте, во вкладке на необходимый продукт. Подробное описание по установке codesys и добавлению target файла можно найти на официальном сайте компании «Овен».

После успешной установки среды разработки и добавления target файла, при первом запуске codesys, откроется стартовое окно, приведенное на рисунке 30.

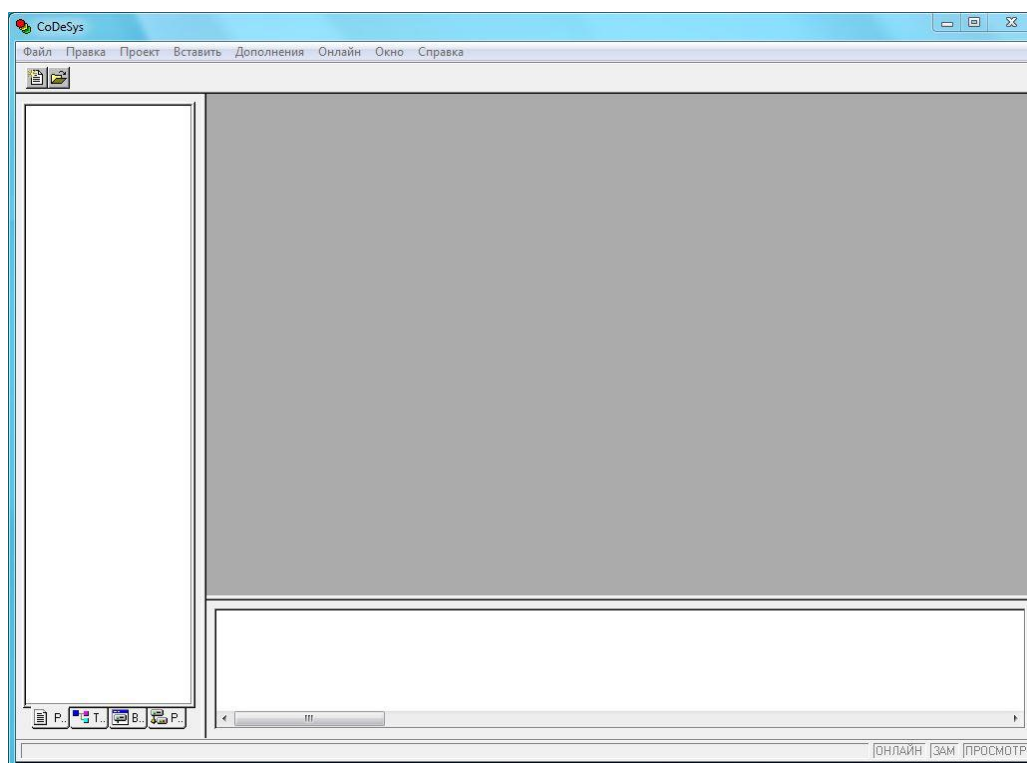


Рисунок 30 – Стартовое окно программы codesys V2.

Для создания нового проекта необходимо перейти во вкладку меню файл и в раскрывшемся списке нажать на пункт создать. После выполнения указанных действий откроется окно «Настройки целевой платформы». Окно «Настройки целевой платформы» приведено на рисунке 31.



Рисунок 31 – Окно «Настройки целевой платформы».

В данном окне необходимо указать конфигурацию ПЛК, для которого будет производиться разработка программного продукта. В данном случае необходимо указать «PLC160M v2», как показано на рисунке 32.

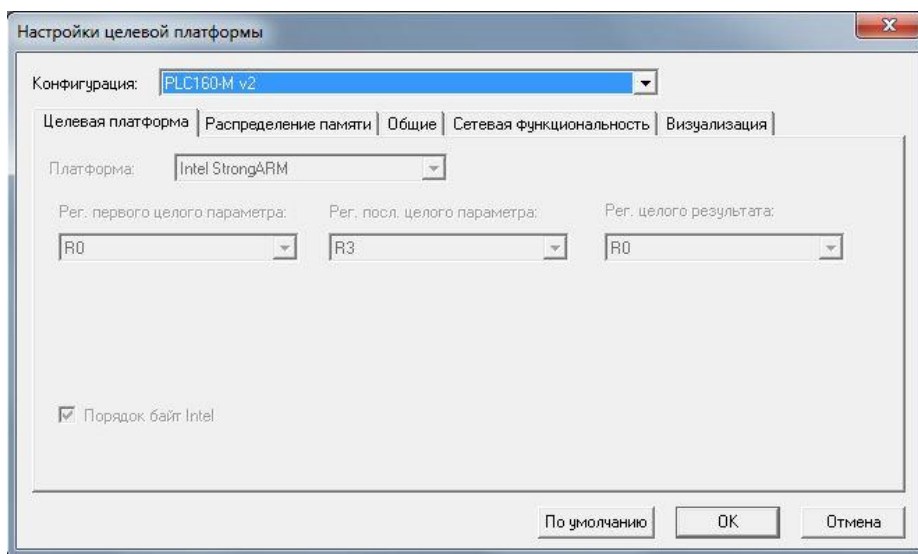


Рисунок 32 – Окно «Настройки целевой платформы» с указанием необходимого ПЛК.

Как видно на рисунке 102, после выбора необходимого ПЛК, в окне «Настройки целевой платформы» появились вкладки, которые содержат различные настройки ПЛК. В проекте вносить изменения в данные целевой платформы не требуется. Далее необходимо нажать на кнопку «ОК». После нажатия кнопки «Ок» среда разработки откроет окно «Новый программный компонент POU», представленное на рисунке 33.

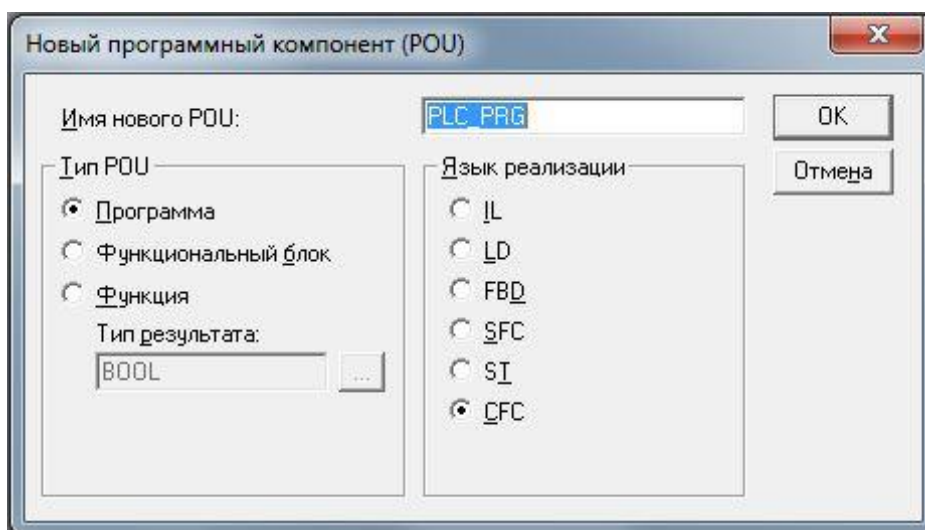


Рисунок 33 – Окно «Новый программный компонент POU».

В окне «Новый программный компонент POU» среда разработки codesys предлагает указать «Имя нового POU», выбрать его тип и язык реализации. На данном этапе создания проекта мы можем выбрать только желаемый язык разработки, а все остальные пункты, необходимо оставить без изменения. После выбора языка написания программы, нажимаем в окне «Новый программный компонент POU» на кнопку «ОК». При успешном создании программы откроется окно проекта с именем программы (PLC\_PRG) и в дереве проекта появится соответствующая запись как показано на рисунке 34.

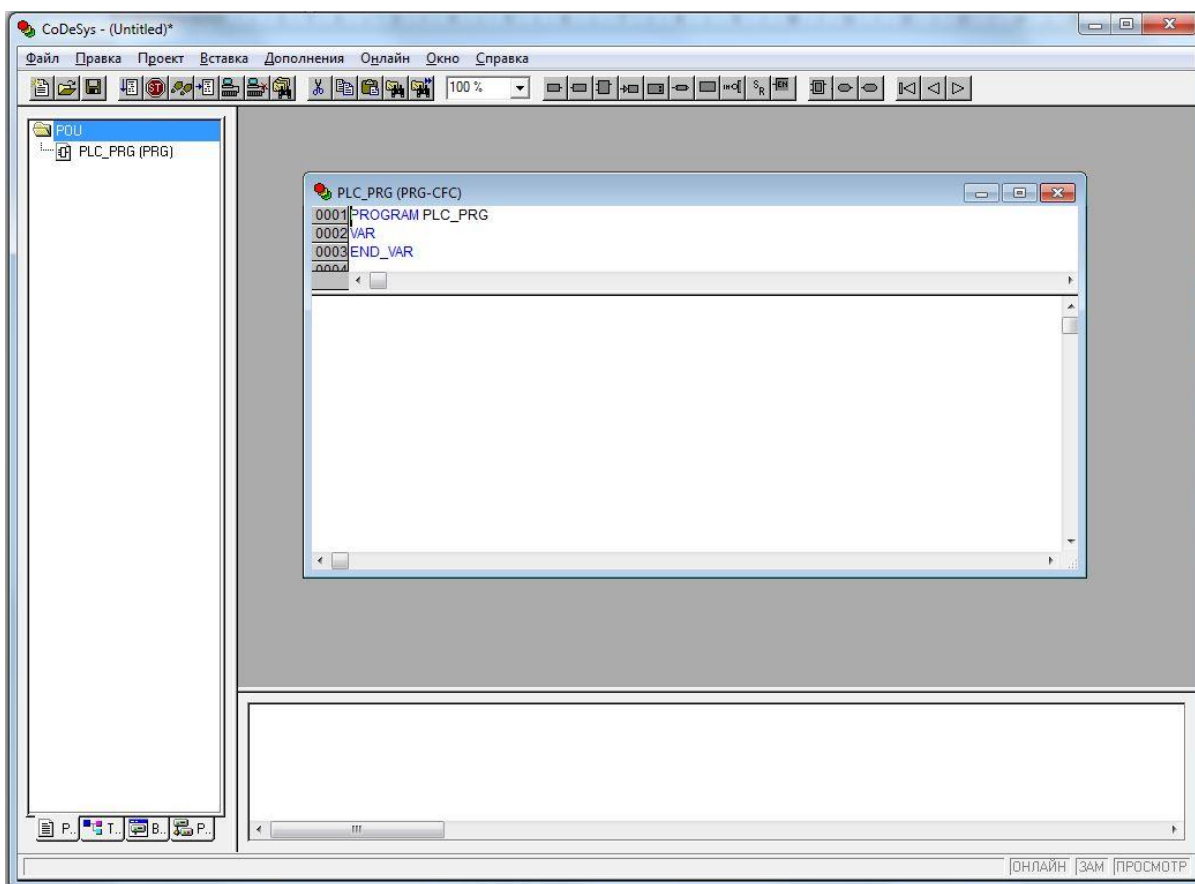


Рисунок 34 – Окно программы codesys при успешном создании проекта.

Далее необходимо сохранить созданный проект. Целью данного действия является не только сохранение созданного проекта, но и включение функции авто сохранения, что позволит сохранить измененные данные проекта при некорректном закрытии среды. Для сохранения проекта необходимо перейти во вкладку меню файл и выбрать в выпавшем списке пункт «Сохранить как». После чего в программе откроется окно «Сохранить как», в котором не обходимо указать место

расположения файла и указать его имя. Окно «Сохранить как» приведено на рисунке 35.

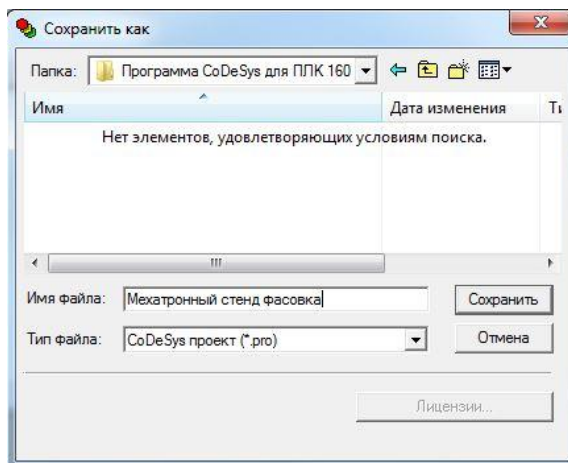


Рисунок 35 – Окно «Сохранить как».

После указаний данных требований, нажимаем на кнопку «Сохранить». При успешном сохранении файла в указанной папке появится файл с указанным именем проекта имеющий расширение «.pro». Так же при успешном сохранении в названии окна среды разработки codesys появится имя проекта, как показано на рисунке 36.

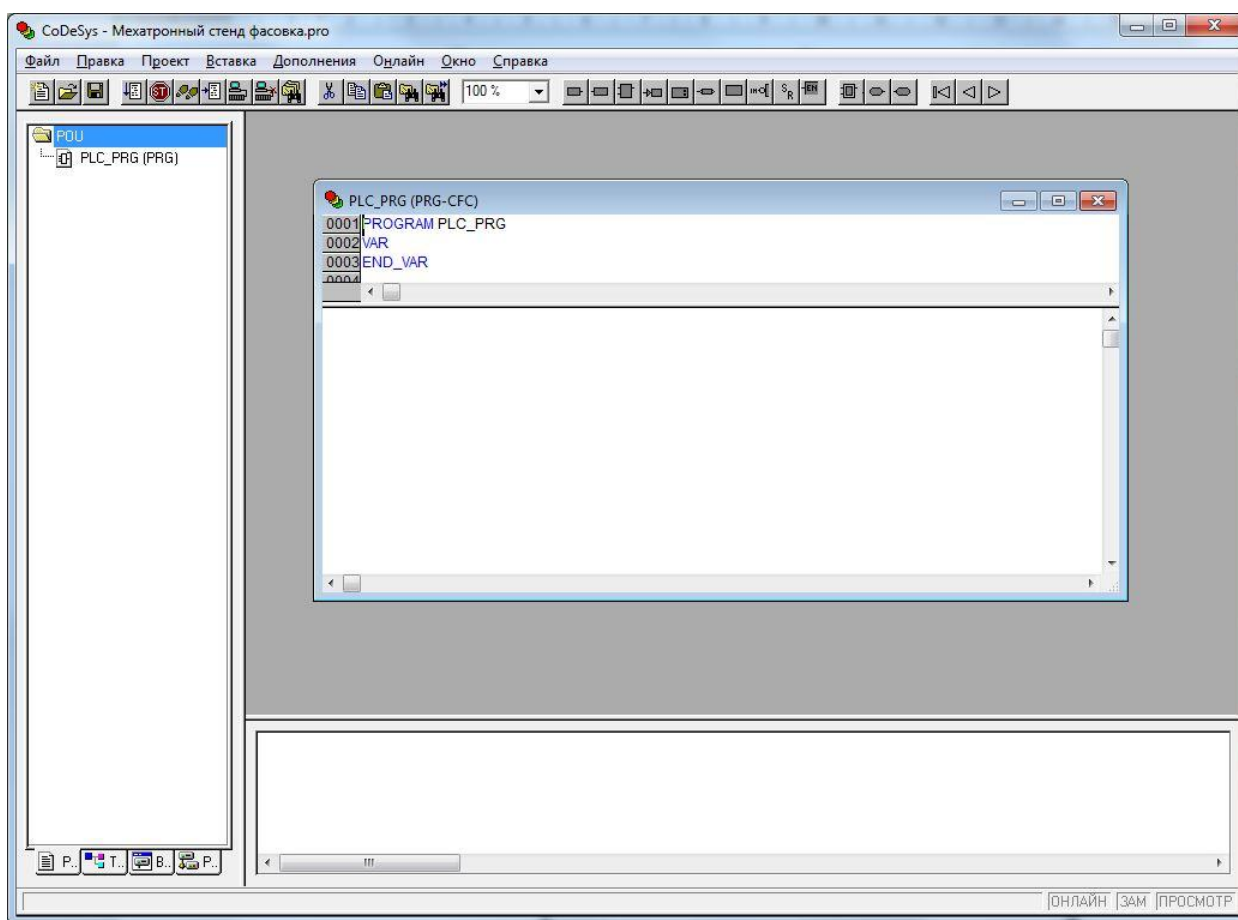


Рисунок 36 – Окно «Codesys», с сохраненным проектом.

Следующим этапом в создании проекта является объявление глобальных переменных. Глобальные переменные – это переменные, которые могут использоваться во всем проекте без ограничений. На данном этапе необходимо присвоить физическим входам и выходам ПЛК глобальные переменные. Для выполнения данного действия необходимо перейти во вкладку «Ресурсы», в области дерева проекта и выбрать в представленном списке пункт «Конфигурация ПЛК», как показано на рисунке 37.

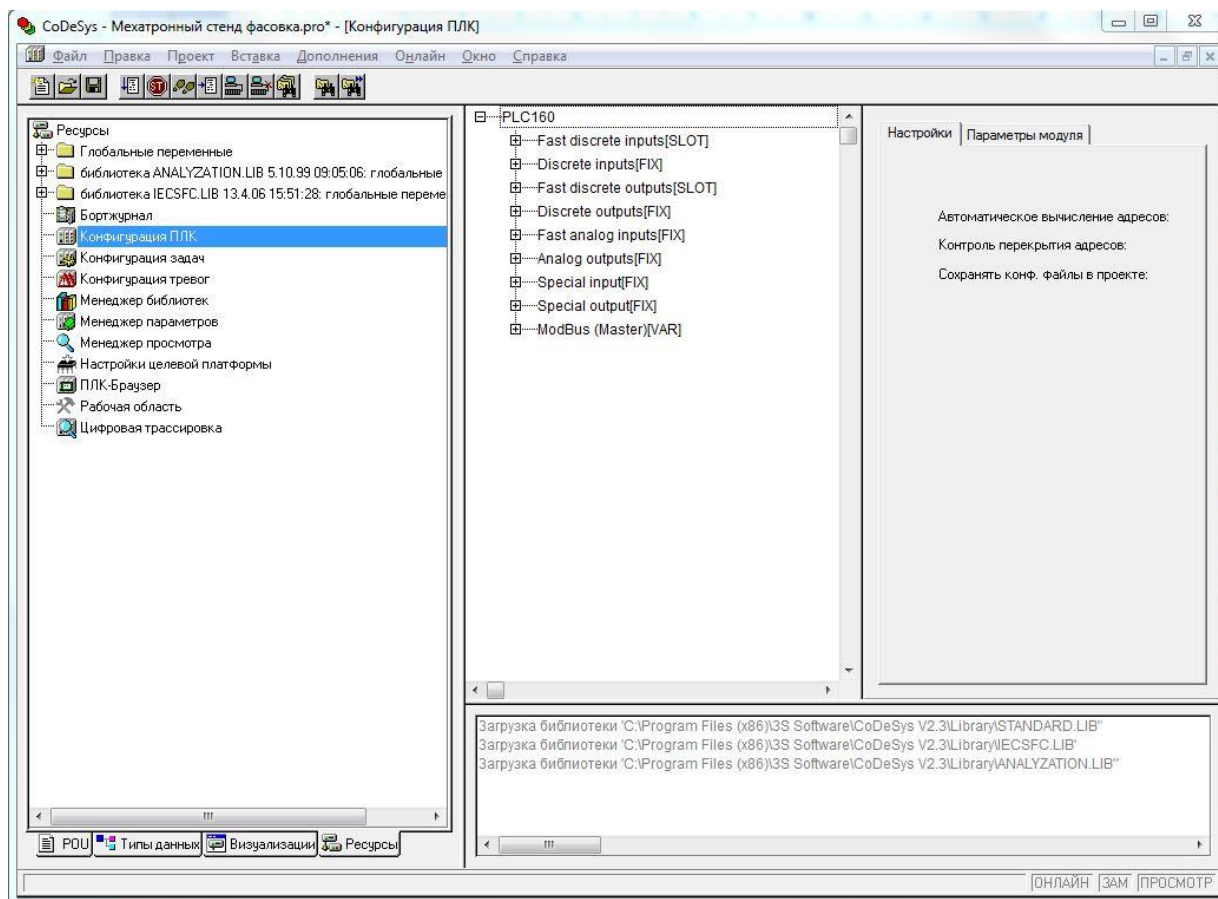


Рисунок 37 – Окно «Конфигурация ПЛК».

В открывшемся окне «Конфигурация ПЛК», как показано на рисунке 37, приведен общий список всех имеющихся входов и выходов контроллера ПЛК 160М. Для структуризации представления данных составим таблицу физических переменных. Укажем в данной таблице номера и типы входов/выходов ПЛК, а также используя электрическую принципиальную схему мехатронного стенда, укажем в данной таблице исполнительные механизмы и их обозначения на схеме. Для более удобного понимания программы, присвоим имена тэгам приближенным к названиям датчиков и исполнительным механизмам.



Таблица 13 – Физические переменные.

№	№ входа	Тип входа/выхода	Номер клеммы ПЛК	Тэг	Датчик/ Исп. мех.	Обозначение на схеме
1	1	DI	7	DK1_1	Концевик	DK1.1
2	2	DI	8	DK2_1	Концевик	DK2.1
3	3	DI	9	DK2_2	Концевик	DK2.2
4	4	DI	10	DK2_3	Концевик	DK2.3
5	5	DI	11	DK2_4	Концевик	DK2.4
6	6	DI	12	DK2_5	Концевик	DK2.5
7	7	DI	13	DI_1	Датчик индуктивности	D2
8	8	DI	14	Pusk_SB	Пуск	SB3
9	9	DI	15	Stop_SB	Аварийный стоп	SB4
10	10	DI	16	Vrabote_kompressor	Статус компрессора Вкл/Выкл	KM1
11	1	DO	52	Valve_K1	Пневмо-клапан	Y1
12	2	DO	54	Valve_K2	Пневмо-клапан	Y2.1
13	3	DO	56	Valve_K3	Пневмо-клапан	Y2.2
14	4	DO	58	Valve_K4	Пневмо-клапан	Y2.3
15	5	DO	60	Valve_K5	Пневмо-клапан	Y2.4
16	6	DO	62	Valve_K6	Пневмо-клапан	Y2.5
17	7	DO	64	Valve_K7	Пневмо-клапан	Y3
18	8	DO	66	Valve_K8	Пневмо-клапан	Y4
19	9	DO	68	Pusc_Com	Пуск компрессор	K1
20	10	DO	69	Stop_Com	Стоп компрессор	K2
21	11	DO	71	Forward_act	Актуатор вперед	K3
22	12	DO	72	Back_act	Актуатор назад	K4

ПЛК 160М оснащен шестнадцатью дискретными входами и двенадцатью выходами. Группа дискретных входов состоит из обычных и быстродействующих входов. ПЛК 160М оснащен четырьмя быстрыми входами. Система управления мехатронным стендом использует быстродействующие входы в качестве обычных. Поэтому в окне «Конфигурация ПЛК» в данном случае необходимо записать переменные не изменяя конфигурацию данных входов.

Присвоение переменных к указанным в таблице 13 входам и выходам происходит путем двойного нажатия на необходимый вход или выход. Далее производится запись соответствующего имени тега, для подтверждения необходимо нажать на клавишу «Enter», либо однократно нажать по пустому месту левой клавишей мыши. Окно «Конфигурация ПЛК» с присвоенными тегами изображено на рисунке 38.

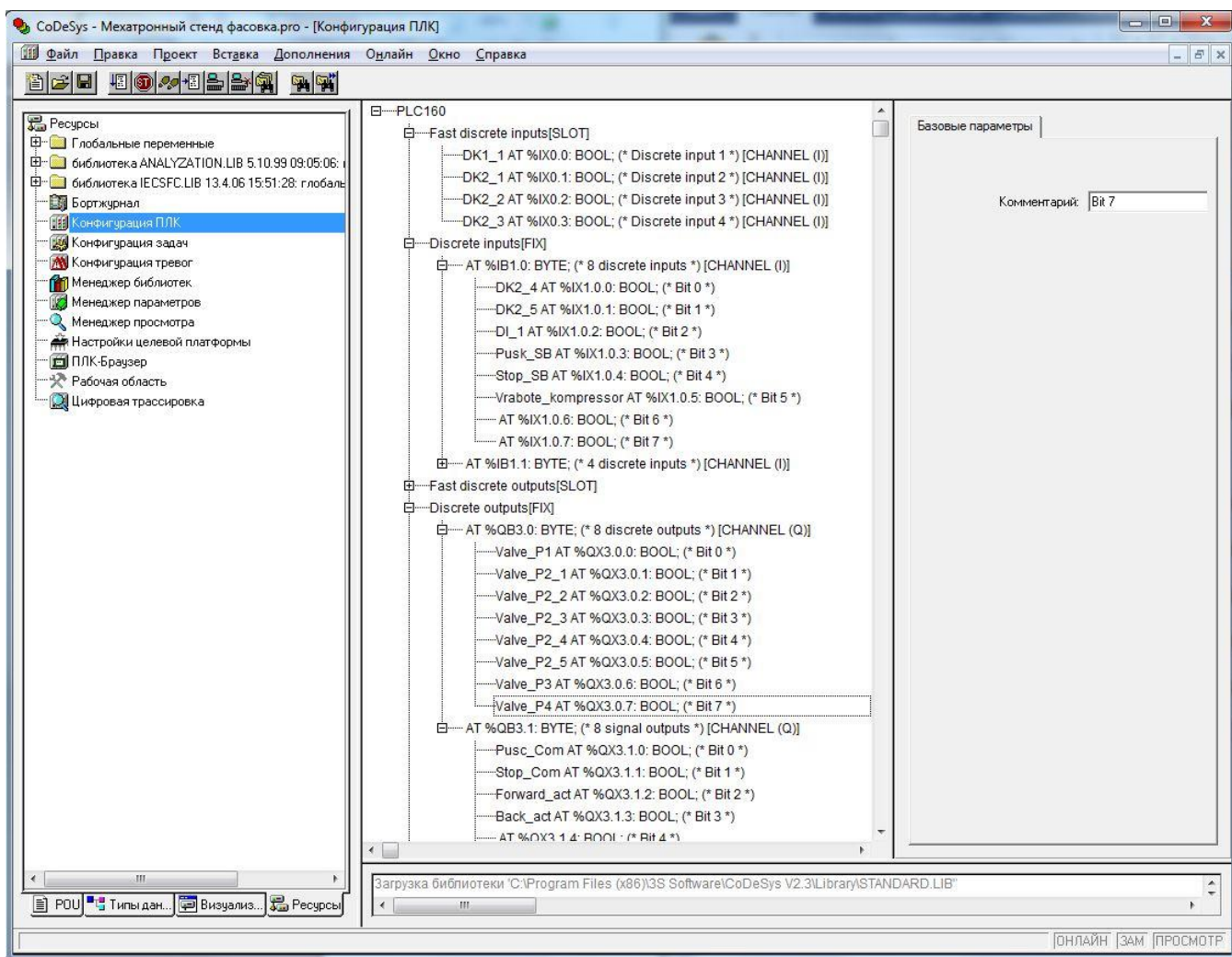


Рисунок 38 – Окно «Конфигурация ПЛК» с присвоенными тегами.

В таблице 13 отражены только физические переменные, которые поступают непосредственно на входы и выходы ПЛК, но в системе управления мехатронным стандом используется еще и сеть среднего уровня. В сеть среднего уровня входит модуль MB 110-224.1TP, драйверы двигателей конвейеров и датчик цвета. Сеть среднего уровня построена на физическом интерфейсе RS – 485 и использует протокол передачи данных modbus RTU. Для настройки ПЛК для работы в данной сети в качестве мастера в окне «Конфигурация ПЛК» необходимо создать вышеуказанное подключение. Создание данного подключения производится путем нажатия правой клавиши мыши по строчке «PLC160» в окне «Конфигурация ПЛК». В открывшемся списке необходимо выбрать пункт «Добавить Подэлемент» и выбрать пункт «ModBus (Master)...», как показано на рисунке 39.

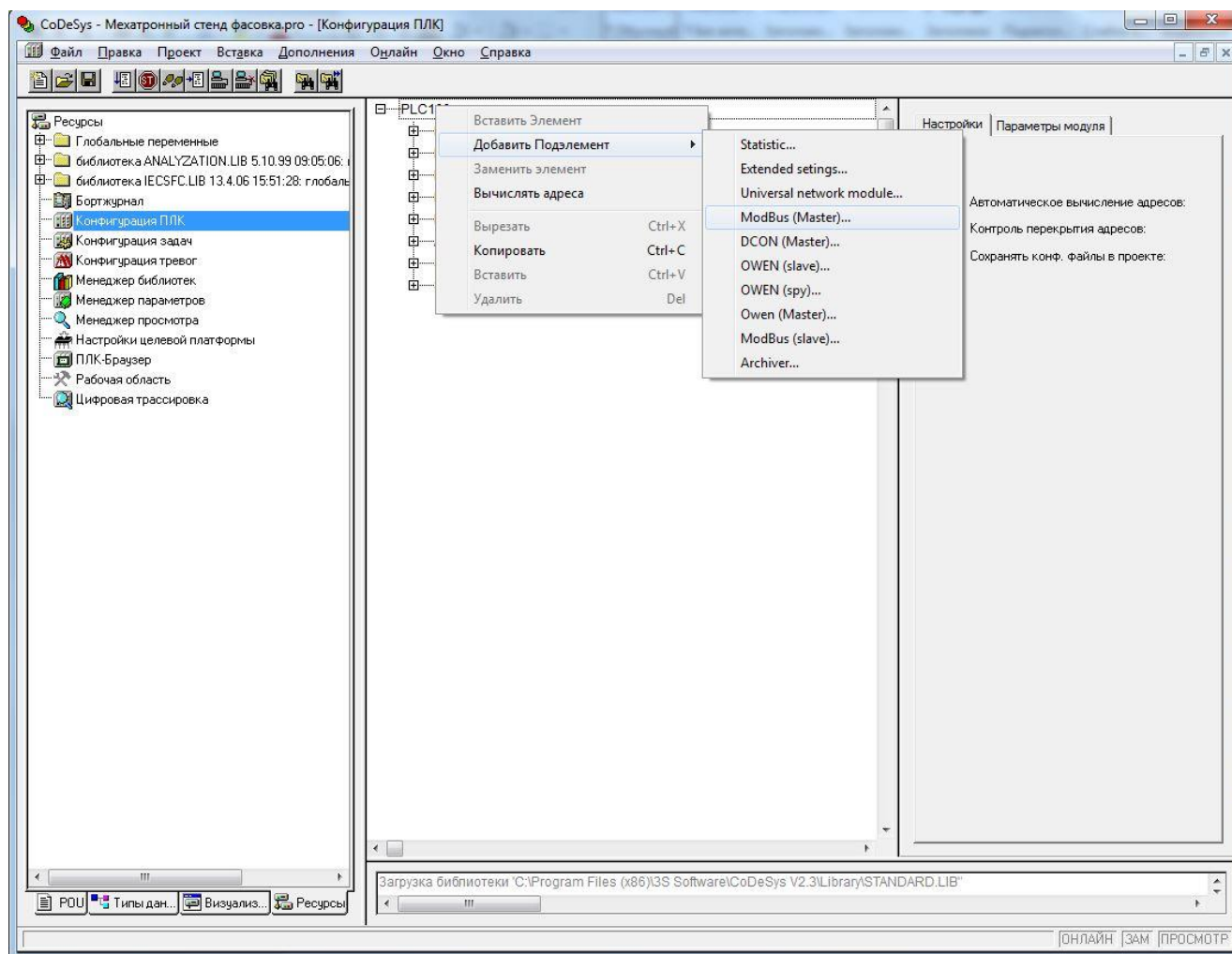


Рисунок 39 – Добавление подключения «ModBus (Master)...».

После выполнения вышеуказанных действий в дереве конфигурации ПЛК появится строка «ModBus (Master)[VAR]». По умолчанию среда разработки codesys

создает подключение по каналу RS – 232. Для изменения данного параметра, необходимо открыть под ветвь подключения «ModBus (Master)[VAR]» и нажать правой клавишей мыши по строке «Debug RS-232[SLOT]». После в выпавшем списке выбрать пункт «Заменить элемент» и указать подключение «RS-485-1», как показано на рисунке 40.

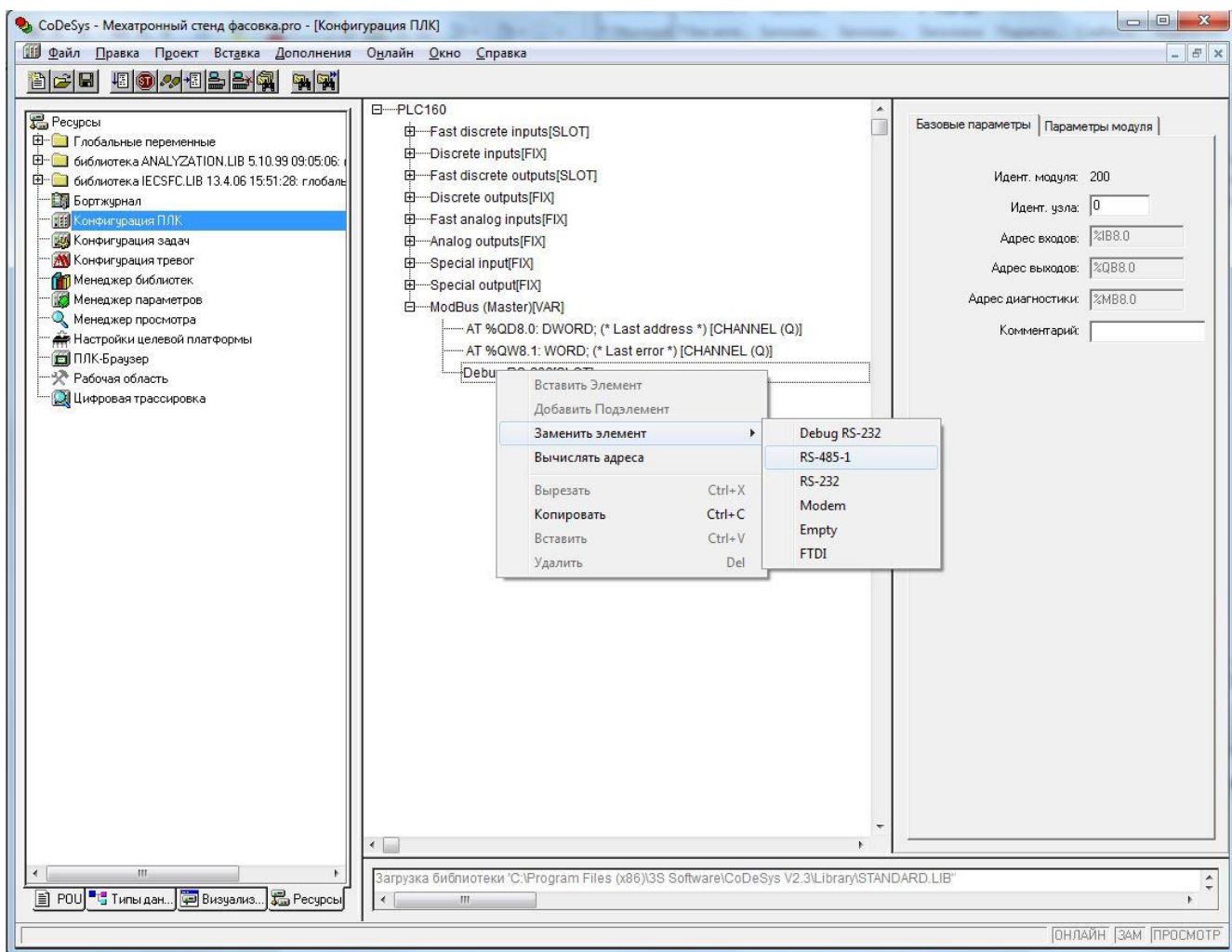


Рисунок 40 – Настройка подключения к сети «RS-485-1».

Далее необходимо создать и настроить устройства входящие в состав сети среднего уровня. Для этого необходимо нажать правой клавишей мыши по строке «ModBus (Master)[VAR]» и выбрать пункт «Добавить Подэлемент» и в списке выбрать «Universal Modbus device...». После этого в проект codesys будет добавлен универсальный модуль modbus сети, который работает в режиме slave.

Устройства, входящие в состав сети среднего уровня имеют адресацию представленную в таблице 14. Для удобства добавим в комментарии «Universal Modbus device» названия устройств.

Таблица 14 – Адреса устройств сети среднего уровня.

№	Устройство / ОБЦ	Адрес
1	МВ 110-224.1ТР / U1	1
2	Драйвер управления электродвигателем (конвейер 1) / U2	2
3	Драйвер управления электродвигателем (конвейер 2) / U3	3
4	Датчик цвета / U4	4

Первым устройством в сети среднего уровня выступает модуль МВ 110-224.1ТР, поэтому необходимо настроить первое универсальное устройство в codesys для работы с этим модулем. Для назначения адреса устройства в проекте, необходимо пометить добавленный «Universal Modbus device», в правой части окна codesys перейти на вкладку «Параметры модуля» и в списке найти строку с параметром «ModuleSlaveAddress», данный параметр располагается на 5-ой строке в указанном списке. Укажем адрес МВ 110-224.1ТР как показано на рисунке 41.

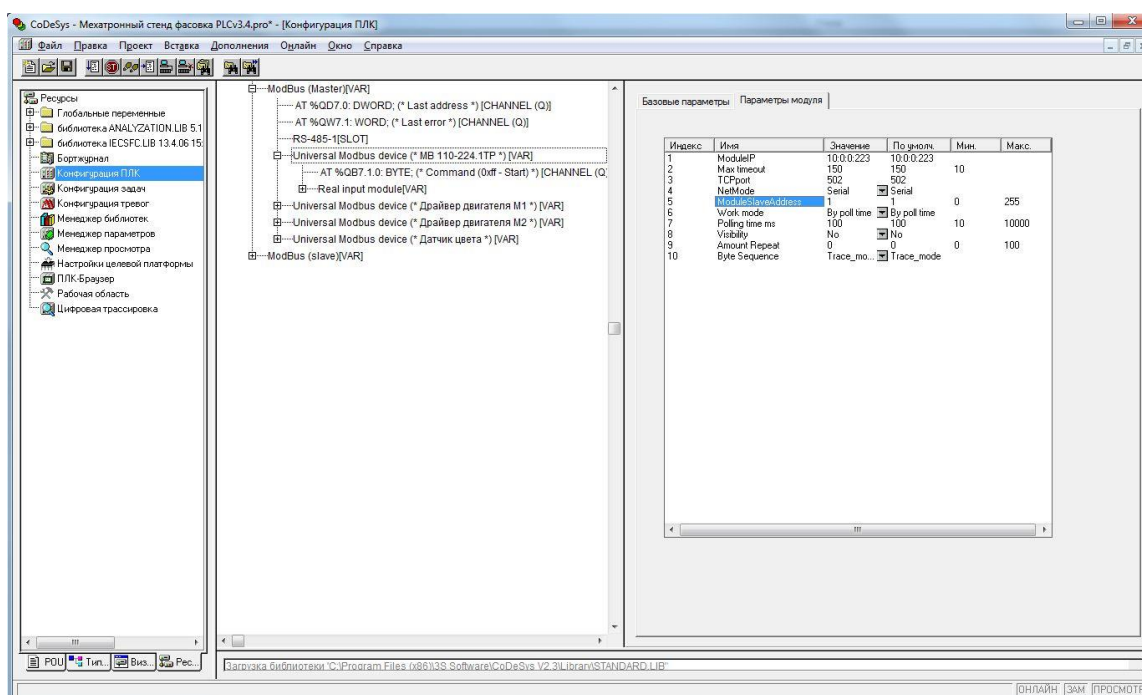


Рисунок 41 – Параметры «Universal Modbus device».

Далее необходимо указать регистр который будет опрашивать ПЛК 160М. Для этого необходимо нажать правой клавишей мыши на строчке «Universal Modbus device», вывести курсор мыши на «Добавить Подэлемент» и в выпавшем списке выбрать «Real input module...». Далее необходимо указать номер регистра, МВ 110-

224.1TP передает значение веса в 46 регистре, функция чтения 0x03 «Read Holding Registers». Для установки данных параметров в дереве конфигурации ПЛК нажмем на «Real input module» и перейдем на вкладку «Параметры модуля». Укажем в первой строке номер регистра, а во второй строке функцию чтения. Окно с указанными параметрами приведено на рисунке 42. Зададим имя переменной используемой в программе «Mod\_Massa».

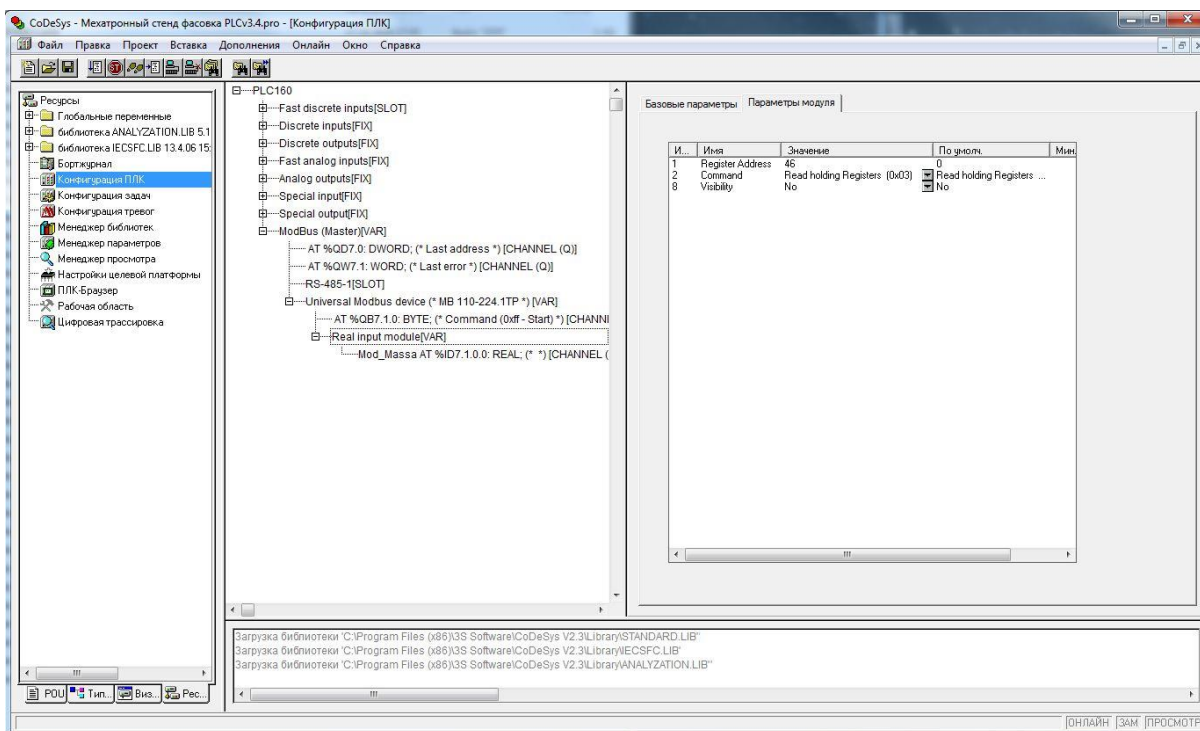


Рисунок 42 – Параметры «Real input module».

Далее по аналогии необходимо создать и настроить в проекте codesys «Universal Modbus device» для драйверов управления электродвигателями и датчика цвета используя таблицы 15 – 17.

Таблица 15 – Регистры драйвера управления электродвигателя U2.

№	№ Регистра	Тип данных	Комментарий	Имя переменной в codesys
1	0.0	BYTE	Пуск	Mod_Pusc_M1
2	0.1	BYTE	Вперед	Mod_Back_M1
3	0.2	BYTE	Назад	Mod_Forward_M1
4	0.3	BYTE	Сброс аварии	Mod_Reset_M1
5	1	WORD	Скорость	Mod_Speed_M1
6	2	WORD	Код аварии	Mod_NumAlarm_M1

Таблица 16 – Регистры драйвера управления электродвигателя U3.

№	№ Регистра	Тип данных	Комментарий	Имя переменной в codesys
1	0.0	BYTE	Пуск	Mod_Pusc_M2
2	0.1	BYTE	Вперед	Mod_Back_M2
3	0.2	BYTE	Назад	Mod_Forward_M2
4	0.3	BYTE	Сброс аварии	Mod_Reset_M2
5	1	WORD	Скорость	Mod_Speed_M2
6	2	WORD	Код аварии	Mod_NumAlarm_M2

В таблицах 15 и 16 приведены регистры данных по которым осуществляется управление драйверами и считывается текущее их состояние. Управление драйверами ПЛК 160М производит с помощью записи команд в нулевой и первый регистры. Драйвер управления электродвигателя передает свое состояние во втором регистре. ПЛК 160М производит чтение второго регистра и определяет наличие ошибки в работе драйвера. Команды управления «Пуск», «Вперед», «Назад» и «Сброс аварии» передаются в нулевом регистре соответствующих битах, которые указаны в таблицах 15 и 16. Для добавления нулевого регистра управления, используемого драйверами управления электродвигателями в «Universal Modbus device» необходимо добавить «8bit output module» и указать имена переменных в соответствии с таблицами 15 и 16. Для добавления переменных типа «WORD», необходимо добавить «Register output module» с функцией записи 0x06 «Preset single register» для переменной «Mod\_Speed\_M2» и «Register input module» для переменной «Mod\_NumAlarm\_M2» с функцией чтения 0x03 «Read Holding Registers».

Таблица 17 – Регистры датчика цвета.

№	№ Регистра	Тип данных	Комментарий	Имя переменной в codesys
1	0	WORD	Красный	Mod_red
2	1	WORD	Зеленый	Mod_green
3	2	WORD	Синий	Mod_blue

В таблице 17 приведены регистры, в которых передаются параметры измеренные датчиком цвета. Для задания регистров работы с датчиком цвета в codesys необходимо использовать «Register input module» с функцией чтения 0x03 «Read Holding Registers». Конфигурация дерева подключения «ModBus (Master)[VAR]» приведена на рисунке 43.

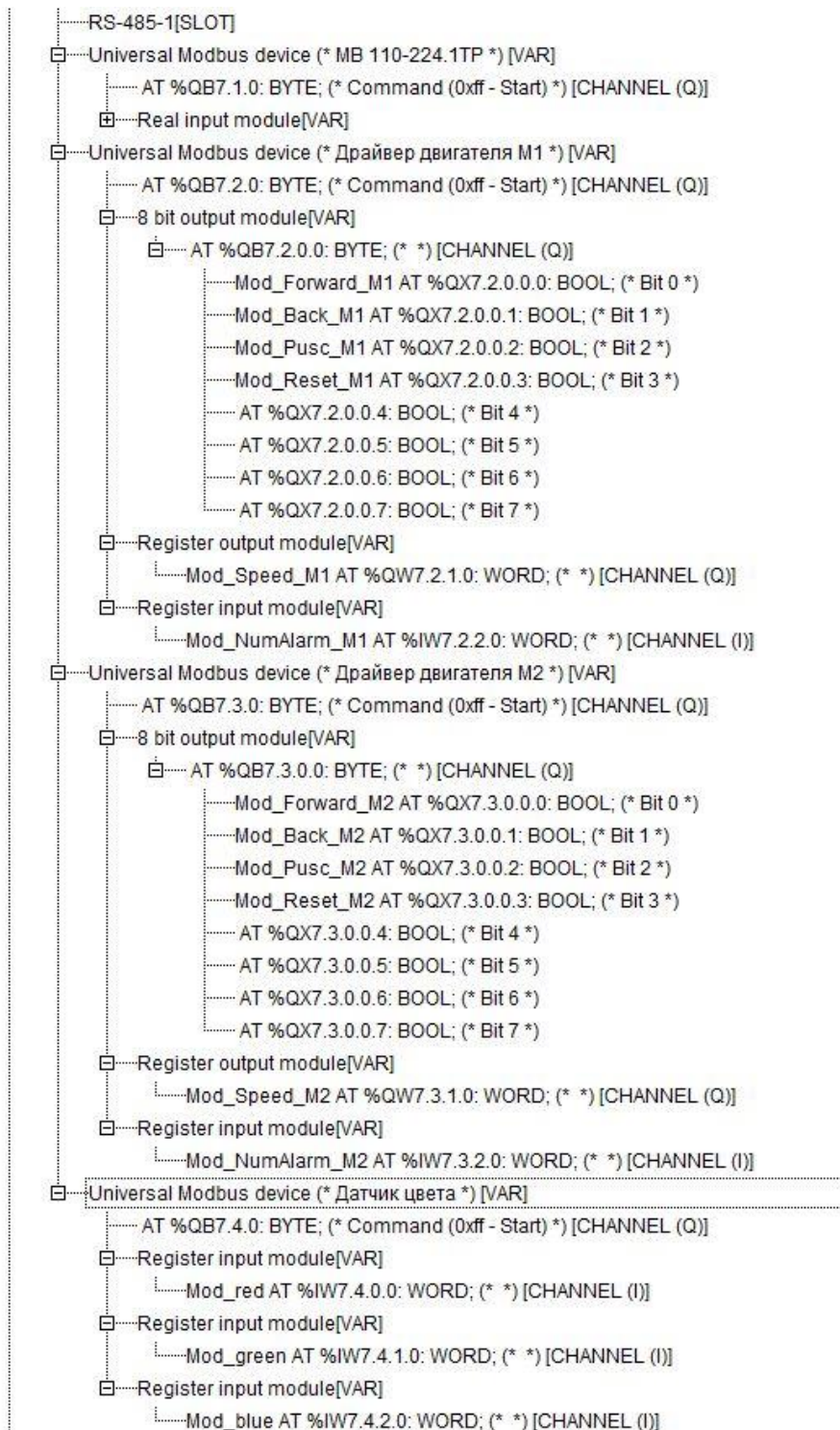


Рисунок 43 – Дерево конфигурации «ModBus (Master)[VAR]».



Далее для создания программы управления мехатронным стендом перейдем в дереве проекта на вкладку «POU». Поскольку проект содержит в своем составе определенное количество повторяющихся элементов, воспользуемся шаблонным методом, т.е. создадим функциональные блоки управления элементами, которые впоследствии будем тиражировать. Для создания функционального блока управления пневмоцилиндром необходимо открыть контекстное меню, путем нажатия правой клавиши мыши по элементу «POU» находящемуся в дереве проекта и выбрать пункт «Добавить объект». Откроется окно «Новый программный компонент (POU)» в котором необходимо задать имя нового POU, выбрать язык реализации и указать тип POU. Заполним данное окно как показано на рисунке 44.

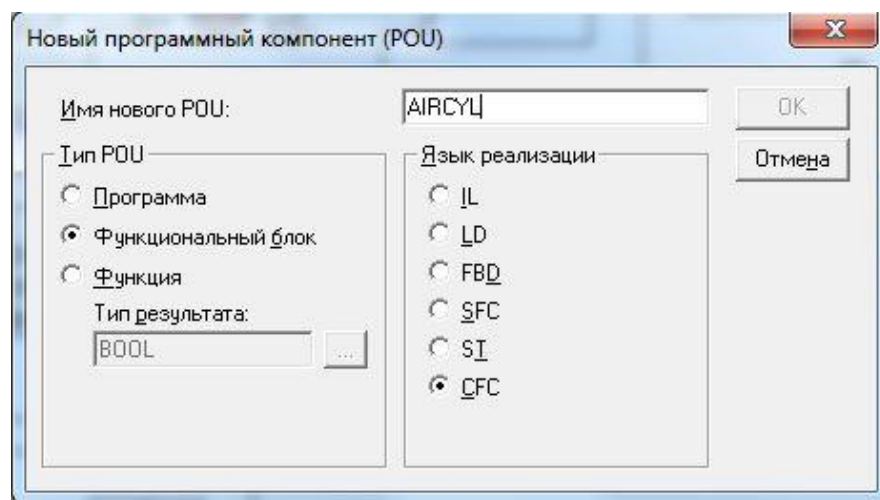


Рисунок 44 – Окно «Новый программный компонент (POU)»

После указания данных настроек откроется окно редактора созданного функционального блока. В данном окне создадим программу управления пневмоцилиндром как показано на рисунке 44.

Зададим необходимые переменные данного блока. Редактор функционального блока имеет три области описания переменных: VAR\_INPUT, VAR\_OUTPUT и VAR. VAR\_INPUT – это область описания входных переменных функционального блока, т.е. в данной области объявляются переменные которые будут поступать из основной программы. VAR\_OUTPUT – это область описания выходных переменных функционального блока, т.е. в данной области объявляются переменные которые будут поступать в основную программу из функционального

блока. Область VAR служит для задания переменных которые используются в границах функционального блока.

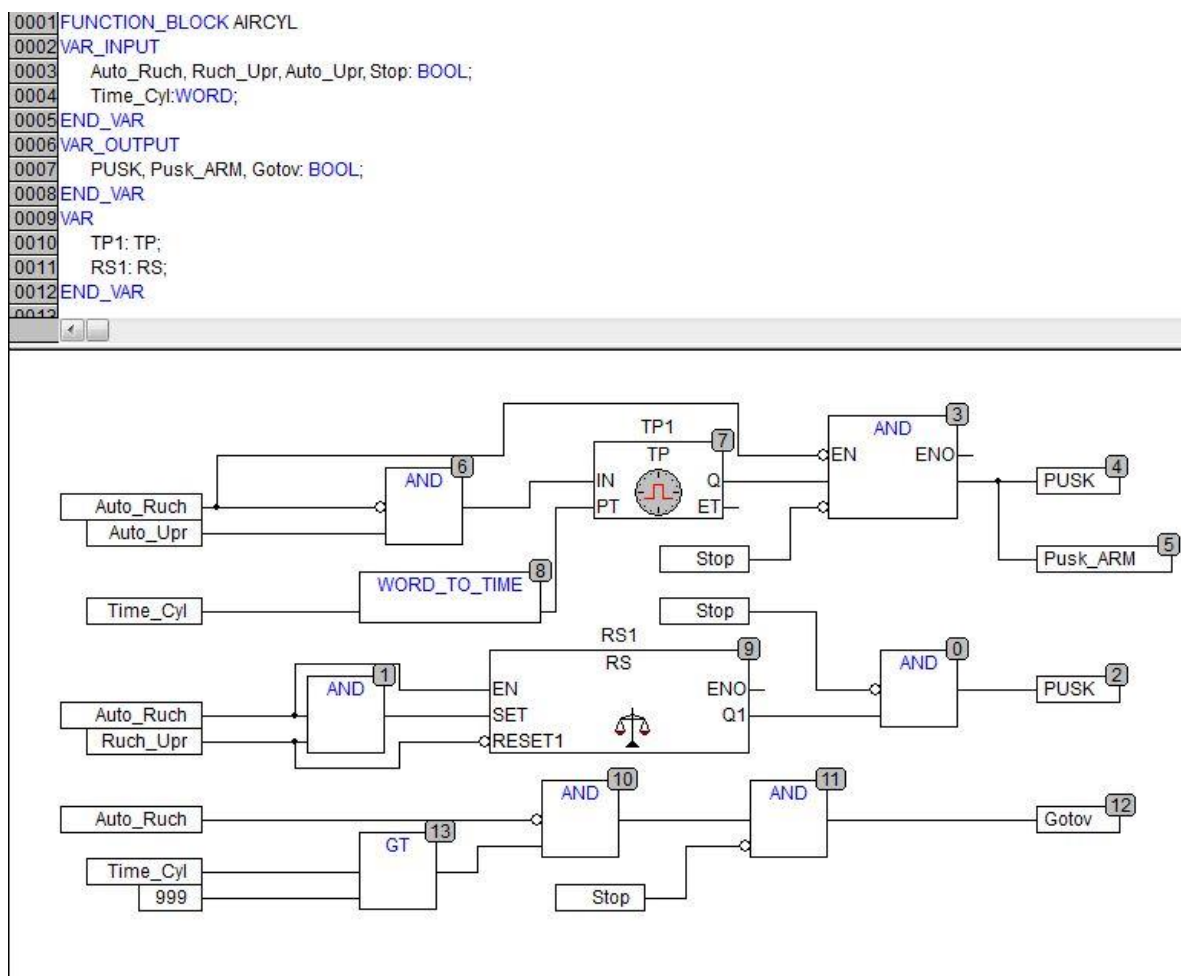


Рисунок 45 – Программа управления пневмоцилиндром.

Как видно из рисунка 45, функциональный блок содержит пять входных переменных и три выходные переменные. Переменная «Auto\_Ruch» является флагом в каком из режимов осуществляется управление пневмоцилиндром (Ручной или автоматический). Переменная «Ruch\_Upr» служит для запуска и остановки пневмоцилиндра в ручном режиме. Переменная «Auto\_Upr» запускает и останавливает пневмоцилиндр в автоматическом режиме. Переменная «Stop» останавливает работу пневмоцилиндра в любом режиме. Все вышеперечисленные переменные имеют тип данных BOOL. Переменная «Time\_Cyl» типа WORD, служит для задания времени работы пневмоцилиндра, т.е. значение переменной указывает время сработки цилиндра и удержание его в сработанном положении. Переменная «Time\_Cyl» учитывается только в автоматическом режиме работы.

Все выходные переменные функционального блока имеют тип BOOL. Переменная «PUSK» служит для передачи сигнала на физический выход ПЛК, который замыкает цепь и подает на катушку пневмораспределителя управляющее напряжение. Переменная «Pusk\_ARM» передает сигнал о сработке пневмоцилиндра в АРМ. Переменная «Gotov» выдает состояние готовности работы блока в автоматическом режиме.

Для задания работы блока в ручном режиме необходимо записать в переменную «Auto\_Ruch» единицу, при этом управление пневмоцилиндром будет осуществляться с АРМа. В переменную «Gotov» будет записано значение ноль, что будет говорить о неготовности пуска станда в автоматическом режиме. Готовность блока для работы в автоматическом режиме будет только при отсутствии записанной единицы в переменной «Auto\_Ruch», а так же необходимо наличие записанного значения в переменную «Time\_Cyl» не менее 1000 мс (данное ограничение вводится исходя из паспортных данных на пневмоцилиндр, т.е. это минимальное значение времени за которое пневмоцилиндр выдвинет шток на максимум).

Аналогичным образом создадим функциональный блок управления конвейером. Конвейер приводится в движение с помощью электродвигателя постоянного тока находящегося под управлением драйвера управления электродвигателем. Зададим имя функционального блока «Mot\_driver».

Зададим переменные функционального блока как показано на рисунке 46.

0001	FUNCTION_BLOCK Mot_driver
0002	VAR_INPUT
0003	Auto_Ruch, Revers, Ruch_Upr, Auto_Upr, Reset_in, DK, STOP: BOOL;
0004	Alarm_in: WORD;
0005	Speed_in: WORD;
0006	END_VAR
0007	VAR_OUTPUT
0008	Pusk, Forward, Back, Reset_out, Alarm_out: BOOL;
0009	Speed_out: WORD;
0010	Gotov: BOOL;
0011	END_VAR
0012	VAR
0013	Rabota, Alarm_pr: BOOL;
0014	END_VAR

Рисунок 46 – Переменные функционального блока «Mot\_driver».

Соберем управляющие цепи функционального блока «Mot\_driver» как показано на рисунке 47.

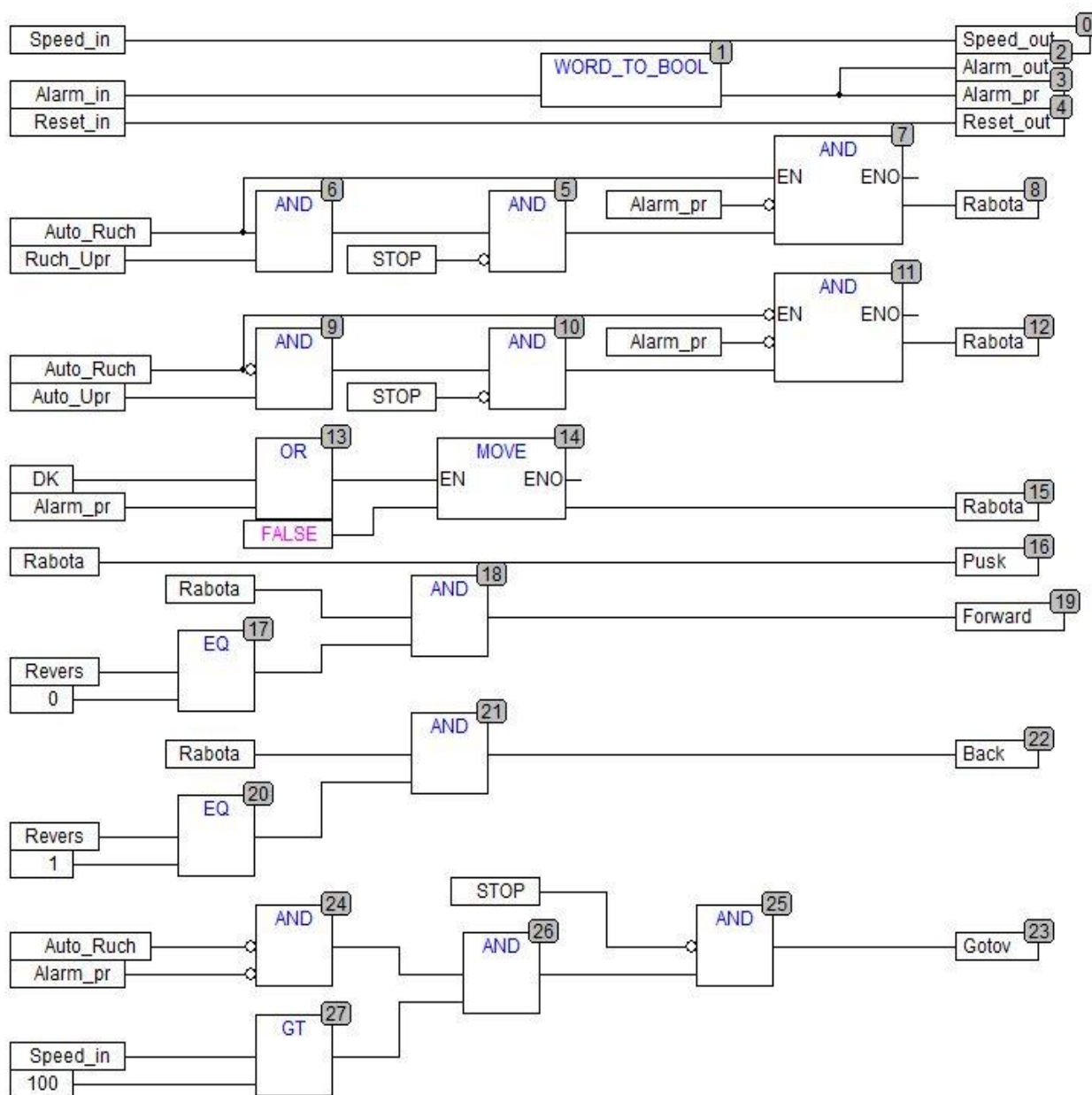


Рисунок 47 – Цепи управления функционального блока «Mot\_driver».

Данный блок содержит девять входных переменных, семь выходных и две внутренние переменные. Переменная «Auto\_Ruch», также как и в функциональном блоке управления пневмоцилиндром, является флагом в каком из режимов будет осуществляется управление драйвером управления электродвигателя. Переменные «Ruch\_Upr» и «Auto\_Upr» также как и в функциональном блоке управления пневмоцилиндром, осуществляют запуск и остановку устройства в соответствующем режиме (ручном/автоматическом). Переменная «Revers» задает

направление вращения электродвигателя. Входная переменная «Speed\_in» имеет тип WORD и служит для передачи значения скорости вращения электродвигателя, от АРМа в драйвер управления электродвигателем. Значение входной переменной «Speed\_in» передается в выходную переменную «Speed\_out». В переменную «Alarm\_in» записывается код ошибки от драйвера управления электродвигателем и передает статус ошибки в АРМ через переменную «Alarm\_out», а также записывает статус во внутреннюю переменную «Alarm\_pr». Переменная «Reset\_in» служит для сброса возможной ошибки связи и передает сигнал из АРМа в драйвер управления электродвигателем. Выходная переменная «Rabota» является управляющей, при записи в данную переменную единицы в драйвер подается сигнал включения электродвигателя. Переменные «Forward» и «Back» указывают драйверу направление вращения электродвигателя, по умолчанию драйвер настроен на движение вперед.

Для задания работы блока в ручном режиме необходимо записать в переменную «Auto\_Ruch» единицу, при этом управление пневмоцилиндром будет осуществляться с АРМа. В переменную «Gotov» будет записано значение ноль, что будет говорить о неготовности пуска станда в автоматическом режиме. Для запуска электродвигателя в автоматическом режиме необходимо указать скорость вращения электродвигателя, указать режим работы авто, а также предварительно настроить направление вращения электродвигателя. Входные переменные «STOP» и «DK» останавливает электродвигатель в любом из режимов работы электродвигателя. В переменную «DK» записывается сигнал от концевого оптического датчика. Данная функция реализована с целью недопущения столкновения заготовки с конвейера.

Функциональный блок управления линейным актуатором. Линейный актуатор приводится в движение с помощью электродвигателя постоянного тока. Схема управления электродвигателем приведена на принципиальной схеме которая приложена к диссертации. Как видно из принципиальной схемы управление линейным актуатором осуществляется при помощи двух реле К3 и К4. Данные реле выполняют функцию гальванической развязки между цепями управления и цепями коммутации актуатора.

Зададим имя функционального блока «АКТУАТОР». Переменные входящие в состав функционального блока показаны на рисунке 48.

```

0001 FUNCTION_BLOCK ACTUATOR
0002 VAR_INPUT
0003     Auto_Ruch, R_Pusk_forward, R_Pusk_back, Ishod: BOOL; (*Сигналы из АРМ*)
0004     Time1, Time2, Time3: WORD; (*Время на перемещение из точки в точку. Сигналы из АРМ.*)
0005     APusk_Forward, APusk_Back: BOOL; (*Сигналы запуска в режиме АВТО*)
0006 END_VAR
0007 VAR_OUTPUT
0008     PUSK_For, PUSK_Back: BOOL; (*Выходы ПЛК*)
0009     Gotov: BOOL;
0010 END_VAR
0011 VAR
0012     TP1: TP;
0013     RS1: RS;
0014 END_VAR

```

Рисунок 48 – Переменные функционального блока «АКТУАТОР».

Цепи управления функционального блока «АКТУАТОР» приведены на рисунке 49.

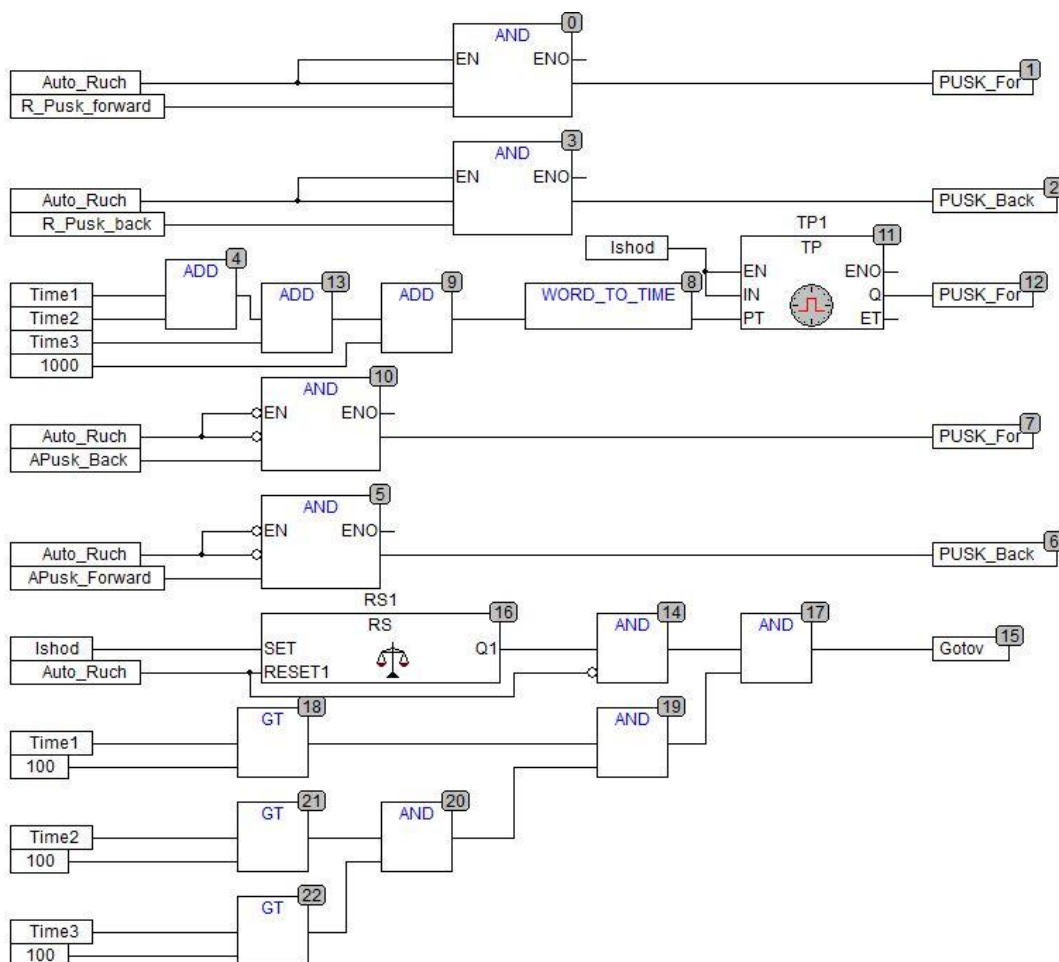


Рисунок 49 – Цепи управления функционального блока «АКТУАТОР».

Функциональный блок управления актуатора имеет девять входных, три выходные переменные. Переменная «Auto\_Ruch», также как и в функциональных

блоках описанных выше, является флагом задающим режим работы блока. Переменные «R\_Pusk\_forward» и «R\_Pusk\_back» имеют тип BOOL. В данные переменные записываются команды управления линейным актуатором в ручном режиме. Переменная «R\_Pusk\_forward» дает команду на движение штока вперед, а «R\_Pusk\_back» соответственно дает команду на движение штока в обратном направлении. Переменные «Time1», «Time2» и «Time3» имеют тип WORD и задают время движения штока между точками. Переменная «Ishod» имеет тип BOOL. В нее записывается команда на перемещение штока в исходное положение. Исходное положение, в данном проекте, линейный актуатор занимает при полностью выдвинутом штоке. Переменные «APusk\_Forward» и «APusk\_Back», тип BOOL, служат для управления линейным актуатором в автоматическом режиме работы. Переменная «Gotov» имеет тип BOOL, ее назначение такое же как и вышеописанных функциональных блоках. Сигнал о готовности работы в автоматическом режиме, как видно из рисунка 47, функциональный блок выдаст при заданных значениях времени перемещения от точки к точке, а так же при выполненном условии, линейный актуатор должен занять исходное положение.

Функциональный блок определения цвета. Данный блок работает с датчиком цвета, он принимает данные от датчика и передает значения в программу. Имя функционального блока «OPR\_Cvet». Переменные входящие в состав функционального блока показаны на рисунке 50.

0001	FUNCTION_BLOCK OPR_Cvet
0002	VAR_INPUT
0003	Reset: BOOL; (*Сброс блока.*)
0004	Red_in, Green_in, Blue_in: WORD;
0005	Rad_inARM, Green_inARM, Blue_inARM: WORD;
0006	END_VAR
0007	VAR_OUTPUT
0008	Red_out, Green_out, Blue_out: WORD;
0009	Red, Green, Blue: BOOL;
0010	Cvet: WORD;
0011	PR_zag_na_stol: BOOL;
0012	END_VAR
0013	VAR
0014	Mov: BOOL;
0015	SR1, SR2, SR3: SR;
0016	END_VAR

Рисунок 50 – Переменные функционального блока «OPR\_Cvet».

Цепи функционального блока «OPR\_Cvet» приведены на рисунке 51.

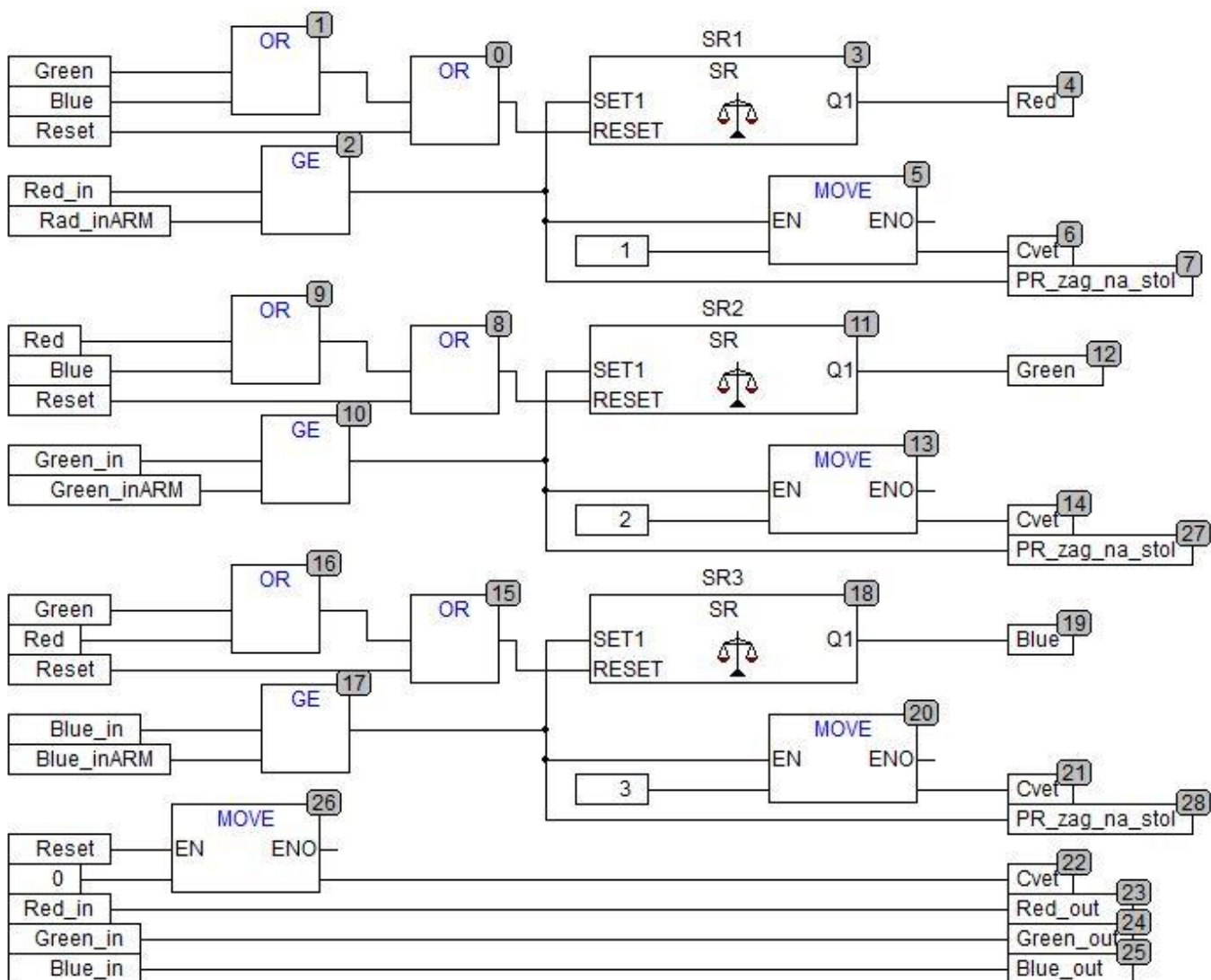


Рисунок 51 – Цепи управления функционального блока «OPR\_Cvet».

Переменные «Red\_in», «Green\_in» и «Blue\_in» предназначены для записи в функциональный блок значений, которые поступают от датчика цвета. Переменные «Red\_inARM», «Green\_inARM» и «Blue\_inARM» предназначены для установки порогов сработки. Данные значения записывает оператор системы, АРМ окно «Настройка датчика цвета». Переменные «Red\_out», «Green\_out» и «Blue\_out» служат для передачи мгновенных показаний датчика цвета в АРМ. Переменные «Red», «Green» и «Blue» являются флагами для задания работы программы ПЛК. В переменную «Cvet» записывается значение цвета заготовки, далее эта переменная передается в АРМ. Переменная «Reset» служит для сброса функционального блока, как при ошибках работы станда, так и при сбросе работы цикла. Переменная



«PR\_zag\_na\_stol» служит флагом программы который сигнализирует о том что заготовка находится в технологической зоне стол.

Функциональный блок «Ves» служит для обработки сигналов поступающих от модуля MB 110. Переменные функционального блока «Ves» приведены на рисунке 52.

0001	FUNCTION_BLOCK Ves
0002	VAR_INPUT
0003	Ves_in: REAL; (*Вес сигнал от преобразователя*)
0004	END_VAR
0005	VAR_OUTPUT
0006	Zag_na_ves: BOOL;
0007	Ves_tec, Ves_tec_zag: WORD; (*Вес сигнал в арм. Моментальный, текущей заготовки*)
0008	END_VAR
0009	VAR
0010	TON1: TON;
0011	END_VAR

Рисунок 52 – Переменные функционального блока «Ves».

Данный блок содержит входную переменную «Ves\_in». В переменную «Ves\_in» записываются моментальные показания от тензометрического датчика.

Выходные переменные «Ves\_tec» и «Ves\_tec\_zag» служат для передачи данных веса заготовки в АРМ и программу ПЛК. Выходная переменная «Zag\_na\_ves» является флагом, который сигнализирует в программу ПЛК о том что заготовка находится на весах.

Цепи управления функционального блока «Ves» приведены на рисунке 53.

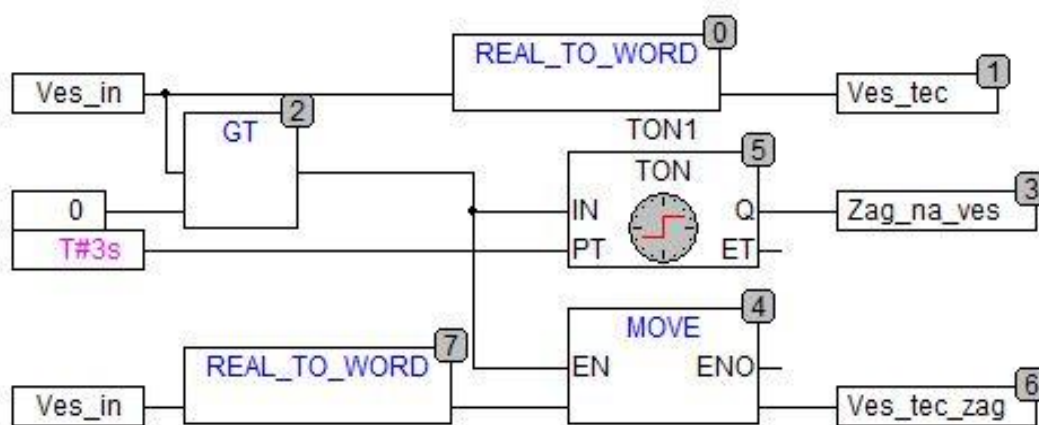


Рисунок 53 – Цепи управления функционального блока «Ves».

Функциональный блок определения материала заготовок «OPR\_material». Данный блок обрабатывает сигналы, поступающие от датчика индуктивности, от

функционального блока «Ves» и оператора (АРМа). Переменные данного блока приведены на рисунке 54.

```
0001 FUNCTION_BLOCK Opr_material
0002 VAR_INPUT
0003     Reset: BOOL; (*Сброс цикла*)
0004     Metal_D3: BOOL; (*Сигнал от датчика индуктивности*)
0005     Ves, vusota_H, Radius_R, Ro_plastik, Ro_wood: WORD;
0006 END_VAR
0007 VAR_OUTPUT
0008     Gotov: BOOL; (*Готовность*)
0009     Metal, Wood, Plastik: BOOL;
0010     Material_ARM: WORD;
0011 END_VAR
0012 VAR
0013     V_zag, Ro_plotnosti: REAL;
0014     RS1, RS2, RS3: RS;
0015 END_VAR
```

Рисунок 54 – Переменные функционального блока «OPR\_material».

Входная переменная «Reset» служит для сброса значений выходных переменных которые генерирует функциональный блок. В переменную «Metal\_D3» поступает сигнал от индуктивного датчика. При наличии металла в заготовке в переменную записывается сигнал логической единицы. Переменная «Ves» служит для работы с данными поступающими от функционального блока «Ves». Переменные «vusota\_H» и «Radius\_R» характеризуют геометрические характеристики заготовок и поступают из АРМа. В переменные «Ro\_plastik» и «Ro\_wood» оператор записывает значения плотности заготовок. Запись в переменных «vusota\_H», «Radius\_R», «Ro\_plastik» и «Ro\_wood» происходит в окне АРМ «Настройки заготовок».

Функциональный блок определяет, из какого материала выполнена заготовка путем сравнения плотности поступившей заготовки с плотностью заданной оператором. Выходные переменные «Metal», «Wood» и «Plastik» предназначены для внутренней логики работы программы ПЛК. Переменная «Material\_ARM» служит для передачи в АРМ значения материала заготовки.

Цепи управления функционального блока «OPR\_material» приведены на рисунке 55.

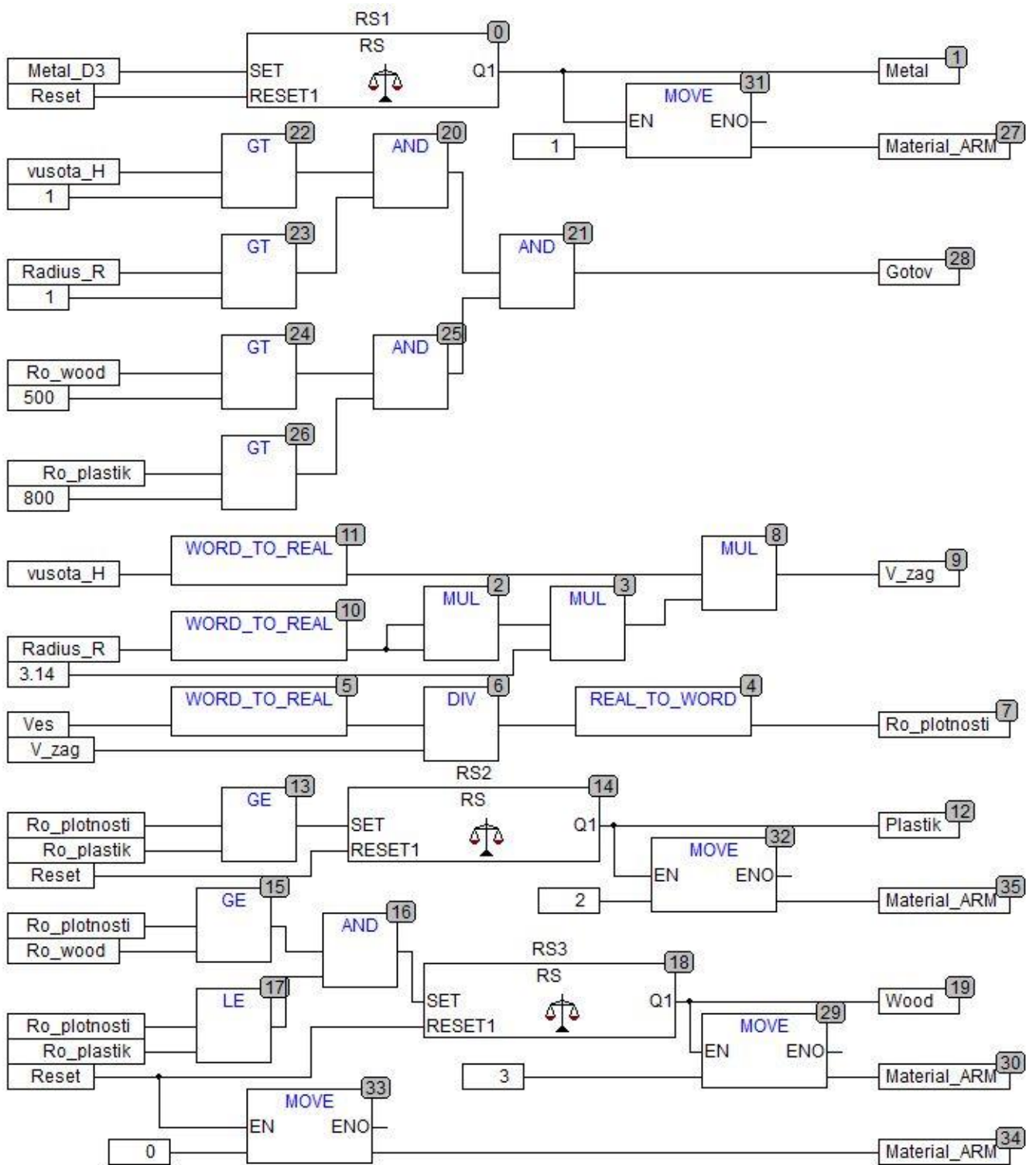


Рисунок 55 – Цепи управления функционального блока «OPR\_material».

Функциональный блок «Sort» предназначен для работы с ячейкой хранения. В данный функциональный блок поступают сигналы о характеристиках текущей заготовки, которые сравнивает данный блок с уставками оператора. Переменные функционального блока «Sort» приведены на рисунке 56.

Цепи управления функционального блока «Sort» приведены на рисунке 57.

```

0001 FUNCTION_BLOCK Sort
0002 VAR_INPUT
0003   DK: BOOL; (*Сигнал от концевика хранилища.*)
0004   Red, Green, Blue, Metal, Wood, Plastik: BOOL; (*Переменные из АРМ. Указания хранения. Признаки*)
0005   Red_Prog, Green_Prog, Blue_Prog, Metal_Prog, Wood_Prog, Plastik_Prog: BOOL; (*Переменные из программы (из поля)*)
0006   Reset: BOOL; (*Сброс блока*)
0007 END_VAR
0008 VAR_OUTPUT
0009   Stop_Con2, Srab_Cyl: BOOL; (*Команда на запуск цилиндра*)
0010 END_VAR
0011 VAR
0012   NetZadania: BOOL; (*Нет задания*)
0013   RS1: RS;
0014 END_VAR

```

Рисунок 56 – Переменные функционального блока «Sort».

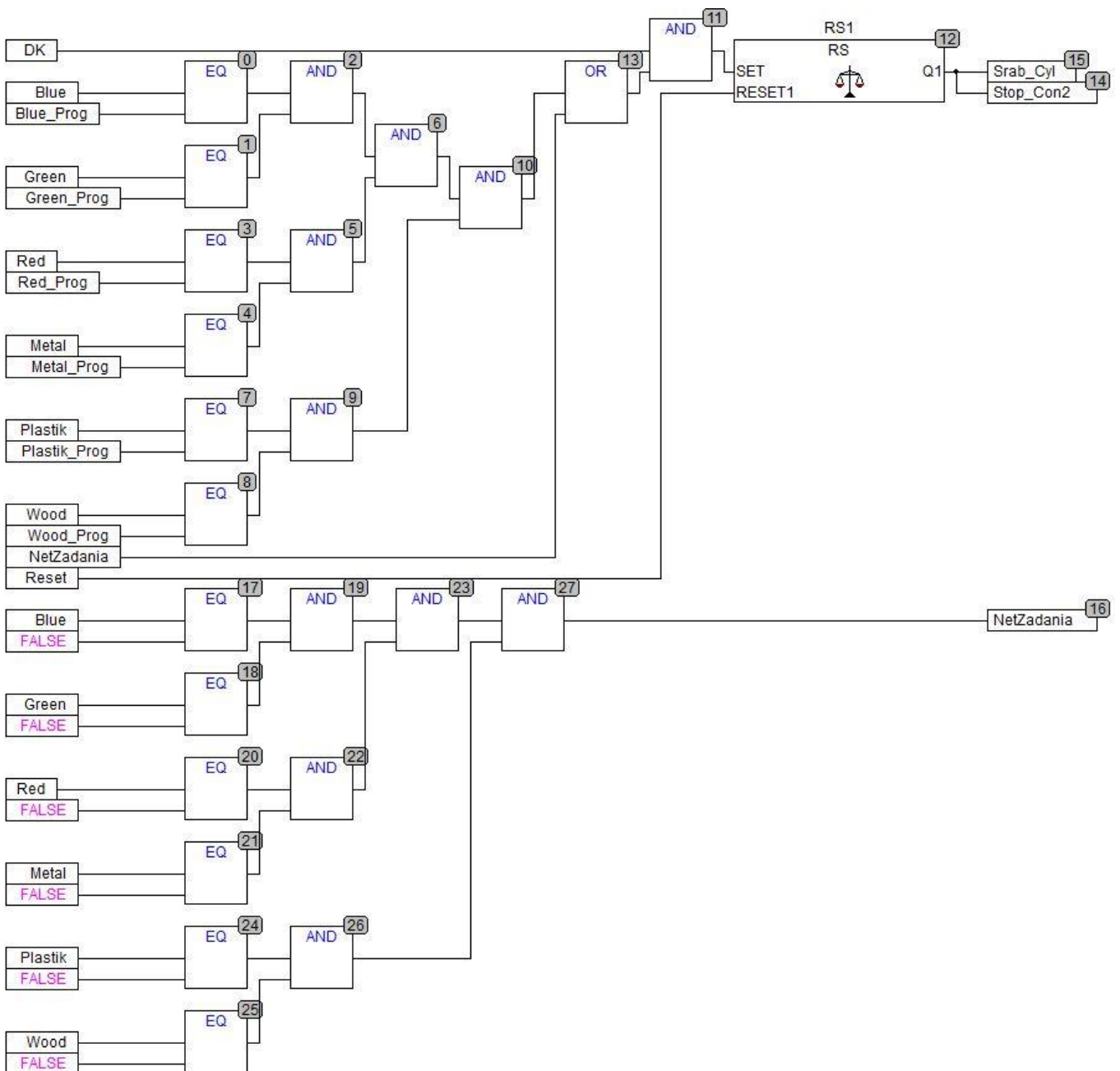


Рисунок 57 – Цепи управления функционального блока «Sort».

Функциональный блок «Pr» управляет процессом переносом заготовок из технологической зоны приема заготовок в зону хранения. Переменные функционального блока «Pr» приведены на рисунке 58.

0001	FUNCTION_BLOCK Pr
0002	VAR_INPUT
0003	Start, Stop: BOOL; (*Запуск/Остановка сортировки*)
0004	DK1, D1, D2: BOOL; (*Входные сигналы текущие*)
0005	Akt_Time1, Akt_Time2, Akt_Time3: WORD; (*Время переноса актуатора*)
0006	C3_Time: WORD; (*Время сработки Ц3*)
0007	(*Статусы заготовок. Свойства.*)
0008	END_VAR
0009	VAR_OUTPUT
0010	PR_Pusk_con1, PR_Pusk_Cyl1, PR_Pusk_Cyl3, PR_Pusk_P4, PR_Pusk_con2: BOOL; (*Переменные запуска устройств.*)
0011	PR_Pusk_Akt_Forward, PR_Pusk_Akt_Back: BOOL; (*Переменные запуска актуатора.*)
0012	D_T1, D_T2, D_T3: BOOL; (*Актуатор. Статусы разрешения перемещения. Сигналы в АРМ*)
0013	END_VAR
0014	VAR
0015	Akt_v_ishod_reset: BOOL;
0016	Pusk_1, Pusk_2, Pusk_3: BOOL;
0017	Sbros_cikl: BOOL; (*Сброс текущего цикла*)
0018	Schet: WORD;
0019	Sh1, Sh2, Sh3, Sh4: BOOL; (*Для счетчика цикла*)
0020	RS1: RS;
0021	Pusk_pr: BOOL; (*Общий пуск функционального блока.*)
0022	Pusk_Con1_Cyl1, Stop_Con1_Cyl1: BOOL; (*Зона выдачи*)
0023	Pusk_Cyl3: BOOL; (*Перенос с 1-го конвейера на весы*)
0024	Pusk_Akt_Back_1: BOOL;
0025	RS2: RS;
0026	RS3: RS;
0027	CTU1: CTU;
0028	TON1: TON;
0029	RS4: RS;
0030	TP1: TP;
0031	TOF1: TOF;
0032	RS5: RS;
0033	TOF2: TOF;
0034	TON2, TON3, TON4: TON;
0035	CTU2: CTU;
0036	Perenos1, Perenos2, Perenos3: Perenos;
0037	RS100, RS101, RS102: RS;
0038	TON100, TON101, TON102: TON;
0039	RS103: RS;
0040	TP103: TP;
0041	F_Trig103: F_TRIG;
0042	TP105: TP;
0043	TON110: TON;
0044	TOF100, TOF101, TOF102: TOF;
0045	Pusk_Con2_pr, Stop_Con2_pr: BOOL;
0046	RS105: RS;
0047	END_VAR

Рисунок 58 – Переменные функционального блока «Pr».

При записи логической единицы во входную переменную «Start» запускается процесс сортировки заготовок. Входная переменная «Stop» останавливает процесс

сортировки на любом из этапов сортировки и переводит систему в исходное состояние. Входные переменные «DK1», «D1» и «D2» являются флагами, которые сигнализируют текущее положение заготовки в технологическом процессе. Переменные «Akt\_Time1», «Akt\_Time2» и «Akt\_Time3» служат для задания времени перемещения заготовки из технологических зон. «Akt\_Time1» - перенос из зоны выдачи на весы. «Akt\_Time2» - перенос с весов на стол. «Akt\_Time3» - перенос со стола в зону сортировки (конвейер 2). Запись переменных осуществляет оператор в окне «Актуатор». Входная переменная «C3\_Time» служит для считывания времени сработки пневмоцилиндра «Ц3», т.е. функциональный блок должен учитывать время пока пневмоцилиндр «Ц3» будет находиться в опущенном положении.

Выходные переменные «PR\_Pusk\_con1», «PR\_Pusk\_Cyl1», «PR\_Pusk\_Cyl3», «PR\_Pusk\_P4», «PR\_Pusk\_con2», «PR\_Pusk\_Akt\_Forward», «PR\_Pusk\_Akt\_Back» служат для запуска устройств входящих в состав зоны выдачи, зоны сортировки и механической руки. Выходные переменные «D\_T1», «D\_T2», «D\_T3» являются флагами которые предназначены для передачи в АРМ информации о положении заготовки.

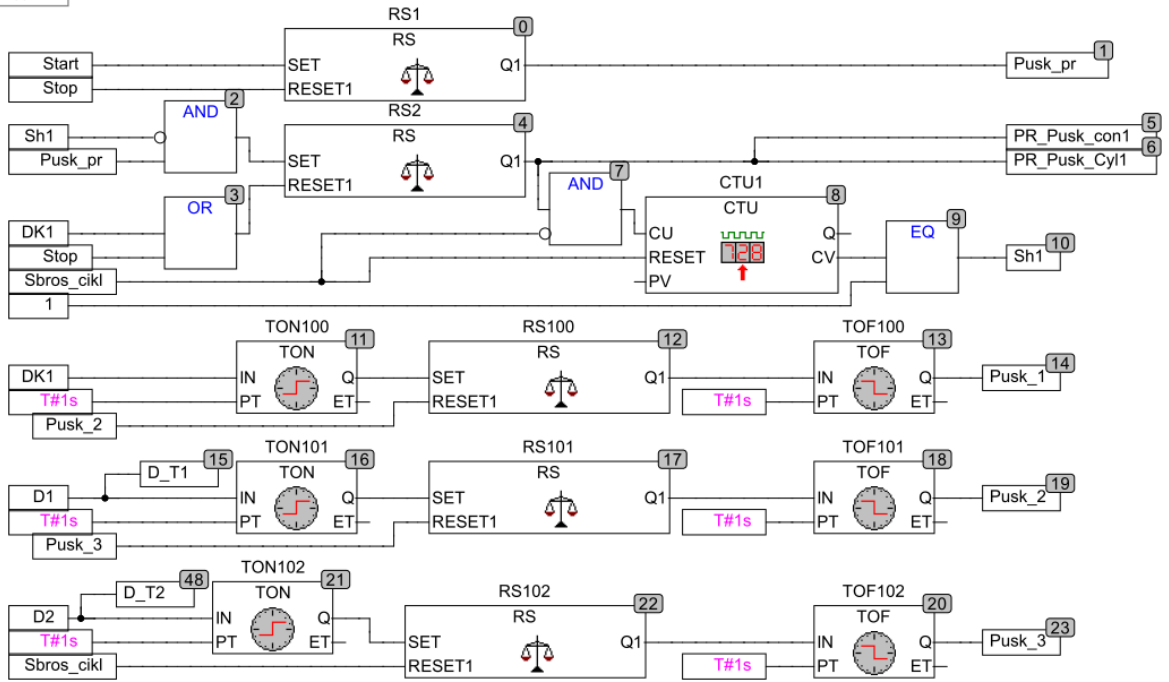
Цепи управления функционального блока «Pr» приведены на рисунке 59.

Как видно из рисунка 59 функциональный блок «Pr» содержит функциональный блок «Perenos». Данный функциональный блок был создан для упрощения написания блока «Pr». В блоке «Perenos» реализован алгоритм управления механической руки. Данный алгоритм выполняется циклично, за один цикл программы механическая рука выполняет три раза аналогичные действия. Выполняет перенос заготовки с приемного конвейера на весы, с весов на стол, со стола на конвейер сортировки.

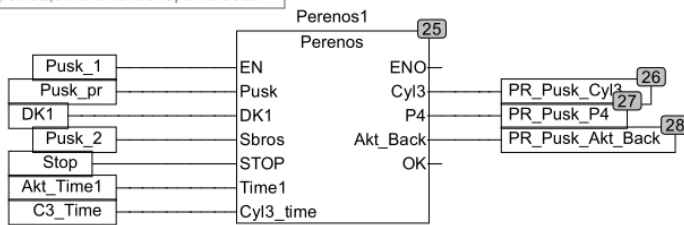
Внутренняя переменная «Sbros\_cikl» формируется на выходе третьего функционального блока «Perenos» и запускает алгоритм возвращения актуатора в исходное положение.

Внутренняя переменная «Akt\_v\_ishod\_reset» формируется по завершению алгоритма возвращения актуатора и служит для автоматического сброса цикла переноса заготовки.

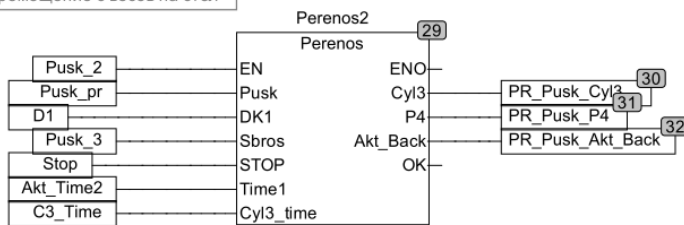
Зона выдачи



Перемещение с конвейера на весы



Перемещение с весов на стол



Перемещение со стола на конвейер 2

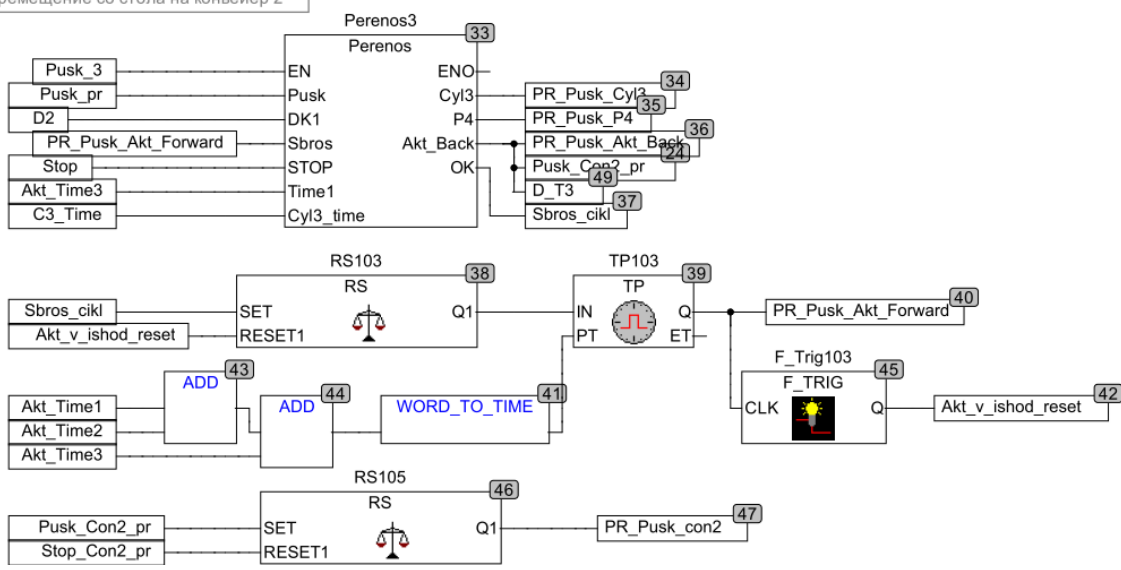


Рисунок 59 – Цепи управления функционального блока «Pг».

В функциональном блоке «Perenos» реализован алгоритм управления механической рукой. Переменные функционального блока «Perenos» приведены на рисунке 60.

```
0001 FUNCTION_BLOCK Perenos
0002 VAR_INPUT
0003     Pusk, DK1, SbroS, STOP: BOOL;
0004     Time1: WORD;
0005     Cyl3_time: WORD;
0006 END_VAR
0007 VAR_OUTPUT
0008     Cyl3, P4, Akt_Back: BOOL;
0009     OK: BOOL;
0010 END_VAR
0011 VAR
0012     Pusk_Akt_Cyl3, Pusk_Cyl3_2, Reset_P4, P4_Blok, Reset_cykl: BOOL;
0013     CV: WORD;
0014     R1: R_TRIG;
0015     TP1: TP;
0016     F1, F2, F3, F4, F5: F_TRIG;
0017     CTU1: CTU;
0018     TON1, TON2: TON;
0019     RS1, RS2, RS3: RS;
0020     TP2, TP3, TP4: TP;
0021 END_VAR
```

Рисунок 60 – Переменные функционального блока «Perenos».

Входная переменная «Pusk» служит для запуска алгоритма работы механической руки. Переменная «DK1» сигнализирует о том что заготовка находится в зоне начала запуска алгоритма. Переменная «SbroS» выполняет функцию сброса алгоритма в исходное состояние. Переменная «Time» предназначена для задания времени работы актуатора. Переменная «Time» - определяет за какое время актуатор перенесет заготовку из начальной точки в конечную. Входная переменная «Cyl3\_Time» служит для задания времени нахождения «Ц3» в сработавшем состоянии.

Выходная переменная «Cyl» запускает цилиндр «Ц3» в сработку на время переноса заготовки. Переменная «P4» запускает в работу присоску. «Akt\_Back» запускает в работу актуатор.

Цепи управления функционального блока «Perenos» приведены на рисунке 57.





## 5.2 Верхний уровень

Программная часть верхнего уровня системы управления мехтронным стендом состоит из OPC – сервера и программы визуализации технологического процесса. OPC – сервера служат в качестве универсального механизма обмена данными, между OPC – клиентами. Программа визуализации технологического процесса, как и программа управления ПЛК являются OPC – клиентами. Структурная схема взаимодействия между клиентами OPC – сервера приведена на рисунке 62.

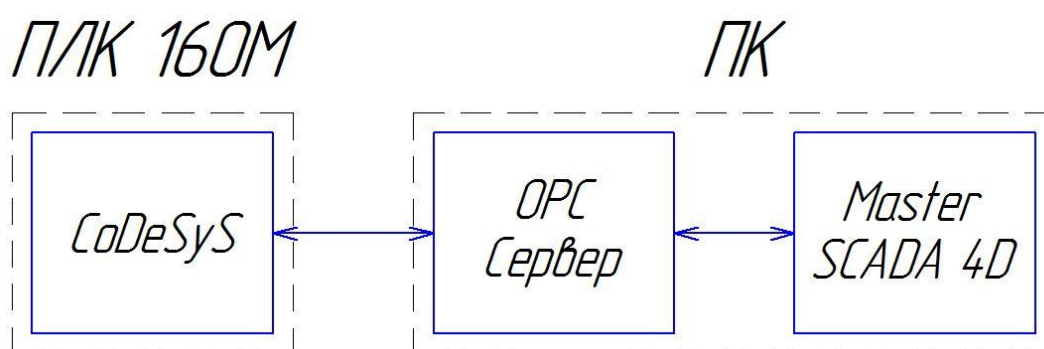


Рисунок 62 – Структурная схема взаимодействия OPC – клиентов.

### 5.2.1 OPC сервер

В системе управления мехатронным стендом применяется OPC – сервер от компании ИнСАТ. Modbus Universal Master OPC Server. Данный OPC – сервер поддерживает передачу данных в сетях Ethernet по протоколу Modbus TCP. Так же данный сервер может работать в сетях RS – 485 и RS - 232 по протоколу как Modbus RTU/ASCII. Modbus Universal Master OPC Server работает в операционных системах семейства windows и поддерживает технологии передачи данных OPC DA, OPC HAD, OPC UA, что позволяет утверждать о кроссплатформенности системы. Компьютер используемый в качестве АРМ, работает под управлением Windows 10 Pro версии, что позволяет гарантировать поддержку всех встроенных функций OPC – сервера.

Компания ИнСАТ предлагает программный продукт Modbus Universal Master OPC Server в различных вариантах, но бесплатно можно работать только с двумя версиями. Обе эти версии имеют определенные ограничения в работе. В первом варианте OPC – сервер имеет ограничения по числу тегов. В данной версии можно использовать 32 тега, без ограничений во времени работы сервера. Вторая версия не имеет ограничений по количеству тегов, но имеет ограничения во времени работы. Данная версия позволяет непрерывно работать без каких-либо ограничений в течении часа, после истечения времени сервер останавливает свою работу. Для восстановления работоспособности системы необходимо перезапустить OPC – сервер и работа будет продолжена. Главное окно Modbus Universal Master OPC Server приведено на рисунке 63.

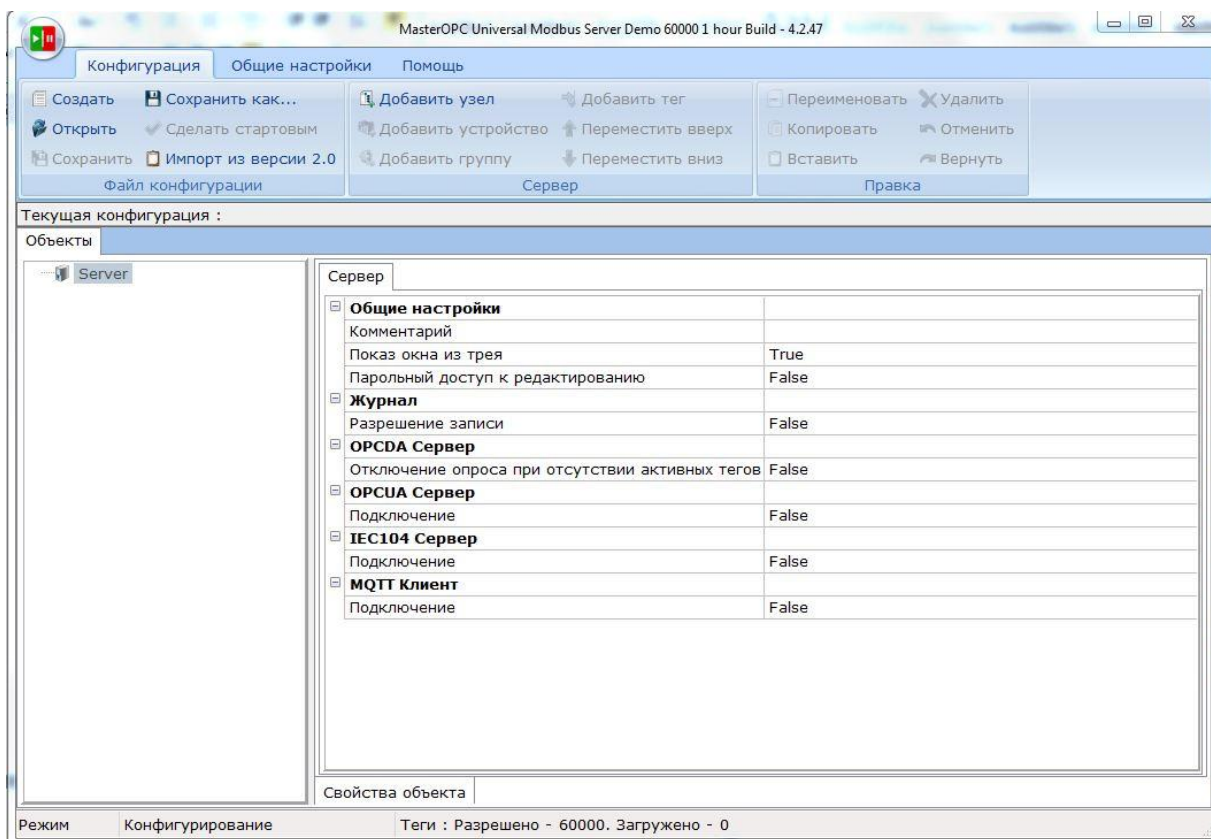


Рисунок 63 – Окно Modbus Universal Master OPC Server.

Для начала работы OPC – сервера необходимо произвести его конфигурацию, добавить и описать коммуникационные узлы, устройства и переменные (теги).

Первым шагом конфигурации OPC – сервера является добавление коммуникационного узла. Для этого необходимо нажать правой клавишей мыши по

объекту «Server», выбрать пункт «Добавить» и в контекстном меню выбрать «Коммуникационный узел». После нажатия левой клавишей мыши на строку «Коммуникационный узел», откроется окно «Редактирование коммуникационного узла», рисунок 64.

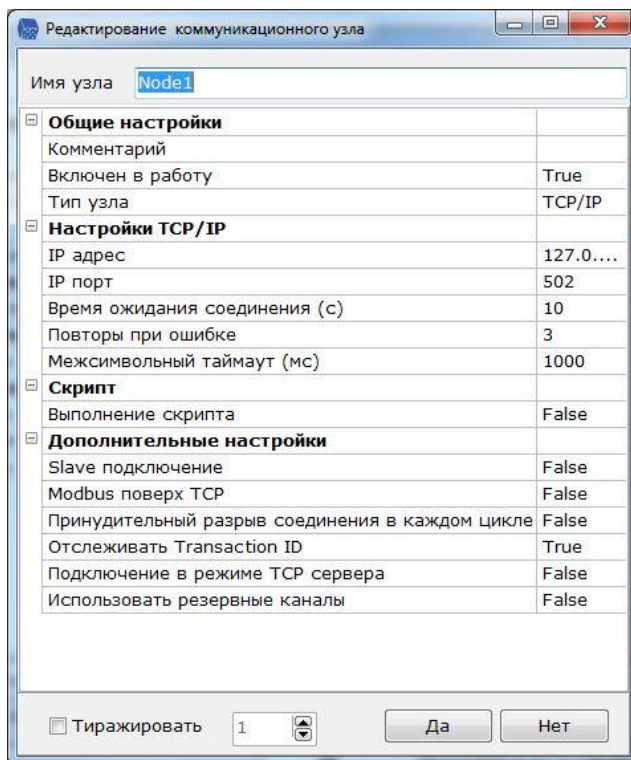


Рисунок 64 – Окно «Редактирование коммуникационного узла».

В данном окне необходимо присвоить имя коммуникационному узлу и задать параметры подключения. Изменим указанные значения параметров в соответствии с таблицей 18, остальные параметры оставим без изменения. После указания всех необходимых параметров в окне «Редактирование коммуникационного узла», необходимо нажать на кнопку «Да». При этом окно «Редактирование коммуникационного узла» закроется, а в дереве объектов появится коммуникационный узел «TCP подключение», рисунок 65.

Таблица 18 – Параметры коммуникационного узла.

№	Параметр	Значение
1	Имя узла	TCP подключение
2	Тип узла	TCP/IP
3	IP адрес	10.0.6.10

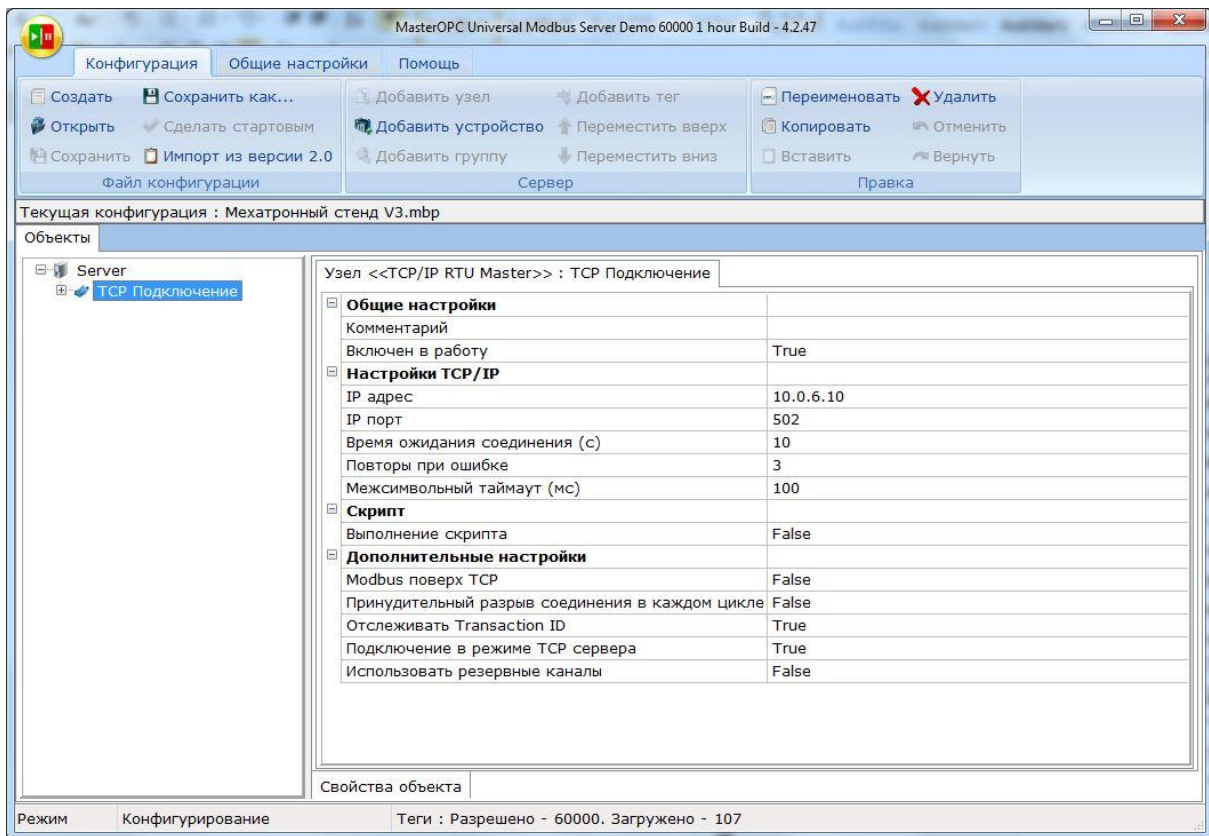


Рисунок 65 – Окно Master OPC Server с добавленным коммуникационным узлом.

Следующим шагом конфигурации OPC –сервера, является добавление устройства в коммуникационный узел. Для этого необходимо нажать левой клавишей мыши на коммуникационный узел и нажать на кнопку «Добавить устройство» во вкладке «Конфигурация», область «Сервер». После откроется окно «Редактирование устройства» изображенное на рисунке 66.

В данном окне необходимо присвоить имя устройству и задать параметры работы OPC – сервера с устройством. Изменим имя устройства на «PLC 160M». Зададим адрес устройства, для этого найдем в списке строку «Адрес» и изменим значение на «1». Важным параметром работы OPC – сервера является «Период опроса». Данный параметр отвечает за количество обращений сервера к устройству за единицу времени. Для достижения стабильных показателей работы системы, значение данного параметра необходимо установить на уровне до 200 мс. Зададим значение данного параметра равным «100» мс. После указания всех необходимых параметров в окне «Редактирование устройства», необходимо нажать на кнопку «Да». При этом окно «Редактирование устройства» закроется, а в дереве объектов появится устройство «PLC 160M», рисунок 67.

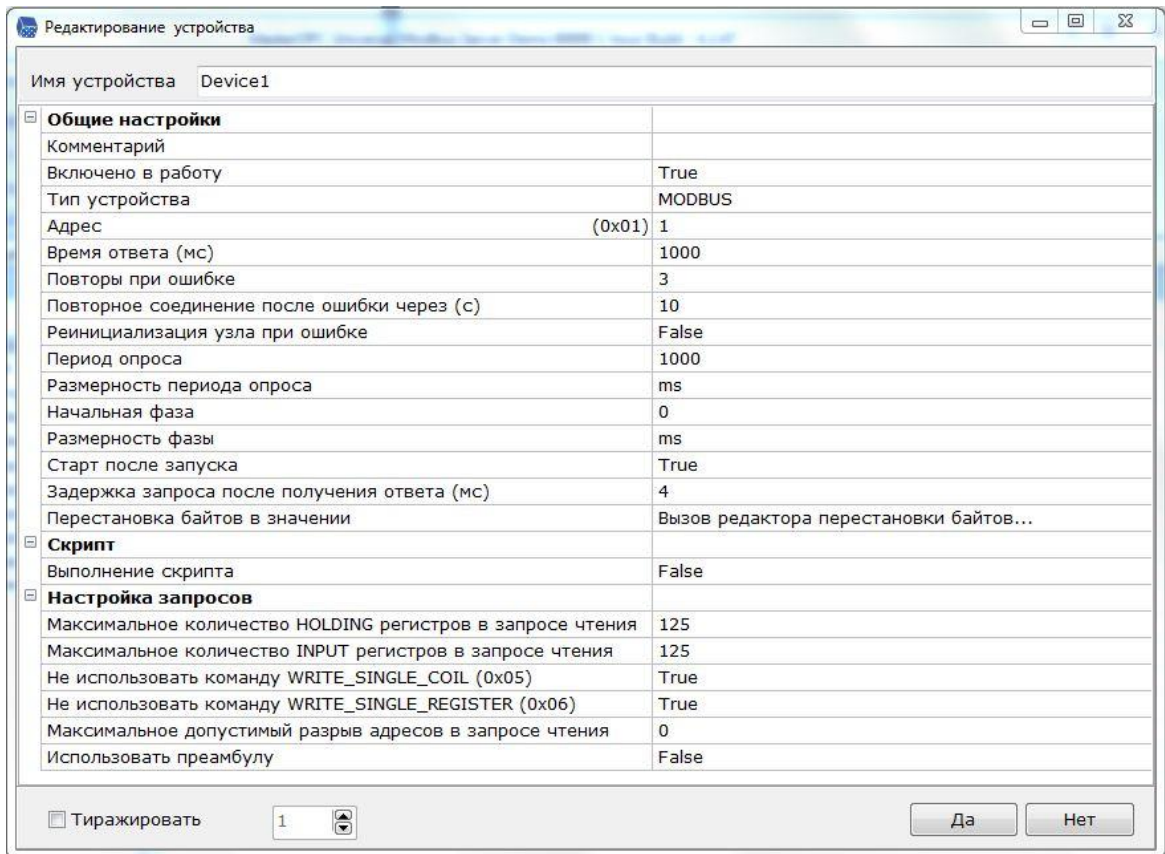


Рисунок 66 – Окно «Редактирование устройства».

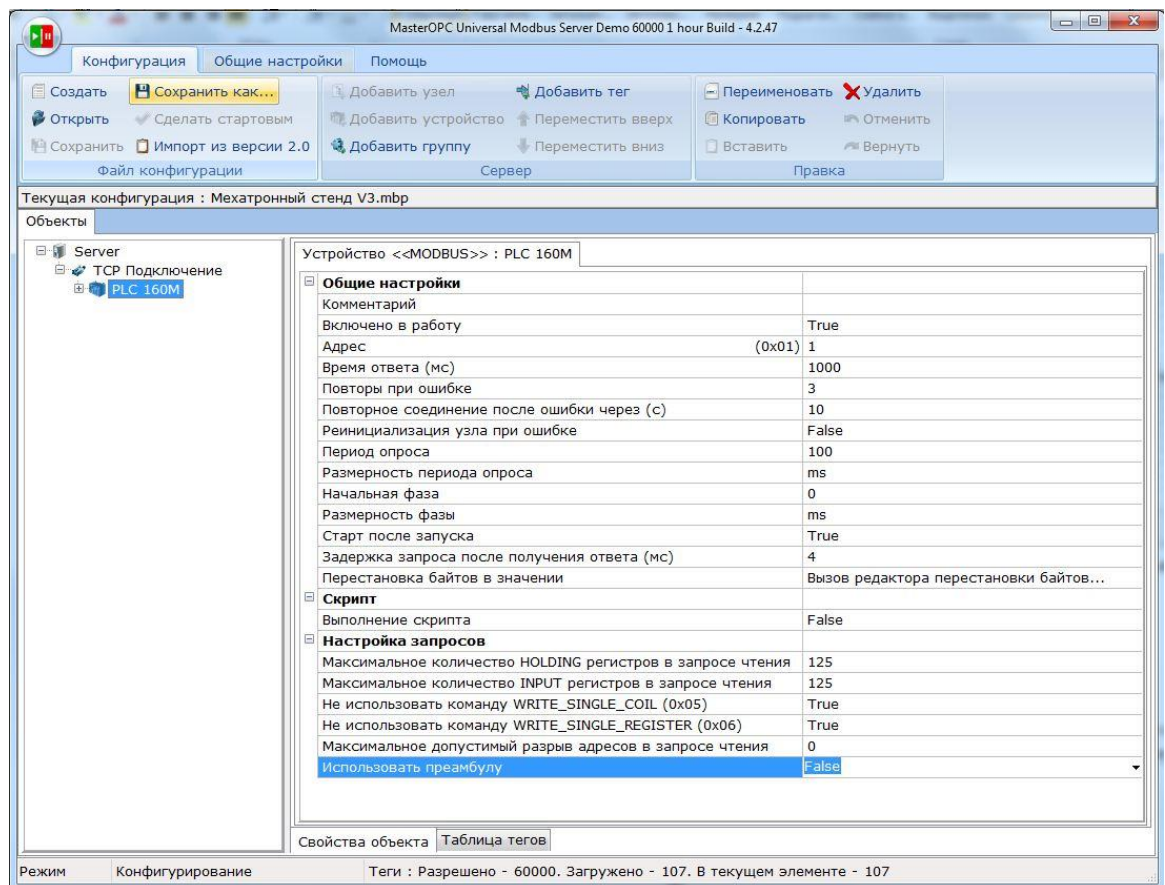


Рисунок 67 – Окно OPC Server с добавленным устройством

Завершающим этапом настройки OPC – сервера является добавление переменных (тегов). Добавить теги в конфигурацию OPC – сервера можно тремя способами, при помощи кнопок на вкладке «Конфигурация» в области «Сервер», либо через контекстное меню, либо при помощи импорта тегов. Самым удобным способом, при работе с большим объемом переменных является способ импорта тегов.

В Modbus Universal Master OPC Server импорт тегов реализован через файл с расширением csv. Длинный файл можно открыть в Excel и заполнить таблицы тегов. Полный список тегов, участвующих в работе OPC – сервера, приведен в приложении В к диссертации. Создать файл импорта тегов, можно выполнив обратную операцию – экспорт тегов. Для этого необходимо создать один тег. При помощи нажатия на кнопку «Добавить тег» во вкладке «Конфигурация». После создания тега необходимо нажать правой клавишей мыши по узлу, и в контекстном меню выбрать строку «Экспорт тегов». После чего откроется окно экспорта тегов, в котором необходимо нажать на кнопку «Выделить», а после «Переместить». После чего тег переместится из области «Теги» в область «Файл CSV». Окно «Экспорт тегов в CSV файл» изображено на рисунке 68. Далее необходимо нажать на кнопку «Готово», расположенную в нижней части окна «Экспорт тегов в CSV файл». После нажатия на кнопку «Готово» откроется окно, в котором необходимо указать имя файла и его расположение на ПК, рисунок 69.

После успешного сохранения CSV файла, его необходимо открыть при помощи программ Excel. Далее необходимо сформировать таблицу тегов, как указано в приложении В и сохранить изменения в файле.

Следующим шагом является импорт тегов в конфигурацию OPC – сервера. Для этого в дереве объектов OPC – сервера, необходимо нажать правой клавишей мыши по устройству «PLC 160M» и выбрать пункт «Импорт тегов». При этом откроется окно «Мастер импорта». В данном окне необходимо указать расположение сформированного CSV файла и нажать на кнопку «Вперед». В следующем окне «Мастер импорта» необходимо нажать на кнопку «Выделить», а после на кнопку «Переместить» и в завершении на кнопку «Готово», рисунок 70.

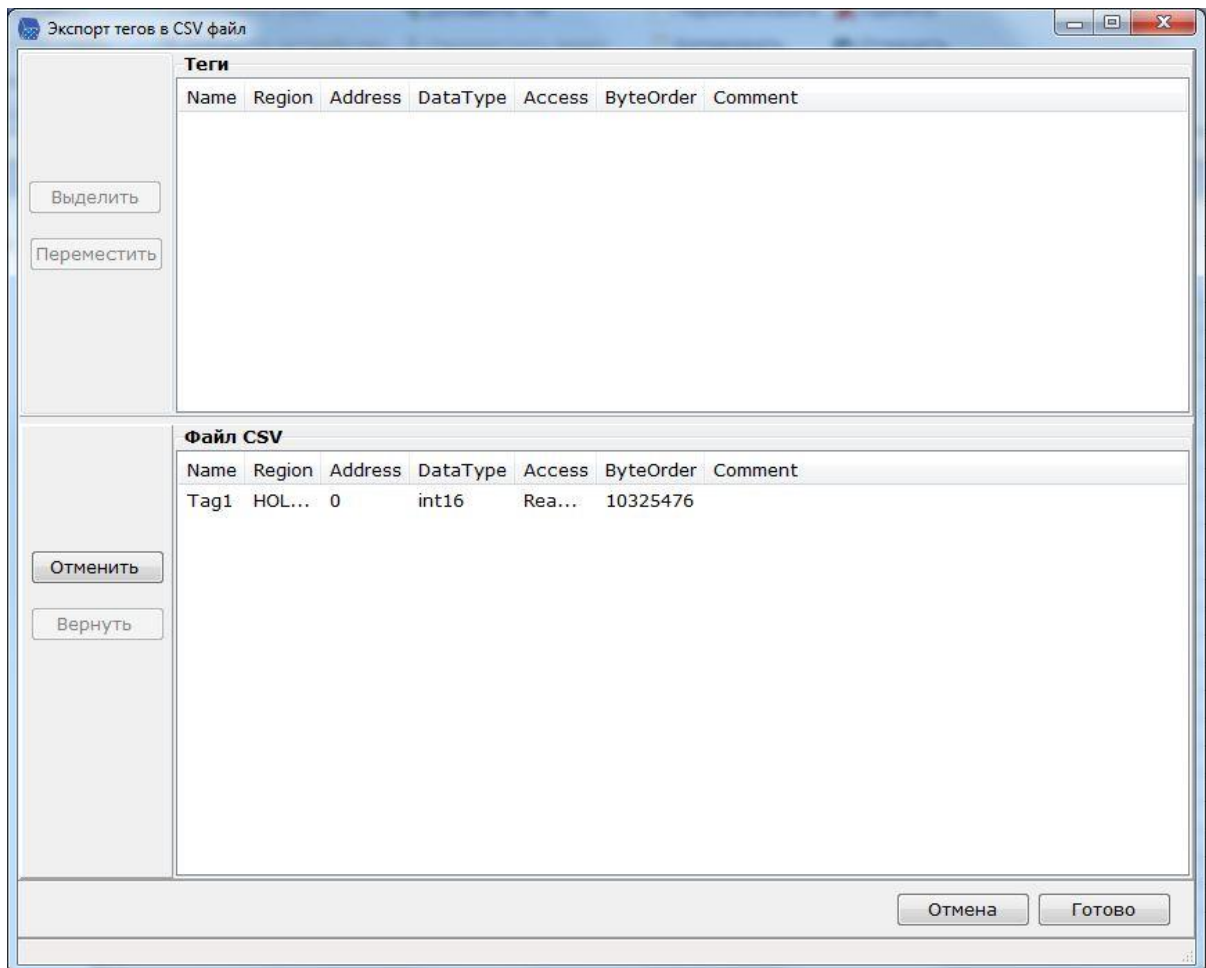


Рисунок 68 – Окно «Экспорт тегов в CSV файл».

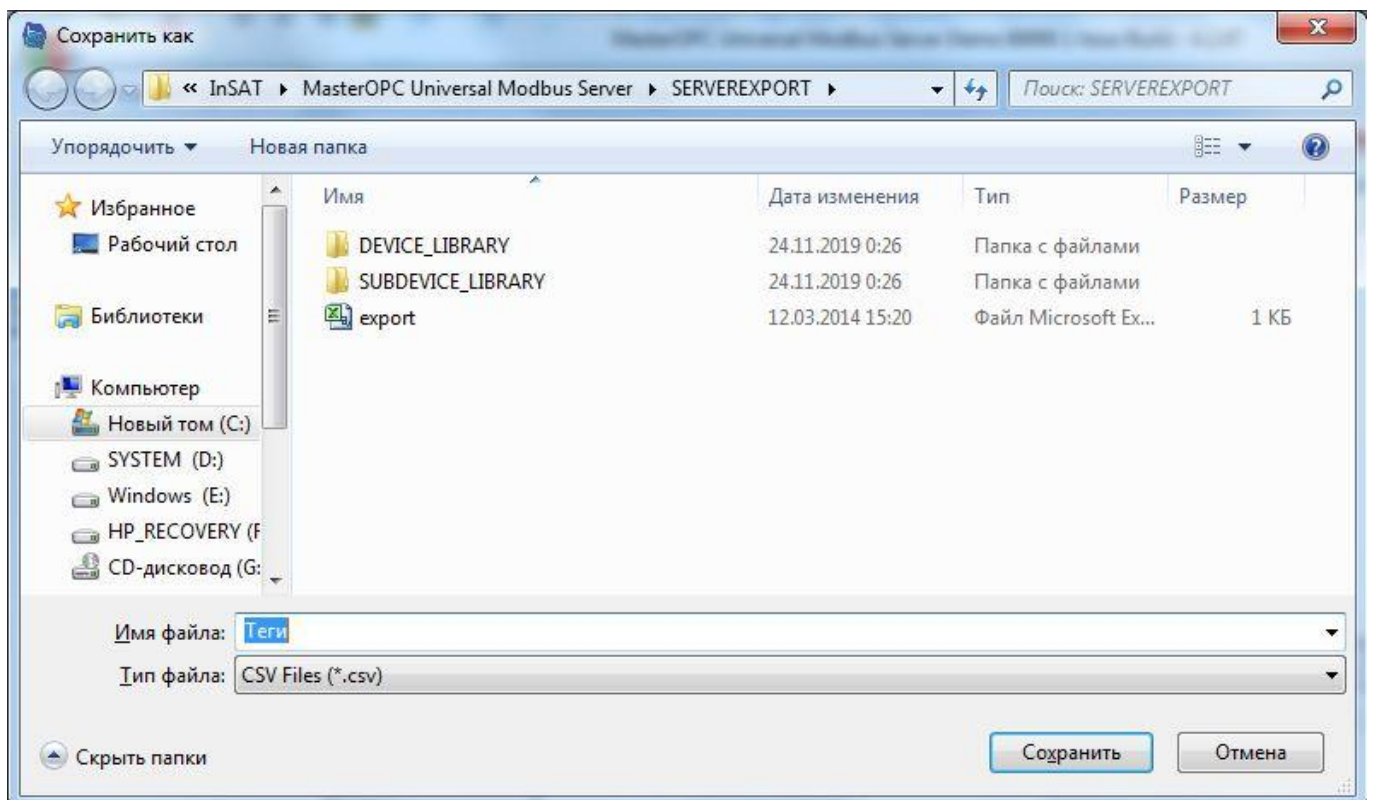


Рисунок 69 – Окно сохранения CSV файла.



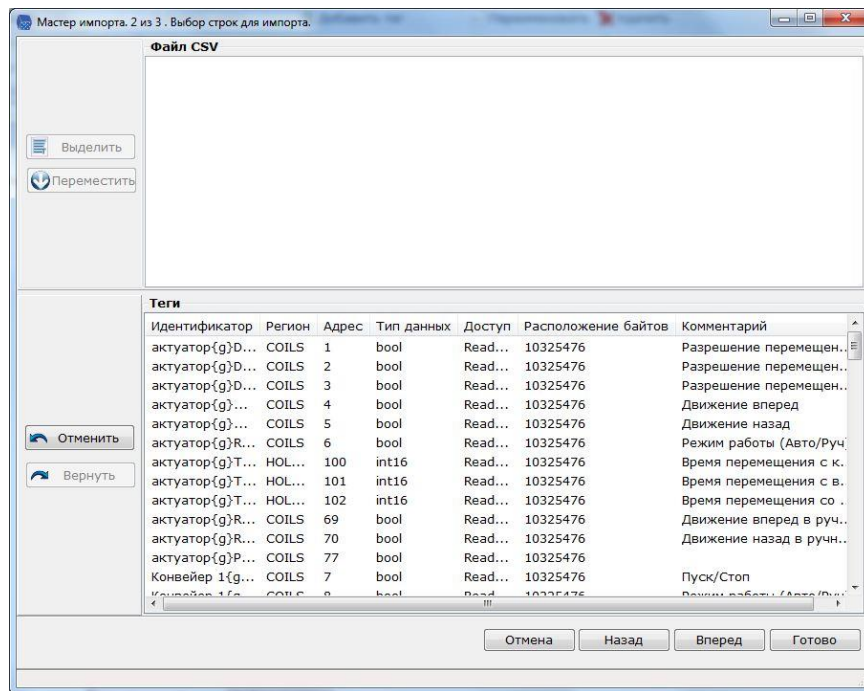


Рисунок 70 – Окно «Мастер импорта».

После нажатия на кнопку «Готово», импорт тегов будет завершен. В главном окне OPC – сервера появятся список тегов, которые разделены по группам, рисунок 71. При этом конфигурация OPC – сервера будет завершена.

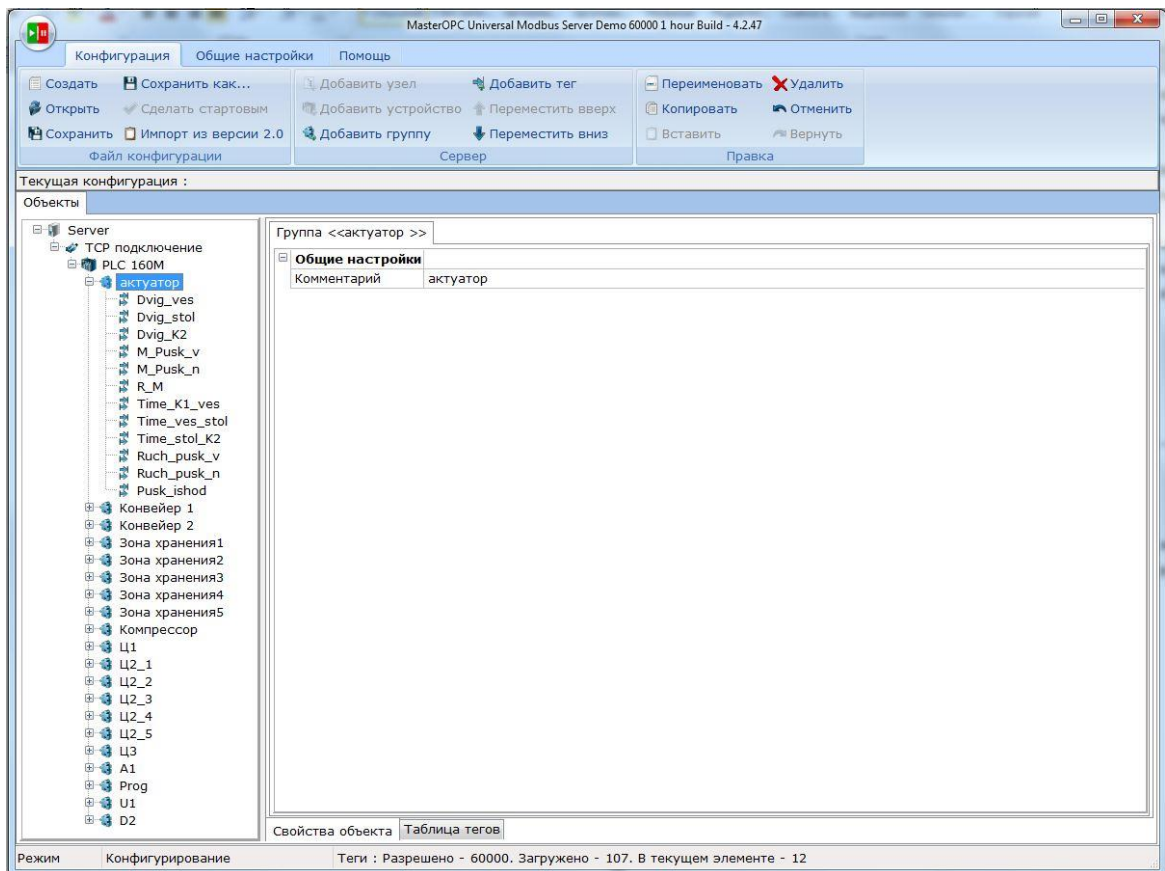


Рисунок 71 – Окно Master OPC Server с добавленными тегам.

## 5.2.2 Master SCADA 4D

В качестве инструмента для написания программы визуализации была выбрана среда разработки Master SCADA 4D, от компании ИнСАТ. Выбор данной среды разработки основывается исходя из ее надежности и популярности. Систему Master SCADA используют в своих проектах крупные компании России, такие как: Газпром, Роснефть, Лукойл, Иркутскэнерго, Калининская АЭС, и многие другие.

Система Master SCADA занимает лидирующую позицию на российском рынке, поскольку является полноценным программным обеспечением, разработанным в России.

Общее количество внедрений Master SCADA составляет тысячи единиц. А география использования распространяется не только на Россию, но и на ближнее и дальнее зарубежье.

Построение программы визуализации в среде разработки Master SCADA 4D основывается на создании программных объектов. Объекты могут содержать в своем составе окна, программы, сообщения и отчеты. Для обмена информацией между объектами проекта используются параметры. Параметры могут передавать информацию любого типа данных. Master SCADA 4D имеет полный набор функций для создания проекта используя только данную платформу, но в данной работе Master SCADA 4D используется только лишь для написания визуализации проекта. Структурная схема программы визуализации, на которой изображены связи объектов, изображена на рисунке 72.

Как видно из структурной схемы, изображенной на рисунке 150, программа состоит из трех основных объектов, «Стартовое окно проекта», «Пневматика» и «АРМ OPC DA». Объекты «Стартовое окно проекта» и «Пневматика» отражают структуру программы визуализации, а объект «АРМ OPC DA» отражает взаимосвязь программы визуализации с OPC – сервером. В системе автоматического управления мехатронным стендом OPC – сервер работает в режиме передачи данных OPC DA.

«Стартовое окно проекта» содержит в своей структуре окно «Главное окно», программу обработки, объекты описывающие электромеханические устройства, датчики, органы управления и отображения системы.

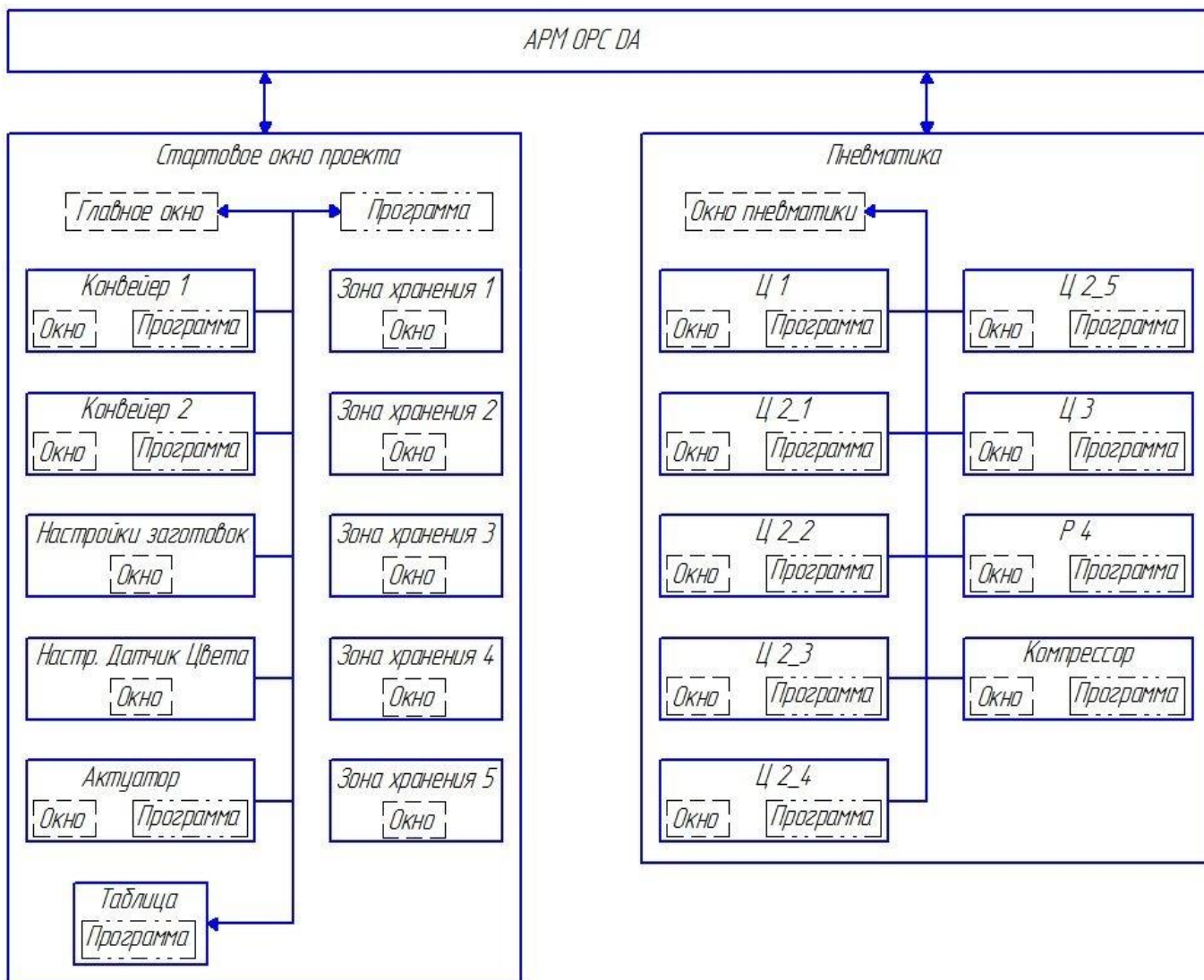


Рисунок 72 – Структурная схема программы визуализации.

Объект «Пневматика» состоит из окна «Окно пневматики», объектов описывающих работу, управление и отображение пневмокомпонентов системы.

Объекты «Стартовое окно проекта» и «Пневматика» получают и обрабатывают значения переменных поступающих от ПЛК, передавая информацию об их состоянии оператору системы автоматического управления мехатронным стандом.

В данном разделе описываются не только программные продукты, используемые в проекте, но и созданные программы управления для

технологического процесса сортировки. В проекте применяется программное обеспечение (ПО) от отечественных производителей. Данное ПО находится в открытом доступе и предоставляется совершенно бесплатно, но с некоторыми ограничениями.

## 6 Интерфейс АРМ

Полное управление мехатронным стандом осуществляется при помощи клиента Master SCADA 4D. Интерфейс стартового окна SCADA системы приведен на рисунке 73.

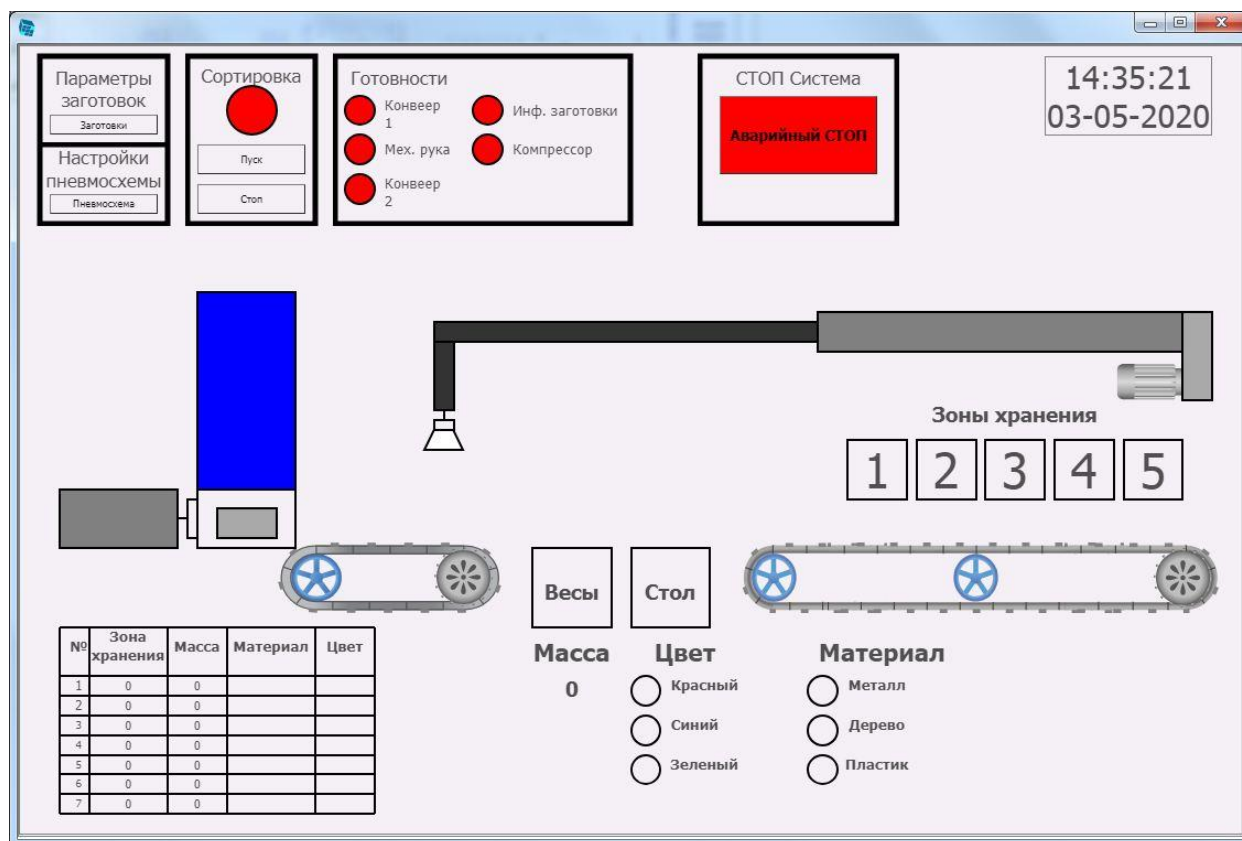


Рисунок 73 – Изображение стартового окна SCADA системы.

Как видно из рисунка 73, стартовое окно отражает в себе полный технологический процесс, органы управления, индикацию состояний, информацию о заготовках и о выполненной сортировке.

Для более удобного восприятия технологического процесса, а так же понимания тонкостей настройки системы, обозначим на рисунке стартового окна SCADA системы области отображения информации и органы управления, рисунок 74.

Область, обозначенная номером 7, отражает технологический процесс сортировки. При запуске автоматической сортировки либо при управлении устройств в ручном режиме, на данной области отражаются их состояния. При

запуске, какого либо устройства, на мнемосхеме устройство окрашивается в зеленый цвет.

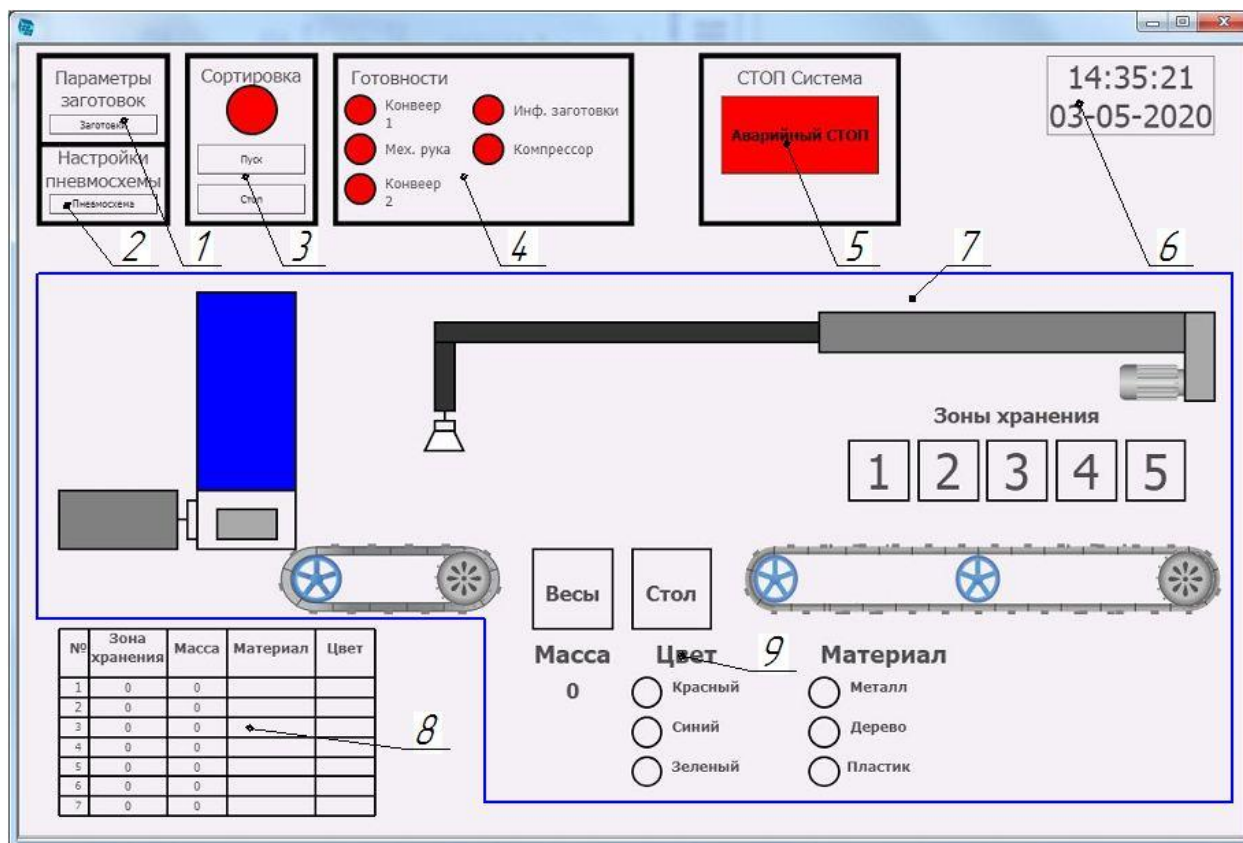


Рисунок 74 – Области стартового окна SCADA системы.

Для начала работы системы, необходимо настроить все параметры и внести необходимые данные. Для более целесообразного использования времени, рекомендуется запустить в работу компрессор. Эта рекомендация следует из конструктивных особенностей компрессора. В мехатронном стенде используется компрессор поршневого типа и во избежание его перегрева необходимо максимально увеличить паузы между его работой.

Для запуска компрессора необходимо нажать на кнопку «Пневмосхема», которая находится в области под номером 1. После откроется окно с мнемосхемой содержащей все пневматические узлы. Пневмосхема изображена на рисунке 75.

Далее необходимо нажать на изображение турбины компрессора. После чего будет открыто всплывающее окно «Компрессор» состоящее из индикатора состояния и кнопок «Пуск», «Стоп». При нажатии на кнопку пуск в контроллер передается сигнал на запуск компрессора и при успешном запуске, в всплывающем

окне «Компрессор» индикатор окрашивается в зеленый цвет и меняет надпись с «Выкл», на «Вкл». Состояния всплывающего окна «Компрессор» приведены на рисунке 76.

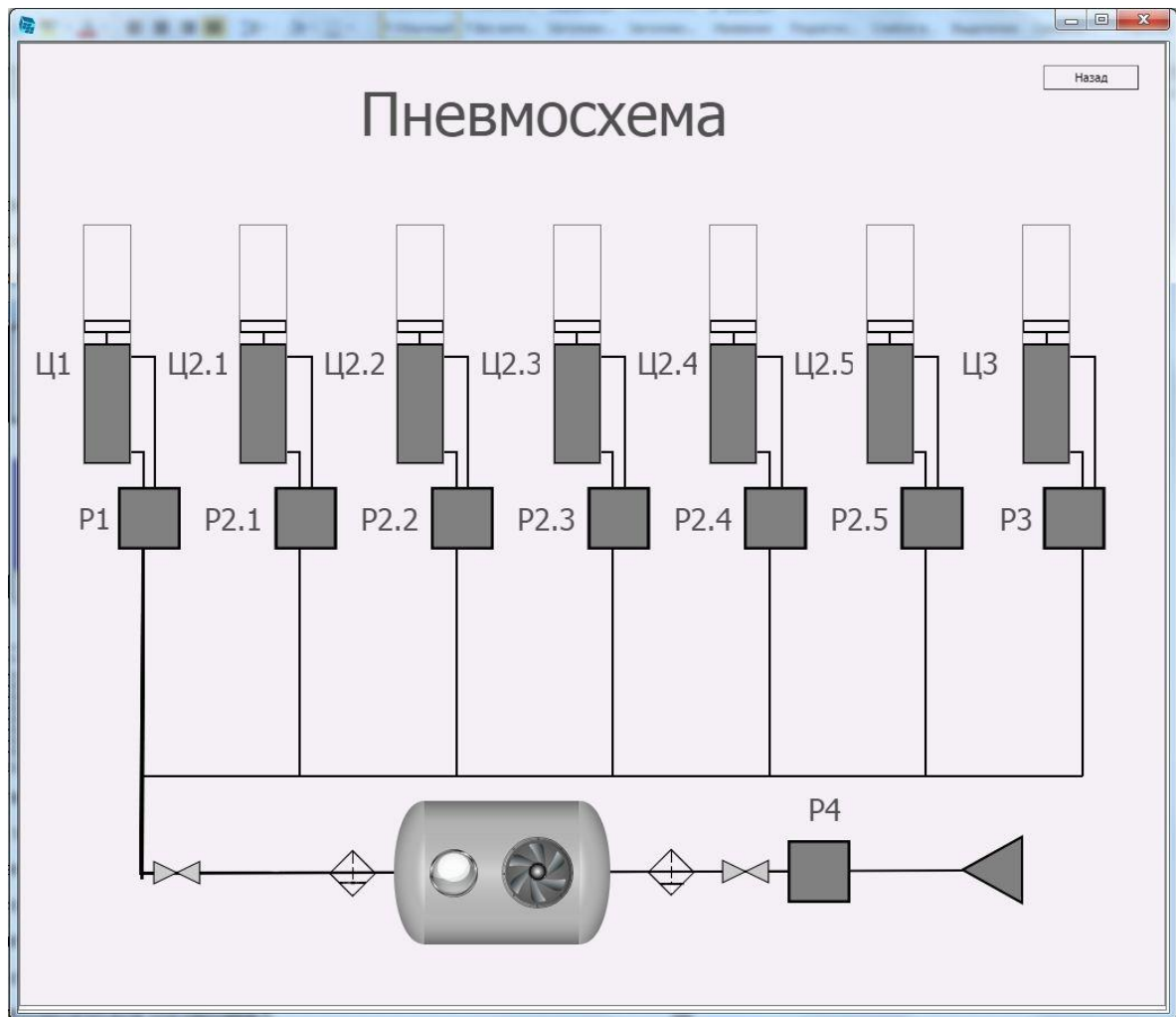


Рисунок 75 – Пневмосхема.

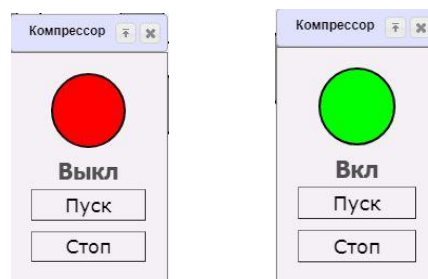


Рисунок 76 – Состояния всплывающего окна «Компрессор».

Состояние компрессора так же отражается в окне «Пневмосхема» и на стартовом окне SCADA системы. На «Пневмосхеме» индикатор находящийся на изображении компрессора окрашивается в зеленый цвет и включается анимация

работы турбины. На стартовом окне располагается индикатор в области номер 4, область готовностей оборудования, который так же окрашивается в зеленый цвет.

После запуска компрессора, необходимо дождаться пока компрессор наберет рабочее давление и перейти к настройке пневмоцилиндров. Для этого необходимо нажать на необходимый пневмоцилиндр, после чего откроется окно его настройки и управления, рисунок 77.



Рисунок 77 – Окно управления «Ц1».

В данном окне необходимо располагается кнопка «Руч», при нажатии на которую пневмоцилиндр переходит в режим ручного управления, а также цифровой ввод времени сработки пневмоцилиндра. Вид окна настройки и управления пневмоцилиндра в режиме ручного управления изображено на рисунке 78.

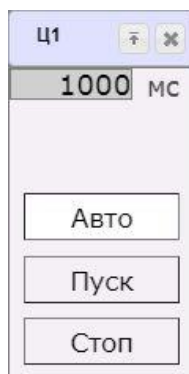


Рисунок 78 – Окно управления «Ц1» в ручном режиме.

При переходе в ручной режим, в окне появляются кнопки «Пуск» и «Стоп». При нажатии на клавишу «Пуск» АРМ отправляет команду на сработку пневмоцилиндра в ПЛК, и пневмоцилиндр будет находиться в таком положении до тех пор, пока оператор не нажмет на кнопку «Стоп». Данная функция позволяет оператору не только управлять в ручном режиме пневмоцилиндром но и измерить



время его фактической сработки, которое необходимо при запуске системы в автоматическом режиме.

По умолчанию все пневмоцилиндры настроены на время сработки в 1000 мс. Это минимальное время сработки, которое не может быть меньше из за конструктивных особенностей пневмоцилиндров применяемых в данном проекте. При попытке ввода значения меньше чем 1000 мс, в окне высветиться соответствующая ошибка и если оператор находится в ручном режиме управления пневмоцилиндром, система заблокирует клавишу пуск, а в ПЛК отправиться команда «Стоп». Изображение окна при неверном вводе значения времени сработки в ручном и автоматическом режиме, приведено на рисунке 79.

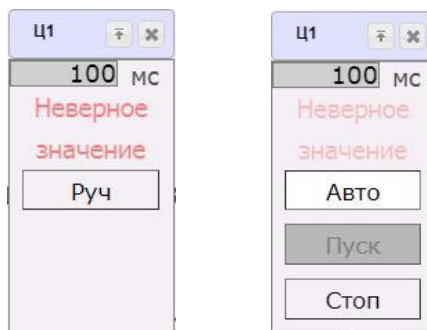


Рисунок 79 – Индикация ошибки.

После успешной настройки пневмоцилиндра, его обязательно необходимо перевести в автоматический режим работы. Аналогичную настройку необходимо произвести абсолютно со всеми пневмоцилиндрами системы.

Помимо пневмоцилиндров в пневмосистему входит присоска. Присоска в данной системе не нуждается в настройке, но может управляться как в автоматическом, так и в ручном режиме работы. Для открытия окна управления присоской необходимо нажать на ее изображение. Изображение окна управления присоской в ручном и автоматическом режиме приведено на рисунке 80.

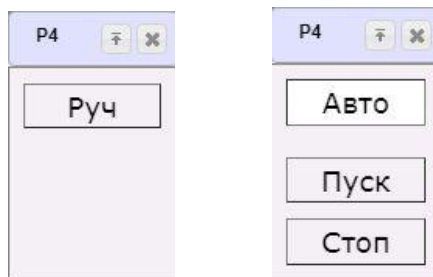


Рисунок 80 – Окно управления присоской.

Для перевода в ручной режим управления присоской, оператор должен нажать на кнопку «Руч», после чего в окне появятся кнопки «Пуск» и «Стоп». При нажатии на кнопку «Пуск» присоска включится, а при нажатии на кнопку «Стоп», соответственно выключится.

Следующим шагом настройки системы является настройка заготовок, для этого необходимо перейти на стартовое окно SCADA системы путем нажатия на кнопку «Назад» в окне «Пневмосхема».

При выполнении автоматической сортировки, системе необходимо задать параметры заготовок: размер, плотность. Для задания данных параметров на стартовом окне SCADA системы необходимо нажать на кнопку «Заготовки» в области 1, рисунок 74. Откроется окно «Настройки заготовок», рисунок 81.

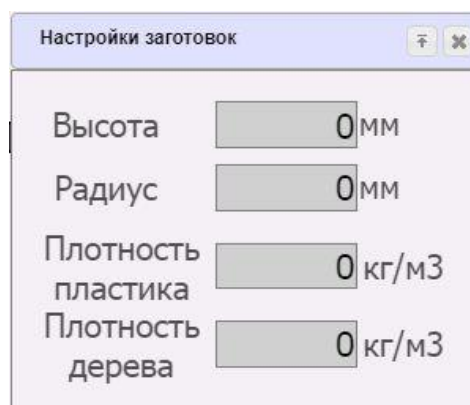


Рисунок 81 – Окно «Настройки заготовок».

В данном окне необходимо указать габаритные характеристики заготовок «Высоту» и «Радиус», а так же «Плотность пластика» и «Плотность дерева». Данные параметры необходимы системе автоматического управления мехатронным стандом, поскольку определение материала заготовок происходит расчетным способом.

Далее необходимо произвести настройку датчика цвета. Для этого необходимо поместить поочередно заготовки различных цветов на «Стол», перед датчиком цвета.

В стартовом окне SCADA системы, для открытия окна, необходимо нажать на надпись «Цвет» (рисунок 74, выноска 9), при этом откроется окно

предназначенное для калибровки датчика цвета. Окно «Настройки датчика цвета» приведено на рисунке 82.



Рисунок 82 – Окно «Настройки датчика цвета».

Необходимо поместить поочередно заготовки разного цвета и зафиксировать измеренные значения датчика, при этом системе необходимо указать соответствующий порог срабатки в указанном окне.

Далее необходимо настроить конвейеры системы управления. Для этого необходимо нажать на изображение двигателя первого конвейера на стартовом окне SCADA системы, рисунок 74. При этом откроется окно настройки конвейера 1. Изображение окна «Конвейер 1» приведено на рисунке 83.

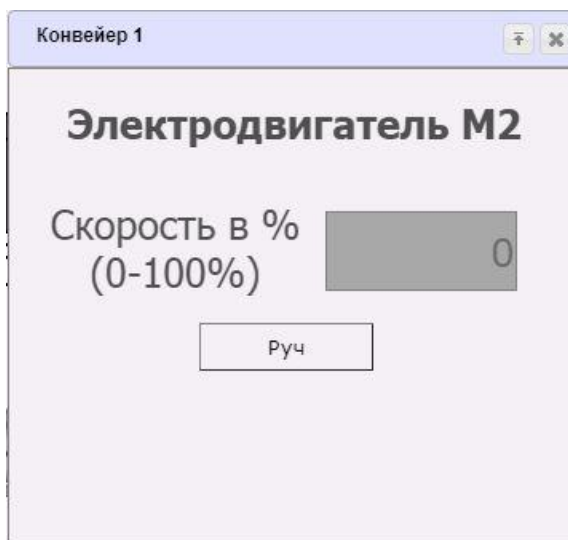


Рисунок 83 – Окно «Конвейер 1».

Далее необходимо перевести конвейер 1 в режим ручного управления, путем нажатия на кнопку «Руч». Изображение окна «Конвейер 1» при ручном режиме управления приведено на рисунке 84.

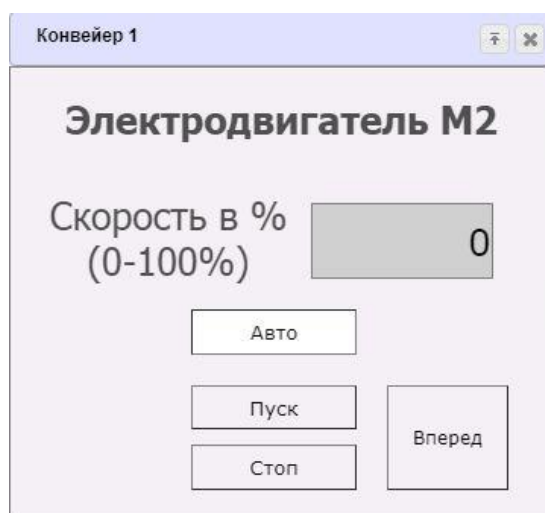


Рисунок 84 – Окно «Конвейер 1» в режиме ручного управления.

В данном окне необходимо задать «Скорость» вращения электродвигателя в процентах, а также настроить направление вращения. Кнопка «Пуск» запускает электродвигатель, кнопка «Стоп» останавливает. Для изменения направления вращения электродвигателя, необходимо нажать на кнопку «Вперед», при этом электродвигатель запустится в обратном направлении, а данная кнопка изменит свое название - «Назад».

После успешной настройки первого конвейера, аналогично необходимо настроить и второй.

Далее необходимо настроить устройство «Мех. рука». Для этого необходимо нажать на изображение линейного – актуатора, при этом откроется окно настройки актуатора, рисунок 85.



Рисунок 85 – Окно «Актуатор».

В данном окне, системе управления необходимо указать временные уставки перемещения между технологическими зонами. Временная уставка определяют время работы двигателя М4, т.е. работу актуатора. Уставка «Т1-Т2» определяет время перемещения с технологической зоны «Конвейера 1» на технологическую зону «Весов». Уставка «Т2-Т3» определяет время перемещения с технологической зоны «Весов» на технологическую зону «Стол». Уставка «Т3-Т4» определяет время перемещения с технологической зоны «Стол» на технологическую зону «Конвейер 2».

Для определения вышеуказанных временных уставок, необходимо перевести актуатор в ручной режим (нажать на кнопку «Руч») и удерживая кнопку «Вперед» засечь время перемещения. Если появляется необходимость вернуть в какое либо положение необходимо нажать на кнопку «Назад». Окно управления актуатором в ручном режиме приведено на рисунке 86.



Рисунок 86 – Окно «Актуатор» в ручном режиме.

При задании временных уставок необходимо учитывать конструктивные особенности актуатора, таким образом суммарное время перемещения между точками не может превышать восьми секунд. Если оператор введет значение суммарно больше чем восемь секунд, то система выдаст ошибку, «Неверное значение».

После успешной настройки, в окне «Актуатор» необходимо нажать на кнопку «Авто», тем самым выбрав автоматический режим. После перехода в

автоматический режим станет доступна кнопка «Исходное положение линейного актуатора». Для запуска системы в автоматическом режиме необходимо выставить актуатор в исходное положение. Оператор нажимая на данную кнопку дает команду системе и актуатор возвращается в исходное положение.

После выполнения всех вышеуказанных этапов настройки система будет сигнализировать о готовности работы. На стартовом окне SCADA системы (рисунок 151, зона номер 4), располагаются соответствующие индикаторы и при успешной настройке они окрасятся в зеленый цвет как показано на рисунке 87.



Рисунок 87 – Индикация готовности системы.

Завершающим этапом настройки системы является указание признаков по которым будет произведена сортировка заготовок. Для этого необходимо нажать на необходимую зону хранения и указать в ней соответствующие признаки. Окно «Зона хранения 1» приведено на рисунке 88.

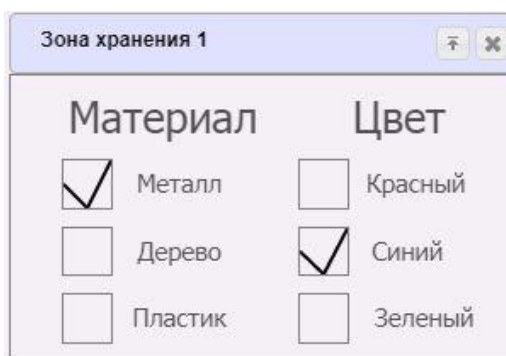


Рисунок 88 – Окно «Зона хранения 1».

При выборе, как указано на рисунке 88, система автоматической сортировки поместит в первую зону хранения заготовки синего цвета с наличием металла.

Если параметры заготовки не будут совпадать с указанными, система автоматического управления мехатронным стендом, поместит заготовку в следующую зону хранения, в которой не указаны параметры заготовки. При

настройке необходимо учитывать данную особенность, если параметры заготовки не будут совпадать с указанными, заготовка остановит процесс сортировки. Оператор при настройке системы управления должен избегать подобную ситуацию, при которой нарушается цикл автоматической сортировки.

Если в процессе автоматической сортировки, либо при работе системы в ручном режиме, произойдет нештатная ситуация, оператор немедленно должен нажать на кнопку «Аварийный СТОП» (рисунок 74, зона 5). При этом система немедленно отреагирует на данную команду, остановив все процессы работы системы.

Система управления мехатронным стендом оснащена упрощенной базой данных. На стартовом окне SCADA системы, в нижнем левом углу расположена таблица в которую заносится информация о параметрах заготовок, рисунок 74, зона 8.

Данная таблица отражает параметры заготовок в порядке их подачи в систему сортировки. Т.е. параметры первой поданной заготовки будут отражены в первой строке. Таблица параметров приведена на рисунке 89.

№	Зона хранения	Масса	Материал	Цвет
1	0	0		
2	0	0		
3	0	0		
4	0	0		
5	0	0		
6	0	0		
7	0	0		

Рисунок 89 – Таблица параметров.

В таблицу заносится номер зоны хранения, масса заготовки, материал и цвет. Таким образом, система формирует для оператора отчет о параметрах первых семи заготовках.

В данном разделе описан интерфейс управления технологическим процессом сортировки. Разработанный интерфейс в полной мере соответствует технологическому процессу, а так же дает возможность оператору производить тонкие настройки технологического оборудования.

## Заключение

В ходе выполнения выпускной квалификационной работы был проведен анализ ключевых особенностей имеющегося на рынке учебного оборудования в области мехатроники. Были определены их основные преимущества и недостатки. Результат анализа показал, что на рынке отсутствуют отечественные производители таких учебных систем. На основании проведенного анализа было принято решение о разработке универсального учебного мехатронного стенда, который обладал бы гибкой системой автоматического управления построенной на современном оборудовании. При выборе оборудования системы учитывались такие факторы как надежность, универсальность и цена. Построенная система управления учебным мехатронным стендом соответствует классической архитектуре представления систем АСУ ТП. Результатом использования вышеуказанных факторов, а так же указанной архитектуры, была достигнута поставленная цель и выполнены все задачи. Как следствие спроектированная система превосходит свои зарубежные аналоги по функциональности. Созданная система управления учебного мехатронного стенда в полной мере несет дидактическую ценность, имеет возможность использования как совместно с ПЛК, так и совместно с микроконтроллерными системами управления, при этом цена данного учебного стенда гораздо ниже своих зарубежных аналогов.

Включение в учебный процесс учебного мехатронного стенда будет являться уверенным шагом в развитии университета. Данный стенд будет служить наглядной учебной единицей изучив которую, студенты смогут повысить свой уровень знаний в области мехатроники, наработать ключевые навыки работы с представленным оборудованием, реализовать на основе данного оборудования различные системы автоматического управления технологических процессов.



## Список используемой литературы и список используемых источников

1. Development of a Poppet-Type Pneumatic Servo Valve/ Takahiro Kanno, Takashi Hasegawa, Tetsuro Miyazaki, Nobuyuki Yamamoto, Daisuke Haraguchi, Kenji Kawashima// Applied Sciences – 2018 – 8 – 2094 URL: <https://www.mdpi.com/2076-3417/8/11/2094> (дата обращения: 15.10.2018).
2. Hiroki Tomori /Control of Pneumatic Artificial Muscles Using Local Cyclic Inputs and Genetic Algorithm// Hiroki Tomori, Kenta Hiyoshi// Actuators – 2018 – 7 – 36 URL: <https://www.mdpi.com/2076-0825/7/3/36> (дата обращения: 16.10.2018).
3. Georgia P. Smyrniou. A Comparative Study of Control Methods for a Robotic Manipulator with Six DOF in Simulation/ Georgia P. Smyrniou, Michail Papoutsidakis Dimitrios Tseles //MATEC Web of Conferences – 2017 – 125 – 6 URL: [https://www.matec-conferences.org/articles/mateconf/abs/2017/39/mateconf\\_csc2017\\_04004/mateconf\\_csc2017\\_04004.html](https://www.matec-conferences.org/articles/mateconf/abs/2017/39/mateconf_csc2017_04004/mateconf_csc2017_04004.html) (дата обращения: 17.10.2018).
4. Samer Jaloudi, Communication Protocols of an Industrial Internet of Things Environment: A Comparative Study/ Samer Jaloudi / Future Internet – 2019 – 11(3) – 66 URL: <https://www.mdpi.com/1999-5903/11/3/66> (дата обращения: 18.10.2018).
5. Shahin Mahdiyoun Rad, Sensorless Commutation Method for Low-Voltage BLDC Motors Based on Unfiltered Line Voltage/ Shahin Mahdiyoun Rad, Mohammad Reza Azizian, Mohammad Hejri // POWER ENGINEERING AND ELECTRICAL ENGINEERING – 2019 – 17 – 1 URL: <http://advances.utc.sk/index.php/AEEE/article/view/3021> (дата обращения: 19.10.2018).
6. Migration of a SCADA system to IaaS clouds – a case study / Philip Church and etc. // Journal of Cloud Computing: Advances, Systems and Applications – 2017 – 6 – 11 URL: <https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-017-0080-5> (дата обращения: 23.10.2018).
7. Passive Wireless LC Proximity Sensor Based on LTCC Technology / Mingsheng Ma and etc. // Sensors – 2019 – 19 – 1110 URL: <https://www.mdpi.com/1424-8220/19/5/1110> (дата обращения: 24.10.2018).

8. Young Soo Suh, Laser Sensors for Displacement, Distance and Position / Young Soo Suh // Sensors – 19 – 1924 URL: <https://www.mdpi.com/1424-8220/19/8/1924> (дата обращения: 25.10.2018).

9. Podržaj Primož, PID Controller Implemented in Festo CDPX / Podržaj Primož // MATEC Web of Conferences – 2019 – 260 – 6 URL: [https://www.matec-conferences.org/articles/matecconf/abs/2019/09/matecconf\\_icpse2018\\_02008/matecconf\\_icpse2018\\_02008.html](https://www.matec-conferences.org/articles/matecconf/abs/2019/09/matecconf_icpse2018_02008/matecconf_icpse2018_02008.html) (дата обращения: 26.10.2018).

10. Color sensors and their applications based on real-time color image segmentation for cyber physical systems / Neal N. Xiong and etc.// EURASIP Journal on Image and Video Processing – 2018 – 23 – 1 URL: <https://link.springer.com/article/10.1186/s13640-018-0258-x> (дата обращения: 27.10.2018).

11. Густав Олсон, Джангуидо Пиани Цифровые системы автоматизации и управления. – СПб.: Невский Диалект, 2001. – 557 с.

12. Влах, И. Машинные методы анализа и проектирования электронных схем / И. Влах, К. Сингхал; пер. с англ. – М.: Радио и связь, 1988.– 560 с.

13. Черношвитов, А. Visual C++7 : учебный курс / А. Черношвитов. – СПб. : Питер, 2002. – 528 с.

14. Павловская, Т.А. С/С++. Программирование на языке высокого уровня : учеб. для вузов / Т.А. Павловская. – СПб. : Питер, 2007. – 460 с.

15. Александров А.А. Электротехнические чертежи и схемы / Александров К.К., Кузьмина Е.Г.- М.:Энергоатомиздат, 1990. - 288с.

16. SCADA-системы. Взгляд изнутри. – РТСофт: О. Синенко, Н. Куцевич, Е. Андреев, 2004г. – 176с.

17. А. В. Герасимов, А. Титовцев. Проектирование АСУТП с использованием SCADA-систем. - М.,:Бином, 2014г. – 128с.

18. Лопатин А.Г., Киреев П.А. Методика разработки систем управления на базе SCADA – Новомосковск, системы Trace Mode 2007г. –112 с.

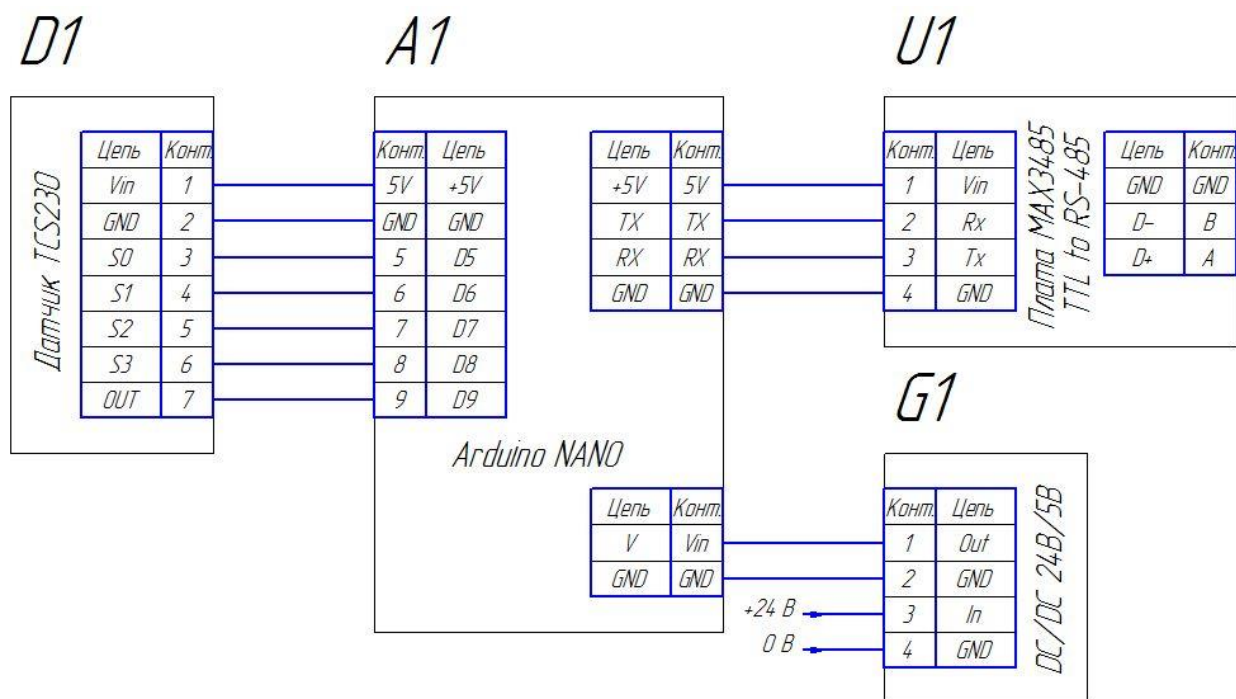
19. Основы автоматизированных систем управления технологическими процессами. Р.Х. Юсупов: Инфра-Инженерия 2018г - 132с.

20. Разработка АСУТП в среде WinCC. Иванов В.Э., Чье Ен Ун: Инфра-Инженерия: 2019г. – 232с.
21. Электронные промышленные устройства: Учеб. для студ. вузов спец. «Пром. электрон.»/ В.И. Васильев, Ю.М. Гусев, В.Н. Миронов и др. – М.: Высш. шк., 1988. – 303 с.
22. Забродин Ю.С. Промышленная электроника: Учебник для вузов. – М.: Высш. школа, 1982. – 496 с., ил.
23. Хьюз, К. Параллельное и распределенное программирование с использованием C++ = Parallel and Distributed Programming Using C++ / К. Хьюз, Т. Хьюз ; пер. с англ. и ред. Н.М. Ручко. – М. : СПб. : Киев : Вильямс, 2004. – 667 с.
24. Кудинов А.К. Методы анализа и расчет электронных схем на ЭВМ: Учебное пособие / – Тольятти : Изд-во ТГУ, 2007
25. Глибин Е.С. Программирование электронных устройств : электронное учеб. пособие / Е.С. Глибин, А.В. Прядилов. – Тольятти : Изд-во ТГУ, 2014.: 1 оптический диск
26. Электроника и микропроцессорная техника: учебник / В.Г. Гусев, Ю.М. Гусев. – 6-е изд., стер. – М.: КНОРУС, 2013. – 800 с.
27. Пьявченко Т. А. Автоматизированные системы управления технологическими процессами и техническими объектами: Учебное пособие. Таганрог: изд-во ТРТУ. 1997. – 128 с.
28. Андреев Е.Б. SCADA – системы: взгляд изнутри./Е.Б. Андреев, Н.А. Куцевич, О.В. Синенко – Москва: «РТСофт», 2004 – 176с.
29. Лопатин А.Г. Методика разработки систем управления на базе SCADA системы Trace Mode./ Лопатин А.Г., Киреев П.А. – Новомосковск: РХТУ им. Д. И. Менделеева, Новомосковск институт Новомосковск, 2007 – 111с.
30. М.В.Позднов, Основы преобразовательной техники: Методические указания по проведению курсового проектирования/ М.В.Позднов. - Тольятти: ТГУ, 2017.– 32 с. – 50 экз.

## Приложение А

### Датчик цвета TCS 230

Принципиальная схема датчика цвета TCS 230.



Перечень элементов датчика цвета.

№	Поз. обозначение	Наименование	Кол.	Примечание
1	A1	Arduino NANO	1	ATmega328p
2	D1	Датчик цвета TCS230	1	TCS 230
3	U1	Преобразователь TTL - RS485	1	MAX3485
4	G1	LM317 DC Linear Regulator	1	LM317

Таблица регистров управления.

№	№ Регистра	Тип данных	Комментарий	Имя переменной в codesys
1	0	WORD	Красный	Mod_red
2	1	WORD	Зеленый	Mod_green
3	2	WORD	Синий	Mod_blue

## Продолжение Приложения А

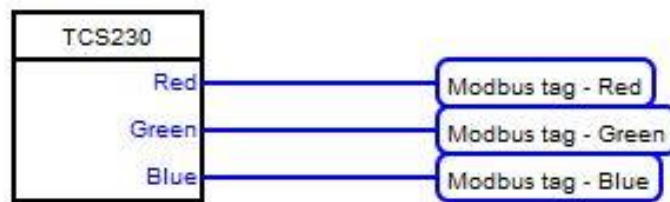
### Программа управления Arduino IDE.

```
#include <ModbusRtu.h>
struct _valuesTCS230
{
int red = 0;
int green = 0;
int blue = 0;
};
_valuesTCS230 _valueTCS230;
Modbus _modbusSlave(1, 0, 3);
uint16_t _ModbusReg[3];
int _tcs2320_1_RedOut = 0;
int _tcs2320_1_greenOut = 0;
int _tcs2320_1_BlueOut = 0;
void setup()
{
  _modbusSlave.begin(115200);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, INPUT);
  digitalWrite(5, 1);
  digitalWrite(6, 0);}
void loop()
{
  _modbusSlave.poll(_ModbusReg, 3);
  _readColorTCS230(5, 6, 7, 8, 9, 0, 0, 0);
  _tcs2320_1_RedOut = constrain(_valueTCS230.red,0,255);
  _tcs2320_1_greenOut = constrain(_valueTCS230.green,0,255);
  _tcs2320_1_BlueOut = constrain(_valueTCS230.blue,0,255);
  _ModbusReg[0] = _tcs2320_1_RedOut;
  _ModbusReg[1] = _tcs2320_1_greenOut;
  _ModbusReg[2] = _tcs2320_1_BlueOut;}
void _readColorTCS230(int s0, int s1, int s2, int s3, int out, int redCal, int greenCal, int
blueCal)
{
  int red = 0;
  int green = 0;
  int blue = 0;
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  _valueTCS230.red =(map(red,255,0, 0,255)) + redCal;
  digitalWrite(s3, HIGH);
```

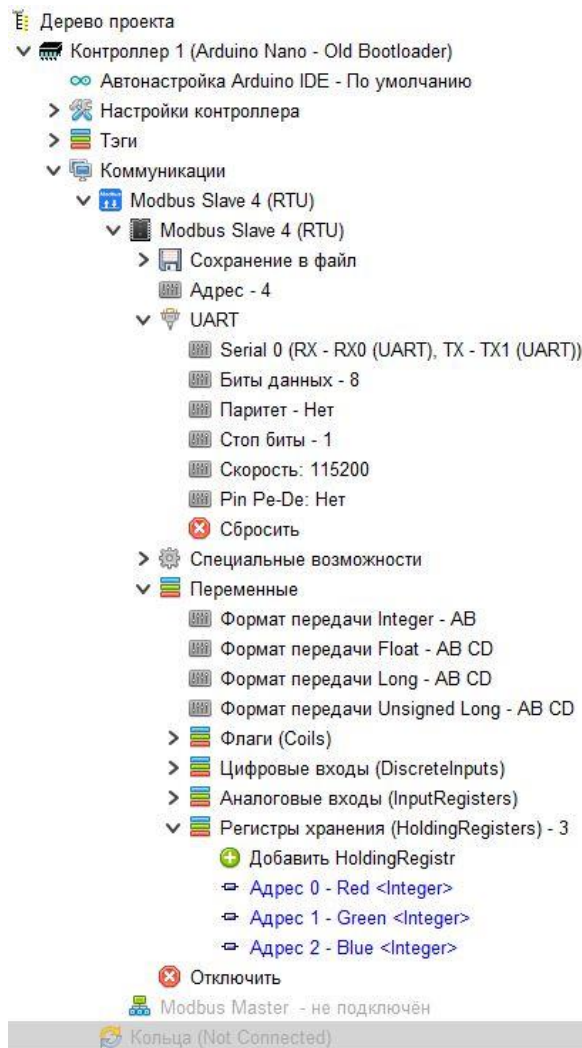
## Продолжение Приложения А

```
blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);  
_valueTCS230.blue = (map(blue,255,0, 0,255))+blueCal;  
digitalWrite(s2, HIGH);  
green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);  
_valueTCS230.green = (map(green,255,0, 0,255))+greenCal; }
```

## Программа управления FLProg.



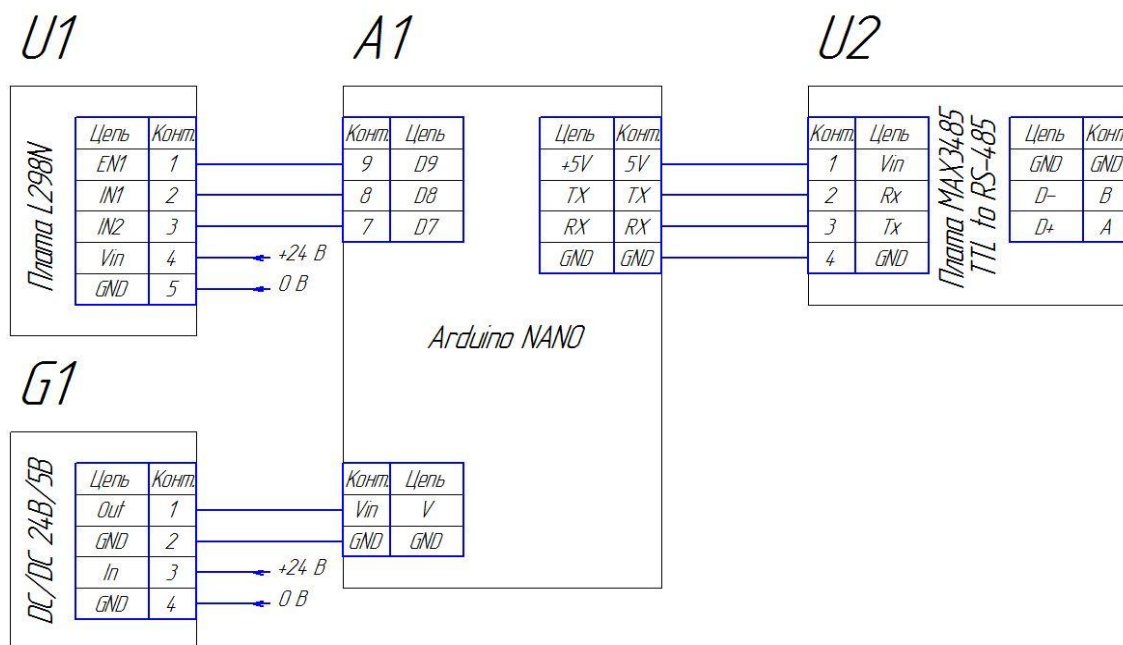
## Дерево проекта FLProg.



## Приложение Б

### Драйвер управления электродвигателем

Принципиальная схема драйвера управления двигателем постоянного тока.



Перечень элементов драйвера.

№	Поз. обозначение	Наименование	Кол.	Примечание
1	A1	Arduino NANO	1	ATmega328p
2	U1	L298N driver board module	1	L298N
3	U2	Преобразователь TTL - RS485	1	MAX3485
4	G1	LM317 DC Linear Regulator	1	LM317

Таблица регистров управления.

№	№ Регистра	Тип данных	Комментарий	Имя переменной в codesys
1	0.0	BYTE	Пуск	Mod_Pusc_M2
2	0.1	BYTE	Вперед	Mod_Back_M2
3	0.2	BYTE	Назад	Mod_Forward_M2
4	0.3	BYTE	Сброс аварии	Mod_Reset_M2
5	1	WORD	Скорость	Mod_Speed_M2
6	2	WORD	Код аварии	Mod_NumAlarm_M2

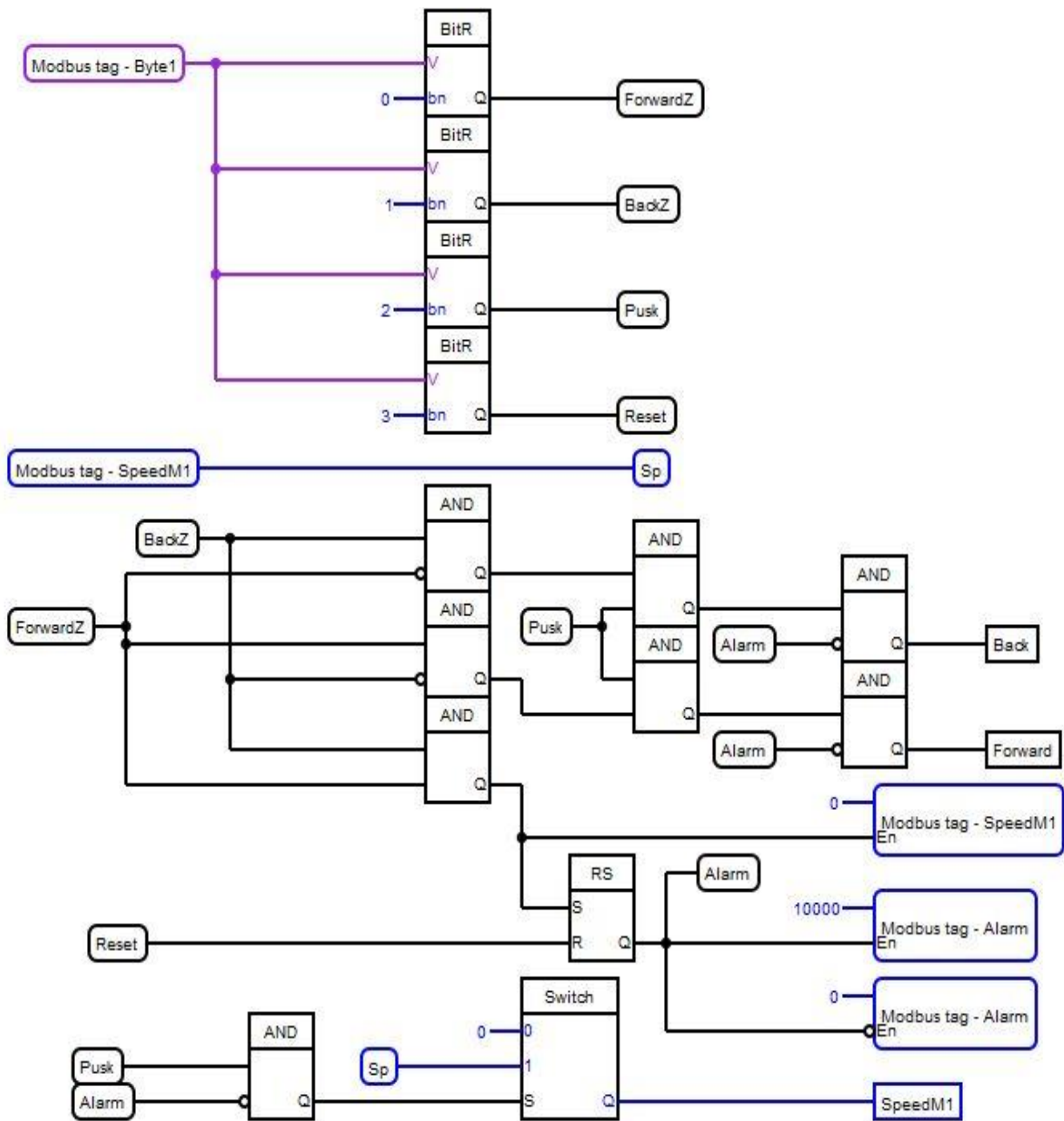
## Продолжение Приложения Б

### Программа управления Arduino IDE.

```
#include <ModbusRtu.h>
Modbus _modbusSlave(3, 0, 0);
uint16_t _ModbusReg[3];
bool _gtv1 = 0;
bool _gtv2 = 0;
bool _gtv3 = 0;
int _gtv4 = 0;
bool _gtv5 = 0;
bool _gtv6 = 0;
int _swi1;
bool _trgs1 = 0;
void setup()
{
  _modbusSlave.begin(115200);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);}
void loop()
{
  _modbusSlave.poll(_ModbusReg, 3);
  if(( (_gtv2) && (_gtv1) )) _trgs1 = 1;
  if(_gtv5) _trgs1 = 0;
  if(( (_gtv3) && !( _gtv6) ))
  {
    _swi1=_gtv4;}
  else
  {
    _swi1=0;}
  _gtv1 = (bitRead(((byte(_ModbusReg[0]))), (0)));
  _gtv2 = (bitRead(((byte(_ModbusReg[0]))), (1)));
  _gtv3 = (bitRead(((byte(_ModbusReg[0]))), (2)));
  _gtv5 = (bitRead(((byte(_ModbusReg[0]))), (3)));
  _gtv4 = (int(_ModbusReg[1]));
  _gtv6 = _trgs1;
  if (( (_gtv2) && (_gtv1) )){
    _ModbusReg[1] = 0;}
  digitalWrite(8, ( (( (_gtv2) && !( _gtv1) )) && (_gtv3) )) && !( _gtv6) ));
  if (_trgs1){
    _ModbusReg[2] = 10000;}
  digitalWrite(7, ( (( (_gtv3) && (( (_gtv1) && !( _gtv2) )) )) && !( _gtv6) ));
  if (!( _trgs1)){
    _ModbusReg[2] = 0;
  }
  analogWrite(9, _swi1);
}
```



Продолжение Приложения Б  
 Программа управления FLProg.



## Продолжение Приложения Б

### Дерево проекта FLProg.

- Дерево проекта
  - Контроллер 1 (Arduino Uno)
    - Автонастройка Arduino IDE - По умолчанию
    - Настройки контроллера
    - Тэги
  - Коммуникации
    - Modbus Slave 2 (RTU)
      - Modbus Slave 2 (RTU)
        - Сохранение в файл
        - Адрес - 2
      - UART
        - Serial 0
        - Биты данных - 8
        - Паритет - Нет
        - Стоп биты - 1
        - Скорость: 115200
        - Pin Pe-De: RX<-0 (UART)
        - Сбросить
      - Специальные возможности
      - Переменные
        - Формат передачи Integer - AB
        - Формат передачи Float - AB CD
        - Формат передачи Long - AB CD
        - Формат передачи Unsigned Long - AB CD
        - Флаги (Coils)
        - Цифровые входы (DiscreteInputs)
        - Аналоговые входы (InputRegisters)
        - Регистры хранения (HoldingRegisters) - 3
          - Добавить HoldingRegistr
            - Адрес 0 - Byte1 <Byte> //Для управления M1
            - Адрес 1 - SpeedM1 <Integer>
            - Адрес 2 - Alarm <Integer>
      - Отключить
    - Modbus Master - не подключён
    - Кольца (Not Connected)

Приложение В  
**Переменные OPC – сервера**

Таблица – Переменные OPC – сервера.

Name	Region	Address	Data Type	Access	ByteOrder	Comment
актуатор {g} Dvig-ves	COILS	1	bool	ReadWrite	10325476	Разрешение перемещения на весы
актуатор {g} Dvig-stol	COILS	2	bool	ReadWrite	10325476	Разрешение перемещения на стол
актуатор {g} Dvig_K2	COILS	3	bool	ReadWrite	10325476	Разрешение перемещения на 2 конвейер
актуатор {g} M_Pusk_v	COILS	4	bool	ReadWrite	10325476	Движение вперед
актуатор {g} M_Pusk_n	COILS	5	bool	ReadWrite	10325476	Движение назад
актуатор {g} R_M	COILS	6	bool	ReadWrite	10325476	Режим работы (Авто/Руч)
актуатор {g} Time_K1-ves	HOLDING_REGISTERS	100	int16	ReadWrite	10325476	Время перемещения с конвейера на весы
актуатор {g} Time-ves-stol	HOLDING_REGISTERS	101	int16	ReadWrite	10325476	Время перемещения с весов на стол
актуатор {g} Time-stol_K2	HOLDING_REGISTERS	102	int16	ReadWrite	10325476	Время перемещения со стола на конвейер
актуатор {g} Ruch_pusk_v	COILS	69	bool	ReadWrite	10325476	Движение вперед в ручном режиме
актуатор {g} Ruch_pusk_n	COILS	70	bool	ReadWrite	10325476	Движение назад в ручном режиме
актуатор {g} Pusk_ishod	COILS	77	bool	ReadWrite	10325476	
Конвейер 1 {g} M_Upr_conveer_1	COILS	7	bool	ReadWrite	10325476	Пуск/Стоп
Конвейер 1 {g} R_M_1	COILS	8	bool	ReadWrite	10325476	Режим работы (Авто/Руч)
Конвейер 1 {g} Speed_driver 1	HOLDING_REGISTERS	103	int16	ReadWrite	10325476	Скорость работы двигателя
Конвейер 1 {g} Ruch_Upr_conveer_1	COILS	7	bool	ReadWrite	10325476	Пуск/Стоп в ручном режиме

## Продолжение Приложения В

Продолжение таблицы - Переменные OPC – сервера.

Конвейер 1{g}Upr_con v1_plc	COILS	71	bool	ReadWrite	10325476	Пуск/Стоп плк
Конвейер 1{g}Alarm_d river1	COILS	72	bool	ReadWrite	10325476	Ошибка связи
Конвейер 1{g}Reset_dr iver1	COILS	73	bool	ReadWrite	10325476	Сброс ошибки
Конвейер 2{g}M_Upr_ conveer_2	COILS	9	bool	ReadWrite	10325476	Пуск/Стоп
Конвейер 2{g}R_M_2	COILS	10	bool	ReadWrite	10325476	Режим работы (Авто/Руч)
Конвейер 2{g}Speed_d river 2	HOLDING_ REGISTERS	104	int16	ReadWrite	10325476	Скорость работы двигателя
Конвейер 2{g}Ruch_U pr_conveer_2	COILS	9	bool	ReadWrite	10325476	Пуск/Стоп
Конвейер 2{g}Upr_con v2_plc	COILS	74	bool	ReadWrite	10325476	Пуск/Стоп плк
Конвейер 2{g}Alarm_d river2	COILS	75	bool	ReadWrite	10325476	Ошибка связи
Конвейер 2{g}Reset_dr iver2	COILS	76	bool	ReadWrite	10325476	Сброс ошибки
Зона хранения1 {g }Metal_Z1	COILS	11	bool	ReadWrite	10325476	Металл
Зона хранения1 {g }Wood_Z1	COILS	12	bool	ReadWrite	10325476	Дерево
Зона хранения1 {g }Plastic_Z1	COILS	13	bool	ReadWrite	10325476	Пластик
Зона хранения1 {g }Red_Z1	COILS	14	bool	ReadWrite	10325476	Красный

## Продолжение Приложения В

Продолжение таблицы - Переменные OPC – сервера.

Зона хранения1 {g }Green_Z1	COILS	15	bool	ReadWrite	10325476	Зеленый
Зона хранения1 {g }Blue_Z1	COILS	16	bool	ReadWrite	10325476	Синий
Зона хранения2 {g }Metal_Z2	COILS	17	bool	ReadWrite	10325476	Металл
Зона хранения2 {g }Wood_Z2	COILS	18	bool	ReadWrite	10325476	Дерево
Зона хранения2 {g }Plastic_Z2	COILS	19	bool	ReadWrite	10325476	Пластик
Зона хранения2 {g }Red_Z2	COILS	20	bool	ReadWrite	10325476	Красный
Зона хранения2 {g }Green_Z2	COILS	21	bool	ReadWrite	10325476	Зеленый
Зона хранения2 {g }Blue_Z2	COILS	22	bool	ReadWrite	10325476	Синий
Зона хранения3 {g }Metal_Z3	COILS	23	bool	ReadWrite	10325476	Металл
Зона хранения3 {g }Wood_Z3	COILS	24	bool	ReadWrite	10325476	Дерево
Зона хранения3 {g }Plastic_Z3	COILS	25	bool	ReadWrite	10325476	Пластик
Зона хранения3 {g }Red_Z3	COILS	26	bool	ReadWrite	10325476	Красный
Зона хранения3 {g }Green_Z3	COILS	27	bool	ReadWrite	10325476	Зеленый
Зона хранения3 {g }Blue_Z3	COILS	28	bool	ReadWrite	10325476	Синий

## Продолжение Приложения В

Продолжение таблицы - Переменные OPC – сервера.

Зона хранения4 {g}Metal_Z4	COILS	29	bool	ReadWrite	10325476	Металл
Зона хранения4 {g}Wood_Z4	COILS	30	bool	ReadWrite	10325476	Дерево
Зона хранения4 {g}Plastic_Z4	COILS	31	bool	ReadWrite	10325476	Пластик
Зона хранения4 {g}Red_Z4	COILS	32	bool	ReadWrite	10325476	Красный
Зона хранения4 {g}Green_Z4	COILS	33	bool	ReadWrite	10325476	Зеленый
Зона хранения4 {g}Blue_Z4	COILS	34	bool	ReadWrite	10325476	Синий
Зона хранения5 {g}Metal_Z5	COILS	35	bool	ReadWrite	10325476	Металл
Зона хранения5 {g}Wood_Z5	COILS	36	bool	ReadWrite	10325476	Дерево
Зона хранения5 {g}Plastic_Z5	COILS	37	bool	ReadWrite	10325476	Пластик
Зона хранения5 {g}Red_Z5	COILS	38	bool	ReadWrite	10325476	Красный
Зона хранения5 {g}Green_Z5	COILS	39	bool	ReadWrite	10325476	Зеленый
Зона хранения5 {g}Blue_Z5	COILS	40	bool	ReadWrite	10325476	Синий
Компрессор {g}Pusc_com	COILS	41	bool	ReadWrite	10325476	Запуск остановка компрессора
Компрессор {g}Vrabote_com	COILS	42	bool	ReadWrite	10325476	Статус о запуске компрессора
Ц1 {g}Pusk_Cyl1	COILS	43	bool	ReadWrite	10325476	Пуск/Стоп пневмоцилиндра

## Продолжение Приложения В

Продолжение таблицы - Переменные OPC – сервера.

Ц1 {g}Time_cyl1	HOLDING_REGISTERS	105	int16	ReadWrite	10325476	Время сработки пневмоцилиндра
Ц1 {g}Pusk_Ruch_cyl1	COILS	53	bool	ReadWrite	10325476	Ручной пуск Ц1
Ц1 {g}Auto_cyl1	COILS	54	bool	ReadWrite	10325476	Режим Авто Ц1
Ц2_1 {g}Pusk_Cyl2_1	COILS	44	bool	ReadWrite	10325476	Пуск/Стоп пневмоцилиндра
Ц2_1 {g}Time_cyl2_1	HOLDING_REGISTERS	106	int16	ReadWrite	10325476	Время сработки пневмоцилиндра
Ц2_1 {g}Pusk_Ruch_cyl2_1	COILS	55	bool	ReadWrite	10325476	Ручной пуск Ц2_1
Ц2_1 {g}Auto_cyl2_1	COILS	56	bool	ReadWrite	10325476	Режим Авто Ц2_1
Ц2_2 {g}Pusk_Cyl2_2	COILS	45	bool	ReadWrite	10325476	Пуск/Стоп пневмоцилиндра
Ц2_2 {g}Time_cyl2_2	HOLDING_REGISTERS	107	int16	ReadWrite	10325476	Время сработки пневмоцилиндра
Ц2_2 {g}Pusk_Ruch_cyl2_2	COILS	57	bool	ReadWrite	10325476	Ручной пуск Ц2_2
Ц2_2 {g}Auto_cyl2_2	COILS	58	bool	ReadWrite	10325476	Режим Авто Ц2_2
Ц2_3 {g}Pusk_Cyl2_3	COILS	46	bool	ReadWrite	10325476	Пуск/Стоп пневмоцилиндра
Ц2_3 {g}Time_cyl2_3	HOLDING_REGISTERS	108	int16	ReadWrite	10325476	Время сработки пневмоцилиндра
Ц2_3 {g}Pusk_Ruch_cyl2_3	COILS	59	bool	ReadWrite	10325476	Ручной пуск Ц2_3
Ц2_3 {g}Auto_cyl2_3	COILS	60	bool	ReadWrite	10325476	Режим Авто Ц2_3
Ц2_4 {g}Pusk_Cyl2_4	COILS	47	bool	ReadWrite	10325476	Пуск/Стоп пневмоцилиндра
Ц2_4 {g}Time_cyl2_4	HOLDING_REGISTERS	109	int16	ReadWrite	10325476	Время сработки пневмоцилиндра
Ц2_4 {g}Pusk_Ruch_cyl2_4	COILS	61	bool	ReadWrite	10325476	Ручной пуск Ц2_4
Ц2_4 {g}Auto_cyl2_4	COILS	62	bool	ReadWrite	10325476	Режим Авто Ц2_4

## Продолжение Приложения В

Продолжение таблицы - Переменные OPC – сервера.

Ц2_5{g}Pusk_Cyl2_5	COILS	48	bool	ReadWrite	10325476	Пуск/Стоп пневмоцилиндра
Ц2_5{g}Time_cyl2_5	HOLDING_REGISTERS	110	int16	ReadWrite	10325476	Время сработки пневмоцилиндра
Ц2_5{g}Pusk_Ruch_cyl2_5	COILS	63	bool	ReadWrite	10325476	Ручной пуск Ц2_5
Ц2_5{g}Auto_cyl2_5	COILS	64	bool	ReadWrite	10325476	Режим Авто Ц2_5
Ц3{g}Pusk_Cyl3	COILS	49	bool	ReadWrite	10325476	Пуск/Стоп пневмоцилиндра
Ц3{g}Time_cyl3	HOLDING_REGISTERS	111	int16	ReadWrite	10325476	Время сработки пневмоцилиндра
Ц3{g}Pusk_Ruch_cyl3	COILS	65	bool	ReadWrite	10325476	Ручной пуск Ц3
Ц3{g}Auto_cyl3	COILS	66	bool	ReadWrite	10325476	Режим Авто Ц3
A1{g}Pusk_P4	COILS	50	bool	ReadWrite	10325476	Запуск/остановка присоски
A1{g}Pusk_Ruch_P4	COILS	67	bool	ReadWrite	10325476	Ручной пуск Ц4
A1{g}Auto_P4	COILS	68	bool	ReadWrite	10325476	Режим Авто Ц4
Prog{g}Настройки заготовки{g}H	HOLDING_REGISTERS	126	int16	ReadWrite	10325476	
Prog{g}Настройки заготовки{g}R	HOLDING_REGISTERS	127	int16	ReadWrite	10325476	
Prog{g}Настройки заготовки{g}Ro_plastik	HOLDING_REGISTERS	128	int16	ReadWrite	10325476	
Prog{g}Настройки заготовки{g}Ro_wood	HOLDING_REGISTERS	129	int16	ReadWrite	10325476	
Prog{g}Настройки заготовки{g}Gotov	COILS	24	bool	ReadWrite	10325476	



## Продолжение Приложения В

Продолжение таблицы - Переменные OPC – сервера.

Prog{g}Zapusc_sort	COILS	51	bool	ReadWrite	10325476	Запуск алгоритма сортировки
Prog{g}Stob_Gloal	COILS	52	bool	ReadWrite	10325476	Аварийный стоп (Кнопка стоп на стартовом экране)
Prog{g}Material	HOLDING_REGISTERS	114	int16	ReadWrite	10325476	Материал
Prog{g}Zona_Hran	HOLDING_REGISTERS	116	int16	ReadWrite	10325476	Зона хранилища
U1{g}Ves	HOLDING_REGISTERS	113	int16	ReadWrite	10325476	Вес
D2{g}Range{g}Red_in_PL C	HOLDING_REGISTERS	118	int16	ReadWrite	10325476	
D2{g}Range{g}Green_in_PL C	HOLDING_REGISTERS	119	int16	ReadWrite	10325476	
D2{g}Range{g}Blue_in_PL C	HOLDING_REGISTERS	120	int16	ReadWrite	10325476	
D2{g}Текущие показания{g}Red_inARM	HOLDING_REGISTERS	121	int16	ReadWrite	10325476	
D2{g}Текущие показания{g}Green_inARM	HOLDING_REGISTERS	122	int16	ReadWrite	10325476	
D2{g}Текущие показания{g}Blue_inARM	HOLDING_REGISTERS	123	int16	ReadWrite	10325476	
D2{g}Cvet	HOLDING_REGISTERS	115	int16	ReadWrite	10325476	Цвет