



Программное обеспечение

СервоМастер[©]

Руководство оператора

Версия 1.0

Москва 2013

Содержание:

1.	Введение.....	5
1.1.	Назначение «Руководства оператора».....	5
1.2.	Основные положения.....	5
2.	Назначение программы и необходимая информация.....	6
2.1.	Информация о программе СервоМастер [©]	6
2.2.	Основное назначение и функции программы СервоМастер [©]	6
2.3.	Архитектура СЧПУ СервоКон [©]	7
2.3.1.	Общие сведения.....	7
2.3.2.	Компоненты СЧПУ СервоКон [©]	8
2.3.3.	Подсистема HAL.....	10
2.3.4.	Подсистема ПЛК.....	18
2.3.4.1.	Структура программы ПЛК.....	18
2.3.4.2.	Взаимодействие с подсистемой HAL.....	19
2.3.5.	Программное представление аппаратных ресурсов СЧПУ СервоКон [©] 22	
2.3.5.1.	Дискретные входы/выходы	22
2.3.5.2.	АЦП	23
2.3.5.3.	ЦАП	23
2.3.5.4.	Шина CAN.....	24
2.3.6.	Логические ресурсы СЧПУ СервоКон [©]	25
2.3.6.1.	Контроллер движения.....	25
2.3.6.2.	Шпиндель.....	41
2.3.6.3.	Программно-управляемые выходы	41
2.3.6.4.	СОЖ.....	42
2.3.6.5.	Сменщик инструментов.....	42
3.	Условия выполнения программы СервоМастер [©]	43
3.1.	Условия, необходимые для выполнения программы.....	43
3.2.	Состав программных средств	43
3.3.	Состав технических средств	43
4.	Выполнение программы СервоМастер [©]	44
4.1.	Установка программы.....	44
4.2.	Запуск	44
4.2.1.	Описание запуска программы. Начало работы	44

4.2.2.	Создание или загрузка конфигурации	46
4.2.2.1.	Создание конфигурации	46
4.2.2.2.	Загрузка сохраненной конфигурации	47
4.2.2.3.	Сохранение конфигурации	47
4.2.2.4.	Работа с проектом	47
4.2.3.	Главное меню	48
4.2.3.1.	Файл	49
4.2.3.2.	Правка	49
4.2.3.3.	Вид	50
4.2.3.4.	СервоКон	50
4.2.3.5.	Справка	51
4.2.4.	Библиотека блоков	51
4.2.4.1.	Раздел библиотеки Motion	52
4.2.4.2.	Раздел библиотеки PLC	57
4.2.4.3.	Раздел библиотеки I/O	58
4.2.4.4.	Раздел библиотеки Functions	63
4.2.4.5.	Раздел библиотеки Kinematics	64
4.3.	Выполняемые процедуры и функции	65
4.3.1.	Работа с блоками	65
4.3.1.1.	Добавление блоков в программу	65
4.3.1.2.	Соединение блоков	66
4.3.1.3.	Удаление блоков и соединений	67
4.3.1.4.	Перемещение блоков	67
4.3.1.5.	Добавление комментариев	67
4.3.2.	Установка значений параметров	67
4.3.3.	Редактирование таблицы инструментов	68
4.3.4.	Написание программ ПЛК	69
4.3.5.	Подключение к СЧПУ	71
4.3.6.	Установка параметров СЧПУ	72
4.3.7.	Настройка значений параметров	72
4.3.8.	Осциллограф	72
4.3.9.	Отладка программ ПЛК	75
5.	Составление конфигурации СЧПУ СервоКон [©]	79
5.1.	Выбор типа кинематики	79

5.2.	Настройка сигнала разрешения работы.....	80
5.3.	Настройка выхода в нулевую позицию.....	82
5.4.	Настройка максимального рассогласования.....	83
5.5.	Настройка аппаратных и программных ограничений положения.....	83
5.6.	Управление по позиции по шине CAN.....	84
5.6.1.	Подключение заданий и обратных связей сервоприводов.....	84
5.6.2.	Настройка передаточных отношений.....	86
5.6.3.	Настройка сервоприводов.....	86
5.7.	Настройка дополнительных функций.....	87
5.7.1.	Ручное управление.....	87
5.7.2.	Управление шпинделем.....	88
5.7.3.	Управление подачей СОЖ.....	88
5.7.4.	Настройка сменщика инструментов.....	88
5.8.	Создание конфигурации для СЧПУ СервоКон [©]	90
5.9.	Удаленная запись конфигурации.....	90
6.	Сообщения оператору.....	91
7.	Приложения.....	95
7.1.	Примеры конфигураций.....	95
7.1.1.	Станок для листового раскроя.....	95
7.1.2.	Фрезерный консольный станок.....	98
7.1.3.	Фрезерный порталный станок.....	100
7.2.	Примеры программ ПЛК.....	103
7.2.1.	Программа ПЛК для управления плазменной горелкой.....	103
7.2.2.	Программа ПЛК для вывода сообщений по внешним входам на дисплей.....	104
7.2.3.	Программа дублирования кнопок клавиатуры внешними выносными кнопками.....	106
7.3.	Перечень модулей Python.....	111

Особые указания по пользованию руководством

Отдельные указания имеют следующее значение:

**ОПАСНОСТЬ:**

Означает, что непринятие соответствующих мер подвергает опасности жизнь и здоровье пользователя и приводит к риску повреждения аппаратуры.

**ВНИМАНИЕ:**

Указывает на то, что не принятие соответствующих мер может привести к неправильной работе СЧПУ СервоКон[®] или программы СервоМастер[®].

**ПРИМЕЧАНИЕ:**

Указывает на то, что данная информация способствует наилучшему пониманию.

Список сокращений:

ПДУ – пульт дистанционного управления;

СПШ – сервопривод на базе гибридного шагового электродвигателя;

СПС – сервопривод на базе синхронного двигателя;

СЧПУ – система числового программного управления;

CAN – интерфейс Controller Area Network по стандарту ISO 11898;

ПО – программное обеспечение;

АЦП – аналого-цифровой преобразователь;

ЦАП – цифро-аналоговый преобразователь;

СОЖ – смазочно-охлаждающая жидкость;

ПЛК – программируемый логический контроллер;

ПИД регулятор – пропорционально интегрально дифференциальный регулятор.

1. Введение

1.1. Назначение «Руководства оператора»

Руководство оператора описывает правила составления конфигурации СЧПУ СервоКон[©] с использованием программного обеспечения СервоМастер[©].

1.2. Основные положения

Руководство Оператора состоит из следующих разделов:

- *Назначение программы.*

Содержит: общие сведения о программе СервоМастер[©], сведения о принципах функционирования и ресурсах СЧПУ СервоКон[©], необходимые для составления конфигураций СЧПУ.

- *Условия выполнения программы.*

Содержит: технические, программные, ресурсные условия, необходимые для установки и работы с программой СервоМастер[©].

- *Выполнение программы.*

Содержит: описание процессов загрузки программы, запуска программы, выполняемых процедур и функций.

- *Составление конфигурации.*

Содержит: описание процесса составления конфигурации СЧПУ.

- *Сообщения оператору.*

Содержит: возможные текстовые сообщения оператору и действия, которые необходимо предпринять оператору в каждом описываемом случае.

- *Приложения.*

Содержит: примеры конфигураций, примеры программ программируемого логического контроллера ПЛК, пояснения к примерам.

2. Назначение программы и необходимая информация

2.1. Информация о программе СервоМастер[©]

Программное обеспечение СервоМастер[©] разработано специалистами ЗАО «Сервотехника». Программное обеспечение СервоМастер[©] распространяется бесплатно через официальный сайт <http://www.servotechnica.ru> и предназначено для настройки работы устройств, изготавливаемых и поставляемых ЗАО «Сервотехника».

2.2. Основное назначение и функции программы СервоМастер[©]

Программа СервоМастер[©] предназначена для конфигурирования, программирования и наладки СЧПУ СервоКон[©] (далее СЧПУ). Программа позволяет:

- задавать конфигурацию подключенного оборудования (сервоприводы, концевые выключатели, шпиндель и т.п.) к СЧПУ;
- настраивать значения параметров СЧПУ, в том числе в режиме он-лайн;
- производить отладку программ ПЛК;
- производить отладку конфигурации и анализировать работу СЧПУ.

Основными функциями программы являются:

- создание и редактирование конфигураций для СЧПУ СервоКон[©];
- написание программ для ПЛК СЧПУ СервоКон[©];
- отладка написанных или загруженных программ ПЛК;
- анализ параметров работы СЧПУ.



ОПАСНОСТЬ: Не приступайте к подключению и конфигурированию СЧПУ без ознакомления с данным руководством оператора. Неправильно составленная конфигурация может повредить ваше оборудование. Для понимания процесса конфигурирования СЧПУ необходимо ознакомиться с комплексом ресурсов СЧПУ, описанных в п. 2.3 «Архитектура СЧПУ СервоКон[©]».



ПРИМЕЧАНИЕ: Для понимания работы с программой СервоМастер[©] необходимо ознакомиться с функционалом, возможностями и принципами работы программного обеспечения СервоМастер[©], – п.п. 3, 4, 6. Для наилучшего понимания процесса конфигурирования в п. 7 Приложения Руководства оператора приведены примеры конфигураций с краткими пояснениями.

2.3. Архитектура СЧПУ СервоКон[©]



ВНИМАНИЕ: Изучение данного раздела необходимо для понимания структурирования конфигурации. Запрещается приступать к отладке конфигурации без ознакомления и изучения данного раздела.

2.3.1. Общие сведения

СЧПУ СервоКон[©] – система ЧПУ общего назначения, предназначенная для выполнения задач управления исполнительными устройствами станков. СЧПУ СервоКон[©] представляет собой вычислительную систему, способную функционировать в режиме реального времени. В качестве источника синхронизации системы используется таймер, каждый период которого соответствует одному циклу управления. Задачи реального времени запускаются на выполнение при каждом срабатывании таймера. Задачи реального времени имеют наивысший приоритет и выполняются в первую очередь.

В течение одного цикла управления выполняются следующие действия:

- обновляются все порты ввода, в том числе и обратная связь от сервоприводов;
- выполняется интерполяция и связанные с ней расчеты в соответствии с текущей конфигурацией;
- пересылается задание в сервоприводы, и обновляются порты вывода.

В течение времени, свободного от выполнения задач реального времени, СЧПУ выполняет прочие задачи, такие как обработку человеко-машинного интерфейса, программы ПЛК и т.д.

На Рис. 1 показана схема обработки задач реального времени, задач нереального времени и время цикла управления.

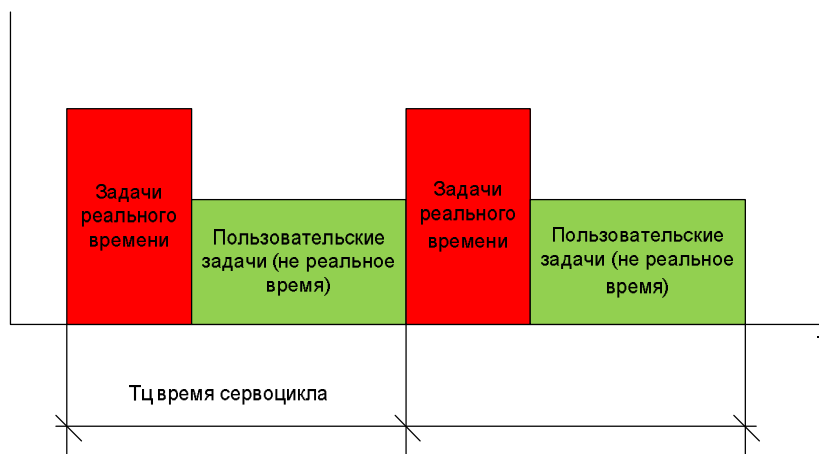


Рис. 1. Задачи реального времени, задачи нереального времени и цикл управления.

2.3.2. Компоненты СЧПУ СервоКон[®]

На Рис. 2 приведена функциональная схема архитектуры СЧПУ СервоКон[®].
На схеме приведены следующие логические элементы и связи между ними:

Элементы схемы, работающие в режиме нереального времени:

- **Interpreter** – компонент считывает исполняемую программу и преобразует её во внутреннее представление;
- **PLC** – компонент предназначен для интерпретации и исполнения программ ПЛК. Программы ПЛК предоставляют возможность выполнения действий, не привязанных к программам движения, например, управления электроавтоматикой;
- **Controller** – компонент предназначен для выполнения СЧПУ команд пользователя, переданных через интерфейс пользователя или другие внешние приложения.

Элементы схемы, работающие в режиме реального времени:

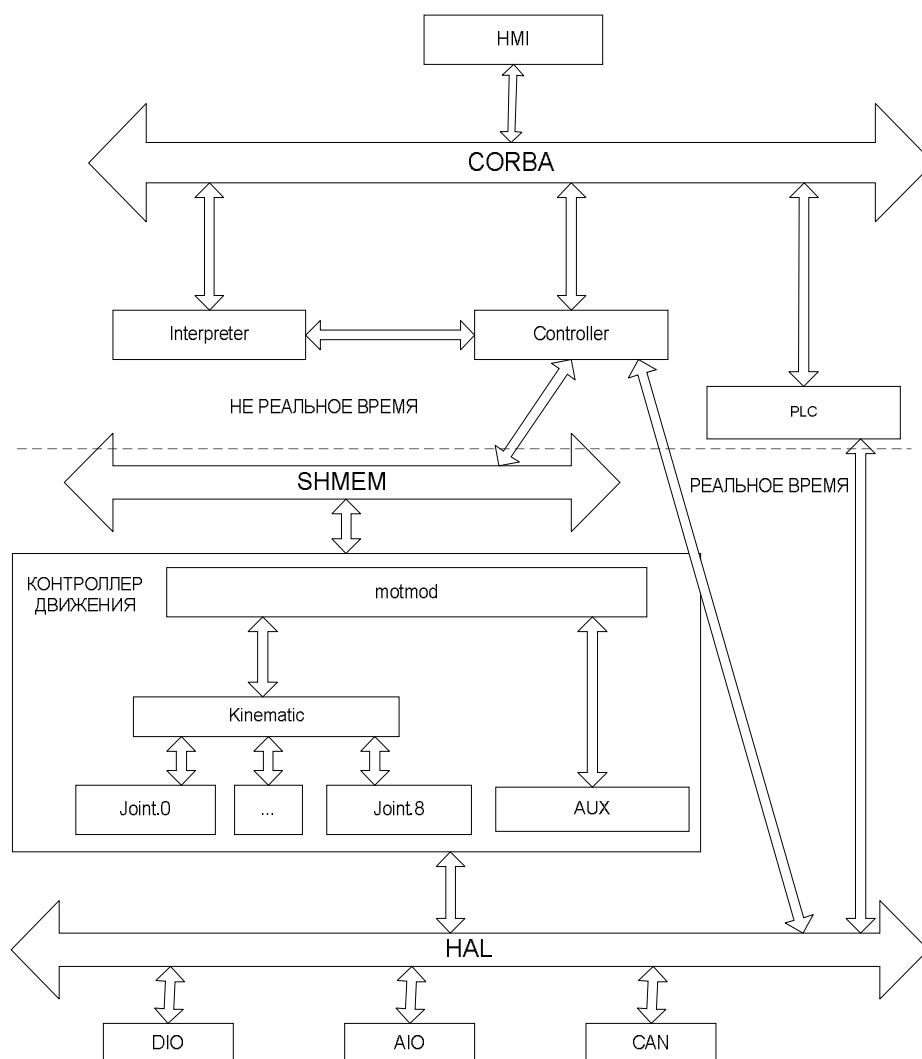
- **Контроллер движения** – компонент, отвечающий за планирование траектории, выполнение кинематических преобразований, вычисление заданий сервоприводам и за иные дополнительные функции, например, управление шпинделем. В состав контроллера движения входят:
 - **kinematics** – компонент, отвечающий за выполнение кинематических преобразований. СЧПУ выполняет преобразование заданий для исполнительных устройств из декартовой системы координат в системы ко-

ординат исполнительных устройств и обратной связи от исполнительных устройств в декартову систему координат;

- **motmod** (*контроллер движения*), **joint** (*привод*) – компоненты, отвечающие за вычисление заданий для исполнительных устройств. В соответствии с заданием, СЧПУ каждый цикл управления выполняет вычисление заданной позиции для всех исполнительных устройств. Также выполняется контроль безопасности выполнения работы исполнительных устройств. СЧПУ постоянно отслеживает величину рассогласования исполнительных устройств, сигналы ошибок, ограничений и аварийного останова;
 - **AUX** (*дополнительные функции*) – компонент, отвечающий за управление смазочно-охлаждающей жидкостью (СОЖ), управление шпинделем, управление сменщиком инструментов.
- **Драйверы ввода/вывода** (**DIO** (*дискретные входы/выходы*), **AIO** (*аналоговые входы/выходы*), **CAN** (*шина CAN*)) – компоненты, отвечающие за выдачу необходимых сигналов управления на физические выходы и считывание сигналов с входов.

Программные шины для связи компонентов между собой:

- **SHMEM** (*разделяемая память*) – область памяти, с помощью которой элементы нереального времени взаимосвязаны с элементами реального времени;
- **HAL** (*hardware abstraction layer*) – программный компонент, позволяющий выполнять передачу и обработку данных в реальном времени;
- **CORBA** (*сетевая архитектура*) – сетевая технология, на основе которой в СЧПУ СервоКон[©] реализуется взаимодействие со сторонним программным обеспечением.

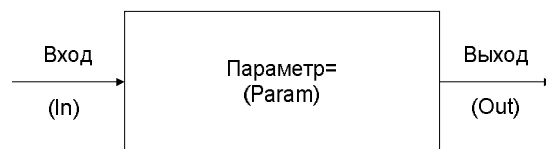
Рис. 2. Функциональная схема СЧПУ СервоКон[®].

2.3.3. Подсистема HAL

СЧПУ имеет в своём составе ряд программных компонентов: контроллер движения, сервоприводы, ПИД-регуляторы, драйверы ввода/выводы и т.д. Для связи программных компонентов в СЧПУ реализована подсистема HAL (Hardware Abstraction Layer – уровень абстракции оборудования). Конфигурация HAL составляется в программе СервоМастер[®]. Компоненты СЧПУ в HAL представлены в виде блоков, логика работы которых определяется:

- входами (In),
- выходами (Out),
- параметрами (Param=).

Принципиальный вид блока показан на Рис. 3.

Рис. 3. Принципиальный вид блока, представленный в программе СервоМастер[®].

Подсистема HAL позволяет путем соединения логических элементов СЧПУ по аналогии с электрическими схемами реализовать вывод нужных управляющих сигналов на выходы СЧПУ и считывание сигналов состояния с входов СЧПУ, а также реализовать простую логику обработки этих сигналов, критичную ко времени выполнения. Эти действия задаются в конфигурации в виде диаграммы функциональных блоков. Описание процесса соединения логических элементов в программе СервоМастер[®] см. в п. 4.3.1.2.

Каждый цикл управления обновляются значения входов всех блоков (на основании значений подключенных к ним элементов) и рассчитываются значения выходов блоков на основании текущих значений входов и параметров блоков. Важно отметить, что выходные значения функциональных блоков рассчитываются не одновременно, а последовательно, и соответственно работа диаграмм, в которых присутствуют циклические структуры, может отличаться от ожидаемой.

Табл. 1. Типы входов/выходов и их значения.

Тип	Описание	Значение
Bool	Логический тип для описания сигналов типа «включен/выключен»	True/False
Float	Значение с плавающей точкой одинарной точности	Вещественное значение, мантисса – 23 бита (6-7 значащих десятичных цифр)
S32	Целочисленный тип	Знаковое целое значение от -2^{32} до $2^{32} - 1$
Index	Специальный тип логического сигнала, который используется для z-метки	-

Значения параметров (Param) устанавливаются в конфигурации для того или иного блока. Параметры задаются через интерфейс пользователя или программы ПЛК.

Имеется два типа блоков: динамические и фиксированные.

К динамическим относятся блоки, при добавлении в конфигурацию которых создается новый экземпляр. Экземпляров динамических блок одного и того же типа может быть сколь угодно много в рамках одной конфигурации. К таким блокам относятся константа (constant), логические «И» (and) и т.д. Все блоки, входящие в состав Табл. 2. «Логические функции», являются динамическими.

К фиксированным относятся блоки, которые могут иметь только один экземпляр в рамках одной конфигурации. Эти блоки представляют компоненты СЧПУ, существующие в единственном экземпляре, такие как контроллер движения и драйверы ввода/вывода. При повторном добавлении в конфигурацию одного и того же фиксированного блока не создается новый объект, а делается две ссылки на один и тот же объект, т.е. на любой фиксированный блок можно делать неограниченное число ссылок в разных частях конфигурации. К фиксированным блокам относятся такие блоки как «Направление вращения шпинделя» (motmod.AUX.Spindle.Forward).

В Табл. 2 «Логические функции» описаны блоки, которые позволяют реализовать логику обработки сигналов в конфигурации.

Табл. 2. Логические функции.

Функция	Описание	Входы	Выходы	Параметры
and	Логическое И – выход принимает значение 1, когда оба входа равны 1, и 0 в противном случае	in0 – Bool in1 – Bool	out – Bool	Нет
or	Логическое ИЛИ – выход принимает значение 1, когда любой из входов равен 1, и 0 в противном случае	in0 – Bool in1 – Bool	Out – Bool	Нет
not	Логическое НЕ – выход принимает значение 1, когда вход равен 0	in – Bool	out – Bool	Нет
rstrigger	RS-триггер – при установке входа r в 1 значение выхода устанавливается в 0 и сохраняется до установки входа s в 1. При наличии значения True на входах r и s одновременно значение на выходе не изменяется	r – Bool s – Bool	out – Bool	Нет
edge	Детектор фронтов выдаёт на выходе значение 1 при появлении на входе фронта. По истечении времени, заданного пользователем в параметре out-width-ns, выход блока снова принимает значение 0	in – Bool	out – Bool	out-width-ns — длительность импульса (округляется до длительности цикла управления) InEdge – тип фронта 1: Переход от 0 в 1 0: Переход от 1 в 0
scale	Масштабирование $out = in * gain + offset$	in – Float	out – Float	gain – масштаб offset – смещение
sum	Сложение $out = in0 * gain0 + in1 * gain1 + offset$	in0 – Float in1 – Float	out – Float	gain0 – масштаб входа 0 gain1 – масштаб входа 1 offset – смещение
comp	Сравнение. Выход блока принимает значение 1, если значения входов	in0 – Float	equal – Bool	hyst – разница значения входов, при которой

	различаются не более чем на hyst	in1 – Float		они считаются равными
abs	Абсолютное значение числа без учета знака	in – Float	out – Float sign – Bool	Нет
timedelay	Задержка, сигнал на выходе изменяется по прошествии соответствующей задержки после изменения сигнала на входе если в течении этого времени значение входа сохранялось постоянным	in – Bool	out – Bool	on_delay – задержка появления значения 1 на выходе off_delay – задержка появления значения 0 на выходе
debounce	Блок предназначен для отбрасывания ложных срабатываний, например, концевых выключателей, которые могут возникать в ряде случаев в условиях помех или дребезга контактов	in – Bool	out – Bool	delay – время, в по истечении которого значение входа передается на выход, при условии сохранения значения входа в течение всего периода

Табл. 2. Логические функции (продолжение)

Функция	Описание	Входы	Выходы	Параметры
pid	ПИД – регулятор	command – Float feedback – Float tnable – Bool	output – Float	Pgain – Float – пропорциональный коэффициент Igain – Float – интегральный коэффициент Dgain – Float – дифференциальный коэффициент FF0 – Float – коэффициент форсирующей связи 0-го порядка FF1 – Float – коэффициент форсирующей связи 1-го порядка FF2 – Float – коэффициент форсирующей связи 2-го порядка bias – Float – смещение выходного сигнала deadband – Float – мёртвая зона ошибки maxoutput – Float – максимальное значение выхода maxerror – Float – максимальное значение ошибки maxerrorI – Float – ограничение ошибки для вычисления интегральной составляющей maxerrorD – Float – ограничение ошибки для вычисления дифференциальной составляющей maxcmdD – Float – ограничение входа форсирующей связи 1-го порядка maxcmdDD – Float – ограничение входа форсирующей связи 2-го порядка

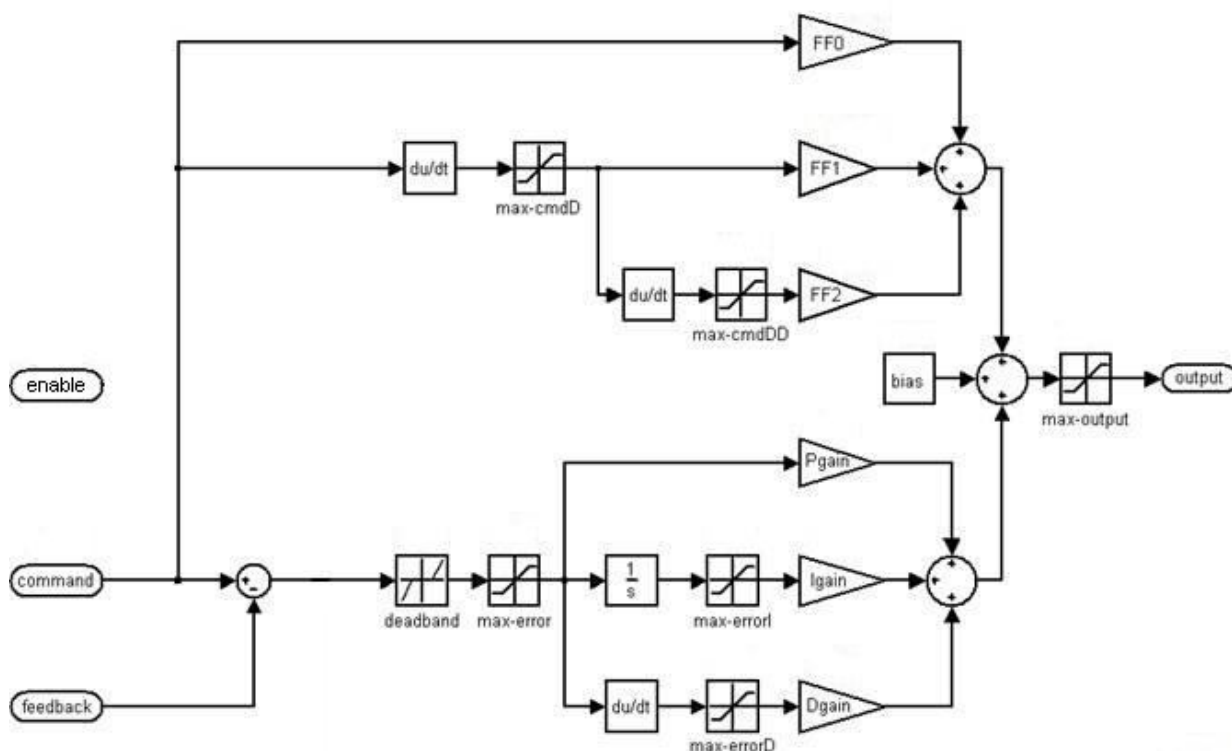


Рис. 4. Структура ПИД-регулятора.

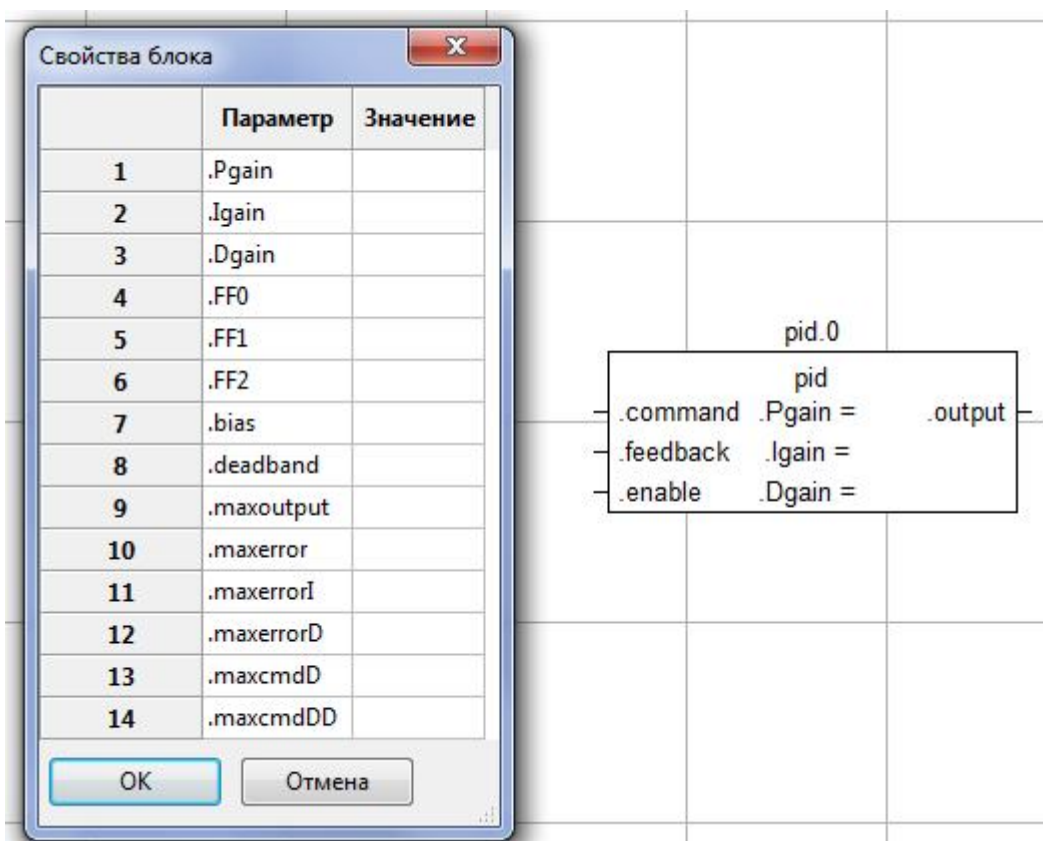


Рис. 5. Параметры настройки ПИД-регуляторов.

Табл. 2. Логические функции (продолжение)

Функция	Описание	Входы	Выходы	Параметры
ilowpass	Фильтр нижних частот общего назначения. Может использоваться, например, для исключения скачкообразных изменений задания при ручном управлении	in – S32	out – S32	scale – Float
mux2	Мультиплексор осуществляет выбор из двух входных сигналов на основании сигнала управления. Если sel принимает значение True, то на выход передается значение входа in1, в противном случае передается сигнал in0	in0 – Float in1 – Float sel – Bool	out – Float	нет
encoder	Программный энкодер – выполняет подсчет фронтов входных сигналов.	phase-A – (Bool) (канал A) phase-B – (Bool) (канал B)	counts – положение без масштабирования position – положение с масштабированием velocity – текущая скорость	position-scale (Float) – коэффициент масштабирования положения
constant	Константа – на выходе постоянное значение, равное значению параметра value	Нет	out – Float	value (Float) – требуемое значение
integ	Интегратор. Формула вычисления $out = out + in * Ts$, где Ts – период цикла управления	in – Float	out – Float	нет
siggen	Генератор сигналов	frequency – Float – частота amplitude – Float – амплитуда offset – Float – смещение	sine – Float cosine – Float sawtooth – Float triangle – Float square – Float	нет

Для логической функции Siggen на Рис. 6 показаны выходы sine – синусоида с параметрами, определяемыми входами, cosine – синусоида с параметрами, определяемыми входами смещённая на $\frac{1}{4}$ периода;

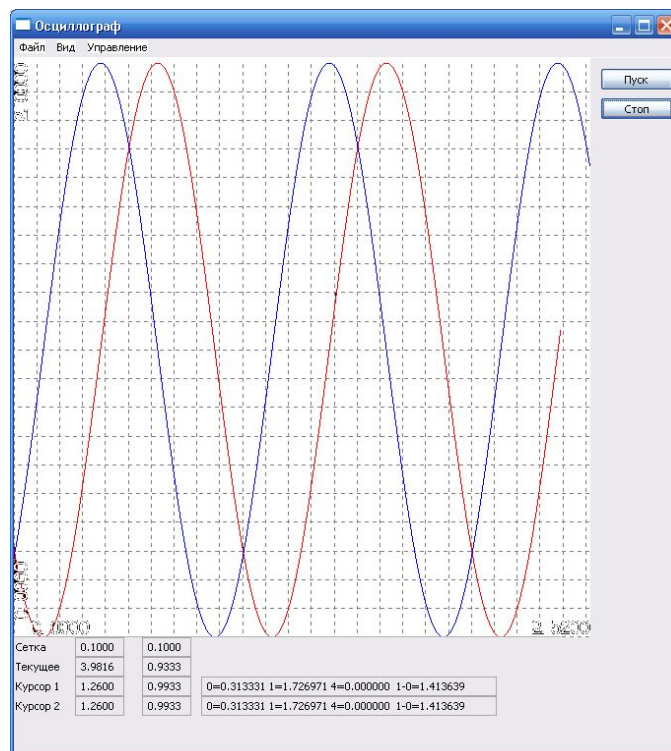


Рис. 6. Синус и косинус с амплитудой, частотой и периодом равными 1.0.

Для логической функции `siggen` на Рис. 7 показаны выходные сигналы `sawtooth` – пилообразный сигнал с параметрами, определяемыми входами, `triangle` – треугольный сигнал с параметрами, определяемыми входами, `square` – меандр с параметрами, определяемыми входами.

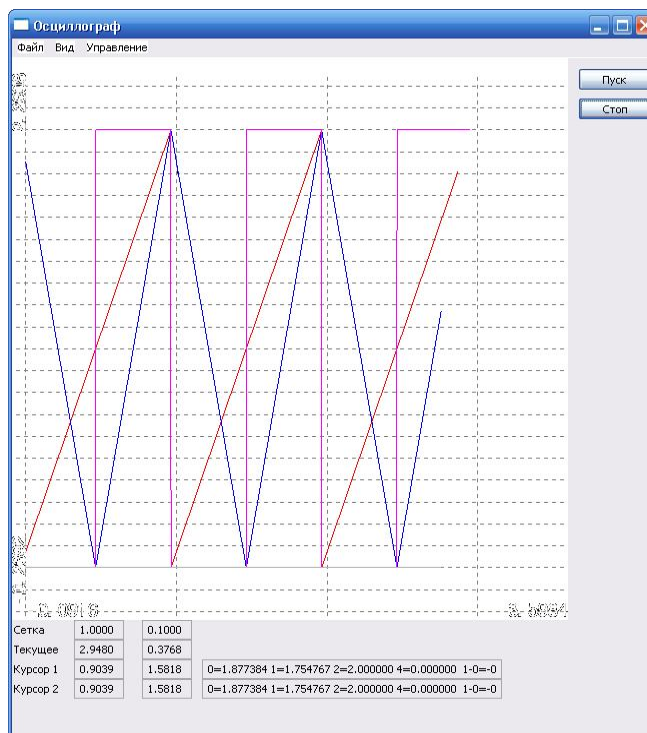


Рис. 7. Пила, треугольник и меандр с амплитудой, частотой и периодом, равными 1.0.

2.3.4. Подсистема ПЛК

Подсистема ПЛК предназначена для выполнения запрограммированных пользователем действий, например, реализации сложных циклов включения оборудования. Программы ПЛК разрабатываются на языке Python и могут взаимодействовать с другими компонентами СЧПУ, используя программные шины HAL и CORBA. Программы ПЛК запускаются на выполнение после контроллера движения и выполняются в режиме не реального времени. В программах ПЛК можно использовать ограниченный набор стандартных модулей Python (см. п. 7.3) и вспомогательные модули plc и hal.

Программы ПЛК разрабатываются и отлаживаются в СервоМастер[®]. Подробнее о редакторе программ см. п. 4.3.4. Подробнее об отладчике программ см. п. 4.3.9.

2.3.4.1. Структура программы ПЛК

Программа ПЛК представляет собой исполняемый модуль на языке Python 2.7. Модуль должен создать объект класса, унаследованного от класса plc.PLC. Этот класс представляет программу ПЛК. В классе необходимо переопределить методы InitIO и Run. Для запуска программы ПЛК на выполнение необходимо вызвать метод Start объекта. При создании программы ПЛК в СервоМастер[®] данный код генерируется автоматически.

Класс `plc.PLC(name)`

Конструктор класса создаёт объект ПЛК. Параметр `name` должен совпадать с именем файла программы без расширения.

`Start()`

Метод запускает программу ПЛК на выполнение.

`wait_for_controller()`

Метод, с помощью которого можно выполнить ожидание запуска всех компонент СЧПУ. Этот метод необходимо вызвать из метода `Run()`, если программе ПЛК требуется взаимодействие с СЧПУ через CORBA.

`InitIO()`

Метод создаёт входы, выходы и параметры, представляющие данную программу ПЛК в HAL. В данном методе необходимо создать входы, выходы и параметры программы ПЛК, которые будут представлять её в HAL. Входы, выходы и параметры создаются с помощью атрибута `io` класса `plc.PLC`, представляющего собой объект класса `hal.Component`.

`Run()`

В данном методе должен быть реализован основной поток программы ПЛК. После завершения метода `Run` программа ПЛК не будет выполнять никаких действий, а значения её входов и выходов в HAL будут оставаться неизменными.

2.3.4.2. Взаимодействие с подсистемой HAL

Программа ПЛК может взаимодействовать с другими компонентами СЧПУ, используя программную шину HAL. При этом программа ПЛК представляется в виде фиксированного блока с входами, выходами и параметрами, созданными в методе `InitIO`, и может быть использована аналогично прочим блокам.

Для создания входов, выходов и параметров блока используется модуль `hal`.

Модуль `hal` экспортирует следующие объекты:

```
hal.HAL_BIT  
hal.HAL_S32  
hal.HAL_FLOAT
```

Эти константы представляют типы HAL Bool, S32 и Float и используются при указании типа создаваемых входов, выходов и параметров.

```
hal.HAL_IN  
hal.HAL_OUT
```

Эти константы представляют направления элементов и используются при создании входов и выходов.

`hal.HAL_RW`

`hal.HAL_RO`

Эти константы представляют типы доступа к параметрам: полный доступ или доступ только на чтение.

Класс `hal.Component`

Данный класс предоставляет возможность создания входов, выходов и параметров.

Объект класса имеет следующие методы:

`newpin(name, typ, direct)`

Создаёт вход или выход типа `typ` с именем `name`. Если параметр `direct` равен `hal.HAL_IN`, то создаётся вход, если параметр `direct` равен `hal.HAL_OUT`, то создаётся выход

`newparam(name, typ, access)`

Создаёт параметр типа `typ`, типом доступа `access` и именем `name`.

Созданные входы, выходы и параметры доступны как атрибуты объекта в том случае, если их имена является корректными идентификаторами python или через операцию индексирования, например:

```
class Plc1(plc.PLC)
    def InitIO(self):
        #создать вход типа Bool
        self.io.newpin('direct', hal.HAL_BIT, hal.HAL_IN)
        #создать параметр типа S32
        self.io.newparam('count', hal.HAL_BIT, hal.HAL_RW)
        #создать выход типа Float
        self.io.newpin('my-out', hal.HAL_FLOAT, hal.HAL_OUT)
    def Run(self):
        while True:
            #увеличить значение параметра count на 1 если вход direct
            #установлен, иначе уменьшить на 1
            if self.io.direct:
                self.io.count += 1
            else:
                self.io.count -= 1
```

```
#вывести на выход 0.0 если count < 100 иначе count  
self.io['my-out'] = 0.0 if self.io.count<100 else  
float(self.io.count)
```

Примеры программ ПЛК и её включения в конфигурацию см. в п. 7 Приложения.

2.3.5. Программное представление аппаратных ресурсов СЧПУ СервоКон[®]

2.3.5.1. Дискретные входы/выходы

Дискретные порты ввода/вывода используются для ввода информации с датчиков и вывода управляющих сигналов релейного типа.

В качестве ресурсов ввода/вывода в СЧПУ СервоКон[®] могут выступать как ресурсы самой СЧПУ, расположенные в ней, так и ресурсы приводов СПС/СПШ, подключенные к ней и находящиеся под ее управлением. При этом частота обновления и считывания будет одинаковой и будет равна частоте цикла управления.

К дискретному вводу, как правило, подключаются: датчики концевого выключения, предназначенные для контроля выхода объекта управления за допустимые пределы рабочей зоны; датчики нулевой позиции, используемые для обнуления системы после включения; кнопки экстренного останова; переключатели режимов; а также другая электроавтоматика станка, используемая для автоматизации работы технологического оборудования. К дискретным выводам подключается оборудование релейного типа или, например, включение шпинделя.

Для получения значения дискретного входа подключите соответствующий ему выход драйвера к нужному элементу HAL.

Для установки значения дискретного выхода подключите к соответствующему ему входу драйвера нужный элемент HAL.

< k>, < m> и < n> в названии группы ресурсов – целочисленные индексы, определяющие номер элемента группы или интерфейса.

Табл. 3. Группа hal_servocon.io_bank<m> – ресурсы СЧПУ.

Имя	Направление	Тип	По умолчанию	Описание
Din.<n>.value	Out	Bool		Значение дискретного входа
Dout.<n>.value	In	Bool	False	Требуемое значение дискретного выхода

Табл. 4. Группа hal_servocon.CAN.<k>.dev.<m> - ресурсы сервоприводов.

Имя	Направление	Тип	По умолчанию	Описание
IO.d_in.<n>	Out	Bool		Значение дискретного входа
IO.d_out.<n>	In	Bool	False	Требуемое значение дискретного выхода

2.3.5.2. АЦП

АЦП используются для ввода аналоговой информации, например, датчиков уровней. Данные входы могут использоваться для автоматизации работы технологического оборудования.

В качестве ресурсов ввода/вывода в СЧПУ СервоКон[©] могут выступать как ресурсы самой СЧПУ, расположенные в ней, так и ресурсы приводов СПС/СПШ, подключенные к ней и находящиеся под ее управлением.

Для получения значения АЦП подключите соответствующий ему выход драйвера к нужному элементу HAL. При необходимости значение можно преобразовать, используя параметры scale и offset – значение выхода равно $(\text{входное напряжение}) * \text{scale} + \text{offset}$.

Табл. 5. Группа hal_servocon.io_bank<m> – ресурсы СЧПУ.

Имя	Направление	Тип	По умолчанию	Описание
Adc.<n>.value	Out	Float		Значение АЦП

Табл. 6. Группа hal_servocon.CAN.<k>.dev.<m> - ресурсы сервоприводов.

Имя	Направление /Параметр	Тип	По умолчанию	Описание
IO.a_in.<n>.value	Out	Float		Значение АЦП
IO.a_in.<n>.scale	Param	Float	1.0	Коэффициент масштабирования АЦП
IO.a_in.<n>.offset	Param	Float	0.0	Смещение АЦП

2.3.5.3. ЦАП

ЦАП используются для управления технологическими параметрами объекта управления, например, скоростью шпинделя. В совокупности с квадратурными входами могут использоваться для управления позицией в замкнутом режиме.

Для установки значения ЦАП подключите к соответствующему ему входу драйвера нужный элемент HAL. При необходимости значение можно преобразовать, используя параметры scale и offset – значение напряжения на ЦАП равно $(\text{value} - \text{offset}) * \text{scale}$ и ограничено диапазоном выходного напряжения ЦАП. Например, если установить Value = 15, Scale = 0.2, Offset = 3, то выходное напряжение будет равно 2,4 В. При установке Value = 1, Scale = 0.2, Offset = 5 выходное напряжение будет равно 0 В.

Табл. 7. Группа hal_servocon.io_bank<m> – ресурсы СЧПУ.

Имя	Направление /Параметр	Тип	По умолчанию	Описание
dac.<n>.value	In	Float	0.0	Требуемое значение ЦАП
dac.<n>.scale	Param	Float	0.0	Коэффициент масштабирования
dac.<n>.offset	Param	Float	0.0	Смещение

2.3.5.4. Шина CAN

CAN – (Controller Area Network) последовательная шина коллективного доступа, специально разработана для обеспечения взаимодействия промышленных управляющих контроллеров. Физический уровень интерфейса совместим со стандартом ISO 11898.

Приемопередатчик обеспечивает:

- совместимость со стандартом ISO 11898;
- скорость обмена до 1000 КБит/сек;
- до 120-ти активных узлов физического сегмента сети;
- «горячее» подключение к работающей сети;
- отсутствие помех и переходных процессов при включении или выключении данного устройства для других работающих устройств;
- гарантированную работу в условиях синфазных помех амплитудой до ± 25 В.

Приемопередатчик поддерживает:

- работоспособность при обрыве общего опорного провода или одной из линий дифференциальной пары;
- термальную защиту и защиту от статического электричества до 16 кВ.

СЧПУ СервоКон[®] может выполнять управление позицией сервоприводов СПШ/СПС, подключенных по шине CAN. Для этого с частотой сервоцикла СЧПУ передаёт в приводы задание позиции, требуемое состояние сигнала разрешения работы и требуемые значения аппаратных выходов приводов. Приводы, в свою очередь, передают в СЧПУ текущую позицию, сигнал готовности к работе и состояние аппаратных входов.

Табл. 8. Группа hal_servocon.CAN.<n>.

Имя	Направление / Параметр	Тип	По умолчанию	Описание
Status	Out	Bool		Состояние сервопривода – принимает значение 1, если сервопривод не готов к работе или с ним отсутствует связь
Enable	In	Bool	False	Сигнал разрешения работы сервопривода
Pos-req	In	Float	0.0	Задание положения [мм]
Vel-req	Зарезервирован (не используется)			
Pos-fb	Out	Float		Текущее положение [мм]
Index-enable	Out	Index		Сигнал поиска Z-метки
activated	Param	S32	0	Разрешение опроса сервопривода 1, если опрос разрешен
input-scale	Param	Float	1.0	Передаточное отношение [мм/инк]
vel-scale	Зарезервирован (не используется)			
index-search-dir	Out	S32	0	Направление поиска Z -метки 1 – в положительном направлении, 0 – в отрицательном
IO.d_in.<n>	Out	Bool		Значение дискретного входа
IO.d_out.<n>	In	Bool		Требуемое значение дискретного выхода
IO.a_in.<n>.value	Out	Float		Значение АЦП
IO.a_in.<n>.scale	Param	Float	1.0	Коэффициент масштабирования АЦП
IO.a_in.<n>.offset	Param	Float	0.0	Смещение АЦП

2.3.6. Логические ресурсы СЧПУ СервоКон[®]

2.3.6.1. Контроллер движения

Контроллер движения СЧПУ формирует задания для всех исполнительных устройств системы, исходя из управляющей программы и заданных ограничений позиции, скорости и ускорения, отслеживает состояния сигналов готовности устройств и сигналов выхода за границы рабочей зоны. Выполнение функций контроллера движения происходит в режиме реального времени. В состав контроллера движения входят интерполятор, функции кинематического преобразования и функции управления сервоприводами.

Кинематическое преобразование

Программы движения описывают траекторию перемещения инструмента в декартовых координатах. Для правильного формирования задания сервоприводам необходимо задать закон соответствия координат в терминах сервоприводов и декартовых координат инструмента. Установка соответствия координат сервоприводов и декартовых координат называется кинематическим преобразованием. Такое соответствие определяется типом и параметрами кинематики.

Обратное кинематическое преобразование – преобразование из декартовых координат в координаты сервоприводов. Присутствует во всех типах поддерживаемых кинематических преобразований.

Прямое кинематическое преобразование – преобразование из координат сервоприводов в декартову систему координат. В некоторых типах кинематических преобразований может отсутствовать, при этом обратная связь по позиции не вычисляется, а копируется из задания положения в декартовой системе координат и ряд функций, предполагающих наличие обратной связи (адаптивные алгоритмы, измерительные движения и т. д.) могут быть недоступны или работать иначе, чем в схемах с прямым преобразованием.

Во всех типах кинематических преобразований доступна матрица коррекции, позволяющая компенсировать неидеальность геометрии станка, например, неперпендикулярность осей. Полученное от интерполятора задание положения умножается на матрицу коррекции перед вычислением заданий сервоприводов.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \cdot \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix} = \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix}$$

При расчёте обратной связи полученные из позиций сервоприводов декартовы координаты умножаются на матрицу, обратную матрице коррекции C^{-1} перед передачей в контроллер движения.

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} \cdot C^{-1} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Обратная матрица вычисляется по формуле $C^{-1} = \frac{1}{|C|} \cdot C^T$, где $|C|$ – определитель матрицы, а C^T – транспонированная матрица. Транспонированная матрица C^T получается из матрицы C , заменой строк на столбцы.

Определитель матрицы вычисляется по формуле:

$$|C| = C_{11} \cdot C_{22} \cdot C_{33} - C_{11} \cdot C_{23} \cdot C_{32} - C_{12} \cdot C_{21} \cdot C_{33} + C_{12} \cdot C_{23} \cdot C_{31} + C_{13} \cdot C_{21} \cdot C_{32} - C_{13} \cdot C_{22} \cdot C_{31}$$



ВНИМАНИЕ: Если определитель матрицы коррекции равен 0, то вычисление обратной матрицы невозможно и функция кинематического преобразования не сможет выполнить преобразование координат приводов в декартовы координаты инструмента.

Формирование задания сервоприводу и получение обратной связи показано на Рис. 8.

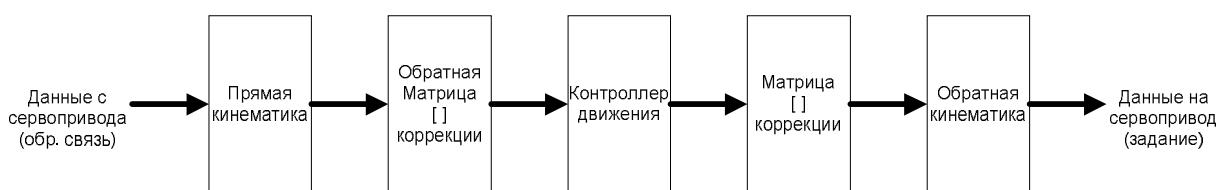


Рис. 8. Структурная схема получения обратной связи и формирования заданий сервоприводам.

В СЧПУ СервоКон[®] поддерживаются следующие типы кинематических преобразований:

Тривиальная кинематика (trivkins) – соответствие декартовых осей и сервоприводов один к одному.

Данная кинематическая схема используется в случаях, когда одна декартова ось приводится в движение одним сервоприводом. Соответствие сервоприводов осям определяется следующим образом:

- Сервопривод 0 – ось X;
- Сервопривод 1 – ось Y;
- Сервопривод 2 – ось Z;
- Сервопривод 3 – ось A;
- Сервопривод 4 – ось B;
- Сервопривод 5 – ось C;
- Сервопривод 6 – ось U;
- Сервопривод 7 – ось V;
- Сервопривод 8 – ось W.

Табл. 9. Параметры тривиальной кинематики (trivkins).

Имя	Параметр	Тип	По умолчанию	Описание
c.<m>.<n>	Param	Float	1 если m=n, иначе 0	Элемент матрицы коррекции

Портальная кинематика (gantrykins) – кинематика, в которой одна или несколько осей приводятся в движение более чем одним сервоприводом, работающих параллельно.

Данная кинематическая схема используется в случаях, когда одна декартова ось приводится в движение двумя или более сервоприводами. В этом случае обратная связь в декартовой системе координат вычисляется, используя положение сервопривода с наибольшим номером перемещающего данную ось. Соответствие декартовых осей сервоприводам определяется параметрами joint.<n>.axis.

Табл. 10. Параметры портальной кинематики (gantrykins).

Имя	Параметр	Тип	По умолчанию	Описание
Joint.<n>.axis	Param	S32	-	Ось сервопривода n
c.<m>.<n>	Param	Float	1 если m=n, иначе 0	Элемент матрицы коррекции



ОПАСНОСТЬ: Некорректная настройка сервоприводов станков с нетривиальной кинематикой может повредить станок. Перед выполнением движения убедитесь, что:

- Правильно настроены передаточные отношения сервоприводов.
- У сервоприводов установлены одинаковые максимальные скорости и ускорения.
- Одинаково настроены ПИД-регуляторы сервоприводов.
- У сервоприводов выбран один и тот же цикл выхода в 0.
- Команды ручного управления и разрешения ручного управления подключены к одному источнику.
- Установлены одинаковые масштабы ручного управления сервоприводов.

Обязательно проверьте корректность направлений вращения и реакции на концевые выключатели и датчики нулевого положения до соединения двигателей портала с механическими компонентами станка.

Интерполятор

Интерполятор (motmod), исходя из программы движения, типа кинематики и ограничений скорости (траекторного и для каждого сервопривода) и ограничений ускорения

(траекторного и для каждого сервопривода), с заданной частотой генерирует задание по позиции для всех сервоприводов.

Табл. 11. Группа элементов и параметров интерполятора motmod.

Имя	Параметр/ Направление	Тип	По умолчанию	Описание
MaxVelocity	Param	Float	0.0	Максимально допустимая контурная скорость
MaxAcceleration	Param	Float	0.0	Максимально допустимое контурное ускорение
servo_period_nsec	Param	S32	2000000	Период цикла управления [нс] (для управления по CAN должен совпадать с периодом обчёта контуров позиции)
Enable	In	Bool	False	Вход разрешения работы – при значении False СЧПУ снимает сигналы разрешения работы со всех приводов
Pause	In	Bool	False	Вход останова программы. Если вход активирован в процессе выполнения программы движения, то программа останавливается с возможностью продолжения с точки останова
FeedHold	In	Bool	False	Вход запрета перемещения – пока этот вход активен, СЧПУ устанавливает заданную нулевую скорость для всех сервоприводов

Управление разрешением работы

СЧПУ выполняет управление сервоприводами только в состоянии готовности. Переход в состояние готовности происходит только по команде от интерфейса пользователя.

СЧПУ переходит в состояние сброса при появлении одного из следующих сигналов ошибки любого из сервоприводов:

- срабатывания сигнала аппаратного ограничения положения;
- превышения программного ограничения положения;
- превышения допустимого уровня ошибки слежения.

В состоянии сброса СЧПУ снимает сигнал разрешения работы всех сервоприводов и игнорирует все команды на перемещение до перевода в состояние готовности.

При снятии сигнала разрешения работы Enable СЧПУ переходит в состояние аварийного останова. Данное состояние аналогично состоянию сброса за исключением того, что из него невозможно перейти в состояние готовности до появления сигнала разрешения работы.



ПРИМЕЧАНИЕ: переход в состояние аварийного останова (то есть остановка загрузки программ и информирование пользователя) осуществляется не модулем реального времени `motmod`, а компонентом `Controller`, работающим не в реальном времени. Таким образом, при кратковременном снятии сигнала разрешения будет только прекращено движение на время пропадания сигнала разрешения. Для реализации функции аварийного останова с блокированием рекомендуется использовать либо соответствующие аппаратные средства (например, кнопки с фиксацией) либо программный компонент `rstrigger`.

Функция запрета движения

При активации входа `FeedHold` СЧПУ устанавливает нулевое задание скорости для всех осей. Данная функция может использоваться для ожидания готовности внешнего оборудования.

Функция останова с продолжением

При активации входа `Pause` или нажатии кнопки «Стоп» СЧПУ остановит выполнение программы движения. При этом останавливается перемещение по всем осям, вращение шпинделя и подача СОЖ. При последующем запуске программы возможно продолжение выполнения с точки останова при следующих условиях:

- В программе до первого движения указан режим задания положения;
- В программе до момента останова задана текущая плоскость;
- В программе не изменяются единицы измерения;
- В программе не выполняется изменение нулей систем координат;
- В программе не используются нумерованные параметры.

Измерительное движение

При получении команды на выполнение измерительного движения СЧПУ выполняет запрограммированное движение до момента активации входа `Probe`. При активации входа СЧПУ захватит текущее положение и выполнит останов с заданными ограничениями ускорений. Если вход `Probe` был активен при начале измерительного движения или не был активирован до окончания запрограммированного движения, то СЧПУ остановит выполнение программы и сообщит об ошибке.

Режимы управления

- Ручной режим;
- Автоматический режим.

Ручной режим подразумевает независимое управление приводами. В ручном режиме каждый из сервоприводов при активном входе JogEnable при изменении сигнала JogCommand со значения cmd1 на значение cmd2 совершает перемещение на величину $(cmd2 - cmd1) * JogScale$.

Автоматический режим подразумевает последовательное чтение и выполнение команд пользователя на перемещение в декартовой системе координат. В данном режиме СЧПУ принимает команды, интерпретирует их, получая линейные или круговые перемещения, рассчитывает допустимые скорости и ускорения, и, исходя из них, формирует задание в декартовой системе координат. Выполняя обратное кинематическое преобразование, СЧПУ вычисляет заданные координаты для всех сервоприводов.

Интерполятор

Интерполяция производится следующим образом:

1. Вычисляются заданные положения и скорости осей исходя из заданных ограничений траекторной скорости, траекторного ускорения и ограничений скоростей и ускорений по осям.
2. С помощью обратного кинематического преобразования вычисляются задания положения для сервоприводов.
3. Выполняется сплайн-интерполяция по 4 точкам положения.

Типы интерполяции

- Линейная интерполяция;
- Круговая интерполяция.

Линейная интерполяция предназначена для выполнения линейных движений одной или более декартовыми осями.

При линейной интерполяции задания положения для всех осей формируются таким образом, чтобы все декартовы оси завершили перемещение одновременно.

Круговая интерполяция предназначена для выполнения круговых или винтовых движений в системе координат XYZ. Круговая интерполяция доступна в одной из плоскостей XY, XZ, YZ. При круговой интерполяции оси A, B, C перемещаются таким образом, чтобы завершить перемещение одновременно с завершением перемещения остальными осями.

Режим прохождения траектории

СЧПУ может выполнять перемещения в двух режимах.

Режим 1. Режим точного прохождения. В этом режиме СЧПУ точно следует заданной траектории движения, что требует полного останова в конечных точках неколлинеарных участков траектории. Зависимость профиля скорости движения от траектории движения в данном режиме показана на Рис. 9.

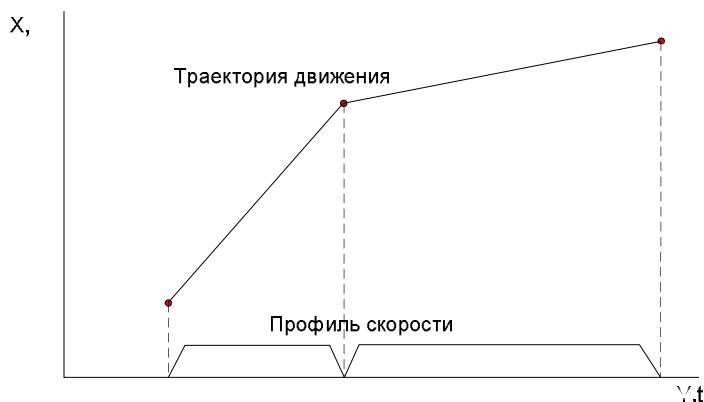


Рис. 9. Режим точного прохождения.

Режим 2. Режим поддержания скорости. В этом режиме СЧПУ выполняет движения таким образом, чтобы сохранить постоянной скорость движения инструмента. В данном режиме доступно два вида прохождения траектории:

Режим 2.1. Режим с неограниченным отклонением, показан на Рис. 10. В данном режиме скорость передвижения ограничивается только длиной сегментов;

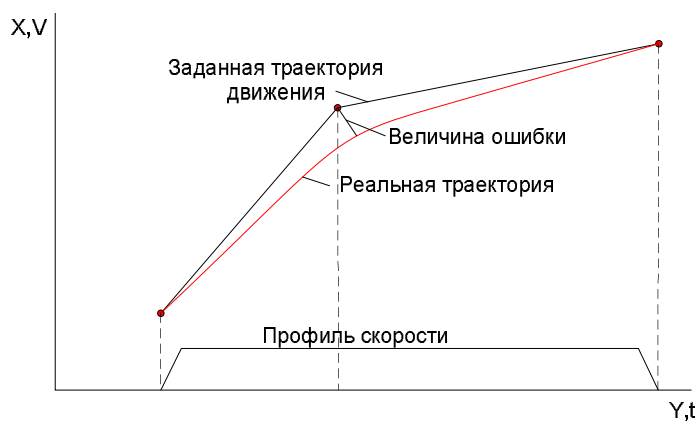


Рис. 10. Режим с неограниченным отклонением.

Режим 2.2. Режим с ограниченным отклонением. В данном режиме скорость прохождения траектории уменьшается таким образом, чтобы траектория прохождения отличалась от заданной не более, чем на указанную величину. Зависимость траектории движения и величины ошибки от профиля скорости показана на Рис. 11.

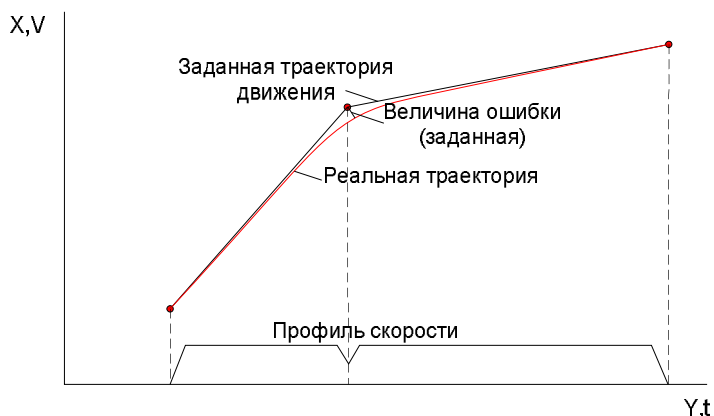


Рис. 11. Режим с ограниченным отклонением.

Кроме того, в данном режиме выполняется преобразование траектории по следующим правилам:

1. Если два последовательных линейных сегмента отличаются от прямой, проходящей через начало первого сегмента и конец второго, не более чем на заданный допуск, то они заменяются одним сегментом.
2. Если дуга отличается от прямой не более чем на заданный допуск, то она заменяется на два линейных сегмента: от начала к середине дуги и от середины к концу.

Функция Look-ahead

В СЧПУ реализована функция Look-ahead, позволяющая на основе информации о траектории движения сформировать наиболее быстрый профиль скорости, возможный без превышения ограничений скоростей и ускорений, заданных в системе. Данная функция активируется при установке в программе движения режима поддержания скорости.

Привод

Привод (joint) представляет собой физическую степень свободы станка. СЧПУ в процессе отработки программы движения генерирует задание по позиции для сервоприводов в зависимости от типа кинематики. Также СЧПУ отслеживает ошибки сервоприводов, рассогласование между заданием и текущим положением и программные и аппаратные ограничения положения.

Табл. 12. Группа motmod.Joint.<n>.

Имя	Направление / Параметр	Тип	По умолчанию	Описание
fault	In	Bool	False	Вход ошибки сервопривода
enable	Out	Bool		Выход разрешения работы сервопривода
pos-cmd	Out	Float		Задание положения
vel-cmd	Out	Float		Задание по скорости (форсирующая связь)
pos-fb	In	Float	0.0	Вход обратной связи по позиции
minlim	In	Bool	False	Вход аппаратного ограничителя в отрицательном направлении
maxlim	In	Bool	False	Вход аппаратного ограничителя в положительном направлении
home	In	Bool	False	Вход датчика нулевого положения
ndex	In	Index		Сигнал поиска Z-метки
JogCommand	In	S32	0	Команда ручного управления
JogEnable	In	Bool	False	Разрешение ручного управления
MaxVelocity	Param	Float	0.0	Максимальная допустимая скорость [мм/с]
MaxAcceleration	Param	Float	0.0	Максимальное допустимое ускорение [мм/с ²]
MinLimit	Param	Float	-10000.0	Минимальное программное ограничение положения
MaxLimit	Param	Float	10000.0	Максимальное программное ограничение положения
FollError	Param	Float	1.0	Максимальная ошибка слежения
Backlash	Param	Float	0.0	Величина люфта [мм]
JogScale	Param	Float	0.001	Коэффициент масштабирования ручного управления [мм/инк]

Система координат сервопривода

Каждый сервопривод имеет свою систему координат. Смещение нулевого положения устанавливается после выполнения процедуры поиска нулевой точки. Далее при расчёте прямого и обратного кинематического преобразования текущее положение в координатах сервопривода смещается на величину смещения.

Сигналы ошибки и аппаратных ограничений положения

В сервоприводе предусмотрены сигналы аппаратной ошибки и аппаратных ограничений положения. При активации сигнала ошибки сервопривода СЧПУ переходит в со-

стояние сброса. При активации сигнала аппаратного ограничения положения СЧПУ переходит в состояние сброса и запрещает перемещение сервопривода в соответствующем направлении до момента деактивации сигнала. В СЧПУ предусмотрен режим игнорирования сигналов аппаратных ограничений положения, при котором активация сигналов не вызовет перехода в состояние сброса, но запрет на движение в направлении сработавшего сигнала сохраняется.

Задание, обратная связь и ошибка слежения

Каждый цикл управления СЧПУ захватывает текущее положение `pos-fb`. Разница между текущим и заданным положением сервопривода называется ошибкой слежения. Большая величина ошибки слежения свидетельствует о том, что исполнительное устройство не может выполнить задание движения от СЧПУ или параметры привода не настроены должным образом. При превышении ошибкой слежения уровня параметра `FollError` СЧПУ переходит в состояние сброса.

Заданное положение `pos-req` рассчитывается исходя из программы движения или команды ручного управления с учётом ограничений скоростей и ускорений сервоприводов.

Поиск нулевой точки

При использовании относительных датчиков положения при выключении сервоприводов и/или СЧПУ теряются данные о положении сервоприводов в машинной системе координат. Для восстановления этих данных необходимо выполнить процедуру поиска нулевой точки. Данная процедура использует сигнал о достижении сервоприводом предопределённого положения в машинной системе координат, после чего соответствующим образом выставляется смещение нулевого положения сервопривода.

На Рис. 12 приведен алгоритм поиска нулевой точки оси.

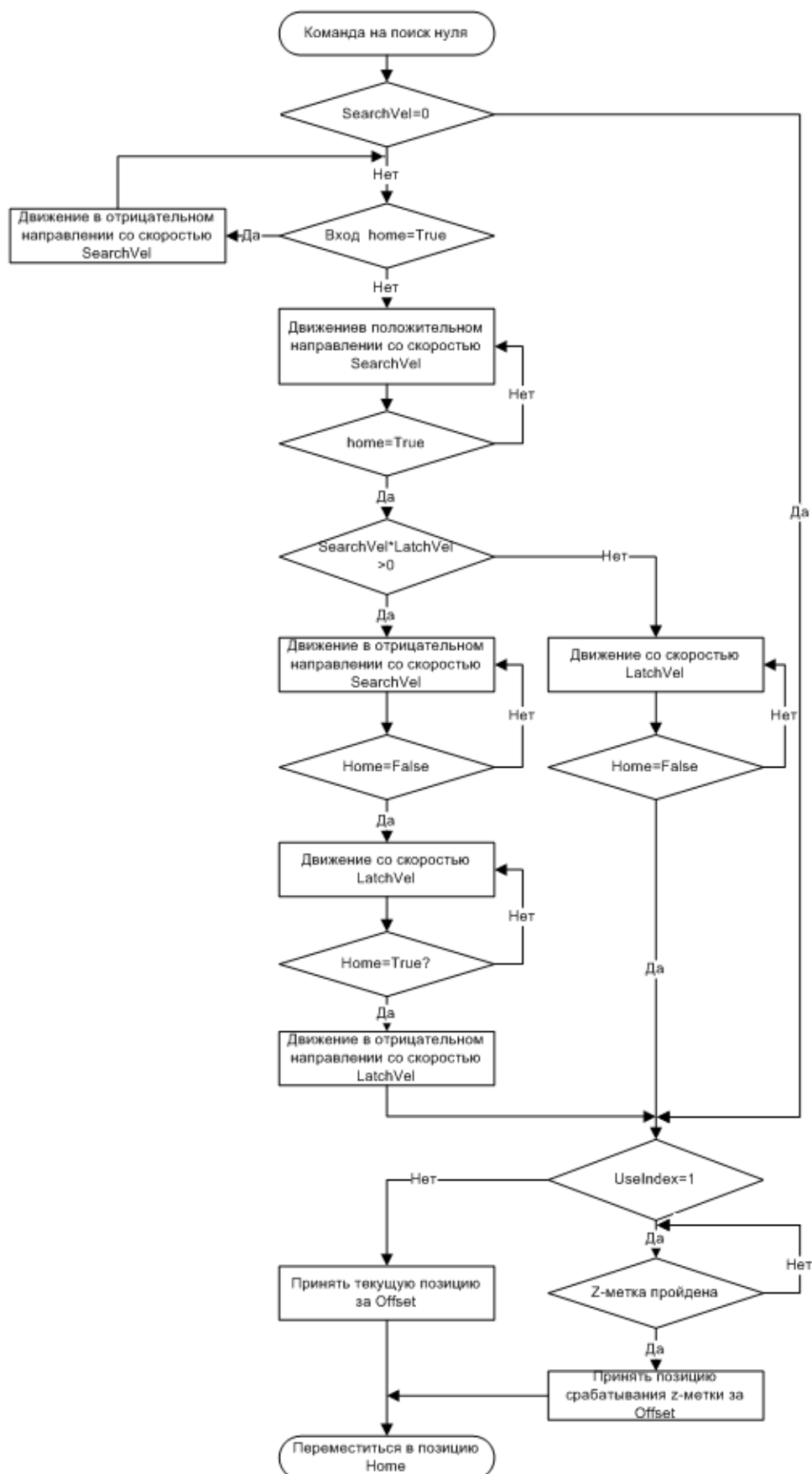


Рис. 12. Алгоритм поиска нулевой точки оси.

В алгоритме встречаются следующие переменные:

1. SearchVel - Скорость выхода в нулевую точку. Величина скорости SearchVel выбирается исходя из минимизации времени выполнения алгоритма с одной стороны. С другой стороны SearchVel и конфигурация станка должна быть таковой, чтобы привод успел остановить ось до достижения упора оси.
2. LatchVel – Скорость фиксации нуля. Величина данной скорости, как правило, на много меньше скорости SearchVel, чтобы обеспечить высокую точность фиксации нулевой позиции, что достаточно важно особенно при отсутствии дополнительного этапа выхода в ноль с помощью Z метки.
3. home – Состояние датчика нулевой позиции оси. В частном случае в качестве датчика может выступать концевой выключатель оси. Однако в качестве аргументов в пользу использования выделенного датчика нулевой позиции выступает тот факт, что в этом случае увеличивается безопасность выхода в ноль и скорость SearchVel, т.к. датчик Home располагается дальше от упора, чем концевой выключатель.
4. UseIndex – параметр активирующий точный поиск нуля с помощью Z метки.
5. Offset – Смещение по оси.
6. Home – положение после выхода в 0. Данный параметр используется в том случае, если в качестве датчика нулевого положения используется концевой выключатель оси, так как в этом случае положение датчика нулевого положения должно находиться вне программных ограничителей положения оси.

Первое действие, которое выполняется в алгоритме, это проверка на равенство нулю скорости выхода в ноль SearchVel. Далее выполняется проверка - находится ли ось уже на датчике нулевой позиции (home). Если алгоритм запускается на выполнение при активном датчике нуля, то выполняется съезд с него на скорости поиска нуля SearchVel.

Следующим шагом следует непосредственно выход в ноль на скорости SearchVel, который завершается активацией датчика home.

Далее существуют два варианта выхода в ноль:

1. Если направление скорости SearchVel и LatchVel совпадают, то СЧПУ выполнит повторный съезд с home на скорости SearchVel и далее выполнит повторный наезд на него на скорости LatchVel. В этом случае фиксация нулевой позиции выполнится в момент активации датчика home.

2. Если направление скорости SearchVel и LatchVel не совпадают, то СЧПУ выполнит съезд с home на скорости LatchVel. В этом случае фиксация нулевой позиции выполнится в момент деактивации датчика Home.

Предпочтение одному или другому способу можно отдать исходя из сравнения точности механизма активации и деактивации используемого датчика Home.

В завершении, при наличии соответствующих настроек, выполняется выход в ноль с помощью Z метки энкодера привода. При выполнении данной процедуры точность выхода в ноль существенно увеличивается в большинстве случаев.

Для инициализации всей машины, необходимо, чтобы все оси выполнили алгоритм поиска нулевой позиции. Существует возможность настройки последовательности выхода в ноль по осям. Например, можно выполнить последовательный выход в ноль по осям. Это способ занимает наибольшее количество времени. Другой способ выхода в – это одновременный выход в ноль всех осей. Данный способ наиболее быстрый, в тоже время для некоторых конфигураций станка является опасным, т.к. существует вероятность повреждения механики станка рабочим органом.

Наиболее распространенной последовательностью выхода в ноль для трехкоординатного станка является двухцикловый. В первом цикле выполняется поиск нуля оси Z, которая отводит рабочий орган в безопасное положение, далее выполняется одновременный поиск по осям X и Y.

В приведенном на Рис. 12 алгоритме внесены некоторые упрощения. В частности, не показана взаимная синхронизация осей при выполнении поиска нулевых позиций. Например, в случае, если в одном цикле поиска задействованы несколько осей, то по завершении поиска ось ожидает завершения поиска по всем остальным осям в данном цикле.

Программные ограничения положения

В конфигурации СЧПУ могут быть заданы программные ограничения положения. Эти ограничения задействуются после выполнения процедуры поиска нулевой точки. СЧПУ не принимает команды, которые приводят к перемещению сервоприводов за программные ограничения положения.

Ограничения скорости и ускорения

Для каждого из сервоприводов могут быть установлены индивидуальные ограничения скорости и ускорения. При выполнении команд задания всех сервоприводов корректируются таким образом, чтобы скорости и ускорения каждого из них не превышали ограничений.

Компенсация люфта

СЧПУ может выполнять компенсацию люфта в механических компонентах станка. Для этого СЧПУ отслеживает текущее направление движения сервопривода и при его изменении прибавляет к заданию величину люфта.

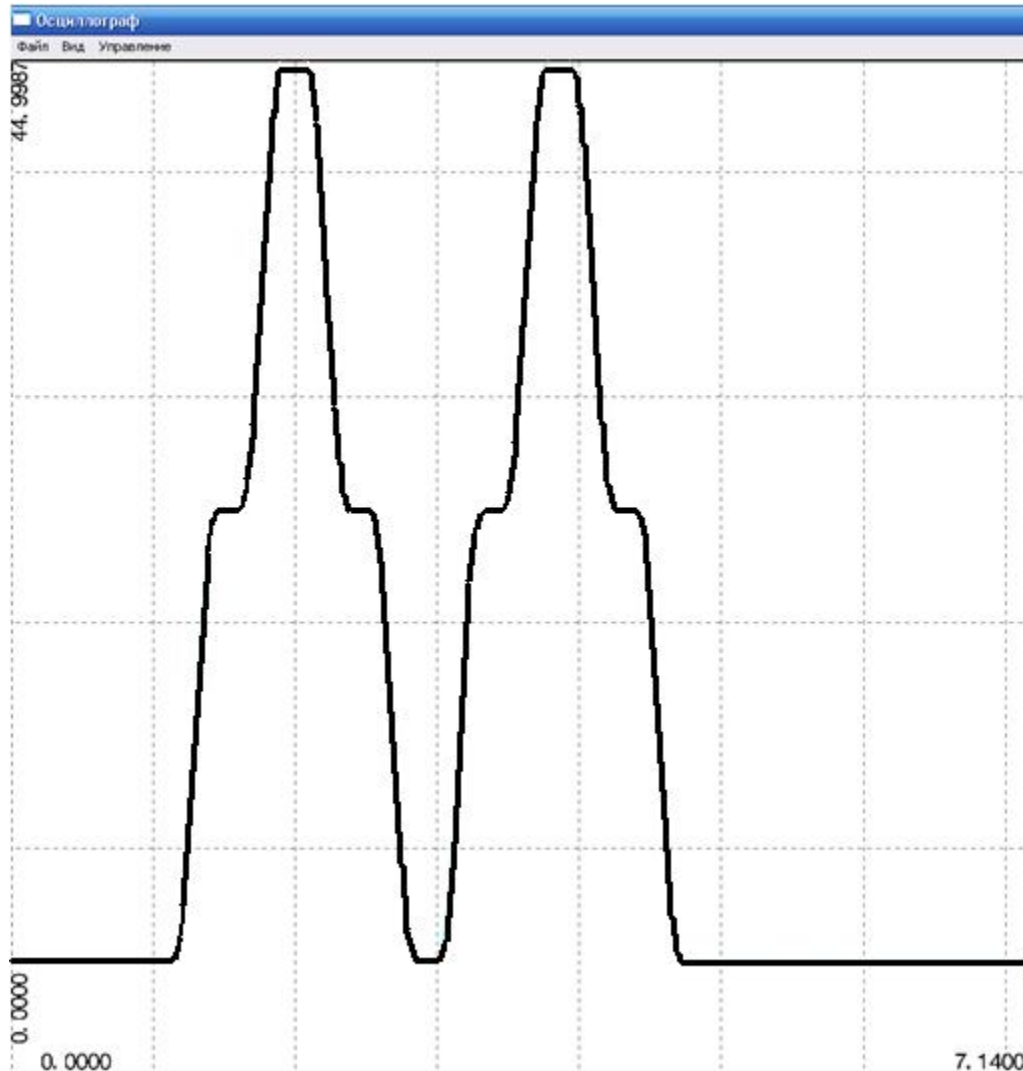


Рис. 13. График позиции при выполнении двух движений в положительном и отрицательном направлении без использования алгоритма компенсации люфта.

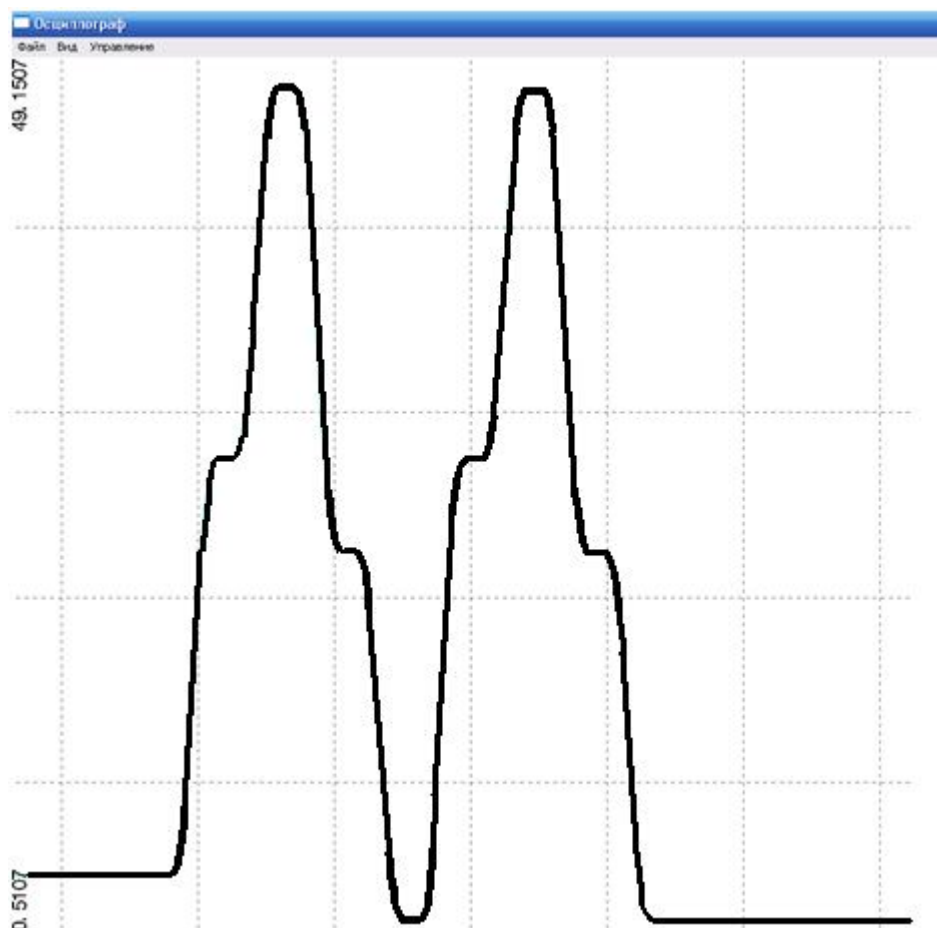


Рис. 14. График позиции при выполнении двух движений в положительном и отрицательном направлении при использовании алгоритма компенсации люфта.

На рисунках Рис. 13, Рис. 14 приведены графики задания позиции при выполнении двух движений в положительном направлении на 20 мм а затем двух движений в отрицательном направлении на 20 мм с величиной люфта 0 и 5 мм.

2.3.6.2. Шпиндель

Шпиндель предназначен для вращения инструмента или заготовки с постоянной скоростью, задаваемой в программе движения. В токарных станках перемещение осей может быть синхронизировано со скоростью вращения шпинделя.

Табл. 13. Логические ресурсы шпинделя motmod.AUX.Spindle

Имя	Направление	Тип	По умолчанию	Описание
Pos-fb	In	Float	0.0	Текущее положение шпинделя
Speed-cmd	Out	Float		Заданная скорость вращения шпинделя
Speed-fb	In	Float	0.0	Текущая скорость вращения шпинделя
On	Out	Bool		Сигнал разрешения работы шпинделя
Forward	Out	Bool		Команда вращения шпинделя вперед
Reverse	Out	Bool		Команда вращения шпинделя назад
Index	In	Index		Вход z-метки шпинделя

При активации функции вращения шпинделя вперёд и ненулевой заданной скорости вращения шпинделя СЧПУ активирует выход AUX.Spindle-forward. При активации функции вращения шпинделя назад и ненулевой заданной скорости вращения шпинделя СЧПУ активирует выход AUX.Spindle-reverse. При деактивации функций вращения шпинделя или установке нулевой скорости шпинделя СЧПУ деактивирует выходы AUX.Spindle-forward и AUX.Spindle-reverse.

2.3.6.3. Программно-управляемые выходы

При выполнении пользовательских программ движения СЧПУ СервоКон[©] может дополнительно управлять значениями 4 дискретных и 4 аналоговых выходов.

Табл. 14. Группа motmod.IO.

Имя	Направление	Тип	По умолчанию	Описание
IO.a_out.<n>	Out	Float	0.0	Программно управляемый аналоговый выход (M67 P<n>)
IO.d_out.<n>	In	Bool	0	Программно управляемый дискретный выход (M64 P<n>, M65 P<n>)

При выполнении функций установки/сброса программируемых дискретных выходов СЧПУ активирует/деактивирует выходы IO.d_out.<n>.

При выполнении функции установки значения программируемых аналоговых выходов СЧПУ устанавливает значения выходов IO.a_out.<n>.



ВНИМАНИЕ: Состояние программируемых выходов не меняется автоматически при нажатии клавиш «Пуск» и «Стоп» и не восстанавливается при использовании функции останова с продолжением.

2.3.6.4. СОЖ

Табл. 15. motmod.AUX.Misc

Имя	Направление	Тип	По умолчанию	Описание
Flood-on	Out	Bool	False	Команда включения охлаждения (M8)
Mist-on	Out	Bool	False	Команда включения охлаждения (M7)

При активации функции включения охлаждения типа 1 СЧПУ активирует выход AUX.mist-on. При активации функции включения охлаждения типа 2 СЧПУ активирует выход AUX.flood-on. При активации функции выключения охлаждения СЧПУ деактивирует выходы AUX.mist-on и AUX.flood-on.

2.3.6.5. Сменщик инструментов

В СЧПУ СервоКон[©] реализована поддержка сменщика инструмента, позволяющая в процессе выполнения программы заменить текущий инструмент вручную или автоматически.

Табл. 16. motmod.AUX.ToolChange

Имя	Направление	Тип	По умолчанию	Описание
change	Out	Bool	False	Команда на смену инструмента
changed	In	Bool	False	Сигнал об окончании смены инструмента
prepare	Out	Bool	False	Команда на подготовку к смене инструмента
prepared	In	Bool	False	Сигнал о готовности к смене инструмента
Tool-number	Out	S32	0	Номер нужного инструмента

При выполнении команды выбора инструмента (T<n>) СЧПУ активирует выход AUX.ToolChange.prepare и выставляет на выход AUX.ToolChange.tool-number номер выбранного инструмента.

При активации функции смены инструмента (M6) СЧПУ ожидает установки сигнала AUX.ToolChange.prepared, затем активирует выход AUX.ToolChange.change и ожидает установки сигнала AUX.ToolChange.changed.

3. Условия выполнения программы СервоМастер[©]

3.1. Условия, необходимые для выполнения программы

Для работы с программой необходимы технические специалисты, владеющие опытом работы с программными продуктами подобного типа, изучившие данное «Руководство оператора СервоМастер[©]» и «Руководство по эксплуатации СЧПУ СервоКон[©]».

3.2. Состав программных средств

Программное обеспечение предназначено для работы с операционными системами Windows[®] XP (32 bit), Windows Vista[®] (32 bit), Windows[®] 7 (32bit).

3.3. Состав технических средств

Технические средства (персональные компьютеры) должны отвечать требованиям, предъявляемым к ним операционными системами, которые используются для работы СервоМастер[©]. Для работы с программным осциллографом рекомендуется наличие видеоадаптера, поддерживающего интерфейс OpenGL 2.0.

Для анализа и отладки конфигурации СЧПУ по сети компьютеры должны быть оснащены сетевыми адаптерами.

4. Выполнение программы СервоМастер[©]

4.1. Установка программы

Описание процесса загрузки и установки СервоМастер[©].

1. Скачайте дистрибутив программы с сайта <http://www.servotechnica.ru/> Дистрибутив располагается в разделе «Документация» – «ЧПУ СервоКон» – «Программное обеспечение».
2. Дистрибутив представляет собой набор устанавливаемых компонентов, собранных в архив. Разархивируйте архив программы архиватором.
3. Запустите процесс установки программы с помощью файла setup.exe.



ВНИМАНИЕ: Если на компьютере установлена старая версия программы СервоМастер[©], необходимо деинсталлировать ее через панель управления – «Установка и удаление программ» до начала установки новой версии программы.

Для некоторых антивирусных программ понадобится отключить антивирусную защиту на время установки программы СервоМастер[©].

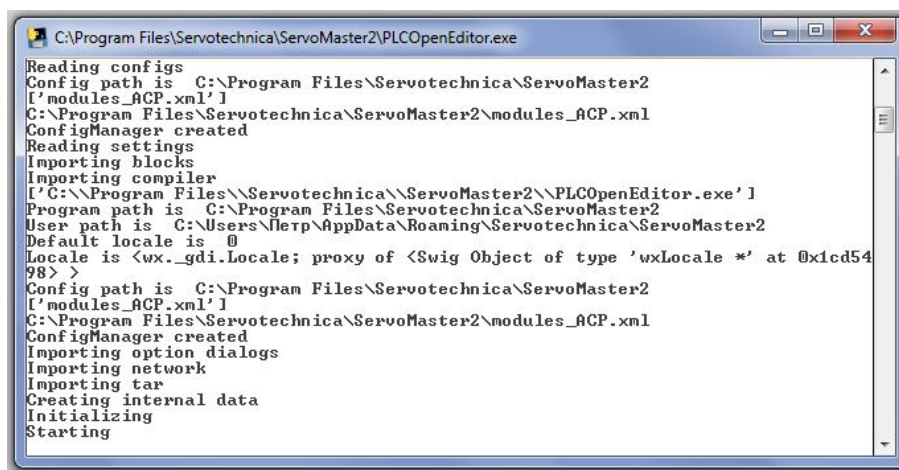
4. Следуйте указаниям мастера установки.

4.2. Запуск

4.2.1. Описание запуска программы. Начало работы

Запустите программу СервоМастер[©] с помощью иконки ServoMaster2 на рабочем столе или в меню «Программы» - «Сервотехника» - «СервоМастер», или с помощью файла PLCOpenEditor.exe, расположенного в папке с установленной программой.

Запуск программы начинается с загрузки консольного окна, которое должно быть открыто на всем протяжении работы с программой СервоМастер[©].



```
C:\Program Files\Servotechnica\ServoMaster2\PLCOpenEditor.exe
Reading configs
Config path is C:\Program Files\Servotechnica\ServoMaster2
['modules_ACP.xml']
C:\Program Files\Servotechnica\ServoMaster2\modules_ACP.xml
ConfigManager created
Reading settings
Importing blocks
Importing compiler
['C:\Program Files\Servotechnica\ServoMaster2\PLCOpenEditor.exe']
Program path is C:\Program Files\Servotechnica\ServoMaster2
User path is C:\Users\Петр\AppData\Roaming\Servotechnica\ServoMaster2
Default locale is 0
Locale is <wx._gdi.Locale; proxy of <Swig Object of type 'wxLocale *' at 0x1cd5498> >
Config path is C:\Program Files\Servotechnica\ServoMaster2
['modules_ACP.xml']
C:\Program Files\Servotechnica\ServoMaster2\modules_ACP.xml
ConfigManager created
Importing option dialogs
Importing network
Importing tar
Creating internal data
Initializing
Starting
```

Рис. 15. Общий вид консоли.



ВНИМАНИЕ: Не закрывайте консольное окно, т.к. это приведет к прекращению работы и закрытию программы СервоМастер[©].

Для некоторых антивирусных программ необходимо настроить антивирусную программу таким образом, чтобы она игнорировала запуск программы СервоМастер[©], первоначально идентифицируя ее как полученную из ненадежного источника, открывая программу в т.н. «песочнице», постоянно осуществляя проверку и анализ данного программного обеспечения.

После загрузки консольного окна произойдет загрузка основного рабочего поля программы СервоМастер[©].

Рабочее пространство программы поделено на несколько рабочих и информационных зон:

- Главное меню;
- Поле панели инструментов (на данном этапе не активное);
- Поле программы;
- Окно Проект;
- Окно Библиотека.

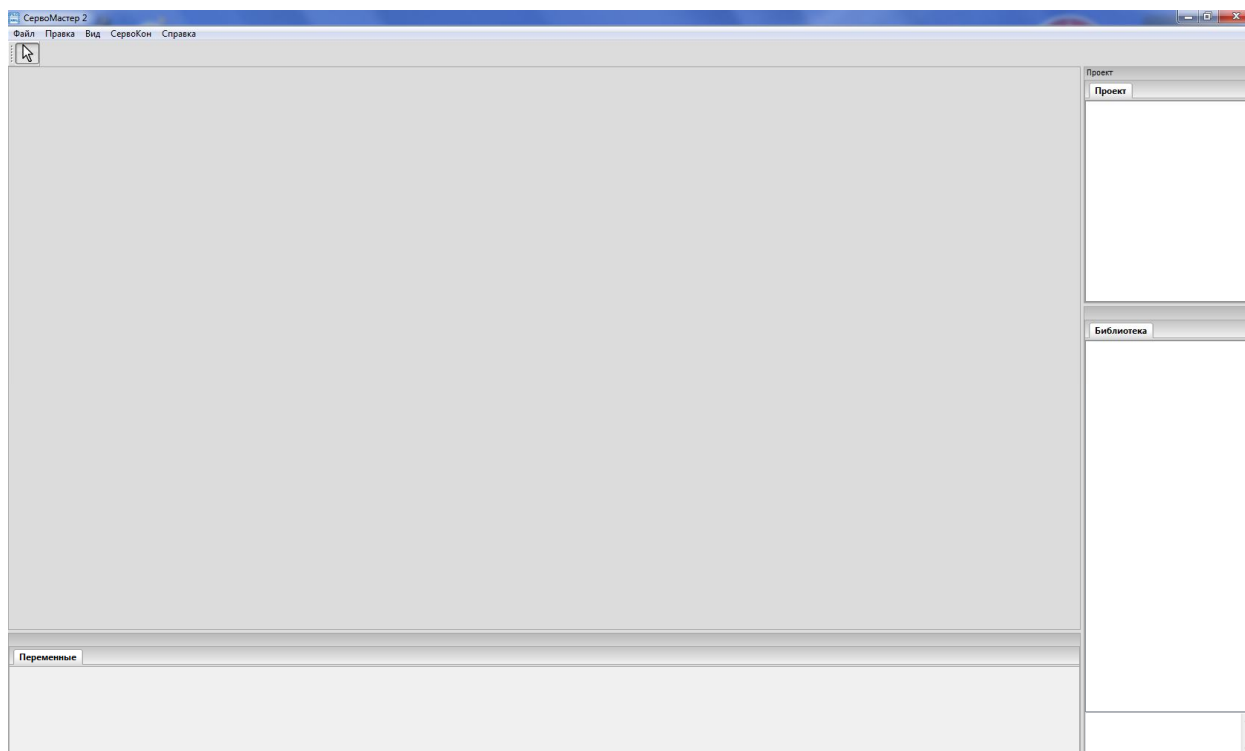


Рис. 16. Рабочее пространство программы.

Для появления рабочего поля программы, дерева проекта, дерева библиотеки блоков необходимо создать новую конфигурацию или загрузить уже созданную для редактирования.

4.2.2. Создание или загрузка конфигурации

4.2.2.1. Создание конфигурации

В главном меню выберите «Файл» - «Новый», в появившемся окне введите имя конфигурации и выберите комплектацию СЧПУ, для которой создается конфигурация, нажатием кнопки мыши и нажмите «ОК».

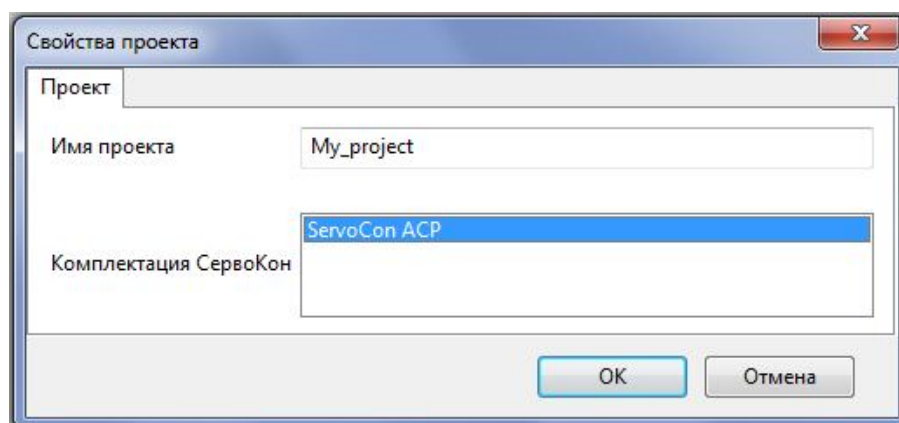


Рис. 17. Окно «Свойства проекта».

4.2.2.2. Загрузка сохраненной конфигурации

В главном меню выберите «Файл» – «Открыть» и выберите нужный файл конфигурации.

В случае если вы установили обновленное описание своей комплектации и пытаетесь открыть конфигурацию, созданную для более старого описания комплектации, то программа предложит выбор: работать со старым описанием или использовать новое. В случае использования нового описания обязательно проверьте конфигурацию на правильность. Рекомендуется использовать новое описание для старых конфигураций только в том случае, если в нём появились необходимые вам функции.

4.2.2.3. Сохранение конфигурации

Для сохранения текущего проекта без создания конфигурации в главном меню выберите «Файл» – «Сохранить» и введите нужное имя файла. При этом будет создан файл проекта (с расширением scrproj). Файл проекта не может быть загружен в СЧПУ.

4.2.2.4. Работа с проектом

В дереве проекта правой кнопкой мыши щелкните на пункт «Программы», в контекстном меню выберите «Добавить программу», в появившемся окне введите имя программы.

Программа появится в дереве проекта в разделе программ. Двойным щелчком мыши откройте ее для редактирования.



ВНИМАНИЕ: Имя программы может состоять из букв, цифр и символа «_», но обязательно должно начинаться с буквы.

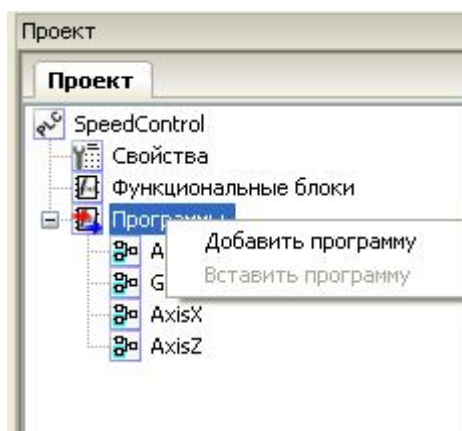


Рис. 18. Дерево «Проект».

После добавления и открытия программы появится рабочее поле программы и меню с панелью инструментов.

Другие элементы дерева «Проект»:

Свойства – выводит окно с именем проекта и комплектацией СЧПУ;

Функциональные блоки – в данный момент не используется.

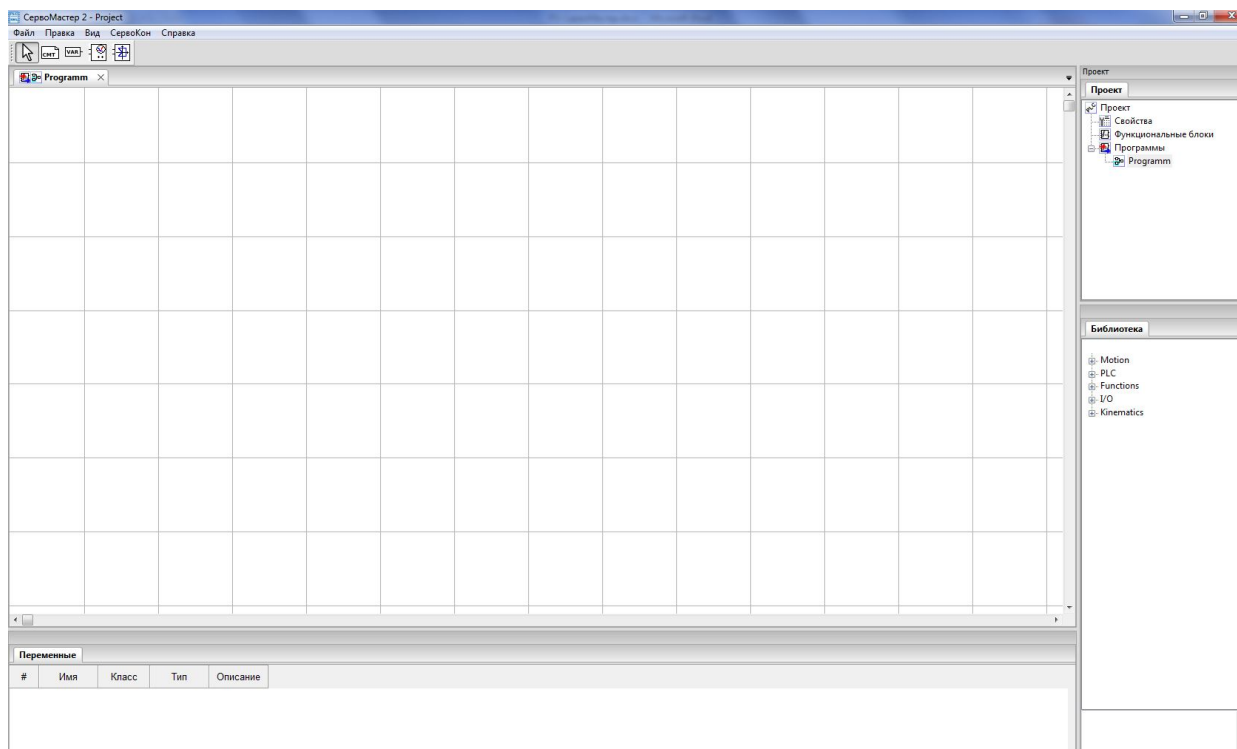


Рис. 19. Рабочее поле программы СервоМастер[®].

Строка меню панели инструментов содержит элементы в виде активных иконок – «Выбрать объект», «Создать комментарий», «Текущее значение выхода», «Создать осциллограф».

4.2.3. Главное меню

Главное меню состоит из пяти основных разделов: «Файл», «Правка», «Вид», «СервоКон», «Справка».

4.2.3.1. Файл

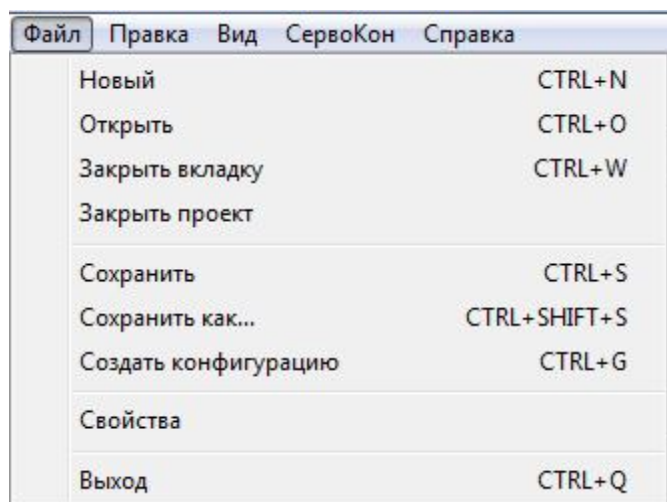


Рис. 20. Раздел главного меню «Файл».

Меню «Файл» предназначено для операций с создаваемыми файлами и проектами. Меню «Файл» состоит из стандартного набора функций и пункта «Создать конфигурацию».

4.2.3.2. Правка

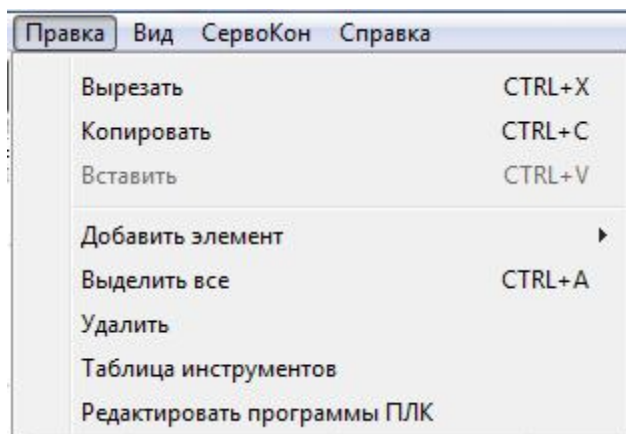


Рис. 21. Раздел главного меню «Правка».

Меню «Правка» состоит из стандартного набора функций, за исключением специальных функций:

«Добавить элемент» – используется для добавления программ;

«Таблица инструментов» - используется для редактирования таблицы инструментов СЧПУ;

«Редактировать программы ПЛК» – используется для создания и редактирования программ ПЛК.

4.2.3.3. Вид

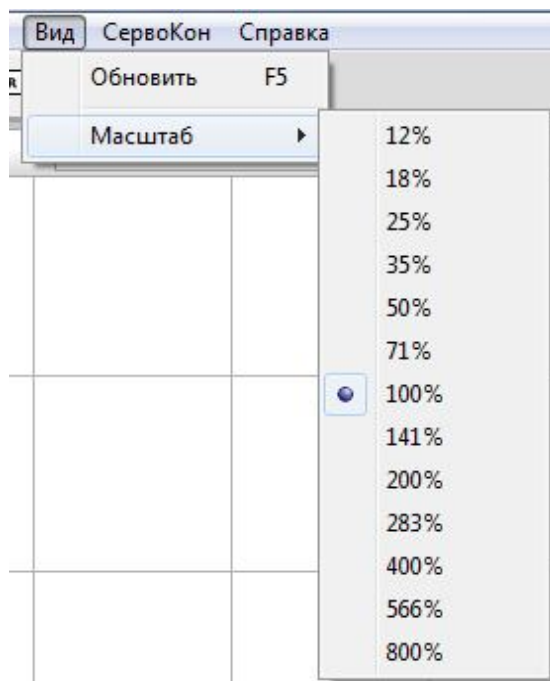


Рис. 22. Раздел главного меню «Вид».

Меню «Вид» для обновления рабочего поля программы и для масштабирования.

4.2.3.4. СервоКон

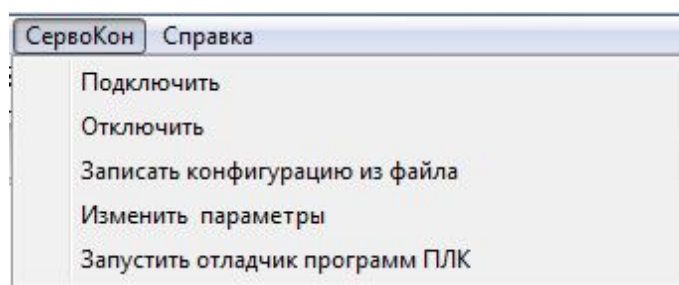


Рис. 23. Раздел главного меню «СервоКон».

Меню «СервоКон» для подключения и отключения СЧПУ СервоКон[©], записи в СЧПУ СервоКон[©] конфигурации, изменения параметров СЧПУ СервоКон[©], запуска отладчика программ ПЛК.

4.2.3.5. Справка

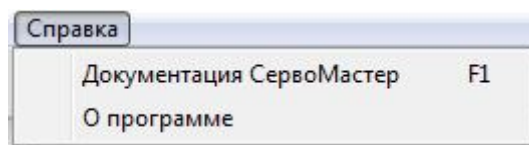


Рис. 24. Раздел главного меню «Справка».

Меню «Справка» состоит из справочной документации и информации о программе СервоМастер[®].

4.2.4. Библиотека блоков

В библиотеке блоков 5 основных разделов:

- Motion – блоки, представляющие контроллер движения, сюда входят блоки, отвечающие за движение сервоприводов, управление шпинделем, подачей СОЖ, сменщиком инструментов и др.
- PLC – блоки, представляющие программы ПЛК. Если программы ПЛК не были загружены, может отсутствовать;
- Functions – блоки, предоставляющие математические и логические функции, применяющиеся для создания конфигурации, например, такие как логическое «не», логическое «или», сумматор, логический мультиплексор, задание постоянной величины, ПИД-регулятор, генератор сигналов и др.;
- I/O – блоки, представляющие аппаратные ресурсы ввода/вывода такие как CAN, платы ввода/вывода, ресурсы передней панели;
- Kinematics – блоки, представляющие типы кинематических преобразований.

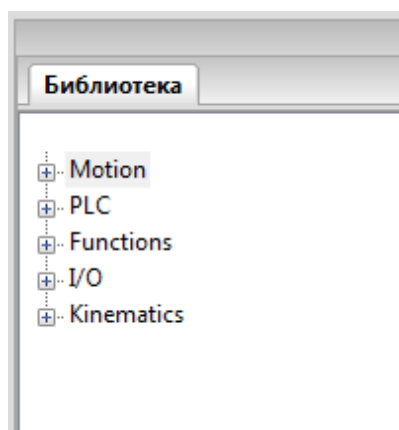


Рис. 25. Библиотека блоков. Основные разделы.

4.2.4.1. Раздел библиотеки Motion

motmod Контроллер движения



Рис. 26. Раздел motmod.

kin – Тип кинематики.

Enable – Вход разрешения работы.

Pause – Вход внешнего сигнала паузы.

Probe – Вход датчика контакта.

FeedHold – Запрет на перемещение.

MaxVelocity – Максимальная траекторная скорость движения.

MaxAcceleration – Максимальное траекторное ускорение.

QueueDepth – Количество кадров в очереди контроллера движения.

LoaderQueueDepth – Количество кадров в очереди загрузчика.

ControllerTime – Время выполнения контроллера движения.

CommandHandlerTime – Время обработки команд контроллером движения.

AUX - Дополнительные функции, состоит из:

- Misc – Управление СОЖ.
- ToolChanger – Управление сменщиком инструментов.
- Spindle – Управление шпинделем.

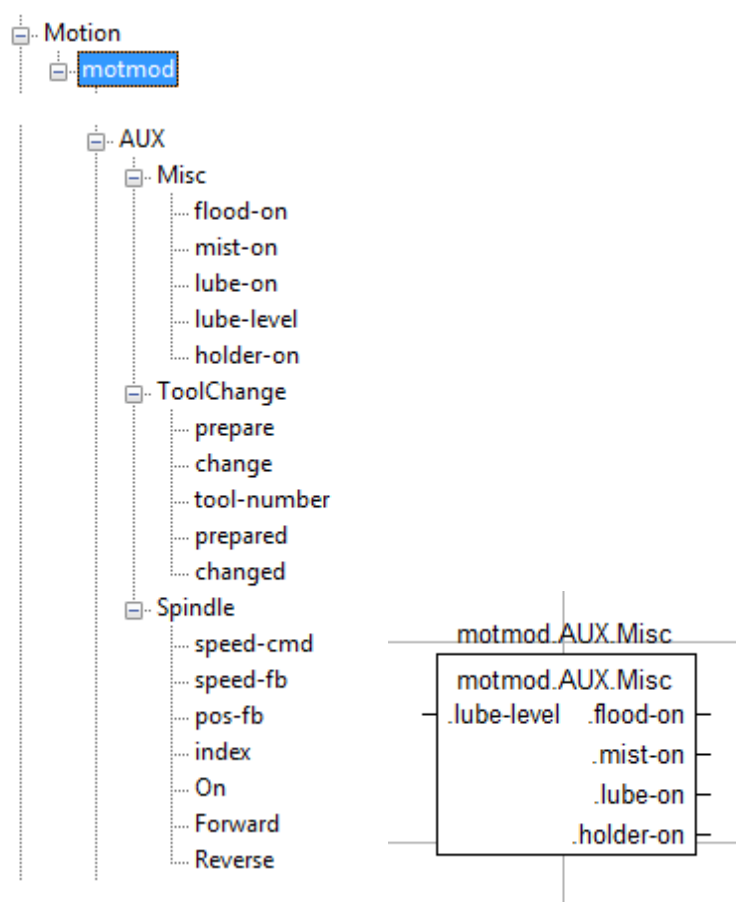


Рис. 27. Раздел motmod.AUX.Misc – управление СОЖ.

lube-level – Уровень смазки.

flood-on – Включение жидкостного охлаждения.

mist-on – Включение охлаждения.

lube-on – Включение смазки.

holder-on – Включение зажима заготовки.

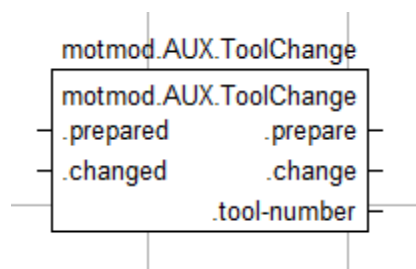


Рис. 28. Раздел motmod.AUX.ToolChanger – Управление сменщиком инструментов.

prepare – Подготовка инструмента.

change – Смена инструмента.

tool-number – Номер инструмента.

prepared – Инструмент подготовлен.

changed – Смена инструмента закончена.

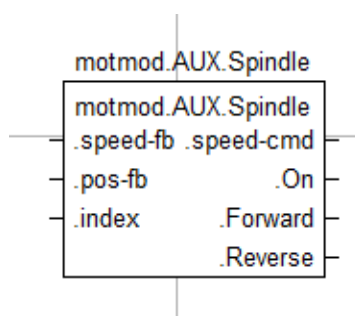


Рис. 29. Раздел motmod.AUX.Spindle – Управление шпинделем.

speed-cmd – Задание скорости шпинделя.

speed-fb – Текущая скорость шпинделя.

pos-fb – Текущее положение шпинделя.

index – z-метка шпинделя.

On – Команда включения шпинделя.

Forward – Команда вращения шпинделя вперед.

Reverse – Команда вращения шпинделя назад.

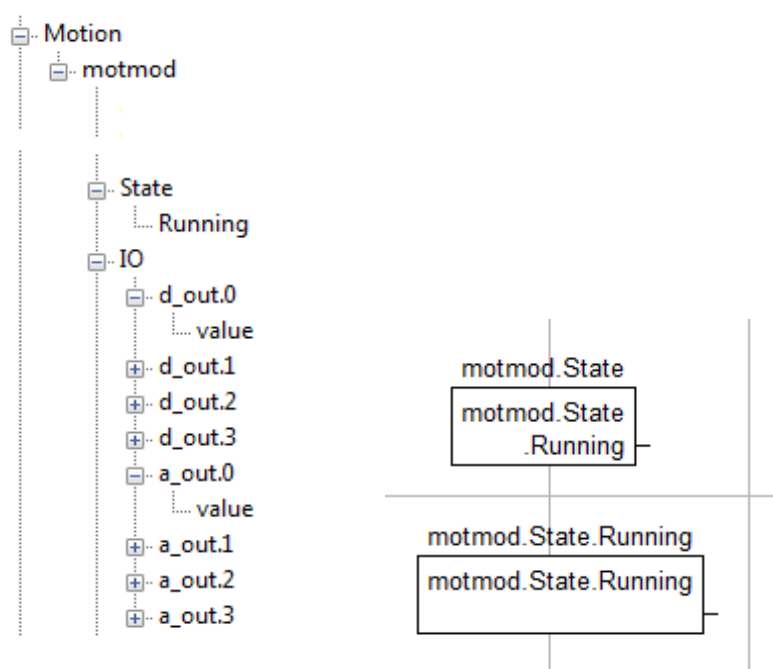


Рис. 30. Раздел motmod.AUX. State - Состояние СЧПУ.

Running – Выполняется ли программа.

IO – Программируемый ввод-вывод.

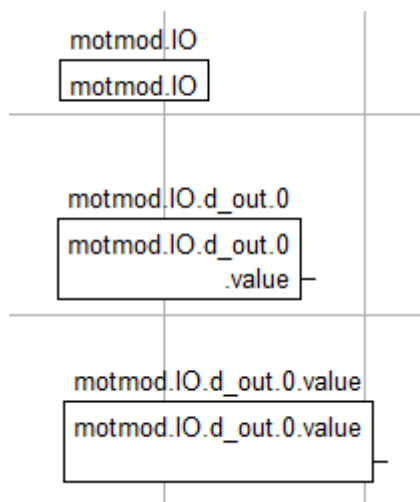


Рис. 31. Блок motmod.IO.d_out.<n> – Дискретные выходы общего назначения.

value – Дискретный выход.

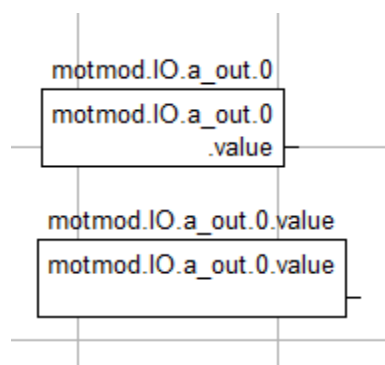


Рис. 32. Блок motmod.IO.d a_out.<n> – Аналоговые выходы общего назначения.

value – Аналоговый выход.

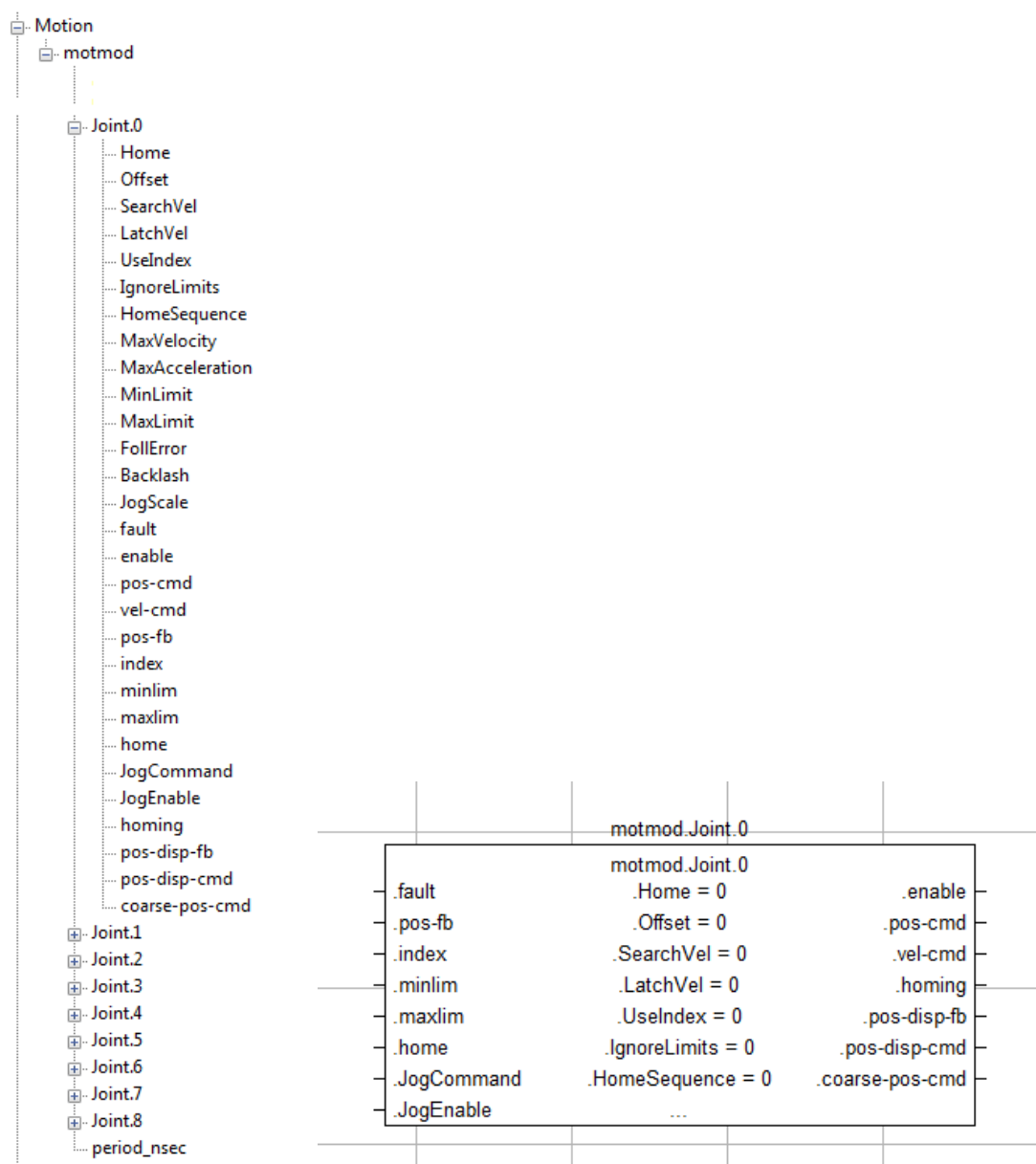


Рис. 33. Раздел motmod.Joint.<n> - Приводы.

Home – Положение после выхода в нуль.

Offset – Смещение нуля привода от датчика нулевого положения.

SearchVel – Скорость поиска датчика нулевого положения.

LatchVel – Скорость съезда с датчика нулевого положения.

UseIndex – Выполнять поиск z-метки двигателя.

IgnoreLimits – Игнорировать концевые выключатели - установите в TRUE если концевой датчик используется как концевой выключатель.

HomeSequence – Номер цикла поиска нуля.

MaxVelocity – Максимальная скорость.

MaxAcceleration – Максимальное ускорение.

MinLimit – Ограничение минимального положения.

MaxLimit – Ограничение максимального положения.

FollError – Максимальное отклонение от траектории.

Backlash – Величина люфта.

JogScale – Перемещение на одну единицу входа ручного управления (мм).

fault – Вход ошибки привода.

enable – Разрешение работы.

pos-cmd – Задание положение.

vel-cmd – Форсирующая связь по скорости.

pos-fb – Текущее положение.

index – Вход сигнала Z-метки.

minlim – Вход концевого выключателя в отрицательном направлении.

maxlim – Вход концевого выключателя в положительном направлении.

home – Вход датчика нулевого положения.

JogCommand – Вход ручного управления.

JogEnable – Разрешение ручного управления.

homing – 1 если выполняется выход в нуль.

pos-disp-fb – Текущее положение относительно нуля станка.

pos-disp-cmd – Заданное положение относительно нуля станка.

coarse-pos-cmd – Заданное положение без сплайн-интерполяции.

4.2.4.2. Раздел библиотеки PLC

Раздел библиотеки PLC содержит блоки, представляющие программы ПЛК, разработанные пользователем.

4.2.4.3. Раздел библиотеки I/O

Раздел библиотеки IO содержит блоки, представляющие аппаратные входы и выходы СЧПУ, а также ресурсы устройств, подключенных к СЧПУ по цифровым каналам связи.

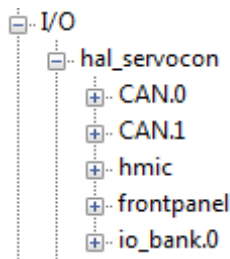


Рис. 34. Раздел IO.

CAN.0, CAN.1 – Каналы CAN.

hmic – Ресурсы пульта дистанционного управления.

frontpanel – Ресурсы передней панели СЧПУ.

io_bank.0 – Плата ввода-вывода СервоКон.

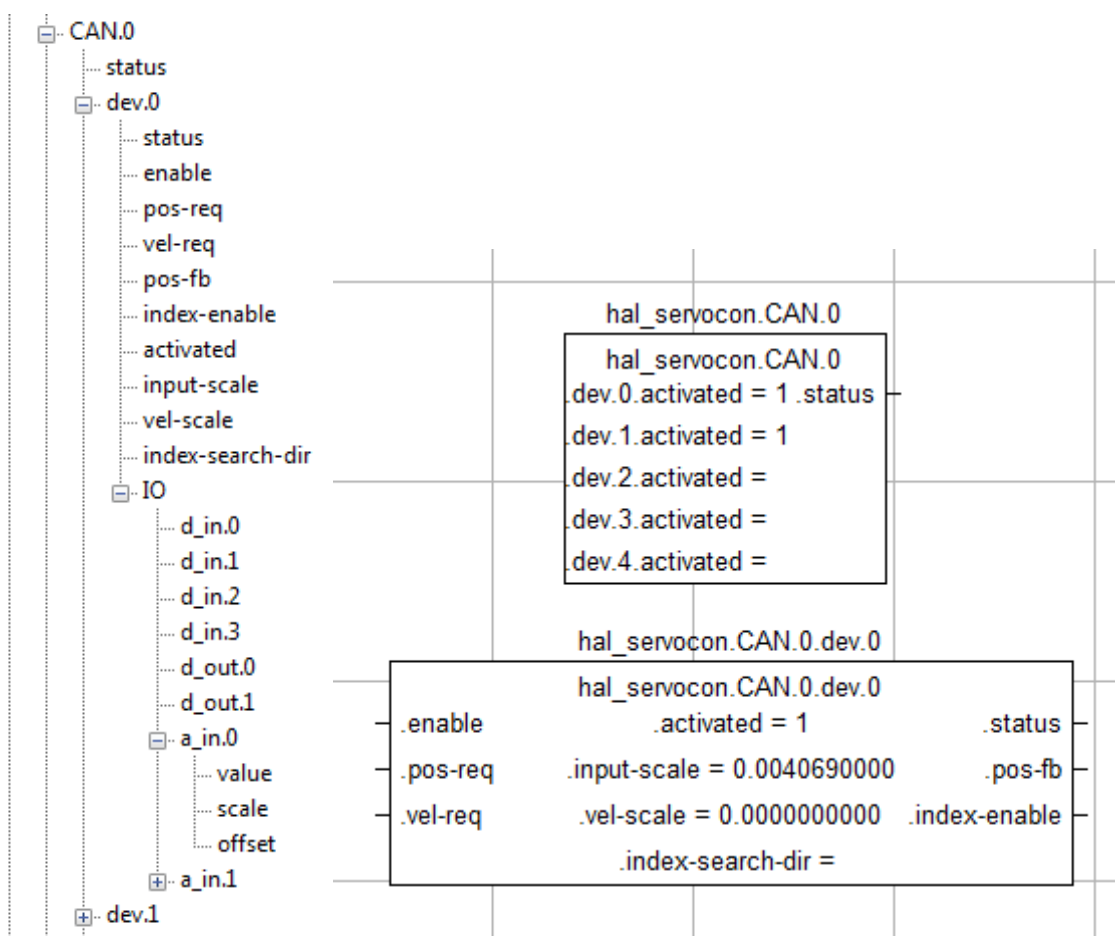


Рис. 35. Раздел IO блок CAN.<n>.

status – Статус шины CAN.

dev.0 – Узел CAN.

status – Состояние устройства, значение True сигнализирует об ошибке или отсутствии связи с устройством.

enable – Разрешение работы.

pos-req – Задание положения.

vel-req – Задание форсирующей связи по скорости.

pos-fb – Текущее положение.

index-enable – Сигнал z-метки.

activated – Установите в 1 для активации узла CAN.

input-scale – Перемещение на 1 инкремент.

vel-scale – Коэффициент форсирующей связи по скорости.

index-serch-dir – Направление поиска z-метки.

IO – Порты ввода/вывода.

d_in.<n>.value – Дискретный вход.

d_out.<n>.value – Дискретный выход.

a_in.<n>.value – Аналоговый вход.

scale – Множитель.

offset – Смещение.

Раздел hmic

Данный раздел описывает ресурсы пульта дистанционного управления.



Рис. 36. Внешний вид пульта дистанционного управления СЧПУ СервоКон[©].

Пульт дистанционного управления (ПДУ) предназначен для управления перемещениями станка по всем осям в ручном режиме. ПДУ поставляется отдельно. См. руководство по эксплуатации СЧПУ СервоКон[©], раздел «Приложения».

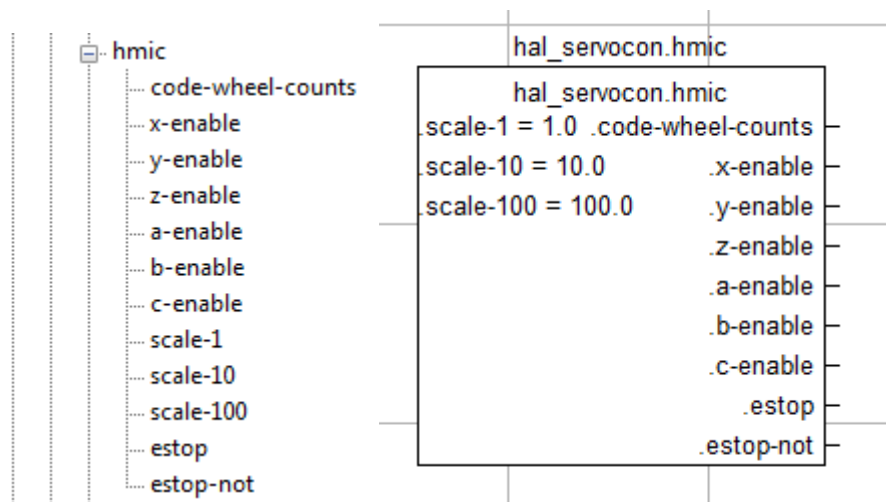


Рис. 37. Раздел IO блок hmic.

Code-wheel-counts – Счетчик импульсов штурвала ПДУ.

x-enable – Сигнал выбора оси X.

y-enable – Сигнал выбора оси Y.

z-enable – Сигнал выбора оси Z.

a-enable – Сигнал выбора оси A.

b-enable – Сигнал выбора оси B.

c-enable – Сигнал выбора оси C.

scale-1 – Масштаб x1.

scale-10 – Масштаб x10.

scale-100 – Масштаб x100.

estop – Кнопка аварийного останова.

estop-not – Кнопка аварийного останова (инвертированный сигнал).

Раздел FRONTPANEL и TOOLCHANGER

Данный раздел содержит ресурсы передней панели СЧПУ СервоКон[®] и ручного сменщика инструмента.

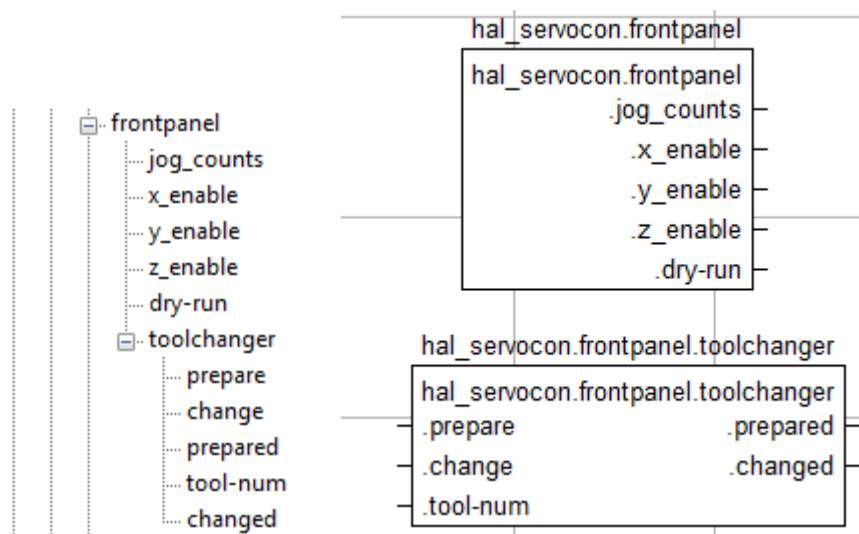


Рис. 38. Раздел IO блоки frontpanel и toolchanger.

jog_counts – Счетчик импульсов ручного управления.

x_enable – Селектор X.

y_enable – Селектор Y.

z_enable – Селектор Z.

dry-run – Работа в холостом режиме.

toolchanger – Ручной сменщик инструментов.

prepare – Команда на подготовку инструмента.

change – Команда на смену инструмента.

prepared – Сигнал готовности.

tool-num – Номер инструмента.

changed – Сигнал готовности.

Раздел библиотеки IO_bank

Данный раздел содержит ресурсы ввода/вывода, которые входят в состав СЧПУ СервоКон[©].

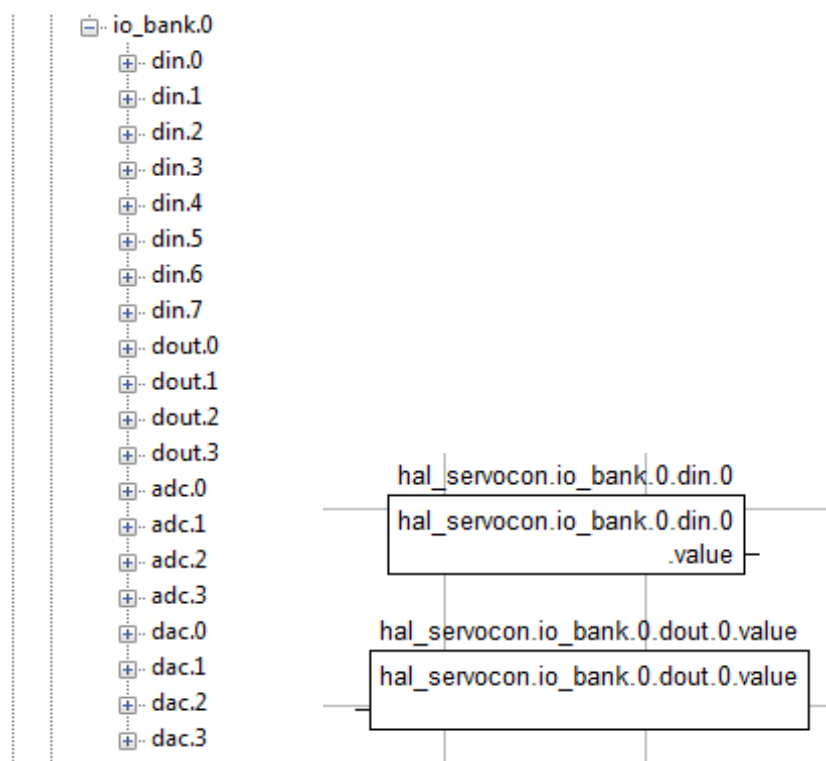


Рис. 39. Раздел IO_bank – Дискретные входы/выходы.

din – Дискретный вход.

dout – Дискретный выход.

value – Значение входа.

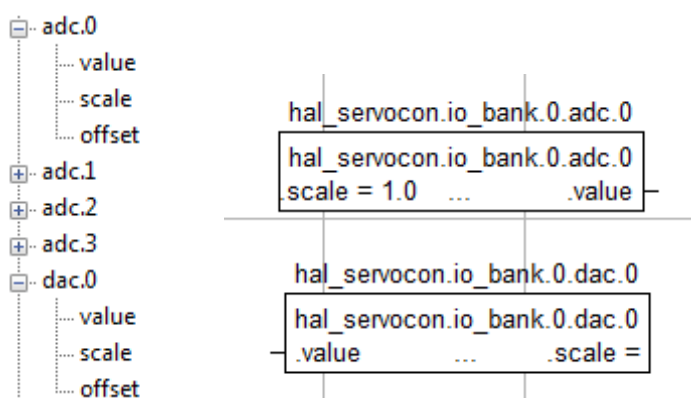


Рис. 40. Раздел IO – АЦП, ЦАП.

adc – АЦП.

dac – ЦАП.

value – Значение выхода.

scale – Множитель.

offset – Смещение.

4.2.4.4. Раздел библиотеки Functions

Раздел библиотеки Functions содержит вспомогательные функции для преобразования сигналов.

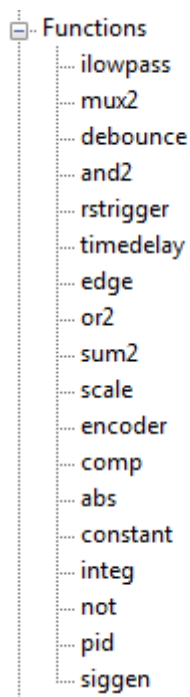


Рис. 41. Раздел Functions.

ilowpass – Фильтр нижних частот.

mux2 – Мультиплексор.

debounce – Фильтрдребезга контактов.

and2 – Логическое И.

rstrigger – RS-триггер.

timedelay – Задержка.

edge – Регистратор фронтов.

or2 – Логическое ИЛИ.

sum2 – Сумматор.

scale – Масштабирование.

encoder – Программный энкодер.

comp – Компаратор.

abs – Абсолютное значение числа.

constant – Постоянная величина.

integ – Интегратор.

not – Логическое НЕ.

pid – ПИД-регулятор.

siggen – Генератор сигналов.

4.2.4.5. Раздел библиотеки Kinematics

Раздел библиотеки Kinematics содержит функции кинематических преобразований.

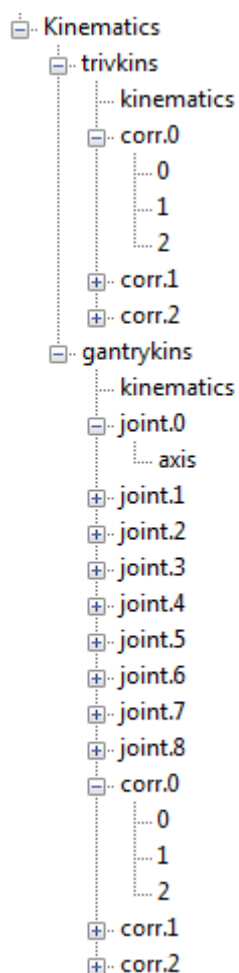


Рис. 42. Раздел Kinematics.

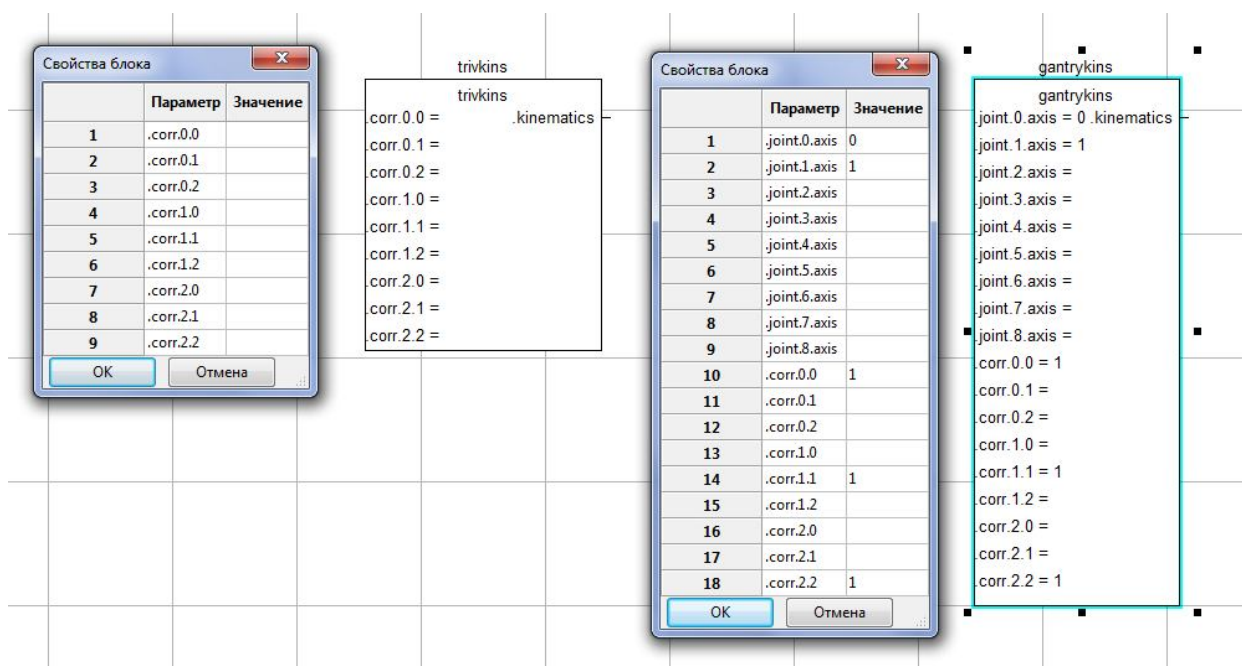


Рис. 43. Блоки тривиальной и портальной кинематики и их свойства.

trivkins – Тривиальная кинематика.

gantrykins – Портальная кинематика.

4.3.Выполняемые процедуры и функции

В разделе описаны общие положения работы с блоками, настройка управления сервоприводами по позиции по шине CAN, настройка управления сервоприводами по скорости, настройка дополнительных функций, редактирование таблиц инструментов, написание программ ПЛК.

4.3.1.Работа с блоками

4.3.1.1. Добавление блоков в программу

Для добавления блока в создаваемую конфигурацию необходимо переместить из библиотеки с помощью мышки в окно редактирования требуемый элемент.



Рис. 44. Внешний вид блока.

4.3.1.2. Соединение блоков

Нажмите левую кнопку мыши на нужном входе/выходе. Удерживая кнопку, подведите курсор ко второму входу/выходу. Если соединение невозможно, то соединение окрасится в красный цвет. Отпустите кнопку - соединение будет создано.

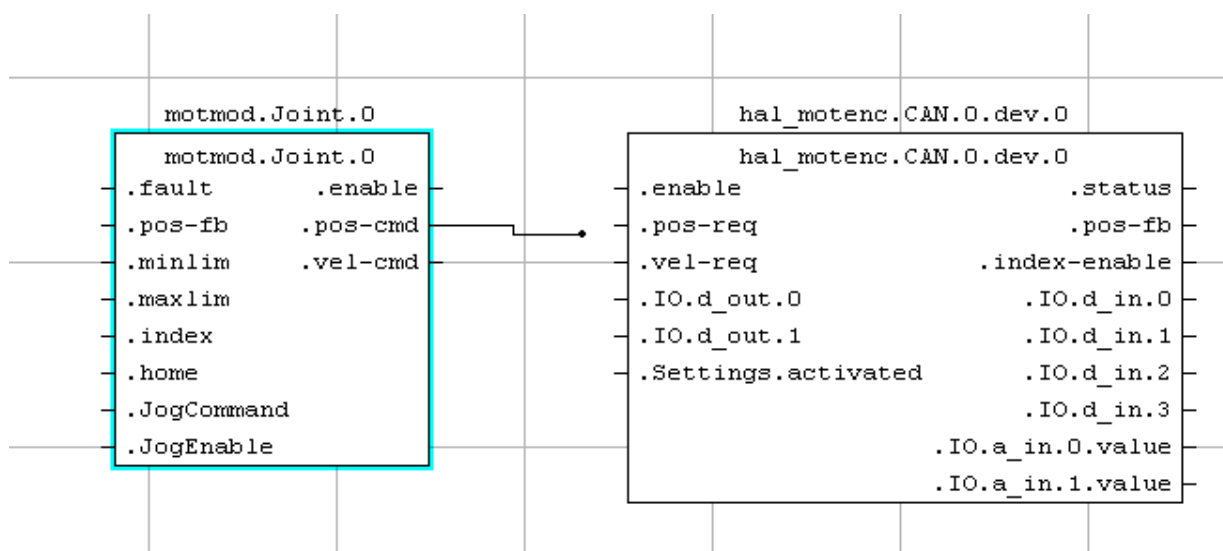


Рис. 45. Способ соединения блоков.

4.3.1.3. Удаление блоков и соединений

Выберите соединение или блок и нажмите клавишу Delete или в контекстном меню выберите "Удалить".

4.3.1.4. Перемещение блоков

Перетаскивайте блоки мышью. Для оптимизации расположения соединения два раза щелкните на него мышью.

4.3.1.5. Добавление комментариев

На верхней панели инструментов выберите "Создать комментарий", обведите рамкой необходимую область, и введите текст в появившееся окно.

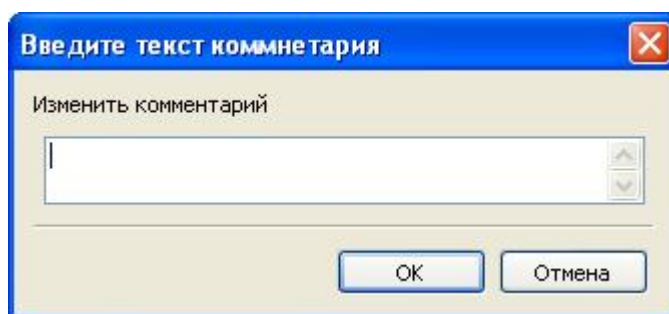


Рис. 46. Окно добавления комментария.

4.3.2. Установка значений параметров

Дважды щёлкните на блоке, параметры которого необходимо настроить. В появившейся таблице параметров введите нужное значение (разделитель целой и дробной части – символ «.») и нажмите кнопку ОК.

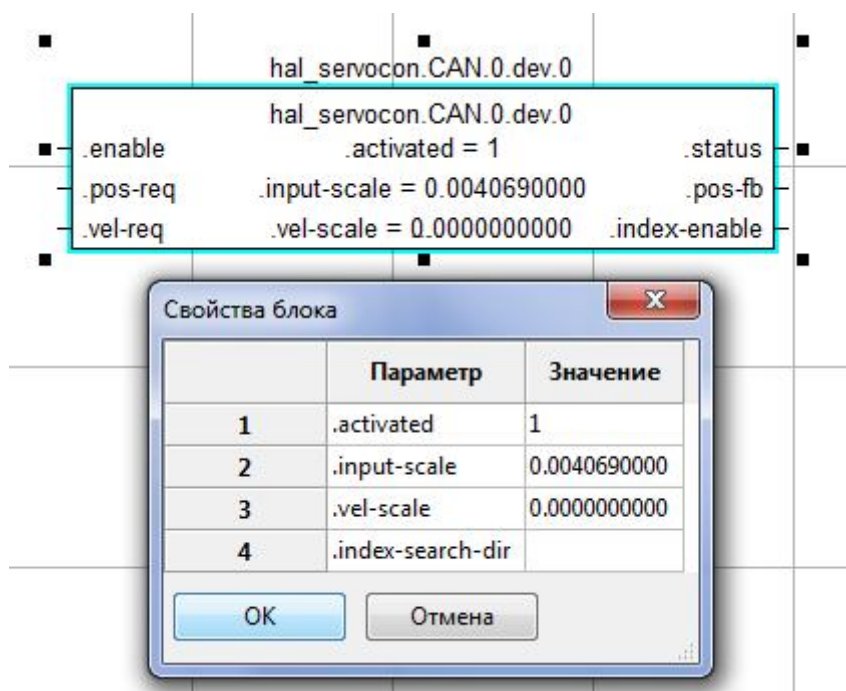


Рис. 47. Окно для задания параметров блока.

4.3.3. Редактирование таблицы инструментов

Для редактирования таблицы инструментов проекта выберите в главном меню пункт «Правка»\ «Таблица инструментов». Откроется окно редактирования таблицы инструментов.

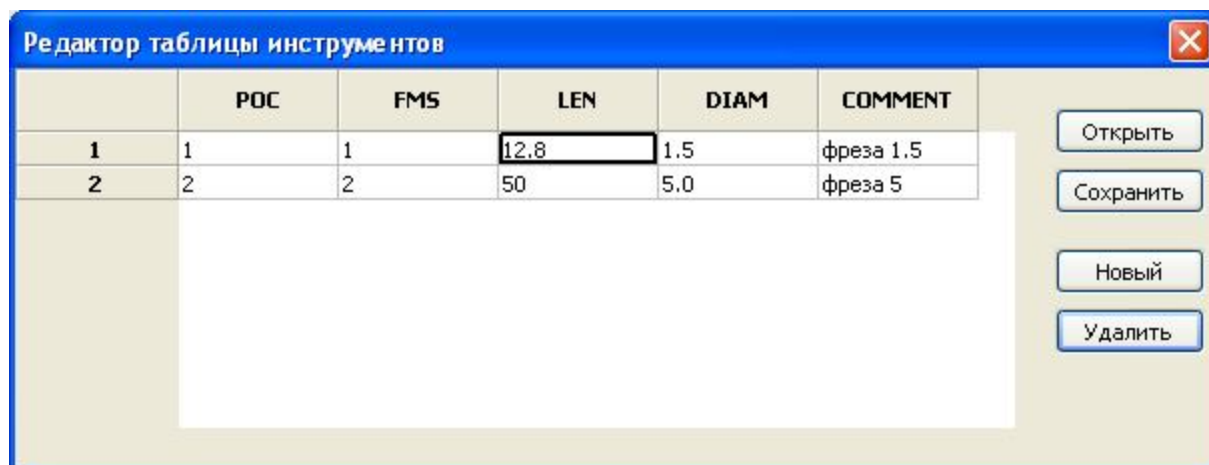


Рис. 48. Редактор таблицы инструментов.

Для добавления таблицы инструментов или выбора новой таблицы нажмите кнопку «Открыть» и выберите нужный файл таблицы или файл шаблона.

Для создания нового инструмента нажмите кнопку «Новый». Для удаления инструмента выберите соответствующую строку и нажмите кнопку «Удалить». Для редакти-

рования параметров инструмента дважды щелкните на соответствующей ячейке, введите нужное значение и нажмите «Enter».

Введите необходимые значения параметров инструментов и закройте окно. Для сохранения таблицы для последующего использования в других конфигурациях нажмите кнопку «Сохранить».

4.3.4. Написание программ ПЛК

Для создания программы ПЛК выберите в главном меню «Правка» – «Редактировать программы ПЛК», нажмите кнопку «Добавить» и введите имя программы.

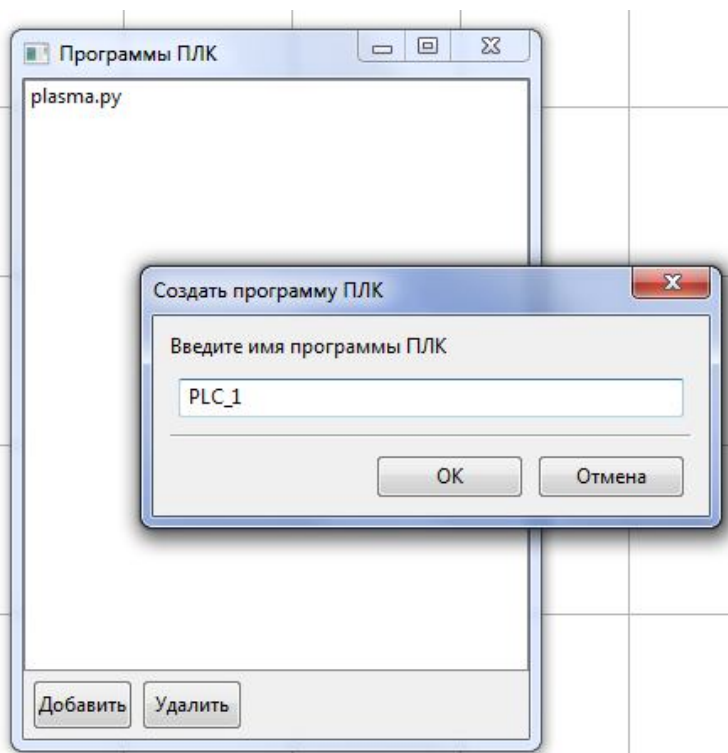


Рис. 49. Окно для создания программы ПЛК.

Для редактирования программы ПЛК дважды щёлкните мышью на имя программы. При этом откроется редактор программ (Рис. 50).

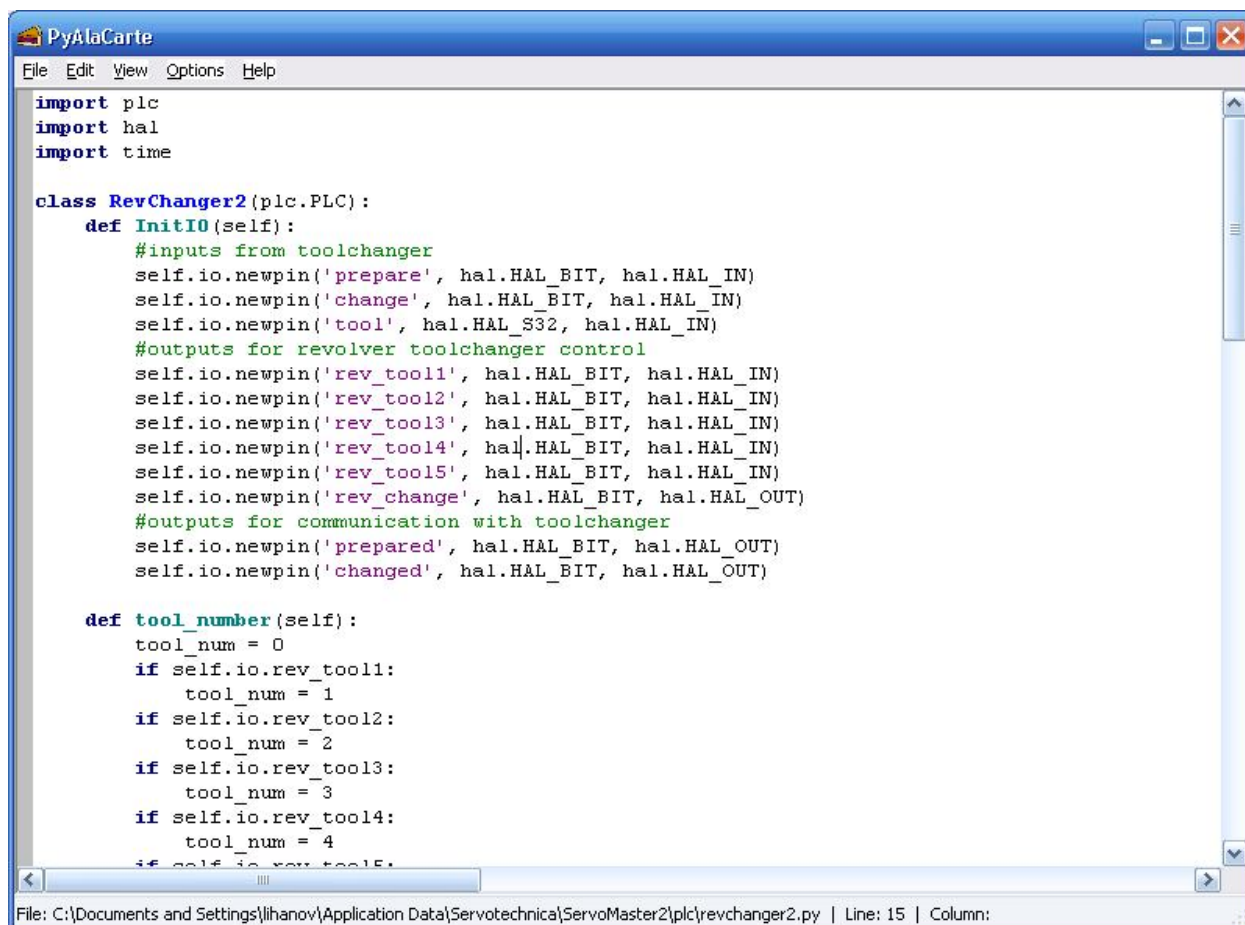


Рис. 50. Окно редактирования программы ПЛК.

После разработки программы данный блок появляется библиотеке функциональных блоков в разделе PLC (Рис. 51, а).

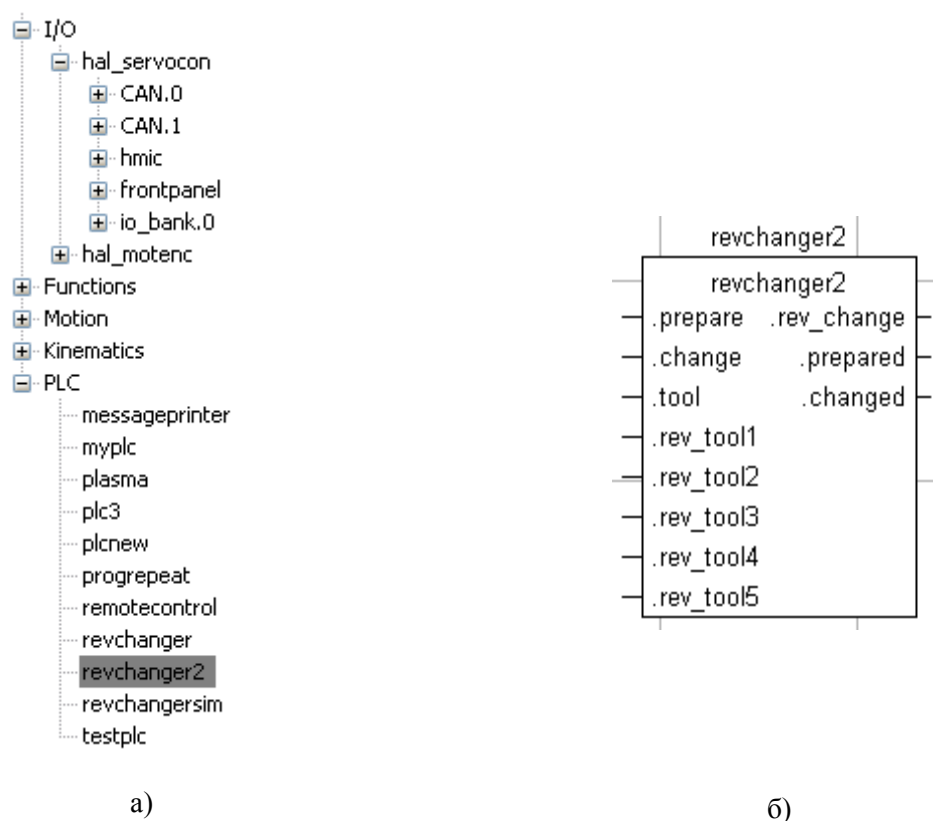


Рис. 51. Графическое представление блока revchanger2.

При этом графическое представление будет выглядеть как показано на Рис. 51, б.

4.3.5. Подключение к СЧПУ

СервоМастер® автоматически обнаружит включённые в локальную сеть СЧПУ при условии правильной настройки на них параметров сети. В случае если к сети необходимо одновременно подключить более одной СЧПУ, требуется предварительно задать каждой из СЧПУ уникальное имя (см. пункт 4.3.6 «Установка параметров СЧПУ»). Для выбора СЧПУ для подключения дважды щелкните мышью на строке с именем СЧПУ в появившемся списке.

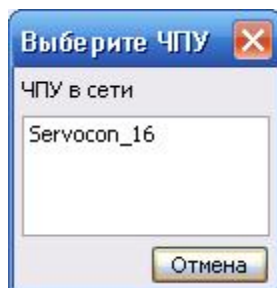


Рис. 52. Окно для подключения СЧПУ.

4.3.6. Установка параметров СЧПУ

Для установки имени и рабочей группы СЧПУ зайдите в меню «СервоКон» - «Изменить параметры», появится окно «Выберите ЧПУ», после выбора СЧПУ появится окно изменения параметров, показанное на Рис. 53. Задайте необходимое имя hostname и название рабочей группы workgroup. Нажмите кнопку «ОК».



Рис. 53. Окно изменения параметров.

4.3.7. Настройка значений параметров

Режим отладки позволяет анализировать работу СЧПУ и изменять параметры в режиме он-лайн. Для перехода в режим отладки в главном меню выберите пункт «СервоКон» – «Подключить», при этом автоматически будет считана текущая конфигурация СЧПУ.

В режиме отладки изменения значений параметров в конфигурации автоматически применяются и сохраняются в СЧПУ.



ОПАСНОСТЬ: Будьте внимательны при изменении параметров. Некорректные значения параметров (например, неверные значения передаточных отношений приводов портала) могут привести к физическому повреждению оборудования.

Рекомендуется настраивать параметры, переведя СЧПУ в состояние сброса.

4.3.8. Осциллограф

Для просмотра текущих значений параметров на панели инструментов выберите «Создать осциллограф», и обведите рамкой необходимую область. В программе появится блок, к входам которого можно подключить любой параметр конфигурации.

Осциллограф имеет два режима работы:

Режим временной развертки показан на Рис. 54. В данном режиме по оси X отображается время с начала сбора данных в секундах, а по оси Y отображаются значения

выбранных параметров. Для использования данного режима подключите необходимые параметры к входам Y1 – Y4.

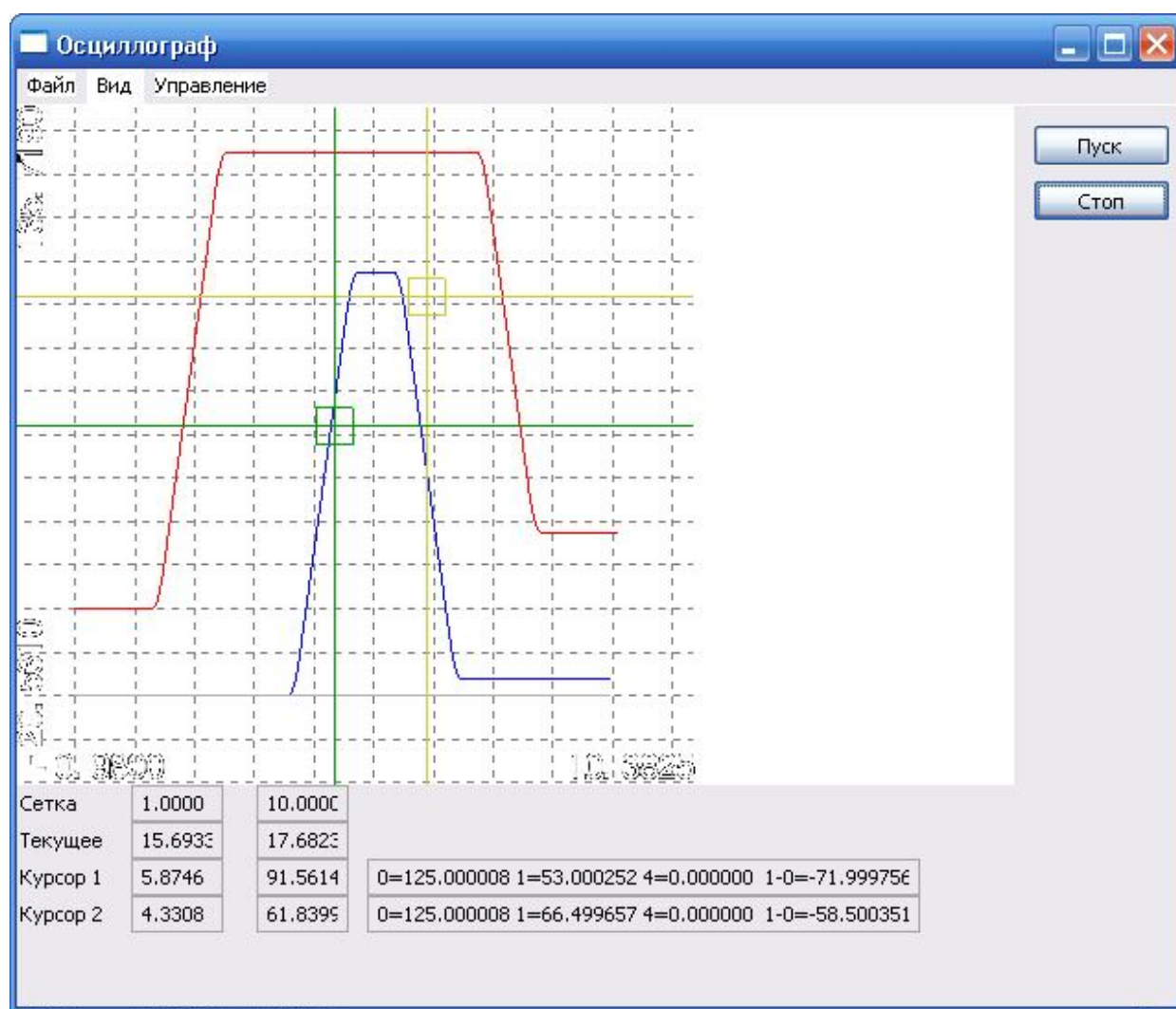


Рис. 54. Режим временной развертки.

Режим плоттера показан на Рис. 55. В данном режиме выбранные пары параметров отображаются в виде кривой в плоскости XY. Этот режим полезен, например, при анализе максимального отклонения реальной траектории движения от заданной. Для использования данного режима подключите необходимые пары параметров к входам X1, Y1 и X2, Y2.

Для анализа изменений параметров дважды щелкните на блоке осциллографа. Отобразится окно осциллографа.

Для начала сбора данных нажмите кнопку «Пуск». Для остановки сбора данных нажмите кнопку «Стоп», при этом осциллограф будет автоматически отмасштабирован под размер собранных данных.

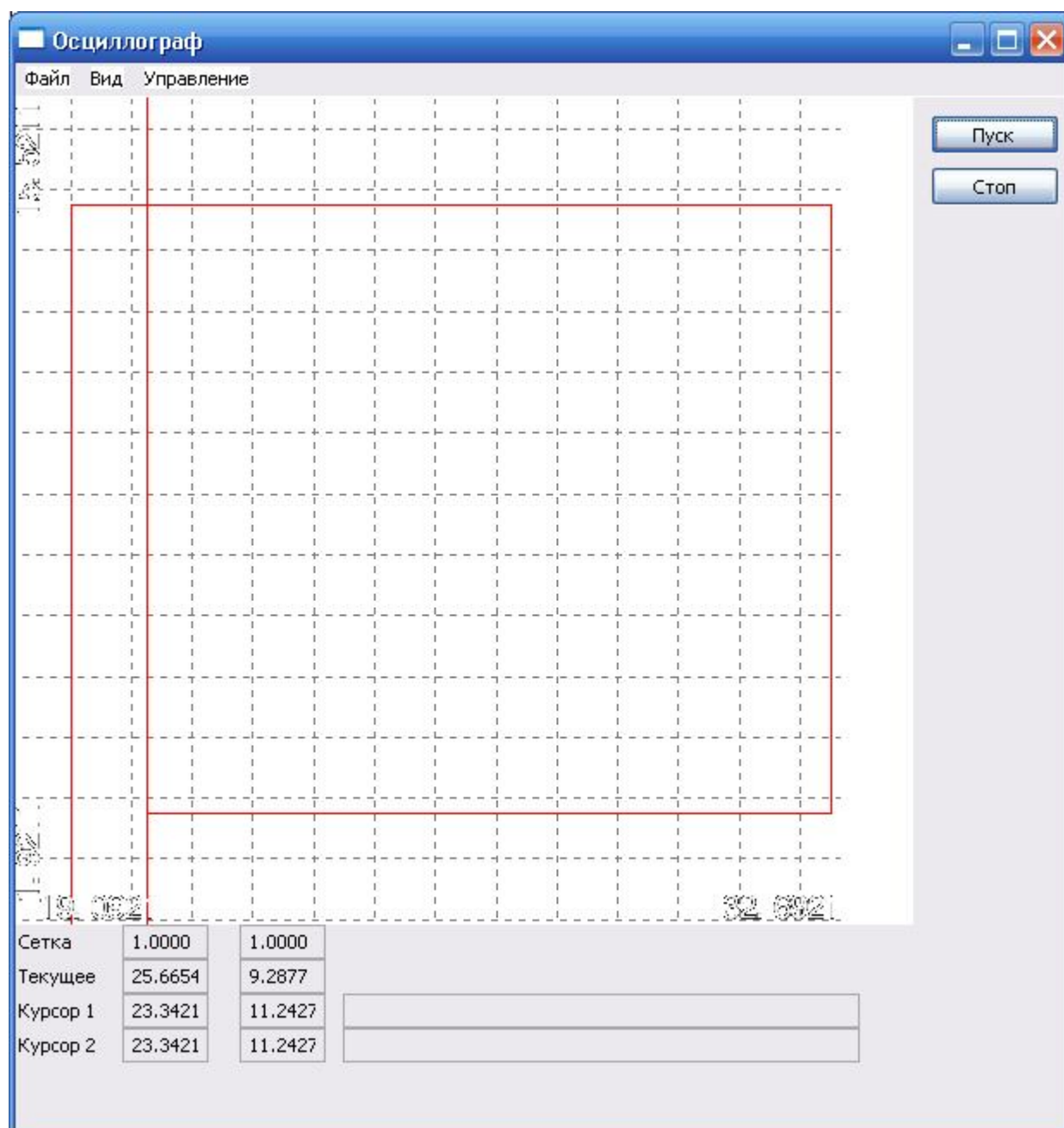


Рис. 55. Режим плоттера.

В левом нижнем углу осциллографа отображаются минимальные значения текущего окна по горизонтали и вертикали. В правом нижнем и левом верхнем углах отображаются максимальные значения.

Для увеличения выбранной части графика, удерживая левую кнопку мыши, обведите рамкой нужную область или подведите курсор к нужной точке и вращайте колесико мыши. Для перемещения отображаемой области перемещайте мышь, удерживая правую кнопку.

Для анализа разницы между показаниями используйте курсоры. При установке флажков Cursor1 и Cursor2 курсоры отобразятся на экране. Для перемещения курсоров перетаскивайте их мышью за среднюю часть.

Под осциллографом отображается текущий шаг сетки, положение указателя мыши и курсоров.

Для настройки отображения данных выберите в меню пункт «Вид» - «Настроить каналы». Отобразится окно со списком каналов. Для настройки свойств канала дважды щёлкните мышью на строке с нужным каналом.

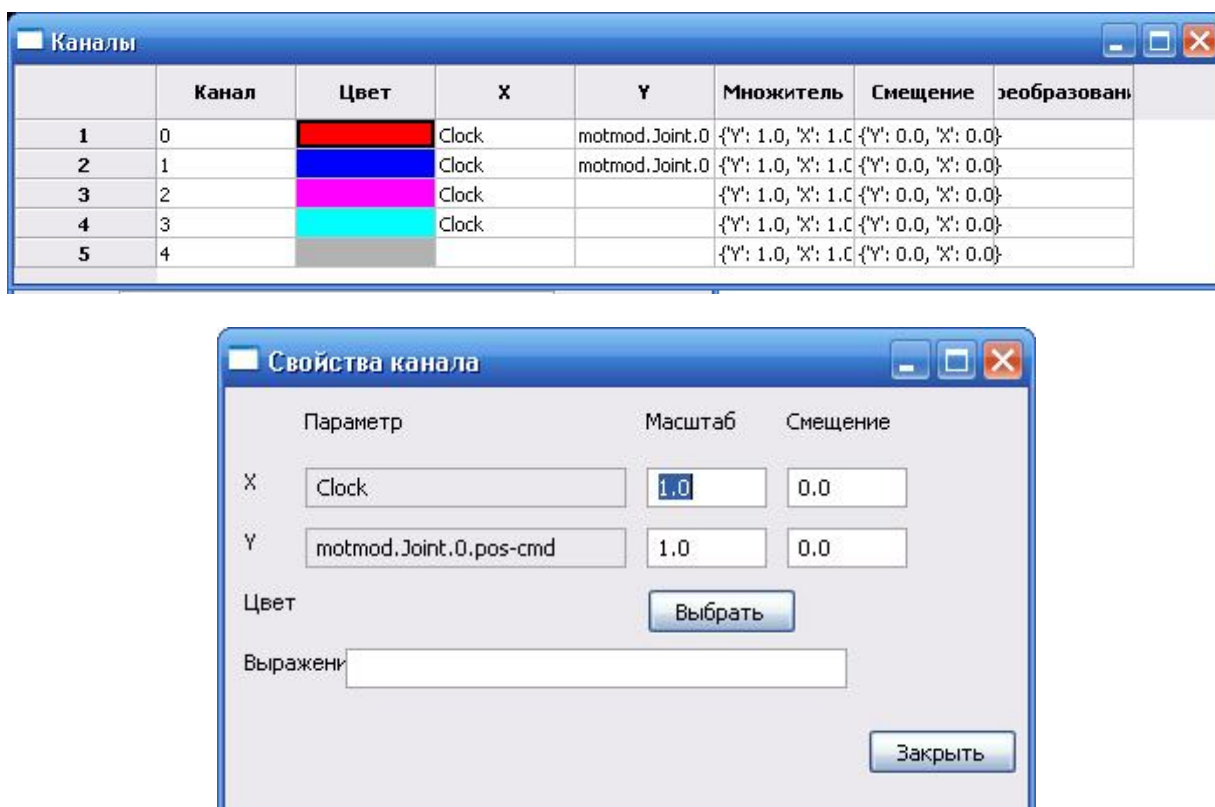


Рис. 56. Настройка каналов.

В настройках можно задать масштаб и смещение канал по осям X и Y и установить цвет отображающей его линии.

4.3.9. Отладка программ ПЛК

Для отладки программ ПЛК в состав программы СервоМастер[©] включён отладчик winpdb.

Для начала отладки:

— Подключитесь к СЧПУ;

- Выберите в меню пункт «СервоКон» - «Запустить отладчик программ ПЛК», при этом отобразится окно отладчика;
- В меню отладчика выберите пункт «File»-«Password» и введите пароль, указанный при запуске отладчика в программе ПЛК;
- В меню отладчика выберите пункт «File»-«Attach» и выберите нужную программу ПЛК из списка, при этом выбранная программа ПЛК будет приостановлена и в окне отладчика отобразится её текст, текущие значения переменных и стек вызовов.

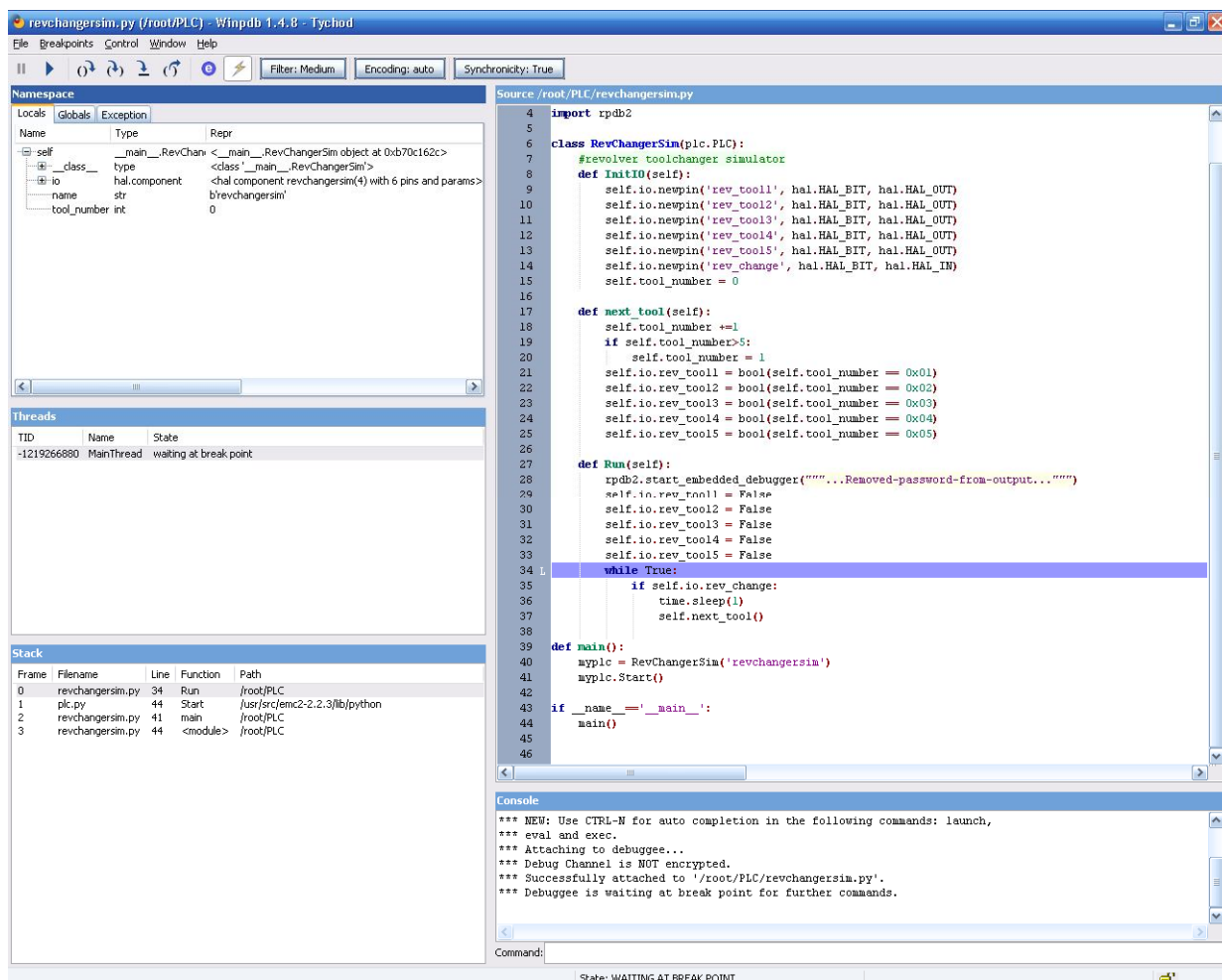


Рис. 57. Окно отладчика программ ПЛК.

Отладчик программ ПЛК состоит из нескольких рабочих окон:

- Namespace;
- Source;
- Threads;
- Stack;

- Console.

Непосредственно программа ПЛК отображается после ее загрузки в отладчик в окне Source . В данном окне можно выполнять программу пошагово при этом текущая строка будет подсвечиваться, а рядом с номером строки появится статус, в виде символа. В данном случае на Рис. 57 это символ L, который указывает на то, что отладчик готов выполнить операцию в текущей строке. Статус C указывает на то, что отладчик готовится выполнить операцию в текущей строке. Статус R, указывает на то, что операция выполнена и можно вернуться к началу выполнения.

Остальные окна являются вспомогательными, для слежения за состоянием выполнения программы при работе с отладчиком:

- Окно Namespace предназначено для отслеживания переменных и их значений при выполнении программы;
- Окно Threads содержит информацию об отлаживаемом потоке выполнения ;
- Окно Stack отображает стек вызовов;
- Окно Console для взаимодействия с отладчиком, добавления и выполнения команд, создания переменных и т.д.

Меню отладчика показано на Рис. 58. Меню позволяет осуществлять быстрый доступ к наиболее часто используемым функциям отладчика.

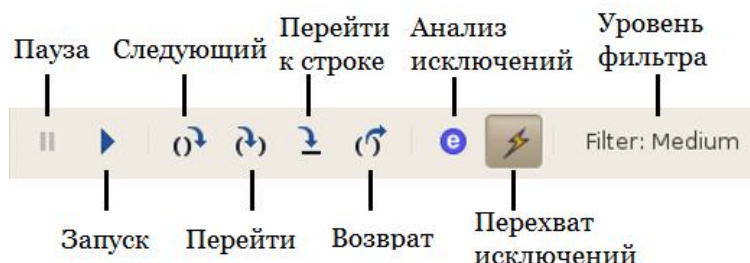


Рис. 58. Меню отладчика.

Добавление и удаление точек останова

Окно Source позволяет добавлять и удалять точки останова. Для добавления точки останова установите курсор на нужную строку в программе в окне Source затем в меню выберите «Breakpoint» - «Toggle» или нажмите F9.

Для удаления точки останова снова выберите нужную строку в программе в окне Source и щелкните по ней, цвет поменяется с красного на белый.

Точки останова представлены в виде подсвеченной строки красного желтого или синего цвета. Красный цвет означает, что точка останова активирована, желтый, что точка останова деактивирована, синий, что точка останова достигнута.

Выполнение по шагам

Для перехода к определенной строке нажмите в меню кнопку «Перейти» или введите в командной строке команду `s «step»`. Для перемещения по отлаживаемому участку программы по шагам нажимайте кнопку «Следующий».

Просмотр значений переменных

Для просмотра значения переменных пользуйтесь окном Namespace. В данном окне отображаются имена, типы и значения глобальных и локальных переменных программы.

Стек вызовов

В окне стека вызовов показан перечень функций, которые вызывались программой в момент достижения точки останова. В самом верху списка в окне вызовов расположена функция, которая вызывалась последней.

Подробная документация по использованию отладчика доступна по адресу <http://winpdb.org/docs/>.

5. Составление конфигурации СЧПУ СервоКон[®]

Для написания конфигурации необходимо: во-первых, разработать структуру конфигурации, исходя из заданных ресурсов СЧПУ; во-вторых, описать саму конфигурацию в виде файла, определив аппаратные и логические ресурсы и связь между ними, загрузить конфигурацию в СЧПУ и отладить ее, проанализировав работу СЧПУ и сделав необходимые изменения.

5.1. Выбор типа кинематики

Выбор типа кинематики осуществляется исходя из конфигурации станка. Для выбора необходимо подключить выход «kinematics» выбранного модуля кинематики к логическому входу «kin» модуля motmod. В простейшем случае (одна координатная ось приводится в движение одним сервоприводом) используется тривиальная кинематическая схема, для которой не нужно настраивать параметры.

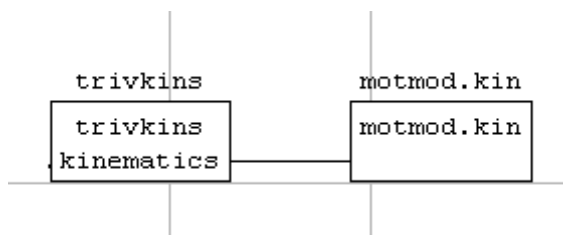


Рис. 59. Выбор типа кинематики.

Для станков портального типа (одна ось приводится в движение несколькими сервоприводами, работающими параллельно) используется портальная кинематическая схема (gantrykins), для которой необходимо указать, какие оси приводятся в движение сервоприводами.

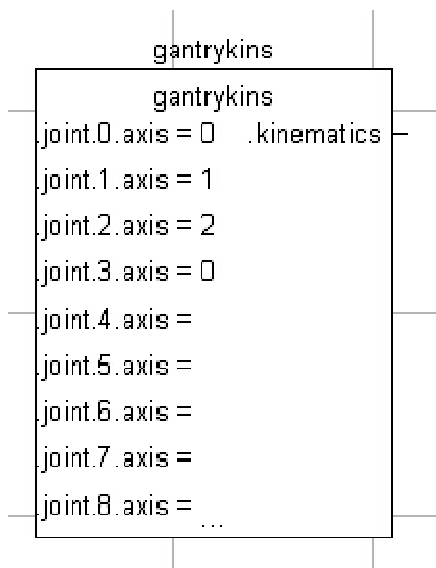


Рис. 60. Пример настройки параметров.

В данном примере ось X приводится в движение сервоприводами 0 и 3.

ОПАСНОСТЬ: Некорректная настройка сервоприводов станков с нетривиальной кинематикой может повредить станок. Перед выполнением движения убедитесь, что:

- Правильно настроены передаточные отношения сервоприводов
- У сервоприводов установлены одинаковые максимальные скорости и ускорения
- Одинаково настроены ПИД-регуляторы сервоприводов
- У сервоприводов выбран один и тот же цикл выхода в 0
- Команды ручного управления и разрешения ручного управления подключены к одному источнику
- Установлены одинаковые масштабы ручного управления сервоприводов
- Обязательно проверьте корректность направлений вращения и реакции на концевые выключатели и датчики нулевого положения до соединения двигателей портала с механическими компонентами станка



5.2. Настройка сигнала разрешения работы

Для реализации функции аварийного останова в СЧПУ используется вход `mot-mod.Enable`. Значение 0 на этом входе переводит СЧПУ в состояние аварийного останова.

В состоянии аварийного останова:

- снимается сигнал разрешения со всех сервоприводов,
- останавливается вращение шпинделя.

Как правило, к входу разрешения подключается сигнал `hal_motenc.estop-not` – состояние кнопки аварийного останова на передней панели и пульте (если он подключен). Если необходимо обеспечить останов движения по каким-либо другим признакам (датчикам столкновения, рассогласованию осей портала и т. п.), то эти сигналы необходимо подключать через логический блок И.

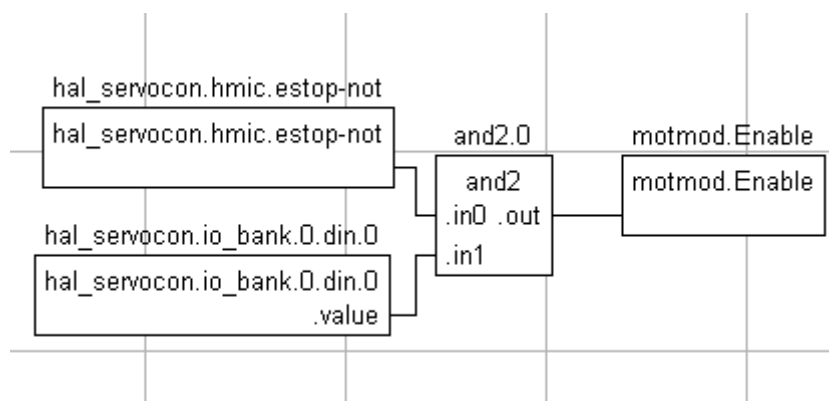


Рис. 61. Настройка сигнала разрешения.

В данном примере аварийный останов произойдет в случае нажатия кнопки аварийного останова или деактивации входа дискретного входа 0 СЧПУ.

5.3. Настройка выхода в нулевую позицию

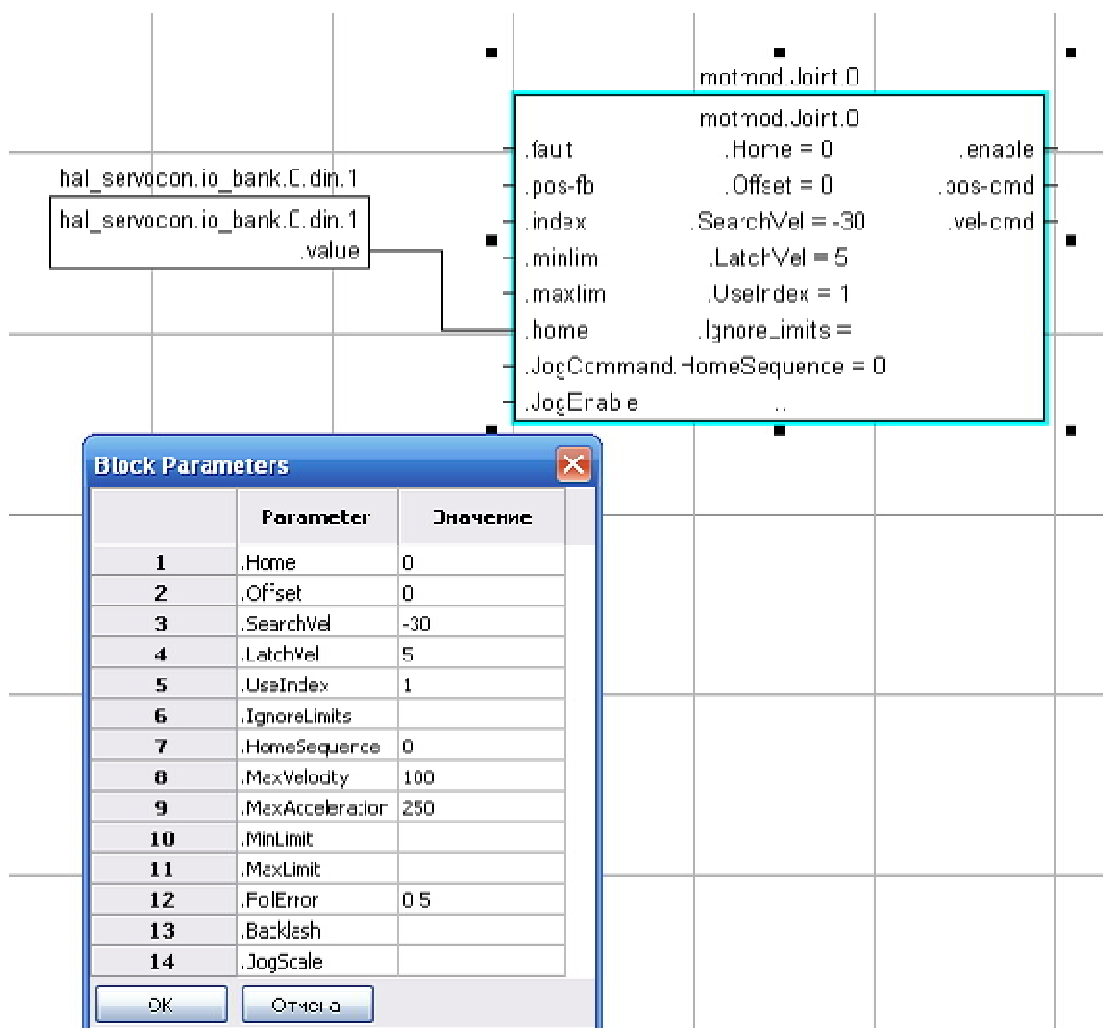


Рис. 62. Настройка выхода в нулевую позицию.



ВНИМАНИЕ: в сервоприводах СПШ/СПС необходимо задать направление поиска z-метки с помощью параметра `index-search-dir`.

Если для определения нулевого положения используется тот же датчик, что и для аппаратного ограничения положения, то необходимо установить параметр `IgnoreLimits` в значение 1, иначе в момент активации датчика движение будет остановлено с ошибкой.



ОПАСНОСТЬ: При таком варианте подключения обязательно убедитесь в правильности направления поиска нуля. В противном случае СЧПУ не остановит движение при срабатывании концевого выключателя в процессе выхода в нуль, что приведет к выходу сервопривода станка из рабочей зоны.

Последовательность поиска нуля сервоприводами определяется параметром Home-Sequence (номер цикла). Сервоприводы с одним номером цикла поиска нуля выполняют процедуру одновременно.



ОПАСНОСТЬ: При использовании портальной кинематики убедитесь, что номера циклов и направления поиска датчика совпадают для параллельно работающих сервоприводов. В противном случае СЧПУ будет давать задание на движение одному из сервоприводов портала, что может привести к механическому повреждению станка.

5.4. Настройка максимального рассогласования

При некорректной настройке сервопривода или недостаточной его мощности для движения со скоростями и ускорениями, задаваемыми СЧПУ, между текущим и заданным положением возникает рассогласование, пропорциональное скорости движения. Рассогласование возникает и в случае нормального режима работы сервопривода, но в этом случае оно сохраняется примерно постоянным. В СЧПУ параметром Joint.n.FolError может быть задано максимальное значение этой ошибки, при превышении заданного значения СЧПУ остановит движение и перейдет в состояние сброса.

5.5. Настройка аппаратных и программных ограничений положения

Для предотвращения повреждения станка при выходе за допустимую область перемещения используют ограничения положения. Поддерживаются ограничения двух типов:

Аппаратные ограничения – подключаются к входам minlim и maxlim сервоприводов. При активации этих входов СЧПУ отключит все сервоприводы и перейдет в состояние сброса.

Программные ограничения – устанавливаются в параметрах MinLimit и MaxLimit. Если СЧПУ получает команду на перемещение за пределы программных ограничений, то перемещение будет завершено в положении программного ограничения, если же программа движения содержит выход за пределы программных ограничений, то СЧПУ остановит выполнение программы и сообщит об ошибке.



ПРИМЕЧАНИЕ: Программные ограничения положения задействуются только после выполнения процедуры выхода в 0.

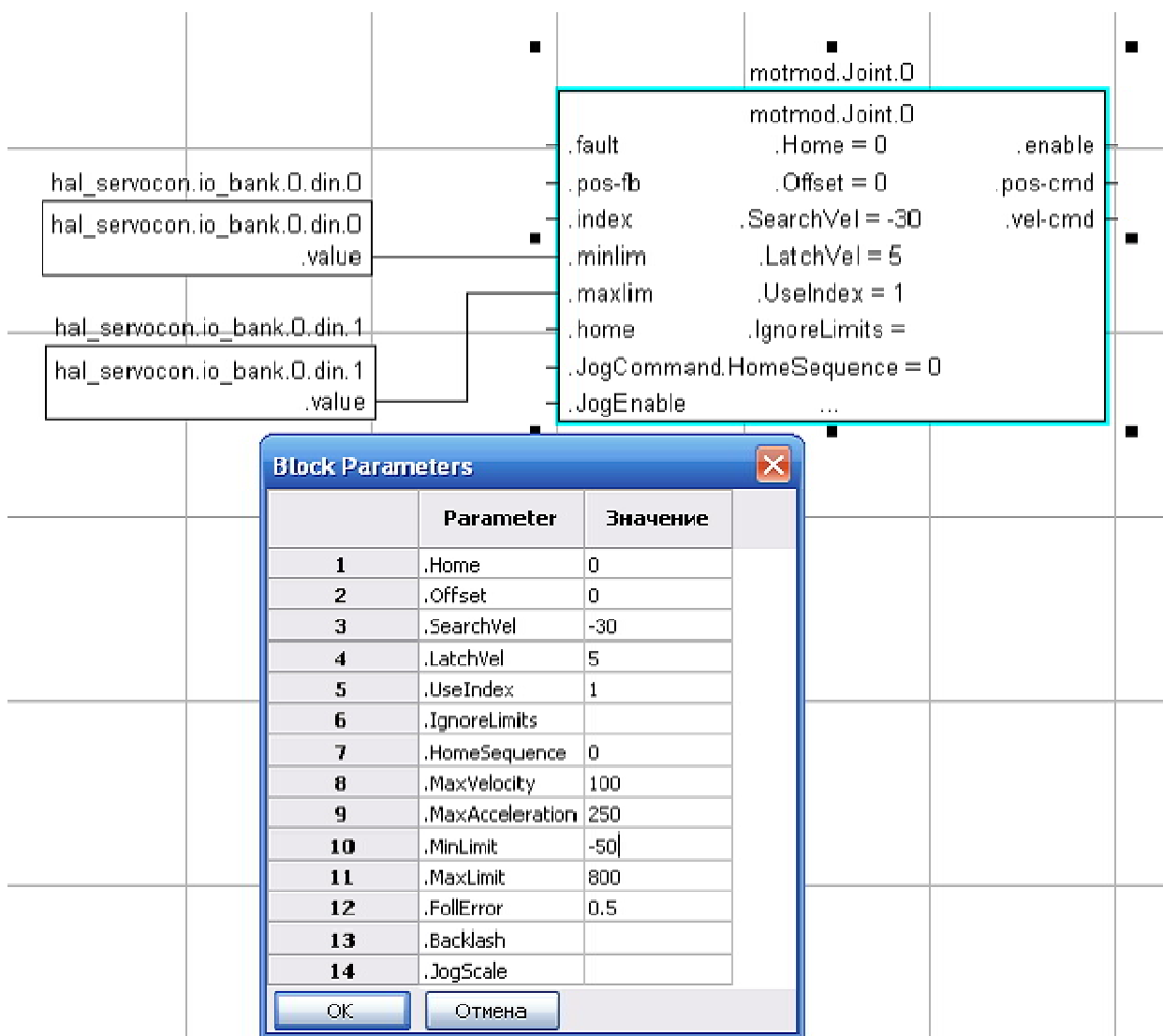


Рис. 63. Настройка ограничений положения.

5.6. Управление по позиции по шине CAN

СЧПУ СервоКон® позволяет управлять приводами СПШ/СПС по позиции с передачей управляющего сигнала и обратной связи по шине CAN. При таком варианте подключения контур позиции рассчитывается непосредственно в приводах, а СЧПУ только формирует необходимое задание по позиции.

5.6.1. Подключение заданий и обратных связей сервоприводов

Для управления сервоприводом по шине CAN необходимо соединить соответствующие входы и выходы сервопривода и узла шины CAN, в частности:

- разрешение работы (enable) сервопривода подключить к входу разрешения работы (enable) узла CAN;

- задание положения (pos-cmd) сервопривода подключить к входу задания положения (pos-req) узла CAN;
- вход ошибки сервопривода (fault) сервопривода подключить к выходу состояния (status) узла CAN;
- вход обратной связи по позиции (pos-fb) сервопривода подключить к выходу текущего положения (pos-fb) узла CAN;
- установить параметр activated узла CAN в 1.

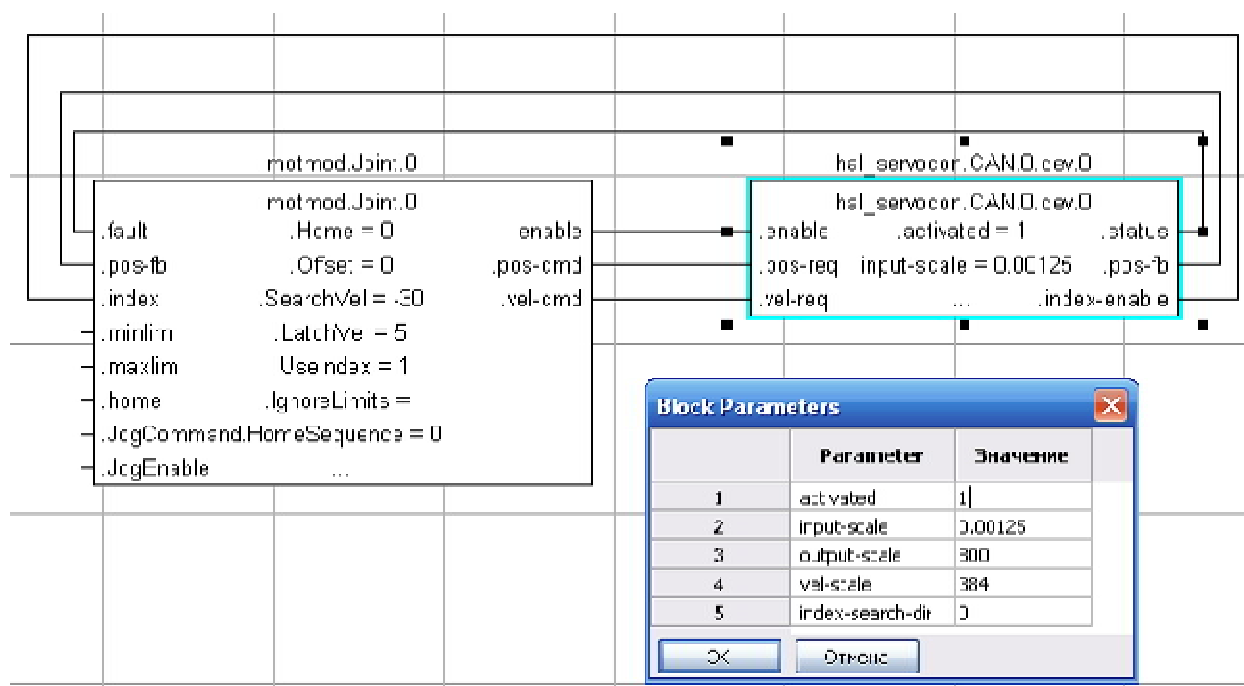


Рис. 64. Настройка управления сервоприводом по CAN.

5.6.2. Настройка передаточных отношений

Для соблюдения правильного масштаба перемещений необходимо указать передаточное отношение. Для узла шины CAN:

Input-scale – перемещение на 1 инкремент энкодера.



ВНИМАНИЕ: Вводите передаточное отношение как можно более точно, округление значения приведёт к появлению систематической погрешности, пропорциональной величине перемещения.

Пример. Сервопривод с разрешением энкодера 4000 инк/оборот приводит в движение шестерню диаметром 80 мм через редуктор с передаточным отношением 10.

Перемещение на 1 инкремент составит

$80 \cdot \pi / (10 \cdot 4000) = 0,006283185307179586476925286766559$ мм

Округлив это значение до 0,00628, получим на перемещении 1000 мм ошибку порядка 80 инкрементов, т.е. 0,5 мм.

5.6.3. Настройка сервоприводов

Для включения управления сервоприводом СПШ/СПС по CAN необходимо установить в нем следующие параметры:

Vp9: режим с максимальной динамикой;

Vp7: замкнут;

Pp5: замкнут;

Ip0: интерфейс CAN;

Ip7: адрес узла в конфигурации +1;

Ip8: 0;

Ip9: режим контурного управления;

Ip15: 1000 кБит/с;

Ip16: Выкл.

Настройку контуров скорости и положения сервоприводов СПШ/СПС при таком способе управления необходимо осуществлять через программу МотоМастер[®], отключив синхронизацию по CAN. Подробные инструкции по подключению и настройке сервоприводов СПШ/СПС приведены в руководстве пользователя.

5.7. Настройка дополнительных функций

5.7.1. Ручное управление

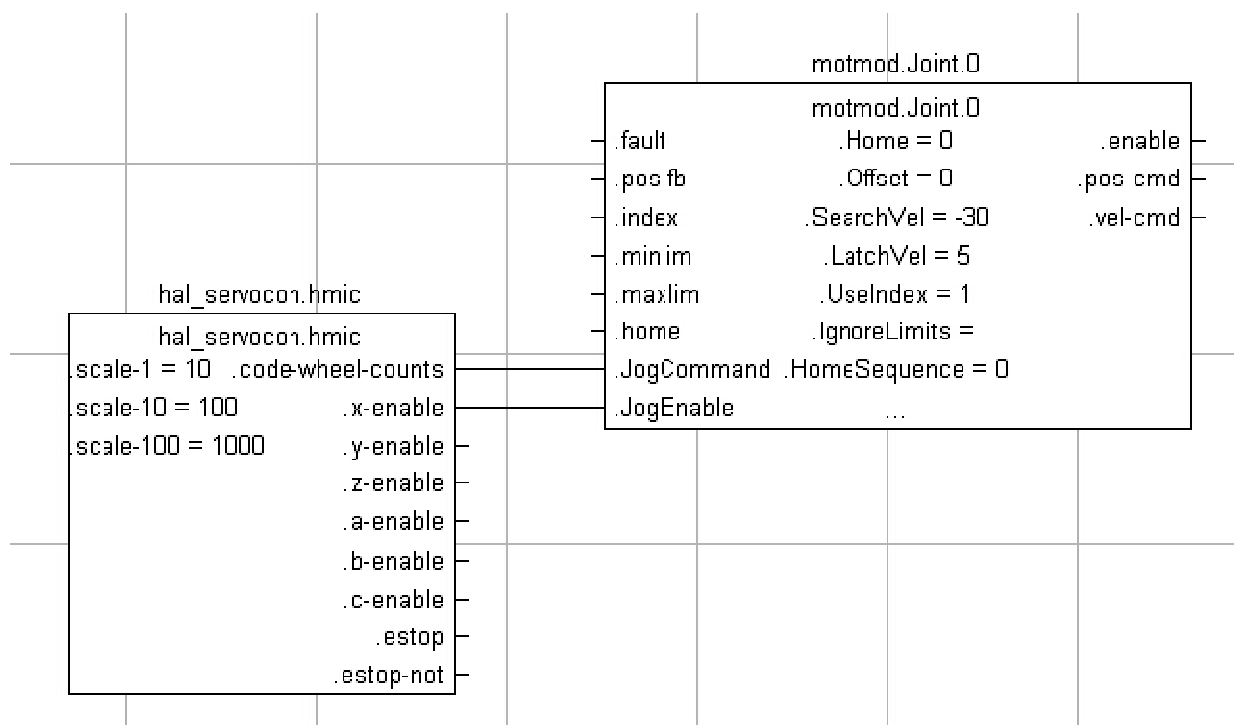


Рис. 65. Настройка ручного управления.

Для ручного управления используются входы сервопривода JogCommand и JogEnable. Активирование входа JogEnable разрешает ручное управление сервоприводом, после этого изменение входа JogCommand приводит к движению сервопривода. Как правило, на вход JogCommand подключаются выходы пульта, передней панели или энкодера. Величина перемещения на единицу изменения JogCommand определяется параметром JogScale сервопривода.

При использовании пульта ручного управления с CAN –интерфейсом изменение управляющего сигнала на инкремент энкодера рукоятки можно изменить в зависимости от селектора масштаба с помощью параметров scale-1, scale-10 и scale-100.

При управлении сервоприводами в ручном режиме рампа по скорости не формируется, соответственно, при управлении сервоприводами с высокими ускорениями задание имеет форму ступеньки, что приводит к рывкам при движении. Для исключения этого эффекта можно использовать функцию ilowpass, представляющую собой фильтр низких частот, со значением параметра gain в диапазоне от 0 до 1.



ОПАСНОСТЬ: При управлении порталом обязательно подключите сигнал разрешения ручного управления приводам портала к одному источнику и задайте одинаковые коэффициенты, в противном случае возможно механическое повреждение станка.

5.7.2. Управление шпинделем

Если используется шпиндель с постоянной скоростью вращения, то для управления им необходимо подключить выходы Spindle.On и Spindle.Forward и/или Spindle.Reverse для включения привода шпинделя.



ВНИМАНИЕ: Если в программе движения не задана скорость шпинделя или установлена скорость шпинделя равная нулю (кодовым словом S), то сигналы включения установлены не будут.

5.7.3. Управление подачей СОЖ

Для использования функций включения/выключения охлаждения, подключите выходы AUX.Flood-on и AUX.Mist-on к дискретным выходам, управляющим соответствующими устройствами.

5.7.4. Настройка сменщика инструментов

Если станок не предполагает смены инструмента, то рекомендуется подключить выходы prepare и change к входам prepared и changed соответственно для того, чтобы избежать бесконечного цикла ожидания смены инструмента.

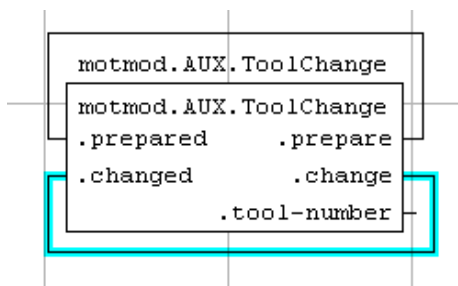


Рис. 66. Пример настройки сменщика инструментов (не предполагается смены).

Если смена инструмента производится вручную, то сообщить СЧПУ об окончании процесса смены инструмента можно либо вручную (используя, например, внешнюю кнопку, подключенную к дискретному входу), либо используя блок frontpanel.toolchanger. При этом при необходимости смены инструмента на дисплее СЧПУ высветится соответствующее сообщение, и выполнение программы будет приостановлено до нажатия клавиши Enter.

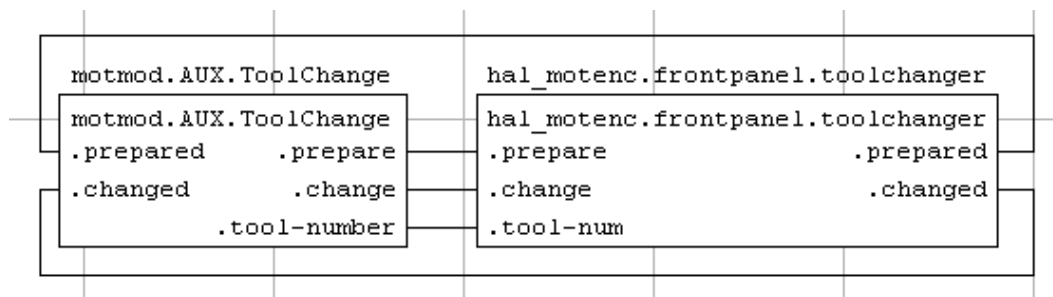


Рис. 67. Пример настройки сменщика инструментов (смена вручную).

Для использования функций коррекции в конфигурации должна присутствовать таблица инструментов. Представление таблицы инструментов в программе СервоМастер[®] см. на Рис. 48.

Инструмент для фрезерного станка имеет следующие параметры:

- POC – номер кармана сменщика инструментов, в который установлен данный инструмент;
- FMS – номер инструмента (задающийся кодовым словом T);
- LEN – длина инструмента;
- DIAM – диаметр инструмента;
- COMMENT – описание (может отсутствовать);

Инструмент для токарного станка имеет следующие параметры:

- POC – номер кармана сменщика инструментов, в который установлен данный инструмент;
- FMS – номер инструмента (задающийся кодовым словом T);
- ZOFFSET – длина инструмента;
- XOFFSET – смещение инструмента по оси X;
- DIAM – диаметр инструмента;
- ORIENTATION – ориентация инструмента;
- COMMENT – описание (может отсутствовать).

Если программа движения содержит траекторию инструмента, то функцию коррекции диаметра можно использовать для компенсации износа инструмента или использования инструмента, отличного от того, который использовался для построения траектории – в этом случае в поле DIAM заносится отличие текущего диаметра инструмента от расчетного.

5.8. Создание конфигурации для СЧПУ СервоКон®

В главном меню выберите «Файл» - «Создать конфигурацию». Отобразится окно с сообщениями о прогрессе операции, в конце будет выведена информация об ошибках и предупреждениях. Если конфигурация создана успешно (не было сообщений об ошибках), программа предложит сохранить полученную конфигурацию.

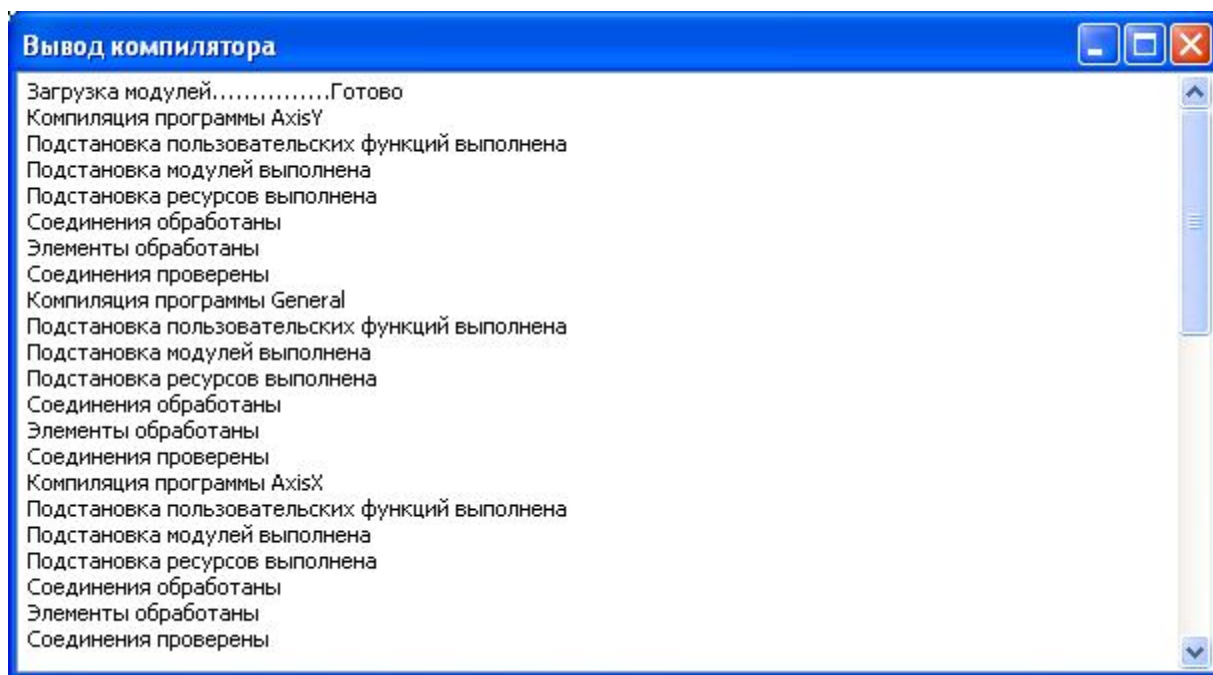


Рис. 68. Окно процесса сохранения конфигурации.

Полученная конфигурация (с расширением `ssconfig`) может быть загружена в СЧПУ через USB-диск или сеть Ethernet.

5.9. Удаленная запись конфигурации

В главном меню выберите пункт «СервоКон» – «Записать конфигурацию» и выберите нужный файл конфигурации.

6. Сообщения оператору

Табл. 17. Перечень сообщений

Текст сообщения	Причина, эффект на СЧПУ	Способ устранения
Не подключен сигнал разрешения работы	Не подключен вход <code>motmod.Enable</code> . СЧПУ будет постоянно находиться в состоянии <code>ESTOP</code>	Подключить нужный сигнал к входу <code>motmod.Enable</code>
Задана нулевая максимальная подача - движение в координатном режиме невозможно	Параметр <code>motmod.MaxVelocity</code> не установлен или установлен в значение 0. Движение СЧПУ будет возможно только в ручном режиме	Установить нужное значение параметра <code>motmod.MaxVelocity</code>
Задано нулевое максимальное ускорение - движение невозможно	Параметр <code>motmod.MaxAcceleration</code> не установлен или установлен в значение 0. Движение СЧПУ будет возможно только в ручном режиме	Установить нужное значение параметра <code>motmod.MaxAcceleration</code>
Не задана скорость поиска датчика нулевого положения	Параметр <code>motmod.Joint<n>.SearchVel</code> не установлен или установлен в значение 0. СЧПУ не будет выполнять поиск концевого выключателя при поиске нулевого положения	Установить нужное значение параметра
Не подключен сигнал z-метки	Параметр <code>motmod.Joint<n>.UseIndex</code> установлен в значение 1, но вход <code>index</code> этого сервопривода не подключен. СЧПУ не сможет выполнить поиск z-метки при выходе в 0	Установить нужное значение параметра в 0 или подключите вход <code>motmod.Joint<n>.index</code>
Сигнал нулевого положения подключен к концевому датчику - установите <code>IgnoreLimits</code> в 1	Вход <code>motmod.Joint<n>.home</code> подключен к тому же блоку, что и один из входов ограничения положения сервопривода Выход в 0 будет заканчиваться ошибкой «Сработал аппаратный ограничитель положения»	Установить значение параметра <code>motmod.Joint<n>.IgnoreLimits</code> в 1
Установлена максимальная скорость оси 0 - движение будет невозможно	<code>motmod.Joint<n>.MaxVelocity</code> не установлен или установлен в значение 0. СЧПУ не сможет выполнить движения, в которых участвует данный сервопривод	Установите нужное значение параметра
Установлено максимальное ускорение оси 0 - движение будет невозможно	<code>motmod.Joint<n>.MaxAcceleration</code> не установлен или установлен в значение 0. СЧПУ не сможет выполнить движения, в которых участвует данный сервопривод	Установите нужное значение параметра
Не установлен гистерезис для компаратора	Параметр <code>hyst</code> компаратора не установлен или установлен в значение 0. Выход <code>equal</code> компаратора будет равен <code>false</code> если только на оба входа не подключен один и тот же сигнал	Установите нужное значение параметра

Не подключен сигнал разрешения сервопривода CAN	Не подключен вход CAN.<n>.dev.<m>.enable сервопривод будет постоянно находиться в выключенном состоянии	Подключите ко входу нужный сигнал
Устройство CAN не активно - установите activated для управления по CAN	Не установлен параметр CAN.<n>.dev.<m>.activated Обратная связь сервопривода не будет обрабатываться и ему не будет посылаться команда управления	Установите параметр в значение 1
Не задано передаточное отношение	Параметр CAN.<n>.dev.<m>.input-scale не установлен или установлен в значение 0. Сервоприводу будет передаваться задание положения 0 и обратная связь сервопривода всегда будет равна 0	Установите нужное значение параметра
Не выбран тип кинематики	В конфигурации должно присутствовать кинематическое преобразование, выход kinematics которого должен быть подключен ко входу motmod.kin Конфигурация не создается	Добавьте в конфигурацию нужное кинематическое преобразование
Не задана скорость съезда с датчика нулевого положения	Параметр motmod.Joint<n>.LatchVel не установлен или установлен в значение 0. СЧПУ не сможет закончить выход в 0 по этой оси.	Установите нужное значение параметра
Неверное значение	Указанное значение не является верным значением для параметра данного типа	Введите правильное значение
Кинематика не выбрана или проверка кинематики выполнена неудачно	Значения параметров конфигурации не допустимы для данного типа кинематики	Измените значение параметров
Обнаружены циклы - порядок вычисления функций может быть неверным	В конфигурации имеются циклические соединения – возможен неожиданный порядок расчёта значений входов/выходов	
Логическая константа имеет неизвестное значение - будет установлен 0	Указано значение параметра типа Bool, отличное от поддерживаемых значений	Установите значение параметра в одно из значений 1,0, TRUE, FALSE
Значение параметра округлено до целого	В качестве значения целочисленного параметра указано вещественное число	
В конфигурации отсутствуют приводы	В конфигурации отсутствуют блоки Joint	Добавьте хотя бы один блок Joint в конфигурацию
Приводы портала имеют разные скорости поиска датчика исходного положения	При использовании портальной кинематики параллельно работающие приводы должны иметь одинаковые скорости поиска датчика нулевого положения	Установить одинаковое значение параметров SearchVel для указанных приводов

Приводы портала имеют разные сигналы разрешения ручного управления	При использовании портальной кинематики ручное управление параллельно работающими приводами должно включаться одним сигналом	Подключите нужный сигнал к входам JogEnable указанных приводов
Приводы портала имеют разный источник команды ручного управления	При использовании портальной кинематики команда ручного управления параллельно работающими приводами должна быть одинаковой	Подключите нужный сигнал к входам JogCommand указанных приводов
Приводы портала имеют разные ускорения	При использовании портальной кинематики параллельно работающие приводы должны иметь одинаковые максимальные ускорения	Установить одинаковое значение параметров MaxAcceleration для указанных приводов
Приводы портала имеют разные максимальные скорости	При использовании портальной кинематики параллельно работающие приводы должны иметь одинаковые максимальные скорости	Установить одинаковое значение параметров MaxVelocity для указанных приводов
В конфигурации должен присутствовать блок motmod	В конфигурации отсутствует блок motmod	Добавьте блок motmod в конфигурацию
К входу подключено более одного выхода	К входу блока допускается подключать только один выход	Удалите лишнее соединение
Найдено новое описание комплектации. Использовать его для этого проекта (соединения могут измениться)?	В конфигурации используется более старая версия описания комплектации, чем установлена в программе СервоМастер	Нажмите «Да» для использования новой версии описания комплектации. Нажмите «Нет» чтобы использовать версию описания комплектации из конфигурации
Конфигурация предназначена для более старой версии ПО СЧПУ. Точно использовать новую конфигурацию?	Описание комплектации в программе СервоМастер требует более новой версии ПО СЧПУ, чем описание комплектации в конфигурации	Нажмите «Да» для использования новой версии описания комплектации. Нажмите «Нет» чтобы использовать версию описания комплектации из конфигурации
Невозможно связаться с СЧПУ	Ошибка связи с СЧПУ	Проверьте сетевые настройки СЧПУ и компьютера
Операция выполнена – СЧПУ отключена	Запись конфигурации в СЧПУ выполнена успешно, для продолжения работы в режиме он-лайн необходимо повторно подключиться к СЧПУ	Выберите в главном меню программы пункт «СервоКон» - «Подключить»
Версия ПО СЧПУ 1.4.5 Для данной конфигурации необходима версия не ниже 1.5. Пожалуйста, обновите ПО СЧПУ.	Для работы данной конфигурации требуется версия ПО СЧПУ не ниже указанной	Обновите версию ПО СЧПУ и повторите действие
Произошла ошибка. Нажмите ОК чтобы сохранить отчет.	Произошёл сбой программы	Свяжитесь со службой технической поддержки

Пожалуйста, отправьте отчет об ошибке на e-mail: support@servotechnica.ru		
---	--	--

7. Приложения

7.1. Примеры конфигураций

В главе приведены примеры конфигураций СЧПУ СервоКон[©] для станка листового раскроя, фрезерного консольного станка и фрезерного портального станка. Приведены краткие характеристики станков, показаны структурные схемы конфигураций и примеры выполнения конфигураций в программе СервоМастер[©]. Все конфигурации доступны для скачивания на сайте www.servotechnica.ru в разделе «Документация» - «ЧПУ СервоКон» - «Программное обеспечение».

7.1.1. Станок для листового раскроя

Станок представляет собой консольный координатный стол, приводимый в движение двумя сервоприводами СПС. Ручное управление станком осуществляется с передней панели СЧПУ СервоКон[©]. В качестве инструмента на станке установлена плазменная горелка, логика управления которой, реализована в программе ПЛК plasma. Также возможно тестовое управление горелкой с передней панели СЧПУ СервоКон[©].

Каждая из осей X Y укомплектована аппаратными концевыми выключателями, подключенными к дискретным входам сервопривода оси X. Концевые выключатели ограничивающий движение в отрицательную сторону по осям X Y одновременно служат датчиками нулевого положения.

На станке установлены дополнительные кнопки аварийного останова, подключенные к дискретному входу IN.2 сервопривода оси Y.

На станке установлен датчик столкновения, подключенный к дискретному входу IN.1 сервопривода оси Y.

Станок оборудован системой вентиляции, которая включается автоматически при запуске программы или вручную с передней панели.

Структурная схема конфигурации станка для листового раскроя приведена на Рис. 69, конфигурация станка для листового раскроя, представленная в программе СервоМастер[©], приведена на Рис. 70.

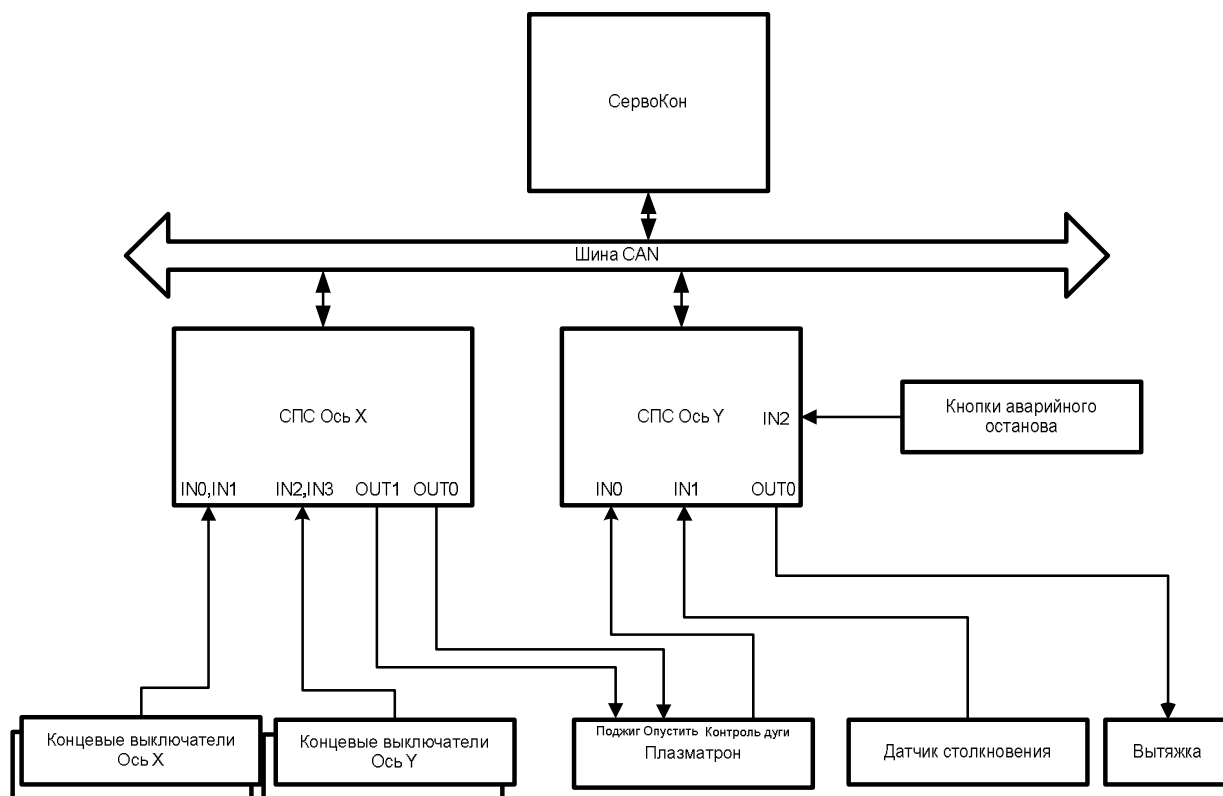


Рис. 69. Структурная схема конфигурации станка для листового раскроя.

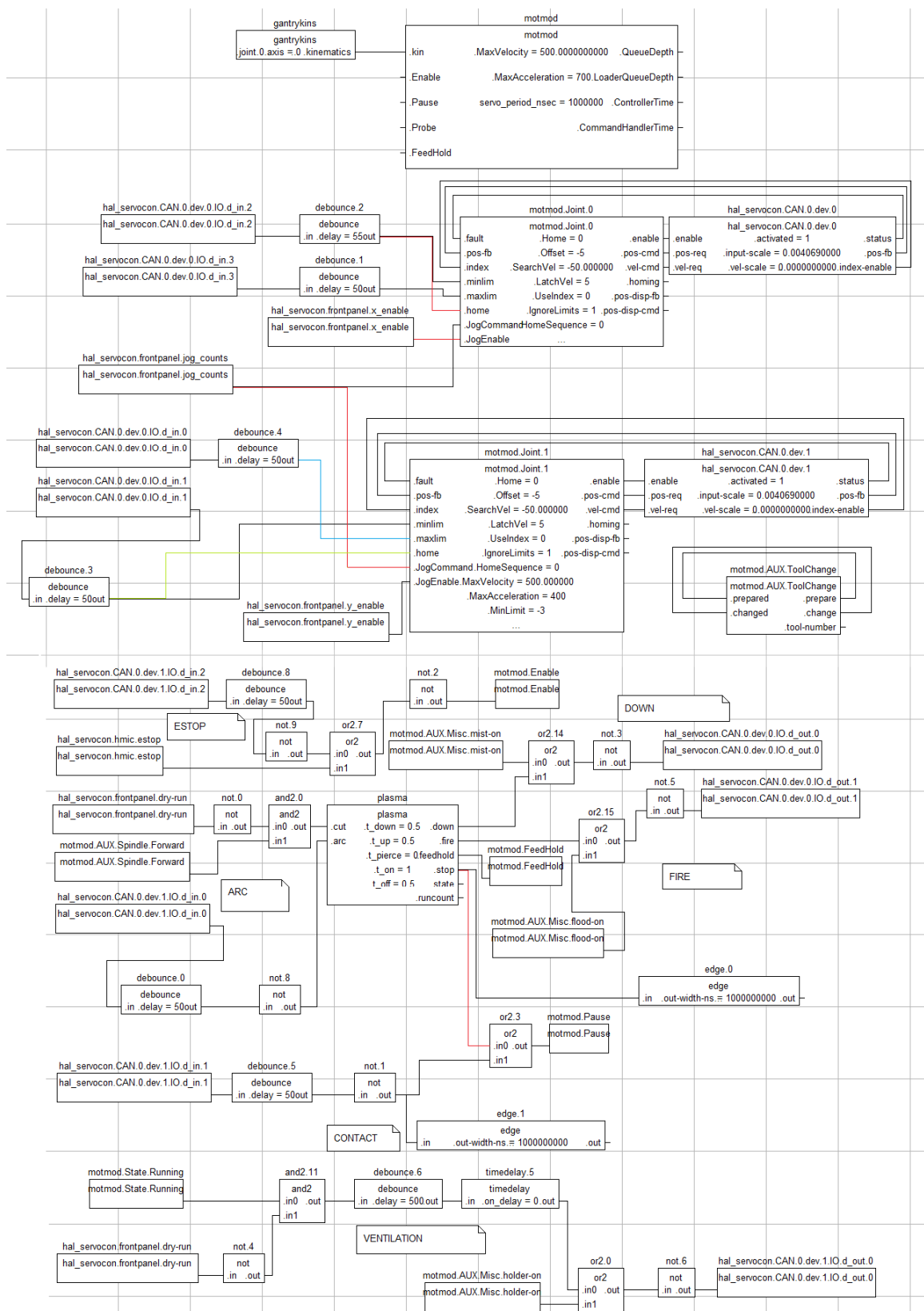


Рис. 70. Конфигурация станка для листового раскроя, представленная в программе СервоМастер[®].

7.1.2. Фрезерный консольный станок

Станок представляет собой консольный фрезерный станок, приводимый в движение тремя сервоприводами СПШ (или СПС). Станок также укомплектован пультом ручного управления для ручного перемещения осей. Шпиндель станка без обратной связи вращается в одну сторону, скорость регулируется сигналом 0-10В.

Каждая из осей X Y Z укомплектована аппаратными концевыми выключателями, подключенными к дискретным входам IN.0 и IN.1 сервоприводов соответствующей оси. Концевые выключатели «-» по осям X Y и «+» по оси Z одновременно служат датчиками нулевого положения. Для более точного поиска нулевого положения используется Z-метка энкодеров сервоприводов.

Структурная схема конфигурации фрезерного консольного привода приведена на Рис. 71, конфигурация фрезерного консольного станка, представленная в программе СервоМастер[©], приведена на Рис. 72.

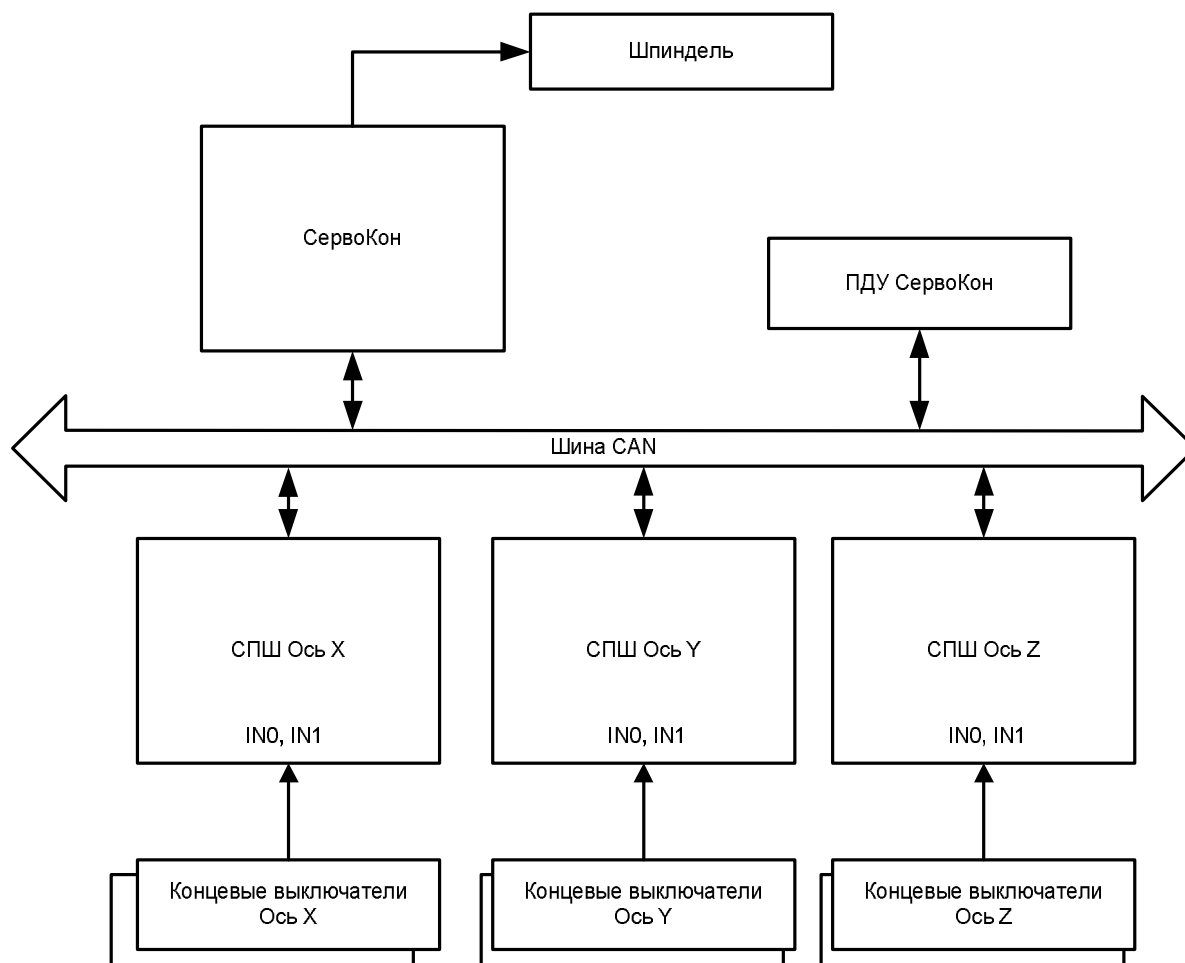


Рис. 71. Структурная схема конфигурации фрезерного консольного станка.

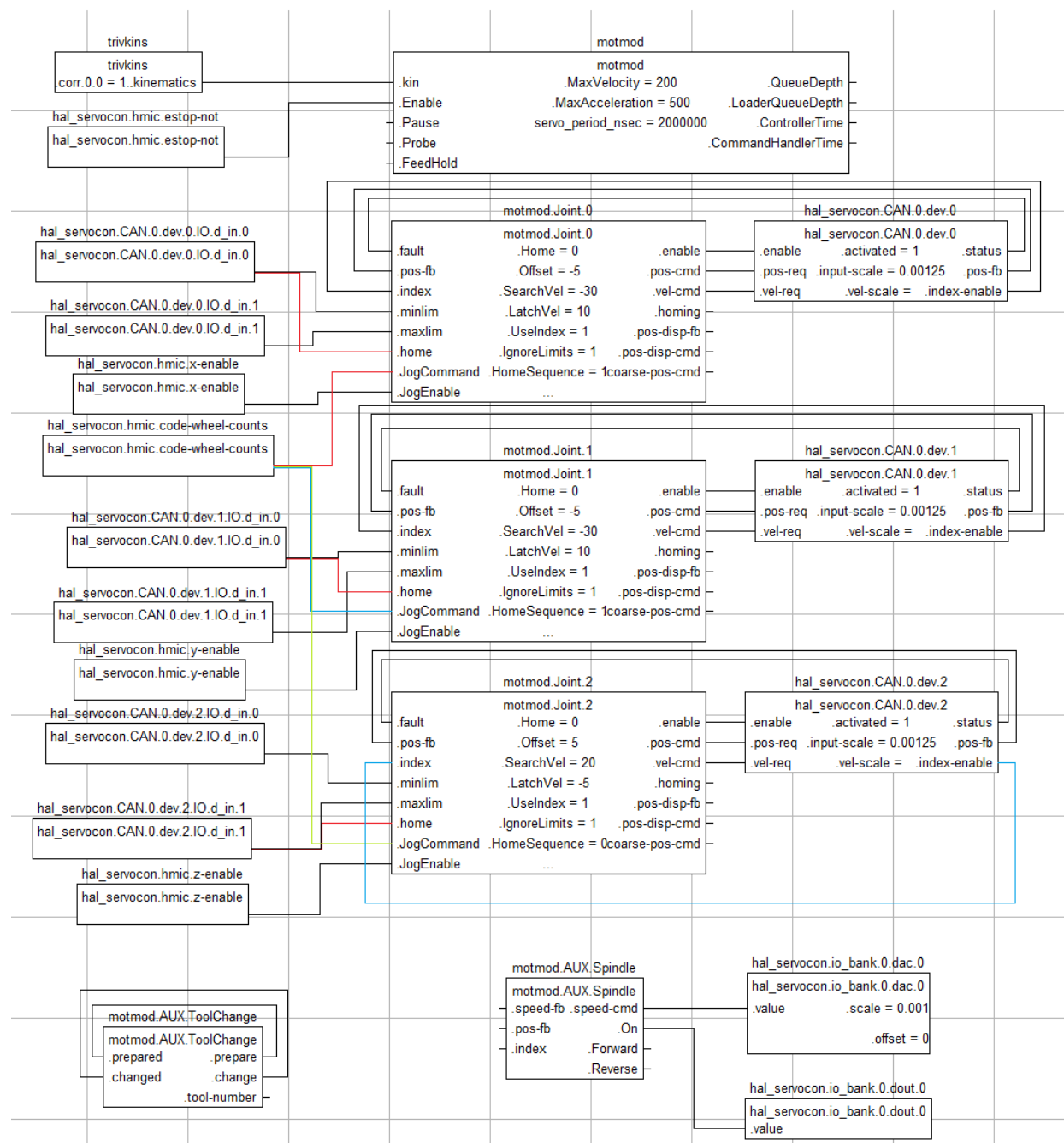


Рис. 72. Конфигурация фрезерного консольного станка, представленная в программе СервоМастер[®].

7.1.3. Фрезерный портальный станок

Станок представляет собой портальный фрезерный станок, приводимый в движение 4-мя сервоприводами СПШ (или СПС). Сервоприводы подключены к 2-м каналам CAN (по 2 привода на канал). Ось X приводится в движение двумя параллельно работающими сервоприводами.

Ручное управление станком осуществляется с передней панели. Шпиндель станка без обратной связи вращается в две стороны; направление вращения выбирается дискретным выходом (OUT.0 для вращения по часовой, OUT.1 для вращения против часовой), скорость регулируется сигналом 0-10 В. Привод шпинделя сигнализирует о выходе на заданную скорость установкой дискретного выхода, подключенного к входу IN.0 СЧПУ.

Станок оснащён системой подачи СОЖ, активируемой дискретным выходом OUT.2 СЧПУ. Каждая из осей X Y Z укомплектована аппаратными концевыми выключателями, концевые выключатели «-» по осям X Y и «+» по оси Z одновременно служат датчиками нулевого положения. Для более точного поиска нулевого положения используется Z-метка энкодеров сервоприводов. Станок оснащён датчиком касания для измерения длины инструмента или высоты заготовки, подключенным к входу IN.2 сервопривода оси Z.

Структурная схема конфигурации фрезерного портального приведена на Рис. 73, конфигурация фрезерного портального станка, представленная в программе СервоМастер[©], приведена на Рис. 74.

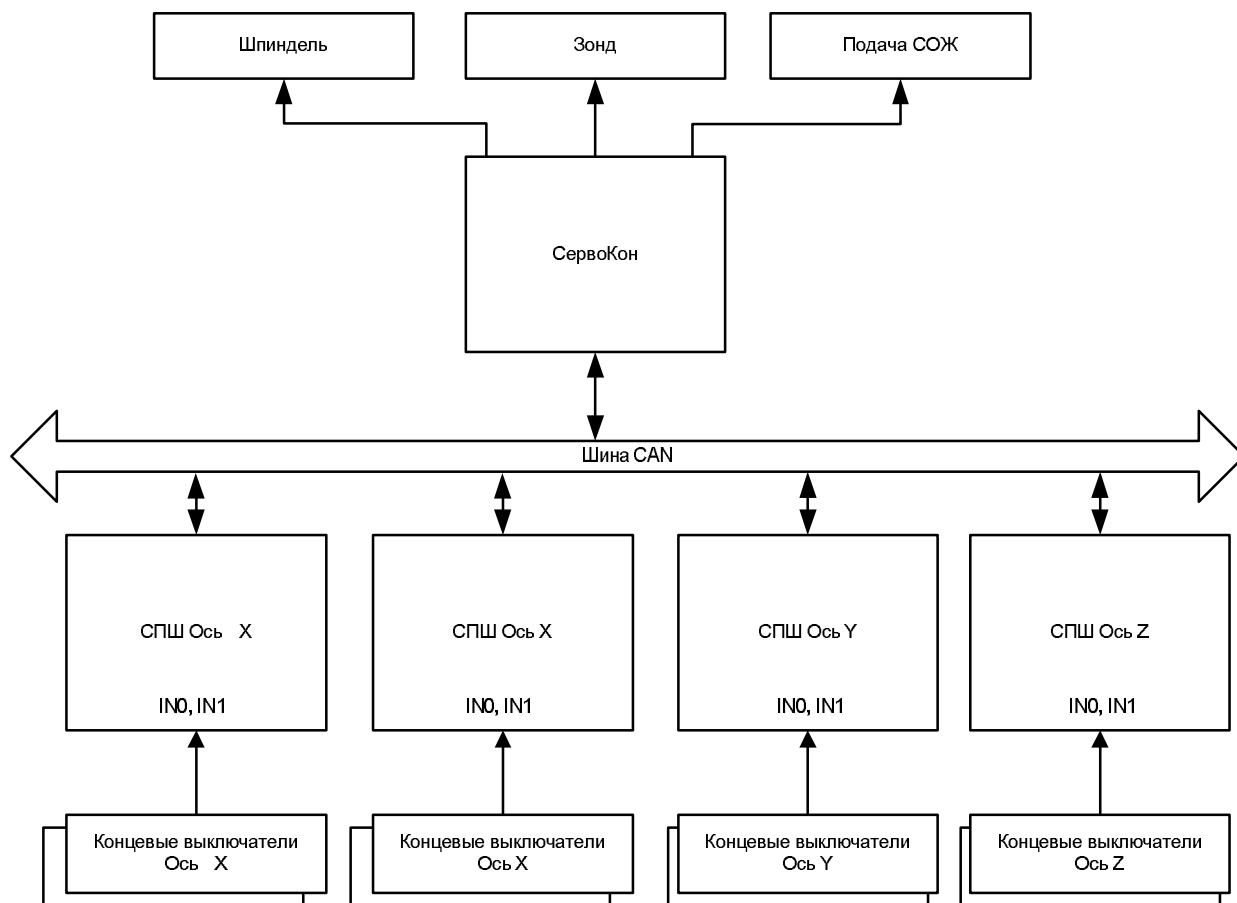


Рис. 73. Структурная схема конфигурации фрезерного портального станка.

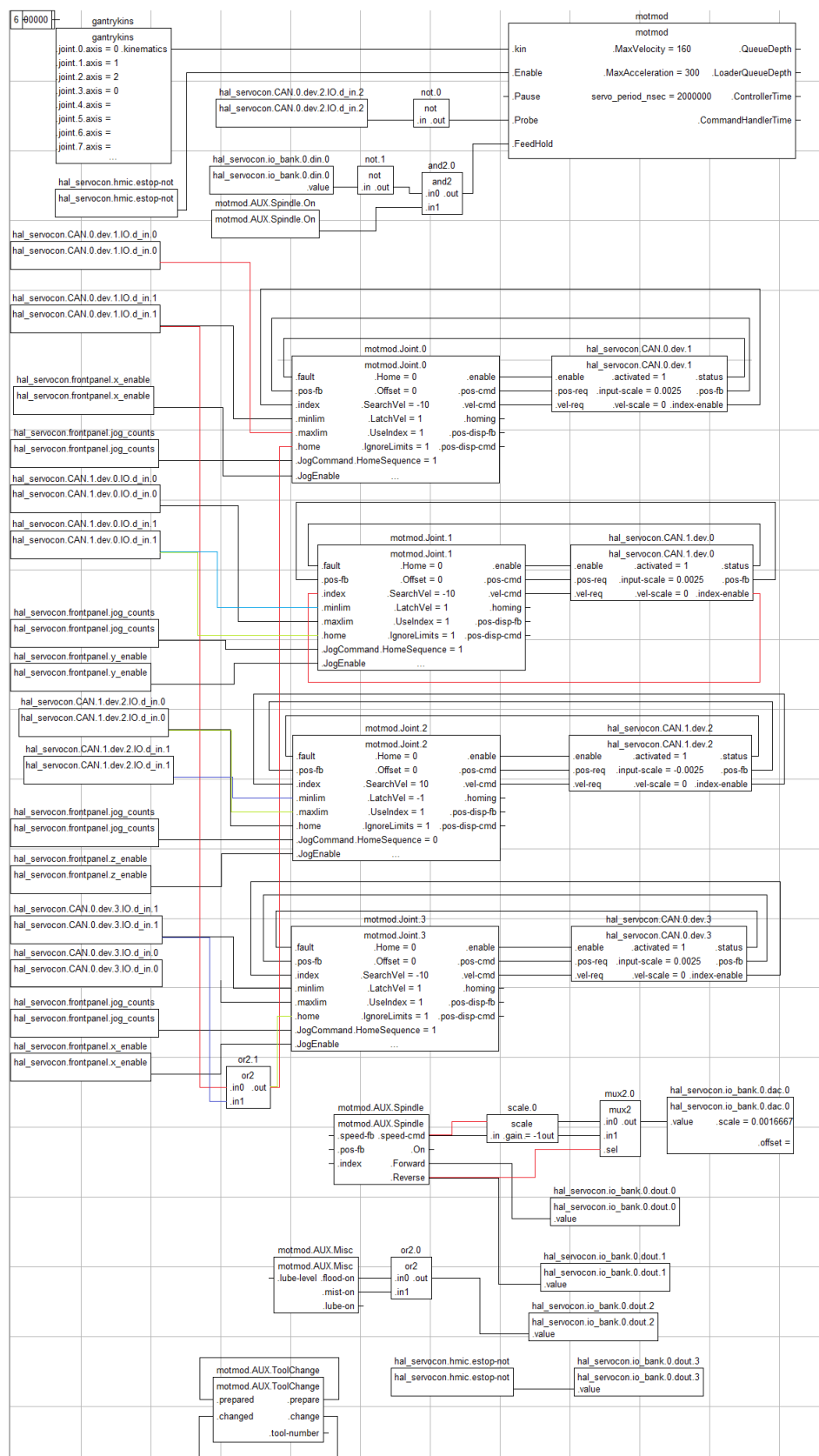


Рис. 74. Конфигурация фрезерного портального станка, представленная в программе СервоМастер[®].

7.2. Примеры программ ПЛК

7.2.1. Программа ПЛК для управления плазменной горелкой

```
#для создания элементов конфигурации
import hal
import os
import time
import plc
#отладчик
import rpdb2

class Plasma(plc.PLC):
    def InitIO(self):
        self.io.newpin("cut", hal.HAL_BIT, hal.HAL_IN)
        self.io.newpin("arc", hal.HAL_BIT, hal.HAL_IN)

        self.io.newpin("down", hal.HAL_BIT, hal.HAL_OUT)
        self.io.newpin("fire", hal.HAL_BIT, hal.HAL_OUT)
        self.io.newpin("feedhold", hal.HAL_BIT, hal.HAL_OUT)
        self.io.newpin("stop", hal.HAL_BIT, hal.HAL_OUT)

        self.io.newparam("t_down", hal.HAL_FLOAT, hal.HAL_RW)
        self.io.newparam("t_up", hal.HAL_FLOAT,
hal.HAL_RWself.io.newparam("t_on", hal.HAL_FLOAT, hal.HAL_RW)
        self.io.newparam("t_off", hal.HAL_FLOAT, hal.HAL_RW)
        self.io.newparam("t_pierce", hal.HAL_FLOAT, hal.HAL_RW)
        self.cutting = False

    def StopCut(self):
        self.io.feedhold = True
        self.io.fire = False
        self.io.down = False
        time.sleep(self.io.t_up)
        self.io.feedhold = False
        self.cutting = False

    def StartCut(self):
        self.io.feedhold = True
        self.io.down = True
        time.sleep(self.io.t_down)
```



```

        self.io.fire = True
        cutstart = time.time()
        while not self.io.arc:
            if time.time() - cutstart > self.io.t_on:
                self.io.stop = True
                self.StopCut()
                self.io.stop = False
                return False
        self.cutting = True
        return True

    def Run(self):
        #здесь реализуем логику ПЛК
        #запустим отладчик, когда ПЛК заработает правильно это можно уб-
        рать

        rpdb2.start_embedded_debugger('123', True, True, 0)
        while True:
            if self.io.cut:
                if not self.cutting:
                    self.StartCut()
                else:
                    if not self.io.arc:
                        self.StopCut()

if __name__=='__main__':
    plc = Plasma('plasma')
    plc.Start()

```

7.2.2. Программа ПЛК для вывода сообщений по внешним входам на дисплей

```

# -*- coding: utf-8 -*-
#кодировка файла должна быть utf-8
import hal
import plc
import sys
#средства CORBA
from omniORB import CORBA
import omniORB
import omniORB.any
import CosNaming

```

```
import CosNotification
import CosNotifyComm, CosNotifyChannelAdmin

class MessagePrinter(plc.PLC):
    def InitIO(self):
        #первый вход
        self.io.newpin('contact', hal.HAL_BIT, hal.HAL_IN)
        #второй вход
        self.io.newpin('arclost', hal.HAL_BIT, hal.HAL_IN)

    def SendMessage(self, message):
        #формируем сообщение
        typ = CosNotification.EventType("ServoCon", "User Message")
        fhdr = CosNotification.FixedEventHeader(typ, "Message")
        hdr = CosNotification.EventHeader(fhdr, [])
        #текст сообщения, которое будет напечатано
        msg = omniORB.any.to_any(message)
        fdata = [CosNotification.Property("Text", msg)]
        rdata = CORBA.Any(CORBA._tc_null, None)
        ev = CosNotification.StructuredEvent(hdr, fdata, rdata)
        #отправляем сообщение в канал сообщений
        self.event_consumer.push_structured_event(ev)

    def InitCORBA(self):
        #инициализируем ORB
        orb = CORBA.ORB_init(sys.argv, CORBA.ORB_ID)
        obj = orb.resolve_initial_references("NameService")
        root_context = obj._narrow(CosNaming.NamingContext)
        #получаем ссылку на канал сообщений
        name = [CosNaming.NameComponent("ServoConMessageChannel", "")]
        chan = root_context.resolve(name)
        chan = chan._narrow(CosNotifyChannelAdmin.EventChannel)
        #регистрация как генератор сообщений
        admin, id = chan.new_for_suppliers(CosNotifyChannelAdmin.AND_OP)
        etyp = CosNotifyChannelAdmin.STRUCTURED_EVENT
        consumer, id = admin.obtain_notification_push_consumer(etyp)
        ctyp = CosNotifyChannelAdmin.StructuredProxyPushConsumer
        self.event_consumer = consumer._narrow(ctyp)
        styp = CosNotifyComm.StructuredPushSupplier
        supplier = admin._narrow(styp)
```

```
#через этот объект будем посылать сообщения
self.event_consumer.connect_structured_push_supplier(supplier)

def Run(self):
    #дождаться инициализации
    self.wait_for_controller()
    #получить ссылку на канал сообщений
    self.InitCORBA()
    #инициализировать флаги, чтобы не передавать сообщения постоянно
    self.contact_msg = False
    self.arc_msg = False
    while True:
        if self.io.contact and not self.contact_msg:
            #анализируем первый вход и передаём сообщение
            self.SendMessage('Активен датчик столкновения')
            self.contact_msg = True
        if self.io.arclost and not self.arc_msg:
            #анализируем второй вход и передаём сообщение
            self.SendMessage('Потеря контроля дуги')
            self.arc_msg = True
        #если условие срабатывания пропало - сбросить флаг
        if not self.io.contact:
            self.contact_msg = False
        if not self.io.arclost:
            self.arc_msg = False

if __name__=='__main__':
    myplc = MessagePrinter('messageprinter')
    myplc.Start()
```

7.2.3. Программа дублирования кнопок клавиатуры внешними выносными кнопками

```
import hal
import plc
import sys
import threading
import time
```

```
from omniORB import CORBA, PortableServer
import CosNotifyComm__POA
import omniORB.any
import CosNaming
import CosNotifyComm, CosNotifyChannelAdmin
import ServoConRemote, ServoConRemote__POA
import CosNotification

class ORBThread(threading.Thread):
    def __init__(self, orb):
        threading.Thread.__init__(self)
        self.orb = orb
        self.done = False
    def run(self):
        while not self.done:
            if self.orb.work_pending():
                self.orb.perform_work()

class RemoteControl(plc.PLC):
    def InitIO(self):
        self.io.newpin('trace_back', hal.HAL_BIT, hal.HAL_IN)
        self.io.newpin('trace_fwd', hal.HAL_BIT, hal.HAL_IN)
        self.io.newpin('run', hal.HAL_BIT, hal.HAL_IN)
        self.io.newpin('stop', hal.HAL_BIT, hal.HAL_IN)

    def press_key(self, code):
        etype = CosNotification.EventType("ServoCon", "HMIC")
        fhdr = CosNotification.FixedEventHeader(etype, "KeyPressed")
        vhdr = [ ]
        hdr = CosNotification.EventHeader(fhdr, vhdr)
        msg = ServoConRemote.IHMIC.HMICMessage(code, 1)
        tc = CORBA.TypeCode(CORBA.id(ServoConRemote.IHMIC.HMICMessage))
        fdata = [ CosNotification.Property("Key", CORBA.Any(tc, msg)) ]
        r = CORBA.Any(CORBA._tc_null, None)
        ev = CosNotification.StructuredEvent(hdr, fdata, r)
        self.eventConsumer.push_structured_event(ev)

    def release_key(self, code):
        etype = CosNotification.EventType("ServoCon", "HMIC")
        fhdr = CosNotification.FixedEventHeader(etype, "KeyPressed")
```

```
vhdr = [ ]
hdr = CosNotification.EventHeader(fhdr, vhdr)
msg = ServoConRemote.IHMIC.HMICMessage(code, 0)
tc = CORBA.TypeCode(CORBA.id(ServoConRemote.IHMIC.HMICMessage))
fdata = [ CosNotification.Property("Key", CORBA.Any(tc, msg)) ]
r = CORBA.Any(CORBA._tc_null, None)
ev = CosNotification.StructuredEvent(hdr, fdata, r)
self.eventConsumer.push_structured_event(ev)

def start_back(self):
    self.press_key(0x12)

def stop_back(self):
    self.release_key(0x12)

def start_fwd(self):
    self.press_key(0x13)

def stop_fwd(self):
    self.release_key(0x13)

def run(self):
    self.press_key(0x14)
    time.sleep(0.5)
    #confirm start if not from first line
    self.press_key(0x04)

def stop(self):
    self.press_key(0x15)

def InitCORBA(self):
    orb = CORBA.ORB_init(sys.argv, CORBA.ORB_ID)
    poa = orb.resolve_initial_references("RootPOA")
    poaManager = poa._get_the_POAManager()
    poaManager.activate()
    naming_obj = orb.resolve_initial_references("NameService")
    naming_context = naming_obj._narrow(CosNaming.NamingContext)

    name = [CosNaming.NameComponent("ServoConKeyboardChannel", "")]
    try:
```

```

        obj = naming_context.resolve(name)
    except CosNaming.NamingContext.NotFound, ex:
        print "Event Channel {0} not found".format(name)
        raise
    channel = obj._narrow(CosNotifyChannelAdmin.EventChannel)
    supplier_admin, adminid = channel.new_for_suppliers(CosNotifyChannelAdmin.AND_OP)
    proxy_consumer, supplieradmin_proxy_id = supplier_admin.obtain_notification_push_consumer(CosNotifyChannelAdmin.STRUCTURED_EVENT)

    self.eventConsumer = proxy_consumer._narrow(CosNotifyChannelAdmin.StructuredProxyPushConsumer)
    supplier = supplieradmin_proxy_id._narrow(CosNotifyComm.StructuredPushSupplier)
    self.eventConsumer.connect_structured_push_supplier(supplier)

    self.orbThread = ORBThread(orb)
    self.orbThread.start()

def Run(self):
    self.wait_for_controller()
    self.run_pressed = False
    self.stop_pressed = False
    self.fwd_pressed = False
    self.back_pressed = False
    self.InitCORBA()
    while True:
        if self.io.trace_back:
            if not self.back_pressed:
                self.back_pressed = True
                self.start_back()
        else:
            if self.back_pressed:
                self.back_pressed = False
                self.stop_back()
        if self.io.trace_fwd:
            if not self.fwd_pressed:
                self.fwd_pressed = True
                self.start_fwd()
        else:
            if self.fwd_pressed:

```

```
        self.fwd_pressed = False
        self.stop_fwd()

    if self.io.run:
        if not self.run_pressed:
            self.run_pressed = True
            self.run()
        else:
            self.run_pressed = False

    if self.io.stop:
        if not self.stop_pressed:
            self.stop_pressed = True
            self.stop()
        else:
            self.stop_pressed = False

if __name__=='__main__':
    myplc = RemoteControl('remotecompl')
    myplc.Start()
```

7.3.Перечень модулей Python

A	B	C	D	E	F
abc	base64	cPickle	datetime	email	fcntl
aifc	bdb	cProfile	dbhash	encodings	filecmp
anydbm	binascii	cStringIO	dbm	errno	fileinput
argparse	binhex	cairo	decimal	exceptions	fnmatch
array	bisect	calendar	difflib		formatter
ast	bwidget	cgi	dircache		fpformat
asynchat	bz2	cgitb	dis		fractions
asyncore		chunk	distutils		ftplib
atexit		cmath	dl		functools
atk		cmd	doctest		future_builtins
		code	drv_libxml2		
		codecs	dsextras		
		codegen	dumbdbm		
		codeop	dummy_thread		
		collections	dum-		
		colorsys	my_threading		
		commands			
		compileall			
		compiler			
		configobj			
		contextlib			
		cookielib			
		copy			
		copy_reg			
		cracklib			
		crypt			
		csv			
		ctypes			
		curses			
G		H	I	J, K	L
gc	gobject	hashlib	ihooks	json	lcms
gconf	grp	heapq	imageop		lib2to3
gdbm	gsf	hershey	imaplib	keyword	libproxy
genericpath	gtk	hmac	imgchr		libxml2
getopt	gtkunixprint	hotshot	imp		libxml2mod
getpass	gzip	htmlentitydefs	importlib		libxslt
gettext		htmllib	inspect		libxsltmod
gio		httplib	io		linecache
git_remote_helpers			itertools		linux_event
rs					linuxaudiodev
glib					locale
glob					logging
gnomevfs					
M	N	O	P		Q, R
magic	netrc	omniORB	parser	pprint	quopri
mailbox	new	omniidl_be	pdb	profile	
mailcap	nf	opcode	pickle	propertywin-	random
markupbase	nis	operator	pickletools	dow	re

marshal
math
md5
mhash
mmap
modulefinder
multifile
multiprocessing
mutex

nntplib
numbers
numpy

optparse
os

pipes
pkg_resources
pkgutil
platform
plistlib
popen2
poplib
posix
posixfile
posixpath

pstats
pty
pwd
py_compile
pyclbr
pydoc
pydoc_data
pyexco
pyexpat

readline
repr
resource
rexec
rfc822
rlcompleter
robotparser
rpdb2
runpy

S

sched
select
sets
setuptools
sgmllib
sha
shelve
shlex
shutil
signal
simplejson
site
smtpd
smtplib
sndhdr
socket
spwd
sre
sre_compile
sre_constants
sre_parse

ssl
stat
statvfs
string
stringold
stringprep
strop
struct
subprocess
symbol
symtable
sys
sysconfig
syslog

T

tabnanny
tarfile
tdb
telnetlib
tempfile
termios
test
textwrap
this
thread
threading
time
timeit
toaiff
token
tokenize
trace
traceback
tty
types

U

unicodedata
unittest
urllib
urllib2
urlparse
user
uu
uuid

V

validate
vismach
vte

W

warnings
wave
weakref
webbrowser
whichdb
winpdb
wsgiref

X

xcgen
xdrlib
xml
xml2po
xmllib
xmlrpclib
xxsubtype

Y

yapps

Z

zeroinstall
zipfile
zipimport
zlib

ЗАО «Сервотехника» /// Россия /// Москва, Клары Цеткин ул., д. 33, корпус 35.

Тел.: +7 495 797-8866 /// info@servotechnica.ru /// www.servotechnica.ru

Контакты технической поддержки:

Тел.: +7 926 216 2121 /// support@servotechnica.ru