

WLDEV.RU

WLMill (Windows)

Содержание

1	Описание.....	4
1.1	Системные требования.....	4
1.2	Взаимодействие с контроллером.....	4
2	Внешний вид.....	4
2.1	Графическое окно.....	5
2.2	Программа	5
2.3	Положения	5
2.4	In/Out. Окно отображения входных/выходных сигналов	6
2.5	Менеджер сообщений.....	6
2.6	Управление. Главное окно управления.....	6
2.7	Панель макросов	6
2.7.2	Ручной режим.....	8
2.7.3	Скрипт.....	9
2.7.4	Пробинг.....	10
2.7.5	Инструмент.....	10
2.7.6	Сервис.....	10
3	Настройка программы	11
3.1	Настройка осей.....	11
3.1.1	Тип.....	12
3.1.2	Размер шага.....	12
3.1.3	Вход.....	12
3.1.4	Выходы.....	12
3.1.5	Подача.....	13
3.1.6	Выходные импульсы.....	13
3.1.7	Пределы.....	13
3.1.8	Поиск положения	13
3.1.9	S кривая	14
3.2	Настройка режимов оси.....	15
3.3	Настройка программы WLMill.....	16
3.3.1	Выход ШИМ.....	17
3.3.2	Корректировка S	17
3.3.3	Вход.....	17
3.3.4	Сглаживание.....	18
3.3.5	Штурвал.....	18
3.3.6	Подача	18
3.3.7	Люфт	18

3.4	Настройка ручного Энкодера (Whell).....	18
4	Типовые операции	19
4.1	Включение станка.....	19
4.2	Ручное перемещение.....	19
4.3	Установка начальных положений двигателя	20
4.4	Задание пределов перемещения оси.....	20
4.5	Установка текущего положения фрезы.....	20
4.6	Работа по программе	21
4.7	Поворот СК.	21
4.8	Работа со шпинделем.	21
4.9	Корректировка выхода S.....	22
4.10	Настройка штурвала	23
4.11	Редактирование программы.	24
5	Скрипты	24
5.1	MACHINE - основной элемент функционала	24
5.2	TIMER - таймеры	24
5.3	DIALOG - диалоги	25
5.4	FILE - работа с файлами.....	25

1 Описание.

Программа WLMill является бесплатной. Она используется для управления фрезерным станком с использованием контроллеров WLMotion.

1.1 Системные требования.

Программа WLMill работает в ОС Windows (x32, x64).

Минимальные системные требования:

- Windows 7
- Частота ЦП не ниже 1,4Гц
- Оперативная память не менее 1Гб
- Интерфейс USB.

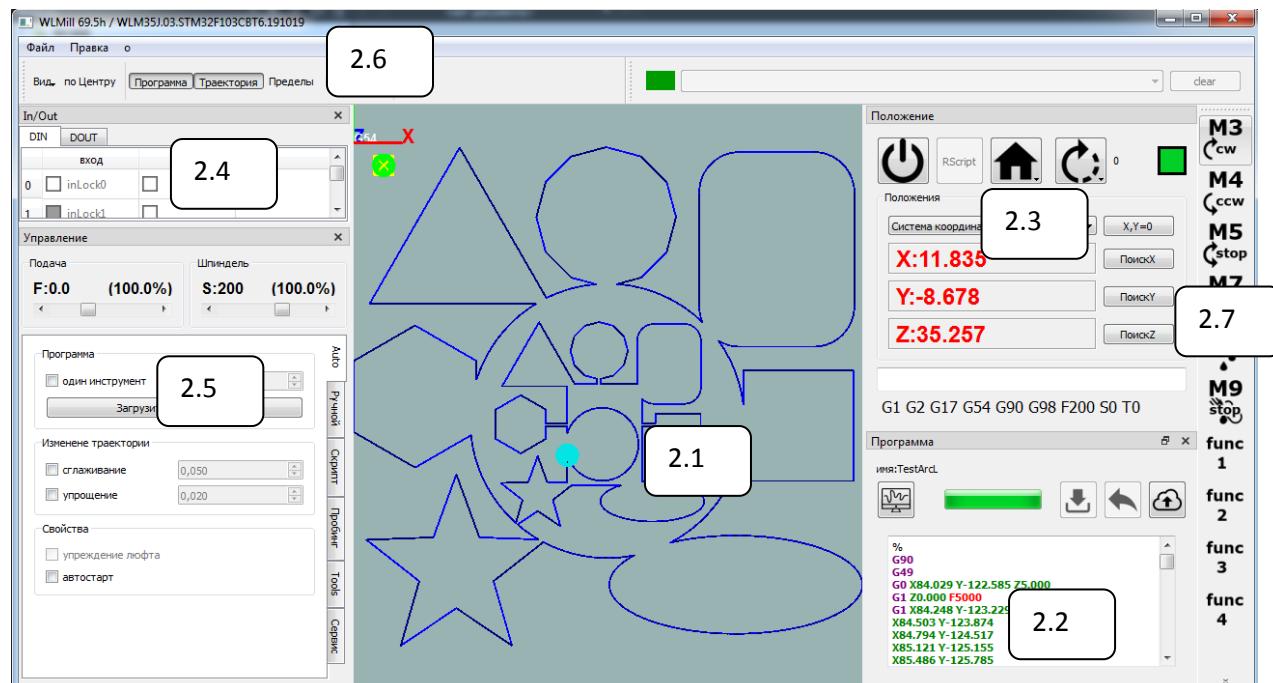
1.2 Взаимодействие с контроллером.

Данная программа сообщает контроллеру какие нужно выполнить действия, либо перемещения. А уже контроллер занимается расчётом траектории для каждой оси.

Например если нам нужно выполнить перемещения и в точку (X100 Y-1 Z50) то программа всего лишь сообщает контроллеру режим перемещения (скорости и ускорения) и конечное положение. Далее контроллер сам осуществляет движение. Аналогично происходит и с другими типами перемещения.

2 Внешний вид.

Программа имеет следующий внешний вид.



2.1 Графическое окно.

Данное окно расположено в центре программы. В нём отображается траектория программы, положение баз, положение фрезы и пр. Для работы с данным окно используется:

- Средняя кнопка мыши + движение - вращение вида
- Колесо - фокусированное увеличение/уменьшение
- Ctrl + средняя кнопка + движение - перетаскивание вида
- Левая кнопка мыши - выбор элемента (для небольших программ)

Для ориентации вида используется кнопки видов сверху, а также кнопки для скрытия траектории движения и программы и пр..

2.2 Программа.

Отображает текст текущей программы в двух режимах.

- режим текста - для небольших программ
- режим строк - для больших программ

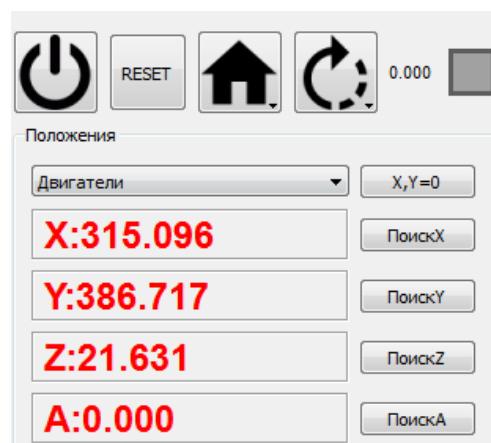
Назначение кнопок:

-  Обновить - обновляет (перестраивает) вид программы в графическом окне.
-  Принять - запоминает внесенные изменения в программу (текстовый режим)
-  Восстановить - восстанавливает изменения в программе (текстовый режим)
-  Перезагрузить - перезагружает программу с жёсткого диска

2.3 Положения.

Отображает положения осей станка.

В данном окне можно выбрать что именно отображать: положение осей в СК или положение двигателей.



Изменения значений производится с помощью двойного нажатия.

Назначение кнопок:

-  Включение/ выключение станка.
-  - произвести перемещение в точку Home (или задание положения точки при удержании кнопки)
-  Поворот СК - поворот СК (относительно базовой точки, см 4.7)
- Поиск (XYZ) - поиск соотв. оси
- X,Y=0 - установка значений X и Y в ноль

2.4 In/Out. Окно отображения входных/выходных сигналов.

В данном окне отображается текущее состояние входов и выходов на соотв. вкладках.

Для реверса входа/выхода необходимо произвести двойное нажатие на квадрат который находится правее. Если реверс установлен то квадрат будет иметь красный цвет, иначе белый.

Для переключения выхода необходимо произвести двойное нажатие на квадрат который находится левее.

Закрашенный квадрат означает активность инверсии либо логическую единицу на соотв. входе или выходе.

В случае если вход/выход неактивен (занят другим модулем) то он будет иметь серый цвет.

2.5 Менеджер сообщений.

Окно используется для отображения важных сообщений которые поступают во время работы станка от контроллера. В нём ведутся записи с фиксацией времени, пояснения, и сколько подобных сообщений было.

2.6 Управление. Главное окно управления.

Вверху данного окна располагается два корректора (рабочая подача и вращения шпинделя).

По середине набор закладок:

- Auto - автоматическое движение
- Ручной - ручные перемещения
- Скрипт - работа со скриптами
- Пробинг - поиск касанием
- Tools - инструменты
- Сервис - сервис станка

2.7 Панель макросов

На панели макросов имеются кнопки часто используемых макросов. Также имеются кнопки запуска макросов пользователя. Иконки данных макро

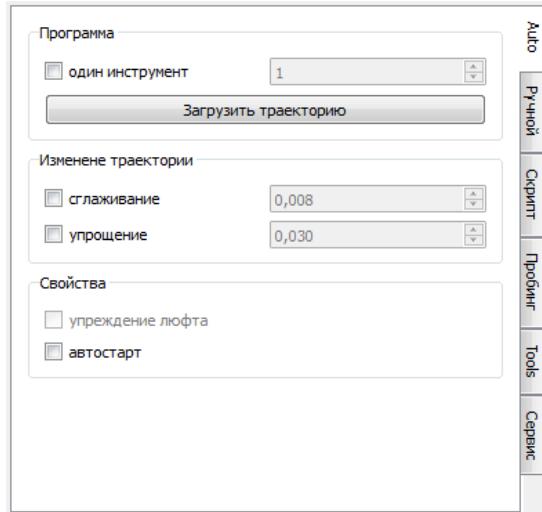
2.7.1.1 Корректоры. F и S.

Данные корректоры используются для корректировки заданных значений рабочей подачи и частоты вращения шпинделя в реальном времени.

Возможна корректировка с помощью клавиатуры при установке фокуса ввода (нужно нажать мышкой на бегунок соотв. корректора)

Для возвращения корректора в 100% необходимо двойное нажатие на проценты.

2.7.1.2 Auto. Закладка автоматического перемещения.



На данной закладке есть несколько блоков:

2.7.1.3 Программа.

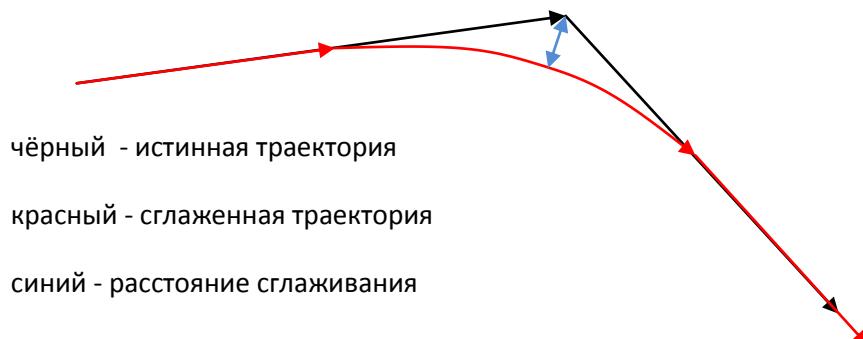
Используется при работе с программой.

- Один инструмент - если необходимо выполнить программу для одного инструмента.
- Загрузить траекторию - загрузить траекторию программы для последующего выполнения.

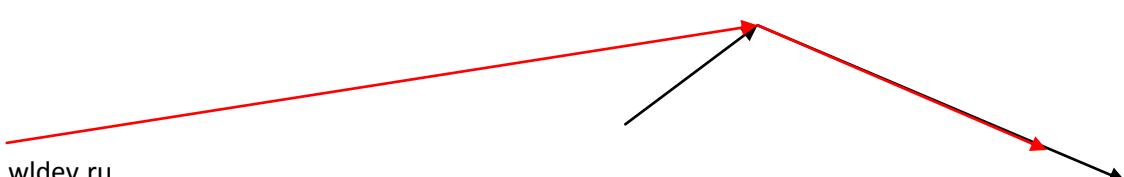
2.7.1.4 Изменение траектории.

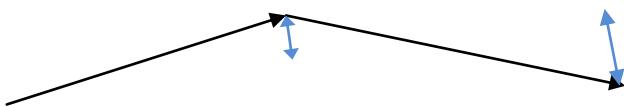
В данном блоке мы устанавливаем параметры изменения траектории.

- сглаживание - это максимальное расстояние между кривой сглаживания и вершиной траектории программы (G64 параметр P).



- упрощение - это максимальное расстояние для отклонения траектории от упрощённой траектории (G64 параметр Q)..





чёрный - истинная траектория

красный - упрощённая траектория

синий - расстояние максимального упрощения

2.7.1.5 Свойства.

В данном блоке имеется свойства для автоматического движения.

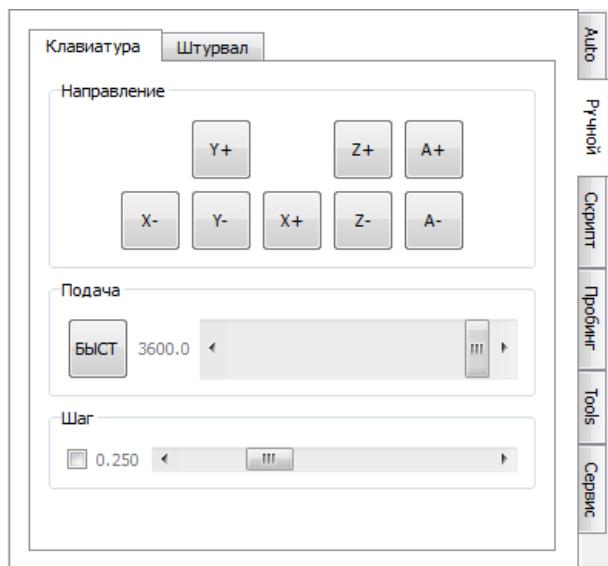
- автостарт - начинать движение сразу после получения данных в буфер перемещения (например при ручном вводе G кода)

2.7.2 Ручной режим.

Данная вкладка используется для работы в ручном режиме.

2.7.2.1 Клавиатура.

Перемещения с помощью клавиатуры.



В блоке направления находятся кнопки перемещения соотв. координат. (X,Y - стрелки на клавиатуре, Z - PageUp/PageDown)

Для движения от клавиатуры, кнопки должны находиться в "фокусе ввода". Когда движение возможно от нажатия на кнопки от клавиатуры - кнопки активны.

Для быстрых перемещений необходимо нажать "shift".

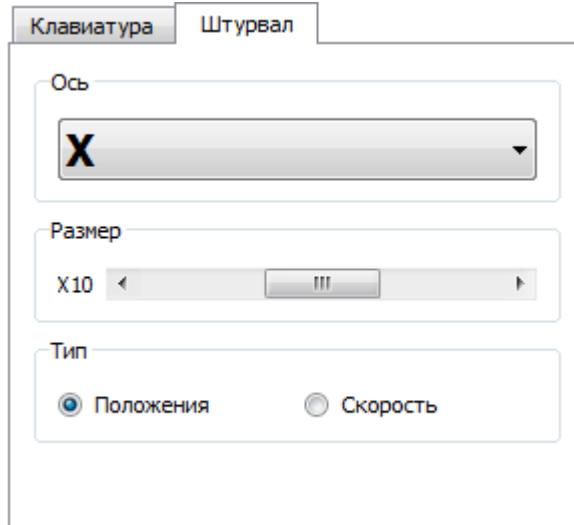
Для шаговых перемещений необходимо установить галочку в блоке "шаг" и установить размер одного шага (при необходимости).

При нажатии кнопок "+" или "-" - произойдёт изменения шага единичного перемещения. Если шаг уже был максимальный то установиться непрерывное движение. И наоборот если было

установлено непрерывное движение и был нажат "-" - установится пошаговое движение с максимальным шагом.

2.7.2.2 Штурвал.

Движение от штурвала (ручной энкодер). В зависимости от того какие элементы управления были выбраны в для переключения режимов работы штурвала, будут доступны соотв. элементы (оставшиеся)



В соотв. блоке "ось" выбираем ось для перемещения с помощью мыши или клавиатура клавишами 1(X),2(Y),3(Z)...

В блоке "размер" устанавливаем размерность одного оборота штурвала. Т.е. на сколько должна переместиться ось за один оборот штурвала.

Выбираем тип перемещения:

- Положение - двигатель будет перемещаться в положение заданное штурвалом
- Скорость - двигатель прекратит позиционирование если штурвал не вращается

Если в режиме "Положение" мы будем задавать перемещения (крутить штурвал) быстрей чем может перемещаться ось. То ось будет накапливать перемещения (отставать от штурвала), и если мы резко перестанем крутить штурвал, то ось продолжит своё движение на заданную позицию (будет догонять позицию). Чтобы этого не было существует режим скорость в нём привод будет останавливаться сразу после того как мы перестали крутить колесо.

2.7.3 Скрипт.

Вкладка для работы со скриптами.

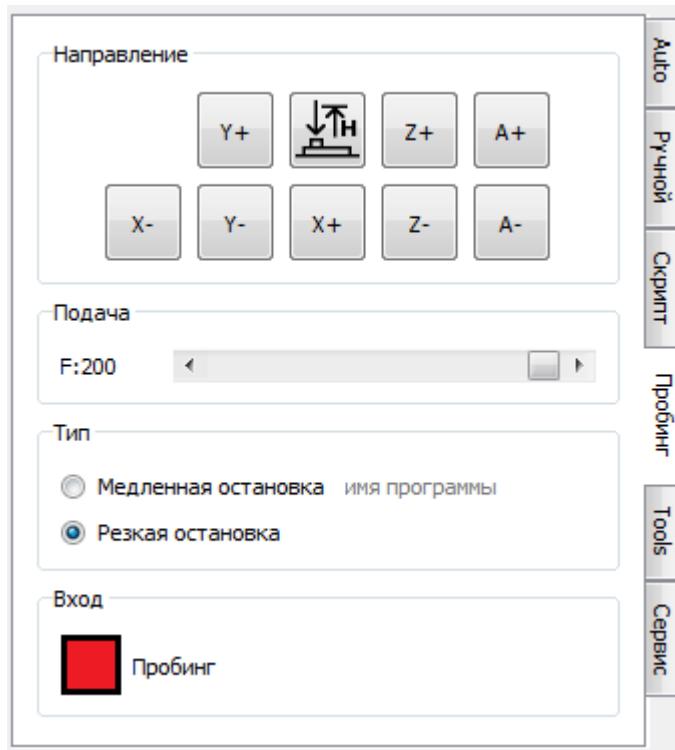
На вкладке расположены кнопки:

- Правка скриптов - редактирование скриптов
- Сброс Выполнения - прерывание выполнения текущего скрипта. Также эта кнопка расположена в окне положения. RScript
- Стока ввода для выполнения команды

О скриптах будет рассказано подробно в WLMill.Script.

2.7.4 Пробинг.

Вкладка используется для поиска положения детали или заготовки.



В блоке направление задаётся направление поиска.

В блоке подача задаётся скорость поиска.

В блоке тип выбирается:

- медленная остановка - после того как произошло изменение состояния входа inProbe двигатель медленно остановится, а затем вернется на позицию где произошло изменение состояния.
- Резкая остановка - двигатель мгновенно остановится при изменение

В блоке "Вход" отображается текущее состояние входа inProbe. Рекомендуется проверить правильное срабатывания перед началом пробинга.

2.7.5 Инструмент.

2.7.6 Сервис.

Вкладка выполнения сервисных операций .

В блоке ось выполняются действия для конкретной оси:

- Обучить - произвести обучение оси по датчикам (найти положение датчиков по двигателю). Чтобы потом находить это положение оси.(находить положение двигателя по датчикам)
- Найти - произвести поиск оси (см выше) .
- Аннулировать поиск - перевести ось в состояния "неизвестного положения" при котором не работают софт лимиты.

С помощью блока "Обучить ось" происходит поиск положения датчиков оси для последующего использования этих данных для поиска положения оси.

3 Настройка программы

Для настройки программы станок должен быть выключен (п 4.1).

Настройка программы осуществляется в несколько этапов:

- Настройка каждой оси (размерности, типы сигналов итд)
- Настройка режимов перемещения каждой оси (скорость, ускорение)
- Настройка программы параметров WLMill

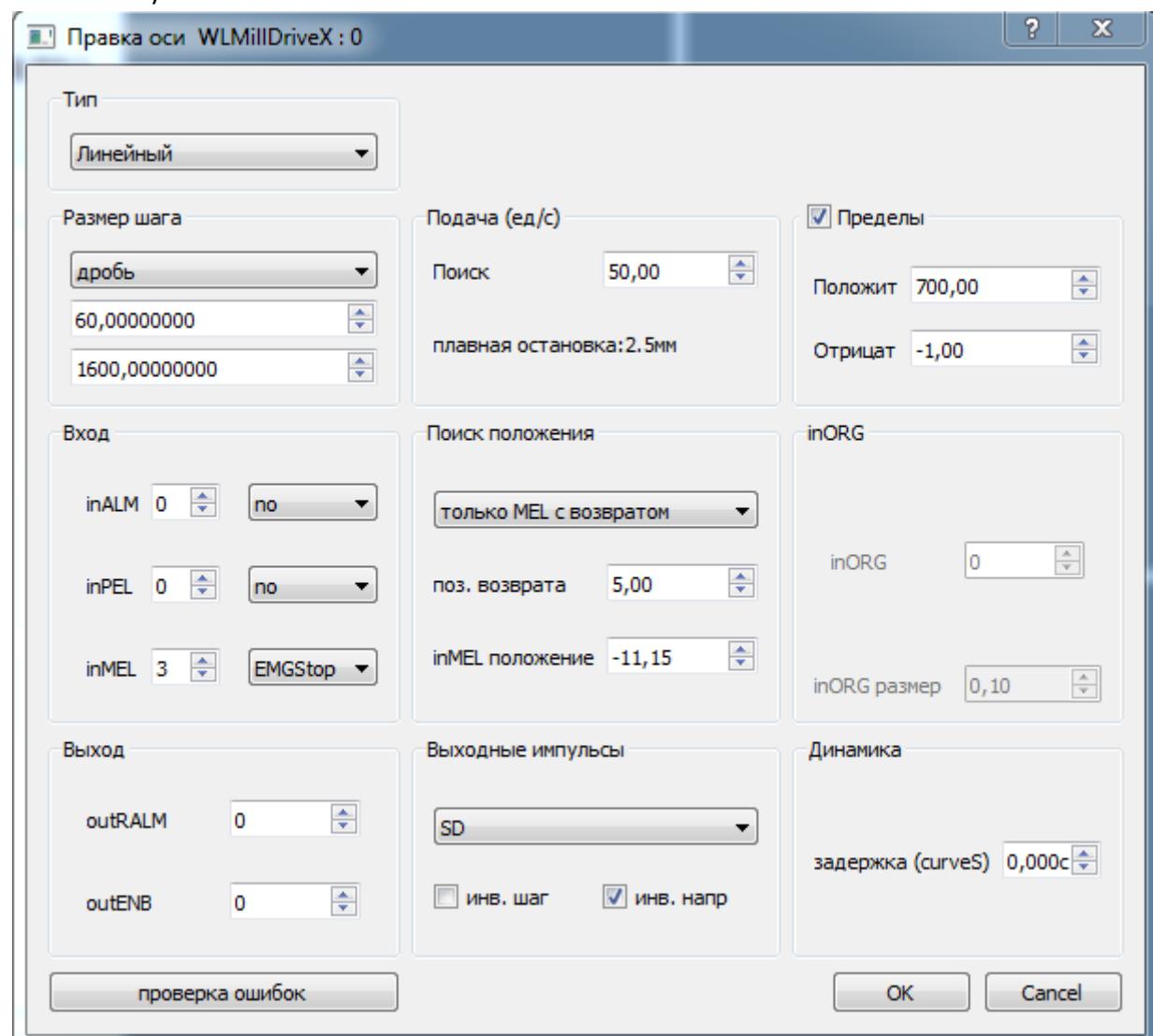
Перед настройкой мы должны подключить датчики, кнопки, концевые датчики, выходные сигналы, а также драйвера приводов к контроллеру WLMotion. Как это сделать смотрите документацию к контроллеру. Для безопасной работы должен быть подключен вход inEMGStop либо inSDStop. Убеждаемся в работе входов с помощью окна In/Out (п 2.4).

В процессе обновления программы поставляются в том числе и файлы конфигурации, чтобы ваша конфигурация сохранялась, необходимо НЕ КОПИРОВАТЬ файлы расположенные в папке millconfig/

3.1 Настройка осей.

Настройка осей осуществляется с помощью меню "Правка" в которой мы выбираем соответствующую Ось (X,Y,...).

После чего у нас появится меню:



3.1.1 Тип

В данном окне выбирается тип оси. Линейный или поворотный.

3.1.2 Размер шага.

В данном блоке мы устанавливаем размер единичного перемещения оси, либо какое количество шагов приходится на единицу (мм, град) перемещения оси (в данном случае Drive0), либо дробное число.

Пример расчёта:

Дано:

Шаговый двигатель 200 Шагов на оборот .

Контроллер ШД с множителем 8.

Шарико-винтовая пара с шагом 5 мм.

Тогда $8 * 200 / 5 = 320$ шагов на 1 мм

При выборе размера шага стоит учитывать:

1. Способен ли будет воспринимать драйвер двигателя максимальную частоту перемещения.
2. При больших делителях, у шаговых двигателей момент может уменьшаться.
3. Точность позиционирования при увеличении коэффициента деления шагов ШД не повысится (она определяется параметрами ШД), изменится только плавность перемещения.
4. Лучше выбирать максимальные частоты работы не выше половины максимальной частоты контроллера. Тогда изменение частоты перемещения будет более плавным.

3.1.3 Вход.

В данном блоке мы указываем приводу какие входные сигналы к нему подключены

- inALM - вход ошибки контроллера привода
- inPEL - положительный концевой датчик
- inMEL - отрицательный концевой датчик
- inORG - датчик оригинальной позиции привода (используется для поиска положения оси)

напротив каждого входа имеется меню для установки реакции (что должно произойти) на датчик:

- noAct - игнорировать (нет реакции)
- EMGStop - мгновенная остановка
- SDStop - плавная остановка

Если ось находится в интерполяции с другими осями то и другие оси будут остановлены (SDStop или EMGStop) соответствующим способом.

Внимание!!! : Вход 0 и Вход 1 - это статические (виртуальные) входа!!!

Если мы не используем какой либо вход то нужно установить для него статический вход 0.

3.1.4 Выходы.

В данном блоке мы указываем какие выходы может использовать привод (0 - никакой)

- outRALM - выход сброса ошибки контроллера привода
- outENB- выход разрешения работы контроллера привода

Внимание!!! : Вход 0 - это статический (виртуальный) выход.!!!

Если мы не используем какой либо выход то нужно установить для него статический выход 0.

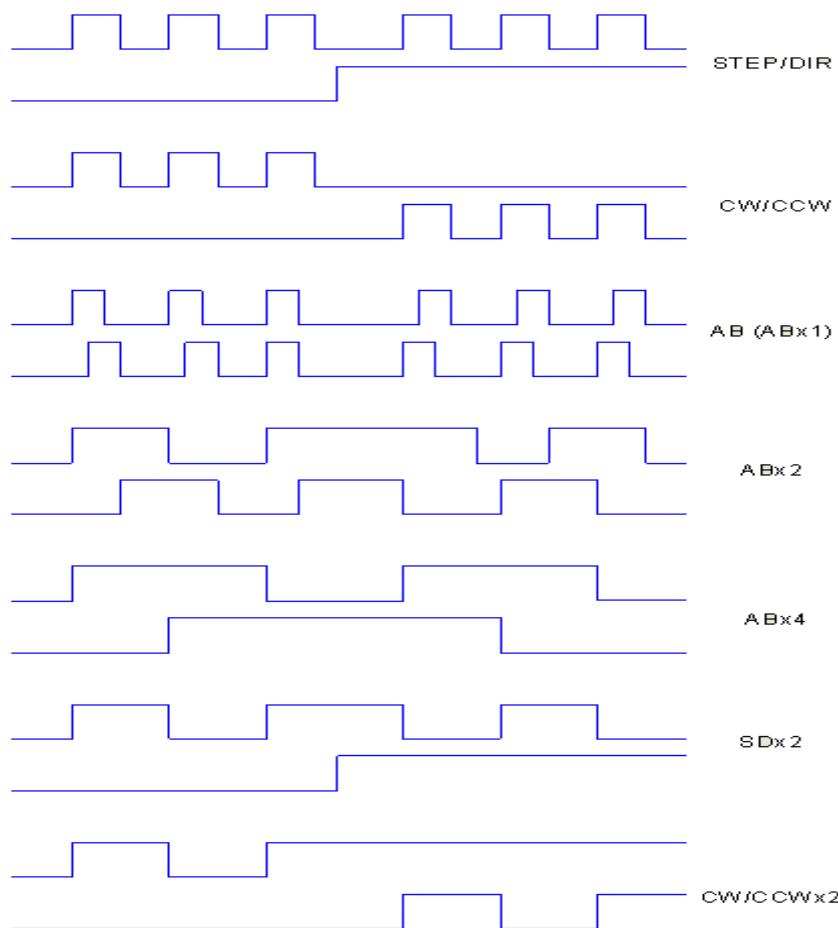
3.1.5 Подача.

В данном блоке мы указываем подачи на которых будет перемещаться привод

- Поиск - подача поиска

3.1.6 Выходные импульсы.

В данном блоке мы выбираем тип выходных, а также можем произвести реверс сигналов и направления перемещения. Как правило используется SD (Step/Dir)



3.1.7 Пределы.

В данном блоке мы указываем пределы перемещения (софт лимиты) для каждой оси. Если они не используются то нужно убрать галочку с этого блока.

Если ось является поворотной то при отсутствие пределов она будет производить перемещение по наименьшему пути. Из расчёта оборота 360 градусов.

3.1.8 Поиск положения

В данном блоке мы выбираем метод с помощью которого ось будет находить своё положение после включения станка либо после аварийного останова.

Существует несколько методов поиска:

- noFind - не производится поиск положения данной оси
 - onlyPEL(back)/onlyMEL(back) - производится поиск по положительному/отрицательному датчику оси.
 - onlyORG(back) - производится поиск по одному датчику ORG. При выборе этого метода необходимо указать номер входа сигнала и размер метки в мм (градуса) если ось линейная(поворотная).
- back (с возвратом) - означает, что после проведения поиска ось автоматически будет с позиционирована в "поз.возврата".

ВНИМАНИЕ!!! при поиске с использованием сигналов inPEL/inMEL должна быть выставлена реакция остановки для соотв. входа. Эта реакция будет использована и при поиске/обучение.

порядок поиска onlyORG:

- ось установлена так чтобы датчик inORG был в активном состоянии.
- ось выходит из датчика из датчика (используется параметр inORG размер).
- останавливается.
- ось проходит назад через два фронта датчика
- останавливается.
- ось проходит вперед через два фронта датчика
- останавливается.
- Берутся позиции срабатывания датчика. Если обучение то оно запоминается если поиск то корректируется текущее положение оси.
- если имеется "back" то происходит позиционирование в это положение.

порядок поиска onlyPEL/onlyMEL:

- ось начинает движение к соотв. датчику
- по срабатыванию ось останавливается согласно реакции (см **пОшибка! Источник ссылки е найден.**)
- Берутся позиции срабатывания датчика. Если обучение то оно запоминается если поиск то корректируется текущее положение оси.
- если имеется "back" то происходит позиционирование в это положение.

После выбора типа поиска положения мы можем ввести или изменить положение датчика на нашей оси (inXX положение). Также мы можем найти положение датчика нашей оси используя режим обучения. 4.3

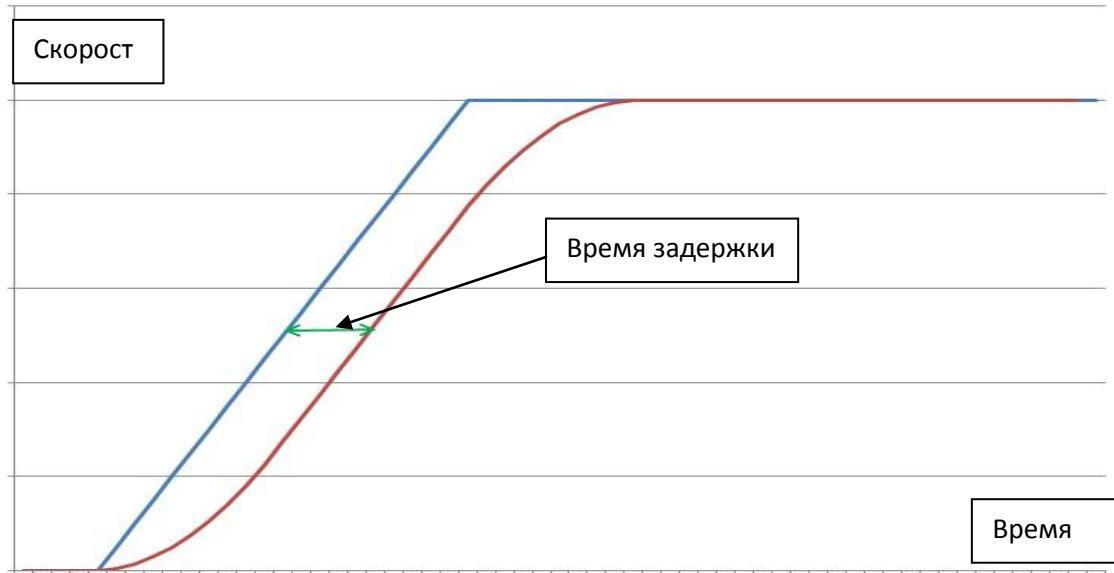
3.1.9 S кривая

Данный параметр отвечает за инерционность конкретной оси. При больших значениях может существенно менять траекторию инструмента.

Данный параметр устанавливается для каждой оси отдельно.

Численно он равен времени за которое ускорение оси измениться с текущего до целевого.

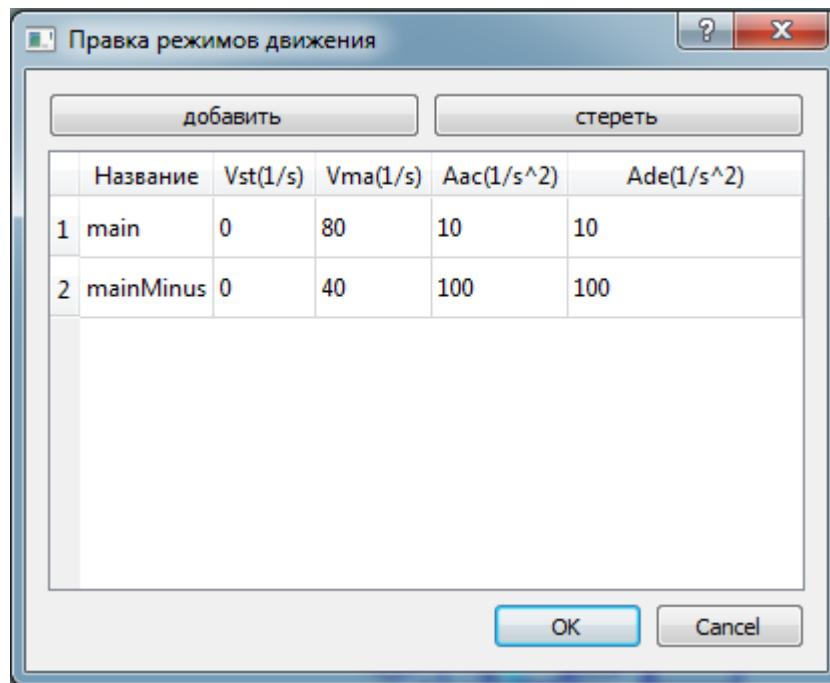
На графике ниже показаны два графика. Синий - время задержки равно нулю. Красный - время задержки не равно нулю.



Рекомендуется устанавливать не более 0,050с. (50мс)

3.2 Настройка режимов оси.

Для настройки режимов оси необходимо выбрать меню Правка->режимы перемещения. После чего у нас появится окно со списком режимов. В котором в виде таблицы представлены режимы перемещения оси.

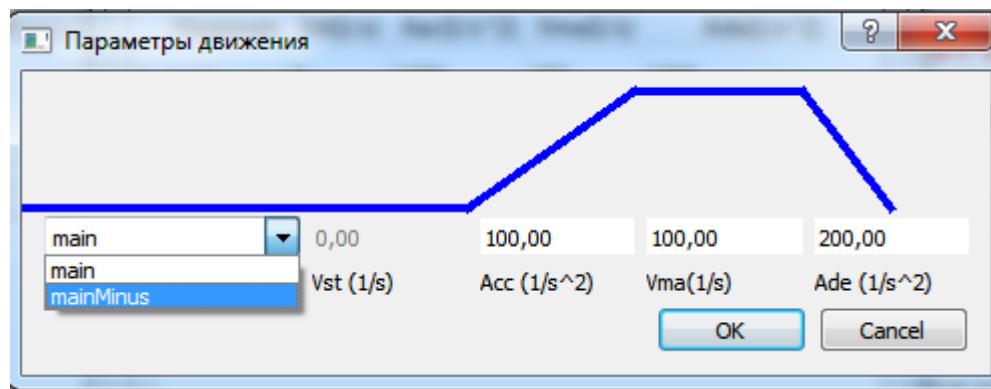


Для добавления или удаления используются соотв. кнопки.

Для каждой оси должен присутствовать режим "main" - основной режим.

В случае наличия режима mainMinus, ось будет осуществлять перемещения в положительном направлении на режиме "main". А в отрицательном "mainMinus".

Для правки режима необходимо произвести двойное нажатие на соотв. строку. После чего появится окно правки выбранного режима:



В окне мы вводим параметры движения:

- V_{st} - стартовая скорость, с которой начнется движение и далее привод будет ускоряться до максимальной.
- V_{ma} - Максимальная скорость - максимальная скорость перемещения
- Acc/Ade - соотв. ускорение и замедление можно ввести двумя способами (ускорение, время)

Для наглядности вверху изображён график изменения скорости от времени. При правке какого-либо параметра выделяется участок графика за который он отвечает.

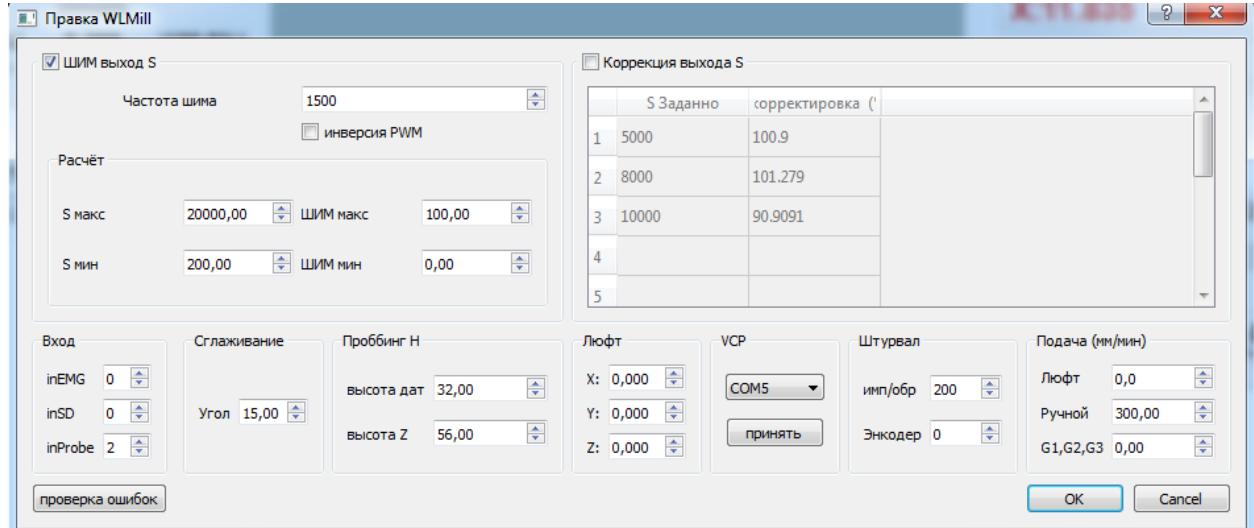
Важно!!! после указания размерности шага, для обеспечения правильной интерполяции в координатах XYZ WLMill приводит все эти размерности к наименьшей из осей.

То есть, если у нас: X 800 шагов/мм, Y 800 шагов/мм, Z 1600 шагов/мм. То максимальная возможная скорость для осей XYZ будет = (максимальная частота перемещения) / 1600.

Для WLM35A ($step/dir$) = $75000/1600 = 46\text{мм/с}$

3.3 Настройка программы WLMill.

Настройка программы осуществляется с помощью меню "Правка" в которой мы выбираем "WLMill".

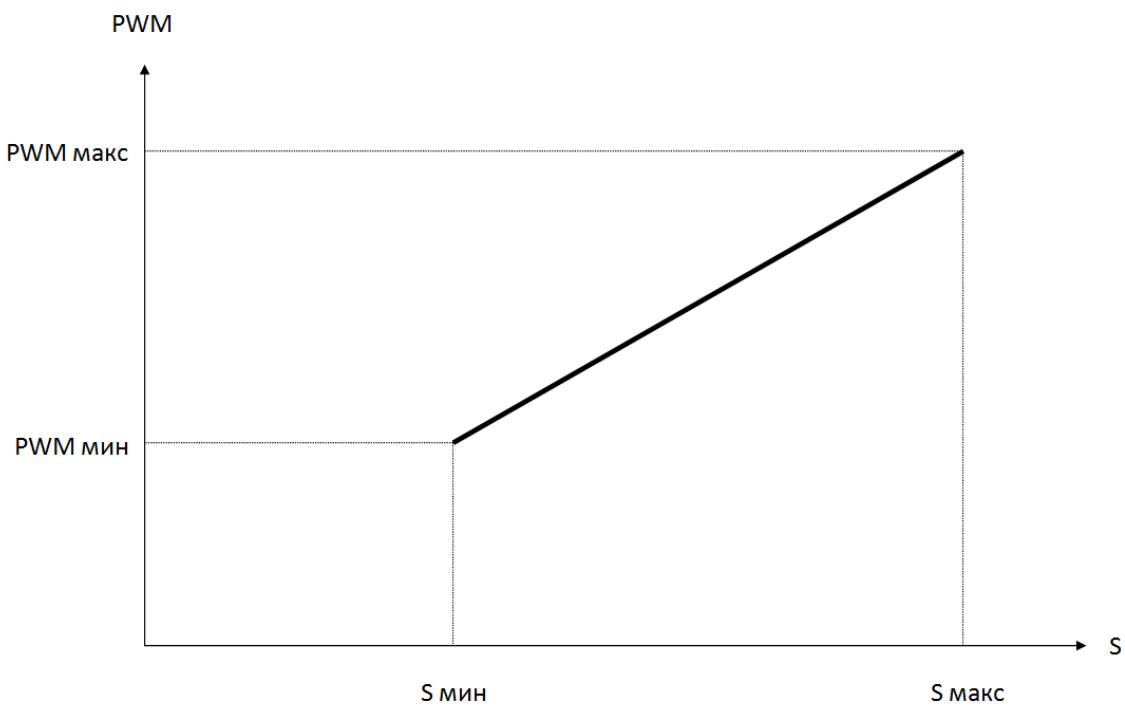


В данном окне есть несколько блоков:

3.3.1 Выход ШИМ.

В данном блоке мы устанавливаем параметры ШИМ выхода который используется для управления шпинделем станка. Галочка в блока ШИМ - означает о что мы будем использовать выход ШИМ (если её нет то выход будет использоваться как дискретный)

- Частота ШИМ - частота ШИМ сигнала в Герцах
- S макс/мин и ШИМ макс/мин - это данные для расчёта ШИМ (т.е. если параметр S будет больше либо равен S макс то выход ШИМ будет установлен в ШИМ макс и наоборот)



Для того, чтобы ШИМ выход стал активным (выдавать заданный сигнал) его нужно активировать при помощи команды макроса (enableSOut(enable)) и наоборот (сделать неактивным)(см 4.8) .

Инверсия сигнала ШИМ производится установкой соотв. галочки.

3.3.2 Корректировка S.

В программе доступна функция корректировки S. То есть можно задать опорные точки в которых мы точно укажем расхождение заданных и полученных данных.

Для этого есть блок "Корректировка выхода S". В нём имеется определенное количество опорных точек, которые мы задаем.

Включение и выключение корректировки S осуществляется с помощью установки галочки блока.

Как производить заполнение таблицы корректировок будет описано в разделе 4.

3.3.3 Вход.

В данном блоке мы устанавливаем общие входы для всех осей.

inEMG - вход незамедлительной остановки

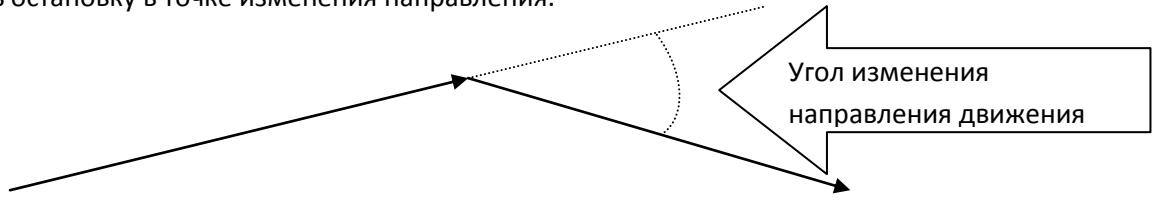
inSD - вход плавной остановки

inProbe - вход пробинга (используется для поиска детали, заготовки)

3.3.4 Сглаживание.

В данном блоке мы устанавливаем параметры сглаживания траектории.

- Угол - это предельный угол сглаживания. Т.е. если в процессе движения траектория изменит направление на угол меньше (либо равен) чем задан то контроллер не будет производить остановку в точке изменения направления.



3.3.5 Штурвал.

В данном блоке мы устанавливаем параметры работы с штурвалом (ручной энкодер).

- имп/обр - количество импульсов наоборот

3.3.6 Подача.

В данном блоке устанавливаем предельные подачи.

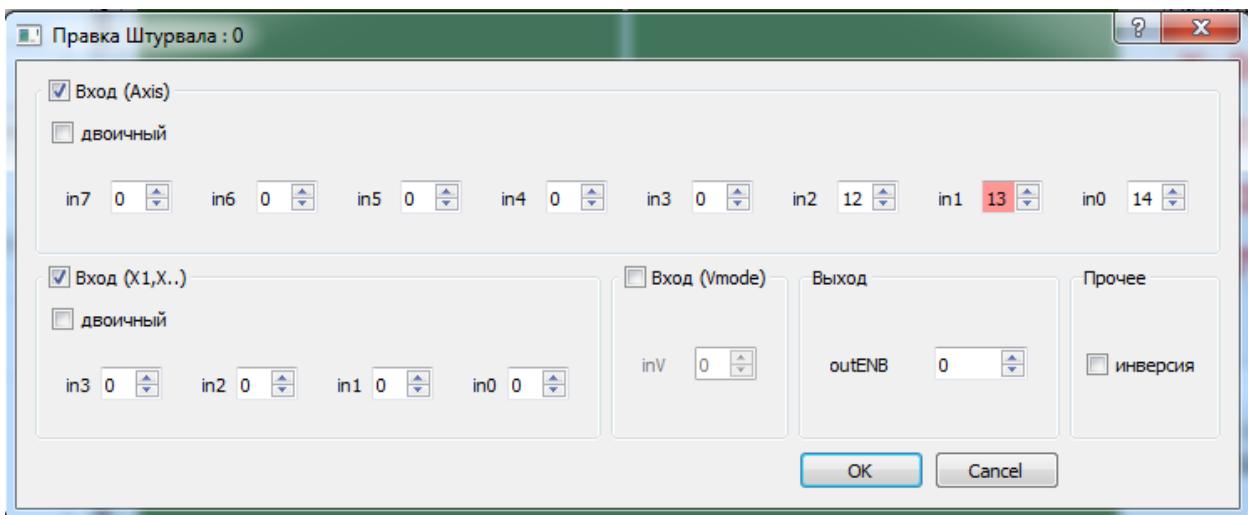
- Люфт - подача отработки люфта (если =0 то используется последняя подача)
- Ручной - подача медленных ручных перемещений
- G1,G2,G3 - максимальная подача рабочих ходов (если =0 то ограничения не действует)

3.3.7 Люфт.

Установка люфтов (свободного хода) осей.

3.4 Настройка ручного Энкодера (Whell)

Для настройки ручного энкодера необходимо выбрать пункт "Правка->Whell". При наличие данного модуля в контроллере должно появится окно:



Если мы будем использовать переключение осей управления с помощью входов, то должны поставить галочку на этом блоке. Далее блоке "Вход (Axis)" мы устанавливаем номера входов, которые мы будем использовать.

Также мы должны поступить и с блоком "Вход (X)" отвечающий за размер единичного перемещения (множитель X1, X10, X100).

В случае использования двоичных сигналов необходимо поставить галочку внутри соотв. блока. "инверсия" меняет направление счёта входных импульсов.

Если мы будем использовать вход установки режима работы от ручного энкодера, то должны поставить галочку на этом блоке. И установить номер входа.

Выход "outENB" устанавливается в логическую единицу если Штурвал активен.

Примечание: в контроллерах WLMotion вход номе "0" и "1" это виртуальные статичные входа. Выход под номером "0" также виртуальный, его нужно указывать если выход не используется.

4 Типовые операции

В данном разделе будут описаны типовые операции в программе WLMill.

4.1 Включение станка

Включения станка в программе WLMill осуществляется с помощью соотв. кнопки на экране в окне "управление" (п2.6).

Если станок выключен/включен то можно/нельзя производить настройку программы (п 3)

4.2 Ручное перемещение.

Необходимо:

- включить станок
- перейти в п.2.6.2
- произвести перемещение от клавиатуры либо от штурвала

Разрешающие входы(inALM,inPEL,inMEL,inSDStop,inEMGStop) должны быть в соотв. состоянии.

4.3 Установка начальных положений двигателя.

Когда мы вводим станок в эксплуатацию то нам необходимо для каждой оси выполнить следующие действия:

- настроить ось (установить размер шага, типы импульсов, правильное направление вращения двигателя)
- включить станок (нажатие кнопки на экране)
- произвести перемещение оси в предполагаемую позицию. визуально
- установить значение положения двигателя в эту позиции. (2.32.3, вкладка двигатели, изменение положения).
- произвести обучение оси (2.7.6), В результате чего мы узнаем положение датчика относительно двигателя (смещение). В дальнейшем при поиске положения оси мы будем находить положение двигателя по датчику (наоборот).
- скорректировать пределы перемещения оси

Для проверки "обучения оси" можно задать неверное положение оси и произвести поиск её положения.

обучение оси - это нахождение положения датчика по двигателю

поиск положения оси - это нахождение положения двигателя по датчику

также мы можем задать положение нашего датчика используя меню настройки оси 3.1.8

ВНИМАНИЕ!!! при поиске положения по концевым датчикам не происходит сброс положения в 0. А восстанавливается позиция оси по известному смещению.

4.4 Задание пределов перемещения оси.

Для задания (выяснения) пределов перемещения нам необходимо:

- установить галочку на блоке "пределы" в настройке оси. Установить заведомо большие значения пределов.
- произвести поиск положения оси. Индикатор положения оси (например "X:100.0") должен стать чёрным.
- переходим в раздел "сервис". В блоке "ось" выбираем ось для которой мы хотим установить пределы.
- нажимаем кнопку "аннулировать поиск". Индикатор положения оси (например "X:100.0") должен стать красным. (это делается если пределы будут больше чем ранее заданные).
- в ручном режиме аккуратно перемещаем ось к отрицательному/положительному пределу и нажимаем кнопку "Взять отр. предел"/"Взять полож. предел". Внимание!!! вы не должны достигнуть срабатывания датчиков inPEL и inMEL (концевые датчики двигателя).
- производим поиск положения оси и проверяем в ручном режиме работу пределов. Ось не должна выйти за них.

Для удобства оценки пределов на экране отображается куб пределов осей X,Y,Z. Если мы нажмем кнопку отображения "по центру", то у нас будет квадрат пределов по X,Y.

4.5 Установка текущего положения фрезы.

Необходимо:

- Задать нужную СК с помощью ввода соотв. команды (п 4.3) (G54,G55...).

- В окне "Положения" (п.2.3) с помощью двойного щелчка установить нужные значения

4.6 Работа по программе

Необходимо:

- включить станок
- в окне 2.6.1.2 нажать кнопку "загрузить траекторию"
- если не был выбран "автостарт", нажать в кнопку "пуск".

4.7 Поворот СК.

В программе WLMill реализована функция поворота СК по реферным точкам. Данные точки



отображаются в виде жёлтых конусов ( пункт "поворот").



- данная кнопка имеет выпадающее меню (нажать и удерживать):

- установить базовую позицию (в СК программы)
- установить вспомогательную позицию (в СК программы)
- корректировать поворот

Общий принцип поворота СК:

1. Задаются координаты двух точек (базовая(вокруг неё происходит поворот) и вспомогательной(малый конус)).
2. Находится координата базовой точки (касание, подвод, или просто задаётся).
3. Подводится инструмент к вспомогательной точке, и нажимается кнопка "корректировать поворот".

после чего рассчитывается угол поворота которому соот. текущее положение детали.

Также можно повернуть СК относительно базовой точки на заданную величину просто нажать на

 и ввести угол поворота.

4.8 Работа со шпинделем.

Для работы со шпинделем имеется 3 макроса.

- function M3() - вращение по часовой стрелке
- function M4() - вращение по против часовой стрелки
- function M5() - остановка шпинделя

В этих макросах нам необходимо определить ,что программа должна сделать.

Если мы используем **дискретное** управление шпинделем (вкл/выкл) в **одну строку то** (выход 3 включает вращение влево):

function M3() { MACHINE.setOutput(3,1); return 1;	function M4() { return 1;	function M5() { MACHINE.setOutput(3,0); return 1;
--	---------------------------------	--

}		}
---	--	---

Если мы используем **дискретное** управление шпинделем (вкл/выкл) в **две стороны то**(выход 3 включает вращение влево 4 вправо):

<pre>function M3() { MACHINE.setOutput(3,1); MACHINE.setOutput(4,0); return 1; }</pre>	<pre>function M4() { MACHINE.setOutput(4,1); MACHINE.setOutput(3,0); return 1; }</pre>	<pre>function M5() { MACHINE.setOutput(3,0); MACHINE.setOutput(4,0); return 1; }</pre>
---	---	---

Для использования **плавной регулировки** в одну сторону нужно активировать выход **S**.

<pre>function M3() { MACHINE.enableSOut(1); return 1; }</pre>	<pre>function M4() { return 1; }</pre>	<pre>function M5() { MACHINE.enableSOut(0); return 1; }</pre>
---	--	---

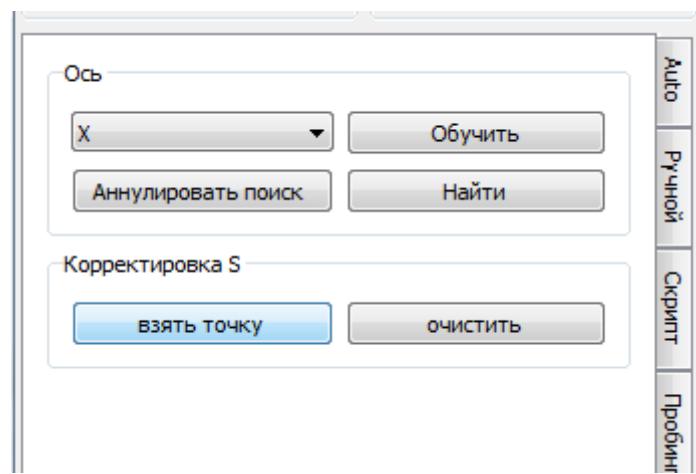
Для организации задержки нужно использовать функцию **DELAY(ms);**(например **DELAY(1000)** задержка в 1секунду)

4.9 Корректировка выхода S.

Корректировка выхода S осуществляется путем задания опорных точек. То есть нам необходимо задать определенное количество точек (от 2 до 10) в которых мы укажем размер корректировки в процентах.

Общий алгоритм задания корректировки S:

1. Настраиваем значения S (**Sмин,Sмакс**) и соотв. значения выхода S (PWM и прочее).
2. Включаем корректировку S.
3. Устанавливаем значение S в минимальное положение.
4. Переходим в раздел "Сервис" нажимаем кнопку "очистить". Для удаление всех ранее введенных корректировок.
5. После чего корректором добиваемся того чтобы заданное значение S заданное совпало с реальным. После чего добавляем точку корректировки. (нажимаем в закладке "Сервис" кнопку "Взять точку " в блоке "Корректировка S").



6. Повторяем п5. для максимального значения S.
7. Повторяем п5 ещё для нескольких значений S.(нужно выбирать такие значения S, где корректировка будет максимальной).
8. Проверяем правильность работы. Значение оборотов должно соотв. S.

4.10 Настройка штурвала

При работе с ручным энкодером, возможны различные подключения.

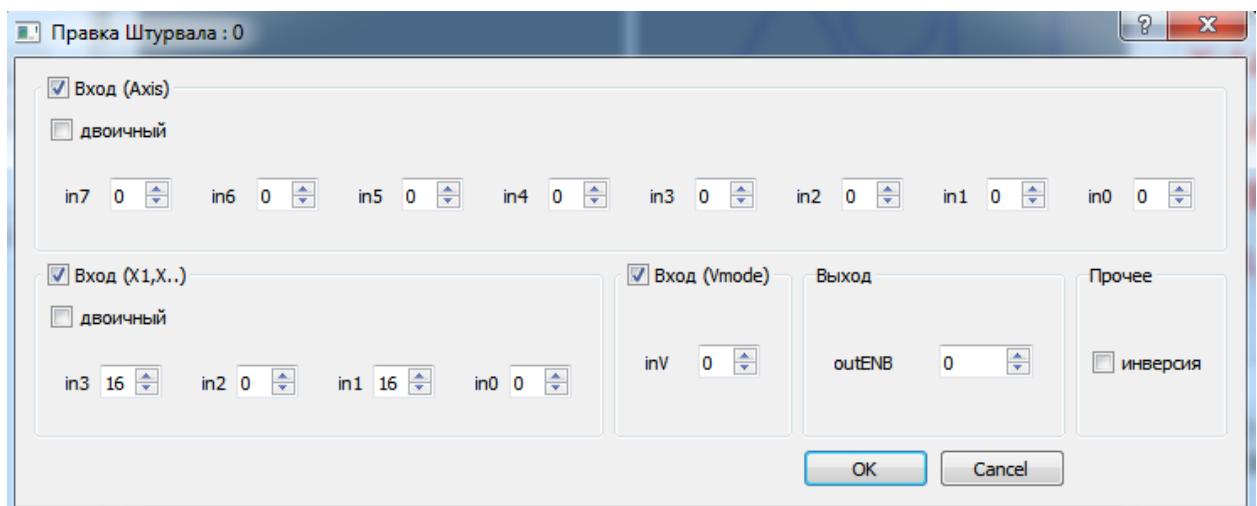
Имеется 3 блока входных данных.:

- (Axis) Номер оси
- (X1,X..)Шаг перемещения
- (Vmode) Вид перемещения (по скорости / по позиции)

Каждый блок можно использовать или не использовать . Если какой либо блок используется (внешнее переключение) , то его нельзя переключить с компьютера и наоборот.

Например если мы не желаем задавать тип перемещения то мы убираем соотв. галочку с блока в окне правки штурвала.

Также мы можем в окне штурвала произвести инверсию колеса, и установить тип входных данных (двоичный или одинарный). Для удобства отображается текущее состояние входов с помощью фона номера входа.



4.11 Редактирование программы.

5 Скрипты

В программе WLMill есть возможность выполнения скриптов написанных на языке JavaScript. Данный инструмент весьма полезен при работе с программой, с помощью скриптов можно выполнять различные операции, которые не были предусмотрены в стандартном функционале WLMill.

В язык JavaScript встроен определенный набор функционала для работы с программой. Каждый элемент функционала имеет имя написанное ЗАГЛАВНЫМИ буквами.

Например:

```
MACHINE.setOutput(2,1) //установит выход 2 в логическую единицу
```

Язык JavaScript достаточно прост, его описание можно найти в интернете, также существует большое количество видео с уроками по этому языку.

Далее будет описание элементов функционала скриптов WLMill.

5.1 MACHINE - основной элемент функционала

Функция	Описание
bool getInput(index)	Возвращает текущее состояние входа
bool getOutput(index)	Возвращает текущее состояние выхода
setOutput(index,state)	Устанавливает выход в заданное состояние
setOutputPulse(index,state,time)	Формирует импульс заданного состояния, и заданной продолжительности (в миллисекундах).
setOutputTog(index)	Изменяет состояние выхода на противоположный
enableSOut(enable)	Активирует выход S
goDriveFind("*")	Запускает процедуру поиска положения двигателя по датчику
bool isDriveActiv("")	Если двигатель активен (выполняет движение или выполняет автоматическую операцию) то возвращает true.
void setCurPositionSC(" ", pos)	Устанавливает текущее значение заданной координаты в активной системе координат
bool runGCode(txt)	Выполняет G код
double setCurPositionSC("")	Возвращает положение оси в текущей СК
bool goDriveProbe("",dir,F,typeStop)	Запуск цикла пробинга для оси. typeStop 0 - резкая остановка 1 - плавная остановка с возвратом 2 - плавная остановка
double getProbePosition("",front)	Считывание регистра защёлки пробинга оси с указанием фронта.
bool getInProbe()	Возвращает текущее положение входа inProbe

5.2 TIMER - таймеры

Элемент таймер содержит в себе 32 независимых таймера. Которые могут использоваться при работе со скриптами.

Функция	Описание
start(index)	Запускает таймер
restart(index)	Перезапускает таймер
stop(index)	Останавливает таймер
long getCount(index)	Возвращает счётчик таймера (в мсек)

5.3 DIALOG - диалоги

Помимо всего прочего имеется интерфейс для общения с пользователем из скрипта. Работа с данным элементом строится по следующему принципу:

- создаётся окно (вызов соотв. диалога: enterNum(txt), enterString(txt) итд)
- ожидается закрытие окна (isShow() должен возвратить 0)
- обрабатывается данные из окна (при необходимости)

Функция	Описание
message(txt)	Вывод окна с сообщением
enterNum(txt)	Вывод окна с подсказкой для ввод числа
enterString(txt)	Вывод окна с подсказкой для ввод строки
enterSaveFile(txt1,lastFile)	Вывод окна выбора сохранения файла
bool isShow()	Отображается ли какое либо окно
bool isOk()	Была ли нажата кнопка подтверждения ввода
bool isCancel()	Была ли нажата кнопка отмены ввода
real getNum()	Возвращает введённое число
txt getString()	Возвращает введённую строку
setString(txt)	Устанавливает строку в окно ввода строки
setNum(real)	Устанавливает число в окно ввода числа.

5.4 FILE - работа с файлами

Элемент макросов FILE даёт возможность работы с файлами.

Функция	Описание
bool createFile(nameFile)	Запись пустого файла. (стирание файла)
bool write(nameFile,string)	Запись строки в файл.
bool saveVal(nameFile,nameVal,data)	Сохранение переменной в файл
data loadVal(nameFile,nameVal)	Чтение переменной из файла