



Albatros
3.2.3

ЧПУ

Tecnologie e Prodotti per l'Automazione

Настоящая документация является собственностью
компании T.P.A S.r.l.
Запрещается копирование без предварительного
согласования.
Компания сохраняет за собой право в любой момент
вносить изменения в текст

Оглавление

1	Введение	1
1.1	Правила пользования справочником	1
1.2	Рабочие окна	1
2	Состав системы	2
2.1	Права доступа к системе	2
2.2	Многоязыковая поддержка	2
2.3	Типовая архитектура системы	2
2.4	Организация и логическая конфигурация	4
2.5	Устройства	5
3	Обзорный Экран	7
3.1	Использование Обзорного Экрана	7
3.2	Порядок работы на Обзорном Экране	7
3.3	Порядок воздействия на Устройства	7
3.4	Перемещение Осей в Ручном режиме	7
4	Технологические параметры и Параметры инструментов	9
4.1	Окно "Технологические параметры"	9
4.2	Окно "Параметры инструментов"	10
5	Диагностика	12
5.1	Окно "Диагностика"	12
5.2	Состав окна "Диагностика"	12
5.3	Представление Устройств	12
5.4	Взаимодействие с Устройствами	13
5.5	Список клавиш для навигации по древовидной структуре	13
5.6	Корректоры линейности	14
5.7	Планшет калибровки осей	14
6	Ошибки и Сигналы	17
6.1	Введение	17
6.2	Ошибки системы	18
6.2.1	Ошибки, генерированные управлением осей	18
	1 ИмяОси: неправильное подключение кодера	18
	2 ИмяОси: движение не завершено	18
	3 ИмяОси: сервоошибка	18
	4 ИмяОси: выход за положительный предел	19

	5 ИмяОси: выход за отрицательный предел	19
	10 ИмяОси: Выполнение в реальном времени быстрее построения профиля	19
6.2.2	Ошибки, генерированные управлением удаленных модулей Входов/Выходов	19
	2049 Приемник номер: неправильная конфигурация	19
	2050 Приемник номер: отключен	19
	2051 Приемник номер: снова подключен	19
	2052 Приемник номер: ошибка повторного считывания неподключенного выхода номер НомерВыхода	19
	2054 Приемник номер: неправильный тип	20
	2055 Приемник номер: инициализирован	20
	2056 Приемник номер: ошибка питания +24 В пост. тока	20
	2057 Ошибка питания GreenBUS	20
	2058 Приемник номер: ошибка повторного считывания ТипУстройства ИмяУстройства	20
	2059 Не пройден тест двухпортовой памяти передатчика	20
	2060 Ошибка при инициализации передатчика	21
	2061 Ошибка при передаче встроенного ПО на передатчик	21
	2062 Ошибка при передаче конфигурации на передатчик	21
	2063 Ошибка при передаче конфигурации на приемник	21
	2064 Приемник номер: несовместимая версия встроенного ПО	21
	2065 Приемник номер: ошибка асинхронной связи	21
	2066 Приемник номер: общая ошибка	22
	2067 Приемник номер: ошибка при передаче конфигурации	22
	2068 Приемник номер: внутренняя ошибка № номерошибки	22
	2069 Приемник номер: ошибка питания +24 В пост. тока стенд номер	22
6.2.3	Ошибки, генерированные управлением МЕCHATROLINK-II	22
	2308 Плата НомерПлаты: инициализация не удалась вследствие неправильного задания параметра конфигурации	22
	2341 Плата НомерПлаты: число сервоприводов превышает максимально допустимое значение	22
	2342 Плата НомерПлаты: аппаратный адрес сервопривода Серво превышает максимальное разрешенное значение	23
	2349 Плата НомерПлаты: сервопривод Серво не подключен	23
6.2.4	Ошибки, генерированные управлением CanBUS	23
	2761 Узел номер: отключен	23
	2762 Узел номер: снова подключен	23
	2763 Ошибка отсутствия передачи	24
	2764 Узел номер: ошибка отсутствия приема	24
	2765 Узел номер: инициализирован	24
	2766 Условие отказа на интерфейсе CAN	24
	2767 Ошибка утраты состояния CANopen	24
	2768 Узел номер: ошибка отсутствия приема PDO	24
	2769 Узел номер: ошибка приема неконфигурированного узла	24
	2770 Узел номер: неправильная конфигурация	24
	2771 Узел номер: ошибка связи SDO	25
	2772 Таймаут цикла CAN опроса узлов	25
	3073 Узел номер: аварийная Ошибка № Номерошибки	25
	3074 Узел номер: общая ошибка CAN № Номерошибки	25
	3088 Плата CAN номер: узел НомерУзел: ошибка связи SDO № Номерошибки - описание	25
6.2.5	Ошибки, генерированные управлением шиной EtherCAT	25
	3329 Ошибка инициализации гнезда связи	25
	3330 Ошибка при сканировании сети EtherCAT	26
	3331 Ошибка конфигурации почтового ящика передачи	26
	3332 Ошибка конфигурации почтового ящика приема	26
	3333 Плата EtherCAT номер: ошибка типа расширений узла НомерУзел	26
	3334 Ошибка на этапе конфигурации PDO	26
	3335 Узел НомерУзел в аварийной ситуации (НомерОшибки)	26
	3336 Плата EtherCAT номер: число расширений узла НомерУзел неправильно	28

	3337 Плата EtherCAT: узел НомерУзел отключен	28
	3338 Плата EtherCAT: узел НомерУзел снова подключен	28
	3340 Плата EtherCAT: узел НомерУзел не ответил на запрос (Код)	28
	3341 Плата EtherCAT: узел НомерУзел не существует	28
	3342 Кабель отключен	28
	3343 Плата EtherCAT номер: узел НомерУзел не переходит в состояние SAFE-OPERATIONAL (Код)	29
	3344 Плата EtherCAT номер: узел НомерУзел не переходит в состояние OPERATIONAL (Код)	29
	3345 Плата EtherCAT: нестабильная связь	29
	4400 Слишком много активных осей в FASTREAD (Функция:ИмяФункция линия:НомерЛиния)	29
6.2.6	Ошибки, генерированные инициализацией	29
	769: Неправильная конфигурация программного обеспечения	29
	770 Номер IRQ сконфигурирован неправильно	29
	772 Ошибка при считывании зоны буферной памяти при инициализации	30
	773 Достигнуто максимальное число конфигурируемых осей	30
	774 Не начала выполнение задача реального времени осей	30
	775 Недостаточно времени для выполнения GPL	30
	776 Слишком большое время выполнения задачи реального времени	30
	777 Истек период работы watchdog	31
	778 Заблокирован код главной функции встроенного ПО	31
	1025 Плата НомерПлаты: не отвечает на команды	31
	1026 Плата НомерПлаты: ошибка при отправке встроенного ПО на плату осей	31
	1028 Плата НомерПлаты: встроенное ПО не присутствует	31
	1029 Плата НомерПлаты: заблокирована главная функция	31
	1031 Плата НомерПлаты: ошибка при инициализации	31
	1032 Плата НомерПлаты: не пройден тест двухпортовой памяти	32
	1033 Плата НомерПлаты: код начальной загрузки встроенного ПО не выполняется	32
	1035 Плата НомерПлаты: не присутствует	32
	1037 Плата НомерПлаты: не удалось открытие двухпортовой памяти	32
	1039 Плата НомерПлаты: истек период работы сторожевой схемы	32
	1040 Плата НомерПлаты: ошибка питания +24 В пост. тока	32
	1047 Плата НомерПлаты: неразрешенная конфигурация программного обеспечения	33
	1052 Плата НомерПлаты: выполняется код начальной загрузки	33
	1053 Плата НомерПлаты: Истек период работы watchdog осей	33
	1055 Истек период работы watchdog на Плате НомерПлаты	33
	1056 Плата НомерПлаты: ошибка питания интерфейса CAN	33
	1057 Плата НомерПлаты: внутренняя ошибка номер НомерОшибки	33
6.2.7	Ошибки, генерированные управлением памяти	33
	1281 Ошибка распределения памяти в зоне хипа	33
	1286 Ошибка при реализации хипа	34
	1287 Слишком много отнятий памяти от хипа	34
	1289 Ошибка при создании глобальных переменных	34
	1290 Ошибка размера энергонезависимых переменных	34
	1291 Ошибка размера переменных только для считывания	34
6.2.8	Ошибки, генерированные отказами	34
	1559 Следы точки останова	34
	1569 Недействительный рабочий код микропроцессора	34
	1586 Значение INTEGER делится на ноль	35
	1600 Переполнение в результате операции с плавающей запятой	35
	1601 Незагруженность в результате операции с плавающей запятой	35
	1602 Недействительный аргумент для операции с плавающей запятой	35
	1603 Значение с плавающей запятой, деленное на ноль	35
	1604 Неточный результат операции с плавающей запятой	35
	1605 Было использовано неправильное значение с плавающей запятой	35

1728	Была сделана попытка доступа к недействительному адресу	36
1735	Общее исключение	36
1736	Невыровненные данные	36
1801	Температурный сигнал тревоги	36
1802	Сигнал тревоги вентилятора	36
1803	Частота ЦП нестабильна	36
6.2.9	Ошибки, генерированные инструкциями GPL	37
4097	Устройство ТипУстройства ИмяУстройства не сконфигурировано	37
4098	Глобальная переменная ИмяПеременной не существует	37
4099	Функция ИмяФункции не найдена	37
4101	Непоследовательное управление осью ИмяОси	37
4105	Инструкция не может быть выполнена на оси ИмяОси	37
4106	Не подключен удаленный модуль, относящийся к шаговой оси ИмяОси	38
4107	Инструкция SYSOK с неправильными аргументами	38
4108	Ось ИмяОси: окончательная квота за пределы программного обеспечения	38
4110	Неправильная скорость	38
4111	Отрицательное ускорение оси ИмяОси	38
4112	Отрицательное замедление оси ИмяОси	38
4114	Ось ИмяОси: не выполнено обнуление на быстром входе	38
4115	Ось ИмяОси: не найдена отметка нуля	39
4353	Рабочий код инструкции неизвестен (Функция:ИмяФункции строка:НомерСтроки)	39
4354	Неправильная математическая операция (Функция:ИмяФункции строка:НомерСтроки)	39
4355	Неправильный адрес матрицы или вектора (Функция:ИмяФункции строка:НомерСтроки)	39
4356	Инструкция RET не вызвана CALL (Функция:ИмяФункции строка:НомерСтроки)	39
4357	Несуществующая локальная переменная (Функция:ИмяФункции строка:НомерСтроки)	40
4358	Несуществующий ярлык перехода (Функция:ИмяФункции строка:НомерСтроки)	40
4359	Неправильный аргумент макропрограммы (Функция:ИмяФункции строка:НомерСтроки)	40
4360	Ошибка распределения памяти при выполнении (Функция:ИмяФункции строка:НомерСтроки)	40
4361	Слишком много активных задач (Функция:ИмяФункции строка:НомерСтроки)	40
4362	Неправильный формат матрицы (Функция:ИмяФункции строка:НомерСтроки)	41
4363	Слишком много активных инструкций ONINPUT (Функция:ИмяФункции строка:НомерСтроки)	41
4364	Ось уже занята в локальной системе координат (Функция:ИмяФункции строка:НомерСтроки)	41
4365	Инструкция ONINPUT активирована на том же входе (Функция:ИмяФункции строка:НомерСтроки)	41
4366	Слишком много активных инструкций ONFLAG (Функция:ИмяФункции строка:НомерСтроки)	41
4367	Инструкция ONFLAG активирована на том же флажке (Функция:ИмяФункции строка:НомерСтроки)	42
4368	Попытка записи переменной типа только для считывания (Функция:ИмяФункции строка:НомерСтроки)	42
4369	Слишком много активных главных осей (Функция:ИмяФункции строка:НомерСтроки)	42
4370	Слишком много активных ведомых осей (Функция:ИмяФункции строка:НомерСтроки)	42
4372	Неправильное использование инструкции (Функция:ИмяФункции строка:НомерСтроки)	42
4373	Невозможно выполнить считывание скорости подачи (Функция:ИмяФункции строка:НомерСтроки)	43
4374	Выполняется слишком много инструкций типа IPC (Функция:ИмяФункции строка:НомерСтроки)	43
4375	FASTREAD выполнена на осях различных плат (Функция:ИмяФункции строка:НомерСтроки)	43

4378	Неподключенная инструкция (Функция:ИмяФункции строка:НомерСтроки)	43
4379	Инструкция не может использоваться в функциях, запущенных прерыванием (Функция:ИмяФункции строка: НомерСтроки)	43
4380	Слишком много запросов записи в зоне буферной памяти (Функция:ИмяФункции строка: НомерСтроки)	44
4381	Невозможно использовать еще неоткрытую последовательную линию (Функция:ИмяФункции строка: НомерСтроки)	44
4382	Невозможно открыть уже открытую последовательную линию (Функция:ИмяФункции строка: НомерСтроки)	44
4383	Была сделана попытка открыть слишком много вспомогательных процессов (Функция:ИмяФункции строка: НомерСтроки)	44
4384	Вспомогательный процесс не выполняется (Функция:ИмяФункции строка: НомерСтроки)	44
4385	Была сделана попытка открыть вспомогательный процесс из другой задачи (Функция:ИмяФункции строка: НомерСтроки)	44
4391	Ошибка при активации SYSOK (Функция:ИмяФункции строка: НомерСтроки)	45
4394	Слишком много ошибок цикла (Функция:ИмяФункции строка:НомерСтроки)	45
4395	Слишком много сообщений (Функция:ИмяФункции строка:НомерСтроки)	45
4397	Переполнение стека на функции ИмяФункции (Функция:ИмяФункции строка:НомерСтроки)	45
4398	Незагруженность стека на функции ИмяФункции (Функция:ИмяФункции строка: НомерСтроки)	45
4399	Параметр за пределами диапазона (Функция:ИмяФункции строка: НомерСтроки)	45
4865	Отсутствует определение станка для интерполяции (G216 или G217)	45
4866	Отсутствует описание индексов конфигурации станка (M6)	46
6.2.10	Ошибки, генерированные драйвером связи CNCTPA	46
16385	Модуль отключен	46
16386	Модуль подключен	46
16387	Модуль снова подключен	46
16388	Модуль инициализирован	46
16389	Модуль прервал связь	46
16641	Встроенное ПО блока управления не реагирует на команды	47
16642	TraSock не реагирует на команды	47
16643	Операционная система не позволяет использовать RTX	47
16645	Ошибка при отправке кода встроенного ПО	47
16646	Оказалось невозможным снова направить в выполнение код встроенного ПО	47
16897	Не установлен RTX	47
16898	Пользователь не имеет прав администратора	48
16899	Неправильный размер ОЗУ модуля	48
16900	Неправильный IP-адрес модуля	48
16901	Модуль уже подключен к другой установке	48
16902	Модуль не сконфигурирован	48
16903	Настройки брандмауэра не допускают связь	48
16904	Сетевая плата отсутствует или отключена	49
16905	Отсутствует код встроенного ПО блока управления	49
16906	Версия RTX несовместима с кодом встроенного ПО блока управления	49
16907	Версия операционной системы несовместима с кодом встроенного ПО блока управления	49
17153	ТипПлаты: нет кода встроенного ПО передатчика GreenBUS	49
17154	ТипПлаты: нет части кода встроенного ПО передатчика GreenBUS	49
17155	ТипПлаты: ошибка отправки кода программы самозагрузки передатчика GreenBUS	50
17156	ТипПлаты: ошибка отправки кода главной функции передатчика GreenBUS	50
17157	ТипПлаты: нет кода программы самозагрузки	50
17158	ТипПлаты: нет кода главной функции	50
17159	ТипПлаты: ошибка отправки кода программы самозагрузки	50

17160 ТипПлаты: ошибка отправки кода главной функции	50
17409 Оказалось невозможным направить вспомогательный исполняемый файл	51
17410 Оказалось невозможным выполнить вспомогательный исполняемый файл	51
17667 ИмяDLL: оказалось невозможным направить в выполнение код встроенного ПО	51
17668 ИмяDLL: оказалось невозможным получить указатель к совместно используемой ОЗУ	51
17921 Оказалось невозможным направить NODETPA	51
17922 NODETPA повторно не запустился	51
17923 NODETPA не выполняется	52
18177 NODETPA попытался получить доступ к недействительному адресу	52
6.3 Общие сигналы	52
6.3.1 Albatros начинает выполнение	52
6.3.2 Albatros завершает выполнение	52
6.3.3 Компьютер входит в режим ожидания	52
6.3.4 Компьютер выходит из режима ожидания	52
6.3.5 Выключить компьютер	52
6.3.6 Текущий уровень доступа	52
6.3.7 Обновление программного обеспечения модулей	52
6.3.8 Отправка конфигурации модулям	53
7 Конфигурация Системы	54
7.1 Введение	54
7.2 Конфигурация устройств	54
7.2.1 Введение	54
7.2.2 Общее устройство	54
7.2.3 Цифровой выход	55
7.2.4 Аналоговый вход	55
7.2.5 Ось	55
Базовые Данные	55
Параметры Движения	56
Параметры Интерполяции	56
Другие Параметры	56
Параметры Эталона	56
Уровни доступа	57
Сцепление осей	57
Корректоры линейности	57
7.3 Логическая конфигурация	58
7.3.1 Конфигурация установки	58
7.3.2 Конфигурация групп	58
7.4 Физическая конфигурация	60
7.4.1 Конфигурация системы	60
7.4.2 Конфигурация аппаратного обеспечения	60
Предопределенные конфигурации	62
Конфигурировать узел шины TPA	62
Конфигурировать узел шины CAN	63
Плата управления шиной	63
Узел CAN	64
Вставить новый узел	64
Конфигурировать узел	64
Характеристики управления EtherCAT в Albatros	64
Вступление	64
Конфигурация аппаратного обеспечения EtherCAT	65
Описание PDO	66

Модификация PDO привода	66	
Дополнительные PDO	67	
Автоматическое получение узлов EtherCAT	68	
7.4.3	Конфигурация Виртуальное-физическое	68
7.4.4	Карты электромонтажа	70
7.5	Список клавиш для навигации по древовидной структуре	70
8	Средства разработки	71
8.1	Редактор GPL	71
8.1.1	Функции редактора GPL	71
	Использование регулярных выражений	72
8.1.2	Вставить сообщение	74
8.1.3	Криптография	74
8.1.4	Список горячих клавиш	75
8.2	Библиотеки	77
8.3	Отладка	77
8.3.1	Отладчик	77
8.3.2	Выполняемые задачи	78
8.3.3	Все задачи	78
8.3.4	Вызовы по функции	79
8.3.5	Точки прерывания	79
8.3.6	Содержание переменной	79
8.3.7	Список горячих клавиш	80
8.4	Инициализация блока управления	80
8.4.1	Сетевые подключения	80
8.4.2	Аппаратная Диагностика	81
	Топология сети EtherCAT	81
	Просмотр и редактирование объектов в узлах	81
8.5	Тест	82
8.5.1	Запомнить глобальную переменную	82
8.5.2	Выполнить функцию	83
8.5.3	Импорт сообщений	83
8.5.4	Примечание пользователя в файле отчета о тревоге	85
8.6	Инструменты	85
8.6.1	Индивидуализировать...	85
8.7	Браузер	87
8.7.1	Браузер	87
8.7.2	Искать идентификатор...	87
8.7.3	Список горячих клавиш	88
9	Вспомогательные программы	89
9.1	XConfMerge: программа для слияния конфигурационных файлов	89
9.2	XParMerge: программа для слияния двух параметрических файлов	90
10	Язык GPL	91
10.1	Базовые концепции	91
10.1.1	Введение в язык GPL	91
10.1.2	Условности и терминология	91
10.1.3	Переменные	93

	Типы данных	93
	Преобразование данных	95
	Объявление и видимость переменных	96
	Модификаторы	96
	Задание RANGE	97
	Права Считывания / Записи	97
10.1.4	Константы	97
	Заданные константы с предустановленным значением	98
	Заданные константы с предустановленным значением при запуске Albatros	99
10.1.5	Ключевые слова	100
10.1.6	Функции	101
10.1.7	Параметры типа устройства	103
10.1.8	Многозадачность	103
10.1.9	Связь	105
10.1.10	Переменные, используемые в программировании	105
10.1.11	Оси	105
10.1.12	Корректоры линейности	108
10.1.13	Управление сообщениями на языке	108
10.1.14	Управление ошибками системы	108
10.2	Специальные функции	109
10.2.1	Персонализация перемещения осей	109
10.2.2	Стандартные функции перемещения и калибровки	111
10.2.3	Функция OnUIEnd#	115
10.2.4	Функция OnUIPlugged#	115
10.2.5	Функция OnUIUnPlugged#	115
10.3	Инструкции	115
10.3.1	Условности	115
10.3.2	Типология инструкций языка GPL	115
10.3.3	Ввод/Вывод	122
	GETFEED	122
	INPANALOG	122
	INPFLAGPORT	122
	INPPORT	122
	MULTIINPPORT	123
	MULTIOUTPUTPORT	123
	MULTIRESETFLAG	123
	MULTIRESETOUT	124
	MULTISETFLAG	124
	MULTISETOUT	124
	MULTIWAITFLAG	124
	MULTIWAITINPUT	125
	OUTANALOG	125
	OUTFLAGPORT	125
	OUTPORT	126
	RESETFLAG	126
	RESETOUT	126
	SETFLAG	126
	SETOUT	126
	WAITFLAG	126
	WAITINPUT	127
	WAITPERSISTINPUT	127
10.3.4	Оси	128
	CHAIN	128
	CIRCABS	129
	CIRCINC	129
	CIRCLE	130

COORDIN	131
DISABLECORRECTION	132
EMERGENCYSTOP	132
ENABLECORRECTION	133
ENDMOV	133
FASTREAD	134
FREE	134
HELICABS	135
HELICINC	135
JERKCONTROL	136
JERKSMOOTH	136
LINEARABS	137
LINEARINC	137
MOVABS	138
MOVINC	138
MULTIABS	139
MULTIINC	140
NORMAL	141
RESRIFLOC	141
SETINDEXINTERP	141
SETLABELINTERP	141
SETPFLY	141
SETPFLYCHAINSTRAT	142
SETPZERO	142
SETPZEROCHAINSTRAT	143
SETQUOTE	143
SETQUOTECHAINSTRAT	143
SETRIFLOC	144
SETTOLERANCE	144
START	146
STARTINTERP	146
STOP	146
SWITCHENC	147
WAITACC	147
WAITCOLL	148
WAITDEC	148
WAITREG	149
WAITSTILL	149
WAITTARGET	149
WAITWIN	149
Параметры оси	150
Считывание / Запись	150
DEVICEID	150
GETAXIS	150
Движение Точка-точка	156
SETACC	156
SETDEC	157
SETDERIV	157
SETFEED	157
SETFEEDF	157
SETFEEDFA	158
SETINTEG	158
SETMULTIFEED	158
SETPROP	158
SETSLOPE	158
SETVEL	159
Интерполированное Движение	159

	LOOKAHEAD	159
	SETACCI	159
	SETACCLIMIT	160
	SETACCSTRATEGY	160
	SETAXPARTYPE	160
	SETCONTORNATURE	161
	SETDECI	161
	SETDERIVI	161
	SETFEEDFAI	162
	SETFEEDI	162
	SETFEEDFI	162
	SETINTEGI	162
	SETPROPI	163
	SETSLOPEI	163
	SETSLOWPARAM	163
	SETVELI	164
	SETVELILIMIT	164
	Скоординированное Движение	164
	SETFEEDCOORD	164
	SETOFFSET	166
	Объединенное Движение	166
	RATIO	166
	SETDYNRATIO	167
	Общие Параметры	167
	DYNLIMIT	167
	ENABLESTARTCONTROL	168
	NOTCHFILTER	168
	RESLIMNEG	168
	RESLIMPOS	169
	SETADJUST	169
	SETBACKLASH	169
	SETBIGWINFACTOR	171
	SETDEADBAND	171
	SETENCLIMIT	172
	SETINDEXEN	172
	SETINTEGTIME	172
	SETIRMPP	172
	SETLIMNEG	173
	SETLIMPOS	173
	SETMAXER	173
	SETMAXERNEG	173
	SETMAXERPOS	174
	SETMAXERTYPE	175
	SETPHASESINV	176
	SETREFINV	176
	SETRESOLUTION	176
10.3.5	Счетчики	177
	DECOUNTER	177
	INCOUNTER	177
	SETCOUNTER	177
10.3.6	Таймеры	177
	HOLDTIMER	177
	SETTIMER	177
	STARTTIMER	178
10.3.7	Переменные, Векторы и Матрицы	178
	CLEAR	178
	FIND	178

	FINDB	179
	LASTELEM	179
	LOCAL	179
	MOVEMAT	180
	PARAM	180
	SETVAL	181
	SORT	181
10.3.8	Строки	181
	ADDSTRING	181
	CONTROLCHAR	182
	LEFT	182
	LEN	182
	MID	183
	RIGHT	183
	SEARCH	183
	SETSTRING	183
	STR	184
	VAL	184
10.3.9	Связь	184
	CLEARRECEIVE	184
	COMCLEARRXBUFFER	184
	COMCLOSE	185
	COMGETERROR	185
	COMGETRXCOUNT	185
	COMOPEN	185
	COMREAD	186
	COMREADSTRING	186
	COMWRITE	187
	COMWRITESTRING	187
	RECEIVE	187
	SEND	192
	SENDIPC	198
	WAITIPC	198
	WAITRECEIVE	199
10.3.10	Математика	199
	ABS	199
	ADD	199
	AND	199
	ARCCOS	200
	ARCSIN	200
	ARCTAN	200
	COS	200
	DIV	201
	EXP	201
	EXPR	201
	LOG	202
	LOGDEC	203
	MOD	203
	MUL	203
	NOT	204
	OR	204
	RANDOM	204
	RESETBIT	205
	ROUND	205
	SETBIT	206
	SHIFTL	206
	SHIFTR	207

	SIN	209
	SQR	210
	SUB	210
	TAN	210
	TRUNC	211
	XOR	211
10.3.11	Многозадачность	211
	ENDMAIL	211
	ENDREALTIMETASK	212
	ENDTASK	212
	GETPRIORITYLEVEL	212
	GETREALTIME	212
	GETREALTIMECOUNT	212
	HOLDTASK	213
	RESUMETASK	213
	SENDMAIL	213
	SETPRIORITYLEVEL	214
	STARTREALTIMETASK	214
	STARTTASK	214
	STOPTASK	214
	WAITMAIL	215
	WAITTASK	215
10.3.12	Управление потоком	215
	CALL	215
	DELONFLAG	215
	DELONINPUT	216
	FCALL	216
	FOR/NEXT	216
	FRET	217
	GOTO	217
	IF/IFVALUE/IF-THEN-ELSE	218
	IFACC	219
	IFAND	219
	IFBIT	219
	IFBLACKBOX	220
	IFCHANGEVEL	220
	IFCOUNTER	221
	IFDEC	221
	IFDIR	221
	IFERRAN	222
	IFERROR	223
	IFFLAG	223
	IFINPUT	224
	IFMESSAGE	224
	IFOR	225
	IFOUTPUT	225
	IFQUOTER	226
	IFQUOTET	227
	IFRECEIVED	227
	IFREG	227
	IFSAME	228
	IFSTILL	228
	IFSTR	228
	IFTARGET	229
	IFTASKHOLD	229
	IFTASKRUN	229
	IFTIMER	230

	IFVEL	230
	IFWIN	231
	IFXOR	231
	ONERRSYS	232
	ONFLAG	232
	ONINPUT	233
	REPEAT/ENDREP	233
	RET	234
	SELECT	234
	TESTIPC	235
	TESTMAIL	235
10.3.13	Разные	236
	CLEARERRORS	236
	CLEARMESSAGES	236
	DEFMSG	236
	DELAY	237
	DELERROR	238
	DELMESSAGE	238
	ERROR	238
	IFDEF/ELSEDEF/ENDDEF	240
	MESSAGE	242
	SYSFAULT	244
	SYSOK	244
	TYPEOF	244
	WATCHDOG	244
10.3.14	MECHATROLINK-II	245
	MECCOMMAND	245
	MECGETPARAM	246
	MECGETSTATUS	246
	MECSETPARAM	248
10.3.15	Стандартная полевая шина	249
	AXCONTROL	249
	AXSTATUS	250
	CNBYDEVICE	252
	READDICTIONARY	252
	WRITEDICTIONARY	252
10.3.16	EtherCAT	253
	ACTIVATEMODE	253
	ECATGETREGISTER	253
	ECATSETREGISTER	253
	GETPDO	253
	SETEOE	254
	SETPDO	254
10.3.17	Платы TMSbus управлением CAN	254
	GETCNSTATE	254
	GETSDOERROR	255
	GETMNSTATE	255
	RECEIVEPDO	255
	SENDPDO	255
	SETNMTSTATE	256
10.3.18	Моделирование	256
	DISABLE	256
	DISABLEFORCEDINPUT	256
	ENABLE	257
	ENABLEFORCEDINPUT	257
	RESETFORCEDINPUT	257
	SETFORCEDANALOG	257

	SETFORCEDINPUT	257
	SETFORCEDPORT	258
10.3.19	BlackBox	258
	ENDBLACKBOX	258
	PAUSEBLACKBOX	258
	STARTBLACKBOX	259
10.3.20	ISO	259
	ISOG0	259
	ISOG1	260
	ISOG9	260
	ISOG90	261
	ISOG91	261
	ISOG93	261
	ISOG94	261
	ISOG216	262
	ISOG217	262
	ISOM2	263
	ISOM6	263
	ISOSETPARAM	264
	KINEMATICEXP	265
10.3.21	Больше недоступные инструкции	266
10.3.22	Инструкции, которые не могут использоваться на событии	267
10.4	Примеры	268
10.4.1	Обнуление при прерывании	268
10.4.2	Сервер перемещения осей	269
10.4.3	Цикл Главной функции с управлением ошибками	271
10.4.4	Операции на строках	272
10.4.5	Последовательное / параллельное выполнение	273
10.4.6	Стандартная процедура обнуления	273
10.4.7	Движения ISO	274

1 Введение

1.1 Правила пользования справочником

В данном справочнике приводится описание функций ЧПУ Albatros.

Справочник составлен таким образом, чтобы позволить оператору понять систему и ее применение.

Основными пунктами каждого раздела справочника являются:

- окна и инструменты Albatros.
- описание типовой Архитектуры системы Albatros.
- порядок отображения устройств и работы на них в ручном режиме и режиме диагностики, используя обзорный Экран.
- порядок отображения и изменения Технологических и Геометрических Параметров и Параметров инструментов
- порядок отображения Устройств и воздействия на них в ручном режиме и режиме диагностики

Во избежание утяжеления данного справочника, просим обращаться к инструкциям на Операционную Систему Windows для полного понимания механизмов использования *мыши, меню, панелей инструментов* и всех классических рабочих функций Windows.

1.2 Рабочие окна

Рабочие окна бывают различных типов и зависят от типа выполняемой операции. Одновременно может быть открыто несколько окон.

Окна делятся на следующие типы:

Окно	Описание
<i>Главное</i>	главное окно Albatros. Позволяет вызывать функции и содержит все остальные окна, содержимое которых зависит от конкретного представляемого ими применения.
Обзорный экран	содержит графическое отображение станка или его частей и позволяет работать на них.
Технологические Параметры	позволяет отображать и менять технологические параметры.
Параметры инструментов	позволяет отображать и менять параметры инструментов.
Диагностика	позволяет отображать состояние устройств, а там, где это допустимо - также работать на них.
Ошибки системы	окно со списком последних обнаруженных ошибок системы. Можно также отобразить ошибки цикла и сообщения.
Конфигурация Системы	позволяет отображать и менять логических и физических устройств станка

2 Состав системы

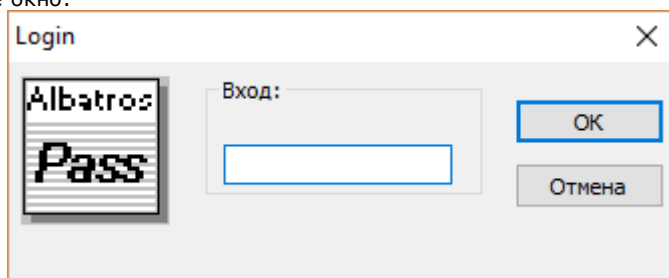
2.1 Права доступа к системе

Albatros предусматривает 4 уровня доступа к системе:

- Пользователь: это уровень с максимальными ограничениями доступа. Невозможно изменить какие-либо настройки устройств. Этот уровень используется для выполнения обработок и нормальных операций станка. При подключении системы автоматически подключается этот уровень доступа.
- Сервис: этот уровень используется для выполнения планового техобслуживания станка. Оператор должен иметь возможность изменять некоторые параметры конфигурации, но не вносить изменения в структуру станка.
- Изготовитель: этот уровень используется для конфигурации установок и станков. На этом уровне могут вноситься почти все возможные изменения. Используется разработчиками.
- Тра: это максимальный уровень доступа к системе. Служит для защиты доступа к некоторым особо ответственным настройкам, изменение которых требует глубокого знания Albatros. Используется очень редко. Пароль доступа к этому уровню должен запрашиваться непосредственно в ТРА.

Для доступа к системе с уровнем доступа выше уровня Пользователя или для приведения системы на уровень Пользователя после внесения изменений на более высоком уровне, необходимо ввести соответствующий пароль.

Для вызова окна для ввода пароля используется сочетание клавиш **CTRL+*** (звездочка). В качестве альтернативы в правой части **Строки приложений** Windows имеется значок . При щелчке правой клавиши мыши на этом значке можно отобразить меню с пунктом **Изменить уровень пароля**. Открывается следующее окно:



Окно входа в систему

Теперь необходимо ввести пароль и нажать кнопку **[ОК]** для подтверждения. Вместо образующих пароль символов отображаются символы "*", с тем, чтобы никто не мог прочитать только что введенный пароль. После ввода пароля вы сразу попадаете на соответствующий уровень доступа. Для подтверждения уровня доступа можно выбрать пункт **Информация о Albatros** из меню ?.


Если введенный пароль неправилен, отображается сообщение об ошибке "Внимание! Неправильный пароль!".

2.2 Многоязыковая поддержка

Albatros поддерживает отображение текста на различных языках.

Переключение языка

Переключение языка может осуществляться на любом [уровне доступа](#) в систему. Для изменения

выбранного языка необходимо использовать сочетание клавиш **CTRL + /** или щелкнуть значок  в "строке приложений" Windows.

В открывающемся окне выбрать нужный язык и щелкнуть кнопку **[ОК]**.

Язык меняется не сразу, а при следующем запуске Albatros.

2.3 Типовая архитектура системы

Ввиду того, что многие аспекты при графическом отображении и в структуре базовых данных станка в большой степени зависят от типа станка, помимо информации общего характера, данное руководство предоставляет описание состава типовой системы только в качестве примера.

Реальные индикации, схемы и графические страницы системы как таковой зависят, естественно, от конкретного применения и поэтому предусматриваются производителем станка.

ЧПУ Albatros образовано ПК-супервизором, отображающим интерфейс оператор-машина, и переменным числом модулей (от 1 до 16) для управления всеми рабочими ресурсами станка или установки.

Таким образом, можно иметь два типа установки:

С одним модулем, состоящей только из одного модуля, подключенного непосредственно к шине ПК.

С несколькими модулями, состоящей минимум из 1 и максимум из 16 модулей и предусмотренной только для применения на установках или линиях с несколькими станками. Блок ПК в этом случае физически отделен от модулей, которые могут находиться в различных точках линии или установки.

В архитектуре обоих типов модули состоят из одной или нескольких плат осей для прямого управления осями станка и логического управления устройствами входа/выхода.

В варианте с одним модулем платы осей установлены непосредственно на ПК-супервизоре, в варианте с несколькими модулями они установлены в промышленный ПК (с или без монитора и клавиатуры), соединенный с ПК-супервизором по сети ethernet. На рисунке ниже приводится схема подключения между ПК-супервизором и удаленным модулем. Кроме того, приводятся основные виды деятельности, выполняемые различными компонентами.

СИСТЕМА ПОДКЛЮЧЕНИЯ УДАЛЕННОГО МОДУЛЯ



Супервизор ПК
с Microsoft Windows

Интерфейс Albatros



Числовое Программное Управление
с Windows Embedded
Windows CE или RTX

Связь GPL
Интерполяции



Плата управления
полевой ШИНЫ

Обновление входов/выходов
Управление Осями

Интеллектуальные удаленные устройства управляют устройствами входа/выхода и осями (удаленный TRS-AX) непосредственно на станке. Эти устройства обеспечивают считывание линий Цифровых (ВКЛ. / ОТКЛ.) или Аналоговых Входов и обновление линий Цифровых или Аналоговых Выходов и подключены к Модулям при помощи GreenBUS (последовательная шина RS485 - 1 Мбод) шины CAN и Ethercat. Работа Albatros в станке защищена аппаратным ключом USB, сконфигурированным компанией TPA.

2.4 Организация и логическая конфигурация

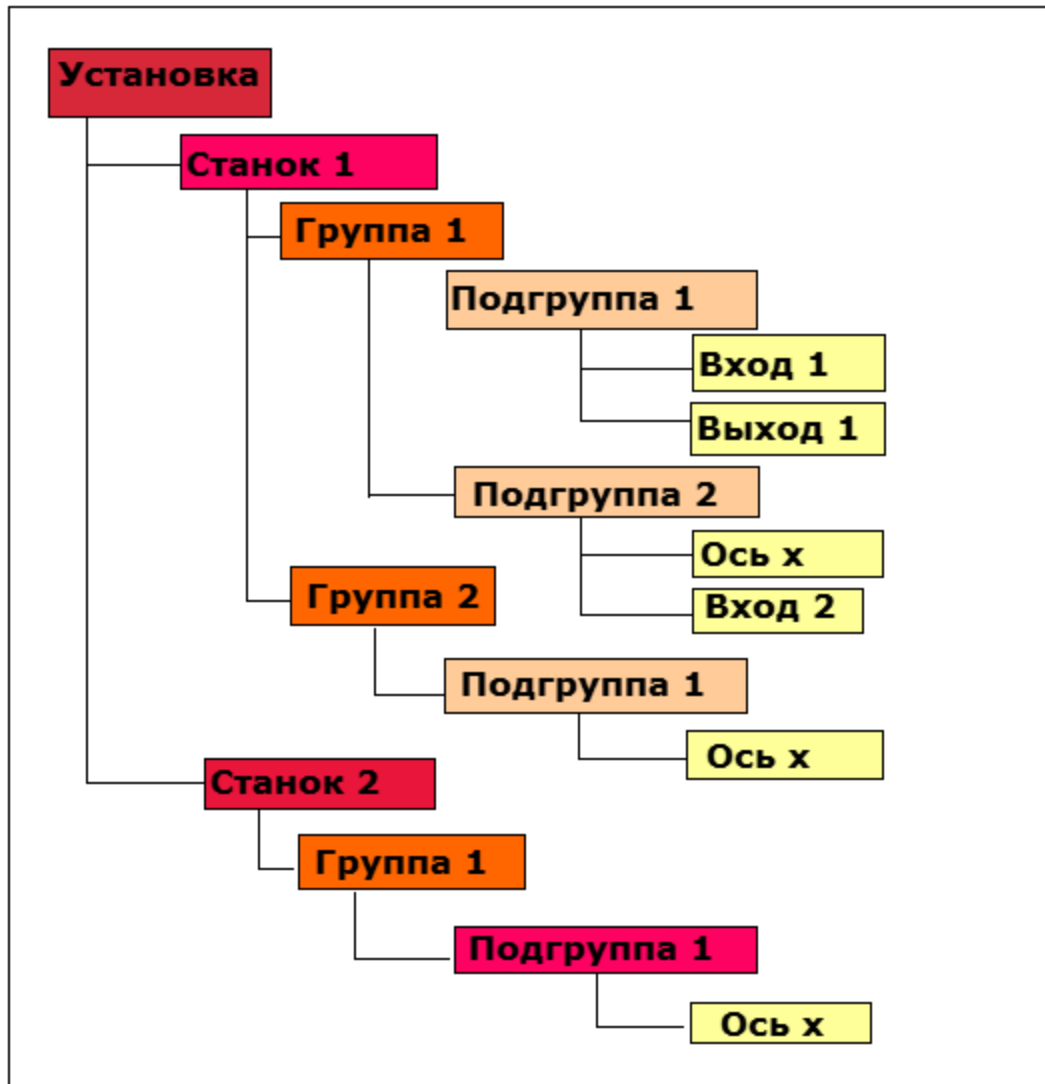
В системе Albatros описательная структура установки или отдельного станка организована в технологическом архиве с иерархической структурой.

Эта настройка отражает потребность в сохранении, на уровне данных конфигурации и порядка доступа, возможной модульной структуры станков, классифицируя ее в плане динамичной агрегации различных модулей, агрегатов и устройств, которые можно вводить или исключать в зависимости от возможного различного оснащения.

Следуя этой логической структуре, в самом общем сложном случае, мы имеем верхний иерархический уровень, образованный следующим:

- 1. Установка** это ничто иное, как совокупность станков. Подразумевается как совокупность рабочих элементов, управляемых ЧПУ. Установка всегда присутствует, даже в случае одного станка, и не должна определяться специально.
- 2. Станок** рассматривается с "логической" точки зрения как совокупность устройств (осей, таймеров и пр.) и циклов управления, т. е. кода, написанного в GPL, реализующего алгоритмы управления самого станка. Как правило, в станке имеется большое число устройств, которые организованы в группы
- 3. Группы** это "емкости", позволяющие организовать компоненты машины в зависимости от логических критериев. Например, можно определить группу "оси", содержащую все оси станка, конечные выключатели, циклы, которые выполняют обнуление осей и пр.
- 4. Подгруппы** представляют собой дальнейшую специализацию группы. Например, группа "оси" может быть разделена на "цифровые оси" и "шаговые оси".
- 5. Устройства** представляют собой самый низкий уровень иерархии. Они являются логическим представлением электрических и механических компонентов станка и независимы от расположенного ниже аппаратного обеспечения.

На рисунке ниже схематично представлена структура воображаемой установки, состоящей из двух станков:



Пример иерархической организации установки.

ПРИМЕЧАНИЕ: Группы могут и не подразделяться на Подгруппы и образовываться непосредственно Устройствами.

Для доступа к некоторым функциям, например, Диагностика, Конфигурация системы, Параметры, в случае установки с несколькими станками необходимо выбрать станок, для которого требуется отображать данные.






2.5 Устройства

Устройства можно разделить на две категории: физические устройства и логические устройства. В системе все устройства определяются именем, обозначающим его применение.

Физические устройства













Под физическими устройствами понимаются все элементы, действующие на электрические или пневматические части станка или определяющие их состояние:

Символ	Устройство	Назначение
	Цифровой вход	определяет состояние "включен" или "отключен" устройства. Например, предохранительный выключатель дверки.
	Цифровой выход	подключает или отключает устройство, приводя его в состояние "включен" или "отключен". Может использоваться, например, для

Символ	Устройство	Назначение
	Аналоговый вход	управления электроклапаном.
	Аналоговый выход	определяет значение напряжения на входе в соответствующий зажим. Например, значение, генерированное тахогенератором.
	Порт входов	задает напряжение на выходе из соответствующего зажима. Может использоваться, например, для управления инвертором.
	Порт выходов	состоит из 8 цифровых входных линий.
	Ось	состоит из 8 цифровых выходных линий.
		управляет движением электрической оси. Можно управлять осями различного типа: с аналоговым управлением, с цифровым управлением, шаговыми двигателями, осями подсчета (только считывание кодера).

Логические устройства

Логические устройства - это элементы, действующие исключительно в рамках рабочих программ и не имеющие физической ответной стороны:

Символ	Устройство	Назначение
	Таймер	элемент подсчета времени. Единица измерения - секунда. Имеет разрешение в 4 мс. Может принимать только положительные значения, а максимальное отображаемое значение составляет 8.589.934 секунды (ок. 99 суток) (с реальным временем 250 Гц). Его значение записывается в энергонезависимую память платы осей.
	Счетчик	элемент подсчета операций. Может принимать значение в диапазоне от - 2.147.483.648 до +2.147.483.647. Его значение записывается в энергонезависимую память платы осей.
	Флаговый бит	элемент, который может принимать значение "включен" или "отключен".
	Флажковый выключатель	это особые флажки, которые могут быть связаны с некоторыми кнопками панели инструментов, например, флажок Пуск.
	Флаговый порт	состоит из 8 строк флагового бита.
	Переменная	глобальная переменная типа <i>integer</i> кода GPL.
	Переменная	глобальная переменная типа <i>char</i> кода GPL
	Переменная	глобальная переменная типа <i>float</i> кода GPL.
	Переменная	глобальная переменная типа <i>double</i> кода GPL.
	Переменная	глобальная переменная типа <i>строка</i> кода GPL.
	Переменная	глобальная переменная типа <i>массив</i> кода GPL.
	Переменная	глобальная переменная типа <i>матрица</i> кода GPL.

3 Обзорный Экран

3.1 Использование Обзорного Экрана

При работе станка можно открыть окно *Обзорный экран*, позволяющее отображать состояние самых значимых устройств.

Информация, отображаемая на обзорных экранах, совпадает с информацией в окне диагностики. Однако если в окне диагностики информация представляется в древовидной структуре (включающей все устройства станка), то обзорные экраны позволяют представлять информацию графически (отображая, например, картинку станка и показывая координаты осей). Кроме того, обзорные экраны позволяют выбирать наиболее значимую информацию и группировать другую информацию во второстепенные экраны, которые могут вызываться пользователем.

3.2 Порядок работы на Обзорном Экране

В целях диагностики Оператор может выбрать различные образующие обзорный экран страницы, *два раза* щелкая мышью на одной из зон станка, выделенной на рисунке пунктирным прямоугольником, которые также называются "горячие зоны".

Для выделения "горячей зоны", устройства или оси достаточно переместить указатель мыши на изображение нужного объекта. Одновременно в Строке Состояния отображается имя устройства, по которому проходит указатель мыши.

Указатель мыши приобретает различные формы в зависимости от типа указываемого объекта, указывая, таким образом, разрешенные на этом объекте действия. Эти формы могут быть следующими:



лупа

если указатель находится в "горячей зоне"



рука

если это устройство выхода



текстовый курсор

если это окошко, в котором можно задать значение

3.3 Порядок воздействия на Устройства

Воздействие на устройства обеспечивается путем установки указателя мыши на нужное устройство, завершая затем действие в соответствии с описанным ниже (зависит от типа устройства).

Порядок представления	Действие	Устройство
<i>Значок устройства</i>	установить указатель мыши и <i>щелкнуть</i>	Цифровой выход Флажковый выключатель Флажковый бит
<i>Клетка задания значения</i>	установить указатель мыши, <i>щелкнуть</i> и задать значение	Аналоговый выход Выходной порт Флажковый порт Координата оси Таймер Счетчик




3.4 Перемещение Осей в Ручном режиме

Для получения доступа к функциям перемещения осей в ручном режиме необходимо иметь соответствующие [права доступа](#). Эти права определяются производителем станка.

Для взаимодействия с одной из осей достаточно *два раза щелкнуть* мышью в нужном поле отображения координат оси. При этом открывается окно для перемещения оси. В случае осей Виртуального, Шагового типа и Подсчета окно содержит сжатый объем информации. Например, если ось - типа Подсчета, отображаются только значения "Реальная координата" и "Скорость".



В окне имеется две зоны, которые содержат:





Зона отображения

- Три окна с отображением *Реальной координаты оси [мм]*, ее *Скорости [м/мин.]* и *Погрешности контура* или преследования.
- Две кнопки выбора, отображающие *Состояние* оси (*Free* = с разомкнутым контуром, напр., вследствие ошибки системы, *Normal* = с замкнутым контуром, т. е. в нормальном состоянии управления положением). Эти кнопки позволяют также задавать состояние.
- Сигнализация, при движении, *Состояния* оси (напр., *Ускорение*).
- Две кнопки для выполнения перемещения оси в *Направлении* отрицательном  или положительном .
- Кнопка  для Остановки, в любой момент, движения оси при перемещениях в режимах Абсолютный или Шаговый.


Зона перемещения

- Два окна для задания *Отрицательной координаты* и *Положительной координаты*, которые будут использоваться в режиме *Абсолютный*.
- Окно для задания *Скорости*, с которой должна перемещаться ось при перемещении в ручном режиме.
- Три кнопки для выбора режима выполнения перемещения: *Толч. ход*, *Абсолютный* или *Шаговый*.
- Окно для задания значения *Шага* для использования в *Шаговом* режиме.

Для перемещения оси необходимо соответствующим образом задать описанные параметры. Необходимо выбрать режим перемещения и нажать кнопку  (для перемещения оси в положительном направлении) или кнопку  (для перемещения оси в отрицательном направлении).

В режиме *Толчковый ход* ось перемещается до тех пор, пока остается нажатой кнопка  или кнопка . В *Шаговом* режиме ось будет перемещаться на величину, указанную в окошке "Шаг", каждый раз при нажатии кнопки  или кнопки .

В режиме *Абсолютный* ось сразу дойдет до координаты, заданной в окошке "Положит. коорд." или "Отриц. коорд."

Вместо кнопок ,  и  можно использовать кнопки "+" (или Ctrl+P), "-" (или Ctrl+M) и "клавишу пробела" клавиатуры.

4 Технологические параметры и Параметры инструментов

4.1 Окно "Технологические параметры"

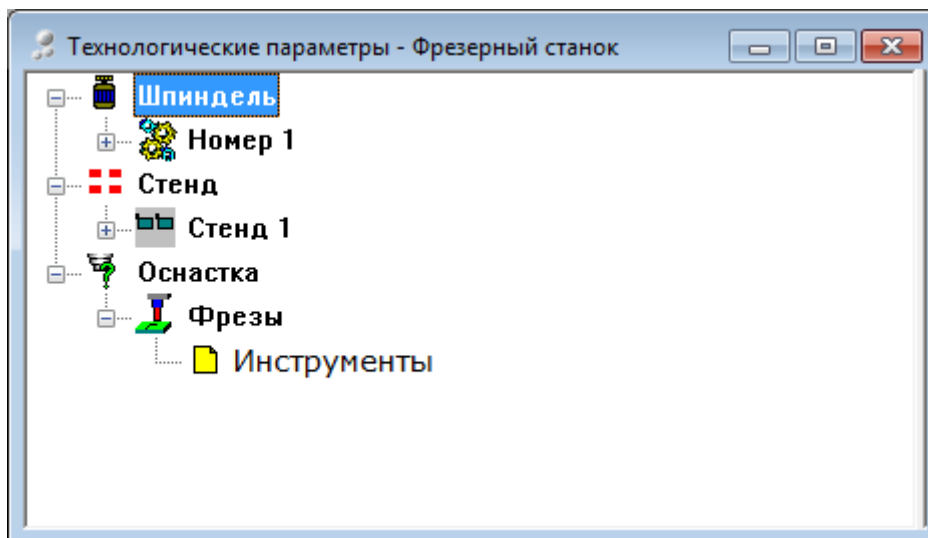
Архив "Технологических параметров" позволяет записывать в память информацию геометрического и технологического характера, относящуюся к станку. Эта информация необходима ЧПУ для правильного управления функционированием станка.

Это окно открывается из меню **Файл ->Открыть технологические параметры**.



Технологические параметры обычно разбиты на Группы / Подгруппы (как правило, группы и подгруппы Технологических параметров не связаны с группами и подгруппами, на которые разделены устройства станка). Режимы отображения определяются Изготовителем Станка и дифференцируются в зависимости от конкретного применения.

Имеющиеся в архиве значения, как правило, задаются производителем на этапе калибровки Станка и обычно пользователь может изменить их только случайным образом. Поэтому некоторые данные могут защищаться паролем для предупреждения непредвиденных изменений, которые могут нарушить работу системы.

Окно Технологических параметров в древовидной структуре отображает все Группы и Подгруппы образующих архив параметров, как показано на следующем рисунке.



Структура архива Технологических параметров

Это окно содержит отображаемые в виде древовидной структуры некоторые Группы с соответствующими Подгруппами параметров. Древовидная структура может разворачиваться или сворачиваться кнопками  и  каждого узла. Открытие и закрытие частей древовидной структуры может выполняться также клавишами: +, -, кнопками со стрелкой вправо и влево.

Порядок работы с Технологическими параметрами

После открытия дерева нужной Группы / Подгруппы открывается страница с данными.

Данные могут отображаться в форме таблицы, в текстовых окнах или в окнах выбора; это зависит от типа данных и от того, как они были заданы производителем.

При внесении изменений в данные необходимо нажать кнопку **[OK]** для записи изменений в память.

Оснастка

Особый случай задания данных станка - оснастка. Обычно информация, относящаяся к совокупности инструментов, которыми оснащен станок (оснастка), записывается в память в архиве Технологических параметров. Информация, относящаяся к инструментам, записывается в память в архиве Параметров Инструментов.

Таким образом, для определения оснащения станка необходимо связать информацию в этих двух архивах. Если в приложении это предусмотрено, можно будет вызвать из архива Технологических параметров

информацию архива Параметров инструментов. Как правило, связь устанавливается кнопкой со значком, подобным показанному ниже.

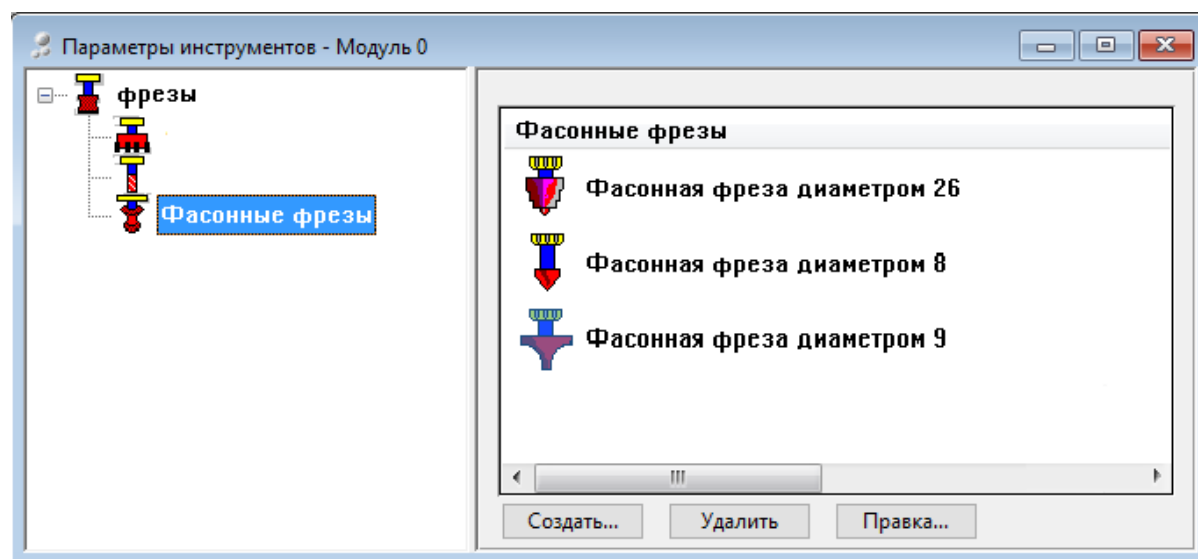


При выборе этого значка и двойном щелчке левой кнопкой мыши открывается окно со списком инструментов, определенных в архиве Параметров инструментов; из него можно выбрать нужный инструмент. После выбора значок кнопки меняется и отображается значок, относящийся к конкретному инструменту.

Кроме того, можно отобразить данные, относящиеся к инструменту, два раза щелкнув на значке правой кнопкой мыши.



4.2 Окно "Параметры инструментов"

Окно "Параметры инструментов" открывается из меню **Файл->Открыть параметры инструментов**. Параметры инструментов, предусмотренные Производителем на основе обработок, которые может выполнять станок, обычно организованы, как показано на рисунке ниже:



Пример окна "Параметры инструментов"

Окно "Параметры инструментов" состоит из двух зон:

- *зона слева* содержит группы с соответствующими подгруппами инструментов, которые отображаются с древовидной структурой. Древовидная структура может разворачиваться или свертываться кнопками  и  каждого узла. Например, можно иметь группу "Фрезы", состоящую из подгрупп фрез с различными характеристиками, например, фасонные фрезы, фрезы с поворотом в горизонтальной плоскости и т. д. Каждая из этих подгрупп имеет присвоенный один или несколько инструментов, характеристики которых задаются в диалоговом окне, определенном производителем. Инструменты, имеющиеся в каждой подгруппе, отображаются в правой зоне экрана.
- *зона справа*, заголовком которой является название выбранной подгруппы, содержит список инструментов, входящих в эту подгруппу. Определенные здесь инструменты не обязательно присутствуют на станке. Связь между инструментом и положением в станке (оснащение) обычно выполняется в архиве технологических параметров.

Порядок воздействия на Параметры инструментов

Операции ввода, изменения и удаления инструментов из архива возможны благодаря *кнопкам*, находящимся в нижней части окна:

[Создать...]	позволяет вводить в подгруппу новый инструмент. Открывает диалоговое окно "Новый инструмент", в котором можно задать следующие данные: - <i>Описание</i> : сообщение, идентифицирующее инструмент. Описание может выбираться из уже имеющихся в списке, при условии, что оно уже не было присвоено другому инструменту, или же можно определить новое. - <i>Изображение</i> : это значок, идентифицирующий инструмент. Может выбираться из уже имеющихся в списке или же может вызываться из папки при помощи кнопки [Изображение] . Инструмент может вводиться с соблюдением алфавитного порядка описаний.
[Удалить]	позволяет удалить инструмент из подгруппы при подтверждении. Соответствующее описание не удаляется и остается доступным для другого инструмента.
[Правка...]	позволяет заменить <i>описание</i> или <i>изображение</i> выбранного инструмента, при этом отображается то же диалоговое окно, которое описано в команде [Создать...] .

5 Диагностика

5.1 Окно "Диагностика"

На этапе выполнения станка можно открыть окно *Диагностика*, позволяющее оператору держать под контролем рабочее состояние станка, контролируя логическое состояние цифровых входных/выходных сигналов, значение аналоговых входов/выходов, значение счетчиков и таймеров и перемещение осей. В зависимости от присвоенных изготовителем [прав доступа](#) можно также изменить состояние устройств.

Если это позволяет уровень доступа, можно иметь в реальном времени:

- отображение состояния (ВКЛ./ВЫКЛ.) всех цифровых входных и выходных сигналов
- возможность подключать и отключать сигналы цифровых выходов
- отображение текущих значений (в интервале +/- 10 В) аналоговых входов
- возможность задавать значение (в интервале +/- 10 В) всем аналоговым выходам
- возможность перемещать оси в ручном режиме, выбирая скорость, значение шага или абсолютной конечной координаты с отображением реального положения, скорости и погрешности контура.
- отображение и изменение глобальных переменных

В последующих параграфах подробно описываются устройства и глобальные переменные и их графическое отображение.



ПРИМЕЧАНИЕ: В окне диагностики отображаются исключительно устройства, подключенные для текущего уровня доступа.

5.2 Состав окна "Диагностика"


Со ссылкой на структуру "Группы / Подгруппы", которая уже описывалась в главе [Состав системы](#), можно получить доступ к устройствам, которые затем отображаются в древовидной структуре. Основой структуры является группа, представленная значком



, за которым следует Имя и Комментарий.

Структуру можно развернуть или свернуть *щелчком* кнопки  или . Открытие и закрытие составляющих дерева частей может выполняться также клавишами: +, -, кнопками со стрелкой вправо и влево.

При открытии Группы в дереве отображаются

- "Список устройств" Группы, представленный символом 
- возможные образующие Группу Подгруппы.

При открытии одной из Подгрупп отображаются образующие ее устройства.

5.3 Представление Устройств

Для каждого из отображаемых устройств представляется следующая информация:

- графический символ;
- текущее состояние и значение;
- имя;
- комментарий.

Ниже представлено графическое отображение устройств, тип устройства и величина, отображаемая в режиме реального времени.

Состояние цифровых входов, цифровых выходов и флажков графически отображается индикатором, который окрашивается определенным цветом в зависимости от того, подключен или отключен вход.

Для портов, т. е. нескольких одновременно представленных линий (8), будет отображаться ряд индикаторов. Первая линия группы представлена крайним правым индикатором, последняя - крайним левым.

Устройство	Символ	Состояние	Отображение в реальном времени
	л		

Цифровой вход			состояние: активен = ЗЕЛЕНЫЙ, неактивен = СЕРЫЙ
Цифровой выход			состояние: активен = КРАСНЫЙ, неактивен = СЕРЫЙ
Аналоговый вход		22.000	значение тока
Аналоговый выход		22.000	цифровое значение тока в Вольт
Порт входов			состояние каждой линии (как Цифровой вход). Состояние: активен = ЗЕЛЕНЫЙ, неактивен = СЕРЫЙ
Порт выходов			состояние каждой линии (как Цифровой выход). Состояние: активен = КРАСНЫЙ, неактивен = СЕРЫЙ
Ось		100.000	абсолютное текущее положение
Таймер		12.000	текущее значение в секундах
Счетчик		58	текущее цифровое значение
Флаговый бит			состояние: активен = ЖЕЛТЫЙ, неактивен = СЕРЫЙ
Флажковый выключатель			состояние (как Флаговый бит). Состояние: активен = ЖЕЛТЫЙ, неактивен = СЕРЫЙ
Флаговый порт			состояние каждой линии (как Флаговый бит). Состояние: активен = ЖЕЛТЫЙ, неактивен = СЕРЫЙ
Глобальная переменная		2	глобальная переменная типа integer кода GPL.
Глобальная переменная		127	глобальная переменная типа char кода GPL
Глобальная переменная		50.00000000	глобальная переменная типа float кода GPL.
Глобальная переменная		200.00000000	глобальная переменная типа double кода GPL.
Глобальная переменная		Area	глобальная переменная типа строка кода GPL.
Глобальная переменная		[256]	глобальная переменная типа массив кода GPL.
Глобальная переменная		[10][3]	глобальная переменная типа матрица кода GPL.

5.4 Взаимодействие с Устройствами

В целях диагностики можно взаимодействовать с устройством для считывания его состояния или изменения его значения. Это невозможно для некоторых типов устройств, например, входных устройств и устройств, защищенных изготовителем. В этом случае сообщение проинформирует оператора, если тот попытается выполнить действия на устройстве.

После выбора устройства *два раза щелкнуть* его мышью или нажать кнопку **ВВОД**, или **Клавишу пробела** для доступа к окну, позволяющему менять состояние или значение этого устройства.

Если это **цифровой выход** или **флаговый бит**, не отображается какого-либо окна, а сразу же меняется состояние устройства. Работа выхода сигнализируется изменением цвета индикатора состояния этого выхода.

В случае **порта выходов** необходимо установить курсор мыши на индикатор, соответствующий нужному выходу, и *щелкнуть два раза* для изменения состояния. Это относится и к **флажковым выключателям**, и к **флаговому порту**.

Для **аналоговых выходов**, **таймеров** и **счетчиков** открывается диалоговое окно, где отображается текущее значение и где можно задать новое значение, которое нужно немедленно присвоить устройству.

Режим взаимодействия с **Осью** описывается в пункте [Перемещение осей в ручном режиме](#).

5.5 Список клавиш для навигации по древовидной структуре

Клавиша	Описание
Стрелка вверх	перемещает выбор на строку, непосредственно до или после данной
Стрелка вниз	
Стрелка вправо	разворачивает на один уровень выбранную ветвь, а если она уже развернута, перемещает выбор на следующую ветвь

Стрелка влево	закрывает выбранную ветвь или, если она уже закрыта, - перемещает выбор на предыдущую ветвь
+	разворачивает на один уровень выбранную ветвь
-	закрывает выбранную ветвь
*	разворачивает все уровни выбранной ветви
CTRL+ALT+SHIFT и ВВОД	отображает таблицы корректоров линейности, присвоенных оси. Если сочетание клавиш подключено при выбранной оси в дереве устройств модуля,
CTRL+ALT+SHIFT и щелчок левой кнопки мыши	отображаются все корректоры, присвоенные оси, как матрица, в которой столбцы - это соответствующие оси (первый столбец - столбец автокорректоров), а строки - значения коррекции. Измененные значения учитываются при движении оси, но не записываются в память на диске.

5.6 Корректоры линейности

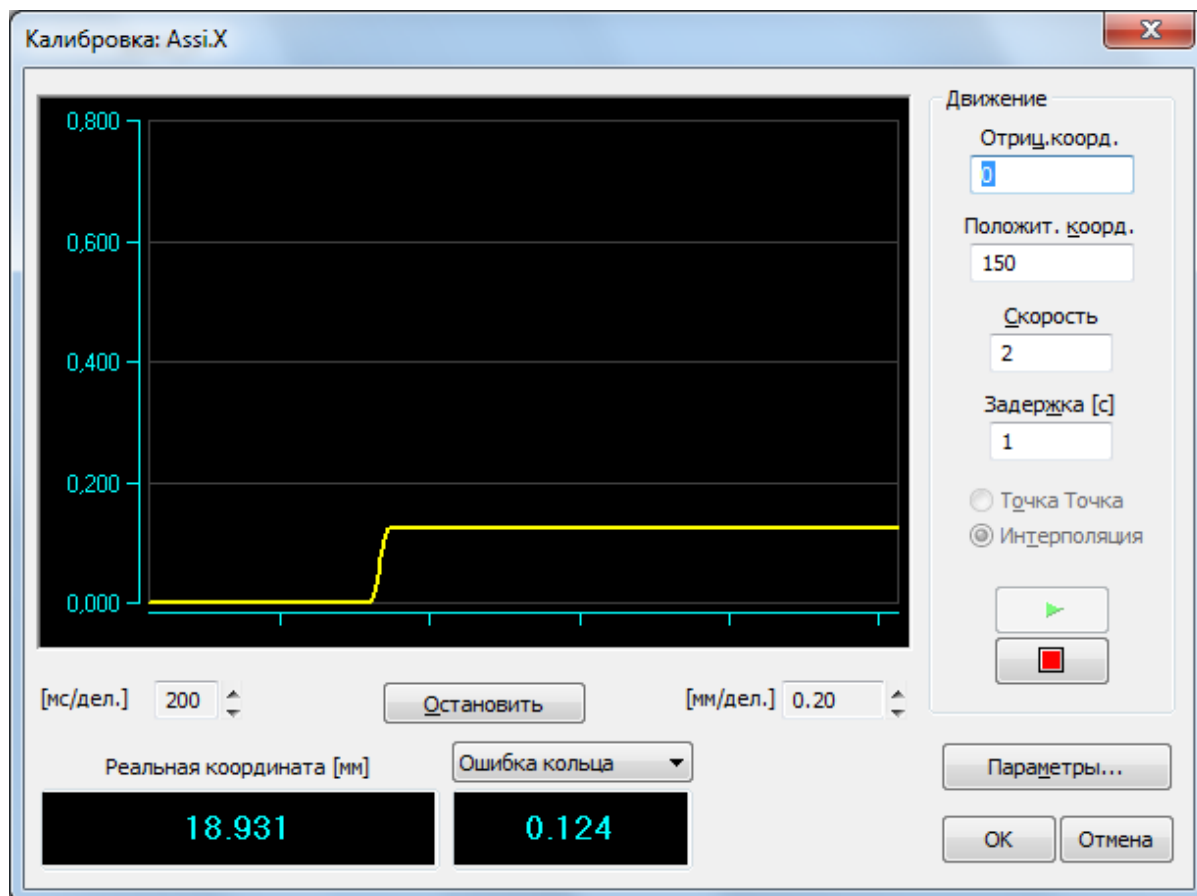
В диагностике можно просмотреть и изменить корректоры линейности выбранной оси, открыв контекстное меню и выбрав пункт [Корректоры линейности](#). Элемент отображается только в том случае, если в конфигурации были определены корректоры линейности. В качестве альтернативы меню можно использовать комбинацию клавиш **[CTRL+SHIFT+ВВОД]**.

5.7 Планшет калибровки осей

Планшет калибровки осей представляет собой инструмент, позволяющий менять параметры конфигурации оси и одновременно перемещать ось и видеть отображенной для нее погрешность контура на виртуальном осциллографе.

Для доступа к планшету калибровки осей необходимо иметь [уровень доступа](#) к системе, больший или равный уровню "Изготовитель". Доступ к планшету калибровки дается из режима диагностики или вручную двойным щелчком на калибруемой оси при удерживании нажатой кнопки **[SHIFT]** или из контекстуального меню при выборе пункта "Калибровка".

Так можно получить доступ к планшету калибровки, который представлен на рисунке ниже:



Для проверки поведения оси при изменении параметров следует постоянно перемещаться между двумя предельными координатами, которые называются **Положительная координата** и **Отрицательная координата**.

Помимо этих параметров должна задаваться **Скорость** перемещения оси. На первых этапах калибровки рекомендуется использовать низкое значение скорости. Кроме того, можно задать **Задержку**, которая будет применяться между двумя последовательными движениями.

В окне осциллографа будет отображаться график погрешности контура или одна из других величин оси. Как и в стационарных осциллографах, можно соответствующим образом масштабировать график для его адаптации к размерам окна или для выделения деталей. При помощи мыши и управляющих клавиш можно еще раз посмотреть последнюю минуту калибровки, отобразить один или два курсора для выполнения измерений и проверок на выборочных данных, увеличить участок графика для анализа деталей выборочных данных, менять сдвиг и масштаб по оси абсцисс и по оси ординат. Кроме того, нажатием кнопки **Остановить** можно заблокировать прокрутку графика для его спокойного рассмотрения без необходимости остановки оси.

Помимо графика, в двух ячейках отображаются реальная координата (слева) и представленная на графике величина (справа). Эта величина может задаваться при помощи комбинированного окна, расположенного над ячейкой отображения.

Для изменения параметров оси необходимо нажать кнопку **[Параметры...]**. При этом открывается окно, представленное на рисунке ниже, в котором можно изменить налету большую часть параметров оси. После внесения изменений в один или несколько параметров их можно будет применить, нажимая кнопку **[Применить]**.

Эта кнопка позволяет сразу же увидеть эффект изменения динамики оси. При нажатии **[ОК]** изменения принимаются, а при нажатии **[Отмена]** восстанавливаются значения, существовавшие перед открытием окна **[Параметры...]**

Обычно регулируются следующие параметры:

- **Пропорциональный** коэффициент
- **Интегративный** коэффициент
- **Производный** коэффициент
- **Подача вперед**: процент текущей скорости, подаваемый непосредственно на привод (независимо от погрешности контура)
- **Подача вперед уск.**: эталонный процент скорости, подаваемый непосредственно на привод на этапах ускорения и замедления оси (дополнительно к feed forward)
- **Ускорение**: продолжительность линейной функции ускорения
- **Замедление**: продолжительность линейной функции замедления

Курсор в графической области

Курсор - это инструмент, который позволяет измерять и отображать некоторые данные трассировки. Он состоит из вертикальной линии цвета, связанного с выбранной трассировкой. Вы можете перемещаться по графику с помощью клавиш управления курсором или с помощью мыши. Во всплывающей подсказке, связанной с вертикальной линией, могут отображаться различные значения, выбираемые из всплывающего меню, которое можно вызвать, щелкнув правой кнопкой мыши рядом с курсором.

Во всплывающем меню вы можете выбрать параметры:

- **Канал**: представляет список трассировок, отображаемых на графике, и выделяет галочкой ту, на которую ссылается курсор. Также можно выбрать новую трассировку, которая будет связана с курсором.
- **Стиль**: представляет список данных, которые могут отображаться в прямоугольнике всплывающей подсказки.
- **Значение X-Y**: отображает значение на оси X и на оси Y точки трассировки, где находится курсор.
- **Значение X**: отображает значение на оси X точки трассировки, в которой находится курсор.
- **Значение Y**: отображает значение на оси Y точки трассировки, в которой находится курсор.
- **Период**: отображает значение расстояния между двумя отображаемыми курсорами при измерении вдоль оси X.
- **Пик - пик**: отображает значение расстояния между двумя отображаемыми курсорами при измерении между двумя точками вдоль оси Y.
- **Частота**: отображает обратное расстояние по оси X.
- **Опции**: настроить как отображается курсор и соответствующая подсказка

- **Использовать цвет канала:** если он включен, курсор рисуется в цвете выбранных данных, к которым он относится, в противном случае выбор цвета, с которым отображается курсор, является случайным.
- **Скрыть подсказку на выпуске:** если эта опция включена, всплывающая подсказка отображается только до тех пор, пока нажата левая кнопка мыши, а затем скрыта.
- **Инvertировать центровку:** подсказка обычно отображается справа от курсора. Если опция включена, курсор отображается слева от курсора.
- **Позиционирование по образцу:** если включено, курсор помещается только на выбранные значения.

Калибровка оси

Калибровка оси - это ответственная операция, которая должна выполняться с максимальным вниманием и осторожностью.

С помощью пункта CalibSampleTime в разделе [Albatros] в Tra.ini можно изменить период выборки данных оси для окна Калибровка. Значение указано в миллисекундах; оно не может быть меньше частоты контроля осей и не может быть больше 100.

Перед калибровкой оси с планшета необходимо задать все параметры в конфигурации, а также задать предел шкалы для скорости привода. Для аналоговых осей значение напряжения, которому в Albatros соответствует максимальная скорость, составляет 9 Вольт.

Для предупреждения повреждения станка при использовании неправильных параметров рекомендуется задать низкую скорость, равную, например, 10% максимальной предусмотренной для оси скорости. При этом даже при задании повышенного усиления не будет слишком бурной реакции оси.

Как правило, выполняется первая калибровка для движений точка-точка, а затем - калибровка для интерполированных движений.

Первая операция, если это еще не было выполнено при конфигурации, - это задание продолжительности ускорения и замедления. Чем больше эта продолжительность, тем меньше будет действующее на ось ускорение.

Затем задается минимальное усиление, позволяющее перемещать ось. Это служит для проверки правильности калибровки привода.

Albatros сконфигурирован на обеспечение эталона в 9 Вольт при максимальной скорости, заданной в конфигурации оси. Например, при перемещении оси со скоростью, равной 10% от максимальной, если привод калиброван правильно, будет считываться эталонное напряжение, равное 10% от максимального, т. е. 0,9 Вольт. Если этого эталонного напряжения не получается, необходимо изменить предел измерения привода.

После калибровки привода необходимо постепенно, все также с максимальным вниманием, начать повышать усиление контура положения. Время от времени следует проверять отсутствие условий избыточного удлинения и неустойчивости. Скорость на этом этапе должна всегда поддерживаться ниже и равной 10% максимальной. На этом этапе рекомендуется всегда глубоко анализировать получаемый профиль скорости при помощи виртуального осциллографа, увеличивая, по возможности, изображение для отображения деталей.

После того, как поведение оси станет достаточно быстрым и устойчивым, постепенно можно повысить скорость, каждый раз проверяя оси и корректируя, при необходимости, усиление. Усиление и скорость никогда не должны повышаться резко. Условия калибровки, которые могут казаться устойчивыми на низких скоростях, могут не быть таковыми на более высоких скоростях.

После определения оптимального значения усиления при необходимости можно постепенно увеличивать интегрирующий и производный коэффициенты, а затем - Подача вперед для снижения погрешности контура и его приведения к значениям, приемлемым с точки зрения точности оси. Подача вперед позволяет почти полностью устранить погрешность контура на режимном этапе перемещения, но не на этапах ускорения и замедления. Чтобы еще больше уменьшить ошибку контура в этих фазах, вы можете увеличить ускорение подачи вперед. Как правило, даже очень низкие значения этого параметра достаточны для получения хорошего результата.

Что касается калибровки оси для интерполированных движений, то можно использовать значения, уже определенные для движения точка-точка, однако необходимо учитывать наличие других осей станка. В частности, для обеспечения высокой точности при интерполированных движениях необходимо сбалансировать погрешности контура осей. Поэтому следует определить ось с наиболее высокой погрешностью контура (при одинаковой скорости) или "ухудшить" (применительно к параметрам интерполяции) калибровку других так, чтобы погрешности контура оказались одинаковыми.

6 Ошибки и Сигналы

6.1 Введение

Albatros управляет событиями об ошибках и отчетах.

Ошибки системы

Это ошибки, которые система Albatros может выявить автоматически как на этапе выполнения программ, так и при операциях техобслуживания и диагностики установки.

Эти ошибки имеют разную природу и включают проблемы от управления осями до проблем, которые могут возникнуть при выполнении программы.

Ошибки системы могут управляться непосредственно в рабочих программах при помощи инструкции [ONERSYS](#), в противном случае при их возникновении завершается выполнение программ на модуле, на котором произошла ошибка. Объяснение каждой системной ошибки описано на следующих страницах данного руководства.

Ошибки цикла

Это ошибки, которые возникают во время выполнения программы, но, как правило, позволяют ее продолжение после устранения ошибки. Ошибки цикла могут генерироваться с помощью инструкции GPL [ERROR](#).

Сообщения

Это предупреждающие сообщения, которые генерируются в определенных ситуациях выполнения программы, или сигналы запроса вмешательства со стороны оператора, но не останавливающие выполнение самой программы. Сообщения могут быть сгенерированы с помощью инструкции GPL [MESSAGE](#).

Сигналы

Это общие сигналы, не связанные с машинным циклом и написанные Albatros. Они не отображаются в Albatros. Описание каждого сигнала представлено на следующих страницах данного руководства.

Строка ошибок

На строке ошибок отображается последняя системная ошибка, произошедшая в порядке времени, вместе с последней ошибкой цикла и последним сообщением.

Ошибки системы выделяются красным цветом.

Ошибки цикла выделяются желтым цветом. Это ошибки, которые возникают при выполнении программы, но, как правило, позволяющие ее продолжение за счет устранения ошибки.

Сообщения выделяются зеленым цветом.

Приемник TRS-AX 1: отключен	00002
Обнуление осей еще не было выполнено	00001
Идет инициализация	00001

Строка ошибок

Ошибки, выявленные после включения системы, отображаются в окне, которое можно открыть двойным щелчком мыши на *Строке ошибок* или с меню **Вид**. Кроме того, в этом окне отображается дополнительная информация об ошибках системы.

Это окно разделено на следующие участки: В **верхней части** отображается следующая информация:

- Тип: определяет тип ошибки (ошибка системы, ошибка цикла и сообщение).
- Время и Дата: это время и дата возникновения ошибки.
- Описание: описание ошибки.
- Код: номер сообщения об ошибке.
- Задача: это имя задачи, генерировавшей ошибку (не присутствует на Строке ошибок).

При *двойном щелчке* на одном из этих столбцов информация упорядочивается в соответствии с содержимым этого столбца.

В **нижней части** имеются следующие окна:

- Ошибки цикла: при подключении отображаются и эти ошибки.
- Сообщения: при подключении отображаются и сообщения.
- Все: при подключении отображаются сообщения всех модулей системы, связанные с типом отображаемой информации.
- Ячейка имени модуля: отображает имя модуля, для которого выполняется отображение, позволяя, кроме того, выбирать, при системе с несколькими модулями, модуль, для которого необходимо отобразить информацию.

Наконец, имеются следующие **кнопки** управления:

- **[Удалить все]** удаляет всю информацию, отображаемую памятью, но не удаляет ее из архива.
- **[Удалить]** позволяет удалять текущую информацию из памяти, не удаляя ее из архива.
- **[ОК]** позволяет закрыть окно.

Хранение ошибок и сигналов в файле отчета

Все ошибки записываются в файл для их архивного воспроизведения. Это текстовый файл в формате TSV. Имя файла - MONTH (номер текущего месяца) .TER. В комплект Albatros входит программа ViewRER, которая загружает и отображает файлы .TER.

6.2 Ошибки системы

6.2.1 Ошибки, генерированные управлением осей

1 ИмяОси: неправильное подключение кодера

Причина:

При остановленной оси возникла разница между теоретической координатой и реальной координатой оси больше 1024 шагов кодера.

Обычно это происходит при пуске оси в эксплуатацию, когда инвертированы между собой фазы кодера.

При нормальном функционировании наблюдается, когда при выключенном приводе ось перемещается вручную, если предварительно она не была переключена в FREE, или когда ось, вследствие неправильной калибровки, подвергается чрезмерному удлинению на этапе прибытия на координату (выброс).

После этой ошибки обнуляется эталонный сигнал и ось приводится в состояние FREE.

Способ устранения:

На этапе пуска оси в эксплуатацию проверить подключение фаз кодера соответствующей оси (при необходимости подключить опцию инверсии фаз кодера в конфигурации оси). Проверить калибровку оси специальным режимом Диагностики.

2 ИмяОси: движение не завершено

Причина:

В конце перемещения, спустя 5 секунд после завершения теоретического движения, разница между теоретической координатой и реальной координатой оказалась больше окна, указанного в конфигурации. Это может просто обуславливаться тем, что привод выключен или не подключен, или же плохой регулировкой сдвига привода. Может, в любом случае, обуславливаться также механическими зазорами на оси или слишком низким усилением контура положения оси.

Способ устранения:

Проверить, что привод включен и подключен. Проверить калибровку оси и отрегулировать сдвиг привода соответствующей оси.

3 ИмяОси: сервоошибка

Причина:

Во время любого типа движения разница между теоретической координатой и реальной координатой оси стала больше максимальной погрешности, указанной в конфигурации, или погрешности, заданной инструкцией [SETMAXER](#).

Обычно обуславливается плохой регулировкой усиления контура положения или предела измерения скорости привода или слишком высокой инерцией оси.

Способ устранения:

Проверить регулировку усиления и предел измерения скорости привода.

Проверить правильность функционирования кодера и группы двигатель/привод.

Кроме того, проверить наличие возможных механических блокировок.

4 ИмяОси: выход за положительный предел

Причина:

Теоретическая координата оси вышла за предельную положительную координату, указанную в Конфигурации или заданную инструкцией [SETLIMPOS](#).

Способ устранения:

Исправить в программе координату, выходящую за предельную положительную координату, или снова определить пределы координаты оси.

5 ИмяОси: выход за отрицательный предел

Причина:

Теоретическая координата оси вышла за предельную отрицательную координату, указанную в Конфигурации или заданную инструкцией [SETLIMNEG](#).

Способ устранения:

Исправить в программе координату, выходящую за предельную отрицательную координату, или снова определить пределы координаты оси.

10 ИмяОси: Выполнение в реальном времени быстрее построения профиля

Причина:

Выполнение в реальном времени профиля движения быстрее генерации GPL этого профиля. Буфер опережающего просмотра опорожняется быстрее, чем наполняется. Эта ошибка может обуславливаться двумя обычно сопутствующими причинами:

- скорость интерполяции слишком велика по сравнению с размерами проходимых отрезков
- проходимые отрезки слишком короткие.

Способ устранения:

Проверить, что заданная скорость интерполяции не слишком велика по сравнению с размерами проходимых отрезков и что проходимые отрезки интерполяции не слишком короткие.

6.2.2 Ошибки, генерированные управлением удаленных модулей Входов/Выходов

2049 Приемник номер: неправильная конфигурация

Причина:

Удаленный приемник получил конфигурацию, отличную от определенной на оборудовании. Это может произойти в случае, если удаленный приемник не такой, какой был выбран в конфигурации аппаратного обеспечения Albatros. Например, удаленный приемник - Albre16, а в Albatros был сконфигурирован удаленный Albre24(GreenBus v3.0), или TRS-IO с неправильным числом расширений TRS-IO-E (GreenBus v4.0).

Способ устранения:

Проверить конфигурацию аппаратного обеспечения.

2050 Приемник номер: отключен

Причина:

Удаленный приемник не реагирует на команды передатчика.

Способ устранения:

Проверить питание приемника и последовательную связь.

2051 Приемник номер: снова подключен

Причина:

Подключение между передатчиком и приемником было восстановлено.

2052 Приемник номер: ошибка повторного считывания неподключенного выхода номер НомерВыхода

Причина:

На указанном цифровом выходе сработала защита или имеется КЗ, в любом случае, он находится в состоянии, отличном от предусмотренного блоком управления. Выход не связан с каким-либо логическим устройством в Конфигурации виртуальное-физическое, поэтому это является признаком непоследовательности между конфигурацией и фактическим электромонтажом станка.

Способ устранения:

Проверить Конфигурацию виртуальное-физическое. Устранить КЗ или проверить, что прилагаемая нагрузка не превышает максимальные пределы (см. техническую документацию).

2054 Приемник номер: неправильный тип**Причина:**

На этапе инициализации удаленных модулей на определенном адресе был выявлен приемник, отличный от указанного в конфигурации.

Способ устранения:

Проверить, что конфигурация аппаратного обеспечения соответствует настройкам удаленных модулей.

2055 Приемник номер: инициализирован**Причина:**

Приемник подключился к передатчику после отключения вследствие отсутствия питания.

2056 Приемник номер: ошибка питания +24 В пост. тока**Причина:**

Полевое питание (+24 В пост. тока) удаленного модуля входов/выходов неактивно или работает неправильно.

Способ устранения:

Проверить работу питания +24 В пост. тока.

2057 Ошибка питания GreenBUS**Причина:**

Плохо работает питание полевой шины, соединяющей удаленные модули входов/выходов с блоком управления. Это питание должно иметь номинальное значение +12 В пост. тока и подаваться от блока управления.

Способ устранения:

Проверить наличие питания GreenBus, проверить кабели GreenBus. Выключить и снова включить. При необходимости заменить плату управления.

**2058 Приемник номер: ошибка повторного считывания ТипУстройства
ИмяУстройства****Причина:**

Состояние указанного выхода не соответствует заданному. Это может обуславливаться наличием КЗ, срабатыванием защиты (избыточная нагрузка) или просто отсутствием питания. Указанный выход может быть цифровым выходом, аналоговым выходом, выходом управления оси. Тип выхода указывается в отображении ошибки.

Способ устранения:

Если это цифровой выход, проверить питание +24 В (сторона оборудования), устранить возможные КЗ или ситуации повышенного потребления выхода (см. техническую документацию). Если это аналоговый выход или выход управления осями, проверить наличие и значение заданного напряжения на выходе (тестером или осциллографом), устранить возможные КЗ или ситуации повышенного потребления выхода (см. техническую документацию).

2059 Не пройден тест двухпортовой памяти передатчика**Причина:**

Произошла ошибка при тестах, выполнявшихся на этапе инициализации платы осей. В частности, не удалась инициализация передатчика GreenBus (микроконтроллер i296). Может обуславливаться неправильной конфигурацией адресов входов/выходов и IRQ платы или конфликтом с другой периферией системы. Наконец, это может быть следствием повреждения платы осей.

Способ устранения:

Проверить конфигурацию платы, проверить отсутствие конфликтов с другой периферией. При использовании в удаленном режиме снова передать встроенное ПО на модуль. Квалифицированные

специалисты могут выполнить тест аппаратного обеспечения двухпортовой памяти микроконтроллера i296. Если проблема возникает снова, обращаться к производителю.

2060 Ошибка при инициализации передатчика

Причина:

Произошла ошибка при тестах, выполнявшихся на этапе инициализации платы осей. В частности, не удалась отправка встроенного ПО на передатчик GreenBUS (микроконтроллер i296). Может обуславливаться неправильной конфигурацией адресов входов/выходов и IRQ платы или конфликтом с другой периферией системы. Наконец, это может быть следствием повреждения платы осей.

Способ устранения:

Проверить конфигурацию платы, проверить отсутствие конфликтов с другой периферией. При использовании в удаленном режиме снова передать встроенное ПО на модуль. Квалифицированные специалисты могут выполнить тест аппаратного обеспечения двухпортовой памяти микроконтроллера i296. Если проблема возникает снова, обращаться к производителю.

2061 Ошибка при передаче встроенного ПО на передатчик

Причина:

Произошла ошибка при тестах, выполнявшихся на этапе инициализации платы осей. В частности, не удалась отправка на передатчик GreenBus (микроконтроллер i296) конфигурации удаленных модулей входов/выходов.

Способ устранения:

Проверить конфигурацию аппаратного обеспечения, при использовании удаленного модуля снова передать встроенное ПО на модуль. Квалифицированные специалисты могут выполнить тест аппаратного обеспечения ОЗУ микроконтроллера i296. Если проблема возникает снова, обращаться к производителю.

2062 Ошибка при передаче конфигурации на передатчик

Причина:

Произошла ошибка при тестах, выполнявшихся на этапе инициализации платы осей. В частности, не удалась инициализация удаленных модулей входов/выходов.

Способ устранения:

Проверить конфигурацию аппаратного обеспечения, при использовании удаленного модуля снова передать встроенное ПО на модуль. Квалифицированные специалисты могут выполнить тест аппаратного обеспечения ОЗУ микроконтроллера i296. Если проблема возникает снова, обращаться к производителю.

2063 Ошибка при передаче конфигурации на приемник

Причина:

Произошла ошибка при инициализации удаленного модуля.

Способ устранения:

Проверить конфигурацию аппаратного обеспечения. Квалифицированные специалисты могут выполнить тест аппаратного обеспечения удаленного модуля. Если проблема возникает снова, обращаться в сервисную службу.

2064 Приемник номер: несовместимая версия встроенного ПО

Причина:

Версия встроенного ПО удаленного приемника несовместима со встроенным ПО блока управления.

Способ устранения:

Проверить установку блока управления. Если проблема не устраняется, обратиться в сервисную службу.

2065 Приемник номер: ошибка асинхронной связи

Причина:

Произошла ошибка или не получен ответ при связи блока управления с удаленным модулем (GreenBus v4.0).

Способ устранения:

Проверить подключения и питание GreenBus. Если проблема не устраняется, обратиться в сервисную службу.

2066 Приемник номер: общая ошибка**Причина:**

Произошла общая ошибка при передаче события или сигнала тревоги от удаленного модуля (GreenBus v4.0)

Способ устранения:

Проверить подключения и питание GreenBus. Если проблема не устраняется, обратиться в сервисную службу.

2067 Приемник номер: ошибка при передаче конфигурации**Причина:**

Произошла ошибка связи при передаче данных конфигурации удаленному модулю (GreenBus v4.0).

Способ устранения:

Проверить подключения и питание GreenBus. Выключить и снова включить. Если проблема не устраняется, обратиться в сервисную службу.

2068 Приемник номер: внутренняя ошибка № номерошибки**Причина:**

Произошла внутренняя ошибка указанного удаленного модуля.

Способ устранения:

Обратиться к изготовителю станка.

2069 Приемник номер: ошибка питания +24 В пост. тока стенд номер**Причина:**

Полевое питание (+24 В пост. тока) выходной группы, подключенной к тому же зажиму, что и питание, неактивно или работает неправильно.

Способ устранения:

Проверить работу питания +24 В пост. тока.

6.2.3 Ошибки, генерированные управлением МЕCHATROLINK-II**2308 Плата НомерПлаты: инициализация не удалась вследствие неправильного задания параметра конфигурации****Причина:**

В параметре "Сопоставление виртуальной и физической конфигурации" не была сопоставлена ни одна ось (логическое устройство) с платой с шиной МЕCHATROLINK-II (физическое устройство).

Способ устранения:

Проверить назначения в параметре "Сопоставление виртуальной и физической конфигурации".

2341 Плата НомерПлаты: число сервоприводов превышает максимально допустимое значение**Причина:**

К плате с шиной МЕCHATROLINK-II было подключено слишком большое число сервоприводов по сравнению с заданной конфигурацией.

Способ устранения:

Проверить в конфигурации системы значение "Частота контроля осей".

В таблице приведены правильные значения, которые нужно задать, в зависимости от числа сервоприводов, управляемых платой.

Плата	Частота контроля осей (Гц)	Максимальное число сервоприводов
AlbMech	1000	8

AlbMech	<=500	16
DualMech Mono	1000	8
DualMech Mono	500	20
DualMech Mono	250	30
DualMech	1000	16
DualMech	500	40
DualMech	250	60

2342 Плата НомерПлаты: аппаратный адрес сервопривода Серво превышает максимальное разрешенное значение

Причина:

На плате с шиной MECHATROLINK-II одна ось (логическое устройство) была сопоставлена аппаратному адресу (физическому устройству), превышающему число сервоприводов, управляемых платой.

Способ устранения:

Проверить в конфигурации системы значение "Частота контроля осей". В таблице приведены правильные значения, которые нужно задать, в зависимости от числа сервоприводов, управляемых платой.

Плата	Частота контроля осей (Гц)	Максимальное число сервоприводов
AlbMech	1000	8
AlbMech	<=500	16
DualMech Mono	1000	8
DualMech Mono	500	20
DualMech Mono	250	30
DualMech	1000	16
DualMech	500	40
DualMech	250	60

В параметре "Сопоставление виртуальной и физической конфигурации" проверить связь между логическим устройством и физическим устройством. Например, если максимальное число сервоприводов равняется 8, то сопоставление между логическим устройством и физическим устройством должно находиться в пределах первых 8 осей (от Ax1 до Ax8).

2349 Плата НомерПлаты: сервопривод Серво не подключен

Причина:

Прервано физическое подключение к сервоприводу указанной платы

Способ устранения:

Проверить электромонтаж шины MECHATROLINK-II и сервопривод.

6.2.4 Ошибки, генерированные управлением CanBUS

2761 Узел номер: отключен

Причина:

Указанный узел CAN представляется в данный момент не подключенным к полевой шине, относящейся к указанной плате, несмотря на то, что он присутствует в конфигурации.

2762 Узел номер: снова подключен

Причина:

Указанный узел CAN представляется только что подключенным к полевой шине, относящейся к указанной плате.

2763 Ошибка отсутствия передачи

Причина:

Внутренняя ошибка платы, обозначенной. Не удалась передача данных на указанный узел CAN.

Способ устранения:

Обратиться к изготовителю станка.

2764 Узел номер: ошибка отсутствия приема

Причина:

Не удался прием данных, ожидаемых от указанного узла CAN.

Способ устранения:

Проверить подключение и питание указанного устройства CAN. Проверить электромонтаж всей линии CAN. Проверить подключение линии к ЧПУ. Проверить соответствие между заданиями протокола указанного устройства CAN по сравнению с заданиями передатчика в ЧПУ (скорость в бодах, адрес, настройки, характерные для принятого протокола).

2765 Узел номер: инициализирован

Причина:

Указанный узел CAN был подключен к полевой шине. Затем он был инициализирован правильно.

2766 Условие отказа на интерфейсе CAN

Причина:

Сигнализируется отказ внутреннего питания интерфейсного устройства CAN.

Способ устранения:

Обратиться к изготовителю станка.

2767 Ошибка утраты состояния CANopen

Причина:

Из-за серьезной проблемы передатчик CAN больше не находится в рабочем состоянии

Способ устранения:

Обратиться к изготовителю станка.

2768 Узел номер: ошибка отсутствия приема PDO

Причина:

Числовое управление не получило ожидаемого PDO, что ждал от узла CAN.

Способ устранения:

Проверьте:

- питание узла.
- что узел не остался в предоперационном режиме.
- данные PDO и шины CAN, настроенные в Albatros.

2769 Узел номер: ошибка приема неконфигурированного узла

Причина:

Обнаружено присутствие узла в сети CAN, которого не было объявлено в аппаратной конфигурации Albatros.

Способ устранения:

Проверьте аппаратный адрес узла и адрес узла, установленного в аппаратной конфигурации.

Проверьте, что узел был объявлен в аппаратной конфигурации, в противном случае необходимо добавить его.

2770 Узел номер: неправильная конфигурация

Причина:

Описание данных и RPDO и TPDO неправильно.

Способ устранения:

Исправьте описание данных pdo передачи и приема в аппаратной конфигурации.

2771 Узел номер: ошибка связи SDO

Причина:

Указанный узел Can не ответил при асинхронной связи (SDO).

Способ устранения:

Проверить состояние подключения узла. Если проблема не устраняется, обратиться к изготовителю.

2772 Таймаут цикла CAN опроса узлов

Причина:

Произошла ошибка таймаута цикла CAN опроса узлов

Способ устранения:

Изменить в аппаратной конфигурации значение времени заданной выборки.

3073 Узел номер: аварийная Ошибка № Номерашибки

Причина:

Устройство CANopen обнаружило ситуацию ошибки узла, указанную отображенным кодом. Код ошибки - шестнадцатеричное число. Это ситуации ошибки, связанные с отдельным узлом, соответствующие стандарту CiA DS301-протокол EMERGENCY.

Способ устранения:

См. документацию на узел.

3074 Узел номер: общая ошибка CAN № Номерашибки

Причина:

Произошла внутренняя ошибка указанного модуля. Код ошибки - шестнадцатеричное число.

Способ устранения:

Обратиться к производителю.

3088 Плата CAN номер: узел НомерУзел: ошибка связи SDO № Номерашибки - описание

Причина:

В инструкции READDICTIONARY или WRITEDICTIONARY не удалось выполнить один или несколько запросов на чтение/запись по протоколу SDO. Невыполнение команд может быть вызвано, например, запросом на чтение объекта CANOpen, не реализованного в устройстве, к которому обращаются; или это может быть связано с записью в регистр CANOpen данного, не совместимого с типом объекта (к примеру, попытка записать строку в объект, тип которого задан как целое число). Код ошибки соответствует спецификации DS402, наряду с числовым кодом выводится также текстовое описание. Код ошибки - шестнадцатеричное число.

Способ устранения:

Проверить правильность параметров Скорости передачи (BaudRate), Время выборки и т.д., заданных в аппаратной конфигурации и параметров команд [READDICTIONARY](#) и/или [WRITEDICTIONARY](#), содержащихся в коде GPL.

6.2.5 Ошибки, генерированные управлением шиной EtherCAT

3329 Ошибка инициализации гнезда связи

Причина:

Встроенное ПО не смогло установить связь с сетевой платой.

Способ устранения:

Если плата была сконфигурирована в системе RTX, проверить, что файлы .ini, присутствующие в подпапке встроенного ПО Albatros, написаны правильно. Для проверки синтаксиса файлов см. руководство по установке RTX в Albatros (InstallationRTXGuide.pdf).

3330 Ошибка при сканировании сети EtherCAT

Причина:

При предварительном сканировании сети EtherCAT главная плата не получила ответа от некоторых или ото всех сконфигурированных ведомых модулей или конфигурация не соответствует фактической сети EtherCAT, присутствующей в поле.

Способ устранения:

Проверить электромонтаж между главным модулем EtherCAT и ведомыми модулями.
Проверить описания устройств в аппаратной конфигурации. Для выявления ошибки поможет окно Диагностика аппаратного обеспечения. В этом окне отображаются имеющиеся узлы, а при неправильной конфигурации, помимо имени найденного устройства, отображается имя ожидаемого устройства.

3331 Ошибка конфигурации почтового ящика передачи

Причина:

Узел EtherCAT не ответил на команду, поданную главным модулем. Причины могут быть следующими: отсутствует связь, неисправен узел...

Способ устранения:

Проверить электромонтаж и работу в удаленном режиме.

3332 Ошибка конфигурации почтового ящика приема

Причина:

Узел EtherCAT не ответил на команду, поданную главным модулем. Причины могут быть следующими: отсутствует связь, неисправен узел...

Способ устранения:

Проверить электромонтаж и работу в удаленном режиме.

3333 Плата EtherCAT номер: ошибка типа расширений узла НомерУзел

Причина:

Тип расширений, сконфигурированных в узле EtherCAT в аппаратной конфигурации, не соответствует типу реально имеющихся расширений. (Например, в аппаратной конфигурации определен TRS-CAT с расширением TRS-IO-E, а в системе имеется TRS-CAT с расширением TRS-AN-E).

Способ устранения:

Проверить, что устройства, определенные в аппаратной конфигурации, соответствуют имеющимся.

3334 Ошибка на этапе конфигурации PDO

Причина:

Узел EtherCAT, для которого делалась попытка сконфигурировать PDO, не присутствует в сети или неисправен.

Способ устранения:

Проверить, что конфигурация сети EtherCAT, описанная в аппаратной конфигурации Albatros, соответствует физической конфигурации сети.

3335 Узел НомерУзел в аварийной ситуации (НомерОшибки)

Причина:

Указанный узел находится в аварийной ситуации.

Способ устранения:

Проверить код сигнала тревоги по следующей таблице

Код сигнала тревоги	Описание
0x0001	Неустановленная ошибка
0x0002	Нет памяти
0x0011	Недействительное изменение запрошенного состояния
0x0012	Неизвестное запрошенное состояние

0x0013	Программа самозагрузки не поддерживается
0x0014	Нет действительного встроенного ПО
0x0015	Недействительная конфигурация почтового ящика
0x0016	Недействительная конфигурация почтового ящика
0x0017	Недействительная конфигурация менеджера синхронизации
0x0018	Нет действительных входов
0x0019	Нет действительных выходов
0x001A	Ошибка синхронизации
0x001B	Сторожевая схема менеджера синхронизации
0x001C	Недействительные типы менеджера синхронизации
0x001D	Недействительная конфигурация выхода
0x001E	Недействительная конфигурация входа
0x001F	Недействительная конфигурация сторожевой схемы
0x0020	Ведомый модуль требует холодного пуска
0x0021	Ведомый модуль требует INIT
0x0022	Ведомый модуль требует PREOP
0x0023	Ведомый модуль требует SAFEOP
0x0024	Недействительное отображение входа
0x0025	Недействительное отображение выхода
0x0026	Непоследовательные настройки
0x0027	Холостой ход не поддерживается
0x0028	Синхронизация не поддерживается
0x0029	Для холостого хода требуется 3-буферный режим
0x002A	Фоновая сторожевая схема
0x002B	Нет действительных входов и выходов
0x002C	Фатальная ошибка синхронизации
0x002D	Нет ошибки синхронизации
0x0030	Недействительная конфигурация DC SYNCH
0x0031	Недействительная конфигурация защелки DC
0x0032	Ошибка PLL
0x0033	Ошибка недействительных входов/выходов DC
0x0034	Ошибка недействительного таймаута DC
0x0035	Недействительное время цикла синхр. DC
0x0036	Время цикла DC Sync0
0x0037	Время цикла DC Sync1
0x0041	MBX_AOE
0x0042	MBX_EOE
0x0043	MBX_COE
0x0044	MBX_FOE
0x0045	MBX_SOE
0x004F	MBX_VOE
0x0050	Нет доступа к EEPROM

0x0051	Ошибка EEPROM
0x0060	Перезапуск вед. модуля локально

3336 Плата EtherCAT номер: число расширений узла НомерУзел неправильно

Причина:

Число расширений, сконфигурированных в узле EtherCAT в Albatros, не соответствует числу имеющихся расширений. (Например, в аппаратной конфигурации был определен TRS-CAT с двумя расширениями TRS-IO-E, а в системе существует только одно расширение).

Способ устранения:

Проверить, что устройства, описанные в аппаратной конфигурации, соответствуют имеющимся.

3337 Плата EtherCAT: узел НомерУзел отключен

Причина:

Узел EtherCAT не вернул никакого ответа на элемент управления. Узел может быть отключен от сети или может быть выключен.

Способ устранения:

Проверьте проводку и убедитесь, что на узел подается питание.

3338 Плата EtherCAT: узел НомерУзел снова подключен

Причина:

Указанный узел EtherCAT снова подключен к сети и начал отвечать на запросы управления.

3340 Плата EtherCAT: узел НомерУзел не ответил на запрос (Код)

Причина:

Узел EtherCAT, допрошен через службу SDO, не ответил на запрос. Кроме этого, если устройство предоставило код прерывания (abort code), этот сообщается в показанном сообщении об ошибке (код). Эта ошибка может появиться во время выполнения инструкций [SETPZERO](#), [SETPFLY](#) и [FASTREAD](#) на осях EtherCAT, поскольку они используют связь SDO для настройки привода. В этом конкретном случае, сообщение об ошибке, кроме стандартных кодов прерывания, может содержать следующие коды:

- 1: внутренний Таймаут истек: узел не ответил в течение 250 мс.
- 00000005: внутренний индекс неправильного почтового ящика.
- 00000006: данные, полученные не в соответствии с запросом.

Способ устранения:

Обратиться к производителю.

3341 Плата EtherCAT: узел НомерУзел не существует

Причина:

Была предпринята попытка выполнить команду на несуществующем узле.

Способ устранения:

Проверьте номер узла в инструкции GPL, которая вызвала ошибку. Подключите узел.

3342 Кабель отключен

Причина:

Кабель EtherCAT не подключен к блоку управления.

Способ устранения:

Убедиться, что кабель правильно подключен к блоку управления и что он не поврежден.

3343 Плата EtherCAT номер: узел НомерУзел не переходит в состояние SAFE-OPERATIONAL (Код)

Причина:

Указанный узел EtherCAT не переходит в состояние SAFE-OPERATIONAL.

Способ устранения:

Обратиться к производителю и сообщить номер указанной ошибки (код), который представляет код ALstatuscode.

3344 Плата EtherCAT номер: узел НомерУзел не переходит в состояние OPERATIONAL (Код)

Причина:

Указанный узел EtherCAT не переходит в состояние OPERATIONAL.

Способ устранения:

Обратиться к производителю и сообщить указанный номер ошибки (код), который представляет код ALstatuscode.

3345 Плата EtherCAT: нестабильная связь

Причина:

Связь в сети EtherCAT нестабильна из-за неисправностей или неисправных кабелей или узлов.

4400 Слишком много активных осей в FASTREAD (Функция:ИмяФункция линия:НомерЛиния)

Причина:

Пыгалось использовать инструкцию FASTREAD с большим количеством осей, чем разрешено. Следует иметь в виду, что оси, оснащенные дополнительным энкодером, эквивалентны двойной оси (см. Инструкцию [SWITCHENC](#)).

Способ устранения:

Уменьшить количество осей использованных с дополнительным энкодером.

6.2.6 Ошибки, генерированные инициализацией

769: Неправильная конфигурация программного обеспечения

Причина:

Конфигурация аппаратного обеспечения удаленного модуля не соответствует указанной конфигурации программного обеспечения на конфигурации системы.

Способ устранения:

Проверить соответствие параметров аппаратного обеспечения удаленного модуля и параметров программного обеспечения.

770 Номер IRQ сконфигурирован неправильно

Причина:

Был неправильно задан IRQ платы осей в конфигурации модуля. Обычно существует аппаратный конфликт с другой периферией системы.

Способ устранения:

В настройках BIOS материнской платы проверить, что использованный платой осей IRQ зарезервирован "Legacy ISA". Проверить, что отсутствует другая периферия, использующая тот же IRQ, заданный плате осей. По возможности изменить IRQ периферии, конфликтующей с платой осей, в противном случае изменить IRQ платы осей.

772 Ошибка при считывании зоны буферной памяти при инициализации

Причина:

Произошла ошибка при тестах, выполнявшихся на этапе инициализации платы осей. В частности, не пройден тест буферной ОЗУ (Dallas). Может обуславливаться неправильной конфигурацией адресов входов/выходов и IRQ платы или конфликтом с другой периферией системы. Наконец, это может быть следствием повреждения платы осей.

Способ устранения:

Проверить конфигурацию аппаратного обеспечения. Квалифицированные специалисты могут выполнить тест аппаратного обеспечения ОЗУ микроконтроллера i296. Обратите внимание, что тест аппаратного обеспечения буферной ОЗУ приводит к удалению всех записанных в ней данных. В буферной ОЗУ хранятся значения некоторых устройств, например, счетчиков, таймеров и сдвиги DAC осей. Перед выполнением теста следует сохранить эти значения.

Если проблема возникает снова, обращаться к производителю.

773 Достигнуто максимальное число конфигурируемых осей

Причина:

Была сделана попытка сконфигурировать число осей больше максимально допустимого числа.

Способ устранения:

Уменьшить число конфигурируемых осей. Обратиться к изготовителю станка за дополнительной информацией.

774 Не начала выполнение задача реального времени осей

Причина:

Встроенное ПО управления осями было инициализировано, но функционирует неправильно. Обычно существует аппаратный конфликт с другой периферией системы.

Способ устранения:

Проверить отсутствие конфликтов с другой периферией, изменить конфигурацию периферий, вызывающих конфликт, или удалить ее из системы.

775 Недостаточно времени для выполнения GPL

Причина:

Выполнение задачи реального времени занимает слишком много времени цикла. Генерируется в случае, когда задача реального времени не завершается до начала следующей задачи реального времени осей (например, был создан бесконечный цикл).

Способ устранения:

Изменить код GPL так, чтобы уменьшить продолжительность задачи реального времени.

776 Слишком большое время выполнения задачи реального времени

Причина:

Выполнение задачи реального времени занимает слишком много времени цикла. Продолжительность выполнения оказывается чуть больше максимального разрешенного предела.

Способ устранения:

Изменить код GPL так, чтобы уменьшить продолжительность задачи реального времени.

777 Истек период работы watchdog

Причина:

Заблокировано встроенное ПО.

Способ устранения:

Обратиться к изготовителю станка.

778 Заблокирован код главной функции встроенного ПО

Причина:

Встроенное ПО заблокировано в течение более 5 задач реального времени.

Способ устранения:

Обратиться к изготовителю станка.

1025 Плата НомерПлаты: не отвечает на команды

Причина:

При инициализации была обнаружена карта осей, но она неправильно реагирует на команды.

Способ устранения:

Квалифицированные специалисты могут выполнить тест аппаратного обеспечения карты осей. Если проблема возникает снова, обращаться к производителю.

1026 Плата НомерПлаты: ошибка при отправке встроенного ПО на плату осей

Причина:

Оказалось невозможным отправить встроенное ПО на плату.

Способ устранения:

Обратиться к производителю.

1028 Плата НомерПлаты: встроенное ПО не присутствует

Причина:

Встроенные ПО, имеющиеся на плате, неправильны для типа обнаруженной платы.

Способ устранения:

Направить правильное встроенное ПО.

1029 Плата НомерПлаты: заблокирована главная функция

Причина:

Встроенное ПО платы не начало выполнение.

Способ устранения:

Обратиться к производителю.

1031 Плата НомерПлаты: ошибка при инициализации

Причина:

Произошла ошибка при процедурах инициализации платы осей.

Способ устранения:

Проверить и устранить причины ошибок системы, произошедших в моменты, предшествующие возникновению текущей ошибки. Затем инициализировать систему.

1032 Плата НомерПлаты: не пройден тест двухпортовой памяти**Причина:**

Произошла ошибка при тестах, выполнявшихся на этапе инициализации платы осей. В частности, не удалась инициализация двухпортовой памяти. Как правило, это обуславливается конфликтом аппаратного обеспечения с другой периферией системы, но может обуславливаться также повреждением платы.

Способ устранения:

Проверить конфигурацию платы, проверить отсутствие конфликтов с другой периферией. При использовании в удаленном режиме снова передать встроенное ПО на модуль. Квалифицированные специалисты могут выполнить тест аппаратного обеспечения двухпортовой памяти. Если проблема возникает снова, обращаться к производителю.

1033 Плата НомерПлаты: код начальной загрузки встроенного ПО не выполняется**Причина:**

Встроенное ПО пуска платы было инициализировано, но функционирует неправильно. Обычно существует аппаратный конфликт с другой периферией системы.

Способ устранения:

Проверить конфигурацию платы, проверить отсутствие конфликтов с другой периферией. При использовании в удаленном режиме снова передать встроенное ПО на модуль. Квалифицированные специалисты могут выполнить тест аппаратного обеспечения двухпортовой памяти на плате. Если проблема возникает снова, обращаться к производителю.

1035 Плата НомерПлаты: не присутствует**Причина:**

Произошла ошибка при тестах, выполнявшихся на этапе инициализации платы осей. В частности, не была обнаружена плата.

Способ устранения:

Проверить, что плата присутствует в системе и что она не повреждена. Квалифицированные специалисты могут выполнить тест аппаратного обеспечения платы. Если проблема возникает снова, обращаться к производителю.

1037 Плата НомерПлаты: не удалось открытие двухпортовой памяти**Причина:**

Не удалось открытие двухпортовой памяти платы.

Способ устранения:

Квалифицированные специалисты могут выполнить тест аппаратного обеспечения платы. Обратиться к производителю.

1039 Плата НомерПлаты: истек период работы сторожевой схемы**Причина:**

Заблокировано встроенное ПО платы осей НомерПлаты.

Способ устранения:

Обратиться к изготовителю станка.

1040 Плата НомерПлаты: ошибка питания +24 В пост. тока**Причина:**

Полевое питание (+24 В пост. тока) выходов не присутствует или неправильно работает.

Способ устранения:

Проверить работу полевого питания +24 В пост. тока.

1047 Плата НомерПлаты: неразрешенная конфигурация программного обеспечения

Причина:

Устройство получило конфигурацию, несовместимую с имеющимся или подключенным аппаратным обеспечением. Например, запрашивается конфигурация неподключенной оси или оси, не присутствующей на устройстве.

Способ устранения:

Проверить соответствие параметров аппаратного обеспечения платы и параметров программного обеспечения.

1052 Плата НомерПлаты: выполняется код начальной загрузки

Причина:

Плата задана на временный режим и выполняет код начальной загрузки.

Способ устранения:

Обратиться к производителю.

1053 Плата НомерПлаты: Истек период работы watchdog осей

Причина:

Произошла серьезная ошибка при выполнении встроенного ПО платы управления осями. Оси отключаются и возможный сигнал SYSOK отключается. Не сбрасывать систему.

Способ устранения:

Обратиться к изготовителю станка.

1055 Истек период работы watchdog на Плате НомерПлаты

Причина:

Заблокировано встроенное ПО платы НомерПлаты.

Способ устранения:

Обратиться к изготовителю станка.

1056 Плата НомерПлаты: ошибка питания интерфейса CAN

Причина:

Устройство передачи на линии CanBus, присутствующее на указанной плате, имеет ошибку питания. Это может обуславливаться наличием КЗ, ошибкой электромонтажа шины или повреждением платы.

Способ устранения:

Проверить электромонтаж всей линии CAN. Проверить подключение линии к ЧПУ. Устранить возможное короткое замыкание. Если после этого связь не восстанавливается, обратиться к изготовителю.

1057 Плата НомерПлаты: внутренняя ошибка номер НомерОшибки

Причина:

Ошибка аппаратного обеспечения узла.

Способ устранения:

Обратиться к изготовителю.

6.2.7 Ошибки, генерированные управлением памяти

1281 Ошибка распределения памяти в зоне хипа

Причина:

Доступная ОЗУ меньше требуемой, например, глобальной матрицей.

Способ устранения:

Уменьшить размер глобальных переменных, помещенных в ОЗУ.

1286 Ошибка при реализации хипа

Причина:

Ошибка управления памятью встроенным ПО.

Способ устранения:

Обратиться к изготовителю станка.

1287 Слишком много отнятий памяти от хипа

Причина:

Ошибка управления памятью встроенным ПО.

Способ устранения:

Обратиться к изготовителю станка.

1289 Ошибка при создании глобальных переменных

Причина:

Было определено слишком большое число [глобальных переменных](#) или были определены глобальные матрицы слишком больших размеров.

Способ устранения:

Уменьшить число глобальных переменных или размер матриц.

1290 Ошибка размера энергонезависимых переменных

Причина:

Было определено слишком большое число энергонезависимых переменных или были определены энергонезависимые матрицы слишком больших размеров.

Способ устранения:

Уменьшить число энергонезависимых переменных или размер энергонезависимых матриц.

1291 Ошибка размера переменных только для считывания

Причина:

Было определено слишком большое число переменных только для считывания или были определены матрицы только для считывания слишком больших размеров.

Способ устранения:

Уменьшить число переменных только для считывания или размеры матриц только для считывания.

6.2.8 Ошибки, генерированные отказами

1559 Следы точки останова

Причина:

Серьезная ошибка встроенного ПО.

Способ устранения:

Обратиться к изготовителю станка.

1569 Недействительный рабочий код микропроцессора

Причина:

Микропроцессор встретил неизвестную инструкцию. Может обуславливаться как проблемами аппаратного обеспечения ПК, так и повреждением файлов, содержащихся во встроенном ПО Albatros.

Способ устранения:

В случае локального модуля проверить, что файлы не повреждены, и при необходимости выполнить повторную установку Albatros. Для удаленных модулей выполнить обновление встроенного ПО.

Выполнить тест аппаратного обеспечения ПК, в частности, ОЗУ. Если проблема не устраняется, обратиться к изготовителю станка.

1586 Значение INTEGER делится на ноль

Причина:

Была сделана попытка разделить INTEGER (Целое число) на ноль.

Способ устранения:

Проверить в функциях GPL, что все деления правильны.

1600 Переполнение в результате операции с плавающей запятой

Причина:

Результат операции FLOAT больше емкости адресата:

$\pm 3,402823E+38$ для float
 $\pm 1,79769313486231E+308$ для double

Способ устранения:

Проверить в функциях GPL правильность расчетов с плавающей запятой.

1601 Незагруженность в результате операции с плавающей запятой

Причина:

Результат операции FLOAT меньше емкости адресата:

$\pm 1,401298E-45$ для float
 $\pm 4,94065645841247E-324$ для double

Способ устранения:

Проверить в функциях GPL правильность расчетов с плавающей запятой.

1602 Недействительный аргумент для операции с плавающей запятой

Причина:

В операции float был использован операнд не типа float.

Способ устранения:

Проверить в функциях GPL правильность расчетов с плавающей запятой (float).

1603 Значение с плавающей запятой, деленное на ноль

Причина:

Была сделана попытка разделить float или double на ноль. Генерируется также при выполнении логарифма нуля.

Способ устранения:

Проверить в функциях GPL, что все деления правильны.

1604 Неточный результат операции с плавающей запятой

Причина:

Неправильный результат операции с плавающей запятой.

Способ устранения:

Проверить в функциях GPL правильность расчетов с плавающей запятой.

1605 Было использовано неправильное значение с плавающей запятой

Причина:

Было использовано значение с плавающей запятой меньше представляемого минимума:

$\pm 1,401298E-45$ для float

$\pm 4,94065645841247E-324$ для double

Способ устранения:

Проверить в функциях GPL правильность расчетов с плавающей запятой.

1728 Была сделана попытка доступа к недействительному адресу

Причина:

Программа вошла в недействительную зону памяти.

Способ устранения:

Проверить соответствие глобальных/локальных переменных и если проблема не устраняется, сообщить о неисправности.

1735 Общее исключение

Причина:

Возникло нераспознанное исключение.

Способ устранения:

Обратиться к изготовителю станка.

1736 Невыровненные данные

Причина:

Серьезная ошибка встроенного ПО.

Способ устранения:

Обратиться к изготовителю станка.

1801 Температурный сигнал тревоги

Причина:

Температура ЦП управления превысила максимально допустимые пределы.

Способ устранения:

Проверить отсутствие проблем вентиляции или причин перегрева. Если проблема возникает снова, обращаться к производителю.

1802 Сигнал тревоги вентилятора

Причина:

Плохо работает вентилятор ЦП управления. Эта проблема за короткое время может вызвать перегрев ЦП.

Способ устранения:

Обратиться к изготовителю станка.

1803 Частота ЦП нестабильна

Причина:

Рабочая частота ЦП нестабильна.

Способ устранения:

Обратиться к изготовителю станка.

6.2.9 Ошибки, генерированные инструкциями GPL

4097 Устройство ТипУстройства ИмяУстройства не сконфигурировано

Причина:

Инструкция GPL использовала не сконфигурированное устройство, т. е. устройство, не имеющее подключения виртуальное-физическое. Может генерироваться всеми инструкциями, которым в качестве параметра передается устройство.

Способ устранения:

В конфигурациях блока управления проверить, что все используемые функцией устройства имеют подключение виртуальное-физическое. Затем снова передать конфигурацию на плату.

4098 Глобальная переменная ИмяПеременной не существует

Причина:

Инструкции GPL была передана в качестве аргумента глобальная переменная, которая не была определена. Как правило, это происходит тогда, когда блок управления не был правильно инициализирован.

Способ устранения:

Снова выполнить компиляцию всего кода GPL и снова инициализировать блок управления.

4099 Функция ИмяФункции не найдена

Причина:

Был сделан вызов отсутствующей функции. Может возникнуть, когда, после изменения кода GPL не была выполнена инициализация блока управления.

Способ устранения:

Снова выполнить компиляцию всего кода GPL и снова инициализировать блок управления.

4101 Непоследовательное управление осью ИмяОси

Причина:

Был выполнен неразрешенный переход состояния на оси. По переходам состояния см. соответствующую документацию.

Может генерироваться всеми инструкциями, управляющими осями, но обычно происходит в следующих случаях:

- при попытке интерполяции, координации с осью, уже занятой в движении точка-точка (или наоборот).
- при выполнении инструкции Chain, SetPFly, SetPZero на оси, которая находится в прозрачном режиме.
- при попытке интерполяции, координации на оси, которая является ведомой другой оси.

Способ устранения:

Проверить, что все движения осей завершены инструкцией ожидания на координате, особенно если оси чередуют движения различного типа (точка-точка, интерполяция и т.д.).

4105 Инструкция не может быть выполнена на оси ИмяОси

Причина:

Была сделана попытка выполнения инструкции на не поддерживаемой ее оси. Например, инструкция интерполяции на шаговой оси.

Способ устранения:

Скорректировать код GPL.

4106 Не подключен удаленный модуль, относящийся к шаговой оси ИмяОси

Причина:

Была сделана попытка воздействия на шаговую ось, не подключенную к блоку управления.

Способ устранения:

Проверить подключение удаленного модуля, управляющего этой осью.

4107 Инструкция SYSOK с неправильными аргументами

Причина:

Была выполнена инструкция SYSOK с неправильными аргументами. Происходит, если один или несколько цифровых выходов, переданных инструкции как аргументы, сконфигурированы неправильно.

Способ устранения:

Проверить код GPL и конфигурацию виртуальное-физическое.

4108 Ось ИмяОси: окончательная квота за пределы программного обеспечения

Причина:

Была сделана попытка перемещения оси за пределы, заданные в конфигурации, или кодами GPL.

Способ устранения:

Скорректировать программу обработки, вызывавшей эту ошибку. При необходимости скорректировать код GPL или конфигурацию оси.

4110 Неправильная скорость

Причина:

Была сделана попытка задать оси нулевую или отрицательную скорость.

Способ устранения:

Скорректировать код GPL.

4111 Отрицательное ускорение оси ИмяОси

Причина:

Была сделана попытка задать оси отрицательное ускорение.

Способ устранения:

Скорректировать код GPL.

4112 Отрицательное замедление оси ИмяОси

Причина:

Была сделана попытка задать оси отрицательное замедление.

Способ устранения:

Скорректировать код GPL.

4114 Ось ИмяОси: не выполнено обнуление на быстром входе

Причина:

Обнуление координаты на быстром входе (обнуление налету) было завершено неправильно. Эта процедура позволяет обнулять координату движущейся оси в момент изменения состояния соответствующего быстрого входа. Если ось завершает текущее движение без переключения входа, генерируется ошибка системы. Причиной может быть неправильное задание параметров перемещения оси или проблема электромонтажа быстрого входа.

Способ устранения:

Проверить код GPL, реализующий обнуление налету, проверить электромонтаж быстрого входа.

4115 Ось ИмяОси: не найдена отметка нуля

Причина:

Обнуление координаты на отметке нуля кодера было завершено неправильно. Эта процедура позволяет обнулять координату движущейся оси в момент обнаружения отметки нуля кодера. Если ось достигает координаты поиска отметки и не обнаруживает ее, генерируется ошибка системы. Причиной может быть неправильное задание параметров перемещения оси или проблема электромонтажа сигнала отметки (фаза С разъема оси).

Способ устранения:

Проверить код GPL, реализующий обнуление на отметке, проверить электромонтаж оси.

4353 Рабочий код инструкции неизвестен (Функция:ИмяФункции строка: НомерСтроки)

Причина:

При выполнении функции GPL была выявлена неразрешенная инструкция. Как правило, это является признаком повреждения файлов, содержащих компилированный код GPL. Если программное обеспечение управления было обновлено, проверить, что вы перекомпилировали код GPL. Предыдущая версия ПО может содержать инструкции, которые больше не поддерживаются новой.

Способ устранения:

Снова выполнить компиляцию всего кода GPL и инициализировать блок управления. Если проблема не устраняется, обратиться к изготовителю станка.

4354 Неправильная математическая операция (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Инструкция GPL сделала попытку выполнения неправильной математической операции, например, деления на ноль. Или же некоторые данные в инструкции GPL непоследовательны. Обычно эта ошибка генерируется инструкциями перемещения с интерполяцией, т. к. является частью встроенного ПО, выполняющего несколько математических расчетов.

Способ устранения:

Проверить, что правильны все параметры, переданные инструкциям интерполяции. Если проблема не устраняется, сообщить о неисправности изготовителю.

4355 Неправильный адрес матрицы или вектора (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Инструкция GPL сделала попытку доступа к элементу массива или матрицы, выходящему за максимальный размер. Например, она сделала попытку доступа к элементу 10 массива из 5 элементов. Может генерироваться всеми инструкциями, принимающими в качестве параметра массив или матрицу.

Способ устранения:

Проверить, что все переданные инструкциям индексы матрицы и массива находятся в пределах размеров массива и матрицы.

4356 Инструкция RET не вызвана CALL (Функция: ИмяФункции строка: НомерСтроки)

Причина:

Была выполнена инструкция RET без наличия в стеке соответствующего адреса возврата. Наиболее частой причиной является объявление подпроцедуры до инструкции FRET выхода из функции, предварительно не защитив ее при помощи GOTO, не допускающей ее случайного исполнения. Или же был выполнен нежелательный переход внутри подпроцедуры.

Способ устранения:

Проверить ход программы GPL. По возможности, позиционировать подпроцедуры в конце тела функций (после инструкции FRET).

4357 Несуществующая локальная переменная (Функция:ИмяФункции строка:НомерСтроки)

Причина:

Инструкция GPL сделала попытку доступа к локальной переменной, которая не была назначена.

Способ устранения:

Снова выполнить компиляцию и снова передать все функции на плату. Если проблема не устраняется, сообщить о неисправности.

4358 Несуществующий ярлык перехода (Функция:ИмяФункции строка:НомерСтроки)

Причина:

Инструкция GPL выполнила переход к несуществующему ярлыку перехода. Может генерироваться GOTO, CALL, FCALL, всеми IF.

Способ устранения:

Снова выполнить компиляцию и снова передать все функции на плату. Если проблема не устраняется, сообщить о неисправности.

4359 Неправильный аргумент макропрограммы (Функция:ИмяФункции строка:НомерСтроки)

Причина:

Инструкции GPL были переданы недействительные аргументы. Может генерироваться всеми инструкциями. Однако в большинстве случаев система GPL пытается автоматически скорректировать эту ситуацию, выполняя автоматические преобразования типа (приведение), которые, однако вызывают потерю времени. Эта ошибка генерируется, когда эти преобразования невозможны, большей частью в следующих случаях:

- инструкции, действующие на определенные устройства (SETTIMER, SETCOUNTER), которым передается инструмент другого типа.
- инструкции, действующие на бит, которым передается число с плавающей запятой (AND, OR и т.д.)
- инструкции, действующие на матрицы или массивы, которым передается простая переменная (SORT, MOVEMAT и т.д.)
- инструкции, действующие на строки, которым не передаются строки.

Ошибка генерируется также когда делается попытка выполнить команду на плате, на которой эта команда не реализована. (Например, команду [SENDPDO](#) или команду [RECEIVEPDO](#) на плате, отличной от TMSCan или TMSCan+).

Способ устранения:

Скорректировать код GPL.

4360 Ошибка распределения памяти при выполнении (Функция:ИмяФункции строка:НомерСтроки)

Причина:

Функция GPL попыталась выделить зону памяти для внутреннего пользования, но не нашла свободной памяти.

Этот сигнал может указывать физиологическую ситуацию, обуславливаемую, например, исполнением слишком большого числа задач одновременно или определением слишком больших размеров глобальных переменных.

Способ устранения:

Проверить размеры глобальных и локальных переменных и попытаться уменьшить их. Проверить, что одновременно не выполняется слишком много задач, и при необходимости уменьшить их количество.

4361 Слишком много активных задач (Функция:ИмяФункции строка:НомерСтроки)

Причина:

Была сделана попытка поставить на выполнение одновременно более 256 задач.

Способ устранения:

Уменьшить число активных одновременно задач.

4362 Неправильный формат матрицы (Функция:ИмяФункции строка:НомерСтроки)**Причина:**

Инструкция, действующая на матрицы, выявила недействительный формат. Эту ошибку системы могут генерировать следующие инструкции:

- MOVEMAT, если не соответствует формат исходной и целевой матрицы.
- CLEAR при попытке обнуления несуществующей строки матрицы.
- GETAXIS, если формат матрицы, переданной для параметра, не соответствует формату, который ожидает инструкция (см. документацию на язык GPL).

Способ устранения:

Проверить перечисленные инструкции внутри генерировавшей ошибку задачи. В частности, проверить, что матрицы, переданные на MOVEMAT, имеют то же число столбцов того же типа, и что матрица, переданная на GETAXIS, имеет соответствующий формат.

4363 Слишком много активных инструкций ONINPUT (Функция:ИмяФункции строка:НомерСтроки)**Причина:**

Было активировано более 128 инструкций OnInput.

Способ устранения:

Уменьшить число инструкций ONINPUT.

4364 Ось уже занята в локальной системе координат (Функция:ИмяФункции строка:НомерСтроки)**Причина:**

Эта ошибка связана с подключением трех осей с переносом и поворотом для выполнения интерполяций на нескольких осях картезианской системы координат.

Была сделана попытка выполнения SETRIFLOC, передавая инструкции ось, которая уже занята в трех осях системы координат. Генерируется также в случае, если выполняется RESRIFLOC на оси, которая не была занята ни в одной системе из трех координат. Последний случай - когда больше не существует доступных систем из трех координат (максимум их может быть 32).

Способ устранения:

Проверить, что три координаты, переданные при помощи SETRIFLOC, не имеют общих осей.

Проверить RESRIFLOC.

Проверить также, что имеются инструкции ожидания на координате до RESRIFLOC.

В любом случае, помните, что пока интерполяция не завершена, RESRIFLOC фактически не выполняется.

4365 Инструкция ONINPUT активирована на том же входе (Функция:ИмяФункции строка: НомерСтроки)**Причина:**

Инструкции ONINPUT один и тот же вход был передан несколько раз.

Способ устранения:

Проверить, что нет двух ONINPUT, которым как параметр передается один и тот же вход.

4366 Слишком много активных инструкций ONFLAG (Функция:ИмяФункции строка: НомерСтроки)**Причина:**

Было активировано более 128 инструкций OnFlag.

Способ устранения:

Уменьшить число инструкций ONFLAG.

4367 Инструкция ONFLAG активирована на том же флажке (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Инструкции ONFLAG один и тот же флажок был передан несколько раз.

Способ устранения:

Проверить, что нет двух ONFLAG, которым как параметр передается один и тот же флажок.

4368 Попытка записи переменной типа только для считывания (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Была сделана попытка доступа к записи переменной только для считывания. Переменные типа только для считывания всегда глобальные и находятся во флеш-памяти блока управления. В редакторе глобальных переменных они указываются как "статические". При попытке записи на одной из этих глобальных переменных генерируется эта ошибка системы. Кроме того, ошибка генерируется также в случае использования переменных, находящихся в буферной (энергонезависимой) ОЗУ, в качестве аргумента некоторых инструкций при записи. Например на инструкции COORDIN переменная, переданная для индикации обрабатываемой строки, должна находиться в ОЗУ.

Способ устранения:

Проверить все статические и энергонезависимые переменные.

4369 Слишком много активных главных осей (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Была сделана попытка подключить в качестве главной более 4 осей одновременно. Эта ошибка генерируется только при выполнении инструкции CHAIN.

Способ устранения:

Уменьшить число главных осей.

4370 Слишком много активных ведомых осей (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Была сделана попытка подключить более 8 осей в качестве ведомых одной главной оси. Эта ошибка генерируется только при выполнении инструкции CHAIN.

Способ устранения:

Уменьшить число ведомых осей.

4372 Неправильное использование инструкции (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Эта ошибка генерируется на одной из следующих ситуаций:

1. Используется инструкция для управления почтовым ящиком (sendmail, waitmail, endmail, ifmail) или инструкция для управления IPC (sendipc, testipc, waitipc) внутри функции, вызванной инструкцией Errsys, Oninput, или Onflag.
2. Используется инструкция IfError или инструкция IfMessage без активации управления предупреждениями состояния.
3. Используется инструкция Сторожевой схемы без наличия платы TMSWD.
4. Параметры заданные в инструкции интерполяции (linearinc, linearabs, circle, circinc, circabs, helinc и helcabs) не последовательны. Например, число заявленных осей отличается от того заявленных координат или были заявлены более осей тех, которыми инструкция может управлять.

Способ устранения:

Ниже описываются способы устранения для каждой указанной причины:

1. Переместить инструкцию, вызывающую ошибку на другую функцию или удалить инструкцию.
2. Проверить, что управление предупреждениями состояния подключено. В файле tra.ini на разделе [ALBATROS] пункту AlarmsHaveState надо задать значение 1.
3. Удалить инструкцию Сторожевой схемы или приобрести плату TMSWD.
4. Проверить, что параметры инструкции GPL правильны. Каждой оси одна координата должна соответствовать. Число осей заявленных в инструкции не должно быть выше числа осей, которыми функция может управлять. Например, инструкция LINEARABS может управлять до 6 осей. Если, среди параметров, заявлены более 6 параметров, инструкция генерирует ошибку системы.

4373 Невозможно выполнить считывание скорости подачи (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Инструкция [GETFEED](#) была использована на плате TmsBus или TmsCap, которая не является главной.

Способ устранения:

Проверить в конфигурации аппаратного обеспечения, что плата, к которой подсоединена скорость подачи, является главной.

4374 Выполняется слишком много инструкций типа IPC (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Был превышен максимальный предел в 16 одновременно выполняемых инструкций IPC.

Способ устранения:

Изменить код GPL.

4375 FASTREAD выполнена на осях различных плат (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Была сделана попытка выполнения инструкции FASTREAD, передавая ей как параметры оси, которые не все подключены к одной плате.

Способ устранения:

Соответствующим образом изменить код GPL или конфигурацию виртуальное-физическое.

4378 Неподключенная инструкция (Функция:ИмяФункции строка:НомерСтроки)

Причина:

Была сделана попытка использования инструкции, выполнение которой не подключено. Возможно, неправильно вставлен или отсутствует аппаратный ключ.

Способ устранения:

Правильно вставить аппаратный ключ. Если проблема не устраняется, обратиться к производителю.

4379 Инструкция не может использоваться в функциях, запущенных прерыванием (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Была сделана попытка использования инструкции, недопустимой в функции, запущенной прерыванием. Функции, запущенные прерыванием, - это функции, передаваемые как параметр инструкциям ONERRSYS, ONINPUT и ONFLAG.

Способ устранения:

Изменить код GPL. См. [список инструкций, которые не могут использоваться на прерывании](#)

4380 Слишком много запросов записи в зоне буферной памяти (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Была сделана попытка выполнения одновременно слишком большого числа операций записи в буферную память (буферная память характеризуется относительно большим временем доступа).

Способ устранения:

Проверить инструкции, выполняющие операции записи на переменных, расположенных в буферной памяти: счетчики, таймеры, матрицы и переменные, заявленные как "nonvolatile".

4381 Невозможно использовать еще неоткрытую последовательную линию (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Была сделана попытка выполнения инструкции, действующей на последовательный порт, до предварительного выполнения для этого порта инструкции COMOPEN.

Способ устранения:

Изменить код GPL.

4382 Невозможно открыть уже открытую последовательную линию (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Была сделана попытка выполнения инструкции COMOPEN на последовательном порте, уже открытом при помощи этой инструкции.

Способ устранения:

Изменить код GPL.

4383 Была сделана попытка открыть слишком много вспомогательных процессов (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Была сделана попытка открыть более 4 вспомогательных процессов одновременно.

Способ устранения:

Изменить код GPL.

4384 Вспомогательный процесс не выполняется (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Была сделана попытка доступа к невыполняемому вспомогательному процессу.

Способ устранения:

Изменить код GPL.

4385 Была сделана попытка открыть вспомогательный процесс из другой задачи (Функция:ИмяФункции строка: НомерСтроки)

Причина:

Была сделана попытка открыть вспомогательный процесс из задачи, отличной от той, которая запустила выполнение. Вспомогательный процесс может использоваться только задачей, запустившей его выполнение.

Способ устранения:

Изменить код GPL.

4391 Ошибка при активации SYSOK (Функция:ИмяФункции строка:НомерСтроки)**Причина:**

Не удалось подключение сигнала SYSOK. Как правило, может обуславливаться неисправностью датчика Greenbus на плате осей.

Способ устранения:

Квалифицированные специалисты могут выполнить тест аппаратного обеспечения двухпортовой памяти микроконтроллера i296. Если проблема возникает снова, обращаться к производителю.

4394 Слишком много ошибок цикла (Функция:ИмяФункции строка:НомерСтроки)**Причина:**

Активно более 2000 ошибок цикла.

Способ устранения:

Скорректировать код GPL, ограничивая число сигналов.

4395 Слишком много сообщений (Функция:ИмяФункции строка:НомерСтроки)**Причина:**

Активно более 2000 сообщений.

Способ устранения:

Скорректировать код GPL, ограничивая число сигналов

4397 Переполнение стека на функции ИмяФункции (Функция:ИмяФункции строка:НомерСтроки)**Причина:**

Стек функции GPL превысил максимальный предел в 2 кбайт.

Способ устранения:

Снова скомпилировать код GPL и проверить в отчете компилятора предполагаемый занимаемый объем стека функцией, которая вызвала ошибку системы. Затем уменьшить число локальных переменных и параметров передаваемых функциям (заменяя их, например, глобальными переменными). Уменьшить число CALL.

4398 Незагруженность стека на функции ИмяФункции (Функция:ИмяФункции строка:НомерСтроки)**Причина:**

Может подаваться только при серьезной ошибке встроенного ПО, например, при неправильном управлении параметрами функции или локальными переменными.

Способ устранения:

Обратиться к изготовителю станка.

4399 Параметр за пределами диапазона (Функция:ИмяФункции строка:НомерСтроки)**Причина:**

Переменной GPL или устройству было задано значение, не входящее в разрешенный диапазон.

Способ устранения:

Исправить и снова скомпилировать код GPL.

4865 Отсутствует определение станка для интерполяции (G216 или G217)**Причина:**

Была сделана попытка перемещения осей с интерполяцией ISO или были заданы индексы конфигурации без предварительного определения матриц конфигурации и образующих станок осей.

Способ устранения:

Исправить и снова скомпилировать код GPL, используя инструкцию [ISOG216](#).

4866 Отсутствует описание индексов конфигурации станка (M6)**Причина:**

Была сделана попытка перемещения осей с интерполяцией ISO без предварительного определения индексов матриц конфигурации станка

Способ устранения:

Исправить и снова скомпилировать код GPL, используя инструкцию [ISOM6](#).

6.2.10 Ошибки, генерированные драйвером связи CNCTPA**16385 Модуль отключен****Причина:**

Была прервана связь между ПК-супервизором и модулем.

Возможные причины:

- отсутствие питания удаленного модуля
- прерывание, даже временное, подключения кабелей ethernet в результате ложного контакта в разъемах или повреждения кабелей
- отсутствие питания или неправильная работа концентратора ethernet (при наличии)
- блокировка встроенного ПО удаленного модуля в результате повреждения файлов конфигурации
- сброс ЦП удаленного модуля в результате перегрева или ЭМ помех

Способ устранения:

Проверить, что модуль включен. Проверить кабели и разъемы ethernet. Обновить встроенное ПО на удаленном модуле. Проверить, что на нем не происходит перегрева в результате недостаточной вентиляции и что на него не действуют электромагнитные помехи. Если проблема не устраняется, обратиться к изготовителю станка.

16386 Модуль подключен**Причина:**

Удаленный модуль подключился к ПК-супервизору после этапа инициализации Albatros. При запуске Albatros стремится подключить все модули, предусмотренные конфигурацией системы. Этот этап длится ок. 4 секунд. Модули, подключающиеся позднее, вызывают ошибку системы.

16387 Модуль снова подключен**Причина:**

Удаленный модуль снова подключился к ПК-супервизору после отключения. Таким образом, эта ошибка всегда следует за ошибкой 16385 "Модуль отключен".

16388 Модуль инициализирован**Причина:**

При нормальном функционировании был повторно инициализирован удаленный модуль. Это подразумевает, что модуль отключился и снова подключился к ПК-супервизору. Таким образом, эта ошибка всегда следует за ошибкой 16385 "Модуль отключен".

Эта ошибка указывает на выполнение сброса модуля в результате, например, отсутствия электропитания.

16389 Модуль прервал связь**Причина:**

Удаленный модуль прервал связь с Albatros. Это происходит, когда модуль не получает команды или опросы от ПК-супервизора в течение длительного времени. Таким образом, эта ошибка указывает на проблему (сильное замедление или блокировку) на ПК-супервизоре.

Способ устранения:

Проверить, что на ПК-супервизоре нет программ, вызывающих блокировку или замедление системы. Отключить экранную заставку на ПК-супервизоре. Если проблема не устраняется, обратиться к производителю станка.

16641 Встроенное ПО блока управления не реагирует на команды

Причина:

Произошла ошибка при инициализации системы. В частности, неправильно отвечает встроенное ПО. Проблема может обуславливаться повреждением файлов встроенного ПО.

Способ устранения:

Попробовать сбросить систему, а при необходимости - также переинсталлировать Albatros. Если проблема не устраняется, обратиться к производителю.

16642 ТраSock не реагирует на команды

Причина:

Произошла ошибка при инициализации системы. В частности, программное обеспечение связи с модулями отвечает неправильно. Проблема может обуславливаться повреждением файлов.

Способ устранения:

Попробовать сбросить систему, а при необходимости - также переинсталлировать Albatros. Если проблема не устраняется, обратиться к изготовителю станка.

16643 Операционная система не позволяет использовать RTX

Причина:

Инсталлированная на ПК операционная система не допускает использование RTX и, соответственно, не допускает правильную работу версий Albatros, предусматривающих ее наличие.

Способ устранения:

Обновить операционную систему ПК. См. минимальные системные требования в руководстве по инсталляции Albatros (InstallationGuide.pdf).

16645 Ошибка при отправке кода встроенного ПО

Причина:

Произошла ошибка при инициализации системы. В частности, не удалась отправка файла встроенного ПО на модуль.

Способ устранения:

Попробовать сбросить блок управления. Если проблема не устраняется, обратиться к изготовителю станка.

16646 Оказалось невозможным снова направить в выполнение код встроенного ПО

Причина:

Произошла ошибка при повторной инициализации системы. В частности, не удался пуск встроенного ПО после того, как оно было ранее остановлено.

Способ устранения:

Попробовать сбросить систему. Если проблема не устраняется, обратиться к изготовителю станка.

16897 Не инсталлирован RTX

Причина:

Инсталлированная версия Albatros требует, чтобы на ПК была инсталлирована RTX, но она не была обнаружена.

Способ устранения:

Инсталлировать RTX, а если он уже присутствует - инсталлировать его снова. См. руководство по инсталляции RTX Albatros (InstallationRTXGuide.pdf).

16898 Пользователь не имеет прав администратора

Причина:

Albatros был запущен пользователем, не имеющим прав администратора ПК. Для правильного функционирования Albatros требуются права администратора.

Способ устранения:

Закрывать текущий сеанс и войти в систему как "Administrator" или как другой пользователь с правами администратора.

16899 Неправильный размер ОЗУ модуля

Причина:

Объем ОЗУ, обнаруженный на удаленном модуле, не соответствует предусмотренному. Как правило, эта ошибка указывает на повреждение аппаратного обеспечения.

Способ устранения:

Если проблема не устраняется, обратиться к изготовителю станка.

16900 Неправильный IP-адрес модуля

Причина:

Был обнаружен удаленный модуль, IP-адрес которого не относится к подсети ПК-супервизора. Albatros не может поддерживать правильную связь с модулем.

Способ устранения:

Проверить, что правильны служебные настройки AlbdHCP и сетевой платы ПК. См. руководство по установке Albatros (InstallationGuide.pdf).

16901 Модуль уже подключен к другой установке

Причина:

Был обнаружен удаленный модуль, который представляется подключенным к другому ПК-супервизору. Это может обуславливаться наличием в сети другого ПК, на котором выполняется Albatros и который использует тот же модуль. Может также обуславливаться неисправностью программного обеспечения связи модуля.

Способ устранения:

Проверить, что удаленный модуль не используется другим ПК-супервизором. Сбросить модуль. Если проблема не устраняется, обратиться к изготовителю станка.

16902 Модуль не сконфигурирован

Причина:

Был обнаружен модуль, который не представляется сконфигурированным в "Конфигурации системы" Albatros.

Способ устранения:

Сконфигурировать модуль.

16903 Настройки брандмауэра не допускают связь

Причина:

Был обнаружен установленный на ПК брандмауэр, не допускающий связь между Albatros и удаленными модулями.

Примечание: Albatros может определять наличие брандмауэра Windows, но не других брандмауэров, например, входящих в некоторые антивирусные программы.

Способ устранения:

Изменить настройки брандмауэра или отключить его.

16904 Сетевая плата отсутствует или отключена

Причина:

Не была обнаружена сетевая плата, которая может быть использована для связи с удаленными модулями.

Примечание: тот факт, что обнаруживается плата, не гарантирует, что она правильно сконфигурирована и подключена.

Способ устранения:

Проверить наличие и правильность конфигурации сетевой платы. Если проблема не устраняется, обратиться к производителю станка.

16905 Отсутствует код встроенного ПО блока управления

Причина:

Albatros не находит файл встроенного ПО на жестком диске ПК. Эта проблема может возникнуть после случайного стирания файлов встроенного ПО или после неправильного обновления.

Способ устранения:

Проверить, что присутствуют файлы, содержащиеся в папке встроенного ПО инсталляции Albatros, и что они имеют соответствующую версию. Обратиться к изготовителю станка.

16906 Версия RTX несовместима с кодом встроенного ПО блока управления

Причина:

Была обнаружена версия RTX, несовместимая с инсталлированным встроенным ПО.

Способ устранения:

Инсталлировать правильную версию RTX или обновить встроенное ПО. Обратиться к изготовителю станка.

16907 Версия операционной системы несовместима с кодом встроенного ПО блока управления

Причина:

Версия операционной системы удаленного модуля несовместима с инсталлированным встроенным ПО.

Способ устранения:

Инсталлировать на удаленный модуль правильную версию операционной системы или обновить встроенное ПО. Обратиться к изготовителю станка.

17153 ТипПлаты: нет кода встроенного ПО передатчика GreenBUS

Причина:

Не был найден файл встроенного ПО в папке встроенного ПО. Как правило, эта проблема возникает при случайном стирании файла или неполной или поврежденной инсталляции.

Способ устранения:

Переинсталлировать Albatros, предварительно выполнив резервную копию системы. Обратиться к изготовителю станка.

17154 ТипПлаты: нет части кода встроенного ПО передатчика GreenBUS

Причина:

Файл с кодом встроенного ПО передатчика GreenBus присутствует в папке встроенного ПО, но представляется поврежденным или неполным.

Способ устранения:

Переинсталлировать Albatros, предварительно выполнив резервную копию системы. Обратиться к изготовителю станка.

17155 ТипПлаты: ошибка отправки кода программы самозагрузки передатчика GreenBUS**Причина:**

Произошла ошибка при инициализации системы. В частности, не удалась отправка файла встроенного ПО на модуль.

Способ устранения:

Попробовать сбросить блок управления. Если проблема не устраняется, обратиться к изготовителю станка.

17156 ТипПлаты: ошибка отправки кода главной функции передатчика GreenBUS**Причина:**

Произошла ошибка при инициализации системы. В частности, не удалась отправка файла встроенного ПО на модуль.

Способ устранения:

Попробовать сбросить блок управления. Если проблема не устраняется, обратиться к изготовителю станка.

17157 ТипПлаты: нет кода программы самозагрузки**Причина:**

Не был найден файл встроенного ПО в папке встроенного ПО. Как правило, эта проблема возникает при случайном стирании файла или неполной или поврежденной инсталляции.

Способ устранения:

Переинсталлировать Albatros, предварительно выполнив резервную копию системы. Обратиться к изготовителю станка.

17158 ТипПлаты: нет кода главной функции**Причина:**

Не был найден файл встроенного ПО в папке встроенного ПО. Как правило, эта проблема возникает при случайном стирании файла или неполной или поврежденной инсталляции.

Способ устранения:

Переинсталлировать Albatros, предварительно выполнив резервную копию системы. Обратиться к изготовителю станка.

17159 ТипПлаты: ошибка отправки кода программы самозагрузки**Причина:**

Произошла ошибка при инициализации системы. В частности, не удалась отправка файла встроенного ПО на модуль.

Способ устранения:

Попробовать сбросить блок управления. Если проблема не устраняется, обратиться к изготовителю станка.

17160 ТипПлаты: ошибка отправки кода главной функции**Причина:**

Произошла ошибка при инициализации системы. В частности, не удалась отправка файла встроенного ПО на модуль.

Способ устранения:

Попробовать сбросить блок управления. Если проблема не устраняется, обратиться к изготовителю станка.

17409 Оказалось невозможным направить вспомогательный исполняемый файл

Причина:

Эта ошибка может возникнуть при обновлении встроенного ПО удаленного модуля. Может обуславливаться моментной неполадкой сети, а также повреждением встроенного ПО модуля. Сообщение об ошибке может включать код ошибки.

Способ устранения:

Попробовать выключить, а затем снова включить удаленный модуль и повторить процедуру обновления. Если проблема не устраняется, обратиться к изготовителю станка.

17410 Оказалось невозможным выполнить вспомогательный исполняемый файл

Причина:

Произошла ошибка при инициализации системы. В частности, оказалось невозможным выполнить вспомогательную программу. Кроме того, в сообщении об ошибке содержится имя вспомогательной программы, а также может содержаться код ошибки.

Способ устранения:

Попробовать сбросить систему. Если проблема не устраняется, обратиться к изготовителю станка.

17667 ИмяDLL: оказалось невозможным направить в выполнение код встроенного ПО

Причина:

Произошла ошибка при инициализации системы. В частности, оказалось невозможным выполнить код встроенного ПО. "ИмяDLL" соответствует компоненту, вызвавшему ошибку.

Способ устранения:

Попробовать сбросить систему. Если проблема не устраняется, обратиться к изготовителю станка.

17668 ИмяDLL: оказалось невозможным получить указатель к совместно используемой ОЗУ

Причина:

Произошла ошибка при инициализации системы. В частности, было невозможно открыть канал связи со встроенным ПО. "ИмяDLL" соответствует компоненту, вызвавшему ошибку.

Способ устранения:

Попробовать сбросить систему. Если проблема не устраняется, обратиться к изготовителю станка.

17921 Оказалось невозможным направить NODETRA

Причина:

Эта ошибка может возникнуть при обновлении встроенного ПО удаленного модуля. Может обуславливаться моментной неполадкой сети, а также повреждением встроенного ПО модуля. Сообщение об ошибке может включать код ошибки.

Способ устранения:

Попробовать выключить, а затем снова включить модуль и повторить процедуру обновления. Если проблема не устраняется, обратиться к изготовителю станка.

17922 NODETRA повторно не запустился

Причина:

Эта ошибка может возникнуть при обновлении встроенного ПО удаленного модуля. Может обуславливаться моментной неполадкой, а также повреждением встроенного ПО модуля. Сообщение об ошибке может включать код ошибки.

Способ устранения:

Попробовать выключить, а затем снова включить модуль и повторить процедуру обновления. Если проблема не устраняется, обратиться к изготовителю станка.

17923 NODETRA не выполняется

Причина:

В сети был обнаружен удаленный модуль, программное обеспечение связи которого не выполняется. Как правило, это свидетельствует о неисправности программного обеспечения связи. Сообщение об ошибке может включать код ошибки.

Способ устранения:

Попробовать выключить, а затем снова включить модуль. Если проблема не устраняется, обратиться к изготовителю станка.

18177 NODETRA попытался получить доступ к недействительному адресу

Причина:

Программное обеспечение для связи удаленного модуля генерировало ошибку. Сообщение об ошибке может включать код ошибки.

Способ устранения:

Попробовать выключить, а затем снова включить модуль. Если проблема не устраняется, обратиться к изготовителю станка.

6.3 Общие сигналы

6.3.1 Albatros начинает выполнение

Он сообщает о запуске Albatros и отображает некоторую полезную информацию о версии программы и среде выполнения.

6.3.2 Albatros завершает выполнение

Он сообщает, что Albatros собирается закончить выполнение.

6.3.3 Компьютер входит в режим ожидания

Он сообщает, что компьютер входит в режим ожидания. С этого момента Albatros больше не может отвечать на запросы и сигналы из цикла GPL.

6.3.4 Компьютер выходит из режима ожидания

Он сообщает, что компьютер только что вышел из режима ожидания. Albatros возобновляет выполнение без перезапуска.

6.3.5 Выключить компьютер

Он сообщает, что компьютер собирается выключиться, пока Albatros еще работает.

6.3.6 Текущий уровень доступа

Он сообщает об изменении уровня доступа к функциям Albatros, обычно для техобслуживания или изменения цикла или конфигурации.

6.3.7 Обновление программного обеспечения модулей

Он сообщает, что было запрошено обновление программного обеспечения и встроенного программного обеспечения, имеющихся в пульте дистанционного управления.

6.3.8 Отправка конфигурации модулям

Он сообщает, что было запрошено обновление конфигурации и цикла, имеющихся в пульте дистанционного управления.

7 Конфигурация Системы

7.1 Введение

В главе, посвященной составу системы, уже рассматривалось, что система Albatros состоит из одного или нескольких модулей, образующих установку, и что каждый из них имеет иерархическую структуру. Для конфигурации станка с точки зрения Albatros необходимо выполнить последовательность операций, с помощью которых будут конфигурироваться различные логические уровни и расположенное далее аппаратное обеспечение.

При конфигурации системы выполняются следующие основные операции:

- [Конфигурация Модуля](#)
- [Определение Групп и Подгрупп](#)
- [Конфигурация Устройств](#)
- [Конфигурация Системы](#)
- [Конфигурация Аппаратного обеспечения](#)
- [Конфигурация Виртуальное-Физическое](#)

Практически, Конфигурации Модуля, Группы и Станка определяют логическую структуру станка, а Конфигурации Системы, Аппаратного обеспечения и Виртуальное-Физическое определяют его структуру на физическом уровне.

Рассмотрим подробно некоторые из этих пунктов в следующих параграфах.

7.2 Конфигурация устройств

7.2.1 Введение

В главе, относящейся к составу системы Albatros, представлены типы устройств, которые могут появляться в модуле. Затем устройства представляются снова, но с точки зрения их конфигурации.

Для каждого типа существует максимальное число устройств, конфигурируемых в соответствии со следующим списком:

Тип устройства	Макс. число
Аналоговый вход	128
Аналоговый выход	128
Цифровой вход	4096
Цифровой выход	4096
Входной порт	512
Выходной порт	512
Ось	240
Таймер	128
Счетчик	128
Флаговый бит	1024
Флажковый выключатель	256
Флаговый порт	256

7.2.2 Общее устройство

Большая часть устройств требует одинаковых параметров конфигурации. Ниже представлена конфигурация цифрового входа, однако эти положения касаются и следующих элементов:

- Флаговый бит
- Флажковый выключатель
- Аналоговые выходы
- Входной порт
- Выходной порт
- Флаговый порт
- Таймеры
- Счетчики

Для конфигурации устройства из приведенных выше необходимо указать следующие настройки:

- **Имя:** имя устройства длиной макс. 40 символов.
- **Комментарий:** краткое описание устройства, которое может переводиться на различные языки.

- **Доступы для чтения:** указывает минимальный уровень доступа, необходимый для того, чтобы устройство могло отображаться в окнах диагностики и на обзорных экранах.
- **Доступы для записи:** указывает минимальный уровень доступа, необходимый для того, чтобы можно было изменить состояние устройства.
- **Общедоступный:** указывает, может ли состояние устройства считываться и меняться кодом GPL, не относящимся к группе, в которой находится устройство.

7.2.3 Цифровой выход

Цифровой выход имеет на один параметр больше по сравнению со стандартными устройствами: Моностабильный

Для конфигурации цифрового выхода необходимо указать следующие настройки:

- **Имя:** имя устройства длиной макс. 40 символов.
- **Комментарий:** краткое описание устройства, которое может переводиться на различные языки.
- **Моностабильный:** при выборе конфигурирует выход как моностабильный, т. е. когда выход задается на ВКЛ., он автоматически возвращается на ОТКЛ. через 200 мс.
- **Доступы для чтения:** указывает минимальный уровень доступа, необходимый для того, чтобы устройство могло отображаться в окнах диагностики и на обзорных экранах.
- **Доступы для записи:** указывает минимальный уровень доступа, необходимый для того, чтобы можно было изменить состояние устройства.
- **Общедоступный:** указывает, может ли состояние устройства считываться и меняться кодом GPL, не относящимся к группе, в которой находится устройство.

7.2.4 Аналоговый вход

Аналоговый вход имеет на один параметр больше по сравнению со стандартными устройствами: *Тип напряжения на входе.*

Для конфигурации аналогового входа необходимо указать следующие настройки:

- **Имя:** имя устройства длиной макс. 40 символов.
- **Комментарий:** краткое описание устройства, которое может переводиться на различные языки.
- **Тип:** позволяет выбирать интервал напряжений, считанных на входе.
- **Доступы для чтения:** указывает минимальный уровень доступа, необходимый для того, чтобы устройство могло отображаться в окнах диагностики и на обзорных экранах.
- **Доступы для записи:** указывает минимальный уровень доступа, необходимый для того, чтобы можно было изменить состояние устройства.
- **Общедоступный:** указывает, может ли состояние устройства считываться и меняться кодом GPL, не относящимся к группе, в которой находится устройство.

7.2.5 Ось

Базовые Данные

Должны указываться следующие базовые данные:

- **Имя:** имя устройства длиной макс. 40 символов.
- **Описание:** краткое описание устройства, которое может переводиться на различные языки.
- **Разрешение:** разрешение кодера, зависит от характеристик кодера и указанных единиц измерения. Следует помнить, что платы осей Albatros подсчитывают как импульс как фронты подъема, так и фронты снижения обеих фаз кодера (таким образом, кодер с 2500 импульсами на оборот будет рассматриваться как кодер с 10000 импульсами на оборот).
- **Тип:** тип оси. Предусматриваются следующие типы: **Аналоговый** (контролируется в аналоговом режиме), **Шаговый**, **Цифровой**, **Подсчет** (только считывание кодера), **Виртуальный**.
- **Единица измерения:** единица измерения, используемая для выражения координат оси. От нее зависят все производные величины, поэтому этот параметр рекомендуется задавать раньше любого другого.
- **Инверсия фаз:** позволяет компенсировать при помощи программного обеспечения возможную инверсию электромонтажа фаз кодера.
- **Инверсия Эталона:** позволяет инвертировать эталон скорости оси. При использовании вместе с инверсией фаз позволяет инвертировать направление оси (если электромонтаж выполнен правильно).
- **Включить Отметки:** имеется только для осей подсчета, автоматически обнуляет координату при обнаружении отметки кодера.

Параметры Движения

Параметры, используемые для перемещения оси точка-точка.

- **Макс. скорость:** максимальная скорость оси.
- **Ускорение:** продолжительность линейной функции ускорения
- **Замедление:** продолжительность линейной функции замедления
- **Мин. скорость:** задается только для шаговых осей, скорость, которую достигает двигатель за один шаг.
- **Тип изменения:** тип линейной функции ускорения и замедления. Не может задаваться для шаговых двигателей
- **Пропорциональное:** пропорциональный коэффициент контроллера PID контура положения
- **Интегрируемое:** интегрирующий коэффициент контроллера PID контура положения
- **Производное:** производный коэффициент контроллера PID контура положения
- **Подача вперед:** процент подачи вперед. Допускает соответствующее снижение погрешности контура на такой же скорости.
- **Подача вперед уск.:** процент подачи вперед с ускорением. Допускает снижение остаточной погрешности контура (не сниженной подачей вперед) на этапах ускорения и замедления оси.
- **Образцы интегрируемого:** задает число образцов погрешности контура, использованных для расчета интегральной составляющей. Действительные значения находятся в пределах от 1 до 200. Значение по умолчанию: 50. См. инструкцию gpl [SETINTEGTIME](#).

Параметры Интерполяции

Параметры, используемые для перемещения при интерполяции оси.

Они имеют то же значение, что и параметры, рассмотренные в заданиях движения, за исключением Минимальной скорости, которая определяется только для шаговых осей в движениях точка-точка. Однако они используются для интерполированных движений.

ВАЖНОЕ ПРИМЕЧАНИЕ: значения ускорения и замедления, заданные в параметрах интерполяции, не могут быть меньше соответствующих значений в Параметрах Движения.

Другие Параметры

- **Скорость в ручном режиме:** указывает максимальную скорость конфигурации, которая может использоваться для перемещений вручную. Она никогда не будет выше максимальной заданной скорости.
- **Динамич. серво-ош.:** подключает или отключает динамическую сервоошибку. Значением по умолчанию является отключенная динамическая сервоошибка, поэтому пороговая сервоошибка остается подключенной. См. инструкцию gpl [SETMAXERTYPE](#).
- **Эталонная скорость и Ошибка кольца:** эти два значения используются для расчета реального линейного отношения двух величин в машине. Чтобы значения учитывались, они должны быть как положительными, так и ненулевыми, и должно быть включено поле **Динамич. серво-ош.**
- **Ожидание остановленной оси:** подключает или отключает функцию восстановления выбега. Вводит в конец каждого движения время ожидания в 50 мс.
- **Таймаут перемещ. оси:** Действительные значения находятся в пределах от 0 до 1024. См. инструкцию GPL [ENABLESTARTCONTROL](#).
- **Неправ. предел подключения кодера:** Заданные значения выражены в единицах измерения, в которых выражено разрешение оси. Задаваемые значения должны находиться в пределах от 128/разрешение оси и 16384/разрешение оси. Умолчание вычисляется на основе числа шагов, равных 1024, т.е. 1024/разрешение оси.
- **Предел серво-ошибки положителен:** максимальное значение погрешности контура для контура регулирования в положительном направлении.
- **Предел серво-ошибки отрицателен:** максимальное значение погрешности контура для контура регулирования в отрицательном направлении.
- **Предел оси положительной:** максимальное значение хода оси в положительном направлении.
- **Предел оси отрицательной:** максимальное значение хода оси в отрицательном направлении.
- **Окно прибытия на координате положительного:** допуск на координате прибытия в положительном направлении.
- **Окно прибытия на координате отрицательного:** допуск на координате прибытия в отрицательном направлении.

Параметры Эталона

- **Эталон:** эталонное значение напряжения, которому соответствует максимальная скорость.

- **Автоматическая настройка:** подключает или отключает расчет восстановления автоматического сдвига. Обычно подключена.
- **Начальный сдвиг:** Значение, на которое должен задаваться начальный сдвиг эталона. Значение должно находиться в пределах от -10 до 10. Значение по умолчанию: 0.
- **Частота фильтра отметки:** Фильтруемое значение частоты. Значение должно находиться в пределах от 0 до 500.
- **Минимальное напряжение:** Задаёт параметры минимального напряжения для указанной оси. Отрицательное значение должно находиться в пределах от -10 до 0, а положительное - от 0 до +10. См. инструкцию `grl SETDEADBAND`.
- **Порог:** Задаёт пороговые значения. Они всегда меньше или равны соответствующим значениям минимального напряжения, поэтому отрицательное значение порога должно находиться между 0 и отрицательным минимальным значением напряжения. Максимальное значение порога должно находиться между 0 и положительным минимальным значением напряжения.

Уровни доступа

- **Доступы для чтения:** указывает минимальный уровень доступа, необходимый для того, чтобы ось могла отображаться в окнах диагностики и на обзорных экранах.
- **Доступы для записи:** указывает минимальный уровень доступа, необходимый для того, чтобы можно было изменить состояние оси.
- **Общедоступный:** указывает, может ли состояние оси считываться и меняться кодом GPL, не относящимся к группе, в которой находится ось.

Сцепление осей

Параметры сцепления осей. Это коэффициенты контроллера PID, компенсирующего различия погрешности контура между главной осью и ведомыми осями.

- **Пропорциональное:** пропорциональный коэффициент
- **Интегрируемое:** интегрирующий коэффициент
- **Производное:** производный коэффициент

Корректоры линейности

Задание корректоров линейности оси. Корректоры позволяют компенсировать погрешности позиционирования оси, обуславливаемые неточностью механики оси (автокорректоры), и погрешности в результате воздействия других осей станка (перекрестные корректоры), обычно связанного с изгибом конструкции. Корректоры не подключаются автоматически, их следует подключать в окне изменения значений коррекции (кнопка **[Правка...]**) и активировать кодом GPL инструкцией [ENABLECORRECTION](#).

- **Интервал коррекции:** позволяет задавать расстояние между двумя последующими коррекциями. Число измерений будет определяться длиной оси, деленной на длину интервала коррекции.
- **Имя файла корректоров:** позволяет задавать имя файла, в котором будут сохраняться значения коррекции. Это файл ASCII, в котором значения разделяются символом ";". Это позволяет выполнять его правку обычным текстовым редактором. Расширение файла не указывается, автоматически задается расширение ".csv" (comma separated values).
- **Данные коррекции:** позволяет указывать список осей, для которых текущая ось генерирует коррекцию. Текущая ось всегда включена в список, т. е. автокорректор всегда присутствует. Можно указать еще до 5 других осей. Для добавления оси выбрать ее в списке слева и нажать кнопку **[>>Добавить]**. Для удаления оси выбрать ее в списке справа и нажать кнопку **[Удалить<<]**. Для указания значений коррекции выбрать ось в списке справа и нажать кнопку **[Правка...]**, при этом открывается окно с таблицей для ввода значений коррекции.

ПРИМЕЧАНИЕ: Существует максимальный предел в **235** корректоров линейности, управляемых системой для каждой оси. Соответственно, учитывая длину оси, интервал измерения должен будет иметь длину больше или равную двести тридцать пятой части длины оси. Например, для оси длиной 2500 мм необходимо задать интервал коррекции, больший или равный 10.63 мм. Кроме того, существует предел максимального значения отдельно взятой коррекции, которая должна быть меньше 1024 шагов кодера, например, для оси с разрешением 256 шагов/мм максимальная коррекция составляет ± 4 мм.

7.3 Логическая конфигурация

7.3.1 Конфигурация установки

Для определения нового станка или модификации уже существующего необходимо войти в экран "Конфигурация модуля". Конфигурацией модуля является конфигурация модулей, образующих установку.

Открытие среды конфигурации возможно на уровне доступа производителя или выше.

Доступ к конфигурации

Выбрать из меню **Файл** пункт **Открыть конфигурацию**

Если в установке уже нет сконфигурированных модулей, автоматически открывается "Конфигурация модуля, или же открывается "Конфигурация станка". В этом случае для перехода к конфигурации модуля можно выполнить следующее:

Выбрать из меню **Правка** пункт **Конфигурация модуля**

Для добавления модуля к установке достаточно нажать кнопку **[Создать]**. Кнопка **[Правка]** позволяет изменить данные уже существующего модуля, кнопка **[Удалить]** позволяет удалять модуль, кнопка **[Закрыть]** позволяет выйти из конфигурации установки.

Данные, которые обозначают станок и которые необходимо указывать:

- номер модуля: целое число по порядку, которое, если не указывается, задается системой
- краткое описание.

Имеются некоторые данные указывающие аппаратное обеспечение:

- **Частота управления осями:** указывает частоту с которой данные периодически обмениваются между числовым управлением и устройствами с ним связанными через полевую шину.
- **Число канала интерполяции:** указывает максимальное число каналов интерполяции (то есть, групп осей, которые выполняют движение интерполированного типа), которыми можно управлять одновременно.
- **Процентное использование ЦП:** указывает процент времени, соответствующего времени управления осями (то есть, обратному "частоты управления осями"), который посвящен выполнению встроенного ПО.

Это же окно можно открыть из ветви модуля Конфигурации групп, из ветви "Модуль Конфигурации станка" и из ветви модуля Конфигурации аппаратного обеспечения.

7.3.2 Конфигурация групп

При первом проектировании станка необходимо определить каждый его компонент и написать все циклы управления. Во многих случаях проектирование выполняется на уже изготовленном станке, который изменяется в зависимости от свойств нового станка.

Создание группы

Для создания новой группы необходимо войти в экран конфигурации групп. Первой ветвью древовидной структуры является модуль, от которого отходят все группы, подгруппы и устройства. При нажатии кнопки **[ВВОД]** или кнопки **[Правка]** открывается диалоговое окно для изменения данных модуля.

Выбрать из меню **Правка** пункт **Группы**

Здесь можно создать новые группы, изменить или удалить существующие и скопировать группу.

Список команд для создания, правки, удаления групп, копирования и вставки групп, подгрупп и устройств

Команда	Действие
---------	----------

Создает новую группу, подгруппу, устройство	[CTRL+BВOD], Кнопка [Создать], Правка-> Создать, контекстное меню
Изменяет группу, подгруппу, устройство	[BВOD], Кнопка [Правка], Правка-> Правка ..., контекстное меню
Удаляет группу, подгруппу, устройство	[Отмена], Кнопка [Удалить], Правка-> Удалить, контекстное меню
Включает или отключает использование группы на станке	Контекстное меню, Кнопка [Включить]
Копирует группу, подгруппу, устройство	[CTRL+C], Кнопка [Копировать], Правка-> Копировать, контекстное меню
Вставляет группу, подгруппу, устройство	[CTRL+V], Кнопка [Вставить], Правка->Вставить, контекстное меню

При создании новой группы отображается следующее окно, в котором необходимо задать:

- Имя группы
- Комментарий (который может быть переведен на языки, управляемые Albatros)

Кроме того, можно отметить группу как **Интергруппа**. По меньшей мере одна группа должна задаваться как интергруппа, т. к. этот выбор используется Albatros для определения "основной" группы станка. Это группа, главная функция которой (с таким же именем, что и группа) автоматически запускается при запуске. Такой механизм служит для инициализации станка и выполнения при пуске задач, проверяющих правильность общей работы перед передачей управления пользователю.

Когда вы отключаете группу, в которой есть устройства, подключенные к физическим устройствам, вас спрашивают, хотите ли вы отменить соединение виртуальное физическое. Если вы решите сохранить соединения, пин-коды физических устройств, к которым они подключены, будут отображаться серым цветом в графическом представлении устройства виртуального физического.

Добавление подгруппы к группе

Для создания подгруппы группы необходимо установиться на группу.

Если вы не желаете присваивать подгруппе какое-либо конкретное имя, можно выбрать пункт *Список устройств* и нажать кнопку **[OK]**. Имя подгруппы задается автоматически.

Теперь можно вводить в подгруппу отдельные устройства. Механизм подобен тому, который используется для создания подгрупп. Отображается окно со списком доступных устройств.

После выбора нужного устройства нажать кнопку **[OK]** для подтверждения.

Отображается окно, позволяющее задавать имя, комментарий и другие данные, которые меняются в зависимости от выбранного устройства. Подробное описание типов устройств и их настроек приводится далее в главе [Конфигурация устройства](#).

Копирование устройства

Функция копирования устройства позволяет копировать устройство. Прежде всего необходимо выбрать устройство, а затем нажать кнопку **[Копировать]**. Для ввода устройства в список необходимо выбрать ветку, в которой вставить устройство и включить команду **[Вставить]**. В диалоговое окно, необходимо ввести новое имя устройства.

Копирование подгруппы

Функция копирования подгруппы позволяет копировать подгруппу, включая все содержащиеся в ней ее устройства.

Для ввода подгруппы, необходимо выбрать ветку, в которой вставить ее и включить команду **[Вставить]**. В диалоговое окно, необходимо ввести новое имя подгруппы.

Копирование группы

Функция копирования группы позволяет копировать группу, включая все содержащиеся в ней ее подгруппы и устройства. Кроме того, копируется также возможный связанный с ней обзорный экран группы (обзорный экран, имя которого совпадает с именем группы).

Позволяет быстро создавать группы со структурой, подобной структуре уже существующей группы, без необходимости повторного создания поочередно всех устройств. Для копирования группы необходимо выбрать копируемую группу, нажать кнопку **[Копировать]** и ввести в диалоговое окно имя новой группы.

Копирование устройств, подгрупп и групп также может выполняться между различными модулями.

Выбор групп, принадлежащих к станке

После создания архива группы фактически присутствующие группы должны быть отключены или включены.

Все группы присутствуют на станке, если вы не решите отключить их, используя кнопку **[Отключить]** или выбрав ту же команду в контекстном меню. Если группа была отключена, рядом с названием группы появляется слово **Не присутствует**.

Для просмотра только групп на станке, выберите пункт **Станок** в меню **Правка**.

Для вставки новой группы, нажмите кнопку **[Вставить]**. Появится окно со списком групп, присутствующих в архиве Групп, но еще не введенных в станок.

На этом этапе вы должны выбрать выбранную группу и **перетащить ее мышью** в окно Конфигурация станка или нажать кнопку **[Вставить]**.

Также можно удалить существующую группу с помощью кнопки **[Удалить]** или выполнить поиск имени группы или устройства в дереве состава станка.

В станке должна присутствовать одна интергруппа.

7.4 Физическая конфигурация

7.4.1 Конфигурация системы


Конфигурация системы позволяет ассоциировать физические ресурсы (блоки управления) с модулями, определенными в логической конфигурации. Это возможно в диалоговом окне "Конфигурация системы". Отображается список **модулей** установки и каждому из них можно присвоить **Узел сети**.

- **Локальный узел:** "Локальные" системы, в которых аппаратное обеспечение, реализующее управление, установлено непосредственно на ПК, образующее пользовательский интерфейс системы
- **Имя узла сети:** "Удаленные" системы, в которых аппаратное обеспечение, реализующее управление, соединено с ПК по сети
- **Не сконфигурирован:** никакой конфигурации. Это начальное умолчание. Если оставляется этот выбор, затем в диалоговом окне **Подключение узлов сети** можно будет ассоциировать удаленный модуль.

Можно сконфигурировать до 16 модулей и только один может быть сконфигурирован как локальный узел.

Для задания модуля выбрать кнопку **[Правка]** или два раза щелкнуть мышью на изменяемом узле сети.

При открытии выпадающего меню отображается список доступных удаленных модулей, выбор использования Локального узла или выбор задания модуля как неконфигурированного. Для

подтверждения выбора необходимо выбрать кнопку .

ВАЖНОЕ ПРИМЕЧАНИЕ: Работа Albatros в станке защищена аппаратным ключом USB, сконфигурированным компанией TPA.

7.4.2 Конфигурация аппаратного обеспечения

На аппаратной конфигурации определяются платы и узлы, составляющие систему.

Плата, занимающая первое положение в списке, называется Главной платой.

Могут конфигурироваться следующие типы платы:

- TMSbus до двух
- TMSbus+ до четырех
- TMSCombo+ до четырех
- DualMech до четырех

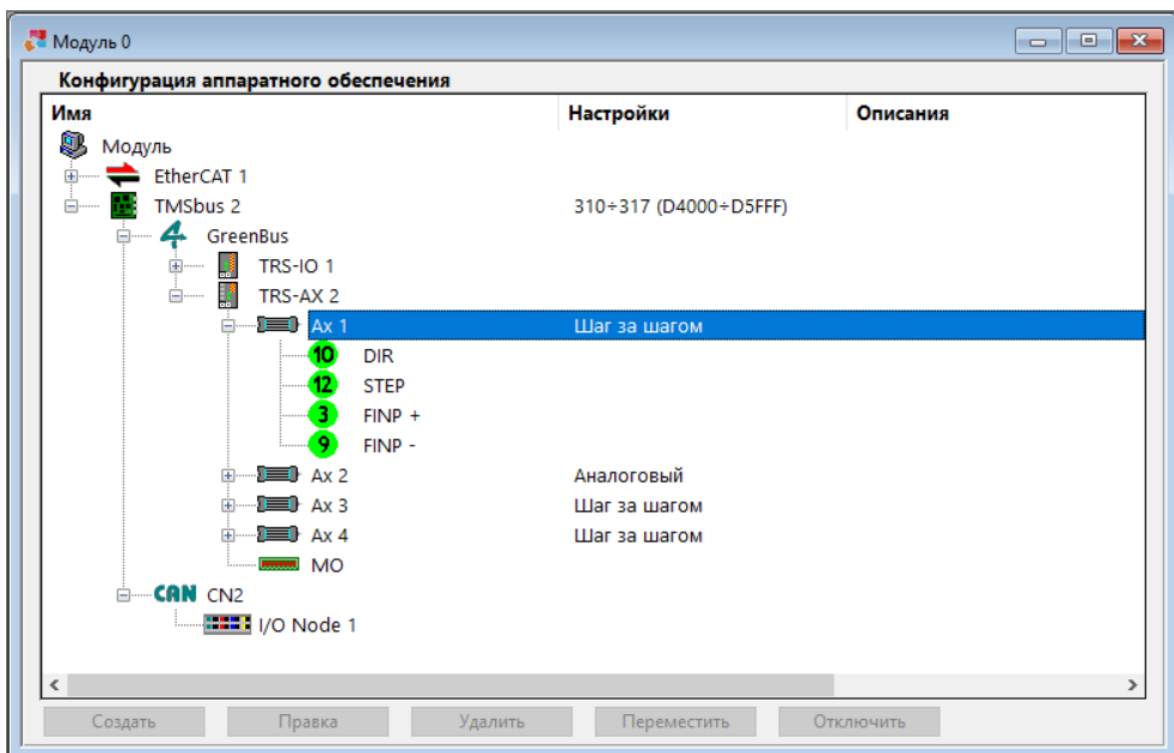
- DualMech Mono до четырех
- TMSCan до двух
- TMSCan+ до четырех
- AlbMech до двух
- EtherCAT одна

Описание окна конфигурации аппаратного обеспечения

Окно конфигурации аппаратного обеспечения открывается при выборе из меню пункта **Правка- >Аппаратное обеспечение**.

Для вставки платы или удаленного модуля входов/выходов или узла CAN или EtherCAT, необходимо нажать кнопку **[Создать]**. При этом откроется окно, позволяющее выбрать плату или удаленный модуль входов/выходов и для шины CAN и GreenBUS положение для ввода.

Как правило, для каждого модуля можно сконфигурировать макс. 4 платы, а в зависимости от типа платы и шины, переменное количество удаленных модулей входов/выходов.



Конфигурация аппаратного обеспечения

В столбце **Настройки** присваивается информация, связанная с платой или узлом.

При командой **[Переместить]** можно переместить плату или узел или расширение TRS-IO или TRS-CAT с одного положения дерева на другое. Расширения можно перемещать только в пределах одного узла. Эта операция сохраняет подключения, которые присутствуют в конфигурации [Виртуальное-Физическое](#).

Кроме того, можно также отключить узел или расширение TRS-IO или TRS-CAT. При отключении сохраняются подключения в конфигурации Виртуальное-Физическое.

Если узел принадлежит к шине GreenBUS, узел и подключенные к нему устройства полностью игнорируются системой. Поэтому не генерируется никакой ошибки, если модуль не обнаруживается при инициализации, и не генерируется никакой ошибки при выполнении инструкции GPL на ассоциированном с модулем устройстве.

Если узел или расширение принадлежит к сети EtherCAT или расширение принадлежит к шине GreenBUS, он не должен присутствовать в сети. Если доступ к отключенному устройству осуществляется по GPL, возникает системная ошибка.

Поэтому при использовании этой функции необходимо быть особенно внимательными.

Для отключения узла или расширения использовать команду **[Отключить]**, для включения узла или расширения использовать команду **[Включить]**.

Поэтому при использовании этой функции необходимо быть особенно внимательными.

Предопределенные конфигурации

Несколько предопределенных конфигураций доступны. Из меню **Правка->Изменить тип управления** или из контекстного меню в ветке Модуля можно выбрать нужную конфигурацию. Если конфигурация новая, дерево заполняется определенными платами и узлами, в противном случае выполняются проверки для проверки совместимости уже имеющегося оборудования с выбранным типом модуля. Все, что не совместимо, устраняется.

Конфигурировать узел шины TRA

Типы удаленного модуля входов/выходов, которые могут конфигурироваться на GreenBus (v3.0):

- Albre8 8 цифровых входов и 8 выходов
- Albre16 16 каналов, конфигурируемых при помощи программного обеспечения как цифровой вход или выход
- Albre24 24 цифровых входа и 24 выхода
- Albre48 48 цифровых входов и 48 выходов
- Albrem 10 входных портов и 10 выходных портов
- AlbSTEP 8 цифровых входов и 6 выходов, шаговый двигатель
- AlbEV 20 или 24 электроклапана (25-контактный разъем D-sub)
- Albrea 4 аналоговых входа и 4 выхода

Типы удаленного модуля, конфигурируемые на GreenBus (v4.0):

- TRS-AX 4 аналоговых или пошаговых оси
- TRS-EV-24 24 электроклапана (25-контактный разъем D-sub)
- TRS-16 16 каналов, конфигурируемых при помощи программного обеспечения как цифровой вход или выход
- TRS-IO 16 каналов, конфигурируемых при помощи программного обеспечения как цифровой вход или выход, расширяемый модулями типа TRS-IO-E и модулями TRS-AN-E до 5 и модулями TRS-AC-E
- TRS-IO-E 16 каналов, конфигурируемых при помощи программного обеспечения как цифровой вход или выход, используемые только в качестве расширений модуля TRS-IO
- TRS-AN-E 1 аналоговый вход и 1 аналоговый выход, используемые только как расширения модуля TRS-IO
- TRS-REM общий удаленный, использованный для реализации особых модулей. Можно их подсоединить до 12 входных портов и 12 выходных портов
- TRS-AC-E 1 ось подсчета и 2 цифровых входа, конфигурируемые как отметка нуля и быстрый вход.
В таблице ниже указывается максимальное число TRS-AC-E, конфигурируемых в TRS-IO.
Если между TRS-IO-E и TRS-AN-E есть 3 расширения, можно настроить только одно расширение TRS-AC-E. Если настроено только одно расширение (TRS-IO-E или TRS-AN-E), можно иметь до 2 расширений TRS-AC-E.

Удаленные модули TRS-AX, TRS-IO, TRS-REM и TRS-16 могут подключаться исключительно к платам TMSbus, TMSbus+ и TMSCombo+.

К каждой плате TMSbus и TMSbus+ можно подсоединить не более 4 удаленных модулей TRS-AX.

Типы удаленного модуля Тра, конфигурируемые на шине EtherCAT:

- TRS-CAT 16 каналов, конфигурируемых при помощи программного обеспечения как цифровой вход или выход, расширяемый модулями типа TRS-IO-E, модулями TRS-AN-E и модулями TRS-AC-E
- STAR-CAT преобразует линейную топологию сети EtherCAT в топологию звездой с использованием одного канала на входе и до 3 различных каналов на выходе

В таблице ниже указывается максимальное число расширений, конфигурируемых в TRS-CAT

Число расширений TRS-IO-E и TRS-AN-E	Число расширений TRS-AC-E
7	0
5	1
3	2
1	3

Для удаленных модулей типа TRS-AX, при увеличении числа включенных TRS-AX уменьшается число используемых TRS-16 и TRS-IO. Для расчета максимального числа вводимых удаленных модулей типа TRS-16 и TRS-IO необходимо использовать следующую формулу: число других удаленных модулей = $32 - (\text{число TRS-AX} * 4)$. Например, если с одной платой TMSbus связаны 3 TRS-AX, при применении формулы получаем: число других удаленных модулей = $32 - (3 * 4)$, таким образом, могут вводиться макс. 20 удаленных модулей типа TRS-16 и/или TRS-IO.

Выбор положения удаленного модуля должен выполняться на основе адреса, заданного при помощи выключателя на удаленном модуле. Пользоваться документацией на аппаратное обеспечение отдельно взятого удаленного модуля.

При выборе удаленного TRS-AX может потребоваться задать управляемый тип осей.

На схеме ниже указываются какие типологии осей могут быть связаны с различным аппаратным обеспечением.

- | | |
|-----------------------|---|
| • плата AlbMech | цифровые оси |
| • плата DualMech | цифровые оси |
| • плата DualMech Mono | цифровые оси |
| • удаленный TRS-AX | аналоговые оси (если конкретное соединение имеет аналоговый тип), оси подсчета (если конкретное соединение имеет аналоговый тип), пошаговые оси (если конкретное соединение имеет аналоговый тип) |
| • удаленный AlbStep | пошаговые оси |
| • расширение TRS-AC-E | оси подсчета |

На платах MECHATROLINK-II каждой осью можно управлять в режиме управления положением или скоростью (по умолчанию). Тип управления для каждой оси можно изменить в окне «Конфигурация аппаратного обеспечения» в столбце «**Настройки**». Число осей, которое можно сконфигурировать, изменяется в зависимости от значения установленной частоты контроля осей:

Плата	Частота контроля осей (Гц)	Максимальное число сервоприводов
AlbMech	1000	8
AlbMech	≤ 500	16
DualMech Mono	1000	8
DualMech Mono	500	20
DualMech Mono	250	30
DualMech	1000	16
DualMech	500	40
DualMech	250	60

Конфигурировать узел шины CAN

Плата управления шиной

Albatros может управлять устройствами на полевых шинах CAN с помощью плат Тра, оснащенных разъемом шины CAN. Шина CAN присутствует на платах **TMSbus**, **TMSbus+**, **TMSCan+** и **TMSCan**.

Конфигурировать основные данные и услуги

Данные шины CAN определяются в конфигурации оборудования. Выберите шину CAN, параметры которого должны быть определены, и нажмите кнопку **[Правка]**.

Основные данные:

- **Время выборки (TIME)**: время выборки в мсек. Оно не может быть выше 60 000 (60 секунд). По умолчанию 2. На шине S-CAN принимается только значение 2.
- **Время для синхронной связи PDO (TIMEPDO)**: время, выраженное в мсек. Указывает время, выделенное для синхронной связи PDO. Установленное значение не может превышать значение TIME (Это не является обязательным значением).
- **Время ожидания (TIMEAFTERRESET)**: время выражается в мсек. Указывает время ожидания на начальном этапе после программного сброса узлов в сети. Оно не может быть выше 60 000 (60 секунд)

- **Число циклов CAN без ответа (LIFETIMEFACTOR):** это число циклов CAN без ответа на вызов Node Guarding (Охрана Узла), до того, как будет сгенерирована ошибка отключенного узла. Значение не может быть больше 100 и меньше 1. (По умолчанию 3).
- **Скорость передачи:** номер скорости связи CAN в килобитах в секунду (значение может быть 1000, 500, 250, 125, 100).

Службы можно включить или отключить, выбрав или отменив выбор опции, которая относится к службе. Полю **Дополнительно** могут быть присвоены значения, которые имеют значение, установленное производителем станка. Управление этим значением не выполняется. По умолчанию это 16.

Узел CAN

Вставить новый узел

Узел вставляется путем выбора ветки CAN дерева и нажатия кнопки **[Создать]**. Тип узла определяется по типу шины. Если шина имеет тип CAN, узел является узлом ввода / вывода, если шина имеет тип S-CAN, тип вставленной шины - тип сервопривода. В диалоговом окне для входа узла необходимо выбрать:

- **Положение:** это номер узла (от 1 и далее).
- **AutoOp:** если выбрано, устройство разрешает автоматический переход в Рабочее состояние после переподключения.

Конфигурировать узел

Для каждого узла можно определить PDO передачи и PDO приема. Если шиной является CanOpen, для плат TMSBus и TMSBus+ можно определить максимум 8 PDO, для плат TMSCan и TMSCan + можно определить максимум 4. Выберите узел в дереве и нажмите кнопку **[Создать]**. Данные, которые будут установлены в диалоговом окне:

- **Тип PDO:** выберите **Передача** или **Прием**, в зависимости от того, что определяете ли вы TPDO или RPDO.
- **Размер:** размер PDO приема или передачи.
- **COB-ID:** значение, которое можно определить только на платах TMSBus и TMSBus +. Значение отображается и сохраняется в десятичном виде. Чтобы отобразить значение в шестнадцатеричном формате, необходимо выбрать флажок **Шестнадцатеричный**, расположенный рядом со значением.
- **Асинхронный:** если он включен, асинхронные PDO настраиваются, т.е. они не обновляются при каждом цикле. Эта опция доступна только для плат TMSCan и TMSCan +. Прием асинхронных PDO должен выполняться в коде GPL с использованием инструкции [RECEIVEPDO](#)

Характеристики управления EtherCAT в Albatros

Режим связи - всегда DC-Synchronous (Синхронный). Первый узел сети предоставляет тактовый сигнал, поэтому очень важно, чтобы этот узел обеспечивал точный надежный сигнал, как он подается, например, от TRS-CAT. Невозможно использовать другие режимы, например, Free-Run. Реализованы следующие протоколы: CoE (CAN application protocol over EtherCAT) и EoE (Ethernet over EtherCAT). В CoE реализованы устройства по DS401 и DS402 с рабочим режимом управления осью по умолчанию *режима циклической синхронной скорости*.

Максимальное число узлов EtherCAT - 200.

Вступление

С каждым физическим устройством EtherCAT связан файл ESI (EtherCAT Slave Information), описывающий характеристики и функции устройства. Файл имеет формат XML. Для каждого устройства должен существовать только один файл ESI. Как правило, файлы ESI можно скачать с интернет-сайта производителя. Albatros ищет эти файлы в папке, определенной в Tra.ini, в разделе [tra] в пункте DirESIFiles. Умолчание - подпапка "\\EtherCAT" SYSTEM.

Из файлов ESI Albatros получается информация об устройстве, анализируя все элементы "\Devices/Device/Type". Каждое устройство определяется следующими параметрами: Vendor ID, Product ID и Revision Number.

Все также из файлов ESI можно получить информацию о расширениях (называемых также модулями) устройств. Albatros находит информацию по типу расширений, отыскивая в файле ESI устройства элементы "Modules/Module".

Конфигурация аппаратного обеспечения EtherCAT

Аппаратное обеспечение конфигурируется путем описания главных плат, а для каждой платы - перечня физических устройств, подключенных по шине к этой плате. Физические устройства называются также "узлы" полевой шины. Для EtherCAT главная плата - это не определенная плата управления шиной, а используется сетевое подключение модуля.

Что касается локальных модулей, сетевое подключение должно быть из реализованных RTX, а для удаленных модулей используется определенное сетевое подключение модуля из предусмотренных Windows CE 6.0. Для каждого модуля, локального или удаленного, можно сконфигурировать только одну главную плату.

Главная плата должна быть вставлена в аппаратной конфигурации, выбрав ветвь Модуль и нажав кнопку **[Создать]**.

Узел EtherCAT

Для вставки узла, выберите ветку платы дерева и нажмите кнопку **[Создать]**. Индексы узла должны быть последовательными, поэтому для каждой команды **[Создать]** узел добавляется к остальным. Для каждого узла должны быть определены следующие данные:

- **Имя устройства:** выберите имя устройства из перечисленных. Отображаются только устройства с самой последней ревизией; если вы хотите увидеть все ревизии, вам нужно выбрать пункт **Отобразить изменения**.
- **Режим оси:** выбрать режим работы, который будет использоваться для узлов типа привода, как указано в регламенте DS402 по отношению к объекту 6060₁₆ «режим работы». Выбор между **Циклическим синхронным положением** и **Циклической синхронной скоростью**. Последняя по умолчанию.
- **Заставить как узел ввода-вывода:** если включено, заставить числовое управление рассматривать привод как узел ввода/вывода. Этот атрибут применяется только к узлам, которые поддерживают DS402 (сервоприводы).

Режим оси - это единственный параметр, который можно изменить после вставки узла.

Узел EtherCAT можно удалять, перемещать, отключать и копировать. Команды доступны в контекстном меню каждой ветви, в главном меню Albatros и в некоторых кнопках в окне конфигурации оборудования. Команда **Удалить** удаляет выбранный узел, все его расширения и его подключения виртуальное-физическое, а все последующие узлы перемещаются на предыдущий адрес.

Команда **Переместить** перемещает выбранный узел на новый адрес. Узлы, следующие за выбранным адресом, также перемещаются на одну позицию.

Команда **Копировать** сохраняет данные выбранного узла для использования в последующей команде Вставить.

Команда **Вставить** вставляет ранее скопированный узел в дерево устройств. Узел вставляется в положение, выбранное пользователем. Все следующие узлы перемещены.

Расширение EtherCAT

Для ввода расширения выберите узел, к которому вы хотите добавить расширение, и нажмите кнопку **[Создать]**. Индексы расширений должны быть последовательными, поэтому расширение в конце остальных добавляется к каждой команде **[Создать]**. Объекты PDO расширений не доступны для редактирования.

Описание PDO

Можно определить до 16 PDO, отправленных узлом (TxPDO) и до 16 PDO, полученных узлом (RxPDO). Каждый RxPDO описывает только один PDO, который узел получает от ведущего, затем цифровые и аналоговые выходы для узлов ввода / вывода или целевой скорости e controlword для узлов оси. Каждый TxPDO описывает только один PDO, который узел отправляет ведущему, затем цифровые и аналоговые входы для узлов ввода / вывода или текущего положения и слово состояния для узлов оси.

Для списка и описания PDO и сопоставляемых объектов на PDO, обратитесь к документации конкретного устройства EtherCAT и его файлу ESI.

Существует три способа описания PDO узла:

- Не указывайте никакой PDO. Таким образом, числовое управление использует PDO, сконфигурированные по умолчанию в устройстве.
- Укажите только PDO без предоставления списка объектов. Используется, когда CN имеет несколько альтернативных и непрограммируемых PDO.
- Полностью опишите PDO, указав объект связи и список объектов для отображения. Этот режим обеспечивает максимальный контроль над информацией, которая отправляется и принимается узлом.

Каждый объект описывается своим индексом в словаре объектов CN, за которым может следовать субиндекс.

Если субиндекс отсутствует, он считается равным 0.

Объект словаря (object dictionary) - это сердце каждого устройства. Обеспечивает доступ ко всем типам данных устройства, параметрам связи, конфигурации и обработки данных.

Внимание! Не все объекты словаря объектов могут быть отображены в PDO.

Модификация PDO привода

Что касается узлов типа сервопривода, для каждой оси имеется PDO, так что еще один TxPDO и еще один RxPDO узла ссылаются на еще одну ось узла. Первые два объекта каждого RxPDO и TxPDO имеют заранее заданное значение и размерность. В следующей таблице описано, как настроить PDO, связанные с осями того же узла, управляемые в режиме оси **Циклической синхронной скорости**.

Привод	RxPDO		TxPDO	
	1-й объект 16 бит ControlWord	2-й объект 32 бит Целевая скорость	1-й объект 16 бит StatusWord	2-й объект 32 бит Фактическое положение
1-я ось	6040 ₁₆	60FF ₁₆	6041 ₁₆	6064 ₁₆
2-я ось	6840 ₁₆	68FF ₁₆	6841 ₁₆	6864 ₁₆
Еще одна ось	Добавьте 800 ₁₆ к каждому объекту оси, которая предшествует ему.			

В следующей таблице описано, как настроить PDO, связанные с осями одного и того же узла, управляемые в режиме оси **Циклического синхронного положения**.

Привод	RxPDO		TxPDO	
	1-й объект 16 бит ControlWord	2-й объект 32 бит Целевое положение	1-й объект 16 бит StatusWord	2-й объект 32 бит Фактическое положение
1-я ось	6040 ₁₆	607A ₁₆	6041 ₁₆	6064 ₁₆
2-я ось	6840 ₁₆	687A ₁₆	6841 ₁₆	6864 ₁₆
Еще одна ось	Добавьте 800 ₁₆ к каждому объекту оси, которая предшествует ему.			

Для изменения объектов оси, выберите ось в дереве и нажмите кнопку **[Правка]**. В диалоговом окне показываются индекс PDO передачи, индекс PDO приема и список сконфигурированных объектов. В частности, данные в диалоговом окне:

- **Индекс PDO передачи:** это значение PDO передачи оси. Это не редактируемое значение.
- **Объекты:** список объектов PDO передачи. Первые два объекта устанавливаются на основе выбранного режима оси.
- **Список объектов:** список объектов PDO передачи, которые могут быть использованы. Для каждого объекта в дополнение к индексу определяются подиндекс и описание.
- **Индекс PDO приема:** это значение PDO приема оси. Это не редактируемое значение.
- **Объекты:** список объектов PDO приема. Первые два объекта устанавливаются на основе выбранного режима оси.
- **Список объектов:** список объектов PDO приема, которые можно использовать. Для каждого объекта в дополнение к индексу определяются подиндекс и описание.

Кнопки **[Вверх]** и **[Вниз]** используются для изменения порядка объектов в списке окна **Объекты**.

Кнопка **[Добавить]** позволяет вставить объект, которого нет в **Списке объектов**.

Для изменения значения объекта, выберите объект и дважды щелкните мышью. Синтаксис для определения нового объекта:

- **номер объекта:** номер объекта в шестнадцатеричном формате.
- **подиндекс:** номер подиндекса. Он должен быть отделен от номера объекта символом «.» (точка). Это необязательное значение, которое, если не определено, принимается равным 0 в качестве значения по умолчанию. Это в десятичном формате.
- **L:** размер объекта в битах. Должно быть кратно 8.

Отсутствующие значения берутся из данных объекта словаря или им присваиваются значения по умолчанию.

Пример:
 объект также с подиндексом и длиной: 1600.1L16
 объект без подиндекса: 1601L24
 простой объект: 1603

Эти значения могут быть прочитаны GPL через инструкцию GETAXIS, на которую мы ссылаемся. Также возможно отслеживать дополнительные объекты как из окна калибровки, так и из осциллографа.

В более общем смысле из GPL можно считывать и записывать конкретные объекты в PDO посредством инструкций [GETPDO](#) и [SETPDO](#), на которые делается ссылка.

Дополнительные PDO

Для определения дополнительных PDO передачи и приема, выберите узел в дереве и затем нажмите кнопку **[Правка]**.

В диалоговом окне для добавления PDO передачи, нажмите кнопку **[Дополнительные TPDO]**, в то время как для добавления PDO приема, нажмите кнопку **[Дополнительные RPDO]**.

Диалоговое окно для настройки дополнительного PDO аналогично диалоговому окну для вставки объектов.

В этом окне вы можете выбрать индекс PDO и объекты, которые будут связаны. Если в PDO есть обязательные и не редактируемые объекты, они отображаются и все кнопки редактирования отключены. Новый PDO добавляется в дерево в конце приводов. Для редактирования данных, выберите и нажмите кнопку **[Правка]**. Для удаления дополнительного PDO, выберите этот PDO в дереве, а затем нажмите кнопку **[Удалить]**.

Автоматическое получение узлов EtherCAT

Если шина EtherCAT присутствует в конфигурации оборудования, можно получить подключенные узлы из сети, используя команду **Автоматическое получение узлов**, которую можно активировать из контекстного меню.

Для выполнения команды данные конфигурации аппаратного обеспечения и данные на числовом управлении должны быть выровнены. И файлы ESI, которые описывают узел в сети, должны присутствовать в папке, определенной в Tra.ini в разделе [tra] под записью DirESIFiles.

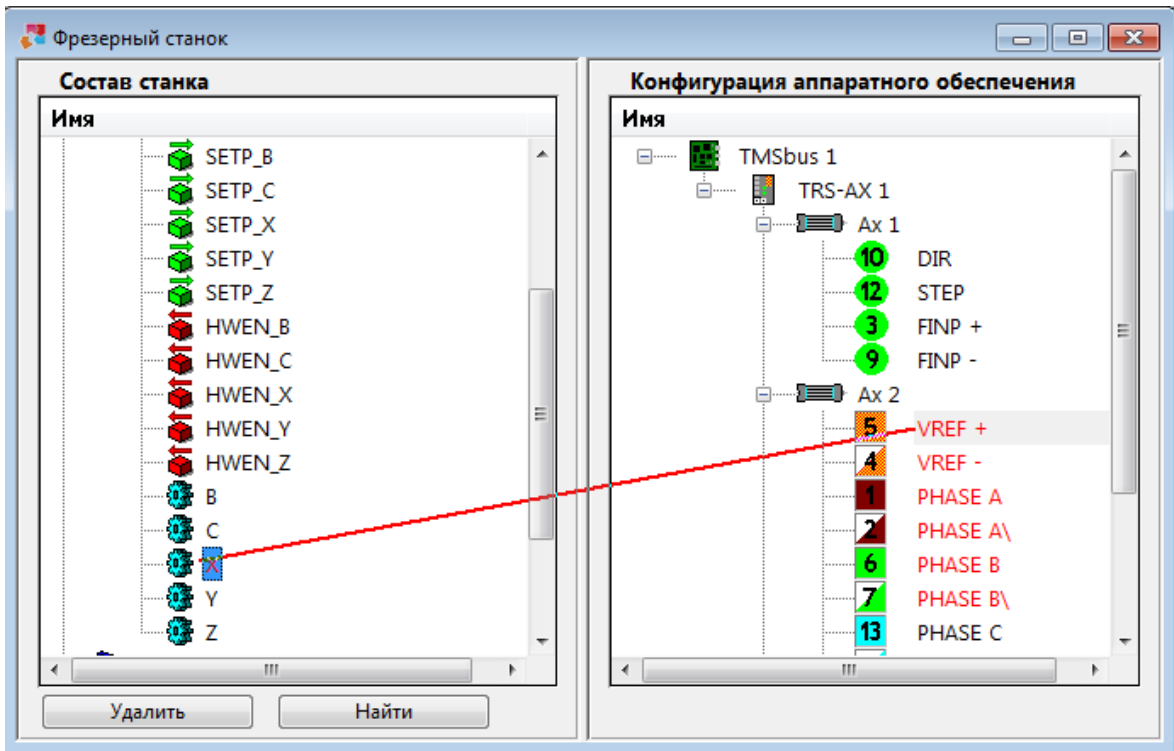
Получение узлов EtherCAT из сети следует перечисленным правилам:

- в конфигурации оборудования ни один узел не настроен: данные, относящиеся к узлам, считываются из сети и отображаются в конфигурации
- Узлы EtherCAT уже были вставлены в аппаратную конфигурацию:
 - если узел, считываемый из сети, имеет тот же адрес, тот же ID Поставщика, тот же ID Продукта и тот же Номер Пересмотра, что и сконфигурированный узел, тогда сконфигурированный узел со всеми его PDO сохраняется
 - если узел, считываемый из сети, имеет тот же адрес, что и сконфигурированный узел, но другой ID Поставщика или другой ID Продукта, затем узел конфигурации удаляется вместе с его физическим- виртуальным и заменяется сетевым узлом. Если сетевой узел является приводом, PDO считываются из файла ESI, в противном случае они считываются из сетевого узла.
 - если узел, считываемый из сети, имеет тот же адрес, тот же ID Поставщика, тот же ID Продукта а другой Номер Пересмотра, что и сконфигурированный узел, тогда заменяется только Номер Пересмотра а PDO сохраняются.
 - если узел, определенный в конфигурации, не имеет соответствующего узла в сети EtherCAT, который занимает тот же адрес, то он отключается в конфигурации и поддерживается виртуальное-физическое.

7.4.3 Конфигурация Виртуальное-физическое

Конфигурация Виртуальное-физическое - это последний этап конфигурации, состоящий в присвоении логических устройств компонентам аппаратного обеспечения.

При открытии Конфигурация Виртуальное-физическое открываются два окна: слева - окно "Состав станка" (виртуальное), справа - "Конфигурация аппаратного обеспечения" (физическое). В обоих окнах в графической форме и с древовидной структурой представлены все образующие систему элементы.



Конфигурация Виртуальное-физическое

Уже выполненные подключения виртуальное-физическое отмечены в окне "Состав станка" красным цветом имени устройств, а в окне "Конфигурация аппаратного обеспечения" - именем типа сигнала, следующим за номером зажима, также выделенным красным цветом.

Для каждой цифровой оси платы МЕCHATROLINK-II в виртуальном-физическом режиме могут конфигурироваться 6 входов и 1 цифровой выход. См. подробное описание в главе **Язык GPL->Инструкции->МЕCHATROLINK-II->MECGETSTATUS**.

Если в модуле имеется шина EtherCAT, в любом случае можно сконфигурировать платы для шины МЕCHATROLINK-II, но с некоторыми ограничениями: с реальным временем 1 мс нельзя подключить более шести осей МЕCHATROLINK-II (для каждой шины); с реальным временем 2 мс предел повышается до 16 осей.

Еще неподключенные устройства и зажимы представлены черным цветом. Устройства, принадлежащие к отключенной группе, чья виртуальная физическая конфигурация была сохранена, показаны серым.

Перед всеми сигналами, относящимися к осям, в окне "Конфигурация аппаратного обеспечения" представлен прямоугольник цвета, соответствующего цвету оболочки провода внутри соединительного кабеля.

Можно выделить подключение, выбирая логическое устройство (или аппаратный компонент) и нажимая клавишу пробела, при этом подключение представляется как красная линия, соединяющая устройство с аппаратным компонентом. Кроме того, можно всегда поддерживать активным отображение подключений при помощи команды **[ALT+ВВОД]**.

Для отображения логического устройства, подключенного к аппаратному компоненту, необходимо выбрать аппаратный компонент и два раза щелкнуть мышью.

Для выбора соединяемых логического устройства и физического устройства можно использовать различные процедуры:

первая процедура

- Отобразить на экране в окне "Конфигурация аппаратного обеспечения" физический зажим, к которому вы желаете подключить устройство
- Выбрать или поставить указатель мыши на нужное логическое устройство в окне "Состав станка"

вторая процедура

- Выбрать или поставить указатель мыши на нужное виртуальное устройство в окне "Состав станка"
- Выбрать команду из меню **Правка->Найти соотв. физич. устройство** или нажать сочетание клавиш **[CTRL+Пробел]**. Автоматически Albaros отображает в окне "Конфигурация аппаратного обеспечения" первое свободное физическое устройство, с которым может быть соединено логическое устройство.

третья возможная процедура

- Выбрать или поставить указатель мыши на нужное виртуальное устройство в окне "Состав станка"
- Выбрать команду из меню **Правка->Найти след. неподключ. устройство** или нажать сочетание клавиш **[CTRL+Цифр. кл.+]** или команду **Правка->Найти предыдущее неподключ. устройство** или нажать сочетание клавиш **[CTRL+Цифр. кл.-]**.

Для выполнения подключения между двумя wybranными устройствами:

- На подключаемом логическом устройстве нажать левую кнопку мыши и, удерживая ее нажатой, переместить мышь в направлении выбранного зажима. Появится красная строка, показывающая выполнение подключения. После того, как вы дойдете до строки зажима, отпустить кнопку для завершения операции или
- выбрать команду **Подключить!** из меню **Правка** или нажать сочетание клавиш **[CTRL+L]**.

Для удаления подключения необходимо выбрать соответствующее устройство или компонент и нажать кнопку **[Удалить]** или кнопку **[Отмена]** на клавиатуре.

7.4.4 Карты электромонтажа

После выполнения подключения между виртуальными устройствами и соответствующими физическими устройствами можно распечатать карты или списки, в которых приводится связь между физическими и виртуальными устройствами.

Для выполнения этой операции необходимо, чтобы в системе был установлен MS-Word (версия 6 или более новая), Albatros использует его функции для форматирования карт. Кроме того, система должна правильно конфигурироваться. Это значит, что в системе должны присутствовать файлы-шаблоны, используемые для компиляции карт. Это серия файлов с расширением ".doc", которые обычно находятся в папке System или другой инсталляционной папке (обычно - папке "Карты"). Важно, чтобы папка, в которой находятся эти файлы, соответствовала папке, указанной в файле tra.ini с ключом "DirMaps". Например:

```
[TRA]
DirMaps=C:\Albatros\Карты
```

Для печати карт электромонтажа необходимо выбрать любой аппаратный компонент, который находится в правом окне [конфигурации Виртуальное-Физическое](#) или который находится в окне [Конфигурации аппаратного обеспечения](#).

При нажатии кнопки "Печать" панели инструментов или при выборе пункта **Печать** меню **Файл** отображается обычное окно выбора опций печати. После выполнения нужного выбора при подтверждении нажатием кнопки **[ОК]** открывается окно со списком аппаратных компонентов, имеющих в конфигурации.

В этом окне можно выбрать компоненты, для которых мы желаем распечатать карту электромонтажа. Наконец, при нажатии кнопки **[ОК]** выполняется печать карт электромонтажа. При отмене выбора опции **Печать на бумаге** карты будут сохранены как документы MS-Word в папке карты + имя текущего модуля (Mod.0, и т.д.).

Учитывая большое число страниц, которые могут быть направлены в печать, рекомендуется выполнить проверку, печатая карту только одного аппаратного компонента и проверяя, что все в порядке. Если вместо карт печатается список логических устройств, возможно, не был выбран компонент (например, плата оси или удаленный модуль) в окне, относящемся к аппаратному обеспечению.

7.5 Список клавиш для навигации по древовидной структуре

Клавиша	Описание
Стрелка Вверх	перемещает выбор на строку, непосредственно до или после данной
Стрелка Вниз	
Стрелка Вправо	разворачивает на один уровень выбранную ветвь, а если она уже развернута, перемещает выбор на следующую ветвь
Стрелка Влево	закрывает выбранную ветвь или, если она уже закрыта, перемещает выбор на предыдущую ветвь
+	разворачивает на один уровень выбранную ветвь
-	закрывает выбранную ветвь
*	разворачивает все уровни выбранной ветви

8 Средства разработки

8.1 Редактор GPL

8.1.1 Функции редактора GPL

Редактор GPL - это инструмент, позволяющий создавать и менять файлы, содержащие код GPL Albatros. Эта функция активна только при уровне пароля, равном или выше пароля изготовителя. Каждому файлу функций присваивается информация, которую можно отобразить в меню **Файл->Информация**.

Рабочие режимы - такие же, как в обычном текстовом редакторе, поэтому здесь есть такие команды, как **Копировать**, **Вставить**, **Найти**, **Заменить** и т. д. Все эти команды выбираются из меню **Правка**.

Отмена

Когда возможно, позволяет отменить последнюю выполненную операцию. Восстанавливается ситуация, существовавшая до последней выполненной операции.

Повторить

Восстанавливает ситуацию, существовавшую до последней команды Отмена

Вырезать

Выбранные текст или данные удаляются и копируются во временную память для обеспечения возможной вставки командой **Вставить**.

Копировать

Выбранный текст или элемент копируется во временную память для обеспечения возможной вставки командой **Вставить**.

Вставить

Содержимое временной памяти вводится по критериям, зависящим от активной функции.

Удалить

Удаляет текст или строки или выбранный элемент. Все удаленное можно вернуть, немедленно выбирая команду **Отмена**.

Выбрать все

Позволяет выбирать весь текст активного файла. К выбранным строкам могут применяться команды Копировать, Вырезать, Вставить.

Найти...

Поиск документа в текущем тексте. Можно задать критерии для использования на этапе поиска: направление поиска, и различие между прописными и строчными буквами, поиск целого слова и поиск через регулярные выражения. Позволяет повторить предыдущий поиск, давая возможность изменить критерии поиска, заданные командой Найти.

Найти следующий

Позволяет повторить предыдущий поиск, давая возможность изменить критерии поиска, заданные командой Найти.

Заменить

Позволяет искать текст в текущем документе и заменять его другим текстом.

Вставить устройство

Вставляет устройство, выбирая его из списка устройств. Эта функция оказывается особенно удобной при работе с большим числом устройств, когда может быть сложно запомнить их имена. Используются только устройства текущего модуля, которые могут вызываться, и все общедоступные устройства других модулей

Вставить функцию


Вставляет функцию или часть функции, начиная с положения курсора, считывая ее из файла-шаблона, написанного изготовителем станка. Можно записать несколько файлов-шаблонов. Файл-шаблон - это текстовый файл, имя которого должно начинаться с префикса GPL и иметь расширение TXT. Он должен записываться в папку, в которой архивированы библиотеки (обычно - system\lib). При определении нескольких файлов-шаблонов при выборе этой команды открывается диалоговое окно, в котором представляется список имен файлов-шаблонов без префикса и без расширения. Файлы-шаблоны могут содержать, например, определения общепотребительных констант, функции управления ошибками системы, общие функции, коды, реализующие общепотребительные алгоритмы и т. д. Могут содержать также комментарии.

Файл-шаблон может создаваться путем сохранения выбранного текста в файле функций GPL. Эта команда доступна только как клавиша-ускоритель [CTRL+SHIFT+C]. Открывается диалоговое окно для ввода имени, которое должно быть присвоено фрагменту кода.

Вставить сообщение

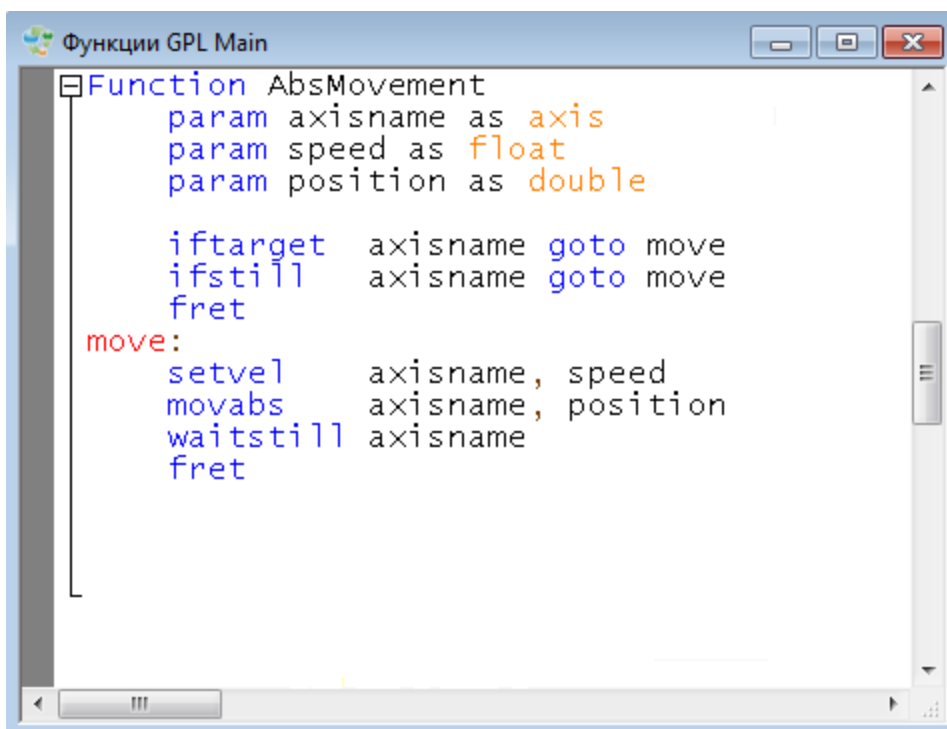
Вставляет в текст GPL цифровой код, ассоциированный с выбранным сообщением. Позволяет вводить новые сообщения в файл языка.

Включить/отключить пропуск страницы

Вводит или удаляет пропуск страницы . Пропуск страницы может использоваться в качестве закладки для перехода к нужным положениям в файле функций.

Перейти к следующему пропуску страницы **Перейти к предыдущему пропуску страницы**

Перемещает курсор правки на строку перехода на следующую страницу относительно той, на которой он расположен
Перемещает курсор правки на строку перехода на предыдущую страницу относительно той, на которой он расположен



Редактор GPL

Проверка правильности синтаксиса выполняется на этапе архивирования, когда текст также компилируется. Однако программист может выполнить предварительный визуальный контроль, т. к. текст отображается разным цветом в зависимости от того, что он представляет. Например, инструкции представляются синим цветом, комментарии - зеленым, ярлыки - красным.

Можно изменить значение табуляторов из меню **Опции->Табуляторы....** Можно определить два типа табуляторов:

- абсолютные табуляторы: задают начальное положение для инструкций кода GPL, начальное положение первого аргумента инструкций и начальное положение для комментария .
 - относительный табулятор (пробелы): задает, сколько пробелов соответствует один табулятор
- Табуляторы помогают сделать более понятным представление кода GPL.

Для каждой инструкции или ключевого слова имеется Справка онлайн для помощи при написании функции. Для вызова справки достаточно установить курсор на инструкцию и нажать кнопку **[F1]**.

На каждой строке текста можно написать только одну инструкцию. Можно продолжать инструкцию на следующей строке, вводя символ '_' (с пробелом перед ним) в качестве последнего символа строки. Это позволяет вводить комментарии посреди инструкции:

```

Message
  1000 ;код отображаемого сообщения
  3 ;ячейка обзорного экрана, в котором будет отображаться [ВВОД].

```

Использование регулярных выражений

Регулярные выражения можно использовать в окне **Найти** и в окне **Заменить**. Регулярное выражение - это последовательность числовых и буквенно-цифровых символов, используемых для поиска и замены частей текста в более крупном тексте. Albatros использует грамматику регулярных выражений ECMAScript. В этом разделе описываются основные способы использования регулярных выражений в Albatros.

Самое простое регулярное выражение - это то, которое представлено одним символом. Существуют исключения, представленные следующими символами:

- **.** (**точка**): точка находит любой символ. Например, «А..» находит все вхождения заглавной буквы А, за которой следуют любые два символа.
- **[]** (**квадратные скобки**): квадратные скобки позволяют указать список символов внутри них. Текст ищет вхождения для каждого из присутствующих символов. Если первым символом является **^** (**циркумфлекс**), будет выполняться поиск всех символов, кроме перечисленных в квадратных скобках.

Например:

[<>]: поиск всех вхождений символа < и символа >.

[.] AX: поиск всех вхождений, имеющих строку "AX".

[a-d]: поиск всех вхождений символов a, b, c и d. Тире указывает на набор символов.

[\ [\]]: выполняется поиск всех вхождений символа [и символа].

[^ +]: ищутся все символы, кроме символа +.

- ***** (**звездочка**): ищутся все вхождения символа (или групп символов), найденные до **звездочки**. Звездочка влияет только на предшествующий ему символ: чтобы заставить его действовать на группу символов, необходимо использовать круглые скобки.

Например:

; - *: поиск всех вхождений символов ; и ; - и ; ----- .

- **+** (**знак плюс**): поиск одного или нескольких вхождений предыдущего символа. Он отличается от использования звездочки, поскольку всегда должен присутствовать символ, предшествующий знаку +. Знак плюс действует только на предшествующий ему символ: чтобы заставить его действовать на группу символов, необходимо использовать круглые скобки. Давайте возьмем пример, используемый для символа звездочки:

; - +: ищутся все вхождения символов ; - и ; --- . Единственный символ не ищется ; (**точка с запятой**) .

- **?** (**вопросительный знак**): делает предшествующий ему символ необязательным, поэтому он может присутствовать не более одного раза.

Например:

Setfeedi?: Все вхождения слова Setfeed и слово Setfeedi.

- **{ }** (**фигурные скобки**): указывают, сколько раз символ (или группа символов) должен присутствовать в тексте.

Например:

ee {2}: все вхождения двух последовательных ee ищутся.

- **^**. (**циркумфлекс и точка**): находит первый символ каждой строки.

- **^** (**циркумфлекс**): находит искомое слово, только если он находится в начале строки

- **|** (**труба**): находит термины, которые появляются как до, так и после символа «|».

Например:

Send | Const: ищутся все вхождения слова Send и слово Const.

- **** (**обратная косая черта**): обратная косая черта имеет двойную функцию:

1) преобразовать нормальный символ в функцию.

\ b найти передний или задний край слова

\ B найти слово, кроме переднего края

\ d найти только цифры

\ D найти все, кроме чисел

\ s найти место

\ S найти все, кроме места

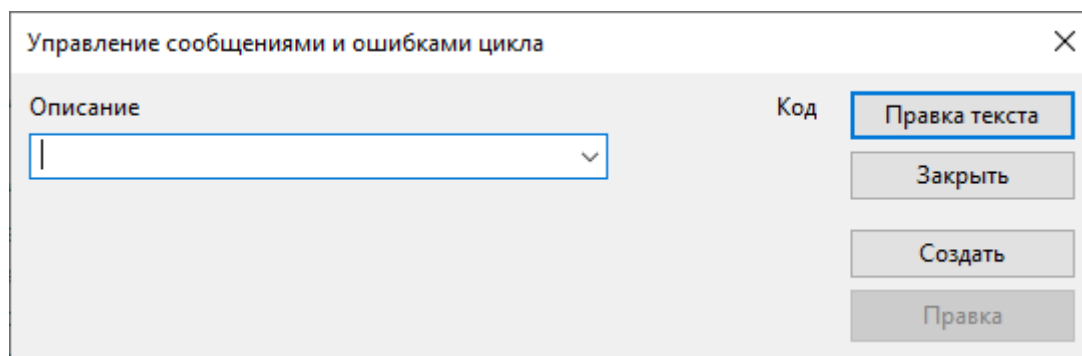
Примеры:

\ bi находит все слова в тексте, начинающиеся с буквы i.

2) превратить специальный символ в нормальный символ. Например, чтобы найти в тексте специальный символ **^** (**циркумфлекс**), просто поставьте перед ним обратную косую черту.

8.1.2 Вставить сообщение

В Albatros существуют сообщения двух типов: сообщения модуля и сообщения группы. Эту команду можно выбрать из меню **Правка->Вставить сообщение**. Сообщения группы вводятся непосредственно в редактор при написании кода GPL при помощи инструкции **DEFMSG**. Они видны и могут использоваться только внутри группы, в которой они определяются, поэтому можно иметь одно определение сообщения в нескольких группах без какого-либо наложения. В отличие от сообщений группы сообщения модуля могут использоваться всеми группами. Они могут вводиться при помощи диалогового окна, позволяющего вызывать сообщение, уже присутствующее в файле языка, или вводить новое.



Окно управления сообщениями

Этим образом не требуется изменить файл. Сообщение будет вводиться на текущем языке и затем должно переводиться на другие языки (с использованием программы TraLangs). Все сообщения в файле языка перечисляются в пункте **Описание**. Для ввода сообщения в функцию необходимо, после выбора текста, нажать кнопку **[Правка текста]**. Для изменения существующего сообщения **[Правка]** или создания нового **[Создать]**, прежде всего необходимо ввести изменение или новый текст, а затем нажать соответствующую кнопку.

8.1.3 Криптография

В Albatros при помощи криптографии можно сделать невозможным отображение исходного текста функций.

Подключение криптографии осуществляется путем задания в tra.ini пункта Tele+= 1. По умолчанию задано значение 0, что означает, что шифрование не включено.

При сохранении файла функций и при подключенной криптографии отображается следующее сообщение: "Желаете криптографировать файл?" При ответе [Да] файл криптографирован. Можно будет криптографировать файл, сохраненный ранее без криптографии, а криптографированный файл останется всегда таким.

С подключенной криптографией с подключенной криптографией, если пароль - ежедневного типа изготовителя, незашифрованный файл не криптографируется. Файл криптографированных функций может отображаться или меняться в Albatros только пользователем, который сохранил файл ранее. Владелец файла зашифрованных функций не может меняться!

Если необходимо убрать криптографию с файла, необходимо использовать внешнюю программу, которая называется SBIANCA и находится в папке Bin Albatros.

Одним из способов использования программы Sbianca является графический интерфейс. Вы выбираете файлы, которые хотите расшифровать. Для каждого файла отображаются свойства **Состояние** и **Учетные данные**. Состояние может быть **Обычный текст**, если файл не зашифрован, или **Зашифрованные**, если файлы зашифрованы.

Свойства **Учетные данные** дают информацию о видимости файла: **Свободно читаемый** означает, что файл может отображаться паролем уровня, на котором мы находимся, в противном случае **Блокированный** означает, что файл не может отображаться. После выбора файлов необходимо щелкнуть на **[Расшифровать!]** для их расшифрования или **[Зашифровать]** для их зашифрования.

Второй способ использования - из командной строки. Команды и файлы передаются программе в качестве аргументов. Если аргументов нет, режим использования активируется через графический интерфейс. Синтаксис:

```
[-l|-e|-d] file1 file2 ...
```

При:

- -l или без опции для просмотра статуса каждого файла
- -d для расшифрования файлов
- -e для зашифрования файлов

В конце выполнения программа выводит результаты запрошенной операции. Вывод программы в формате уценки (markdown).

8.1.4 Список горячих клавиш

Удаление текста

Клавиша

Возврат
CTRL+Возврат
Del
CTRL+T
CTRL+Del

Описание

удаляет один символ слева или удаляет выбранный текст
удаляет слово слева
удаляет один символ справа или удаляет выбранный текст
удаляет слова или пробелы справа
удаляет слово справа и все последующие пробелы до начала нового слова

Комментарий из нескольких текстовых строк

Клавиша

CTRL+';'. На итальянских клавиатурах нажимается также клавиша [SHIFT]

Описание

добавляет или удаляет символ комментария с выбранных строк.

Позиционирование курсора

Клавиша

Стрелка вверх
Стрелка вниз
Стрелка вправо
Стрелка влево
Home
End
CTRL+Home
CTRL+End
CTRL+Стрелка влево
CTRL+Стрелка вправо
CTRL+ВВОД

Описание

перемещает курсор в выбранном направлении

перемещает курсор в начало строки и на первый символ строки
перемещает курсор в конец строки
перемещает курсор в начало документа
перемещает курсор в конец документа
перемещает курсор на одно слово влево
перемещает курсор на одно слово вправо
перемещает курсор на первый символ следующей строки

Выбор

Клавиша

SHIFT+Home
CTRL+SHIFT+Home
CTRL+SHIFT+End
CTRL+SHIFT+Стрелка влево
CTRL+SHIFT+Стрелка вправо
SHIFT+Страница вверх
SHIFT+Страница вниз
CTRL+W
CTRL+A

Описание

выбирает до начала строки из положения курсора
выбирает до начала документа из положения курсора
выбирает до конца документа из положения курсора
выбирает слово или пробелы слева от курсора
выбирает слово или пробелы справа от курсора
выбирает одну страницу вверх из текущего положения курсора
выбирает одну страницу вниз из текущего положения курсора
выбирает слово, на котором установлен курсор
выбирает весь документ

Выбор прямоугольником

Клавиша

ALT+
SHIFT+Стрелка вверх
SHIFT+Стрелка вниз
SHIFT+Стрелка влево
SHIFT+Стрелка вправо
или
ALT+выбор мышью

Описание

выбирает прямоугольный блок кода

Табуляторы

Клавиша

Описание

Tab	если нет выбранного текста, вводит пробел между символами, определенный в Опции->Табуляторы . При выборе нескольких строк вводит справа пробел, заданный для соответствующего табулятора.
SHIFT+Tab	если нет выбранного текста, перемещает курсор слева от пробела, определенного в Опции->Табуляторы . При выборе одной или нескольких строк они перемещаются влево от пробела, заданного для соответствующего табулятора.
Копирование и вставка	
Клавиша CTRL+C CTRL+Ins CTRL+X SHIFT+Del CTRL+V SHIFT+Ins CTRL+Y	Описание копирует выбранный текст в буфер обмена удаляет выбранный текст и копирует его в буфер обмена вставляет содержимое буфера начиная с положения курсора удаляет строку, на которой позиционирован курсор, и копирует ее содержимое в буфер обмена
Буксировка (мышью)	буксирует выбранный текст и при отпускании перемещает его в новое положение
CTRL+Буксировка (мышью)	буксирует выбранный текст и при отпускании копирует его в новое положение
Отмена / Восстановление	
Клавиша CTRL+Z ALT+Возврат CTRL+SHIFT+Z	Описание отменяет последние введенные данные восстанавливает последние введенные данные
Поиск и замена	
Клавиша CTRL+F3	Описание осуществляет поиск вниз во всем документе слова, на котором установлен курсор
CTRL+SHIFT+F3	осуществляет поиск вверх во всем документе слова, на котором установлен курсор
F3	поиск следующего случая употребления. Диалоговое окно Найти должно быть закрыто.
SHIFT+F3	поиск предыдущего случая употребления. Диалоговое окно Найти должно быть закрыто.
ALT+F3	открывает диалоговое окно Найти и задает в качестве искомого текста слово, на которое установлен курсор
Отображение ошибок компиляции	
Клавиша Двойной щелчок на ошибке	Описание позиционирует курсор на строке функции GPL, на которой произошла описанная ошибка
F4	позиционирует курсор на строку функции GPL, на которой произошла ошибка, следующая за последней выбранной ошибкой
SHIFT+F4	позиционирует курсор на строку функции GPL, на которой произошла ошибка, предшествующая последней выбранной ошибке.
Создание файла-шаблона	
Клавиша CTRL+SHIFT+C	Описание сохраняет выбранный текст в файле функций GPL. Открывается диалоговое окно для ввода имени, которое должно быть присвоено фрагменту кода.
Управление свертыванием	
Клавиша CTRL+M	Описание Открывает или закрывает выбранное свертывание.

8.2 Библиотеки

Библиотека представляет собой набор функций GPL, которые могут вызываться внутри пользовательского кода GPL и не быть связаны с какой-либо особой конфигурацией. Библиотеки очень полезны, т. к. могут легко копироваться с одного станка на другой. При этом можно избежать необходимости заново писать общий код при внедрении новых станков. Например, можно создать библиотеку геометрических и математических функций.

Файлы библиотеки архивируются в папке system\lib. Они компилируются при выполнении одной из следующих команд: **ЧПУ->Инициализация, Файл->Компилировать все, Сохранить** файл библиотеки или файл глобальных переменных.

Если в коде GPL станка присваивается имя функции или переменной, уже существующей в библиотеке, на этапе компиляции будет всегда иметь преимущество имя станка. Если одно и то же имя существует в двух различных библиотеках, для обозначения нужного вам в коде GPL рекомендуется использовать следующий расширенный синтаксис: **имябиблиотеки.имяфункции**. Например, если функция **ШиринаСегмента** существует как в библиотеке LIBGEO, так и в библиотеке LIBMAT, а вам необходимо использовать функцию библиотеки LIBGEO, следует написать: LIBGEO.ДлинаСегмента.

Все связанные с библиотекой операции выполняются через одно диалоговое окно. Можно создать **[Новая]** библиотеку. Присвоенное имя приводится в списке установленных библиотек. Можно импортировать уже существующие библиотеки или же преобразовывать файлы групп в библиотеки, вызывая их при помощи диалогового окна, открываемого кнопкой **[Импорт..]**. Библиотеки удаленные используя команду **[Удалить]** перемещаются в мусорное ведро Windows.

Для изменения кода библиотеки выбрать кнопку **[Правка]**. При этом библиотека открывается редактором GPL. При составлении функций библиотеки необходимо соблюдать следующие основные правила:

- невозможно получить доступ к устройствам, функциям, переменным, свойственным конфигурации, внутри которых пишется функция.
- можно вызывать общедоступные функции и переменные других библиотек
- функции, объявленные внутри библиотеки, по умолчанию определяются частными. Для того, чтобы их можно было вызвать из других файлов функции, необходимо объявить их типа **PUBLIC**.

Возможность правки библиотеки зависит от уровня доступа того, кто использует Albatros. Разрешения доступа к библиотеке можно присвоить или изменить, нажимая кнопку **[Свойства]**.

Глобальные переменные, объявленные в библиотеке, отображаются в **Диагностике** в соответствующем разделе. Отображение элементов библиотеки связано с правами доступа того, кто использует Albatros.

8.3 Отладка

8.3.1 Отладчик

Отладчик - это функция Albatros, позволяющая шаг за шагом отслеживать последовательность инструкций задачи GPL, позволяя, таким образом, понять и скорректировать возможные логические ошибки и нежелательные действия кода.

Эта функция активна только при уровне пароля, равном или выше пароля изготовителя.

При помощи отладчика можно, например:

- задать точки прерывания
- прервать выполнение задачи и отобразить значение переменной
- следить за последовательностью выполнения функции
- проверить значение, принятое локальной переменной
- проверить, что в случае инструкции If выбрана соответствующая ветвь

Команды, используемые в режиме отладки, можно выбрать из меню **Отладка**. Основные таковы:

Продолжить	Возобновляет выполнение заблокированной задачи. Задача будет выполняться до тех пор, пока она снова не будет заблокирована или не будет достигнута точка прерывания.
Перезапустить	Перезапускает отладку выбранной задачи.

Прервать сейчас	Блокирует выполнение задачи, для которой выполняется отладка. Курсор позиционируется на строку, на которой была прервана инструкция. После блокировки задачи можно управлять ее выполнением и проверить состояние локальных переменных
Выполнить инструкцию	Выполняет отдельную инструкцию GPL. Задача предварительно должна была быть прервана.
Выполнить до конца функции	Выполняет все инструкции до первой инструкции функции, следующей за текущей
Выполнить инструкцию/функцию	Выполняет отдельную инструкцию GPL или, если инструкция является вводом функции, - выполняет всю функцию
Выполнить до курсора	Выполняет инструкцию до положения курсора.
Завершить отладку	Завершает выполнение отладки. Файл функций, из которого выполнялась отладка, открывается в режиме редактора.

Для доступа к отладке необходимо отобразить [список выполняемых задач](#) (из меню **Отладка->Выполняемая задача**) или [список всех задач](#) (из меню **Отладка->Все задачи**) и выбрать задачу, на которой должна выполняться отладка.

Перед выполнением отладки необходимо проверить, что нет ошибок компиляции функций (например, ошибок синтаксиса, необъявленных переменных) и что модуль, на котором должна выполняться отладка, был правильно инициализирован.

Окно отладки подобно окну редактора GPL, однако в нем нельзя вносить изменения в код. Фон окна серый, а выполняемая строка выделяется желтым цветом.

Примечание: Невозможно выполнить одновременно отладку нескольких задач, относящихся к одному модулю.

8.3.2 Выполняемые задачи










Эту команду можно выбрать из меню **Отладка->Выполняемые задачи**. Отображает список выполняемых задач, связанных со станком или модулем. При выборе задачи можно [выполнить ее отладку](#) или завершить ее выполнение, используя, соответственно, кнопки **[Отладка]** или **[Завершить]**.

8.3.3 Все задачи

Представляет в диалоговом окне список всех задач, которые были определены в коде GPL. Графически они представляются в древовидной структуре. При выборе функции открывается файл, в котором она определена, и курсор позиционируется на ее первую инструкцию. Таким образом можно задать [Точки прерывания](#) еще до начала выполнения.

Возможно выполнить отладку всякой задачи и функции без параметров.

Ниже приводится описание значения каждого символа, использованного для составления дерева исполнения задач. Особый символ - символ, указывающий рекурсивную функцию. Это функция, вызывающая функцию, из которой она сама вызывается.

Символ	Описание
	задача главной функции интергруппы
	задача автозапуска
	произвольная задача
	задача реального времени
	функция группы
	функция группы, выполняемая такими инструкциями, как ONINPUT, ONFLAG
	функция библиотеки
	функция библиотеки, выполняемая такими инструкциями, как ONINPUT, ONFLAG.
	рекурсивная функция

8.3.4 Вызовы по функции

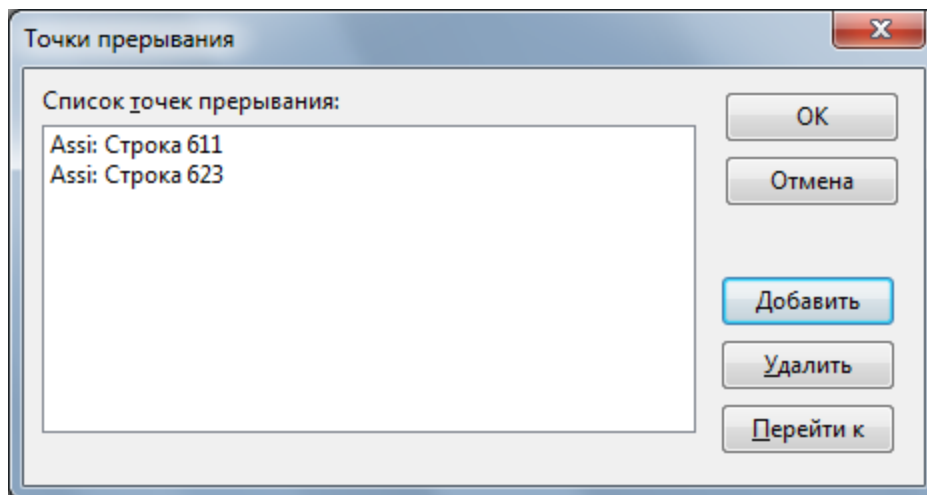
При отладке можно отобразить список функций, которые были вызваны, но еще не возвратили (т. е. все функции, в которых еще не была выполнена инструкция FRET). Открывается диалоговое окно со списком вызовов функции, приведших к текущей инструкции. В верхней части списка приводится функция, выполненная последней.

Для наблюдения за поведением вызова по функции:

- переместить курсор в нужное положение внутри функции.
- выбрать **Отладка->Выполнить до курсора** для приведения выполнения программы в нужное положение.
- выбрать **Отладка->Вызовы по функции** или горячую клавишу CTRL+K.
- из диалогового окна Вызовы можно выбрать имя какой-либо функции. Курсор переместится на первую инструкцию выбранной функции.

8.3.5 Точки прерывания

Точка прерывания позволяет более подробно видеть последовательность выполнения инструкций, анализировать или менять переменные и устройства, анализировать список вызовов по функции и т. д. Выполнение задачи блокируется при достижении инструкции, в которую была введена точка прерывания. Точки прерывания могут задаваться как до того, как определенная задача направляется на выполнение, так и в ходе ее выполнения (из меню **Отладка->Точки прерывания**). Можно также удалить точки прерывания, когда они больше не требуются.



Список точек прерывания

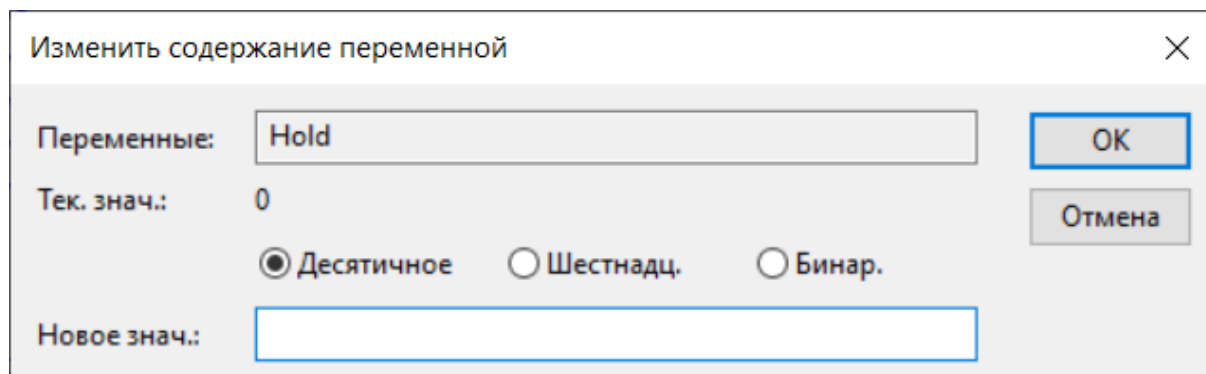
Существуют ситуации, в которых, несмотря на то, что были введены точки прерывания, задача не блокируется, так как выполнение никогда не достигает точки прерывания. В этих случаях можно заблокировать задачу командой **Отладка->Прервать сейчас**. Курсор устанавливается на инструкцию GPL, которая должна была выполняться в момент прерывания.

8.3.6 Содержание переменной

Эту команду можно выбрать из меню **Отладка->Содержание переменной**.

После прерывания выполнения задачи можно отобразить:

- значение локальных переменных, объявленных в функции, в которой остановлена задача
- глобальные переменные
- значение, принятое выражением
- состояние устройств и параметров типа устройства.



Отображение/Изменение содержания переменной

Если переменная (или устройство) не является только для считывания, ее содержимое можно изменить. Естественно, изменение повлияет на последующее выполнение этой задачи. Изменение значения переменной или устройства позволяет попробовать выполнение в условиях, отличных от тех, в которых осуществляется нормальное выполнение, скорректировать возможные выявленные ошибки и продолжить выполнение последующих инструкций.

Можно отобразить содержимое переменной, устройства или константы также путем перемещения мыши на переменную, на имя устройства или константы. Отображается контекстное окно указателя, в котором указывается тип, имя и значение данных. При выборе выражения отображается результат. Если курсор мыши находится внутри выбора, используется весь выбор, в противном случае - только слово, на котором установлен курсор мыши. Если курсор мыши не находится внутри слова, используется весь аргумент. Например, для отображения значения матрицы $Mx[3][\text{столбец}]$, если курсор мыши находится на "3", в контекстном окне отображается 3, если курсор находится на "столбец", отображается значение столбца, а если на "матрица" - не отображается ничего, если на одной квадратной скобке - отображает значение $Mx[3][\text{столбец}]$.

8.3.7 Список горячих клавиш

Для подключения команд **Отладка** можно выбрать пункты меню **Отладка** или же использовать клавиатуру. Используются следующие клавиши:

Клавиша	Описание
CTRL+F5	открывает диалоговое окно с перечнем выполняемых задач
CTRL+SHIFT+F5	открывает диалоговое окно со списком всех задач
CTRL+B	открывает диалоговое окно для ввода или удаления точек прерывания
CTRL+F9	вводит или удаляет точку прерывания на строке, на которой установлен курсор
CTRL+K	открывает диалоговое окно для отображения списка функций, которые вызывались, но которые еще не вернули
SHIFT+F9	открывает диалоговое окно для отображения содержимого переменной
F8	выполняет инструкцию. Если это функция, входит в функцию.
SHIFT+F7	выполняет все инструкции функции
F10	выполняет инструкцию. Если это функция, выполняет ее без вхождения в нее.
F7	выполняет все инструкции до инструкции, на которой установлен курсор. Курсор должен устанавливаться на инструкции внутри функции
ALT+ПРЕКР	прерывает выполнение кода последней выполненной инструкции
F5	возобновляет выполнение кода после прерывания
SHIFT+F5	завершает текущую задачу и выполняет ее снова
ALT+F5	завершает отладку

8.4 Инициализация блока управления


8.4.1 Сетевые подключения

Работа Albatros в станке защищена аппаратным ключом USB, сконфигурированным компанией TPA. Эту команду можно выбрать из меню **ЧПУ->Сетевые подключения**. Она отображает состояние подключенных к системе удаленных модулей. Если модуль не подключен, представляющий его символ отмечается красным крестиком X.

Для каждого модуля имеется два поля. Первое поле - имя модуля, второе - имя станции сети. Обычно станция сети имеет первые знаки, которые обозначают имя и остаются без изменений, "TPA" за которыми следует серийный номер удаленного модуля.

Присвоение узла сети (удаленного модуля) логическому модулю

Можно присвоить узел сети модулю, выбирая указателем мыши пункт "Не сконфигурирован" или нажимая кнопку **[Правка]**. Через несколько секунд отобразится окно со списком доступных в сети удаленных модулей (каждый удаленный модуль должен быть включен и должен иметь правильно присвоенный IP-адрес).

Теперь можно выбрать узел сети, который мы желаем подключить к логическому модулю, и подтвердить выбор нажатием кнопки .

Обратите внимание, что эта операция может выполняться с уровнем пароля "Сервис", без необходимости входа в конфигурацию системы Albatros, который требует уровень пароля "Изготовитель".

Обновление программного обеспечения удаленного модуля.

Можно обновить полностью программное обеспечение блока управления, присутствующего на внутренней памяти удаленного модуля, нажимая кнопку **[Обновить]**. До выполнения обновления, необходимо проверить, что выбранный удаленный модуль подключен к Albatros.

8.4.2 Аппаратная Диагностика

Эту команду можно выбрать из меню **ЧПУ->Аппаратная Диагностика**. Аппаратная Диагностика представляет список и состояние сконфигурированных модулей, плат и узлов относящихся к ним в соответствии с определенным в конфигурации аппаратного обеспечения. Если символ платы или узла отмечается красным крестиком X, это означает, что этот блок не был обнаружен среди аппаратного обеспечения, имеющегося на блоке управления, или что его было невозможно правильно инициализировать. Обозначение желтым вопросительным знаком означает, что система обнаружила наличие платы или узла, но он не соответствует типу, определенному в конфигурации.

Топология сети EtherCAT

В окне диагностики оборудования, когда в *дереве* выбран узел сети EtherCAT, активируется кнопка **[Детали]**, которая, в свою очередь, включает графическое отображение топологии сети EtherCAT.

Этот график показывает информацию об узлах, физически присутствующих в сети: состояние каждого узла, состояние осей, если узел является сервоприводом, и качество связи. Каждый узел и каждая ось представлены в виде прямоугольника, цвет которого определяет состояние.

При наведении курсора мыши на прямоугольник отображается всплывающая подсказка, которая описывает состояние и ошибки связи узла или состояние оси.

Просмотр и редактирование объектов в узлах

В окне диагностики оборудования, когда в *дереве* выбран узел сети EtherCAT, активируется кнопка **[Объект словаря]** для отображения и редактирования объектов узла. Редактирование данных возможно только на уровне **Изготовитель**.

Объекты сгруппированы по их адресу.

Начальный адрес	Окончательный адрес	Название области
0x0000	0x0FFF	Data type area
0x1000	0x1FFF	Communication area

0x2000	0x5FFF	Manufacture specific area
0x6000	0x6FFF	Input area
0x7000	0x7FFF	Output area
0x8000	0x8FFF	Configuration area
0x9000	0x9FFF	Information area
0xA000	0xAFFF	Diagnosis area
0xB000	0xBFFF	Service transfer area
0xC000	0xEFFF	Reserved area
0xF000	0xFFFF	Device area

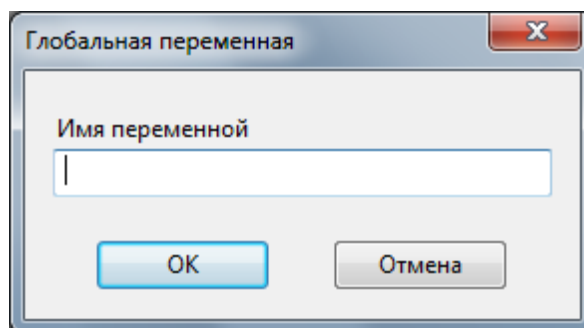
Целочисленные числовые значения отображаются как в десятичном, так и в шестнадцатеричном виде. При изменении их значения число также может быть введено в шестнадцатеричном формате, используя обозначения \$... H, и в двоичном, с помощью \$... B (как делается в GPL).

8.5 Тест

8.5.1 Запомнить глобальную переменную

Эту команду можно выбрать из меню **Тест->Запомнить глобальную переменную**.

Записывает содержание глобальной переменной на диск в форме файла форматированного текста. Файл имеет имя, заданное *имяпеременной.txt*, и будет сохранен в папке *Отчет*. Эта операция возможна только если уровень доступа при считывании глобальной переменной совместим с текущим уровнем доступа.

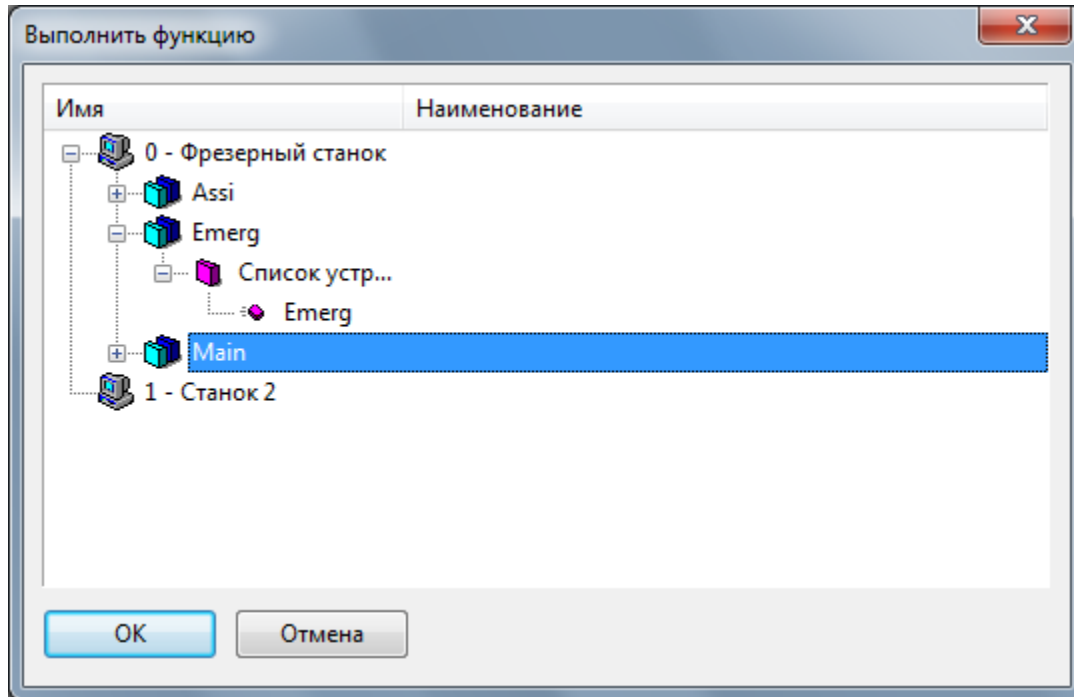


Запись в память глобальной переменной

8.5.2 Выполнить функцию

Эту команду можно выбрать из меню **Тест->Выполнить функцию**.

Выполняет функцию независимо от остальной системы. Затем создается задача, которая начинает свое выполнение с выбранной функции и берущая ее имя. Могут выполняться только функции, не имеющие параметров на входе и уровень доступа к которым при считывании совместим с текущим уровнем доступа. Если выполняемая функция - это главная функция интергруппы, затем выполняются и все задачи с автозапуском.



Выбор функции для выполнения вручную

8.5.3 Импорт сообщений

Сообщения группы, заданные при помощи инструкции GPL [DEFMSG](#), хранятся в файле в формате xmlng. Сообщения, содержащиеся в файле, могут быть изменены, добавлены или удалены.

Для просмотра изменений в коде GPL необходимо использовать команду **Тест->Импортировать сообщения группы**, включенную только тогда, когда нет открытых окон.

Для выполнения импорта весь код GPL должен быть скомпилирован без ошибок.

В противном случае пользователю представляется сообщение "Компилирован не весь код GPL".

Не импортируются сообщения группы, относящиеся к [зашифрованным](#) файлам (См. главу **Средства разработки->Редактор GPL->Криптография**), для которых у пользователя нет разрешения отображения в расшифрованном виде.

Импортируются только сообщения, которые уже были определены в коде GPL. Изменение текста GPL не выполняется, если имеется по меньшей мере одна DEFMSG после инструкции IFDEF.

При импорте могут возникать ошибки, если:

- среди текстов конкретного сообщения группы идентификатор языка присутствует более одного раза
- любой текст пуст (т. е. "")
- имя группы или библиотеки определено более одного раза.

По завершении импорта выполняется компиляция всех модулей, в которых были изменены группы или библиотеки.

Все файлы, перечисленные ниже, могут быть использованы в файлах XMLNG

Идентификаторы, которые можно использовать AUXLANG:

- "AFK" Африкаанс
- "ARA" Арабский
- "AZE" Азербайджанский
- "BAS" Башкирский
- "BEL" Белорусский

"BGR" Болгарский
"BSB" Боснийский (латинский алфавит)
"BSC" Боснийский (кириллица)
"BRE" Бретонский
"CAT" Каталанский
"CHS" Упрощенный Китайский
"CHT" Традиционный Китайский
"COS" Корсиканский
"CSY" Чешский
"CYM" Гэльский
"DAN" Датский
"DEA" Немецкий (Австрия)
"DEU" Немецкий (Германия)
"ELL" Греческий
"ENG" Английский
"ENU" Английский (Соединенные Штаты Америки)
"ESP" Испанский
"ETI" Эстонский
"EUQ" Баскский
"FAR" Персидский
"FIN" Финский
"FRA" Французский
"FPO" Филиппинский
"FRB" Французский (Белгия)
"FYN" Фризский
"GLC" Галицийский
"HAU" Хауса
"HEB" Еврейский
"HRB" Хорватский (Босния-Герцеговина)
"HRV" Хорватский (Хорватия)
"HUN" Венгерский
"IBO" Игбо
"IND" Индонезийский
"IRE" Ирландский
"ISL" Исландский
"ITA" Итальянский
"JPN" Японский
"KAL" Гренландский
"KOR" Корейский
"SAH" Сакский
"KYR" Киргизский
"LVI" Латышский
"LTH" Литовский
"LBX" Люксембургский
"MNN" Монгольский
"NON" Норвежский Ньюорск
"NOR" Норвежский Букмол
"NLB" Нидерландский (Белгия)
"NLD" Нидерландский (Нидерланды)
"OCI" Окситанский
"PLK" Польский
"PTB" Португальский (Бразилия)
"PTG" Португальский (Португалия)
"RMC" Ретороманский
"ROM" Румынский
"RUS" Русский
"SKY" Словацкий
"SLV" Словенский
"SQI" Албанский
"SRM" Сербский (латинский алфавит, Сербия)
"SRN" Сербский (кириллица, Босния-Герцеговина)
"SRO" Сербский (кириллица, Сербия)
"SRP" Сербский (латинский алфавит, Монтенегро)
"SRQ" Сербский (кириллица, Монтенегро)
"SRS" Сербский (латинский алфавит, Босния-Герцеговина)
"SVE" Шведский
"TAJ" Таджикский
"TRK" Турецкий
"TTT" Татарский
"TUK" Туркменский

"UKR" Украинский
"URD" Урду
"UZB" Узбекский
"VIT" Вьетнамский
"WOL" Волоф
"XHO" Коса
"YOR" Йоруба
"ZUL" Зулусский

8.5.4 Примечание пользователя в файле отчета о тревоге

С помощью команды **Вставить заметку** из меню **Тест** можно вставить уведомление в файл отчета тревог за текущий месяц (MONTHхх.ТЕР). Пункт меню активируется с уровнем пароля, равным или высшим сервисного пароля.

8.6 Инструменты

8.6.1 Индивидуализировать...

Эту команду можно выбрать из меню **Инструменты->Индивидуализировать...**

Позволяет задавать максимум 20 программ, выполнение которых может запускаться из меню **Инструменты** Albatros.

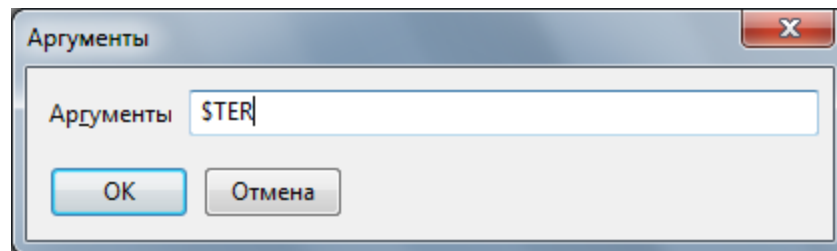
Конфигурация меню Инструменты

- Структура меню:** перечисляются программы, отображаемые в меню **Инструменты**.
- Команда:** имя выполняемой программы. Дополнительно может определяться также папка, в которой хранится программа, если она отлична от папки, в которой выполняется Albatros, или папки, в которой операционная система выполняет поиск выполняемых файлов (разная в среде Windows PATH).
- Текст в меню :** это имя, которое должно отображаться в меню **Инструменты** для обозначения выполняемой программы.
- Аргументы:** любое сочетание аргументов из командной строки, которое требуется программе для правильного выполнения. Можно ввести динамические аргументы. Например, при использовании строки \$Ter при выполнении ViewRER открывается файл отчета за текущий месяц.
Ниже представлен список аргументов:

\$File	Полное имя маршрута текущего файла.
\$FileName	Имя файла и расширение текущего файла.
\$FileDir	Диск и папка текущего файла.
\$Ter	Полное имя маршрута файла отчета об ошибках за текущий месяц.
\$DirModule	Диск и папка с MODx текущего файла.
\$Module	Номер модуля текущего файла.
\$Bin	Диск и папка с исполняемым файлом Albatros.
\$TraIni	Полное имя маршрута файла инициализации tra.ini

\$ReqDirModule	Маршрут (диск и папки) модуля Albatros. Если сконфигуровано несколько модулей, открывается окно выбора модуля.
\$ReqModule	Номер модуля Albatros. Если сконфигуровано несколько модулей, открывается окно выбора номера модуля.
\$ReqFile	Имя файла. Открывается окно запроса имени файла. Выбранный файл будет передаваться в аргументы выполняемой программы.

Запрос аргументов: при выборе каждый раз при запросе выполнения программы открывается диалоговое окно для ввода аргументов, отличных от заданных в пункте Аргументы, которые могут меняться в зависимости от режима запуска программы.



Указывает аргументы запуска программы

Уровень доступа: задает уровень отображения программы в меню *Инструменты*. Программам теста или изменения данных Albatros обычно задается уровень изготовителя. Программам для правки обработок станка задается уровень доступа пользователя.

Редактирование некоторых полей может выполняться также при помощи кнопки **[Добавить...]**. При этом открывается диалоговое окно **Добавить инструмент** для выбора выполняемой программы. Разрешенные типологии исполнимых файлов - .EXE, .COM, .PIF, .BAT.

При закрытии диалогового окна с подтверждением данных программа вводится в окно **Структура меню**, имя программы и папки, к которой она относится, - в строку **Команда**.

Имеются другие кнопки, **[Удалить]**, **[Сместить вверх]**, **[Сместить вниз]**, которые используются, соответственно, для удаления программы и упорядочения списка программ.

8.7 Браузер

8.7.1 Браузер

Браузер - это функция Albatros, которая использует информацию, генерированную компилятором для создания базы данных для быстрого поиска символов, определенных в функциях.

Эта функция активна только при уровне пароля, равном или выше пароля изготовителя. Используемые команды можно выбрать из меню **Отладка**.

При помощи браузера можно:

- установить курсор на строку, в которой впервые определялась функция, переменная или константа модуля, группы или библиотеки. (из меню **Отладка->Перейти к определению**)
- установить курсор на строки, в которых приводится эталон для функции, устройства, переменной модуля или группы, инструкции GPL (за исключением инструкций FCALL и FRET) (из меню **Отладка->Перейти к эталону** для отображения предыдущего или последующего эталона, выбрать из меню, соответственно, пункты **Отладка->Предыдущий** или **Отладка->Следующий**)

Управление переменными группы осуществляется только в окне редактирования группы, к которой они относятся.

Для обновления браузера для перехода на новую версию рекомендуется прежде всего сохранить глобальные переменные, а затем выполнить команду **Файл->Компилировать все**.

На этапе редактирования функций утрачивается соответствие между текстом и искомыми символами, которое возобновляется на этапе архивирования.

8.7.2 Искать идентификатор...
















Эту команду можно выбрать из меню **Отладка->Искать идентификатор...** Она открывает диалоговое окно для ввода имени символа для поиска в коде GPL. В зависимости от заданного **Типа поиска** отыскивается определение или первая ссылка символа.

Вводимое имя может иметь следующие характеристики:

- не содержит ни одного знака '.' (точка) : поиск ведется во всех файлах функций
- содержит только один знак '.' (точка): имя до точки считается именем группы и символ ищется только в этой группе. Например, при определении двух функций с именем VisError, одна в группе ГЛАВНАЯ, другая в группе ОСИ, при записи в качестве идентификатора ОСИ.VisError курсор будет установлен на первую строку функции VisError группы ОСИ.
- содержит два символа '.' (точка): имя до первой точки считается именем группы, а имя до второй точки считается именем подгруппы и поиск символа выполняется только в этой подгруппе.
- если завершается символом '*' (звездочка), учитываются все символы, начинающиеся с символов, указанных до звездочки.

При неопределенности в поиске символа открывается диалоговое окно, в котором отображаются все символы с запрошенным именем. Отсюда можно выбрать нужный символ.

Ниже описывается значение особых символов, которые используются в списке для выбора идентификатора.

Символ	Описание
	инструкция GPL
	константа модуля или группы или библиотеки
	переменная модуля или группы
	переменная библиотеки
	вектор библиотеки
	матрица библиотеки
	функция библиотеки
	сообщение группы
	ярлык
	локальная переменная
	локальный вектор
	локальная матрица
	простой параметр
	параметр вектора
	параметр матрицы

8.7.3 Список горячих клавиш

Для подключения команд Браузера можно выбрать пункты меню **Отладка** или же использовать клавиатуру. Используются следующие клавиши:

Клавиша	Описание
F2	устанавливает курсор на строку, на которой определен выбранный символ. Если в базе данных определено несколько символов с выбранным именем, открывается диалоговое окно, в котором пользователю предоставляется возможность выбирать требуемый символ.
SHIFT +F2	позиционирует курсор на первую ссылку выбранного символа. При неопределенности открывается диалоговое окно для выбора выбранного символа.
CTRL+F2	открывает диалоговое окно для выбора выбранного символа
CTRL+'+' или CTRL+PgUp	позиционирует курсор на следующую ссылку (использовать '+' цифровой клавиатуры)
CTRL+'-' или	позиционирует курсор на предыдущую ссылку (использовать '-' цифровой клавиатуры)

9 Вспомогательные программы

9.1 XConfMerge: программа для слияния конфигурационных файлов

XConfMerge - это инструмент, который объединяет файлы конфигурации. Он запускается из командной строки из папки bin, так как XConfMerge читает файл tra.ini.

Файлы, прочитанные XConfMerge:

- hardware.xconf: содержит данные виртуального-физического и аппаратной конфигурации (физические)
- devices.xconf: содержит данные конфигурации групп, подгрупп и устройств (логические)
- devices.xmlng: содержит переводимые сообщения. В любом случае все языковые файлы в папке считаются
- address.xdb: содержит логические адреса устройств.

Аргументы для прохождения:

- папка для чтения новых файлов
- номер модуля, на который ссылается пользовательское значение. Если номер не указан, 0 считается значением по умолчанию.

Путь для чтения файла, подлежащего обновлению, и путь для записи файла, полученный с помощью процедуры слияния, выводится из набора данных в tra.ini.

ВНИМАНИЕ! Обновляемые файлы перезаписываются и автоматическое резервное копирование не выполняется.

Правила слияния файлов конфигурации групп:

1. если в обоих файлах присутствует одна и та же группа, подгруппа или устройство, данные и включение старого файла сохраняются.
2. если группа, подгруппа или устройство присутствует только в новом файле, данные и включения копируются.
3. если группа, подгруппа или устройство присутствует только в старом файле, она/о удаляется.

Правила слияния файла конфигурации аппаратного обеспечения и виртуального-физического:

1. если в обоих файлах присутствует одно и то же аппаратное обеспечение, аппаратное обеспечение и виртуальное-физическое нового файла сохраняются с включениями, определенными в старом файле.
2. если аппаратное обеспечение существует только в новом файле, новое оборудование поддерживается с его включением и новым виртуальным-физическим.
3. если аппаратное обеспечение существует только в старом файле, оно удаляется вместе с виртуальным-физическим.

Правила слияния файла логических адресов:

1. если файл не находится в папке импортируемых файлов, то сохраняются логические адреса пользователя и новые адреса присваиваются новым устройствам, если таковые имеются.
2. если файл находится в папке импортируемых файлов, то логические адреса, содержащиеся в нем, считываются и используются.

Правила слияния файлов сообщений:

1. Слияние не выполняется, но новый файл сообщений копируется в папку модуля.

В конце выполнения инструмент слияния возвращает следующие значения:

0	все в порядке
1	слияние файлов не было успешным
2	аргументы не определены
3	номер модуля, переданный в качестве аргумента, неверен
4	папка, указанная в качестве первого аргумента, не существует

9.2 XParMerge: программа для слияния двух параметрических файлов

XParMerge - это инструмент, который объединяет два параметрических файла.

Он запускается из командной строки из папки bin, так как XParMerge читает файл tra.ini. Аргументы для передачи: имя нового файла и номер модуля, к которому он относится.

Путь для чтения файла, подлежащего обновлению, и путь для записи файла, полученный с помощью процедуры слияния, выводится из набора данных в tra.ini.

ВНИМАНИЕ! Автоматическое резервное копирование старого параметрического файла не выполняется, но оно перезаписывается.

Правила слияния файлов Технологических Параметров :

- 1) если один и тот же элемент управления существует в обоих файлах, значение старого файла сохраняется, но другие параметры, которые его определяют, обновляются (отключены, видимы, имя переменной GPL), извлекая их из нового файла;
- 2) если элемент управления определен в новом файле, который не существует в старом файле, элемент управления сохраняется;
- 3) если элемент управления определен в старом файле, который не существует в новом файле, элемент управления удаляется.

Правила слияния файлов Параметров Инструмента:

- 1) если эталонный диалог нового файла отличается от эталонного диалога старого файла, сохраняются все диалоги нового файла, если таковые имеются, и обновляются диалоги старого файла (удаляются или добавляются новые элементы управления);
- 2) если эталонный диалог нового файла совпадает с эталонным диалогом старого файла, диалоги нового добавляются к диалогам старого;
- 3) если эталонный диалог присутствует только в старом файле, он и его диалоги удаляются.

10 Язык GPL

10.1 Базовые концепции

10.1.1 Введение в язык GPL

Язык GPL (General Purpose Language - универсальный язык) представляет собой язык, используемый для создания функций системы Albatros. Его структура частично похожа на структуру языка BASIC, однако характеризуется наличием множества инструкций, предназначенных для управления устройствами. Этот язык состоит из более 200 инструкций, которые для удобства подразделяются на группы, состоящие из инструкций со сходными функциями. Кроме того, язык является [многозадачным](#), т. е. позволяет выполнять одновременно несколько задач.

Типовой синтаксис инструкции GPL

Все инструкции GPL имеют близкую структуру, построенную по следующей схеме:

имяинструкции параметр-1, параметр-2, параметр-N

Число фактически присутствующих параметров будет зависеть от инструкции и контекста ее использования. Как правило, максимальное число параметров, допустимых для одной инструкции или функции GPL, составляет 120. В некоторых случаях параметры могут не присутствовать.

Минимальный блок кода GPL - [функция](#).

Разделение кода на группы

Код GPL делится на блоки, отражающие логическое деление станка на группы. Таким образом, для каждой группы у нас будет файл со связанным с ней кодом. К файлам с кодом групп, имеющихся на станке, добавляются файл глобальных переменных и констант, которые может видеть код GPL всех групп, и [библиотеки](#). Библиотеки содержат код, не связанный с конфигурацией станка, поэтому легко переносимый с одного станка на другой.

10.1.2 Условности и терминология

Основные использованные термины

АРГУМЕНТ	Это один из аргументов инструкции. Может определяться как <i>константа</i> , <i>переменная</i> или как <i>параметр</i> , в зависимости от типа инструкции. Если он заключен в квадратные скобки ([]), это означает, что его можно опустить, что обуславливает, естественно, другой способ исполнения инструкции
КЛЮЧЕВОЕ СЛОВО	Это аргумент, который должен выбираться между аргументами с определенным значением и, как правило, представляется прописными буквами. Список ключевых слов представлен на специальной странице справки
ПАРАМЕТР	Это аргумент инструкции, не определенный в самой инструкции, а передаваемый как параметр функции в момент ее выполнения. Иногда называется также термином <i>параметризированный аргумент</i>
КОНСТАНТА	Это аргумент, определенный фиксированным образом при помощи метакоманды CONST или жестко зафиксированный в инструкции
ПЕРЕМЕННАЯ	Это аргумент, определенный как глобальная переменная модуля или группы или при помощи инструкции LOCAL, который может быть организован как простая переменная или как вектор или как матрица. См. Переменные
ПАРАМЕТР КОНФИГУРАЦИИ	Это аргумент, который был определен в конфигурации, например, параметры оси

Аргументы, наиболее часто встречающиеся в описании инструкций

Далее перечислены термины, относящиеся к темам, которые часто встречаются в синтаксисе инструкций GPL, и их описание. Если один аргумент может принимать значения, отличные от описанных ниже, его описание снова приводится в разделе *Аргументы* страницы справки инструкции.

имявхода	имя устройства цифрового входа
имявыхода	имя устройства цифрового выхода
имяфлажка	имя устройства флажкового выключателя или флагового бита
имяпорта	имя устройства входного порта, выходного порта или флагового порта
имятаймера	имя устройства таймера
имясчетчика	имя устройства счетчика
имяфункции	имя функции (действительно также как параметр устройства в случае ERRSYS.)
имяподпрограммы	имя подпрограммы, эквивалентно <i>ярлыку</i> , на который необходимо ссылаться для пояснения. Вызов подпрограммы выполняется при помощи инструкции "CALL имяподпрограммы".
ось	имя оси
константа	символ или целое число или число двойной точности или ключевое слово константа или переменная (<i>тип</i> зависит от инструкции)
значение	имя переменной, элемента вектора или элемента матрицы
переменная	имя <i>параметра устройства</i>
переменнаяустройства	имя матрицы
матрица	имя вектора
вектор	имя ярлыка перехода или имя подпрограммы.
ярлык	логическое состояние, может быть ON или OFF или 1 или 0
состояние	количество времени, в течение которого должно произойти определенное событие, или продолжительность задержки (константа или переменная)
таймаут	значение координаты (константа double или переменная double)
координата	значение радиуса (константа double или переменная double)
радиус	значение угла (константа double или переменная double)
угол	число оборотов (константа double или переменная double)
числооборотов	значение скорости (константа float или переменная float)
скорость	вращение по или против часовой стрелки (переменная или константа: CW или CCW)
направление	(константа или переменная или имяустройства)
операнд	результат операции (переменная или имяустройства)
результат	имя любого типа устройства (или параметр устройства)
имяустройства	последовательность символов, заключенная в двойные кавычки (напр., "строка")
строковаяконстанта	имя вектора символов или строка
строковаяпеременная	операторы сравнения, в т. ч. в сочетании между собой: > (больше) = (равно) < (меньше)
оператор	могут комбинироваться при их размещении рядом, напр., >= (т. е. больше или равно)
тип	тип константы или переменной: "char" (8 бит), "integer" (32 бит), "float" (32 бит), "double" (64 бит), "string"
параметр устройства	это переменная, представляющая устройство. Устройства определяются в конфигурации.

Основные термины, используемые для оси

теоретическая (или целевая) координата "Теоретическое" текущее положение, задаваемое момент за моментом ЧПУ в зависимости от алгоритма генерирования профиля скорости.

реальная координата Фактическое положение оси, определяемое датчиком положения. Отличается от теоретической координаты на величину, называемую "погрешность преследования" или "ошибка кольца".

конечная координата Соответствует запрограммированной координате прибытия движения. Алгоритм расчета профиля скорости приводит теоретическую координату к точному достижению конечного значения.

окно прибытия на координату	Программируемый интервал, центральной точкой которого является конечная теоретическая координата: когда реальная координата оказывается внутри этого интервала, фактическое движение считается завершённым.
большое окно прибытия на координату	Окно прибытия на координату, умноженное на коэффициент, задаваемый при помощи инструкции SETBIGWINFACTOR
погрешность преследования	Разница, момент за моментом, между теоретической и реальной координатой оси. Обычно пропорциональна скорости перемещения и обратно пропорциональна "пропорциональному усилению контура".
пропорциональное усиление [контур]	Программируемый параметр регулирования оси. Определяет отношение между текущей скоростью и относительной погрешностью преследования.
Подача вперед	Программируемый параметр регулирования оси. Определяет прямой вклад (пропорциональный запрограммированной скорости), поданный на блок управления скоростью привода. Позволяет снижать, при одинаковой скорости и одинаковом пропорциональном усилении, значение погрешности преследования.
Скорость подачи override	Процент запрограммированной скорости. Этот параметр позволяет снижать скорость выполнения по сравнению с запрограммированной в процентном отношении, которое может быть от 0% до 100%.
допуск	Значение смещения, на которое ось отклоняется от изначальной траектории при многоосевой интерполяции между двумя последовательными блоками смещения.
механический зазор	Расстояние между пазом и зубом зубчатой пары.

10.1.3 Переменные

Переменные являются контейнерами информации, которые используются языком GPL для размещения в них значений, необходимых для выполнения программы. Переменные характеризуются "типом", отражающим характеристики размещаемой информации. Кроме того, с каждой переменной ассоциируется видимость, определяющая множества или подмножества кода, которые могут оперировать (записывать и считывать) на ней.

Типы данных

ПРОСТЫЕ или СКАЛЯРНЫЕ ДАННЫЕ

GPL поддерживает данные простого и агрегированного типа. Данные простого типа подобны данным, имеющимся в большинстве языков программирования:

Char

Это целое число со знаком в интервале [-128 ; +127] длиной 1 байт.

Объявление переменной типа char выполняется при помощи следующего синтаксиса:

```
ИмяПеременной as char
```

Integer

Это целое число со знаком в интервале [-2147483647 ; +2147483647] длиной 4 байт (соответствует типу long в C).

Объявление переменной типа Integer выполняется при помощи следующего синтаксиса:

```
ИмяПеременной as integer
```

Float

Это число с плавающей запятой в интервале [-3,402 E+38 ; +3,402 E+38] длиной 4 байт (обычно используется для представления скоростей).

Объявление переменной типа Float выполняется при помощи следующего синтаксиса:

```
ИмяПеременной as float
```

Double Это число с плавающей запятой в интервале [-1,797 E+308 ; 1,797 E+308] длиной 8 байт (обычно используется для представления координат)

Объявление переменной типа Double выполняется при помощи следующего синтаксиса:

```
ИмяПеременной as double
```

Данные этих типов могут использоваться одновременно в одном выражении. GPL выполняет автоматическое преобразование без предоставления какого-либо предупредительного сообщения. Поэтому необходимо уделять внимание возможной утрате информации при использовании в одном выражении данных различных типов.

В некоторых ситуациях преобразование не разрешается. В этих случаях, как правило, представляются предупреждения компилятора или ошибки системы.

АГРЕГИРОВАННЫЕ ДАННЫЕ**Массив**

Это совокупность переменных простого типа, все одного типа, полученная при присвоении индекса имени переменной. Индекс заключается в квадратные скобки. Если массив называется, например, "параметры", первый элемент множества будет указываться как "параметры[1]", второй - как "параметры[2]" и т. д.

Массив имеет фиксированное число элементов, которое должно определяться при объявлении.

Типовое объявление массива имеет следующий синтаксис:

```
параметры[10] as integer
```

Где *параметры[10]* указывает, что имя массива - "параметры" и что он состоит из 10 элементов, *as integer* определяет простой тип данных отдельных элементов массива, который в этом случае - integer. Массивы могут состоять из данных простого типа или строк.

Массив может состоять максимум из 262144 элементов.

Вектора могут инициализироваться непосредственно в коде GPL в момент их объявления. Синтаксис GPL может быть:

```
[READONLY] значение[числострок] as integer = 1,2,3,4
```

```
[READONLY] вектор [числострок] as string = "одна","две","три","четыре"
```

Матрицы

Матрицы - это двухмерные массивы, т. е. переменные, для которых заданы два индекса. Матрицу можно ассоциировать таблице данных, разделенную на строки и столбцы. Для указания одной из ячеек таблицы можно указать, в какой строке и в каком столбце таблицы она находится. Первый индекс матрицы соответствует номеру строки, второй - номеру столбца.

В отличие от массивов матрицы могут содержать данные различных типов, но со следующим ограничением: можно использовать один тип простых данных для каждого столбца, но нельзя менять тип в рамках одного столбца.

Например, можно определить матрицу, в которой первый столбец - типа integer, а второй - типа float.

Однако нельзя иметь матрицу, в которой в первой строке мы имеем - integer и float, а во второй строке - char и double. Все строки должны быть одинаковыми в плане типа содержащихся в них данных элементов.

Объявление матрицы может выполняться со следующим синтаксисом:

```
сдвиг[10] as double double double
```

```
раз_дет[50] as float:длина float:ширина float:толщина
```

В объявлении второго типа каждому столбцу задается ярлык или символическое имя. Символические имена столбцов оказываются очень полезными при работе с матрицами больших размеров.

Действительно, в этом случае трудно запомнить значение величин, записанных в каждом столбце матрицы. Символическое имя позволяет немедленно определить тип данных, с которыми мы работаем. Например, "Сдвиг[1][3]" менее понятно, чем "Сдвиг[1].ось_X".

Матрицы могут содержать данные только простоготипа. Например, нельзя создать матрицы, содержащие строки. Матрица может состоять максимум из 262144 строк.

Матрицы могут инициализироваться непосредственно кодом GPL в момент их объявления. Синтаксис GPL может быть:

```
[READONLY] имяматрицы[числострок] as double double integer double = _
  1.1, 2.2, 3, 0.1 _
  1.2, 3.4, 5, 0.1 _
  2.1, 5.6, 6, 0.1
```

Строки

Строки представляют собой совокупность символов, т. е. данных типа char, управление которыми осуществляется особым образом, т. к. они представляют собой читаемый текст.

Строка очень похожа на массив char. Наибольшая разница состоит в наличии символа-терминатора, который автоматически добавляется в конце строк. Кроме того GPL, предоставляет некоторые инструкции, позволяющие работать со строками.

Обычно строки используются для написания сообщений, которые может читать пользователь на экране или в файле отчета.

Объявление переменной типа строки может выполняться со следующим синтаксисом:

```
ИмяПеременной as String
```

```
ИмяПеременной as String[20]
```

В первом типе объявления задается размер по умолчанию в 256 байт. Во втором случае определяется максимальный размер для строки.

Значениями Строки являются последовательности символов заключенные в двойные кавычки. Например, "Нажать кнопку".

Для введения символа "" (двойная кавычка), надо вставить его два раза. Например: "Нажать кнопку""Пуск""".

Для введения символов, через числовой код, написать внутри строки символы \u и дальше числовое значение шестнадцатеричного символа. Например, \u20ac - символ евро. Если пишется "\u20ac 15,6" получается € 15,6.

Преобразование данных

Во всех математических выражениях, за исключением инструкции EXPR, типы данных операндов преобразуются в тип данных переменной результата, а затем выполняется операция. Важно уделять большое внимание объявлению типов данных, т. к. они могут повлиять на результат. Таблица ниже является примером того, как могут меняться результаты в зависимости от типа данных:

DIV	Операнд 1 (Integer)	Операнд 2 (Double)	Результат (char)
	3	5.0	0
	5	1.9	5
	1200	107.2	Неопределенный
	1200	250.0	Неопределенный
DIV	Операнд 1(Double)	Операнд 2 (Double)	Результат (Double)
	3	5.0	0.6
	5	1.9	2.631
	1200	107.2	11.194
	1200	250.0	4.8

Если в инструкции EXPR не все операнды одного типа, выполняется автоматическое преобразование и тип результата операции - больший из двух в соответствии со следующим правилом:

- char < integer
- float < double
- char или integer < float или double.
-

После решения выражения результат преобразуется в тип переменной результата.

```
EXPR Операнд 1 (Double) + Операнд 2 (Integer) / Операнд 3 (Float) Результат (Integer)
    900.0 + 100 / 400.0 900
```

```
EXPR Операнд 1 (Double) + Операнд 2 (Integer) / Операнд 3 (Float) Результат (Double)
    900.0 + 100 / 400.0 900.25
```

Объявление и видимость переменных

Объявления переменных и констант могут выполняться только в некоторых определенных точках кода GPL. Можно определить следующие переменные:

- Глобальные переменные модуля
- Глобальные переменные группы
- Локальные (только переменные)
- Глобальные переменные библиотеки

Максимальное число объявляемых глобальных переменных (модуля и группы) составляет 2048.

Можно определить *модификаторы*, придающие переменным дополнительные характеристики.

Глобальные переменные модуля

Глобальные переменные модуля содержатся в определенном файле, доступ к которому осуществляется путем выбора пункта меню **Файл->Открыть глобальные переменные**.

Объявление, как говорилось в предыдущих параграфах, выполняется путем указания имени переменной, за которым следует ключевое слово "AS", за которым следует тип данных (или типы данных в случае матрицы).

Эти переменные видимы непосредственно кодом других групп.

Глобальные переменные группы

Глобальные переменные группы определяются в начале кода, относящегося к группе. Они должны объявляться до функций GPL.

Эти переменные видны непосредственно всем кодом, содержащимся внутри группы. Кроме того, можно расширить видимость этих переменных за пределы группы, объявляя их как "общедоступные" переменные.

Общедоступные переменные недоступны напрямую извне группы. Для доступа к ним необходимо использовать их имя, указывая перед ним имя группы, к которой они относятся. Например, если необходимо изменить общедоступную переменную "сдвиг" группы "оси" из кода группы "главная", следует написать "SETVAL 10 оси.сдвиг".

Для объявления глобальной переменной группы используется тот же синтаксис, который используется для глобальных переменных модуля. Основное различие состоит в определении общедоступных переменных.

Для определения одной или нескольких общедоступных или индивидуальных переменных используются ярлыки "Public:" и "Private:". Например:

```
Public:
    сдвиг as double
    скорость as float
Private:
    инструмент as integer
```

Локальные переменные

Локальные переменные объявляются в теле функции. Они должны объявляться до любой другой инструкции за исключением объявления параметров функции.

Локальные переменные доступны только внутри функции.

Эти переменные создаются со значением 0 (выделяется необходимая память) только в начале выполнения функции и уничтожаются (память освобождается) по завершении выполнения. Глобальные переменные создаются при инициализации модуля и всегда видны в "Диагностике".

Объявление локальной переменной соответствует уже рассмотренному синтаксису, но перед ним должно приводиться ключевое слово "LOCAL".

Например:

```
Function обработка
local координата_центра as double
movabs X,координата_центр
fret
```

Глобальные переменные библиотеки

Глобальные переменные библиотеки содержатся в [библиотеках кода GPL](#). Они аналогичным глобальным переменным группы.

Модификаторы

Модификаторы: READONLY

Глобальные переменные модуля и глобальные переменные группы могут объявляться как READONLY.

Переменная readonly - это переменная, значение которой не может быть изменено кодом GPL, но может быть изменено "извне", т. е. из архива технологических параметров Albatros.

Архив технологических параметров - это база данных, в которой хранятся значения, характеризующие станок, которые могут меняться со временем после изменений в станке или проведения экстренного техобслуживания. Эта информация обычно вводится в матрицу GPL при инициализации блока управления.

Примером информации этого типа являются сдвиги участков обработки или размеры и технологические параметры инструментов.

При объявлении этих переменных как `readonly` мы защищаем себя от случайных изменений информации, которая, как правило, не должна меняться в ходе нормальной работы станка.

Максимальный размер переменной `readonly` составляет 128 кбайт.

Объявление переменной как `readonly` должно выполняться со следующим синтаксисом:

```
readonly ИмяПеременной as тип
```

Модификаторы: **NONVOLATILE**

Переменные, объявленные переменными класса **NONVOLATILE**, записываются в энергонезависимой ОЗУ (с аккумулятором), а не в обычной ОЗУ. Соответственно, сохраненные в этих переменных значения не утрачиваются при выключении ЧПУ.

Объявление переменной как `nonvolatile` должно выполняться со следующим синтаксисом:

```
nonvolatile ИмяПеременной as тип
```

Например:

```
nonvolatile Областисдвига[2] as double:сдвигX double:сдвигY double:сдвигZ
```

"Nonvolatile" могут определяться только глобальные переменные группы и модуля.

Максимальный суммарный размер переменных, записанных в энергонезависимую ОЗУ, составляет 65536 байт. Максимальный размер отдельной энергонезависимой матрицы составляет 1024 байт.

Задание RANGE

При объявлении переменной можно задать диапазон ее значений. Однако пока не выполняется какого-либо контроля соблюдения пределов на этапе выполнения, выполняется только контроль компилятором в случае, если задаются постоянные значения (напр., для инициализации переменной).

Таким образом, основным преимуществом является некоторое самодокументирование кода.

Определение диапазонов осуществляется по следующему синтаксису:

```
ИмяПеременной Range: минзнач..максзнач AS тип
```

Например:

```
НомерИнструмента Range:1..100 as integer
```

Права Считывания / Записи

Права считывания и записи позволяют указывать [минимальный уровень доступа](#) к системе, требуемый для отображения (право считывания) и изменения (право записи) значения в ней.

Синтаксис следующий:

```
ИмяПеременной Read=S Write=M AS тип
```

Ключевые слова, используемые для определения прав, следующие:

- READ считывание
- WRITE запись

Могут задаваться следующие значения:

- U или USER пользователь
- S или SERVICE сервисная служба
- M или MANUFACTURER изготовитель
- T или TPA tpa

Значения по умолчанию:

- READ считывание для сервисной службы (S или SERVICE)
- WRITE запись для изготовителя (M или MANUFACTURER) и Tpa (T или TPA)

10.1.4 Константы

GPL предусматривает 4 типа констант:

- Integer

- Double
- Char
- Строка

Константы типа Char объявляются при помощи кавычек следующим образом:

```
Const COD = 'A'
```

Константы типа String (строковые) объявляются при помощи двойных кавычек следующим образом:

```
Const MSG = "Начало обработки"
```

Константы типа Integer и типа Double объявляются при помощи следующего синтаксиса:

```
Const PI = 3.14
Const MSGBOX = 12
```

Для констант типа integer предусматривается также бинарная и шестнадцатеричная запись:

```
Const MASK = $11001001b ; бинарная
Const MASK = $F5h      ; шестнадцатеричная
```

Константы группы и библиотеки также могут быть общедоступными или частными. Синтаксис аналогичен предусмотренному для переменных.

Пример:

```
Public:
Const PI = 3.14
Const MSGBOX = 12
Private:
Const MASK = $11001001b
```

ПРИМЕЧАНИЕ: не существует констант типа Float. Дробные значения должны обязательно объявляться как Double. Иногда при этом могут подаваться предупреждения компилятора (при использовании инструкций GPL, оптимизированных для использования типов Float).

Можно определить константы как результат выражений расчетов, используя следующий синтаксис:

```
Const a = 10
Const b = 20
Const c = a + b
```

Допустимые операторы - те же, которые используются в инструкции [EXPR](#).

Заданные константы с предустановленным значением

В языке GPL предусматривается несколько предварительно заданных констант. Они могут использоваться напрямую, без необходимости их определения.

Заданные константы и соответствующие значения следующие:

ON	1
OFF	0
UP	+1
DOWN	-1
POSITIVE	+1
NEGATIVE	-1
CW	1
CCW	0

TRUE	1
FALSE	0
NOWAIT	0
WAIT	1
WAITACK	2
STORE	1
NOSTORE	0
NOPLACE	0
COM1	0
COM2	1
COM3	2
COM4	3
COM5	4
COM6	5
COM7	6
COM8	7
NOPARITY	0
ODDPARITY	1
EVENPARITY	2

Заданные константы с предустановленным значением при запуске Albatros

Язык GPL включает несколько предопределенных констант, значение которых определяется при запуске Albatros. Их можно использовать в инструкции [IFDEF](#).

_ID_MODULE	номер текущего модуля. Номер модуля от 0 и 15.
_REMOTE_MODULE	тип модуля. Константа 1, если модуль удаленный, равна 0, если модуль локальный, она не определена, если модуль не настроен в Конфигурации системы.
_VER_MAJOR	главный номер версии Albatros. Если версия Albatros - 3.2.1, главное значение версии - 3.
_VER_MINOR	младший номер версии Albatros. Если версия Albatros - 3.2.1, младшее значение версии - 2.
_VER_REVISION	номер ревизии Albatros. Если версия Albatros - 3.2.1, значение ревизии - 1.
_VER_SP	строка, описывающая пакет обновления, если он установлен, например «Пакет обновления 1f», в противном случае она не определена.
_VER_FULL	полный номер версии. В случае версии 3.2.1 значит \$00030201H.

10.1.5 Ключевые слова

Ключевые слова - это идентификаторы с зарезервированным применением, которые не могут использоваться иным образом.

Предусматриваются следующие ключевые слова:

Все имена инструкций GPL	Описание всех инструкций GPL см. в части "Инструкции" руководства
Все типы данных	См. Переменные
Параметры типа устройства	См. Параметры типа устройства
EXIST	используется в инструкции IFDEF для проверки существования группы. См. инструкцию IFDEF
NOTEXIST	используется в инструкции IFDEF для проверки несуществования группы. См. инструкцию IFDEF
LINKED	используется в инструкции IFDEF для подключения компиляции блоков кода, если устройство подключено в виртуальном-физическом пространстве. См. инструкцию IFDEF
UNLINKED	используется в инструкции IFDEF для подключения компиляции блоков кода, если устройство не подключено в виртуальном-физическом пространстве. См. инструкцию IFDEF
FUNCTION	Объявление функции. См. Функции
AS	Используется для объявления переменных. См. Переменные
PUBLIC	Атрибут функции. См. Функции
AUTORUN	Атрибут функции. Указывает, что функция запускается автоматически. См. Функции
R= или READ	Атрибут функции или переменной. Сигнализирует уровень доступа при считывании. См. Функции , Переменные и Права доступа
W=или WRITE	Атрибут функции или переменной. Сигнализирует уровень доступа при записи. См. Функции , Переменные и Права доступа
CONST	Позволяет задавать значимое имя, называемое символической константой, вместо номера, знака или строки. См. Переменные
READONLY	Атрибут глобальной переменной. См. Переменные
NONVOLATILE	Атрибут глобальной переменной. См. Переменные
PRIVATE	Атрибут функции. См. Функции
RANGE	Используется для определения интервала значений для переменной. См. Переменные
USER	Атрибут функции или переменной. Сигнализирует тип доступа. В этом случае - пользователь. См. Функции или Переменные .
SERVICE	Атрибут функции или переменной. Сигнализирует тип доступа. В этом случае - техобслуживание. См. Функции или Переменные .
MANUFACTURER	Атрибут функции или переменной. Сигнализирует тип доступа. В этом случае - изготовитель. См. Функции или Переменные .

ТРА

Атрибут функции или переменной. Сигнализирует тип доступа. В этом случае - ТРА. См. [Функции](#) или [Переменные](#).

10.1.6 Функции

Функции - это минимальный блок кода GPL. Инструкции GPL не могут включаться последовательно в файл, а должны группироваться в функции. Максимальное число функций объявляемых - 8191.

С точки зрения компилятора функциями являются все блоки кода GPL, начинающиеся со строки, первым словом которой является FUNCTION. Однако не существует ключевого слова, указывающего конец текста функции: функция заканчивается на строке, предшествующей началу другой функции, или в конце файла с функциями.

Синтаксис для определения функции:

```
FUNCTION      ИмяФункции Атрибуты
                Параметры
                Локальные переменные
                Список инструкций GPL
```

Кроме того, функция - это особый тип устройства Albatros. Она разделяет некоторые свойства с устройствами: однозначное имя (непереводимое), индикатор видимости (является ли устройство общедоступным или нет), [уровень доступа](#) при считывании и уровень доступа при записи (см. следующий параграф).

Права доступа

Права доступа позволяют определить минимальный уровень доступа к системе, обеспечивающий видимость (право считывания) и выполнение (право записи). Синтаксис следующий:

```
Function      ИмяФункции      READ=S WRITE=M
```

Права определяются ключевыми словами READ (считывание) и WRITE (выполнение) Значения, соответствующие уровням доступа, следующие:

- U или USER пользователь
- S или SERVICE сервисная служба
- M или MANUFACTURER изготовитель
- T или TPA tpa

Значения по умолчанию:

- READ считывание для сервисной службы (S или SERVICE) и пользователя (U или USER)
- WRITE запись для изготовителя (M или MANUFACTURER) и tpa (T или TPA)

Функции автозапуска

Функция типа автозапуска выполняется автоматически при инициализации станка.

Функции автозапуска имеют характеристику автоматического перезапуска при завершении после ошибки системы.

Синтаксис следующий:

```
Function      ИмяФункции      autorun
```

Таким образом, достаточно добавить модификатор "autorun" к объявлению функции.

Общедоступные функции

Обычно функция может направляться на выполнение (вызов) только из кода, находящегося внутри файла группы. Для того, чтобы функция могла быть направлена на выполнение кодом GPL другой группы, она должна быть определена **общедоступного типа**. Синтаксис для определения общедоступной функции:

```
Function      ИмяФункции      public
```

Таким образом, достаточно добавить модификатор "public" к объявлению функции.

Исключение составляют функции, относящиеся к интергруппе, которые всегда являются **общедоступными**.

Функции подгруппы

Функция может ассоциироваться с подгруппой путем простого указания имени подгруппы до имени функции. Имя подгруппы и имя функции должны отделяться точкой ".". Например, следующая функция относится к подгруппе "X" группы "Оси".

```
Function   X.обнуление
  local    vel as float
  movabs   X,100
  waitstill X
Fret
```

Асинхронные функции

Асинхронная функция вызывается автоматически блоком управления при генерировании события, с которым связана эта функция.

Такие события могут быть трех типов:

- Изменение состояния цифрового входа: инструкция ONINPUT
- Изменение состояния флагового бита или флажкового выключателя: инструкция ONFLAG
- Генерирование ошибки системы: инструкция ONERRSYS

При генерировании события вызывается функция (не как автономная задача, а в контексте задач, в которой была выполнена соответствующая инструкция ON...) как подразумеваемая FCALL, сразу после того, как завершит выполнение текущая инструкция.

Обычно асинхронные функции служат для управления экстренными ситуациями и должны быть очень быстрыми. Поэтому эти функции не могут использовать какие-либо инструкции GPL, а используют совокупность, обеспечивающую короткое время исполнения.

Функции с входными параметрами (параметрические)

Функция может иметь объявленные параметры на входе, но ни при каких обстоятельствах не возвращает значение.

Параметры могут рассматриваться как особые локальные переменные, значение которых инициализируется извне в момент отправки функции на выполнение. Параметры объявляются ключевым словом [PARAM](#) и имеют такой же синтаксис, который используется для локальных переменных. Параметры должны перечисляться в первых строках тела функции, до любой другой инструкции и до локальных переменных.

Существует два способа передачи параметров:

- **по значению:** передаются по значению все типы простых данных, т. е. CHAR, INTEGER, FLOAT и DOUBLE. Передача по значению обуславливает создание копии первоначального значения. Изменения, введенные в параметр, действуют только в контексте этой функции.
- **по ссылке:** передаются по ссылке структурированные параметры, т. е. МАССИВЫ, МАТРИЦЫ и СТРОКИ. Передача по ссылке обуславливает использование первоначальной переменной, соответственно, внесенные в параметр изменения действуют в контексте вызывающей функции. Эту характеристику можно использовать для возврата возвращаемых значений вызывающей функции.

Как правило, функция направляется на исполнение инструкцией FCALL. Если это параметрическая функция, после имени функции необходимо указать список значений, которые должны задаваться параметрам.

В примере ниже рассматривается параметрическая функция, выполняющая сверление. Координаты центра отверстия и скорость подачи оси Z передаются функции как параметры.

```
Function Сверление
  Param Qx as Double      ; координата X центра отверстия
  Param Qy as Double
  Param vel as Float      ; скорость подачи

  Movabs   X, Qx, Y, Qy
  Waitstill X,Y
  ....
Fret
```

Вывод этой функции для выполнения, например, отверстия на координатах (12.5, 25.7) со скоростью подачи 3 м в минуту, может выполняться следующим образом:

```
Fcall Сверление 12.5, 25.7, 3.0
```

Передаваемые функции параметры должны соответствовать числу и типу параметров, объявленных в вызываемой функции. Выполнение вызываемой функции возобновляется по завершении выполнения вызываемой функции.

В качестве параметра функции может также [объявляться устройство](#). Это позволяет писать универсальные функции, как например, функция обнуления, которая может использоваться со всеми осями, имеющимися на станке:

```
Function ОБНУЛЕНИЕ PUBLIC
    param    ось as Axis
    movabs   ось,100
```

```
Fret
```

```
Function MAIN
```

```
.....
```

```
Оси.Обнуление x
```

```
Fret
```

Функция обнуления относится группе Оси и объявляется PUBLIC (общедоступной) так, чтобы ее можно было видеть также из функций, объявленных в других группах. Функция Main вызывает функцию обнуления группы осей, указывая в качестве параметра на входе ось, которую необходимо переместить.

10.1.7 Параметры типа устройства

Параметры типа устройства - это особые переменные, позволяющие ссылаться на устройство станка. Данные этого типа могут использоваться **исключительно** в объявлении [параметров функции](#). Таким образом, нельзя объявлять переменные этого типа. Определение имен и других характеристик устройств остается в ведении Конфигурации системы.

В таблице ниже приведены типы устройства и соответствующие ключевые слова, которые должны использоваться для объявления соответствующих параметров.

Тип	Ключевое слово
Цифровой вход	INPUTDIG
Цифровой выход	OUTPUTDIG
Аналоговый вход	INPUTANALOG
Аналоговый выход	OUTPUTANALOG
Ось	AXIS
Таймер	TIMER
Счетчик	COUNTER
Флаговый бит	FLAGBIT
Флажковый выключатель	FLAGSWITCH
Флаговый порт	FLAGPORT
Входной порт	INPUTPORT
Выходной порт	OUTPUTPORT
Функция	FUNCTION
Общее устройство	DEVICE
Задача	TASK

В примере ниже объявляется и используется параметр типа оси:

```
Function проба
    Param ось as axis

    MovAbs   ось,100
    WaitStill ось
```

```
Fret
```

10.1.8 Многозадачность

Система Albatros является системой многозадачного типа, поэтому одновременно могут выполняться несколько задач GPL, где под задачей понимается процесс управления логической единицей (обычно - группой).

Предусматривается два типа задач: обычные задачи и задачи реального времени.

Обычные задачи

Многозадачность основана на кооперативном алгоритме, основанном на приоритетности. Это обеспечивает циклическое выполнение всех задач и изменение их приоритета. Алгоритм планирования предусматривает выполнение одной инструкции для каждой активной задачи (состояние выполнения). С каждой задачей связан приоритет, задаваемый инструкцией [SETPRIORITYLEVEL](#). Приоритет определяется целым числом от 0 (максимальный приоритет) до 255 (минимальный приоритет). Для задач с приоритетом 0

(ноль) выполняется одна инструкция при каждом цикле планирования, для задач с приоритетом 1 выполняется одна инструкция каждые два цикла планирования и так далее до задач с приоритетом 255, для которых выполняется одна инструкция каждые 256 циклов планирования.

Выполнение обычных задач осуществляется асинхронно относительно частоты обновления осей. Это означает отсутствие гарантии того, что функция GPL будет выполнена за время между двумя обновлениями состояния осей.

Задача определяется именем функции GPL, с которой начинается ее выполнение.

Выполнение задачи может начинаться:

- автоматически при инициализации системы: главная функция интергруппы и функции автозапуска.
- после выполнения инструкции [STARTTASK](#).
- после пуска ручного режима с графического интерфейса Albatros.

Каждая задача характеризуется внутренним состоянием:

RUNNING	задача выполняется
HOLD	задача приостановлена
BREAK	задача прервана отладчиком

Иерархия задач имеет древовидную структуру. Каждая задача создается другой задачей, для которой она является дочерней, что означает, что при завершении родительской задачи завершаются и все дочерние задачи.

Одновременно может выполнять максимум 500 задач.

Помните, что большое число выполняемых одновременно задач вызывает снижение скорости выполнения каждой отдельной задачи.

Если вы считаете, что создаваемое приложение может потребовать использования больше 200 задач, необходимо использовать соответствующее аппаратное обеспечение, например, модули Cn2128.

Задачи реального времени

Задачи реального времени отличаются от предыдущих тем, что на них не распространяется механизм планирования и они не имеют заданного приоритета, а полностью выполняются при каждом обновлении состояния осей (реальное время оси).

Абсолютно необходимо, чтобы выполнение этих задач завершалось в течение определенного времени, т. е. выполнение описанных ранее задач GPL приостанавливается на время выполнения задач реального времени.

Система выполняет проверки продолжительности выполнения задач реального времени и если они превышают максимально допустимое время, генерируется ошибка системы.

Поэтому в этих задачах не рекомендуется создавать бесконечные циклы (например, с использованием GOTO), циклы, которые, к тому же, не нужны, т. е. выполнение кода начинается сначала при каждой задаче реального времени осей.

Все также для предупреждения слишком длительного выполнения на задачи реального времени распространяются ограничения в применении некоторых инструкций GPL. Инструкции, использование которых не разрешено, - это инструкции, [не используемые на прерывании](#).

Рекомендуется использовать задачи реального времени только для тех действий, которые должны обязательно выполняться синхронно с обновлением координат осей. Для большинства действий управления лучше использовать обычные задачи.

Задачи реального времени запускаются инструкцией [STARTREALTIMETASK](#) и могут прерываться инструкцией [ENDREALTIMETASK](#). Одновременно можно подключить максимум 256 задач реального времени.

Больше не действует механизм наследственности, поэтому если задача, запустившая задачу реального времени, завершается, задача реального времени будет продолжать выполняться.

Локальные переменные, объявленные в задаче реального времени, инициализируются только при запуске задачи и затем сохраняют значение последнего выполнения.

Задачи реального времени не характеризуются состояниями, предусмотренными для обычных задач.

Можно выполнить отладку задачи реального времени, но в этом случае на все время выполнения отладки система автоматически переводит задачу в разряд "обычных задач".

Если в задаче реального времени выявляется ошибка системы, задача переводится в разряд обычных задач и переводится в состояние HOLD для выполнения ее анализа отладчиком.

10.1.9 Связь

Связь между GPL и внешним миром осуществляется тремя различными способами:

- SEND / RECEIVE
- Последовательная связь
- IPC

Send / Receive

Инструкции [SEND](#) и [RECEIVE](#) реализуют механизм связи, ориентированный на сообщение.

Связь может реализовываться внутри одного модуля (малополезная), между несколькими модулями одной линии или между модулями и супервизором Albatros или же при помощи приложений OLE.

Эта связь работает подобно почте: каждое сообщение имеет получателя, идентификатор направленной (или запрошенной) информации, саму информацию, а также некоторую служебную информацию. Albatros выполняет функцию сбора и сортировки этой информации и в некоторых случаях напрямую предоставляет запрошенную информацию.

Этот режим связи обычно используется для отправки программ обработки между супервизором и модулями управления для синхронизации работы станков одной линии и для сопряжения с внешними приложениями (сервер OLE).

Последовательная связь

Язык GPL предоставляет некоторую информацию, например, [COMREAD](#) и [COMWRITE](#), позволяющую направлять и получать данные через последовательные порты блока управления. Таким образом, можно обеспечивать сопряжение блока управления с внешними устройствами, например, инверторами, терминалами или ПЛК. Эти инструкции при правильном использовании позволяют реализовывать протоколы последовательной связи, например, MODBUS-RTU и т. д.

IPC

IPC, или Inter Process Communication, - это режим связи между процессами. В частности, этот режим позволяет определять область памяти, совместно используемую двумя или более процессами, которая может использоваться для обмена информацией.

Как правило, он используется, когда характеристики интерфейса OLE Albatros оказываются несоответствующими.

На стороне GPL связь IPC реализуется инструкциями [SENDIPC](#), [WAITIPC](#) и [TESTIPC](#). В случае локального модуля внешние процессы могут ссылаться на API, поставляемые RTX или на компонент COM [gplipc2.dll](#), поставляемый TPA, упрощающий его использование.

В случае удаленного модуля процессы, выполняющиеся в супервизоре, используют компонент COM [gplipc2net.dll](#).

Для получения большей информации обращаться на TPA.

10.1.10 Переменные, используемые в программировании

Большая часть инструкций была составлена так, чтобы можно было работать с различными типами переменных (CHAR, INTEGER, FLOAT, DOUBLE). Однако каждая инструкция имеет предпочтительный тип переменной, для которой она была оптимизирована. Для обеспечения максимальной отдачи при выполнении кода GPL рекомендуется использовать типы переменных, рекомендованных в описании каждой инструкции. Как правило, рекомендуется соблюдать указания следующей таблицы, в которой для основных используемых в программировании величин указываются соответствующие оптимальные типы переменных:

величина	тип
координата	<i>double</i>
скорость	<i>float</i>
время	<i>double</i>
счетчик	<i>integer</i>
значение порта / флагового порта	<i>integer</i>
таймаут	<i>double</i>
аналоговый вход / выход	<i>float</i>
направляющие косинусы	<i>double</i>
управляющий символ в строке	<i>char</i>
ускорения / замедления	<i>integer</i>

10.1.11 Оси

Обычно именем "ось" обозначается электромеханическая система, целью которой является контролируемое перемещение органа станка.

Эту систему можно описать с точки зрения образующих ее элементов, которые могут быть разделены в зависимости от их технологических характеристик.

Таким образом, мы получим механические компоненты, среди которых:

- несущая конструкция
- направляющие
- подшипники
- винт + подшипники с круговым движением шариков

задачей которых является противодействие действующим усилиям, снижение трения, преобразование движения из вращательного в поступательное и пр.

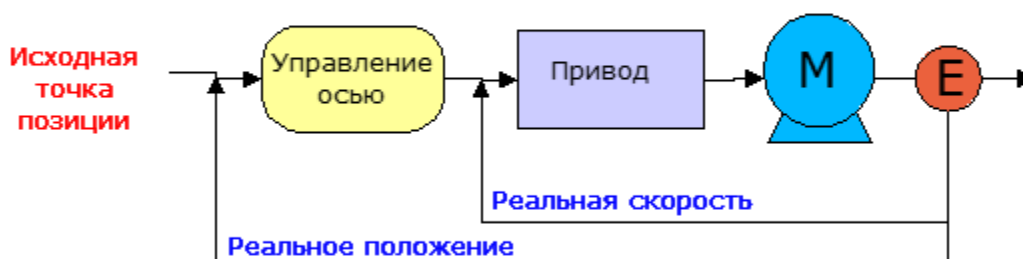
Мы получим также электрические и электронные компоненты, среди которых:

- двигатель
- конечные выключатели
- кодер
- тахогенератор

задачей которых является обеспечение мощности, необходимой для перемещения и определения состояния системы.

Эти элементы связаны между собой так, чтобы обеспечить контролируемое выполнение движений.

Схема управления с обратной связью

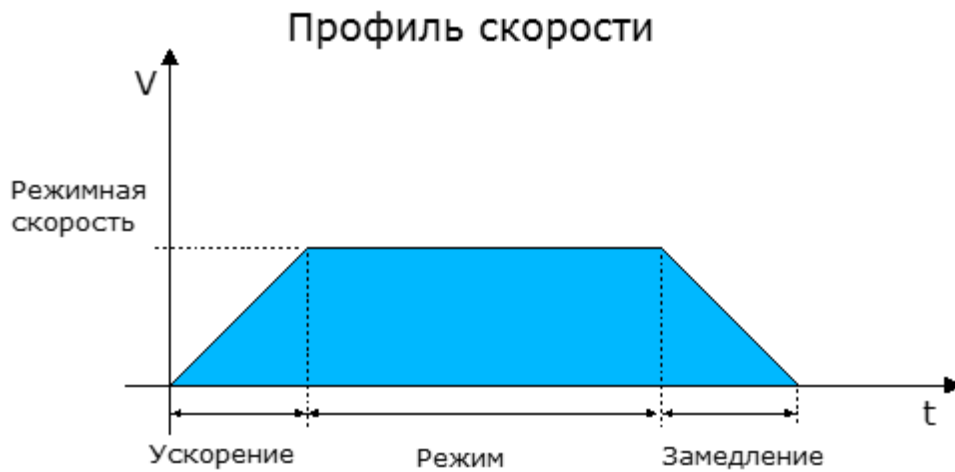


Целью ЧПУ является контроль положения и перемещений оси.

Движение оси можно разложить на 5 этапов:

Ускорение	начальная фаза, в ходе которой ось постепенно увеличивает скорость до достижения нужной
Режим	промежуточный этап, в ходе которого ось перемещается с постоянной скоростью (этот этап может отсутствовать, если расстояние, которое требуется пройти, меньше расстояния, проходимого при ускорении и замедлении)
Замедление	этап, в ходе которого ось постепенно уменьшает скорость до 0
Окно	этап ожидания, пока погрешность контура снизится до значения, указанного в конфигурации как "Окно прибытия на координату"
Координата	конец движения

В конце движения ось должна быть позиционирована внутри интервала, называемого "Окном прибытия на координату" (определяет допуск позиционирования оси). Если этого не происходит в течение 5 секунд с предусмотренного окончания движения, система генерирует ошибку системы "движение не завершено".



Для каждого движения ЧПУ вычисляет профиль скорости, как показано на рисунке, затем вычисляет целевые координаты, разделяя профиль скорости на временные интервалы, равные времени обновления оси и вычисляя площадь каждой части. Площадь представляет собой инкремент координаты, которую ось должна достичь за этот временной интервал для соблюдения указанного выше профиля скорости.

Управление осью осуществляется при помощи PID-контроллера, который обеспечивает "замыкание контура положения", т. е. обеспечения эталона скорости привода, вычисленного в зависимости от координаты, до которой необходимо дойти (целевая координата) и реальной координаты, считанной кодером. Разница между реальной координатой и целевой координатой называется **Ошибка кольца**.

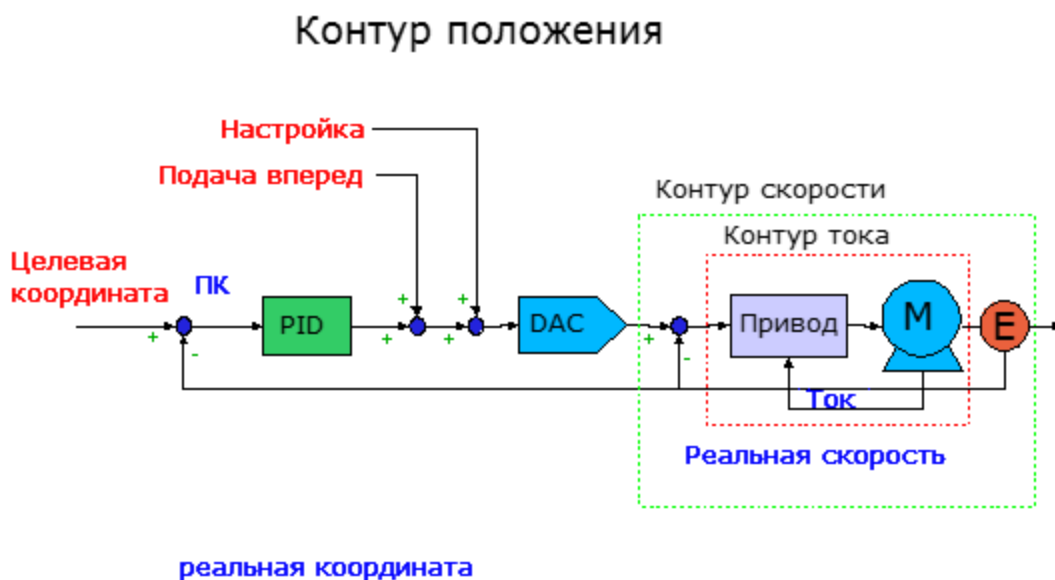


Схема управления осью Albatros

10.1.12 Корректоры линейности

Таблица корректоров линейности оси представляет собой матрицу, имя которой *ИмяГруппы.ИмяПодгруппы.ИмяОси#correctors* или *ИмяГруппы.ИмяОси#correctors* и может использоваться во всех инструкциях, в которых обращается к матрицам, матричным элементам и строкам матрицы.

Количество столбцов в матрице соответствует количеству осей, заданных в окне корректоров линейности в конфигурациях осей.

Автокорректоры вставляются в первый столбец. Все значения коррекции относятся к типу float. Общее количество строк матрицы можно получить с помощью инструкции GPL [LASTELEM](#).

Эти матрицы доступны как для чтения, так и для записи, и в случае изменения их значения немедленно используются для поправки размера, только если поправка была разрешена.

Пример:

```
Function ReadCorr
local i as integer
local j as integer
local row as integer
local column as integer
local firstvalue as float

; чтение первого значения самокоррекции оси AX
firstvalue = X.AX#correctors[1][1]
; количество осей, в окне корректоров линейности
setval 3 column
lastelem X.AX#correctors row
; увеличивает все поправки на постоянное значение
for i 1 row
  for j 1 column
    X.AX#correctors[i][j] = X.AX#correctors[i][j] + 0.025
  next
next
fret
```

10.1.13 Управление сообщениями на языке

Как описывалось в главе "Состав системы", Albatros поддерживает [отображение текстовых сообщений на нескольких языках](#).

Эта поддержка обеспечивается за счет использования TraLangs, программы не входящей в Albatros, которая управляет архивами сообщений. Эта программа помогает в переводе сообщений на различные языки.

Текст, связанный с Ошибками Цикла и Сообщениями

Особым типом текстов, которые обычно используются в Albatros, являются сообщения и ошибки цикла, генерированные кодом GPL.

Они обычно определяются разработчиком кода GPL при написании кода. Для облегчения работы программиста редактор GPL позволяет вводить текст сообщения непосредственно из Albatros, т. е. без необходимости использования TraLangs.

Еще одной возможностью для управления сообщениями на языке является использование инструкции GPL [DEFMSG](#).

10.1.14 Управление ошибками системы

При возникновении [ошибки системы](#) (См. главу **Ошибки системы->Введение в ошибки системы**) нормальным поведением блока управления является прекращение всех задач: управление ошибками системы позволяет избежать прекращения задач, для которых оно было подключено.

Управление ошибками системы, генерируемыми в результате сбоя, незагруженности стека и переполнения стека, осуществляется непосредственно блоком управления без вызова функции управления ошибками системы: задача переводится в состояние HOLD.

Функция управления ошибками

В коде GPL должны быть определена одна или несколько функций для анализа ошибки системы и для соответствующего определения соответствующих действий для обеспечения безопасности станка.

Вывываемая инструкция передается как параметр инструкции GPL [ONERRSYS](#). (См. главу **Язык GPL->Инструкции->Управление потоком->ONERRSYS**).

При возникновении ошибки системы генерировавшая ошибку задача приводится в состояние HOLD. Если задачи автозапуска генерируют ошибки системы, они запускаются снова только если ошибка системы не является FAULT (сбоем).

Если ошибка системы генерируется без номера задачи, приводится в состояние HOLD текущая задача.

10.2 Специальные функции

10.2.1 Персонализация перемещения осей

Графический интерфейс системы Albatros предусматривает возможность перемещения осей в ручном режиме, и предоставляет графическую поддержку для их калибровки.

Перемещение осей в ручном режиме осуществляется при помощи планшета перемещения осей, а калибровка может выполняться при помощи планшета калибровки. Оба этих способа можно вызывать из функции Диагностика и из обзорных экранов.

В обоих случаях управление перемещением осей осуществляется группой функций GPL, исполнение которых остается невидимым для пользователя.

Система имеет заданную совокупность этих функций, которые подходят для большинства случаев. Однако может потребоваться персонализировать их, например, для ввода ограничений для перемещений осей, связанных с состоянием станка или для управления вспомогательными устройствами, например, тормозами оси.

Персонализация осуществляется за счет создания двух функций GPL для каждой оси: одна для перемещения в ручном режиме, другая - для калибровки. Эти функции являются дополнительными, поэтому при их обнаружении системой они используются, в противном случае используются стандартные функции. Кроме того, предусматривается также частичная персонализация функций перемещения.

Перемещение в ручном режиме

Персонализированные функции *перемещения в ручном режиме* должны отвечать следующим требованиям:

- Функция должна относиться к той же группе или подгруппе, к которой относится соответствующая ось.
- Имя функции должно быть **MoveAx#имя_оси**, где имя_оси должно заменяться именем оси, определенной в конфигурации. Например, для оси X имя функции будет следующим: MoveAx#X .
- Функция должна предусматривать следующие параметры:
 1. **Требуемое действие**. Это может быть движение на абсолютную координату, движение с приращением, стоп и т. д. Действия обозначаются целым числом, компилятор GPL предусматривает определенные связанные с действиями константы:

<code>_MOVAXABS</code>	перемещение на абсолютную координату
<code>_MOVAXINC</code>	перемещение на инкрементальную координату
<code>_MOVAXSET</code>	задание координаты
<code>_MOVAXFREE</code>	задание свободного состояния
<code>_MOVAXNORMAL</code>	задание нормального состояния
<code>_MOVAXEND</code>	сброс состояния оси в конце движения (не служит для остановки оси)
 2. **Результат**. Служит для системы, чтобы знать, выполняется ли управление требуемым действием персонализированной функцией. Если управление действием не выполняется, используется соответствующая стандартная функция. Таким образом, это возвращаемое значение, которое должно задать персонализированная функция, и с этой целью оно определяется как параметр, переданный для ссылки (массив только одного элемента).
 3. **Скорость**. Значимо только когда требуемое действие является движением, и задает его скорость.
 4. **Координата**. Значимо только для действий движения и задания координаты.

Пример персонализированной функции перемещения:

```
Function MoveAx#X
  param action as integer
  param result[1] as integer
  param speed as float
  param position as double

  setval      1,result[1]

  select action
```

```

case _MOVAXEND
    fcall EndMovement
case _MOVAXABS
    fcall AbsMovement X, speed, position
case _MOVAXINC
    fcall IncMovement X, speed, position
case _MOVAXSET
    fcall PositionSet X, position
case _MOVAXFREE
    fcall FreeAxis
case _MOVAXNORMAL
    fcall NormalAxis
case else
    call Unknown
endselect

fret

Unknown:
    setval    0, result[1]
    ret

```

Функции EndMovement, AbsMovement и т. д. (эти имена не являются обязательными) должны реализовывать персонализированное управление требуемыми действиями. Для облегчения работы программиста приводим [стандартные функции перемещения](#), которые могут служить основой для разработки персонализированных функций.

Калибровка

Персонализированные функции калибровки должны отвечать следующим требованиям:

- Функция должна относиться к той же группе или подгруппе, к которой относится соответствующая ось.
- Имя функции должно быть **CalibAx#имя_оси**, где имя_оси должно заменяться именем оси, определенной в конфигурации. Например, для оси X имя функции будет следующим: CalibAx#X.
- Функция должна предусматривать следующие параметры:
 1. **Требуемое действие.** Это может быть движение точка-точка или интерполированное движение.
 2. **Результат.** Служит системе для того, чтобы знать, выполняется ли управление требуемым действием персонализированной функцией. Если управление действием не выполняется, используется соответствующая стандартная функция.
 3. **Скорость.** Скорость движения при калибровке
 4. **Положительная координата.** Положительная координата возвратно-поступательного движения калибровки.
 5. **Отрицательная координата.** Отрицательная координата возвратно-поступательного движения калибровки.
 6. **Ожидание.** Ожидание между двумя последовательными движениями

ПРИМЕЧАНИЕ: необходимо помнить, что в некоторых случаях действия, выполненные на планшете калибровки, обуславливают выполнение функции перемещения оси. Например, в конце движения калибровки (при нажатии кнопки стоп) выполняется операция сброса оси, для которой вызывается персонализированная функция перемещения оси параметром действия, заданным на _MOVAXEND. Таким же образом, при изменении координаты оси с планшета калибровки вызывается функция перемещения оси параметром действия, заданным на _MOVAXSET.

Пример функции персонализированной калибровки:

```

Function CalibAx#X
    param action as integer
    param result[1] as integer
    param speed as float
    param PosPosition as double
    param NegPosition as double
    param waitTime as float

```

```

setval      1,result[1]

select action
case _CALAXPP
    fcall PPCalibration X, speed, PosPosition, NegPosition, _
        waitTime
case _CALAXINT
    fcall IntCalibration X, speed, PosPosition, NegPosition,_
        waitTime
case else
    call   Unknown
endselect

fret

Unknown:
setval      0, result[1]
ret

```

Функции PPCalibration и IntCalibration (эти имена не являются обязательными) должны реализовывать персонализированное управление требуемыми действиями. Для облегчения работы программиста приводим стандартные функции калибровки, которые могут служить основой для разработки персонализированных функций.

Взаимодействие с окном Перемещения оси в ручном режиме

Функции взаимодействия с окном перемещения в ручном режиме должны отвечать следующим требованиям:

- Функция должна относиться к той же группе или подгруппе, к которой относится соответствующая ось.
- Имя функции должно быть **MoveAx#имя_оси#Действие**, где имя_оси должно заменяться именем оси, определенной в конфигурации, а Действие может принимать одно из следующих определений:

OPEN	указывает, что пользователь только что открыл окно перемещения оси
CLOSE	указывает, что пользователь собирается закрыть окно перемещения оси
ACTIVE	сигнализирует, что окно перемещения оси активно
INACTIVE	сигнализирует, что окно перемещения оси неактивно
JOG	указывает, что задано движение для смещения, управляемое рабочим циклом оператором
STEP	указывает, что задано движение со смещением с заданным шагом
ABSOLUTE	указывает, что задано движение со смещением на заданную координату

Например, если было открыто окно перемещения оси для оси X, будет вызываться функция с именем MoveAx#X#Open.

Изменения окна Перемещения оси в ручном режиме

К окну перемещения осей можно добавить до 4 кнопок. В той же группе или подгруппе, в которой определена соответствующая ось, должны определяться функции GPL с фиксированным именем MoveAx#ИмяОси#BUTTONтекст. ИмяОси является именем соответствующей оси, а текст указывает текст, который будет отображен на кнопке. Текст может содержать знак '&' для введения ускорителя клавиатуры. Если текст начинается с номера от 1 до 4, этот номер рассматривается как положение, в которое должна вводиться кнопка в окне перемещения оси. Текст кнопки может переводиться с вводом DEFMSG в группу, в которой находится ось, имеющая идентификатор MOVEAX#BUTTONтекст. Нажатие персонализированной кнопки обуславливает выполнение связанной функции GPL. Не выполняется какого-либо ожидания конца выполнения или какой-либо проверки начала выполнения функции.

10.2.2 Стандартные функции перемещения и калибровки

Ниже представлены стандартные функции, используемые планшетом перемещения оси и планшетом калибровки.

Функции разные в зависимости от типа соответствующей оси: подсчета, шаговая и т. д. Эти функции могут [персонализироваться](#).

Стандартные функции ручного перемещения

Перемещение на абсолютную координату

```

; для шаговых осей
Function AbsMovement
    param axisname as axis
    param speed as float
    param position as double

    ifstill      axisname goto move
    fret

move:
    setvel      axisname, speed
    movabs     axisname, position
    waitstill   axisname
    fret

```

```

; для всех остальных типов оси
Function AbsMovement
    param axisname as axis
    param speed as float
    param position as double

    iftarget    axisname goto move
    ifstill     axisname goto move
    fret

move:
    setvel      axisname, speed
    movabs     axisname, position
    waitstill   axisname
    fret

```

Движение с приращением

```

; для шаговых осей
Function IncMovement
    param axisname as axis
    param speed as float
    param position as double

    ifstill     axisname goto move
    fret

move:
    setvel      axisname, speed
    movinc     axisname, position
    waitstill   axisname
    fret

```

```

; для всех остальных типов оси
Function IncMovement
    param axisname as axis
    param speed as float
    param position as double

    iftarget    axisname goto move
    ifstill     axisname goto move
    fret

move:
    setvel      axisname, speed
    movinc     axisname, position
    waitstill   axisname
    fret

```


Задание координаты

```

; для осей подсчета
Function PositionSet
    param axisname as axis
    param position as double

    setquote    axisname, position
    fret

; для шаговых осей
Function PositionSet
    param axisname as axis
    param position as double

    ifstill     axisname goto set
    fret
set:
    setquote    axisname, position
    fret

; для всех остальных типов оси
Function PositionSet
    param axisname as axis
    param position as double

    iftarget    axisname goto set
    ifstill     axisname goto set
    fret
set:
    setquote    axisname, position
    fret

```

Задание свободного состояния

```

Function FreeAxis
    param axisname as axis

    free        axisname
    fret

```

Задание нормального состояния

```

Function NormalAxis
    param axisname as axis

    normal     axisname
    fret

```

Стандартные функции калибровки**Калибровка в точка-точка**

```

; для шаговых осей
Function PPCalibration
    param axisname as axis
    param speed as float
    param PosPosition as double

```

```

    param NegPosition as double
    param waitTime as float

    setvel      axisname, speed
loop:
    movabs     axisname, PosPosition
    waitstill  axisname
    delay      waitTime
    movabs     axisname, NegPosition
    waitstill  axisname
    delay      waitTime
    goto       loop
    fret

; для всех остальных типов оси
Function PPCalibration
    param axisname as axis
    param speed as float
    param PosPosition as double
    param NegPosition as double
    param waitTime as float

    setvel      axisname, speed
loop:
    movabs     axisname, PosPosition
    waitstill  axisname
    ifquotet   axisname,<>,PosPosition goto exit
    delay      waitTime
    movabs     axisname, NegPosition
    waitstill  axisname
    ifquotet   axisname,<>,NegPosition goto exit
    delay      waitTime
    goto       loop
exit:
    fret

```

Калибровка в интерполяции

```

Function IntCalibration
    param axisname as axis
    param speed as float
    param PosPosition as double
    param NegPosition as double
    param waitTime as float

    setveli    axisname, speed
loop:
    linearabs  axisname, PosPosition
    waitstill  axisname
    ifquotet   axisname,<>,PosPosition goto exit
    delay      waitTime
    linearabs  axisname, NegPosition
    waitstill  axisname
    ifquotet   axisname,<>,NegPosition goto exit
    delay      waitTime
    goto       loop
exit:
    fret

```

10.2.3 Функция OnUIEnd#

Функция "OnUIEnd#" выполняется, при наличии, Albatros до завершения всех задач модуля. Она должна определяться в файле функций интергруппы. Функция "OnUIEnd#" имеет максимальное время выполнения в 2 секунды, после чего Albatros завершит все задачи.

Максимальное время ожидания завершения выполнения этой функции может конфигурироваться в Тра.ini, в разделе [Albatros], в пункте Timeout.OnUIEnd=*значение*, где *значение* выражено в миллисекундах и не может быть больше 60000.

10.2.4 Функция OnUIPlugged#

Функция "OnUIPlugged#" выполняется тогда, когда необходимо знать, например, при включении установки, была ли связь Albatros с удаленным модулем. Эта функция должна быть определена в интергруппе.

10.2.5 Функция OnUIUnPlugged#

Функция "OnUIUnplugged#" выполняется до завершения выполнения Albatros (т. е. до того, как Albatros отключится от модуля). Эта функция должна быть определена в интергруппе. Albatros выполняет эту функцию в течение максимум 2 секунд, в ходе которых считываются:

- ошибки цикла
- ошибки системы
- сообщения

В конце выполнения Albatros закрывается.

Максимальное время ожидания завершения выполнения этой функции может конфигурироваться в Тра.ini, в разделе [Albatros], в пункте Timeout.OnUIUnplugged=*значение*, где *значение* выражено в миллисекундах и не может быть больше 60000.

10.3 Инструкции

10.3.1 Условности

Следующие страницы организованы в виде карт и для каждой инструкции описывают:

- синтаксис
- описание аргументов: тип данных и допустимые значения
- описание функционирования
- возможные примечания
- возможные примеры

Однотипные инструкции были сгруппированы для облегчения обучения и консультации.

10.3.2 Типология инструкций языка GPL

Этот язык состоит из инструкций, которые могут быть сгруппированы следующим образом:

Инструкции для управления входами/выходами

GETFEED	считывает скорость подачи override
INPANALOG	считывает аналоговый вход
INPFLAGPORT	считывает флаговый порт
INPPORT	считывает цифровой порт
MULTIINPPORT	считывает до 4 выходных портов
MULTIOUTPORT	задает до 4 выходных портов
MULTIRESETFLAG	задает несколько флажков на 0
MULTIRESETOUT	задает несколько выходов на 0

MULTISETFLAG	задает несколько флажков на 1
MULTISETOUT	задает несколько выходов на 1
MULTIWAITFLAG	ожидает состояние флагового бита или флажкового выключателя
MULTIWAITINPUT	ожидает состояние нескольких входов
OUTANALOG	изменяет аналоговый выход
OUTFLAGPORT	меняет флаговый порт
OUTPORT	меняет цифровой порт
RESETFLAG	задает флажок на 0
RESETOUT	задает выход на 0
SETFLAG	задает флажок на 1
SETOUT	задает выход на 1
WAITFLAG	ожидает состояние флагового бита или флажкового выключателя
WAITINPUT	ожидает состояние входа
WAITPERSISTINPUT	ожидает устойчивое состояние входа

Инструкции для управления Осями

CHAIN	сцепляет одну ось с другой
CIRCABS	абсолютная круговая интерполяция
CIRCINC	круговая интерполяция с приращением
CIRCLE	выполняет окружность
COORDIN	координированное движение осей
DISABLECORRECTION	отключает линейную коррекцию для указанной оси
EMERGENCYSTOP	вызывает аварийный останов осей
ENABLECORRECTION	подключает линейную коррекцию для указанной оси
ENDMOV	завершает перемещение оси
FASTREAD	быстрое считывание координат осей
FREE	приводит ось в свободное состояние
HELICABS	абсолютная винтовая интерполяция
HELICINC	винтовая интерполяция с приращением
JERKCONTROL	подключает или отключает блок управления на интерполированных движениях
JERKSMOOTH	соединяет с непрерывным ускорением и скоростью профили скорости осей в ходе движений контурирования
LINEARABS	абсолютная линейная интерполяция
LINEARINC	линейная интерполяция с приращением
MOVABS	перемещение осей в абсолютном режиме
MOVINC	перемещение осей в режиме с приращением
MULTIABS	многоосевая линейная интерполяция в абсолютном режиме
MULTIINC	многоосевая линейная интерполяция в режиме с приращением
NORMAL	выводит ось из свободного состояния
RESRIFLOC	восстанавливает изначальные ссылки
SETINDEXINTERP	присваивает оси переменную для подсчета выполненных блоков интерполяции
SETLABELINTERP	присваивает оси переменную для идентификации блока смещения
SETPFLY	обнуление налету
SETPFLYCHAINSTRAT	проверяет поведение ведомой оси при инструкции setpfly на главной оси
SETPZERO	обнуление на отметке нуля
SETPZEROCHAINSTRAT	проверяет поведение ведомой оси при инструкции setpzero на главной оси
SETQUOTE	задает координату
SETQUOTECHAINSTRAT	проверяет поведение ведомой оси при инструкции setquote на главной оси
SETRIFLOC	задает пространственные эталоны
SETTOLERANCE	задает значения допуска для линейной интерполяции
START	возобновляет перемещение оси
STARTINTERP	принудительно задает начало интерполяции
STOP	приостанавливает перемещение оси
SWITCHENC	позволяет заменять кодер одной оси кодером другой оси
WAITACC	ожидание ускорения оси
WAITCOLL	ожидание, когда ось зайдет за координату, с которой начинать проверку наличия столкновения
WAITDEC	ожидание замедления оси
WAITREG	ожидание режима оси
WAITSTILL	ожидание, когда конечная координата будет равна целевой координате
WAITTARGET	ожидание, когда ось будет на целевой координате
WAITWIN	ожидание, когда ось будет в окне

Инструкции для управления Параметрами осей

Считывание/Запись

[DEVICEID](#) записывает логический адрес связанный с устройством

[GETAXIS](#) считывает одно данное или больше оси

Движение точка-точка

[SETACC](#) задание ускорения

[SETDEC](#) задание замедления

[SETDERIV](#) задание коэффициента производного действия

[SETFEED](#) задание скорости подачи точка-точка

[SETFEEDF](#) задание подачи вперед

[SETFEEDFA](#) задание подачи вперед ускорения

[SETINTEG](#) задание коэффициента интегрального действия

[SETMULTIFEED](#) задание процентного значения скорости подачи override указанных осей

[SETPROP](#) задание коэффициента пропорционального действия

[SETSLOPE](#) задание типа линейной функции на движениях в быстром режиме

[SETVEL](#) задание скорости

Интерполированное движение

[LOOKAHEAD](#) задание опережающего просмотра интерполяции

[SETACCI](#) задание ускорения для интерполяции

[SETACCLIMIT](#) подключение и отключение автоматического расчета скорости режима интерполяции

[SETACCSTRATEGY](#) выбор типа ускорения

[SETAXPARTYPE](#) изменение используемого набора параметров оси

[SETCONTORNATURE](#) задание угла контурирования

[SETDECI](#) задание замедления для интерполяции

[SETDERIVI](#) задание коэффициента производного действия интерполяции

[SETFEEDFAI](#) задание подачу вперед ускорения при интерполяции

[SETFEEDI](#) задание скорость подачи при интерполяции

[SETFEEDFI](#) задание подачу вперед при интерполяции

[SETINTEGI](#) задание коэффициента интегрального действия при интерполяции

[SETPROPI](#) задание коэффициента пропорционального действия интерполяции

[SETSLOPEI](#) задание типа линейной функции на движениях при интерполяции

[SETSLOWPARAM](#) изменение параметров, необходимых для расчета скорости замедления в случае, если активна функция замедления при контурировании

[SETVELI](#) задание скорости интерполяции

[SETVELILIMIT](#) задание отдельных составляющих скорости указанной оси

Координированное движение

[SETFEEDCOORD](#) задание процентного значения максимального мгновенного изменения скорости подачи оси

[SETOFFSET](#) задание сдвига координаты

Объединённое движение

[RATIO](#) задание отношения сцепления ведомой оси относительно соответствующей главной

[SETDYNRATIO](#) динамическое изменение отношения сцепления при движении главной оси, отношение сцепления

Общие параметры

[DYNLIMIT](#) подключение или отключение теста выхода за пределы оси динамически

[ENABLESTARTCONTROL](#) подключение и задание таймаута для управления невыполненным пуском или неожиданной остановкой оси

[NOTCHFILTER](#) задание предельной частоты фильтра отметки для указанной оси

[RESLIMNEG](#) отключение отрицательного предела оси

[RESLIMPOS](#) отключение положительного предела оси

[SETADJUST](#) задание настройки оси

SETBACKLASH	уменьшение или устранение влияния механических зазоров на траекторию оси
SETBIGWINFACTOR	изменение множителя для расчета большого окна на выбранной оси
SETDEADBAND	задание параметров минимального напряжения для указанной оси
SETENCLIMIT	задание пределов подключения неправильного кодера
SETINDEXEN	подключение или отключение на оси обнуления координаты у отметки нуля
SETINTEGTIME	задание числа образцов погрешности контура, использованных для расчета интегральной составляющей
SETIRMP	задание скорости начала изменения
SETLIMNEG	задание отрицательного предела оси
SETLIMPOS	задание положительного предела оси
SETMAXER	задание максимального значения допустимого преследования
SETMAXERNEG	задание максимального значения допустимого преследования (отрицательное направление)
SETMAXERPOS	задание максимального значения допустимого преследования (положительное направление)
SETMAXERTYPE	задание типа теста на сервоошибке
SETPHASESINV	подключение или отключение инверсии фаз на указанной оси
SETREFINV	подключение и отключение инверсии эталона скорости на указанной оси
SETRESOLUTION	изменение разрешения оси

Инструкции для управления Счетчиками

DECOUNTER	уменьшает счетчик
INCOUNTER	приражает счетчик
SETCOUNTER	задает счетчик

Инструкции для управления Таймерами

HOLDTIMER	блокирует таймер
SETTIMER	задает таймер
STARTTIMER	запускает таймер

Инструкции для управления Матрицами

CLEAR	обнуление переменной, вектора, матрицы
FIND	поиск элемента
FINDB	поиск элемента в векторе или матрице, упорядоченных по возрастающей
LASTELEM	последний элемент вектора или матрицы
LOCAL	объявление локальной переменной, вектора, матрицы
MOVEMAT	копирование одной строки матрицы в другую
PARAM	объявление параметра функции
SETVAL	изменение переменной
SORT	упорядочение вектора или матрицы

Инструкции для управления Строками

ADDSTRING	сцепление двух строк
CONTROLCHAR	задание управляющего символа в строковой переменной
LEFT	извлечение первых символов
LEN	считывание длины строки
MID	извлечение некоторых символов
RIGHT	извлечение последних символов
SEARCH	поиск строки
SETSTRING	изменение строковой переменной
STR	преобразование из числа в строку
VAL	преобразование из строки в число

Инструкции для управления Связью

CLEARRECEIVE	опорожняет список требующих удовлетворения RECEIVE
------------------------------	--

COMCLEARXBUFFER	опорожняет буфер приема последовательного порта
COMCLOSE	закрывает последовательный порт
COMGETERROR	считывает код ошибки
COMGETRXCOUNT	считывает число байтов, имеющих в буфере приема
COMOPEN	открывает последовательный порт
COMREAD	выполняет считывание с последовательного порта
COMREADSTRING	считывает строку с последовательного порта
COMWRITE	записывает на последовательный порт
COMWRITESTRING	записывает строку на последовательный порт
RECEIVE	прием данных извне
SEND	отправка данных наружу
SENDIPC	направляет информацию IPC
WAITIPC	ожидает прибытия информации IPC
WAITRECEIVE	прием данных извне с ожиданием

Математические Инструкции

ABS	абсолютное значение
ADD	сумма
AND	бинарный AND
ARCCOS	арккосинус
ARCSIN	арксинус
ARCTAN	арктангенс
COS	косинус
DIV	деление
EXP	экспонента
EXPR	решает математические выражения
LOG	натуральный логарифм
LOGDEC	десятичный логарифм
MOD	модуль
MUL	умножение
NOT	бинарный НЕТ
OR	бинарный ИЛИ
RANDOM	генерирует случайное число
RESETBIT	задает бит на 0
ROUND	самоокругление
SETBIT	задает бит на 1
SHIFTL	поворачивает биты влево
SHIFTR	поворачивает биты вправо
SIN	синус
SQR	квадратный корень
SUB	вычитание
TAN	тангенс
TRUNC	усечение
XOR	бинарное исключающее ИЛИ

Инструкции для управления Многозадачным режимом

ENDMAIL	сигнализирует окончание выполнения команды
ENDREALTIMETASK	завершает задачу реального времени
ENDTASK	завершает задачу
GETPRIORITYLEVEL	считывает уровень приоритета текущей задачи
GETREALTIME	возвращает время, прошедшее с начала задачи реального времени осей
GETREALTIMECOUNT	возвращает число прошедших задач реального времени
HOLDTASK	прерывает выполнение задачи
RESUMETASK	возобновляет выполнение задачи
SENDMAIL	отправляет команду на почтовый ящик 'mail'
SETPRIORITYLEVEL	задает уровень приоритета текущей задачи
STARTREALTIMETASK	запускает задачу реального времени
STARTTASK	запускает выполнение задачи
STOPTASK	приостанавливает выполнение задачи и останавливает движение соответствующих осей
WAITMAIL	получает команду от почтового ящика 'mail'

[WAITTASK](#)

ожидает завершения выполнения задач

Инструкции для управления Поток

[CALL](#)

вызов подпрограммы

[DELONFLAG](#)

отключение управления экстренным случаем на флаговом бите или флажковом выключателе

[DELONINPUT](#)

отключение управления экстренным случаем на цифровом входе

[ENDREP](#)

окончание повторения блока при помощи REPEAT

[FCALL](#)

вызов функции

[FOR](#)

расширение REPEAT

[FRET](#)

возврат от вызова функции

[GOTO](#)

переход к ярлыку

[IF](#)

тест переменной

[IFACC](#)

тест, находится ли ось в состоянии ускорения

[IFAND](#)

тест на операции AND

[IFBIT](#)

тест на бите

[IFBLACKBOX](#)

тест, активна ли регистрация действия логических устройств.

[IFCHANGEVEL](#)

тест, находится ли ось в состоянии изменения скорости

[IFCOUNTER](#)

тест счетчика

[IFDEC](#)

тест, находится ли ось в состоянии замедления

[IFDIR](#)

тест направления оси

[IFERRAN](#)

тест погрешности контура

[IFERROR](#)

тест погрешности активного цикла

[IFFLAG](#)

тест флажка

[IFINPUT](#)

тест входа

[IFMESSAGE](#)

тест активного сообщения

[IFOR](#)

тест на операции OR

[IFOUTPUT](#)

тест выхода

[IFQUOTER](#)

тест реальной координаты

[IFQUOTET](#)

тест целевой координаты

[IFRECEIVED](#)

тест приема данных

[IFREG](#)

тест, находится ли ось на режиме

[IFSAME](#)

проверка того, что два аргумента относятся к одним и тем же данным

[IFSTILL](#)

тест остановки оси

[IFSTR](#)

тест строки

[IFTARGET](#)

тест, находится ли ось на целевой координате

[IFTASKHOLD](#)

тест, приостановлена ли параллельная функция

[IFTASKRUN](#)

тест, выполняется ли параллельная функция

[IFTIMER](#)

тест таймера

[IFVALUE](#)

тест переменной

[IFVEL](#)

тест скорости оси

[IFWIN](#)

тест, находится ли ось в окне

[IFXOR](#)

тест на операции XOR

[NEXT](#)

окончание повторения блока при помощи FOR

[ONERRSYS](#)

задает вызов функции при ошибке системы

[ONFLAG](#)

экстренный случай на флаговом бите или флажковом выключателе

[ONINPUT](#)

экстренный случай на цифровом входе

[REPEAT](#)

повторение блока инструкций

[RET](#)

возврат от подпрограммы

[SELECT](#)

множественный выбор с переходом

[TESTIPC](#)

проверка наличия информации IPC

[TESTMAIL](#)

тест и прием команды

Различные инструкции

[CLEARERRORS](#)

стирание всех ошибок цикла модуля

[CLEARMESSAGES](#)

стирание всех сообщений модуля

[DEFMSG](#)

определение сообщения группы

[DELAY](#)

блокировка текущей функции на определенный временной период

[DELERROR](#)

стирание предыдущей ошибки цикла

[DELMESSAGE](#)

стирание предыдущего сообщения

[ERROR](#)

отправка ошибки цикла на ПК

[IFDEF/ELSEDEF/ENDIF](#)

тест для условной компиляции

[MESSAGE](#)

отправка сообщения на ПК

[SYSFAULT](#)

отключение сигнала SYSOK

[SYSOK](#)

подключение сигнала SYSOK

TYPEOF	тип аргумента
WATCHDOG	подключает, обновляет, отключает сторожевую схему из GPL на аппаратном модуле TMSWD

Инструкции для управления MECHATROLINK-II

MECCOMMAND	отправка команды на привод оси
MECGETPARAM	считывание параметра указанной оси
MECGETSTATUS	считывание значений STATUS, ALARM и IO_MON
MECSETPARAM	запись параметра в указанную ось

Инструкции для управления стандартными полевыми шинами

AXCONTROL	задание значения для ControlWord
AXSTATUS	возвращение значения, содержащегося в StatusWord
CNBYDEVICE	возвращает номер платы и узла устройства
READDICTIONARY	считывание содержимого объекта словаря
WRITEDICTIONARY	запись содержимого объекта словаря

Инструкции для управления EtherCAT

ACTIVATEMODE	задание рабочего режима
ECATGETREGISTER	возвращение содержимого регистра ESC (EtherCAT Slave Controller)
ECATSETREGISTER	запись содержимого регистра ESC (EtherCAT Slave Controller)
GETPDO	возвращает объект в PDO EtherCAT
SETEOF	подключение или отключение Sniffer
SETPDO	задает объект в PDO EtherCAT

Инструкции для управления шиной CAN

GETCNSTATE	возврат состояния протокола NMT для узла платы.
GETSDOERROR	возврат последней возникшей ошибки
GETMNSTATE	возврат состояния протокола NMT для главного узла платы
RECEIVEPDO	считывание содержимого асинхронного PDO
SENDPDO	запись содержимого асинхронного PDO
SETNMTSTATE	задание состояния протокола NMT для узла платы

Инструкции для Моделирования

DISABLE	отключение одной или нескольких осей
DISABLEFORCEDINPUT	отключение принудительного задания входов
ENABLE	подключение одной или нескольких осей
ENABLEFORCEDINPUT	подключение принудительного задания входов
RESETFORCEDINPUT	принудительное задание входа на OFF
SETFORCEDANALOG	принудительное задание аналогового входа
SETFORCEDINPUT	принудительное задание входа на ON
SETFORCEDPORT	принудительное задание порта входов

Инструкции для функций "Blackbox"

ENDBLACKBOX	завершает функцию регистрации
PAUSEBLACKBOX	приостанавливает функцию регистрации
STARTBLACKBOX	запускает функцию регистрации

Инструкции для управления ISO

ISO G0	задание движения в быстром режиме
ISO G1	задание интерполированного движения
ISO G9	задание принудительной остановки движения
ISO G90	задание толкования координат как абсолютных координат
ISO G91	задание толкования координат как относительных координат
ISO G93	задание толкования скорости обратно пропорционально продолжительности
ISO G94	задание интерпретации скорости как единицы измерения в минуту
ISO G216	определение матриц для определения параметров станка
ISO G217	описание образующих станок физических осей и виртуальных осей
ISO M2	освобождение осей от движения ISO
ISO M6	выбор индексов матриц определения параметров
ISO SETPARAM	задание некоторых параметров, характеризующих плавность интерполированного движения ISO
KINEMATICEXPR	задание отдельных выражений обратной и прямой кинематики

10.3.3 Ввод/Вывод

GETFEED

Синтаксис

GETFEED **переменная**

Аргументы

переменная скорость подачи

Описание

Копирует значение скорости подачи, считанной платой удаленных входов/выходов, в указанную **переменную**. Значение feedrate находится в пределах от 0 до 100 и выражается в процентах. Все платы TRA, обеспечивающие управление скоростью подачи, имеют специальный разъем.

INPANALOG

Синтаксис

INPANALOG **имяаналогвхода, переменная**

Аргументы

имяаналогвхода **переменная** имя устройства типа аналогового входа
переменная переменная

Описание

Копирует значение аналогового входа, указанного **имяаналогвхода**, в указанную **переменную**.

INPFLAGPORT

Синтаксис

INPFLAGPORT **имяфлаговогопорта, переменная**

Аргументы

имяфлаговогопорта **переменная** имя устройства типа флагового порта
переменная переменная

Описание

Копирует состояние флагового порта, указанного **имяфлаговогопорта**, в указанную **переменную**. Флаговый порт рассматривается как битовая маска. Каждому флаговому порту присваивается один бит. Если флажок находится в состоянии "ON", соответствующий бит задается на 1.

INPPORT

Синтаксис

INPPORT **имяпорта, переменная**

Аргументы

имяпорта **переменная** имя устройства типа входного порта
переменная переменная integer или char

Описание

Копирует состояние входного порта **имяпорта** в указанную **переменную**.
Входной порт рассматривается как битовая маска. Если вход порта находится в состоянии "ON", соответствующий бит задается на 1.

MULTIINPPORT**Синтаксис**

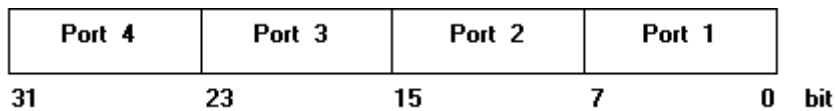
MULTIINPPORT **порт1[,...,порт4], переменная**

Аргументы

порт1 выдает биты от 0 до 7
порт2 выдает биты от 8 до 15
порт3 выдает биты от 16 до 23
порт4 выдает биты от 24 до 31
переменная переменная целого типа, которой присваиваются входные порты

Описание

Считывает одновременно до четырех входных портов и записывает их в одну **переменную**.
Считывание портов выполняется быстрым способом, что обеспечивает чтение в один и тот же отрезок реального времени. Порт1 соответствует самому низкому байту, а порт4 – самому высокому.

**MULTIOUTPORT****Синтаксис**

MULTIOUTPORT **значение, имя порта1[,...,имяпорта4]**

Аргументы

значение число или переменная целого типа для записи в выходные порты
имяпорта1 получает биты от 0 до 7
имяпорта2 получает биты от 8 до 15
имяпорта3 получает биты от 16 до 23
имяпорта4 получает биты от 24 до 31

Описание

Записывает **значение** одновременно в четыре выходных порта. Запись в указанные порты выполняется быстрым способом, что обеспечивает запись в один и тот же отрезок реального времени.
Если **имяпорта2**, **имяпорта3**, **имяпорта4** не заданы, соответствующему байту присваивается значение 0.

**MULTIRESETFLAG****Синтаксис**

MULTIRESETFLAG **маска, имяфлажка1[..., имяфлажка32]**

Аргументы

маска маска задействованных флажков - константа или переменная
имяфлажка1 имя устройства типа флажка

Описание

Из всех флажков **имяфлажка** (1÷32) отключает, т. е. переключает в состояние "OFF" те, бит которых в аргументе **маска** задан на 1.
Бит 0 **маски** (младший) соответствует **именифлажка1**.

MULTIRESETOUT

Синтаксис

MULTIRESETOUT **маска, имявыхода1[, ..., имявыхода32]**

Аргументы

маска маска задействованных выходов - константа или переменная
имявыхода1 имя устройства типа выхода

Описание

Из всех выходов **имявыхода** (1÷32) отключает те, бит которых в аргументе **маска** задан на 1. Бит 0 **маски** (младший) соответствует **именивыхода1**.

MULTISETFLAG

Синтаксис

MULTISETFLAG **маска, имяфлажка1[, ..., имяфлажка32]**

Аргументы

маска маска задействованных флажков - константа или переменная
имяфлажка1 имя устройства типа флажка

Описание

Из всех флажков **имяфлажка** (1÷32) подключает, т. е. переключает в состояние "ON" те, бит которых в аргументе **маска** задан на 1. Бит 0 **маски** (младший) соответствует **именифлажка1**.

MULTISETOUT

Синтаксис

MULTISETOUT **маска, имявыхода1[, ..., имявыхода32]**

Аргументы

маска маска задействованных выходов - константа или переменная
имявыхода1 имя устройства типа выхода

Описание

Из всех выходов **имявыхода** (1÷32) подключает те, бит которых в аргументе **маска** задан на 1. Бит 0 **маски** (младший) соответствует **именивыхода1**. Если выход - моностабильного типа, он отключается автоматически при истечении фиксированного таймаута в 200 миллисекунд.

MULTIWAITFLAG

Синтаксис

MULTIWAITFLAG **маска, флажок1[, ..., флажок32], состояние [, таймаут [, GOTO ярлык]]**
MULTIWAITFLAG **маска, флажок1[, ..., флажок32], состояние [, таймаут [, CALL имяподпрограммы]]**
MULTIWAITFLAG **маска, флажок1[, ..., флажок32], состояние [, таймаут [, CALL имяфункции]]**

Аргументы

маска константа или переменная. Маска задействованных флажков
флажок1[,...флажок32] имя устройства типа флажка
состояние заданная константа. Допускаются следующие значения:
- **ON** активное состояние флажка
- **OFF** отключенное состояние флажка
таймаут константа или переменная. Это предельное время ожидания
ярлык ярлык перехода (GOTO)
имяподпрограммы ярлык подпрограммы (CALL)
ы
имяфункции имя функции

Описание

Ожидает, пока флажки, определенные **флажок1...флажок32**, окажутся в состоянии, указанном параметром **состояние** (ON/OFF). Из всех флажков проверяет те, бит которых в аргументе **маска** подключен (ON). Бит 0 аргумента **маска** (младший) соответствует биту, определенному **флажком1**, бит 1 соответствует биту, определенному **флажком2** и так далее до бита, определенного **флажком32**.

Параметр **таймаут** позволяет задать таймаут, отличный от таймаута по умолчанию, составляющего одну секунду.
 При наличии **ярлыка** или **имени подпрограммы** или **имени функции** по истечении таймаута программа переходит на **ярлык** или вызывает **имя подпрограммы** или **имя функции**.

MULTIWAITINPUT

Синтаксис

MULTIWAITINPUT **маска, вход1[, ..., вход32], состояние [, таймаут [, GOTO ярлык]]**
MULTIWAITINPUT **маска, вход1[, ..., вход32], состояние [, таймаут [, CALL**
 имя подпрограммы]]
MULTIWAITINPUT **маска, вход1[, ..., вход32], состояние [, таймаут [, имя функции]]**

Аргументы

маска константа или переменная. Маска задействованных входов
вход1[,...вход32] имя входа
состояние заданная константа. Допускаются следующие значения:
 - **ON** активное состояние входа
 - **OFF** отключенное состояние входа
таймаут константа или переменная. Это предельное время ожидания
ярлык ярлык перехода (GOTO)
имя подпрограммы ярлык подпрограммы (CALL)
ы
имя функции имя функции

Описание

Ожидает, пока входы, определенные **входом1...входом32**, окажутся в состоянии, указанном параметром **состояние** (ON/OFF).
 Из всех входов проверяет те, бит которых в аргументе **маска** подключен (ON). Бит 0 аргумента **маска** (младший) соответствует биту, определенному **входом1**, бит 1 соответствует биту, определенному **входом2** и так далее до бита, определенного **входом32**.
 Если не указываются дополнительные аргументы, по истечении 1 секунды с начала выполнения инструкции (таймаут по умолчанию) автоматически генерируется следующее параметризованное сообщение: "Wait inputn ON/OFF". Имя сигнализированного входа - это имя первого подключенного входа, который еще не удовлетворил состояние. При наличии параметра **таймаут** описанное выше сообщение генерируется по истечении требуемого таймаута. Если после истечения таймаута требуемое условие возникает, автоматически генерируется параметризованное сообщение для стирания направленного ранее.
 При наличии **ярлыка** или **имени подпрограммы** или **имени функции** по истечении таймаута программа переходит на **ярлык** или вызывает **имя подпрограммы** или **имя функции**.

OUTANALOG

Синтаксис

OUTANALOG **имя аналогового выхода, значение**

Аргументы

имя аналогового выхода имя устройства типа аналогового выхода или оси
значение константа или переменная

Описание

Задаёт напряжение, указанное **значением**, на аналоговом выходе или в оси, выраженной **именем аналогового выхода**.

OUTFLAGPORT

Синтаксис

OUTFLAGPORT **имя флагового порта, значение**

Аргументы

имя флагового порта имя устройства типа флагового порта
значение константа или переменная

Описание

Копирует **значение** во флаговый порт, указанный **именем флагового порта**.
 Параметр **значение** рассматривается как битовая маска. Каждому биту присваивается флаговый порт. Если бит равен 1, соответствующий флажок задается на "ON".

OUTPORT

Синтаксис

OUTPORT **имяпорта, значение**

Аргументы

имяпорта имя устройства типа выходного порта
значение константа или переменная, integer или char

Описание

Копирует **значений** выходной порт **имяпорта**.
Выходной порт рассматривается как битовая маска. Если бит равен 1, соответствующий выход задается на "ON".

RESETFLAG

Синтаксис

RESETFLAG **имяфлажка**

Аргументы

имяфлажка имя устройства типа флажка

Описание

Отключает (переключает на OFF) флажок **имяфлажка**.

RESETOUT

Синтаксис

RESETOUT **имявыхода**

Аргументы

имявыхода имя устройства типа цифрового выхода

Описание

Отключает (переключает на OFF) выход **имявыхода**.

SETFLAG

Синтаксис

SETFLAG **имяфлажка**

Аргументы

имяфлажка имя устройства типа флажка

Описание

Подключает (переключает на ON) флажок **имяфлажка**.

SETOUT

Синтаксис

SETOUT **имявыхода**

Аргументы

имявыхода имя устройства типа цифрового выхода

Описание

Подключает (переключает на ON) выход **имявыхода**.
Если выход сконфигурирован моностабильным, он отключается автоматически при истечении фиксированного таймаута в 200 миллисекунд.

WAITFLAG

Синтаксис

WAITFLAG **имяфлажка, состояние [, таймаут [, GOTO ярлык]]**
WAITFLAG **имяфлажка, состояние [, таймаут [, CALL имяподпрограммы]]**
WAITFLAG **имяфлажка, состояние [, таймаут [, имяфункции]]**

Аргументы

имяфлажка имя флагового устройства

состояние	заданная константа. Допускаются следующие значения: - ON активное состояние флажка - OFF отключенное состояние флажка
таймаут	константа или переменная. Это предельное время ожидания
ярлык	ярлык перехода (GOTO)
имяподпрограммы	ярлык подпрограммы (CALL)
имяфункции	имя функции

Описание

Ожидает, пока флажок **имяфлажка** не примет состояние, указанное параметром **состояние** (ON/OFF).

Если из дополнительных аргументов присутствует только **таймаут**, ошибка цикла "Флажок **имяфлажка** в ожидании **состояния**" генерируется по истечении таймаута.

Если это условие возникает после истечения таймаута, направленная ранее ошибка цикла для этой задачи автоматически стирается.

При наличии **ярлика** или **имениподпрограммы** или **именифункции** по истечении таймаута программа переходит на **ярлык** или вызывает **имяподпрограммы** или **имяфункции** без генерирования какого-либо автоматического отображения.

Примечание

Для предупреждения ситуации ожидания флажка в ходе рабочего цикла рекомендуется задавать таймаут.

WAITINPUT**Синтаксис**

```

WAITINPUT имявхода, состояние [, таймаут [, GOTO ярлык]]
WAITINPUT имявхода, состояние [, таймаут [, CALL имяподпрограммы]]
WAITINPUT имявхода, состояние [, таймаут [, имяфункции]]

```

Аргументы

имявхода	имя входа
состояние	заданная константа. Допускаются следующие значения: - ON активное состояние входа - OFF отключенное состояние входа
таймаут	константа или переменная. Предельное время ожидания
ярлык	ярлык перехода (GOTO)
имяподпрограммы	ярлык подпрограммы (CALL)
имяфункции	имя функции

Описание

Ожидает, пока вход **имявхода** не примет состояние, указанное параметром **состояние** (ON/OFF).

Если не указываются дополнительные аргументы, по истечении 20 секунд с начала выполнения инструкции автоматически генерируется ошибка цикла: "Цифровой вход **имявхода** в ожидании **состояния**".

Если из дополнительных аргументов присутствует только **таймаут**, указанное выше сообщение будет генерироваться по его истечении.

Если это условие возникает после истечения **таймаута**, направленная ранее ошибка цикла для этой задачи автоматически стирается.

При наличии **ярлика** или **имениподпрограммы** или **именифункции** по истечении таймаута программа переходит на **ярлык** или вызывает **имяподпрограммы** или **имяфункции** без генерирования какого-либо автоматического отображения.

Примечание

Для предупреждения ситуации ожидания входного сигнала в ходе рабочего цикла рекомендуется задавать таймаут меньше времени по умолчанию (20 секунд).

Пример

[Стандартная процедура обнуления оси](#)

WAITPERSISTINPUT**Синтаксис**

```

WAITPERSISTINPUT имявхода, состояние, timepersist [, таймаут [, GOTO ярлык]]
WAITPERSISTINPUT имявхода, состояние, timepersist [, таймаут [, CALL
имяподпрограммы]]
WAITPERSISTINPUT имявхода, состояние, timepersist [, таймаут [, имяфункции]]

```

Аргументы

имявхода	имя устройства типа цифрового входа
состояние	заданная константа. Допускаются следующие значения: - ON активное состояние входа - OFF отключенное состояние входа
timepersist	константа или переменная
таймаут	константа или переменная. Предельное время ожидания
ярлык	ярлык перехода (GOTO)
имяподпрограммы	ярлык подпрограммы (CALL)
имяфункции	имя функции

Описание

Ожидает, пока вход **имявхода** не примет состояние, указанное параметром **состояние** (ON/OFF), и остается в этом состоянии в течение времени, указанного в **timepersist**.

Если не указываются дополнительные аргументы, по истечении 20 секунд с начала выполнения инструкции автоматически генерируется ошибка цикла: "Цифровой вход **имявхода** в ожидании **состояния**".

Если из дополнительных аргументов присутствует только **таймаут**, указанное выше сообщение будет генерироваться по его истечении.

Если это условие возникает после истечения **таймаута**, направленная ранее ошибка цикла для этой задачи автоматически стирается.

При наличии **ярлыка** или **имениподпрограммы** или **именифункции** по истечении таймаута программа переходит на **ярлык** или вызывает **имяподпрограммы** или **имяфункции** без генерирования какого-либо автоматического отображения.

Примечание

Для предупреждения ситуации ожидания входного сигнала в ходе рабочего цикла рекомендуется задавать таймаут меньше времени по умолчанию (20 секунд).

10.3.4 Оси**CHAIN****Синтаксис**

CHAIN **главная_ось, ведомая_ось1 [...ведомая_ось5]**

Аргументы

главная_ось	имя устройства типа оси, которое будет выступать в качестве главной
ведомая_ось1...ведомая_ось5	имя устройства типа оси, которое будет выступать в качестве ведомой

Описание

После выполнения этой инструкции **ведомые_оси** (1÷5) выполнят движения, связанные с движениями главной оси отношением сцепления, заданным инструкцией **RATIO**. Окажутся сцепленными как движения точка-точка, так и интерполированные движения.

Ведомая_ось1 не является дополнительным параметром, а должна определяться всегда.

Ведомая ось для сцепления не должна быть занята в интерполяции и не может, в свою очередь, быть главной для других ведомых осей.

Главная ось, в свою очередь, не может быть ведомой других осей.

Сцепление может выполняться как с осями на координате, так и с осями в движении.

Для отключения сцепления осей достаточно выполнить инструкцию **NORMAL** на главной оси. Последняя указанная операция может выполняться как с осями на координате, так и с осями в движении. При отключении сцепления во время движения осей ведомая ось выполняет замедление и останавливается.

Одновременно можно определить максимум 8 главных осей.

Эта инструкция может выполняться и с шаговыми осями при условии, что все они управляются при помощи **TRS-AX**.

Кроме того, все эти оси должны иметь реальный кодер, а не смоделированный. В противном случае генерируется ошибка системы "4101- Непоследовательное управление осью имяоси".

См. также **RATIO**.

Пример

```
; сцепление оси Y с осью X
CHAIN X, Y
; перемещение оси X.
; Y повторение движения X
MOVINC X, 100
```


CIRCABS

Синтаксис CIRCABS

[ярлык],ось1, координата1, ось2, координата2, направление, ±радиус [, угол]

Аргументы

ярлык	константа или переменная integer. Ярлык, определяющий блок смещения
ось1, ось2	имена устройств типа оси
координата1, координата2	константа или переменная. Представляет координату абсолютного смещения
направление	переменная integer. Определяет тип вращения. Допускаются следующие значения: CW по часовой стрелке CCW против часовой стрелке
радиус	константа или переменная. Представляет значение радиуса окружности
угол	константа или переменная. Представляет исходный угол

Описание

Круговая интерполяция по 2 осям с *абсолютным смещением*, основанная на запрограммированных координатах **координата1** и **координата2**.

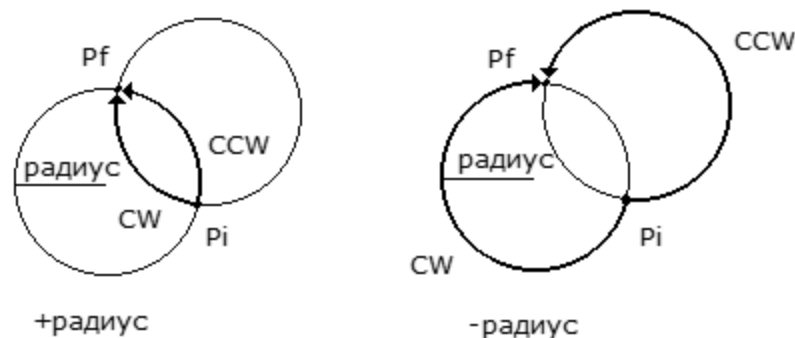
Дуга определяется начальной точкой (текущая точка), конечной точкой, значением **радиуса** и **направлением** перемещения.

Знак, применимый к **радиусу**, позволяет выбирать меньшую дугу (+радиус) или большую дугу (-радиус).

В особом случае, когда начальная координата оси 1 совпадает с конечной координатой **координата1**, а начальная координата оси 2 совпадает с конечной координатой **координата2**, получается полный круг. В этом случае необходимо указать аргумент **угол** с тем же значением, что и инструкция [CIRCLE](#) (на которую делать ссылку).

Параметр "угол" служит для однозначного определения центра круга с тем же значением, что и инструкция [CIRCLE](#). Принимается во внимание только тогда, когда перед выполнением инструкции **координата1** и **координата2** соответствуют текущим координатам осей. Дополнительный параметр **ярлык** используется вместе с инструкцией [SETLABELINTERP](#) для однозначного определения блока смещения.

Оси шагового типа могут использоваться в этой инструкции только при их управлении удаленным TRS-AX. В этом случае необходимо учитывать, что под термином "интерполяция" понимается координированное движение нескольких осей с определенной погрешностью, обуславливаемой методом управления осью.



CIRCINC

Синтаксис CIRCINC

[ярлык],ось1, координата1, ось2, координата2, направление, ±радиус [, угол]

Аргументы

ярлык	константа или переменная integer. Ярлык, определяющий блок смещения
ось1, ось2	имена устройств типа оси

координата1, координата2	константа или переменная. Представляет координату смещения с приращением
направление	переменная integer. Определяет тип вращения. Допускаются следующие значения: CW по часовой стрелке CCW против часовой стрелке
радиус	константа или переменная. Представляет значение радиуса окружности
угол	константа или переменная. Представляет исходный угол

Описание

Круговая интерполяция по 2 осям со *смещением с приращением*, основанная на запрограммированных координатах **координата1** и **координата2**.

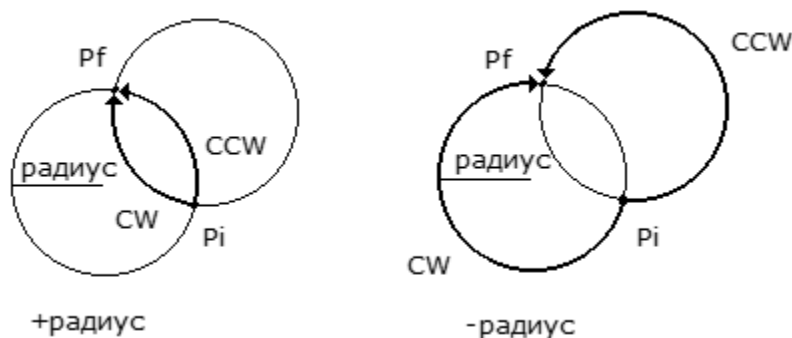
Дуга определяется начальной точкой (текущая точка), конечной точкой, значением **радиуса** и **направлением** перемещения.

Знак, применимый к **радиусу**, позволяет выбирать меньшую дугу (+радиус) или большую дугу (-радиус).

В особом случае, когда координата1 = координата2 = 0, получается полный круг. В этом случае необходимо указать аргумент **угол**, с тем же значением, что и инструкция **CIRCLE** (на которую делать ссылку).

Параметр "угол" служит для однозначного определения центра круга с тем же значением, что и инструкция **CIRCLE**. Дополнительный параметр **ярылок** используется вместе с инструкцией **SETLABELINTERP** для однозначного определения блока смещения.

Оси шагового типа могут использоваться в этой инструкции только при их управлении удаленным TRS-AX. В этом случае необходимо учитывать, что под термином "интерполяция" понимается координированное движение нескольких осей с определенной погрешностью, обуславливаемой методом управления осью.

**CIRCLE****Синтаксис**

CIRCLE [ярылок], ось1, ось2, направление, радиус, угол

Аргументы

ярылок	константа или переменная integer. Ярылок, определяющий блок смещения
ось1, ось2	имена устройств типа оси
направление	переменная integer. Определяет тип вращения. Допускаются следующие значения: CW по часовой стрелке CCW против часовой стрелке
радиус	константа или переменная. Представляет значение радиуса окружности
угол	константа или переменная. Представляет исходный угол

Описание

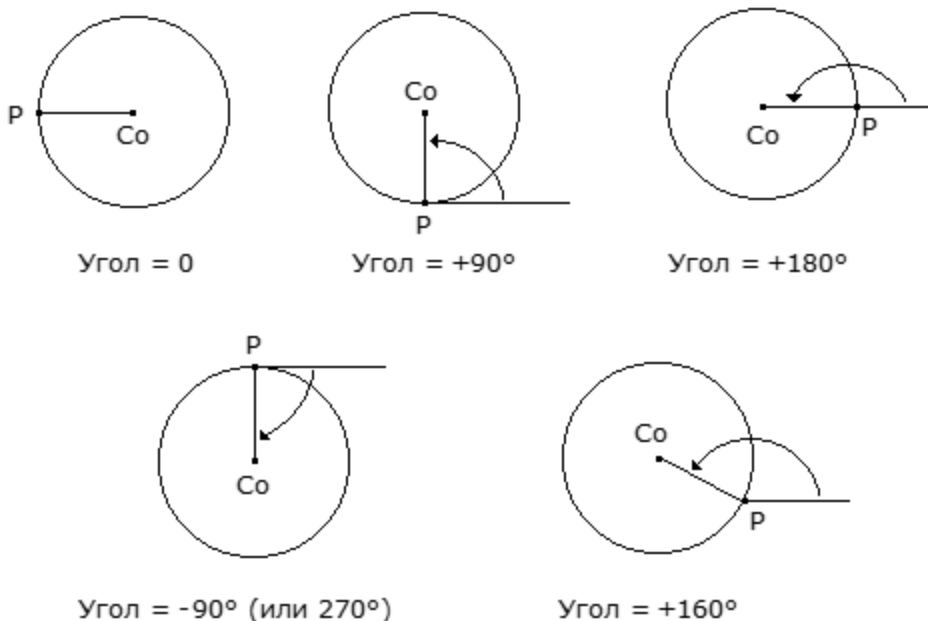
Полная круговая интерполяция.

Генерирует круг с **ось1** и **ось2**, в указанном **направлении**, с **радиусом**, равным заданному значению и исходя из заданного исходного **угла**.

Радиус может принимать только положительные значения.

Угол должен указываться в положительном тригонометрическом выражении против часовой стрелки начиная с оси X. Остается определить положение центра S_0 окружности, указывая угол, образованный радиусом, проходящим через запрограммированную начальную точку P (текущая точка), и

горизонтальным направлением X+. Дополнительный параметр **ярлык** используется вместе с инструкцией [SETLABELINTERP](#) для однозначного определения блока смещения. Оси шагового типа могут использоваться в этой инструкции только при их управлении удаленным TRS-AH. В этом случае необходимо учитывать, что под термином "интерполяция" понимается координированное движение нескольких осей с определенной погрешностью, обуславливаемой методом управления осью.



COORDIN

Синтаксис

COORDIN

матрица, значение **дельтаT**, **направление**, **ini**, **fin**, **маска**, **ось1**, **№_столбца_ось1**

[, (**ось2**, **№_столбца_ось2**) ÷ (**ось32**, **№_столбца_ось32**)]

Аргументы

матрица	матрица данных
значение дельтаT	константа или переменная. Масштаб по оси времени
направление	заданная константа. Направление считывания данных в матрице UP с самой нижней до самой верхней строки DOWN с самой верхней до самой нижней строки
ini	глобальная переменная integer. Это номер начальной строки
fin	глобальная переменная integer. Это номер конечной строки
маска	маска подключаемых осей
ось1 [...ось32]	имена устройств типа оси
№_столбца_ось1[...№_столбца_ось32]	номер столбца матрицы, относящейся к оси

Описание

Эта инструкция позволяет выполнять синхронизированные движения осей **ось1**, **ось2** и т. д. за счет смещений с приращением (микровекторов), определенных **матрицей** данных. Параметры **ось1** и **№_столбца_ось1** должны всегда определяться. Значения, содержащиеся в **матрице**, указывают абсолютные координаты по мере их достижения различными осями. Соответствующие смещения с приращением (разница координат между строкой (n) и строкой (n-1)) выполняются за временной интервал, равный **кратному** масштабу по оси времени (1мс = реальное

время обновления осей), определенному аргументом **значение Δt** , которое, таким образом, должно быть выражено целым числом.

После задания этого значения позволяет выполнять координированное перемещение максимум 32 осей по любому криволинейному пути в пространстве, как они генерированы методами SPLINE.

Эта инструкция является сквозной и для запуска не требует инструкции STARTINTERP. Однако в ее конец необходимо поместить инструкцию WAITSTILL для ожидания правильного прибытия осей на координату. Возможные изменения feedrate override должны выполняться при помощи инструкции SETFEEDI и управляться инструкцией SETFEEDCOORD.

Параметр **направление** позволяет определять направление перемещения по матрице, позволяя выполнять траекторию в обоих направлениях.

Столбцы сканируемой матрицы могут быть типа float или типа double, но не обоих типов одновременно.

Помимо перемещения осей по конечному маршруту (определенному числом строк матрицы), можно выполнять бесконечное перемещение, используя:

- матрицу только с одной строкой. В этом режиме работы блок управления всегда считывает одну строку матрицы и применяет к осям приведенные в ней координаты. Для того, чтобы оси перемещались, потребуется соответствующим образом изменить строку матрицы, используя, по возможности, задачу реального времени, обеспечивающую синхронизацию обновления координат с частотой обновления осей. При этом можно реализовать электронные кулачки или сцепления с отношением, отличным от 1:1. Для подключения этого режима работы необходимо задать **ini** = 1, **fin** = 0 и **направление** = UP. При использовании этого режима работы HE должна использоваться инструкция STOP.
- матрица с несколькими строками. Можно сканировать матрицу циклами от первой до последней строки бесконечно, задавая значения **ini** = 1, **fin** = 0 и **направление** = UP. Если требуется выполнить только одну строку многострочной матрицы, необходимо задать параметры **ini**, **fin** и **направление** следующим образом: **ini** = номер строки, которую необходимо выполнить, **fin** = номер строки, предшествующей той, которую необходимо выполнить, **направление** = UP. В другом случае генерируется ошибка системы.

Оси шагового типа могут использоваться в этой инструкции только при их управлении удаленным TRS-AH.

DISABLECORRECTION

Синтаксис

DISABLECORRECTION ось [, ось1, ..., ось6]

Аргументы

ось имя устройства типа оси
ось1, ..., ось6 имя устройства типа оси

Описание

Отключает линейную коррекцию для указанной **оси**.

Первый параметр - это ось, для которой требуется отключить коррекцию. Если это единственный указанный параметр, будут отключены все коррекции, имеющиеся в конфигурации. Последующие параметры позволяют указать, какие коррекции отключать. Если один из них совпадает с первым параметром, будет отключена автокоррекция.

Более подробное описание см. в [ENABLECORRECTION](#).

Пример

; отключает только автокоррекцию для оси X
DISABLECORRECTION X, X

; отключает перекрестную коррекцию (к X и Y) для оси Z, но не автокоррекцию
DISABLECORRECTION Z, X, Y

EMERGENCYSTOP

Синтаксис

EMERGENCYSTOP ось, время

Аргументы

ось имя устройств типа оси
время константа или целая переменная. Продолжительность линейной функции в мс

Описание

Останавливает указанную ось и с ней, - все оси, задействованные в интерполированном движении. Движение останавливается с линейной функцией замедления за время, указанное переменной **[время]**.

В движениях точка-точка, если заданное время превышает продолжительность замедления, указанную в конфигурации, используется продолжительность, указанная в конфигурации.

Если в интерполированных движениях заданное время превышает максимальную продолжительность замедления всех задействованных осей, используется максимальная продолжительность, заданная в конфигурации.

Движение может возобновляться командой [START](#).

Эта инструкция не может использоваться, если **[ось]** - ведомая.

Эта инструкция может генерировать следующие системные ошибки:

- "4101 - Непоследовательное управление осью", когда **[ось]** выполняет синхронизированное движение или многолинейную интерполяцию или движение ISO.
- "4105 - инструкция не может быть выполнена на оси", когда **[ось]** - ось подсчета.
- "4399 - параметр за пределами диапазона", если указанное **[время]** равно или меньше 0.

ENABLECORRECTION

Синтаксис

ENABLECORRECTION ось [, ось1, ..., ось6]

Аргументы

ось имя устройства типа оси
ось1, ..., ось6 имя устройства типа оси

Описание

Подключает линейную коррекцию для указанной **оси**. Коррекция состоит из автокоррекции и из перекрестной коррекции. Автокоррекция - это коррекция реальной координаты оси в зависимости от ее положения, перекрестная коррекция - это коррекция реальной координаты оси в зависимости от положения других осей. Можно определить максимум пять перекрестных корректоров.

Первый параметр - это ось, для которой требуется подключить коррекцию. Если это единственный указанный параметр, будут подключены все коррекции, имеющиеся в конфигурации.

Последующие параметры позволяют указать, какие коррекции подключать. Если один из них совпадает с первым параметром, будет подключена автокоррекция.

См. также [DISABLECORRECTION](#).

ПРИМЕЧАНИЕ: Для того, чтобы эта инструкция имела действие, коррекция должна быть подключена и в конфигурации.

Пример

; подключает все коррекции, предусмотренные в конфигурации для оси X
ENABLECORRECTION X

; подключает только автокоррекцию для оси X
ENABLECORRECTION X, X

; подключает автокоррекцию и перекрестную коррекцию (к X и Y) для оси Z
ENABLECORRECTION Z, X, Y, Z

ENDMOV

Синтаксис

ENDMOV ось [, координата]

Аргументы

ось имя устройств типа оси
координата константа или переменная.

Описание

Останавливает движение указанной оси. Отличается от инструкции [STOP](#), т. к. движение завершается и больше не может возобновляться другой возможной инструкцией [START](#). При указании параметр **координата** позволяет задавать положение, в котором ось завершит движение, в противном случае точка, в которой остановится ось, зависит от текущей скорости и последнего запрограммирования замедления. При необходимости для достижения точки конца движения блок управления выполняет инверсию движения оси.

Примечание

Параметр координата используется только если речь идет о движении точка-точка. При интерполированном движении движение оси останавливается без учета значения **координаты**.

Пример

```
; останавливает текущее движение и приводит ось на координату 0.0
ENDMOV X, 0.0
```

FASTREAD

Синтаксис

FASTREAD ось1, состояние, переменная1 [,ось2, переменная2],[..., ось8, переменная8]

Аргументы

ось1...[...ось8]	имена устройств типа оси. Ось1 - главная ось.
состояние	заданная константа. Принимает следующие значения: ON фронт подъема OFF фронт снижения
переменная1... [...переменная8]	переменная или элемент матрицы/вектора двойного типа. Координата, записанная в память

Описание

Координаты указанных **осей** считываются и записываются в память в **переменных** в момент, когда быстрый вход **ось1** (главной оси) переключается в заданное состояние.

Если указаны аналоговые оси, они должны относиться к одной и той же плате (4 для TRS-AX). Если указаны цифровые оси, сигнал быстрого входа находится непосредственно на приводе, следовательно, в случае FASTREAD множественных сигнал следует параллельно сопоставить различным устройствам.

При использовании шины EtherCAT для каждой оси, подвергаемой обмену энкодером (см. Инструкцию [SWITCHENC](#)), максимально допустимое количество осей уменьшается на одну.

Как только этот предел превышен, генерируется системная ошибка «4400 Слишком много активных осей в FASTREAD». Кроме того, дополнительный энкодер должен быть подключен к расширению TRS-AC-E на TRS-CAT. В противном случае, генерируется системная ошибка «4375 FASTREAD выполнена на осях различных плат».

Инструкция завершается, когда вход переключается в указанное **состояние (ON/OFF)**.

Если перед переключением быстрого входа выполняется инструкция STOP, эти инструкции остаются активными и возобновляются после инструкции START.

На одной плате осей можно активировать несколько быстрых считываний.

При выполнении команды невозможно одновременно выполнять команды [SETPZERO](#) и [SETPFLY](#) на одной оси при подключении к платам шиной MECHATROLINK-II.

Примечание

Быстрый вход для осей цифрового типа на плате с шиной MECHATROLINK-II находится на входе **EXTI2**, и не требуется конфигурировать его в виртуальном-физическом пространстве. Быстрые входы цифровых осей MECHATROLINK-II должны закорачиваться, т. к. запись координаты оси в память выполняется только со ссылкой на собственный быстрый вход.

FREE

Синтаксис

FREE ось [, напряжение]

Аргументы

ось	имя устройства типа оси
напряжение	константа float или переменная float. Эталонное напряжение

Описание

Переводит **ось** в состояние «разомкнутый контур» (Free), отключая *контроль ее положения*. Если **ось** является ведомой в сцеплении с другими осями, ограничение нарушается и движение **оси** останавливается. Если **задан** параметр **напряжения**, эталонное напряжение оси устанавливается равным ему.

Эта инструкция может использоваться в случае осей измерения для определения координат или осей, движение которых может принудительно задаваться внешними механическими органами, которые могут изменить его положение.

При функционировании координата оси обнаруживается и обновляется, поэтому можно позиционировать ось в абсолютном отношении после повторного подключения контроля положения (инструкция [NORMAL](#)).

HELICABS

Синтаксис

HELICABS [ярлык], ось1, координата1, ось2, координата2, ось3, координата3, направление, ±радиус [,угол[, числооборотов[, ось4, координата4 [, ..., ось6, координата6]]]]

Аргументы

ярлык	константа или переменная integer. Ярлык, определяющий блок смещения
ось1...ось3[...ось6]	имена устройств типа оси
координата1...координата3[...координата6]	константа или переменная. Координата абсолютного смещения
направление	переменная integer. Тип вращения по/против часовой стрелки (CW/CCW)
радиус	константа или переменная. Радиус спирали
угол	константа или переменная. Исходный угол
числооборотов	константа или переменная. Число оборотов

Описание

Винтовая интерполяция с абсолютным смещением, равным запрограммированным координатам **координата1**, **координата2** и **координата3**. Движение образовано круговой интерполяцией, связанной с осями **ось1** и **ось2** (с такими же синтаксическими правилами, что и [CIRCABS](#) /[CIRCINC](#), применительно к аргументам **направление**, **±радиус** и **угол**), и линейной интерполяцией, связанной с **ось3** (может также быть связана с **ось4**, **ось5** и **ось6**). Винтовое движение может развертываться на нескольких оборотах, указанных аргументом **числооборотов**. Координата, относящаяся к оси с линейным движением (например, возможные координаты **ось4**, **ось5**, **ось6**), относится к суммарному смещению (т. е. не к смещению на оборот). Дополнительный параметр **ярлык** используется вместе с инструкцией [SETLABELINTERP](#) для однозначного определения блока смещения. Оси шагового типа могут использоваться в этой инструкции только при их управлении удаленным TRS-AX. В этом случае необходимо учитывать, что под термином "интерполяция" понимается координированное движение нескольких осей с определенной погрешностью, обуславливаемой методом управления осью.

Примечание

1. Условие контурирования оценивается только на первых трех осях, которые образуют систему координат. При добавлении следующей и ее изменении происходит неправильное управление профилем скорости. Для обеспечения правильного движения между одной инструкцией HELICABS и другой необходимо вставить инструкцию [WAITSTILL](#).
2. при задании локальной системы координат при помощи инструкции [SETRIFLOC](#) три оси, определяющие новую систему координат, должны всегда указываться в параметрах инструкции HELICABS даже если они не выполняют смещения.

HELICINC

Синтаксис

HELICINC [ярлык], ось1, координата1, ось2, координата2, ось3, координата3, направление, ± радиус [,угол [, числооборотов [, ось4, координата4 [, ..., ось6, координата6]]]]

Аргументы

ярлык	константа или переменная integer. Ярлык, определяющий блок смещения
ось1...ось3[...ось6]	имена устройств типа оси
координата1...координата3[...координата6]	константа или переменная. Координата смещения с приращением
направление	переменная integer. Тип вращения по/против часовой стрелки (CW/CCW)
радиус	константа или переменная. Радиус спирали
угол	константа или переменная. Исходный угол
числооборотов	константа или переменная. Число оборотов

Описание

Винтовая интерполяция со смещением с приращением, равным запрограммированным координатам **координата1**, **координата2** и **координата3**.

Движение образовано круговой интерполяцией, связанной с осями **ось1** и **ось2** (с такими же синтаксическими правилами, что и [CIRCABS](#) /[CIRCINC](#), применительно к аргументам **направление**, **±радиус** и **угол**), и линейной интерполяцией, связанной с **ось3** (может также быть связана с **ось4**, **ось5** и **ось6**). Винтовое движение может разворачиваться на нескольких оборотах, указанных аргументом **числооборотов**.

Координата, относящаяся к оси с линейным движением (например, возможные координаты **ось4**, **ось5**, **ось6**), относится к суммарному смещению (т. е. не к смещению на оборот). Дополнительный параметр **ярлык** используется вместе с инструкцией [SETLABELINTERP](#) для однозначного определения блока смещения.

Оси шагового типа могут использоваться в этой инструкции только при их управлении удаленным TRS-AX. В этом случае необходимо учитывать, что под термином "интерполяция" понимается координированное движение нескольких осей с определенной погрешностью, обуславливаемой методом управления осью.

Примечание

1. Условие контурирования оценивается только на первых трех осях, которые образуют систему координат. При добавлении следующей и ее изменении происходит неправильное управление профилем скорости. Для обеспечения правильного движения между одной инструкцией HELICINC и другой необходимо вставить инструкцию [WAITSTILL](#).
2. при задании локальной системы координат при помощи инструкции [SETRIFLOC](#) три оси, определяющие новую систему координат, должны всегда указываться в параметрах инструкции HELICINC даже если они не выполняют смещения.

JERKCONTROL

Синтаксис

JERKCONTROL ось, состояние

Аргументы

ось	имена устройств типа оси
состояние	заданная константа. Допускаются следующие значения: ON состояние активного флажка OFF состояние неактивного флажка

Описание

Подключает, если параметру **состояние** присваивается значение ON, или отключает, если параметру **состояние** присваивается значение OFF, управление толчковым режимом на движениях с интерполяцией и точка-точка **оси**. Управление толчковым режимом подключается только с осями, имеющими сконфигурированную S-образную функцию ускорения и замедления. Если ось имеет сконфигурированную линейную функцию, управление толчковым режимом не выполняется.

JERKSMOOTH

Синтаксис

JERKSMOOTH ось, значение

Аргументы

ось	имена устройств типа оси
значение	константа или переменная типа float.

Описание

При выполнении классических интерполированных движений оси могут перемещаться в условиях контурирования, т. е. без остановки между двумя последовательными блоками перемещения, если угол между касательными к траектории меньше параметра "Макс. угол контурирования" (умолчание - 15 градусов, которые можно изменить инструкцией [SETCONTORNATURE](#)).

В противном случае оси останавливаются на ребре двух блоков с контролируемым замедлением и запускаются повторно по новому блоку с контролируемыми ускорениями. Однако остановка и повторный запуск снижают кпд движения станка. В случаях, когда угол контурирования принимает немаловажные значения, например, значение разрыва касательной больше 5 градусов, определяются значительные скачки скорости для задействованных в движении осей с соответствующими бесконечными значениями ускорения, толчкового режима и значительными механическими нагрузками, которые могут сказаться в т. ч. на качестве обработки.

Инструкция JERKSMOOTH, на основе значения, определенного пользователем, позволяет плавно, т. е. непрерывно, соединить ускорение и скорость, профили скорости, сохраняемые осями в ходе движений контурирования. Необходимо отметить, что такое плавное соединение вводит небольшие изменения в выполняемую траекторию по сравнению с теоретической, т. к. около точки контурирования оси имеют профиль скорости, отличный от теоретического.

Переменная **значение**, выраженная при помощи процентного значения от 0 до 100, определяет, в какой степени требуется плавно соединять профили скорости. Значение, равное 0, сохраняет теоретический профиль, создавая разрывы в ускорениях и в профилях скорости. Значение, равное

100, дает плавно соединенные профили, более высокие характеристики, но и максимальное смещение от теоретической траектории, пропорционально скорости, поддерживаемой на траектории.

Примечание

Эта инструкция применяется только в движениях с классической интерполяцией (инструкции [LINEARABS](#), [LINEARINC](#), [CIRCABS](#), [CIRCINC](#), [HELICABS](#), [HELICINC](#)). Она не применяется в движениях с многоосевой интерполяцией (инструкции [MULTIABS](#) и [MULTIINC](#)).

LINEARABS

Синтаксис

LINEARABS [ярлык],ось1, координата1, [ось2, координата2 [, ось3, координата3 [, ..., ось6, координата6]]]

Аргументы

ярлык константа или переменная integer. Ярлык, определяющий блок смещения
ось1[...ось2[...ось6]] имена устройств типа оси
координата1[...координата2[...координата6]] константа или переменная. Координата абсолютного смещения

Описание

Линейная интерполяция, с *абсолютным смещением*, на координаты, указанные **координата1**, **координата2**, и т. д. Дополнительный параметр **ярлык** используется вместе с инструкцией [SETLABELINTERP](#) для однозначного определения блока смещения

Оси шагового типа могут использоваться в этой инструкции только при их управлении удаленным TRS-AX. В этом случае необходимо учитывать, что под термином "интерполяция" понимается координированное движение нескольких осей с определенной погрешностью, обуславливаемой методом управления осью.

Примечание

1. Условие контурирования оценивается только на первых трех осях, которые образуют систему координат. При добавлении следующей и ее изменении происходит неправильное управление профилем скорости. Для обеспечения правильного движения между одной инструкцией LINEARABS и другой необходимо вставить инструкцию [WAITSTILL](#).
2. при задании локальной системы координат при помощи инструкции [SETRIFLOC](#) три оси, определяющие новую систему координат, должны всегда указываться среди первых трех параметров инструкции LINEARABS даже если они не выполняют смещения.

LINEARINC

Синтаксис

LINEARINC [ярлык], ось1, координата1, [ось2, координата2 [, ось3, координата3 [, ..., ось6, координата6]]]

Аргументы

ярлык константа или переменная integer. Ярлык, определяющий блок смещения
ось1[...ось2[...ось6]] имена устройств типа оси
координата1[...координата2[...координата6]] константа или переменная. Координата смещения с приращением

Описание

Линейная интерполяция, со *смещением с приращением*, равная координатам, указанным **координата1**, **координата2**, и т. д. Дополнительный параметр **ярлык** используется вместе с инструкцией [SETLABELINTERP](#) для однозначного определения блока смещения.

Оси шагового типа могут использоваться в этой инструкции только при их управлении удаленным TRS-AX. В этом случае необходимо учитывать, что под термином "интерполяция" понимается координированное движение нескольких осей с определенной погрешностью, обуславливаемой методом управления осью.

Примечание

1. Условие контурирования оценивается только на первых трех осях, которые образуют систему координат. При добавлении следующей и ее изменении происходит неправильное управление профилем скорости. Для обеспечения правильного движения между одной инструкцией LINEARINC и другой необходимо вставить инструкцию [WAITSTILL](#).
2. при задании локальной системы координат при помощи инструкции [SETRIFLOC](#) три оси, определяющие новую систему координат, должны всегда указываться в параметрах инструкции LINEARINC даже если они не выполняют смещения.

MOVABS

Синтаксис

MOVABS ось1, значение1 [, ось2, значение2 [, ..., ось6, значение6]]

Аргументы

ось1...[...ось6] имена устройств типа оси
значение1...[...значение6] константа или переменная. Значение абсолютного смещения

Описание

Заставляет выполнять указанные оси *абсолютное движение*, равное координатам, указанным **значение1** [,...**значение6**].

Для выполнения этого движения ось не должна быть занята в интерполированном движении и должна находиться на координате или в окне. Движение оси начинается как только выполняется инструкция. Если в одной задаче выполняется несколько инструкций перемещения точка-точка, они сцепляются. Если другая задача пытается выполнить инструкции точка-точка на оси, которая уже занята в движении, эта задача останется в ожидании завершения движения, команда на которое была дана первой задачей.

Кроме того, можно изменить скорость между одним движением точка-точка и последующим при помощи инструкции [SETVEL](#). Два движения будут соединены функцией скорости без остановки осей. Если инструкция [SETVEL](#) не используется, максимальная возможная скорость представляется значением скорости в ручном режиме, определенной в конфигурации.

Движение точка-точка может прерываться инструкцией [STOP](#), а затем возобновляться инструкцией [START](#). При прерывании движения ось остается в состоянии "на режиме", даже если физически она не перемещается.

Движение может прерываться инструкцией [ENDMOV](#). В этом случае оно не может возобновляться.

Примечание

1) Ранее движения точка-точка:

- позволяли менять скорость только при остановленной оси. Текущее поведение подобно поведению при интерполированных движениях.
 - при прерывании инструкцией STOP соответствующая ось приходила в состояние "на координате".
- 2) Если число блоков перемещения превышает 32 и блоки образованы микроотрезками, рекомендуется использовать инструкции линейной интерполяции вместо инструкций движения точка-точка. Для получения более подробной информации запросите на T.P.A документ "Limiti Firmware Movimento Punto Punto.doc".

Пример 1

[Стандартная процедура обнуления на прерывании](#)

Пример 2

```
; изменение скорости
Function cambiovel
  setvel X, 20
  setvel X, 20
  movabs X, 100, Y, 200
  movabs X, 150, Y, 180
  setvel X, 5
  movabs X, 80, Y, 100
  waitstill X, Y
fret
```

MOVINC

Синтаксис

MOVINC ось1, значение1 [, ось2, значение2 [, ..., ось6, значение6]]

Аргументы

ось1...[...ось6] имена устройств типа оси
значение1...[...значение6] константа или переменная. Значение смещения с приращением

Описание

Заставляет каждую ось выполнять *движение с приращением*, равное соответствующему **значению**. Для выполнения этого движения ось не должна быть занята в интерполированном движении и должна находиться на координате или в окне. Движение оси начинается как только выполняется инструкция. Если в одной задаче выполняется несколько инструкций перемещения точка-точка, они сцепляются. Если другая задача пытается выполнить инструкции точка-точка на оси, которая уже занята в движении, эта задача останется в ожидании завершения движения, команда на которое была дана первой задачей.

Кроме того, можно изменить скорость между одним движением точка-точка и последующим при помощи инструкции [SETVEL](#). Два движения будут соединены функцией скорости без остановки осей. Если инструкция [SETVEL](#) не используется, максимальная возможная скорость представляется значением скорости в ручном режиме, определенной в конфигурации. Движение точка-точка может прерываться инструкцией [STOP](#), а затем возобновляться инструкцией [START](#). При прерывании движения ось остается в состоянии "на режиме", даже если физически она не перемещается. Движение может прерываться инструкцией [ENDMOV](#). В этом случае оно не может возобновляться.

Примечание

- 1) Ранее движения точка-точка:
 - позволяли менять скорость только при остановленной оси. Текущее поведение подобно поведению при интерполированных движениях.
 - при прерывании инструкцией STOP соответствующая ось приходила в состояние "на координате".
- 2) Если число блоков перемещения превышает 32 и блоки образованы микроотрезками, рекомендуется использовать инструкции линейной интерполяции вместо инструкций движения точка-точка. Для получения более подробной информации запросите на TPA документ "Limiti Firmware Movimento Punto Punto.doc".

Пример 1

[Стандартная процедура обнуления оси](#)

Пример 2

```
; изменение скорости
Function cambiovel
  setvel X, 20
  setvel Y, 20
  movinc X, 100, Y, 200
  movinc X, 150, Y, 180
  setvel X, 5
  movinc X, 80, Y, 100
  waitsti X, Y
ll
fret
```

MULTIABS

Синтаксис

MULTIABS [**ярлык**],**ось1**, **значение1**, [**ось2**, **значение2** [, **ось3**, **значение3** [, ..., **ось16**, **значение 16**]]]

Аргументы

ярлык	константа или переменная integer. Ярлык, определяющий блок смещения
ось1...ось16	имена устройств типа оси
значение1...[...значение16]	константа или переменная. Значение теоретической координаты конца блока смещения

Описание

Абсолютная многолинейная интерполяция до 16 осей. Это движение интерполяции допускает опережение профилей скорости, задавая с помощью инструкцией [SETTOLERANCE](#) соответствующие допуски на осях (под допуском оси понимается часть маршрута, на котором может не существовать отношения постоянной интерполяции). Порядок ввода осей в инструкцию MULTIABS **должен** всегда быть одинаковым и **все** участвующие в движении оси должны присутствовать. Блоки смещения помещаются в очередь в обычный буфер опережающего просмотра и движение запускается при выполнении инструкции [WAITSTILL](#), [STARTINTERP](#) или при заполнении буфера опережающего просмотра. Среди осей, задействованных в движении, одна может использоваться в качестве ускорителя при помощи инструкции [WAITCOLL](#). Дополнительный параметр **ярлык** используется вместе с инструкцией [SETLABELINTERP](#) для однозначного определения блока смещения. Оси шагового типа могут использоваться в этой инструкции только при их управлении удаленным TRS-AX. В этом случае необходимо учитывать, что под термином "интерполяция" понимается координированное движение нескольких осей с определенной погрешностью, обуславливаемой методом управления осью.

Примечание

При интерполяции такого типа невозможно использовать виртуальные системы координат (инструкция [SETRIFLOC](#) и [RESRIFLOC](#)). Можно выполнять движения с сцепленными осями (в [CHAIN](#)). Оси, задействованные в многоосевом интерполированном движении, должны объявляться главными для других осей, не задействованных в движении. Кроме того, можно применять FeedRate Override.

Пример

```

setquote      x, 0
setquote      y, 0
setquote      z, 0
; первый блок
setveli       x, velx1
setveli       y, vely1
setveli       z, velz1
multiabs      x, координатаx1, y, координатаy1,
              z, координатаz1
; второй блок
settolerance  x, tollx2, y, tolly2, z, tollz2
setveli       x, velx2
setveli       y, vely2
setveli       z, velz2
multiabs      x, координатаx2, y, координатаy2,
              z, координатаz2
; третий блок
settolerance  x, tollx3, y, tolly3, z, tollz3
setveli       x, velx3
setveli       y, vely3
setveli       z, velz3
multiabs      x, координатаx3, y, координатаy3,
              z, координатаz3
; четвертый блок
settolerance  x, tollx4, y, tolly4, z, tollz4
setveli       x, velx4
setveli       y, vely4
setveli       z, velz4
multiabs      x, координатаx4, y, координатаy4,
              z, координатаz4
waitstill     x, y, z

```

MULTIINC**Синтаксис****MULTIINC****[ярлык],ось1, значение1, [ось2, значение2 [, ось3, значение3 [, ..., ось16, значение 16]]]****Аргументы**

ярлык константа или переменная integer. Ярлык, определяющий блок смещения
ось1...ось16 имена устройств типа оси
значение1...[...значение16] константа или переменная. Значение приращения теоретической координаты конца блока смещения

Описание

Многолинейная интерполяция с приращением до 16 осей. Это движение интерполяции допускает опережение профилей скорости, задавая соответствующим образом инструкцией [SETTOLERANCE](#) на осях соответствующие допуски (под допуском оси понимается часть маршрута, на котором может не существовать отношения постоянной интерполяции). Порядок ввода осей в инструкцию MULTIINC **должен** всегда быть одинаковым и **все** участвующие в движении оси должны присутствовать. Блоки смещения помещаются в очередь в обычный буфер опережающего просмотра и движение запускается при выполнении инструкции [WAITSTILL](#), [STARTINTERP](#) или при заполнении буфера опережающего просмотра. Среди осей, задействованных в движении, одна может использоваться в качестве ускорителя при помощи инструкции [WAITCOLL](#). Дополнительный параметр **ярлык** используется вместе с инструкцией [SETLABELINTERP](#) для однозначного определения блока смещения. Оси шагового типа могут использоваться в этой инструкции только при их управлении удаленным TRS-AH. В этом случае необходимо учитывать, что под термином "интерполяция" понимается координированное движение нескольких осей с определенной погрешностью, обуславливаемой методом управления осью.

Примечание

При интерполяции такого типа невозможно использовать виртуальные системы координат (инструкция [SETRIFLOC](#) и [RESRIFLOC](#)). Можно выполнять движения с сцепленными осями (в [CHAIN](#)). Оси, задействованные в многоосевом интерполированном движении, должны объявляться главными для других осей, не задействованных в движении. Кроме того, можно применять FeedRate Override.

NORMAL

Синтаксис
NORMAL **ось**

Аргументы
ось имя устройства типа оси

Описание
 Подключает контроль положения на **оси** и отключает сцепление осей.
 При включении системы все сконфигурированные оси приводятся в свободное состояние и переходят в нормальное состояние при выполнении этой инструкции или при выполнении первого движения.
 В любом случае, рекомендуется выполнить эту инструкцию перед выполнением процедуры обнуления осей для сброса возможных аварийных условий.

RESRIFLOC

Синтаксис
RESRIFLOC **ось1, ось2, ось3**

Аргументы
ось1...ось3 имена устройств типа оси

Описание
 Восстанавливает абсолютную систему координат для осей X Y Z (**ось1, ось2, ось3**).
 Обычно эта инструкция используется после задания системы координат с переносом и поворотом при помощи инструкции [SETRIFLOC](#).

SETINDEXINTERP

Синтаксис
SETINDEXINTERP **ось, имяпер**

Аргументы
ось имя устройства типа оси
имяпер имя глобальной переменной типа integer

Описание
 Определяет индекс, подсчитывающий число выполненных осью блоков интерполяции.
 В ходе интерполированного движения при каждой смене блока переменная **имяпер** приращается на 1.

Примечание
 Используемая в качестве индекса переменная должна быть глобальной переменной группы или глобальной переменной станка.

SETLABELINTERP

Синтаксис
SETLABELINTERP **ось, значение**

Аргументы
ось имя устройства типа оси
значение имя глобальной переменной типа integer

Описание
 В переменной **значение** при интерполированном движении каждой смене блока задается значение ярлыка нового блока. Ярлык определяется в инструкциях интерполированного движения.

Примечание
 Переменная **значение** должна быть глобальной переменной группы или глобальной переменной модуля.

SETPFLY

Синтаксис
SETPFLY **ось, состояние, скорость, координата, [ошибка]**

Аргументы
ось имя устройства типа оси

состояние	заданная константа. Указывает состояние проверяемого микровыключателя. Могут задаваться следующие значения: ON OFF
скорость	константа или плавающая переменная.
координата	константа или переменная.
ошибка	целая переменная. Код ошибки.

Описание

Позволяет выполнять обнуление координаты оси "на лету". Обнуление запускается выключателем, подключенным к быстрому входу разъема оси (на платах с шиной MECHATROLINK-II используется EXT11).

При движении **оси**, выполняемом при помощи инструкции MOVABS, инструкция ожидает, пока соответствующий микровыключатель обнуления переключится в указанное **состояние**. После определения этого перехода реальная координата обнуляется без остановки движения и автоматически вновь динамически определяются целевая **координата** прибытия и **скорость**. Таким образом, восстанавливается правильное движение для достижения координаты прибытия и при необходимости выполняется также инверсия движения. Если заданная координата достигается без определения переключения входа и задания параметра **ошибка**, генерируется ошибка системы. Если был задан параметр **ошибка**, он будет содержать цифровой код соответствующей ошибки системы. В этом случае обнуление не было выполнено и для повторного задания поиска микровыключателя необходимо выполнить инструкцию [SETQUOTE](#).

Для прерывания выполнения обнуления "налету" достаточно выполнить инструкцию NORMAL на оси или завершить задачу, которая потребовала выполнения обнуления.

При выполнении команды невозможно одновременно выполнять команды [SETPZERO](#) и [FASTREAD](#) на одной оси при подключении к платам шиной MECHATROLINK-II.

Пример

[Стандартная процедура обнуления на прерывании](#)

SETPFLYCHAINSTRAT**Синтаксис**

SETPFLYCHAINSTRAT ось, [тип]

Аргументы

ось	имя устройства типа оси
тип	Константа типа integer. Допускаются следующие значения: 0 = только главная ось обнуляет координату, ведомая ось сохраняет предыдущую координату отлично от 0 = главная и ведомая оси синхронно обнуляют координату

Описание

Эта инструкция позволяет задавать порядок поведения указанной ведомой оси при подаче инструкции [SETPFLY](#) на главную ось.

Эта инструкция выполняется на ведомой оси. Если переменная тип опускается, принимается значение по умолчанию, равное 0.

SETPZERO**Синтаксис**

SETPZERO ось, координата [,ошибка]

Аргументы

ось	имя устройства типа оси
координата	константа или переменная. Это инкрементальная координата
ошибка	целая переменная. Код ошибки.

Описание

Запускает движение с приращением **оси** до указанной **координаты** и ожидает обнаружения отметки нуля энкодера (до достижения указанной координаты).

В момент обнаружения отметки реальная координата задается на ноль и ось останавливается.

Если заданная координата достигается без обнаружения отметки нуля и задания параметра **ошибка**, генерируется ошибка системы. Если был задан параметр **ошибка**, он будет содержать цифровой код соответствующей ошибки системы. В этом случае уставка не была выполнена и для повторного задания поиска отметки необходимо выполнить инструкцию [SETQUOTE](#).

Перемещение осей, генерированное этой инструкцией, может прерываться инструкцией STOP и возобновляться инструкцией START.

Если команда выполняется для осей S-CAN и для осей EtherCAT, необходимо предварительно выполнить команду FREE.

При выполнении команды невозможно одновременно выполнять команды [SETPFLY](#) и [FASTREAD](#) на одной оси при подключении к платам шиной MECHATROLINK-II.

Пример

```
FREE          X
SETPZERO     X, 100
```

SETPZEROCHAINSTRAT

Синтаксис

```
SETPZEROCHAINSTRAT    ось, [значение]
```

Аргументы

ось	имя устройства типа оси
значение	переменная типа integer. Допускаются следующие значения: 0 = только главная ось обнуляет координату, ведомая ось сохраняет предыдущую координату отлично от 0 = главная и ведущая оси синхронно обнуляют координату

Описание

Эта инструкция позволяет задавать порядок поведения указанной ведомой оси при подаче инструкции [SETPFLY](#) на главную ось.

Эта инструкция выполняется на ведомой оси.

Если переменная **значение** опускается, принимается значение по умолчанию, равное 0.

SETQUOTE

Синтаксис

```
SETQUOTE              ось, координата
```

Аргументы

ось	имя устройства типа оси
координата	константа или переменная

Описание

Эта инструкция контекстуально принудительно задает теоретическую координату и реальную координату оси на значение, указанное **координата**. Если ось движется, это вызывает резкую остановку оси, учитывая, что для блока управления ось неожиданно оказывается на координате (реальная координата совпадает с целевой координатой). Поэтому не рекомендуется использовать эту инструкцию на оси при движении, если она не имеет слишком низкую скорость.

Пример

[Стандартная процедура обнуления оси](#)

SETQUOTECHAINSTRAT

Синтаксис

```
SETQUOTECHAINSTRAT   ось, [значение]
```

Аргументы

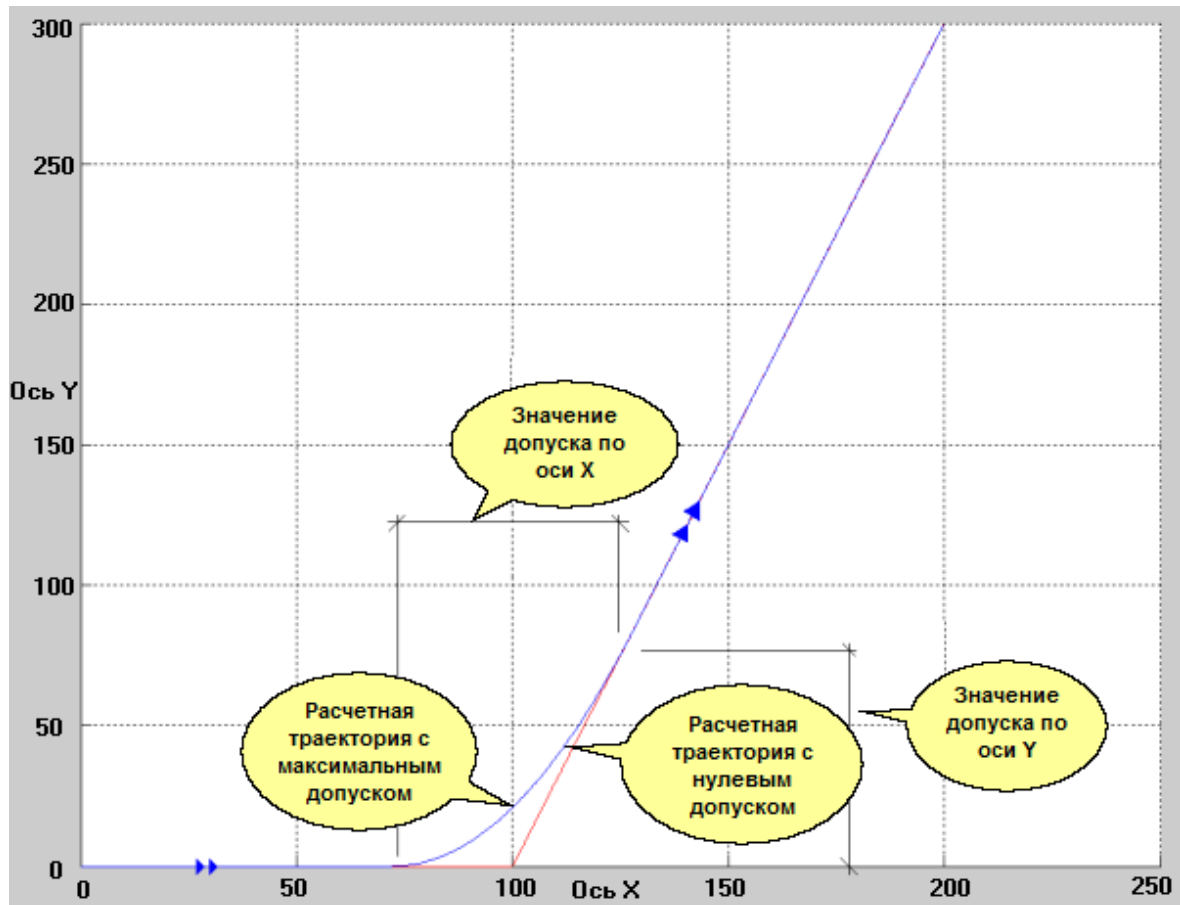
ось	имя устройства типа оси
значение	Переменная типа integer. Допускаются следующие значения: 0 = только главная ось инициализируется на новую координату, ведомая ось сохраняет предыдущую координату отлично от 0 = координаты ведомой оси инициализируются синхронно с координатами главной оси

Описание

Эта инструкция позволяет задавать порядок поведения указанной ведомой оси при подаче инструкции [SETQUOTE](#) на главную ось.

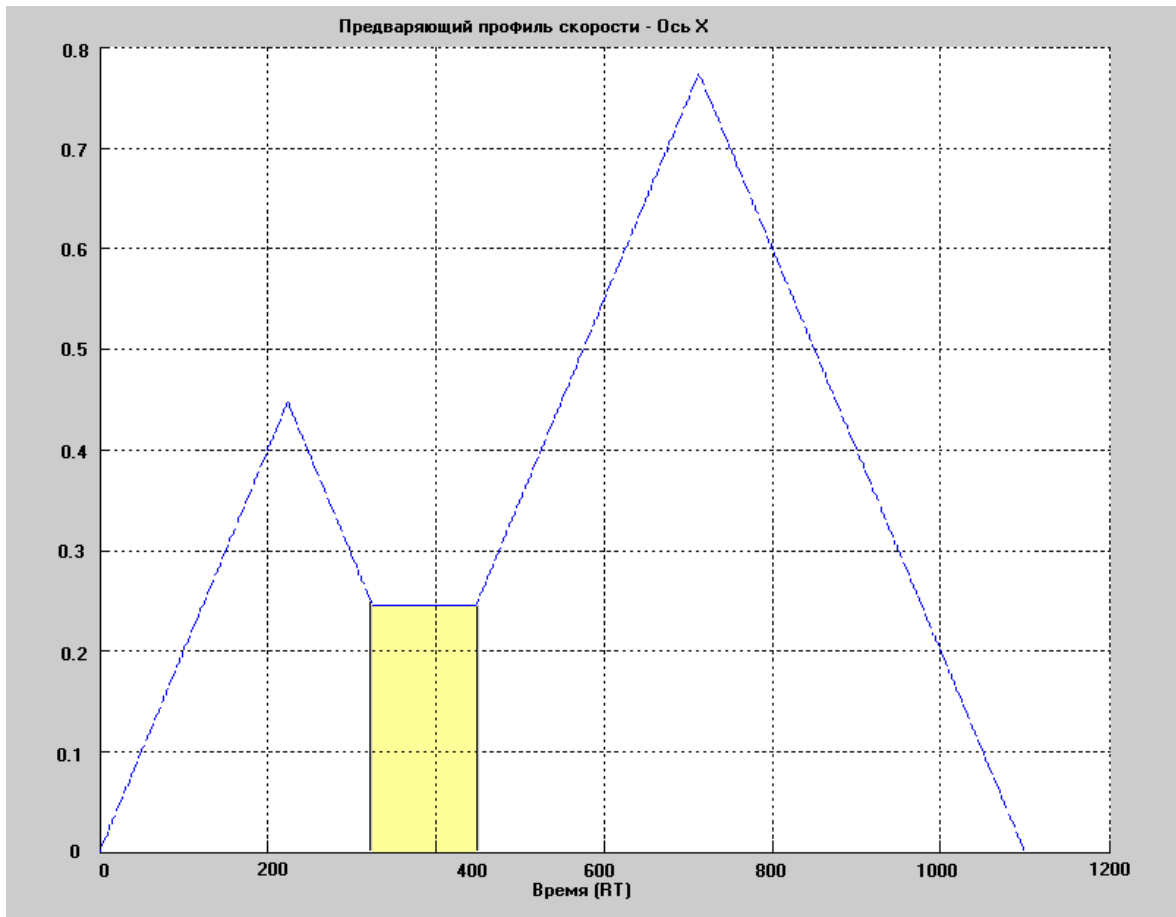
Эта инструкция выполняется на ведомой оси.

Если переменная **значение** опускается, принимается значение по умолчанию, равное 0.



На рисунке выше представлена классическая многоосевая траектория, состоящая из двух блоков движения, первый из которых состоит в смещении на 100 оси X, а второй - в перемещении оси Y на 300 и перемещении все также на 100 оси X. Красная линия показывает траекторию при нулевом допуске, а синяя - траекторию при максимальном допуске.

Допуск может также рассматриваться как зона, противоположная профилю скорости в интервале опережения, как показано на рисунке ниже.



START

Синтаксис

START **ось**

Аргументы

ось имя устройства типа оси

Описание

Восстановление движения **оси** после остановки.

STARTINTERP

Синтаксис

STARTINTERP **ось**

Аргументы

ось имя устройства типа оси

Описание

Запускает интерполяцию, канал которой определяется **осью**. Обычно движение осей, ассоциированных с каналом интерполяции, начинается при полном заполнении буфера интерполяции (512 инструкций) или при достижении инструкции WAITSTILL, которая затем завершает движение. Это позволяет алгоритму интерполятора определить оптимальные профили скорости, имея в распоряжении информацию, относящуюся к высокому числу (или всем) отрезков интерполированного движения. Инструкция STARTINTERP позволяет принудительно задавать запуск движения осей даже если не удовлетворяются только что описанные условия.

STOP

Синтаксис

STOP **ось**

Аргументы

ось имя устройства типа оси

Описание

Остановка движения **оси**. Ось выполнит функцию замедления, продолжительность которой зависит от фактической скорости и от параметров конфигурации.

Пример

[Стандартная процедура обнуления оси](#)

SWITCHENC**Синтаксис**

SWITCHENC ось1, [ось2, [направление, координата]]

Аргументы

ось1 имя устройства типа оси
ось2 имя устройства типа оси. Указывает ось подсчета
направление заданная константа.
UP=переключение кодера при превышении координаты в положительном направлении
DOWN=переключение кодера при превышении координаты в отрицательном направлении
координата константа double или переменная double

Описание

Позволяет заменить кодер **оси1** кодером **оси2**. Переключение кодера выполняется при превышении в положительном (UP) или отрицательном (DOWN) **направлении указанной координаты**. При опускании параметров **направление** и **координата** переключение кодера выполняется сразу же, независимо от положения осей.

Если объявляется только **ось1**, возобновляется работа с одним кодером.

Ось1 не может быть шаговой, осью подсчета или виртуальной, **Ось2** может быть только осью подсчета. Кроме того, как **ось1**, так и **ось2** не могут быть задействованы в сцепленных движениях как ведомая ось.

Эта инструкция генерирует ошибку системы 4101 - Непоследовательное управление осью ИмяОси, если **ось1** или **ось2** объявлена ведомой в сцепленном движении, или если **ось1** выполняет инструкцию FASTREAD или инструкцию SETPFLY. Кроме того, может быть сгенерирована системная ошибка 4105 - Инструкция не может быть выполнена на оси ИмяОси, если объявленный тип оси не является одним из возможных.

WAITACC**Синтаксис**

WAITACC ось1 [, ..., ось6]

Аргументы

ось1 [, ...,ось6] имя устройства типа оси

Описание

Ожидание состояния ускорения или одного из следующих состояний на всех указанных **осях** (1ч6). Задача, в которой выполняется инструкция, помещается в ожидание до тех пор, пока ось не окажется в состоянии ускорения или одном из последующих состояний.

Состояния оси определяются целым числом:

- ускорение = 1
- режим = 2
- замедление = 3
- координата = 4
- ожидание большого окна = 5
- ожидание маленького окна = 6
- ожидание остановленной оси = 7

WAITCOLL

Синтаксис
WAITCOLL

ось, значение, таймаут, дельта

Аргументы

ось
значение
таймаут
дельта

имя устройства типа оси
 константа или переменная. Значение абсолютной координаты
 константа или переменная. Это время ожидания при остановленной оси
 константа или переменная. Это значение окна для получения
 остановленной оси

Описание

При перемещении оси достижению запрограммированной координаты может мешать препятствие механического характера, иногда представленное самой обрабатываемой деталью. В этом случае система генерирует ошибку системы "сервоошибка" или "движение не завершено". Этой инструкцией определяется **значение** координаты, на которой система начинает проверять наличие столкновения, продолжительность ожидания (**таймаут**) до того, как **ось**, после столкновения, приводится в состояние "координата" и **дельта**, определяющая допуск на позиционировании оси. После того, как ось зайдет за координату, определенную в **значении**, проверяется, движется ли еще ось. После перехватывания препятствия определяется критическая ситуация и, хотя при этом и гарантируется толкающее усилие двигателя, больше не проверяется превышение погрешности контура. Направление движения, на котором проверяется произошедшее столкновение, - то же, что и направление последнего находящегося в очереди движения. **Таймаут** выражается в секундах, значение **дельта** должно быть больше 0.001 мм и меньше разницы между запрограммированной координатой прибытия и координатой **значения**. Эта инструкция может использоваться с многоосным интерполятором с момента, когда в этом интерполяторе принимается временная утрата связи интерполяции. Эта инструкция может применяться также к главным осям в сцепленном движении.

В следующих случаях генерируется ошибка системы:

- **ось** выполняет классическое интерполированное движение (см. инструкции LINEARBS, LINEARINC, CIRCABS, CIRCINC, HELICABS, HELICINC) или в координированном движении или интерполированные движения ISO.
- **ось** - ведомая ось
- **ось** - это ось подсчета или шаговая ось или виртуальная ось
- заданное **значение** превышает координату конца движения

Пример

```
; задает координату оси X
SETQUOTE X, 0.0
; перемещает ось X на абсолютную координату 1000
MOVABS X, 1000.0
; ожидает координату столкновения, ожидает 2 секунды
; перед приведением оси в состояние "координата" после
; перехватывания столкновения с точностью 0.01 мм
WAITCOLL X, 980.0,2.0,0.01
```

WAITDEC

Синтаксис
WAITDEC

ось1 [, ..., ось6]

Аргументы

ось1 [...,ось6]

имя устройства типа оси

Описание

Ожидание состояния замедления или одного из следующих состояний на всех указанных **осях** (1ч6). Задача, в которой выполняется инструкция, помещается в ожидание до тех пор, пока ось не окажется в состоянии замедления, координаты, ожидания большого окна, ожидания маленького окна и ожидания остановленной оси.

Состояния оси определяются целым числом:

- ускорение = 1
- режим = 2
- замедление = 3
- координата = 4
- ожидание большого окна = 5
- ожидание маленького окна = 6
- ожидание остановленной оси = 7

WAITREG

Синтаксис

WAITREG ось1 [, ..., ось6]

Аргументы

ось1 [...,ось6] имя устройства типа оси

Описание

Ожидание состояния режима или одного из следующих состояний на всех указанных **осях** (1ч6).
Задача, в которой выполняется инструкция, помещается в ожидание до тех пор, пока ось не окажется в состояниях режима, замедления, координаты, ожидания большого окна, ожидания маленького окна и ожидания остановленной оси.

Состояния оси определяются целым числом:

- ускорение = 1
- режим = 2
- замедление = 3
- координата = 4
- ожидание большого окна = 5
- ожидание маленького окна = 6
- ожидание остановленной оси = 7

WAITSTILL

Синтаксис

WAITSTILL ось1 [, ..., ось6]

Аргументы

ось1 [...,ось6] имя устройства типа оси

Описание

Ожидание для всех **осей** (1÷6), определенных окончанием движения (состояние Координата).

Пример

[Стандартная процедура обнуления оси](#)

WAITTARGET

Синтаксис

WAITTARGET ось1 [, ..., ось6]

Аргументы

ось1 [...,ось6] имя устройства типа оси

Описание

Ожидание, пока для всех указанных **осей** (1÷6) текущая теоретическая координата будет равна конечной координате. Реальная координата будет отличаться от теоретической до тех пор, пока не будет сброшена погрешность контура.

WAITWIN

Синтаксис

WAITWIN ось1 [, ..., ось6]

Аргументы

ось1 [...,ось6] имя устройства типа оси

Описание

Ожидание состояния окна или одного из следующих состояний на всех указанных **осях** (1ч6).
Задача, в которой выполняется инструкция, помещается в ожидание до тех пор, пока ось не окажется в состоянии ожидания большого окна, ожидания маленького окна и ожидания остановленной оси.

Состояния оси определяются целым числом:

- ускорение = 1
- режим = 2
- замедление = 3
- координата = 4
- ожидание большого окна = 5

- ожидание остановленной оси = 6
- ожидание маленького окна = 7

Параметры оси

Считывание / Запись

DEVICEID

Синтаксис

DEVICEID

устройство, переменная

Аргументы

устройство
переменная

имя устройства или параметр устройства
целая переменная, которой присваивается логический адрес

Описание

Записывают в **переменной** логический адрес, назначенный **устройству**, какого бы типа оно ни было. Эта команда дает возможность иметь однозначный "ключ", присвоенный устройству, такой как индекс или ключ поиска в структуре данных.

GETAXIS

Синтаксис

GETAXIS
GETAXIS

ось, имяданных, имяпер
ось, имяданных1, имяданных2, [... , имяданных20,] матрица[строка]

Аргументы

ось
имяданных

имя устройства типа оси
заданная константа (См. перечень в следующей части). Параметр оси (1ч20)

имяпер
строка
матрица

переменная или имя устройства
константа или переменная integer. Номер строки матрицы
имя матрицы

Описание

В первой форме инструкция считывает одно из данных (**имяданных**) оси и обеспечивает его запись в переменной.

Во второй форме эта инструкция считывает в один момент несколько данных одной оси (от 1 до 20) и записывает их в порядке, в котором они были запрошены, в элементы строки указанной матрицы.

В этом случае число столбцов матрицы должно быть равно числу требуемых данных.

Приведенный ниже перечень содержит все заданные константы, которые могут задаваться параметру **имяданных**.

Первый столбец - имя константы.

Второй столбец - описание величины оси, которая считывается инструкцией.

Третий - это формат данных, который возвращается в переменной **имяпер** или в **матрица[строка]**, где:

- **d** означает **double**
- **f** означает **float**
- **i** означает **integer (целый)**
- **b** означает **char (признак)**

Если описание переменной, в которой будут запоминаться данные, отличается от значения, возвращенного командой, компилятор выполняет преобразование (приведение) данных к типу, требуемому пользователем. Иногда это приводит к значительной утрате данных. Например, значение типа **double**, равное 12.345, при преобразовании в целое число превращается в 12. Поэтому рекомендуется соблюдать в объявлении переменных **имяпер** и **матрица[строка]** требуемую типологию.

В последнем столбце указывается или возвращаемое значение, или единица измерения соответствующего параметра.

Константы, начинающиеся с "_CFG", позволяют считывать значения конфигурации, т. е. значения, которые задаются при включении станка.

константа	описание	тип	возвращаемое значение
_CFGTYPE	тип оси	i	1=аналоговая, 3=шаговая, 4=цифровая, 5=подсчета, 6=частоты/направления, 7=виртуальная
_CFGUM	единица измерения	i	0=миллиметры, 1=дюймы, 2=градусы, 3=обороты

константа	описание	тип	возвращаемое значение
_CFGRIS	разрешение	d	импульсы на ед. изм.
_CFGVMAX	максимальная скорость	f	м/1' или дюймы/1" или градусы/1" или об./1'
_CFGVMAXD	максимальная скорость в ручном режиме	f	м/1' или дюймы/1" или градусы/1" или об./1'
_CFGVMAXI	максимальная скорость интерполяции	f	м/1' или дюймы/1" или градусы/1" или об./1'
_CFGPHINV	инверсия фаз кодера	b	0=нет инверсии, 1=инверсия
_CFGRFINV	инверсия эталона	b	0=нет инверсии, 1=инверсия
_CFGZIND	подключение сброса координаты отметки нуля	b	0=отключен, 1=подключен
_CFGSLOPE	тип линейной функции ускорения/замедления при движении точка-точка	b	0=линейная, 1=S-образная, 2= в виде двойной S
_CFGKFFA	Подача вперед по цепи для ускорения	f	
_CFGKFFAI	Подача вперед по цепи для интерполяции ускорения	f	
_CFGSRPP	скорость начала линейной функции шагового движения	f	м/1' или дюймы/1" или градусы/1" или об./1'
_CFGACC	продолжительность ускорения от 0 до _CFGVMAX	i	мс
_CFGDEC	продолжительность замедления от _CFGVMAX до 0	i	мс
_CFGACCI	продолжительность ускорения от 0 до _CFGVMAXI	i	мс
_CFGDECI	продолжительность замедления от _CFGVMAXI до 0	i	мс
_CFGQLP	предел оси в положительном направлении	d	координата
_CFGQLN	предел оси в отрицательном направлении	d	координата
_CFGKP	пропорциональный коэффициент	f	
_CFGKI	интегрирующий коэффициент	f	
_CFGKD	производный коэффициент	f	
_CFGKFF	связь вперед по цепи	f	процент
_CFGKPS	пропорциональный коэффициент ведомой оси	f	
_CFGKIS	интегрирующий коэффициент ведомой оси	f	
_CFGKDS	производный коэффициент ведомой оси	f	
_CFGQEAP	погрешность контура в положит. направлении	d	координата
_CFGQEAN	погрешность контура в отриц. направлении	d	координата
_CFGKPI	пропорциональный коэффициент интерполяции	f	

константа	описание	тип	возвращаемое значение
_CFGKII	интегрирующий коэффициент интерполяции	f	
_CFGKDI	производный коэффициент интерполяции	f	
_CFGTMINP	минимальное положительное напряжение	f	Вольт
_CFGTMINN	минимальное отрицательное напряжение	f	Вольт
_CFGSTMINP	положительное пороговое напряжение	f	Вольт
_CFGSTMINN	отрицательное пороговое напряжение	f	Вольт
_CFGESC	таймаут перемещ. оси	i	мс
_CFGDSE	подключение динамической сервошибки	b	0=отключен, 1=подключен
_CFGAEAN	подключение автоматической настройки	b	0=отключен, 1=подключен
_CFGOFFSET	напряжение настройки - начальный сдвиг	f	Вольт
_CFGCEE	неправильная координата подключения кодера	d	координата
_CFGNOTCH	частота фильтра отметки	i	Гц
_CFGBUFI	размеры буфера интегрирующего расчета	i	[1, 200]
_CFGQAP	окно прибытия на коорд. положит.	d	
_CFGQAN	окно прибытия на коорд. отриц.	d	
_CFGSLOPEI	тип линейной функции ускорения/замедления при интерполяции	f	0=линейная, 1=S-образная, 2= в виде двойной S
_CFGKFFI	связь вперед по цепи для интерполяции	f	процент
_CFGAAF	ожидание остановленной оси	b	0=отключен, 1=подключен
_CFGENCTYPE	тип кодера	i	0=смоделированный или отсутствует, 1=реальный
_SRPP	скорость начала линейной функции шагового движения	f	м/1' или дюймы/1" или градусы/1" или об./1'
_ACC	продолжительность ускорения от 0 до _VMAX	i	мс
_DEC	продолжительность замедления от _VMAX до 0	i	мс
_ACCI	продолжительность ускорения от 0 до VMAXI с интерполяцией	i	мс
_DECI	продолжительность замедления от _VMAX до 0 с интерполяцией	i	мс
_QLP	предел оси в положительном направлении	d	координата
_QLN	предел оси в отрицательном направлении	d	координата

константа	описание	тип	возвращаемое значение
_KP	пропорциональный коэффициент	f	
_KI	интегрирующий коэффициент	f	
_KD	производный коэффициент	f	
_KFF	связь вперед по цепи	f	процент
_KPS	пропорциональный коэффициент ведомой оси	f	
_KIS	интегрирующий коэффициент ведомой оси	f	
_KDS	производный коэффициент ведомой оси	f	
_QEAP	погрешность контура в положит. направлении	d	координата
_QEAN	погрешность контура в отриц. направлении	d	координата
_VEL	скорость при движении точка-точка	f	м/1' или дюймы/1" или градусы/1" или об./1'
_VELI	скорость интерполяции	f	м/1' или дюймы/1" или градусы/1" или об./1'
_MODE	режим функционирования оси	b	1=нормальный, 2=свободный, 8=интерпол., 10=скоорд.
_PHINV	инверсия фаз кодера	b	0=нет инверсии, 1=инверсия
_RFINV	инверсия эталона	b	0=нет инверсии, 1=инверсия
_ZIND	подключение сброса координаты отметки нуля	b	0=отключен, 1=подключен
_KPI	пропорциональный коэффициент интерполяции	f	
_KII	интегрирующий коэффициент интерполяции	f	
_KDI	производный коэффициент интерполяции	f	
_KFFI	связь вперед по цепи для интерполяции	f	процент
_KFFA	связь вперед по цепи для ускорения	f	процент
_KFFAI	связь вперед по цепи для ускорения интерполяции	f	процент
_ESC	таймаут перемещ. оси	i	мс
_CEE	неправильная координата подключения кодера	d	координата
_NOTCH	частота фильтра отметки	i	Гц
_BUFI	размеры буфера интегрирующего расчета	i	[1,200]
_QAP	окно прибытия на коорд. положит.	d	
_QAN	окно прибытия на коорд. отриц.	d	
_QEAPINV	предел погрешности контура в положительном направлении с инверсией	d	
_QEANINV	предел погрешности контура в отрицательном направлении с инверсией	d	
_OFSCCOORD	координата сдвига скоординированного движения	d	

константа	описание	тип	возвращаемое значение
_MS	тип оси: главная или ведомая	b	0=не в цепи, 4=главная, 5=ведомая
_QENC	координата кодера	d	координата
_QR	реальная координата	d	координата
_RIS	разрешение, используемое осью	d	
_ST	состояние оси	b	1=ускорение, 2=режим, 3=замедление, 4=координата, 5=ожидание большого окна, 6=ожидание остановленной оси, 7=ожидание маленького окна, 8=пуск
_QT	теоретическая координата	d	координата
_EA	погрешность контура	d	координата
_FF	связь вперед по цепи	i	
_VC	текущая скорость	f	
_P	корректор пропорциональной составляющей	i	
_I	корректор интегрирующей составляющей	i	
_D	корректор производной составляющей	i	
_FLGS	флажки осей	b	
_VCR	реальная текущая скорость	f	
_ADJUST	компенсация сдвига оси	i	целое значение, представляющее напряжение, которое должно передаваться приводу, если смотреть со стороны платы осей dac. Предел измерения для привода составляет 10 В, а для dac - 32767
_DAC	DAC	i	целое значение, представляющее напряжение, которое должно передаваться приводу, если смотреть со стороны платы осей dac. Предел измерения для привода составляет 10 В, а для dac - 32767.
_ACCINST	мгновенное значение ускорения	f	
_FFA	связь вперед по цепи для ускорения	i	
_GONETIME	время, истекшее с начала движения	f	с (0 для ведомых осей и пошаговых осей)
_RESTIME	время, остающееся до конца движения. Значения относятся к движению, находящемуся в буфере в момент запроса	f	с (0 для ведомых осей, осей с координированным движением и пошаговых осей и движение ISO)
_TARGETTIME	время, затраченное на генерирование конечной координаты	f	микросекунды
_GONESPACE	пространство, пройденное с начала движения. Значения относятся к движению, находящемуся в буфере в момент запроса	f	процент (100 для ведомых осей, осей с координированным движением и пошаговых осей)
_RESSPACE	пространство, остающееся до конца движения. Значения относятся к движению,	f	процент (0 для ведомых осей, осей с координированным движением и пошаговых осей)

константа	описание	тип	возвращаемое значение
	находящемуся в буфере в момент запроса		
_AXESJERK	подключение на оси управления толчковым режимом	b	1=управление подключено, 0=управление не подключено
_MOVEJERK	подключение управления толчковым режимом на движении, которое выполняет ось	b	1=управление подключено, 0=управление не подключено
_MOVETYPE	тип движения, которое выполняет ось	b	1=классическое интерполированное движение, 2=многоосевое интерполированное движение, 3=скоординированное движение, 4=движение точка-точка, 5=сцепленное движение (только ведомые оси)
_PARTYPESET	тип параметров оси, используемых при движении	i	1=интерполяция, 0=точка-точка
_AXINRIFLOC	ось захвачена в локальной системе координат	i	1=да, 0=нет
_QTARGETTOOL	целевая координата оси. Если интерполяция - ISO, целевая координата кончика инструмента соответствует оси	d	
_QREALTOOL	реальная координата оси. Если интерполяция - ISO, реальная координата кончика инструмента соответствует оси	d	
_BACKLASH	значение механического зазора, определенного для оси	d	
_DISABLED	отключение оси	b	1=ось отключена, 0=ось подключена
_DYNLIMIT	подключение динамического контроля пределов осей	b	1=управление подключено, 0=управление не подключено
_AXESFEED	значение feedrate override, примененное в данный момент к оси	f	
_CORRLIN	тип используемого корректора линейности	i	0=не используется никакого корректора, 1=автокорректор, 2=перекрестный корректор, 3=автокорректор вместе с перекрестным корректором
_VELISO	скорость кончика инструмента при перемещении 5 осей	f	
_ISOSTOPS	число принудительных остановок интерполированного движения в результате предельных ситуаций при управлении опережающим просмотром	i	
_CURRATIO	значение используемого в данный момент отношения сцепления	d	
_DYNRATIO	возвращает, если выполняется динамическое изменение отношения сцепления	i	0=нет, 1 = да

константа	описание	тип	возвращаемое значение
_RESBLOCK	число блоков смещения, которые еще необходимо выполнить	i	
_EXECBLOCK	число блоков смещения выполнены	i	
_TOTALBLOCK	число блоков смещения в общей сложности в очереди в движении (текущее значение)	i	
_SWITCHENC	контролирует, выполняется ли в данный момент переключение кодера	i	-1=ось не использует инструкцию SWITCHENC, 0=была выполнена инструкция SWITCHENC, но ось использует свой кодер, 1= была выполнена инструкция SWITCHENC и ось использует кодер оси подсчета
_QOFSENC	значение сдвига кодера	d	
_LENSPZERO	расстояние, пройденное для достижения отметка нуля	d	координата
_TORQUEINST	мгновенное значение момента	i	
_CURRSLOPE	возвращает тип функции, используемой в данный момент в движениях в быстром режиме	i	0=линейная, 1=S-образная, 2= в виде двойной S
_CURRSLOPEI	возвращает тип функции, используемой в данный момент в интерполированных движениях	i	0=линейная, 1=S-образная, 2= в виде двойной S
_REALSLOPEI	возвращает тип функции, используемой в интерполированных движениях	i	0=линейная, 1=S-образная, 2= в виде двойной S
_AXISPAR1...AXIS PAR8	дополнительные параметры для оси EtherCAT	i	число
_ISOMOVETYPE	Тип движения ISO на выполнении	i	0= движение ISO быстрое, 1 = движение ISO интерполированное, -1 = другое
_QMAINENC	Реальная координата основного кодера при использовании вторичного кодера	d	

Движение Точка-точка

SETACC

Синтаксис

SETACC

ось, [значение]

Аргументы

ось

имя устройства типа оси

значение

константа или переменная. Длительность ускорения

Описание

Задает **оси** продолжительность ускорения, определенную **значением**. Продолжительность ускорения выражается в миллисекундах.

Если **значение** опускается, принимается параметр конфигурации. Если эта инструкция помещается между двумя инструкциями MOVABS или MOVINC, выполняется первая инструкция перемещения (с остановкой движения) с использованием заданных ранее параметров ускорения и замедления. Вторая инструкция выполняется с применением новых параметров ускорения. Инструкция SETACC воздействует только на движения, следующие за ее выполнением.

Если указанное **значение** меньше значения конфигурации, принимается значение конфигурации.

См. также [SETDEC](#), [SETACCI](#) и [SETDECI](#).

SETDEC

Синтаксис
SETDEC

ось, [значение]

Аргументы

ось имя устройства типа оси
значение константа или переменная. Длительность замедления

Описание

Задаёт **оси** продолжительность замедления, определённую **значением**. Продолжительность замедления выражается в миллисекундах. Если **значение** опускается, принимается параметр конфигурации. Если эта инструкция помещается между двумя инструкциями MOVABS или MOVINC, выполняется первая инструкция перемещения (с остановкой движения) с использованием заданных ранее параметров ускорения и замедления. Вторая инструкция выполняется с применением новых параметров замедления. Инструкция SETACC воздействует только на движения, следующие за ее выполнением. Если указанное **значение** меньше значения конфигурации, принимается значение конфигурации.

См. также [SETACC](#), [SETACCI](#) и [SETDECI](#).

SETDERIV

Синтаксис
SETDERIV

ось [, значение]

Аргументы

ось имя устройства типа оси
значение константа или переменная. Коэффициент производного действия. Не допускаются значения типа char и integer

Описание

Задаёт **оси значение** коэффициент производного действия.
Если **значение** опускается, используется коэффициент производного действия конфигурации.
Эта инструкция неприменима к шаговому двигателю.
См. также инструкцию [SETDERIVI](#).

SETFEED

Синтаксис
SETFEED

ось, значение

Аргументы

ось имя устройства типа оси
значение константа или переменная. Представляет процент feed rate override

Описание

Меняет **значение** в процентах feed rate override **оси** относительно *движений точка-точка*. См. также [SETFEEDI](#).

SETFEEDF

Синтаксис
SETFEEDF

ось [, значение]

Аргументы

ось имя устройства типа оси
значение константа или переменная. Процент feed forward

Описание

Задаёт **оси значение** в процентах feed forward.
Если **значение** опускается, используется коэффициент feed forward конфигурации.
При применении этой инструкции к шаговому двигателю генерируется ошибка системы. То же происходит в случае, если переменной **значение** задается значение, не находящееся в интервале от 0 до 100.
См. также инструкции [SETFEEDI](#), [SETFEEDFA](#), [SETFEEDFAI](#).

SETFEEDFA

Синтаксис
SETFEEDFA

ось [, значение]

Аргументы

ось
значение

имя устройства типа оси
константа или переменная. Процент feed forward

Описание

Задаёт **оси значение** в процентах *feed forward* ускорения для движения точка-точка. Если **значение** опускается, используется коэффициент *feed forward* конфигурации. При применении этой инструкции к шаговому двигателю генерируется ошибка системы. То же происходит в случае, если переменной **значение** задается значение, не находящееся в интервале от 0 до 100. См. также инструкции [SETFEEDF](#), [SETFEEDFI](#), [SETFEEDFAI](#).

SETINTEG

Синтаксис
SETINTEG

ось [, значение]

Аргументы

ось
значение

имя устройства типа оси
константа или переменная. Коэффициент интегрального действия. Не допускаются значения типа char и integer

Описание

Задаёт **оси значение** коэффициент интегрального действия. Если **значение** опускается, используется коэффициент интегрального действия конфигурации. Эта инструкция неприменима к шаговому двигателю. См. также инструкцию [SETINTEGI](#).

SETMULTIFEED

Синтаксис
SETMULTIFEED

ось1, значение1, ось2, значение2 [, ось3, значение3 [, ..., ось16, значение16]]]

Аргументы

ось1...ось16
значение1...[...значение16]

имена устройств типа оси
константа или переменная. Представляет процент feed rate override

Описание

Меняет **значение** в процентах feed rate override указанных **осей** относительно *движений точка-точка*. Для каждой оси может быть задано свое значение.

SETPROP

Синтаксис
SETPROP

ось [, значение]

Аргументы

ось
значение

имя устройства типа оси
константа или переменная. Коэффициент пропорционального действия. Не допускаются значения типа char и integer

Описание

Задаёт **оси значение** коэффициент пропорционального действия. Если **значение** опускается, используется коэффициент пропорционального действия конфигурации. Эта инструкция неприменима к шаговому двигателю. См. также инструкцию [SETPROPI](#).

SETSLOPE

Синтаксис
SETSLOPE

ось [, значение]

Аргументы

ось

имя устройства типа оси

значение константа или целая переменная. Тип изменения.

Описание

Задаёт тип функции, используемой для движения в быстром режиме:

- 0 линейная функция
- 1 S-образная функция
- 2 функция в виде двойной S

Если **значение** опускается, восстанавливается функция конфигурации.

Изменение типа функции может выполняться только с осью, остановленной в состоянии КООРДИНАТА. В противном случае генерируется ошибка системы 4101- Непоследовательное управление осью имяоси. Вместе с этой инструкцией можно проверить тип функции, используемой в данный момент осью, при помощи инструкции [GETAXIS](#) с параметром `_CURRSLOPE`.

См. также инструкцию [SETSLOPEI](#).

SETVEL

Синтаксис

SETVEL ось [, скорость]

Аргументы

ось имя устройства типа оси
скорость константа float или переменная float

Описание

Задаёт **максимальную** скорость **оси** для движений точка-точка.

Скорость выражается в единицах измерения оси, указанных в конфигурации.

Если запрограммированная **скорость** превышает максимальную скорость конфигурации, используется скорость конфигурации.

Если аргумент **скорость** опускается, принимается скорость конфигурации. Допускаются только положительные значения **скорости**.

См. инструкцию [SETVELI](#).

Пример

[Стандартная процедура обнуления оси](#)

Интерполированное Движение

LOOKAHEAD

Синтаксис

LOOKAHEAD [значение]

Аргументы

значение константа или переменная. Значение опережающего просмотра

Описание

Задаёт значение опережающего просмотра интерполятора. Опережающий просмотр - это число блоков интерполяции, которые обрабатываются до начала движения осей. Опережающий просмотр позволяет генерировать оптимальные профили движения, особенно при использовании S-образных функций.

Если не указывается параметр **значение**, система использует опережающий просмотр 512 блоков (по умолчанию).

Максимальное допустимое значение равно **4096/числоканалов**, где **числоканалов** - число каналов интерполяции, определенных в конфигурации модуля. Минимальное допустимое значение: 256.

Примечание

Под блоком интерполяции понимается совокупность информации, ассоциированной с любой инструкцией интерполированного движения (напр., LINEARABS).

Пример

`LOOKAHEAD 1024`

SETACCI

Синтаксис

SETACCI ось1 [, ..., ось6] [, значение]

Аргументы

ось1 [...,ось6] имя устройства типа оси

значение константа или переменная. Длительность ускорения

Описание

Задаёт осям **ось1**, **ось2** продолжительность ускорения для интерполированных движений, определённую **значением**. Время выражается в миллисекундах. Если **значение** опускается, принимается параметр конфигурации.

См. также [SETACC](#), [SETDEC](#) и [SETDECI](#).

SETACCLIMIT

Синтаксис

SETACCLIMIT ось,[значение]

Аргументы

ось имя устройства типа оси
значение константа времени привода

Описание

Подключает и отключает автоматический расчёт скорости режима интерполяции в зависимости от ускорений, допускаемых осями. Параметр **значение** - это временная константа, используемая для определения ступени скорости, допустимой для **оси**, в миллисекундах. Этот параметр является дополнительным. Если параметр опущен, инструкция отключает автоматический расчёт. Стандартное значение для этого параметра составляет 30 миллисекунд. При уменьшении этого времени произойдет замедление профиля, делая движение более плавным. При его увеличении будет достигнут противоположный эффект. Эта инструкция не может применяться к винтовым интерполяциям.

SETACCSTRATEGY

Синтаксис

SETACCSTRATEGY ось, [значение]

Аргументы

ось имя устройства типа оси
[значение] переменная или константа integer

Описание

Позволяет выбирать тип ускорения для следующих движений интерполяции. Эта инструкция должна выполняться для всех участвующих в интерполяции осей. Значения, допустимые для параметра **значение**: 0, 1 и 2. При передаче значения 0 будет приниматься обычная стратегия ускорения (в качестве ускорения профиля будет выбираться наименьшее среди всех участвующих в интерполяции осей). При значении, равном 2 и линейная интерполяция, будет принято максимальное ускорение, которое могут выдержать отдельные оси (с учетом отдельных компонентов всех осей, линейных и / или поворотных), управление ускорениями в случае круговой интерполяции остается неизменным. Случай значения 1 вызывает устаревшее управление, сохраненное для совместимости.

SETAXPARTYPE

Синтаксис

SETAXPARTYPE ось, [значение]

Аргументы

ось имя устройства типа оси
[значение] переменная или константа integer

Описание

При выполнении многолинейной интерполяции эта инструкция позволяет менять используемый набор параметров оси с переходом от типовых для интерполяции (**значение** = 1) к параметрам, используемым для перемещения точка-точка (**значение** = 0). При опускании переменной **значение**, используемые значения - значения интерполяции.

Изменение набора параметров может выполняться только при оси, остановленной в состоянии КООРДИНАТА, в противном случае инструкция генерирует ошибку системы 4101- Непоследовательное управление осью ИмяОси.

SETCONTORNATURE

Синтаксис

SETCONTORNATURE [значение1[,значение2]]

Аргументы

значение1 константа или переменная. Максимальный угол контурирования
значение2 константа или переменная. Максимальный угол замедления

Описание

Задаёт минимальный угол между касательными двух траекторий, выполненных с интерполяцией, за пределами которого станок не будет выполнять контурирование, т. е. оси будут остановлены по завершении первой траектории, а затем снова запущены по второй. Для этого определяется максимальный угол контурирования **значение1**, который представляет собой максимальный угол между двумя отрезками смещения, меньше которого движение не останавливается. Если угол между двумя блоками смещения превышает максимальный угол контурирования, движение останавливается. Для предупреждения остановки может задаваться значение значение максимального угла замедления **значение2**. Если угол между двумя блоками смещения находится между *максимальным углом контурирования* и *максимальным углом замедления*, движение не останавливается, а только замедляется. Таким образом, *максимальный угол замедления* представляет собой угол, за пределами которого движение должно обязательно останавливаться. Для углов меньше максимального угла контурирования движение **не замедляется**, для углов между максимальным углом контурирования и максимальным углом замедления движение **замедляется**, для углов больше максимального угла замедления движение **останавливается**.

Значение1 и **значение2** - это дополнительные параметры. Если оба они не задаются, по умолчанию принимается значение в 15 градусов. При задании только первого параметра подразумевается, что *максимальный угол замедления* равен *максимальному углу контурирования*. Функция замедления отключена, когда *максимальный угол замедления* меньше или равен *максимальному углу контурирования*. Максимальный угол замедления равен 180 градусам. При задании большего значения генерируется ошибка системы 4399: "Параметр за пределами диапазона". Функция замедления подключена только когда активна функция [JERKSMOOTH](#), а контурирование активно всегда.

Примечание

Использование этой инструкции связано с использованием инструкций [JERKSMOOTH](#) и [SETSLOWPARAM](#) и действует только в движениях с классической интерполяцией (инструкции [LINEARABS](#), [LINEARINC](#), [CIRCABS](#), [CIRCINC](#), [HELICABS](#), [HELICINC](#))

SETDECI

Синтаксис

SETDECI ось1 [, ..., ось6] [, значение]

Аргументы

ось1 [...,**ось6**] имя устройства типа оси
значение константа или переменная. Длительность замедления

Описание

Задаёт осям **ось1**, **ось2** продолжительность замедления для интерполированных движений, определённую **значением**. Время выражается в миллисекундах. Если **значение** опускается, принимается параметр конфигурации.

См. также [SETACC](#), [SETDEC](#) и [SETACCI](#).

SETDERIVI

Синтаксис

SETDERIVI ось [, значение]

Аргументы

ось имя устройства типа оси
значение константа или переменная. Коэффициент производного действия. Не допускаются значения типа char и integer

Описание

Задаёт **оси значение** коэффициент производного действия, используемый при движениях интерполяции осей.

Если **значение** опускается, используется коэффициент производного действия конфигурации. Эта инструкция неприменима к шаговому двигателю.

См. также инструкцию [SETDERIV](#).

SETFEEDFAI

Синтаксис

SETFEEDFAI ось [, значение]

Аргументы

ось имя устройства типа оси
значение константа или переменная. Процент feed forward

Описание

Задаёт **ось значение** в процентах *feed forward* ускорения для интерполированных движений. Если **значение** опускается, используется коэффициент *feed forward* конфигурации. При применении этой инструкции к шаговому двигателю генерируется ошибка системы. То же происходит в случае, если переменной **значение** задается значение, не находящееся в интервале от 0 до 100.

См. также инструкции [SETFEEDF](#), [SETFEEDFI](#), [SETFEEDFA](#).

SETFEEDI

Синтаксис

SETFEEDI ось, значение

Аргументы

ось имя устройства типа оси
значение константа или переменная. Представляет процент feed rate override

Описание

Изменяет **значение** в процентах *feed rate override* **ось** применительно к *интерполированным движениям*. См. также инструкцию [SETFEED](#).

SETFEEDFI

Синтаксис

SETFEEDFI ось [, значение]

Аргументы

ось имя устройства типа оси
значение константа или переменная. Процент feed forward

Описание

Задаёт **ось значение** в процентах *feed forward* для интерполированных движений. Если аргумент **значение** опускается, система принимает в качестве процента *feed forward* процент, определенный в параметрах конфигурации задействованного устройства оси.

Эта инструкция неприменима к шаговому двигателю.

Для переменной **значение** допускаются значения от 0 до 100.

См. также инструкции [SETFEEDF](#), [SETFEEDFA](#), [SETFEEDFAI](#).

SETINTEGI

Синтаксис

SETINTEGI ось [, значение]

Аргументы

ось имя устройства типа оси
значение константа или переменная. Коэффициент интегрального действия. Не допускаются значения типа *char* и *integer*

Описание

Задаёт **ось значение** коэффициент интегрального действия, используемый при движениях интерполяции осей.

Если **значение** опускается, используется коэффициент интегрального действия конфигурации.

Эта инструкция неприменима к шаговому двигателю.

См. также инструкцию [SETINTEG](#).

SETPROP1

Синтаксис
SETPROP1

ось [, значение]

Аргументы

ось
значение

имя устройства типа оси
константа или переменная. Коэффициент пропорционального действия.
Не допускаются значения типа char и integer

Описание

Задаёт **оси значение** коэффициент пропорционального действия, используемый при движениях интерполяции осей.

Если **значение** опускается, используется коэффициент пропорционального действия конфигурации. Эта инструкция неприменима к шаговому двигателю.

См. также инструкцию [SETPROP](#).

SETSLOPE1

Синтаксис
SETSLOPE1

ось [, значение]

Аргументы

ось
значение

имя устройства типа оси
константа или целая переменная. Тип изменения.

Описание

Задаёт тип функции, используемой для интерполированного движения (где допустимо):

- 0 линейная функция
- 1 S-образная функция
- 2 функция в виде двойной S

Если **значение** опускается, восстанавливается функция конфигурации.

Изменение типа функции может выполняться только с осью, еще не задействованной в канале интерполяции. В противном случае генерируется ошибка системы 4101- Непоследовательное управление осью имяоси.

Вместе с этой инструкцией, при помощи инструкции [GETAXIS](#) и параметра `_CURRSLOPE1` можно проверить тип функции, заданной в данный момент для оси, а при помощи параметра `_REALSLOPE1` можно отобразить тип функции (канала, в котором задействована ось), используемой осью.

См. также инструкцию [SETSLOPE](#).

SETSLOWPARAM

Синтаксис
SETSLOWPARAM

ось [,значение1,значение2]

Аргументы

ось
значение1
значение2

имя устройства типа оси
константа или переменная double. Общий коэффициент уменьшения
константа или переменная double. Коэффициент уменьшения инверсии

Описание

Эта инструкция меняет значения параметров, необходимых для расчета скорости замедления в случае, если активна функция замедления с контурированием (см. инструкцию [SETCONTORNATURE](#)).

Скорость замедления вычисляется вначале теоретически ось за осью. В случае инверсии движения может уменьшаться с использованием в расчетах **значения2**. Затем среди всех расчетных скоростей будет учитываться минимальная для соблюдения динамики наиболее ограничивающей оси. Наконец, можно еще больше уменьшить скорость замедления на коэффициент, зависящий от **значения1**.

Если **значение1** или **значение2** опускаются, принимаются значения по умолчанию, соответствующие одному их нулевому действию. Параметр **значение1** представляет процентное значение снижения теоретической скорости замедления. Применяемая скорость замедления равна $(100 - \text{значение1}) / 100$ раз теоретическая скорость замедления. Максимальное значение снижения равно 100. В этом случае результирующая скорость соответствует 1% теоретической скорости. И наоборот, когда она равна нулю или опускается, принимается значение по умолчанию, т. е. принимается полная теоретическая скорость.

Параметр **значение2** представляет собой процент снижения, от 1 до 10 раз, теоретической скорости замедления в случае инверсии движения оси. В частности, принимается, что когда **значение2** равно 100, скорость снижается в 10 раз. И наоборот, когда оно равно нулю или опускается, скорость не уменьшается.
Если заданное значение оказывается меньше нуля или больше 100, эта инструкция генерирует ошибку системы 4399: "Параметр за пределами диапазона". Важно помнить, что при опускании параметра **значение1**, должен опускаться также параметр **значение2**.

Примечание

Использование этой инструкции связано с использованием инструкций [JERKSMOOTH](#) и [SETCONTORNATURE](#) действует только в движениях с классической интерполяцией (инструкции [LINEARABS](#), [LINEARINC](#), [CIRCABS](#), [CIRCINC](#), [HELICABS](#), [HELICINC](#)).

SETVELI

Синтаксис

SETVELI ось1 [, ..., ось6] [, скорость]

Аргументы

ось1 [...,**ось6**] имя устройства типа интерполируемой оси
скорость константа float или переменная float

Описание

Задаёт **максимальную** скорость осей **ось1**, **ось2**, для интерполированных движений. Скорость выражается в единицах измерения оси, указанных параметром конфигурации. Если аргумент **скорость** опускается, принимается максимальная скорость конфигурации. Оси шагового типа могут использоваться в этой инструкции только при их управлении удалённым TRS-AH.
См. инструкцию [SETVEL](#).

SETVELILIMIT

Синтаксис

SETVELILIMIT ось, скорость

Аргументы

ось имя устройства типа оси
скорость константа float или переменная float

Описание

Задаёт отдельные компоненты **скорости указанной оси** для интерполированных движений. Скорость выражается в единицах измерения оси.

Скоординированное Движение

SETFEEDCOORD

Синтаксис

SETFEEDCOORD ось, значение1, значение2

Аргументы

ось имя устройства типа оси
значение1 константа или переменная double. Представляет собой максимальный процент мгновенного изменения feedrate
значение2 константа или переменная integer. Представляет собой число задач реального времени, к которым не применяется никакого изменения feedrate

Описание

Изменяет **значение1** в процентах максимального мгновенного изменения feed rate **оси**. Feed rate больше не меняется в течение временного интервала, выраженного в реальном времени, определенном в переменной **значение2**. Другими словами, после применения изменения feedrate override максимум на **значение1** для числа задач реального времени, равного **значение2**, не может применяться какое-либо другое изменение feedrate. Сочетание этих двух параметров определяет нечто вроде ускорения/замедления, которое может выдержать ось. При модулировании этих двух параметров можно получить функции изменения "лесенкой" с нужным уклоном.

Примечание

Для каждой оси, задействованной в координированном движении, должны задаваться значение feedrate и временной интервал, в противном случае принимаются значения по умолчанию: **значение1**=100 и **значение2**=1. При выполнении координированного движения (инструкция [COORDIN](#)) система пересчитывает параметры **значение1** и **значение2** для применения к движению в зависимости от параметров всех задействованных осей. Из контроля исключаются остановленные оси. Эти два параметра вычисляются таким образом, чтобы:

значение1: минимальное значение, заданное на перемещаемых осях

значение2: значение, полученное путем деления **значения1** на минимальное отношение **значение1/значение2**.

Пример

Function Координированное Движение

```
Setquote    X,0
Setquote    Y,0
Setquote    Z,0
```

```
setFeedCoord    X, 20, 80
setFeedCoord    Y, 10, 1
setFeedCoord    Z, 3, 3
```

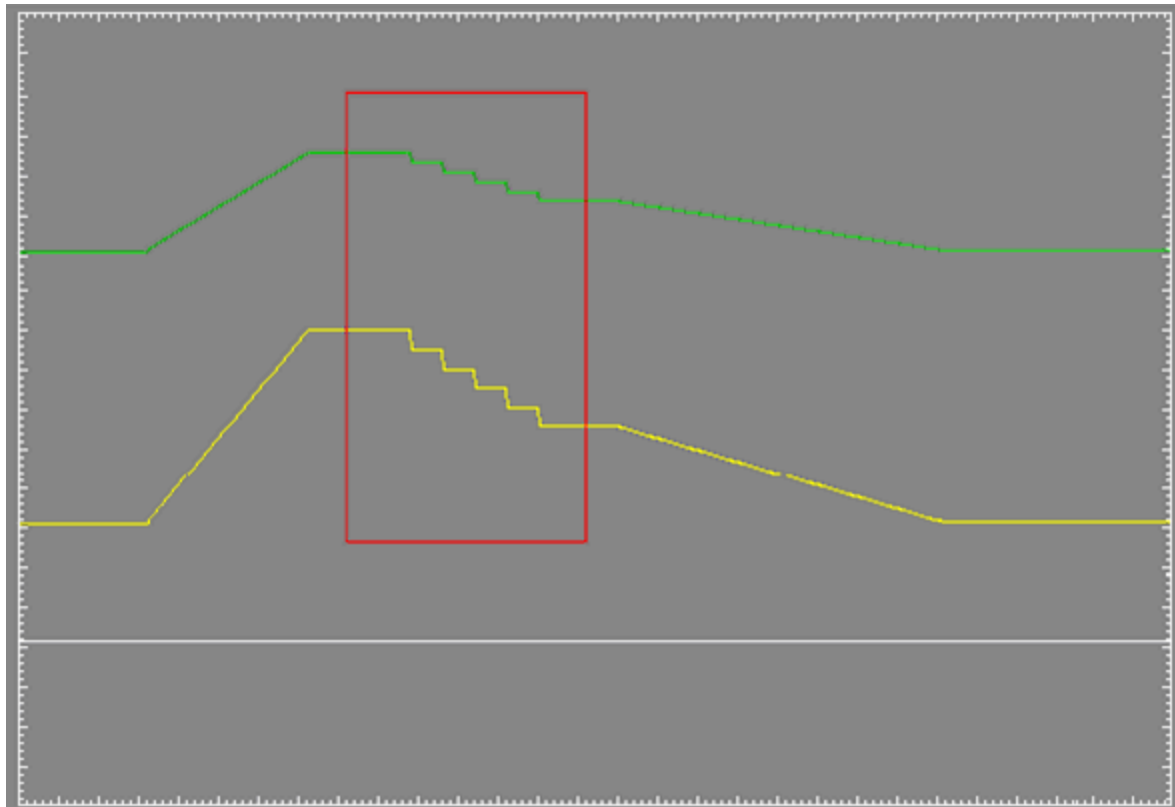
```
coordin    matrix, дельтаT, UP, строкаInit, строкаEnd,mask,_
            X,columnX, Y,columnY, Z,columnZ
waitstill  X,y,z
```

fret

Предположим, что на определенном переходе координированного движения ось Z не перемещается. Заданные параметры таковы:

```
Макс._Изменение    = 10
Дельта_T           = 10 / 0.25 = 40
```

Таким образом, мы получаем следующий график осциллографа, на котором зеленым цветом представлен профиль скорости оси X, а желтым - профиль скорости оси Y.



SETOFFSET

Синтаксис

SETOFFSET

ось, координата

Аргументы

ось
координата

имя устройства типа оси
константа или переменная. Сдвиг для координированного движения

Описание

Позволяет применять сдвиг к координатам координированного движения. Сдвиг, определенный параметром **координата**, будет использоваться в последующих координированных движениях с прибавлением указанной координаты ко всем координатам в таблице. См. также инструкцию [COORDIN](#).

Объединенное Движение

RATIO

Синтаксис

RATIO

ось, [значение]

Аргументы

ось
значение

имя устройства типа оси.
константа или переменная. Отношение сокращения.

Описание

Задаёт отношение сцепления ведомой оси относительно соответствующей главной. Движения ведомой оси будут сдвинуты относительно движений главной оси на заданное отношение сцепления. При опускании параметра **значение** отношение сбрасывается на 1.0 (идентичные движения). Эта инструкция при выполнении при оси не в состоянии ведомая и при соответствующей главной оси не в состоянии "на координате" генерирует ошибку системы. См. инструкцию [CHAIN](#).

Пример

CHAIN

X, Y

RATIO	Y, 0.5	; отношение сокращения 1/2
MOVABS WAITSTILL	X, 100 X	; ось Y переместится на координату 50

SETDYNRATIO

Синтаксис

SETDYNRATIO ось, значение

Аргументы

ось имя устройства типа оси
значение константа или переменная double

Описание

Эта инструкция позволяет динамически, при движении главной оси, менять отношение сцепления. Можно применить новое значение отношения сцепления, даже если предыдущее изменение не завершилось. Объявленная **ось** должна быть ведомой.

Если эта инструкция выполняется с главной осью в состоянии КООРДИНАТА, новое значение отношения сцепления **значение** применяется мгновенно.

Изменение отношения сцепления выполняется при помощи линейной функции ускорения (или замедления). Используемое значение ускорения определяется ускорением главной оси, которая в данный момент захвачена для движения точка-точка. Это означает, что можно также изменить эту функцию изменения заданием нового значения ускорения с использованием инструкции [SETACC](#).

Эта инструкция может генерировать следующую ошибку системы:

- "4101: Непоследовательное управление осью", если объявленная **ось** не является ведомой.

Общие Параметры

DYNLIMIT

Синтаксис

DYNLIMIT ось, состояние

Аргументы

ось имя устройства типа оси
состояние заданная константа. Допускаются следующие значения:
ON подключение динамического контроля пределов оси
OFF отключение динамического контроля пределов оси

Описание

Подключает или отключает динамический тест на превышение пределов оси. Динамический тест на превышение пределов оси отличается от статического теста на превышение пределов оси, т. к. при каждом цикле реального времени он проверяет, превышает ли ось свои пределы, исходя из ее текущей скорости и ее максимального замедления. Статический тест проверяет в каждый момент, что фактическая координата прибытия каждой оси находится в рамках заданных положительного и отрицательного пределов оси. Кроме того, перед пуском движения статический тест проверяет, превышают ли заданные пределы координаты, переданные при помощи инструкций движения. Для определения новых пределов осей до инструкции DYNLIMIT должна задаваться инструкция [SETLIMPOS](#) и инструкция [SETLIMNEG](#), в противном случае рассмотренные предельные значения, уменьшенные на минимальное пространство для остановки, равны координате 0.0 для обоих направлений движения.

Пример

Проверка пределов оси двумя типами теста, статического и динамического, с осями на одной направляющей движения.

Статический текст

В произвольном движении **Ось X1** не может быть больше изначального положительного предела, определенного координатой **Оси X2**. Проверка пределов осей генерирует ошибку системы № 4108 "Ось X1: конечная координата вышла за программные пределы".

Динамический текст

В произвольном движении проверяется, что мгновенная координата **координата X1** находится в пределах оси, с соответствующим знаком в зависимости от направления движения оси, уменьшенных на минимальное пространство остановки этой оси. Минимальное пространство остановки вычисляется в зависимости от мгновенной скорости и замедления, заданного в конфигурации для движения точка-точка. Кроме того, отключается начальный контроль превышения заданных пределов координатами, переданными инструкциями перемещения.



ENABLESTARTCONTROL

Синтаксис

ENABLESTARTCONTROL ось, [таймаут]

Аргументы

ось

таймаут

имя устройства типа оси
переменная или константа integer. Это предельное время ожидания, выраженное в режиме реального времени.

Описание

Эта инструкция позволяет подключать и задавать **таймаут** для управления невыполненным пуском или неожиданной остановкой оси.

Если при запросе движения ось не переместится по меньшей мере на 2 шага за 200 циклов реального времени, генерируется ошибка системы № 3 "Серво-ошибка".

Если параметр **таймаут** задан на ноль, управление отключается. Эта инструкция не имеет никакого действия, если теоретическая скорость меньше двух шагов в 200 циклах реального времени или если движение завершается менее, чем за 200 циклов реального времени.

Пример

```
; таймаут пуска осей равен 10 реального времени
ENABLESTARTCONTROL X, 10
```

NOTCHFILTER

Синтаксис

NOTCHFILTER ось, значение

Аргументы

ось

значение

имя устройства типа оси
константа или переменная. Значение частоты [Гц]. Допускаются значения от **0** до **500**.

Описание

Задаёт предельную частоту фильтра отметки для указанной оси. Если **значение** равно 0, фильтр отключается. Если параметр **значение** опускается, используется значение, заданное в конфигурации.

Пример

```
; предел частоты 97 Гц
NOTCHFILTER X, 97
```

RESLIMNEG

Синтаксис

ResLimNeg ось

Аргументы

ось

имя устройства типа оси

Описание

Отключает тест на отрицательном пределе указанной **оси**. Эти инструкции обычно используются в стандартных процедурах обнуления для поиска выключателей обнуления, позволяя, таким образом, осям превышать заданные значения конфигурации. См. также инструкции [SETLIMNEG](#), [SETLIMPOS](#), [RESLIMPOS](#).

Пример

[Стандартная процедура обнуления оси](#)

RESLIMPOS**Синтаксис**

RESLIMPOS **ось**

Аргументы

ось имя устройства типа оси

Описание

Отключает тест на положительном пределе указанной **оси**. Эти инструкции обычно используются в стандартных процедурах обнуления для поиска выключателей обнуления, позволяя, таким образом, осям превышать заданные значения конфигурации. См. также инструкции [RESLIMNEG](#), [SETLIMPOS](#), [SETLIMNEG](#).

Пример

[Стандартная процедура обнуления оси](#)

SETADJUST**Синтаксис**

SETADJUST **ось, состояние, [значение]**

Аргументы

ось имя устройства типа оси
состояние заданная константа. Может принимать следующие значения:
ON подключить
OFF отключить
[значение] переменная или константа float. Напряжение [Вольт]

Описание

Подключает или отключает на указанной **оси** расчет автоматической компенсации сдвига, т. е. adjust. Adjust позволяет компенсировать небольшие сдвиги положения в конце движения оси. Обычно adjust подключен.

Может оказаться удобным отключить adjust для осей, перемещаемых двигателями с повышенным гистерезисом положения, которые не получили бы какого-либо преимущества от использования этой функции управления.

При подключении adjust после отключения блок управления не принимает во внимание вычисленное ранее значение. Таким образом, эта инструкция может использоваться для обнуления накопленного осью adjust без необходимости повторной инициализации блока управления.

При наличии третьего параметра сдвиг задается на указанное **значение** независимо от подключения или отключения автоматического adjust. Такое использование этой инструкции позволит компенсировать программой возможный сдвиг эталона скорости вместо того, чтобы компенсировать его на приводе, даже если компенсация на приводе должна предпочитаться компенсации программой.

Эта инструкция применима только на осях, управляемых в аналоговом режиме.

SETBACKLASH**Синтаксис**

SETBACKLASH **ось, значение**

Аргументы

ось имя устройства типа оси
значение переменная или константа float. Значение зазора.

Описание

Эта инструкция позволяет уменьшить или устранить влияние механических зазоров на траекторию **оси**. Задаваемое **значение** зазора должно находиться в пределах от 0.0 до 3.0. Это значение не зависит от выбранных единиц измерения.

Особые ситуации наблюдаются в следующих случаях:

- при отключенной оси функция компенсации зазоров, даже если требуется, не применяется.

- с вертикальной осью, учитывая особую конфигурацию, не наблюдается какого-либо зазора.
- с осью с нагрузкой с большой инерцией может наблюдаться частичная, а иногда и полная компенсация нагрузки. Действительно, учитывая массу груза, движение осей может останавливаться с задержкой по сравнению с остановкой двигателя. Поэтому результирующее положение зубьев зубчатого колеса редуктора относительно положения зубьев зубчатого колеса двигателя может быть таким, что оно уменьшит и даже устранил зазор.
- отображение реальных координат и кодера **оси**, выборка которых была сделана осциллографом в точках, в которых была подключена компенсация зазоров (инверсия движения), показывает пик, равный значению этого зазора.

Эта инструкция генерирует ошибку системы при использовании:

- на шаговых осях, управление которыми не осуществляется удаленными модулями TRS-AX, осях подсчета и виртуальных осях.
- на шаговых осях, управление которыми осуществляется удаленными модулями TRS-AX с смоделированным кодером.

Пример

; функция с отключенной компенсацией зазоров (на чертеже показано красным цветом)

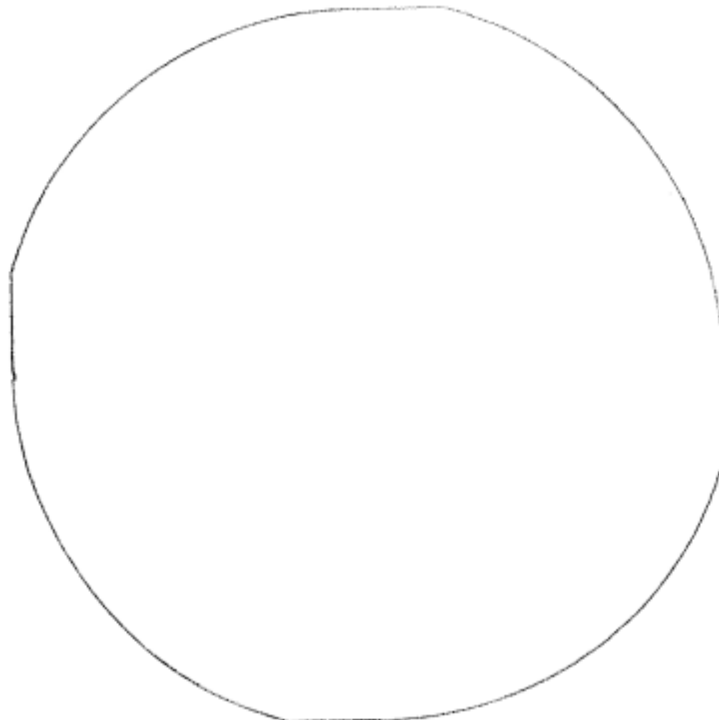
```
SETQUOTE      X, 0
SETQUOTE      Y, 0
SETVELI       X, 1.0
CIRCLE        X,Y,cw,100,90
WAITSTILL     X,Y
```

; функция с подключенной компенсацией зазоров (на чертеже показано черным цветом)

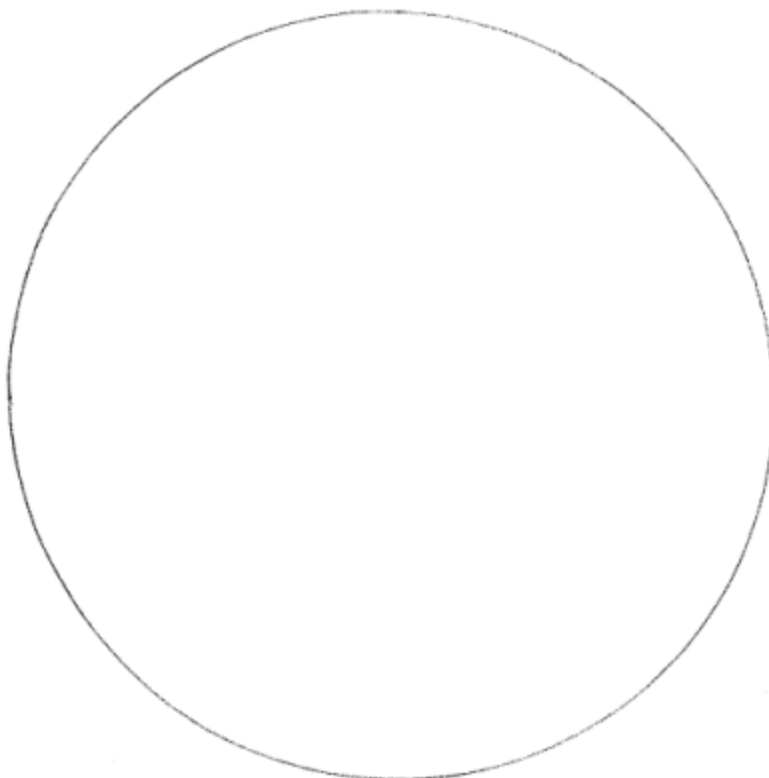
```
SETQUOTE      X, 0
SETQUOTE      Y, 0
SETVELI       X, 1.0
SETBACKLASH   X, 1.9
SETBACKLASH   Y, 1.8
CIRCLE        X,Y,cw,100,90
WAITSTILL     X,Y
```

Выполнение этих двух функций дает две различные траектории.

На первом рисунке показана интерполяция двух осей, имеющих зазор в паре двигатель-редуктор.



На втором рисунке показана та же интерполяция, но с использованием инструкции компенсации зазоров.



SETBIGWINFACTOR

Синтаксис

SETBIGWINFACTOR ось, значение

Аргументы

ось имя устройства типа оси
значение переменная или константа double. Множитель для расчета большого окна.

Описание

Эта инструкция позволяет менять множитель для расчета большого окна на выбранной **оси**. Для расчета большого окна переменная **значение** умножается на параметр, определенный в конфигурации осей Окно прибытия на координату. Задаваемое **значение** должно находиться в пределах 1 и 257 включительно. Умолчание - 4.0.

SETDEADBAND

Синтаксис

SETDEADBAND ось, **VМинПол**, **VМинОтр**, **VПорогПоз**, **VПорогОтр**

Аргументы

ось имя устройства типа оси
VМинПол переменная или константа float. Минимальное положительное напряжение [Вольт]
VМинОтр переменная или константа float. Минимальное отрицательное напряжение [Вольт]
VПорогПоз переменная или константа float. Положительный порог [Вольт]
VПорогОтр переменная или константа float. Отрицательный порог [Вольт]

Описание

Задаёт параметры минимального напряжения для указанной оси. Значения минимального напряжения (положительного/отрицательного) суммируются с теоретическим эталонным напряжением (положительным/отрицательным), если его значение превышает заданное пороговое значение (положительное/отрицательное). Если теоретическое эталонное напряжение находится в пределах пороговых значений, фактическое эталонное напряжение принудительно задается на ноль. Можно

отключить управление минимальным напряжением, задавая все значения на ноль. Пороговые значения должны быть всегда меньше или равны соответствующим значениям минимального напряжения. При инициализации системы управление минимальным напряжением отключено.

SETENCLIMIT

Синтаксис

SETENCLIMIT ось [, значение]

Аргументы

ось имя устройства типа оси
значение константа или переменная double

Описание

Меняет неправильный предел подключения кодера. Этот параметр выражается в единицах измерения оси. Допустимые значения должны находиться в интервале, эквивалентном 128 – 16384 шагам кодера. Если этот параметр опускается, восстанавливается значение по умолчанию, равное 1024 шагам.

Например, для оси с разрешением 1000 импульсов/мм допустимые значения будут находиться в пределе от 0,128 до 16,384 мм.

При задании параметра **значение** на ноль отключается контроль неправильного предела подключения кодера.

Пример

; задаю предел подключения кодера равным 3.5

SETENCLIMIT X, 3.5

SETINDEXEN

Синтаксис

SETINDEXEN ось, состояние

Аргументы

ось имя устройства типа оси
состояние заданная константа. Допускаются следующие значения:
ON состояние отметки нуля активно
OFF состояние отметки нуля отключено

Описание

Подключает или отключает на указанной **оси** обнуление координаты у отметки нуля. Для выполнения этой инструкции ось должна быть типа подсчета.

SETINTEGTIME

Синтаксис

SETINTEGTIME ось [, значение]

Аргументы

ось имя устройства типа оси
значение константа integer или переменная

Описание

Задаёт число образцов погрешности контура, использованных для расчета интегральной составляющей. Действительные значения находятся в пределах от 1 до 200.. Изменение этого параметра налёту возможно, но может создать ступеньки на эталоне скорости оси, поэтому рекомендуется воздействовать на этот параметр при остановленных и отключенных осях и, по возможности, - в свободном состоянии.

SETIRMP

Синтаксис

SETIRMP ось, скорость

Аргументы

ось имя устройства типа оси
скорость константа float или переменная float. Скорость начала изменения

Описание

Задаёт **оси** значение **скорости начала изменения**. Это минимальная скорость шагового двигателя. Эта инструкция используется для осей, перемещаемых шаговыми двигателями.

SETLIMNEG**Синтаксис**

SETLIMNEG ось[, координата]

Аргументы

ось имя устройства типа оси
координата константа или переменная. Отрицательный предел

Описание

Задаёт **оси** предельную отрицательную **координату**.
Если параметр **координата** опускается, подключается отрицательный предел конфигурации. Эти инструкции обычно используются в стандартных процедурах обнуления для поиска выключателей обнуления, позволяя, таким образом, осям превышать заданные значения конфигурации. См. также инструкции [RESLIMNEG](#), [SETLIMPOS](#), [RESLIMPOS](#).

Пример

[Стандартная процедура обнуления оси](#)

SETLIMPOS**Синтаксис**

SETLIMPOS ось [, координата]

Аргументы

ось имя устройства типа оси
координата константа или переменная. Положительный предел

Описание

Задаёт **оси** предельную положительную **координату**.
Если параметр **координата** опускается, подключается положительный предел конфигурации. Эти инструкции обычно используются в стандартных процедурах обнуления для поиска выключателей обнуления, позволяя, таким образом, осям превышать заданные значения конфигурации. См. также инструкции [RESLIMNEG](#), [RESLIMPOS](#), [SETLIMNEG](#).

Пример

[Стандартная процедура обнуления оси](#)

SETMAXER**Синтаксис**

SETMAXER ось, значение [, направление]

Аргументы

ось имя устройства типа оси
значение константа или переменная. Максимальная погрешность преследования
направление заданная константа. Направление оси. Может принимать следующие значения: **POSITIVE**, **NEGATIVE**

Описание

Задаёт **оси**:

- максимальное **значение** преследования, допустимое при статическом тесте сервоошибки
- **значение**, которое при прибавлении к теоретической погрешности, пропорциональной скорости, определяет максимальное значение преследования, допустимое при динамическом тесте сервоошибки.

Если **направление** опускается, максимальное значение преследования задается для обоих направлений.

SETMAXERNEG**Синтаксис**

SETMAXERNEG ось, задержка, опережение

Аргументы

ось имя устройства типа оси

**задержка
опережение**

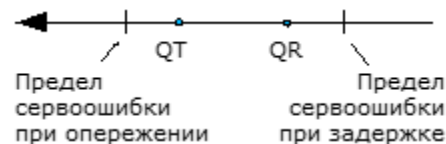
константа или переменная. Максимальная погрешность преследования
константа или переменная. Максимальная погрешность опережения

Описание

Задаёт **оси** максимальные значения погрешности преследования (**задержка**) и опережения (**опережение**), допустимые блоком управления, только для отрицательного направления перед генерированием сигнала "сервоошибка".

- **задержка:** это максимальное значение преследования, допустимое при статическом тесте сервоошибки или, при динамическом тесте сервоошибки, - это значение, которое при прибавлении к теоретической погрешности, пропорциональной скорости, определяет максимальное допустимое значение преследования.
- **опережение:** это максимальное значение преследования, допустимое при инверсии движения с отрицательного на положительное

Погрешность преследования - это разница между теоретической координатой (в которой должна находиться ось) и реальной координатой. При движении оси в отрицательном направлении погрешность преследования с отрицательным знаком указывает условие задержки оси, погрешность преследования с положительным знаком указывает условие опережения. Если эта инструкция не используется, блок управления использует максимальные значения погрешности преследования, имеющиеся в конфигурации оси. В этом случае предел опережения равен 1/4 пределу задержки.

**Пример**

SETMAXERNEG Оси.X, 10, 5

;Максимальная задержка оси составляет 10 мм , максимальное опережение - 5 мм

SETMAXERPOS**Синтаксис**

SETMAXERPOS ось, задержка, опережение

Аргументы

ось имя устройства типа оси
задержка константа или переменная. Максимальная погрешность преследования
опережение константа или переменная. Максимальная погрешность опережения

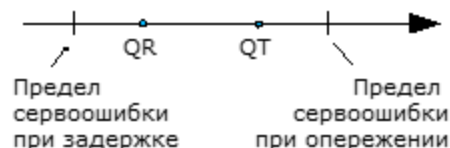
Описание

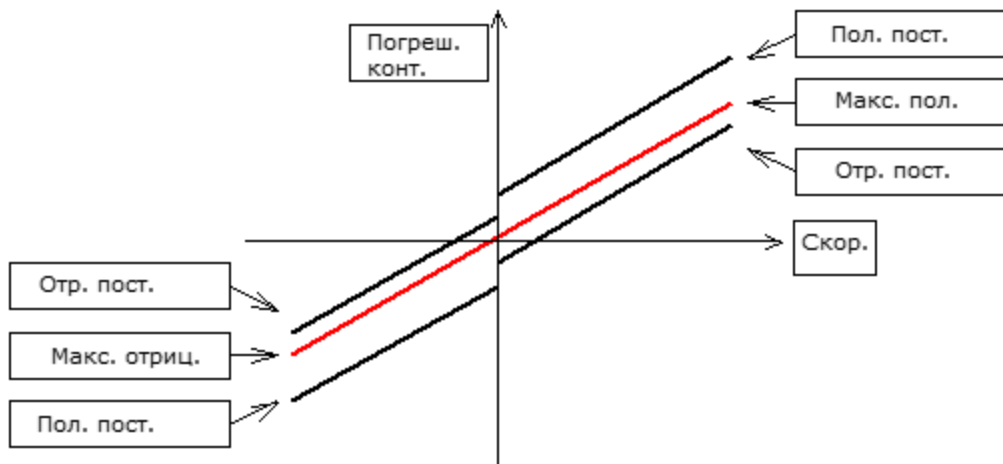
Задаёт **оси** максимальные значения погрешности преследования (**задержка**) и опережения (**опережение**), допустимые блоком управления, только для положительного направления перед генерированием сигнала "сервоошибка".

- **задержка:** это максимальное значение преследования, допустимое при статическом тесте сервоошибки или, при динамическом тесте сервоошибки, - это значение, которое при прибавлении к теоретической погрешности, пропорциональной скорости, определяет максимальное допустимое значение преследования.
- **опережение:** это максимальное значение преследования, допустимое при инверсии движения с отрицательного на положительное

Погрешность преследования - это разница между теоретической координатой (в которой должна находиться ось) и реальной координатой. При движении оси в положительном направлении погрешность преследования с положительным знаком указывает условие задержки оси, погрешность преследования с отрицательным знаком указывает условие опережения.

Если эта инструкция не используется, блок управления использует максимальные значения погрешности преследования, имеющиеся в конфигурации оси. В этом случае предел опережения равен 1/4 пределу задержки.





SETPHASESINV

Синтаксис

SETPHASESINV

ось, состояние

Аргументы

ось
состояние

имя устройства типа оси
заданная константа. Допускаются следующие значения:

ON состояние инверсии фаз активно

OFF состояние инверсии фаз не активно

Описание

Подключает или отключает инверсию фаз на указанной **оси**. Позволяет компенсировать при помощи программного обеспечения возможную инверсию электромонтажа фаз кодера. При использовании вместе с инверсией эталона позволяет инвертировать направление оси (если электромонтаж выполнен правильно).

Для выполнения этой инструкции ось должна быть в состоянии FREE.

SETREFINV

Синтаксис

SETREFINV

ось, состояние

Аргументы

ось
состояние

имя устройства типа оси
заданная константа. Допускаются следующие значения:

ON подключает инверсию эталона скорости

OFF отключает инверсию эталона скорости

Описание

Подключает или отключает инверсию эталона скорости на указанной оси. При использовании вместе с инверсией фаз позволяет инвертировать направление оси (если электромонтаж выполнен правильно).

Для выполнения этой инструкции ось должна быть в состоянии FREE.

См. также [SETPHASESINV](#).

SETRESOLUTION

Синтаксис

SETRESOLUTION

ось [, значение]

Аргументы

ось
значение

имя устройства типа оси
константа или переменная double

Описание

Меняет разрешение указанной оси. Если **значение** опускается, используется значение разрешения, заданное в конфигурации. Изменение значения разрешения может выполняться только при остановленной оси (состояние оси = координата), в противном случае генерируется ошибка системы № 4101 "Непоследовательное управление осью".

Если значение не было найдено, **переменная** будет содержать значение -1.

FINDB

Синтаксис

FINDB **матрица, столбец, мин_предел, макс_предел, значение, переменная**
FINDB **вектор, мин_предел, макс_предел, значение, переменная**

Аргументы

матрица имя матрицы. Матрица, в которой выполнять поиск
вектор имя вектора. Вектор, на котором выполнять поиск
столбец целая константа или переменная или имясчетчика. Номер столбца матрицы, в которой выполнять поиск
мин_предел константа или переменная. Минимальный индекс вектора или матрицы, с которого начинать поиск
макс_предел константа или переменная. Максимальный индекс вектора или матрицы, где завершать поиск
значение константа или переменная. Искомое значение
переменная переменная. Результат поиска

Описание

Выполняет быстрый поиск значения внутри **вектора** или **столбца матрицы** и помещает индекс элемента в **переменную**. Для того, чтобы поиск был успешным, необходимо, чтобы **вектор** или **столбец матрицы** были предварительно упорядочены инструкцией SORT по возрастающей. Если значение не было найдено, **переменная** будет содержать значение -1.

LASTELEM

Синтаксис

LASTELEM **вектор, элементывек**
LASTELEM **матрица, строкикат**

Аргументы

матрица имя матрицы
вектор имя вектора
элементывек переменная. Число элементов вектора
строкикат переменная. Число строк матрицы

Описание

Записывает в переменную **элементывек** число элементов **вектора** или в переменную **строкикат** - число строк **матрицы**.

LOCAL

Синтаксис

LOCAL **имяпер AS тип**
LOCAL **вектор[число элементов] AS тип**
LOCAL **матрица[числострок] AS тип, тип, тип и т. д.**
LOCAL **матрица[число строк] AS тип:имястолб1, тип:имястолб2, тип:имястолб3 и т.д.**

Аргументы

имяпер имя переменной
[число элементов] переменная или константа (обязательный аргумент). Число элементов вектора
[число строк] константа или переменная (обязательный аргумент). Число строк матрицы
тип char, integer (32 бит), float (32 бит), double (64 бит), строка, таймер
имястолб1...имястолбN имя столбца. Ярлык

Описание

Объявление локальной переменной. Перед этой инструкцией может появиться только инструкция PARAM, определяющая параметры функции. Более подробная информация о локальных переменных приводится в [Локальные переменные](#).

MOVEMAT

Синтаксис

MOVEMAT	имяисходматрицы, имяцелевойматрицы
MOVEMAT	имяисходматрицы[исходстрока], имяцелевойматрицы[целеваястрока]
MOVEMAT	имяисходматрицы[исходстрока], имяцелевойматрицы[целеваястрока], числострок

Аргументы

имяисходматрицы	имя исходной матрицы
исходстрока	номер исходной строки для копирования исходной матрицы (обязательный аргумент)
имяцелевойматрицы	имя целевой матрицы
целеваястрока	номер исходной строки для копирования в целевую матрицу (обязательный аргумент)
числострок	число копируемых строк

Описание

Копирует содержимое целой матрицы **имяисходматрицы** в другую матрицу **имяцелевойматрицы** или одну или несколько строк **числострок** строки матрицы **имяисходматрицы[исходстрока]** в строку матрицы **имяцелевойматрицы[целеваястрока]**. Если параметр **числострок** не указан, копируется только одна строка. Две матрицы должны иметь структуру одного типа (число столбцов и тип данных каждого столбца), а в случае целых матриц - также одинаковое число строк. В одной матрице можно перемещать строки данных.

Пример

```

Movemat Mx1, Mx2 ; копирует матрицу Mx1 в Mx2

; копирует строку 10 матрицы Mx1 в строку 3 Mx2
Movemat Mx1[10], Mx2[3]

; копирует строку 1 матрицы Mx1 в строку 7 Mx1
Movemat Mx1[1], Mx1[7]
; копирует 6 строк начиная со строки 2 матрицы Mx1 в матрицу
; Mx2 начиная со строки 8
Movemat Mx1[2], Mx2[8], 6

; копирует 4 строки начиная со строки 2
; матрицы Mx1 в эту же матрицу
; Mx1 начиная со строки 10
Movemat Mx1[2], Mx1[10], 4

```

PARAM

Синтаксис

[PARAM]	имяпер AS тип
[PARAM]	вектор[число элементов] AS тип
[PARAM]	матрица[число строк] AS тип, тип, тип и т. д.
[PARAM]	матрица[число строк] AS тип:псевдоимя, тип:псевдоимя, тип:псевдоимя и т. д.

Аргументы

имяпер	имя переменной
[число элементов]	константа (обязательный аргумент)
[число строк]	константа (обязательный аргумент)
тип	char, integer (32 бит), float (32 бит), double (64 бит), строка

Описание

Параметры ведут себя как локальные переменные (см. [LOCAL](#)), но инициализируются тем, кто вызывает функцию. Синтаксис для объявления параметров - такой же, который используется для локальных переменных. Параметры передаются по значению или по ссылке в зависимости от их типа. См. "[Функции](#)".

Они должны объявляться до любой другой инструкции.

Более подробно см. [Локальные переменные](#).

SETVAL

Синтаксис
SETVAL

значение, имяпер

Аргументы
значение
имяпер

константа или переменная или имяустройства
переменная или имяустройства

Описание

Задаёт указанное **значение** переменной **имяпер** или *i*-му элементу вектора или матрицы.

SORT

Синтаксис
SORT
SORT

матрица, столбец [, порядок], мин_предел, макс_предел
вектор [,порядок], мин_предел, макс_предел

Аргументы

матрица	имя матрицы
вектор	имя вектора
столбец	константа или переменная. Номер столбца матрицы
порядок	заданная константа. Указывает режим упорядочения Допускаются следующие значения: UP упорядочение по возрастающей DOWN упорядочение по убывающей
мин_предел	константа или переменная. Минимальный индекс вектора или матрицы, с которого начинать упорядочение
макс_предел	константа или переменная. Максимальный индекс вектора или матрицы, где завершать упорядочение

Описание

Выполняет упорядочение значений, имеющих в **векторе** или **матрице**, с направлением, определённым константой **порядок**.

В случае матрицы упорядочение строк определяется расположением по возвращающей (UP) или убывающей (DOWN) значений, содержащихся в выбранном **столбце**.

Если аргумент **порядок** опускается, автоматически принимается режим UP.



10.3.8 Строки

ADDSTRING

Синтаксис
ADDSTRING

имястроки1, имястроки2, имястроки3

Аргументы
имястроки1
имястроки2
имястроки3

строковая константа или строковая переменная. Исходная строка
строковая константа или строковая переменная. Добавляемая строка
строковая переменная. Строка результата

MID

Синтаксис
MID

имяисходнойстроки, первыйзн [, числознаков], имяцелевойстроки

Аргументы

имяисходнойстроки
числознаков
имяцелевойстроки
первыйзн

строковая константа или строковая переменная. Исходная строка константа или переменная. Число копируемых знаков строковая переменная. Целевая строка константа или переменная. Положение знака отправки копии

Описание

Извлекает из строки, определенной **имяисходнойстроки**, число знаков, определенных **числознаков**, начиная с положения **первыйзн**.

Извлеченная подстрока помещается в строку, обозначенную **имяцелевойстроки**.

Если **числознаков** опускается, копируется **исходнаястрока** начиная с положения **первыйзн** до ее конца. Практически, берется центральная часть исходной строки. См. также инструкции [LEFT](#) и [RIGHT](#).

Пример

См. пример [Операции на строках](#)

RIGHT

Синтаксис
RIGHT

имяисходнойстроки, числознаков, имяцелевойстроки

Аргументы

имяисходнойстроки
числознаков
имяцелевойстроки

строковая константа или строковая переменная. Исходная строка константа или переменная. Число копируемых знаков строковая переменная. Целевая строка

Описание

Копирует последние **числознаков** строки **имяисходнойстроки** в строку **имяцелевойстроки**.

Практически, берется правая часть исходной строки. См. также инструкции [LEFT](#) и [MID](#)

Пример

См. [Операции на строках](#)

SEARCH

Синтаксис
SEARCH

имястроки, символ, переменная

Аргументы

имястроки
символ

строковая переменная.
константа шаг или строковая константа или строковая переменная.
Искомый символ или строка
переменная

переменная

Описание

Выполняет поиск положения символа ASCII, обозначенного **символ** (который может быть также строкой) внутри строки **имястроки** и помещает индекс результата в **переменную**.

Если **символ** не был найден, **переменная** будет содержать значение -1.

Пример

См. [Операции на строках](#)

SETSTRING

Синтаксис
SETSTRING

"значение", имястроки

Аргументы

значение

строковая константа или строковая переменная (заклученная в двойные кавычки)

имястроки

целевая строка

Описание

Копия строки.
 Выполняет копирование символов ASCII, присутствующих в строке, определенной "**значение**", в строку, определенную **имястроки**.
 Для ввода непечатаемых символов в строку см. инструкцию [CONTROLCHAR](#).

Пример

См. [Операции на строках](#)

STR**Синтаксис**

STR **значение, имястроки**

Аргументы

значение константа или переменная. Преобразуемое исходное значение
имястроки строковая переменная. Строка-получатель

Описание

Преобразует в символы ASCII **значение** и помещает результат в строку **имястроки**. Может использоваться для преобразования переменной типа integer в строку. Например, число 10 преобразуется в строку "10".

Пример

См. [Операции на строках](#)

VAL**Синтаксис**

VAL **имястроки, результат**

Аргументы

имястроки строковая переменная. Преобразуемая строка
результат переменная. Преобразованная строка

Описание

Преобразует содержимое строки **имястроки** в десятичное число и помещает результат в **переменную**.
 Например, строка "123" преобразуется в значение 123.

Пример

См. [Операции на строках](#)

10.3.9 Связь**CLEARRECEIVE****Синтаксис**

CLEARRECEIVE

Аргументы

никакого аргумента

Описание

Опорожняет список RECEIVE выполненных, но еще не удовлетворенных.

COMCLEARRXBUFFER**Синтаксис**

COMCLEARRXBUFFER **номерCOM**

Аргументы

номерCOM заданная константа. Номер последовательного порта. Могут задаваться следующие значения: от **COM1** до **COM8**

Описание

Эта инструкция опорожняет буфер приема последовательного порта **номерCOM**. Все имеющиеся данные стираются.

COMCLOSE

Синтаксис
COMCLOSE

номерCOM

Аргументы
номерCOM

заданная константа. Номер последовательного порта. Могут задаваться следующие значения: от **COM1** до **COM8**

Описание

Закрывает последовательную линию **НомерCOM**, открывшую **COMOPEN**. Необходимо закрывать последовательную линию также когда задача, открывшая последовательный порт, завершается по любым причинам.

COMGETERROR

Синтаксис
COMGETERROR

номерCOM, переменная

Аргументы
номерCOM

заданная константа. Номер последовательного порта. Могут задаваться следующие значения: от **COM1** до **COM8**

переменная

переменная integer. Результат последней операции, выполненной на последовательном порте

Описание

Инструкция считывает код возврата последней инструкции последовательной связи, вызванной на порте **номерCOM**. При помощи этой инструкции можно знать, успешно ли была выполнена операция считывания или записи, в противном случае возвращается также код ошибки. Далее перечисляются коды ошибки:

Нормальный возврат	0
Полон буфер передачи	2
Устройство уже открыто	3
Порт недействителен или неконфигурирован	6
Не удалось подключение порта входов/выходов	7
Невозможно подключиться к прерыванию	8
Последовательный порт (com) еще не открыт	9
Последовательное устройство (com) занято	12
Невозможна связь с RTX	14

COMGETRXCOUNT

Синтаксис
COMGETRXCOUNT

номерCOM, числознаков

Аргументы
номерCOM

заданная константа. Номер последовательного порта. Могут задаваться следующие значения: от **COM1** до **COM8**

числознаков

число знаков, присутствующих в буфере

Описание

Эта инструкция возвращает число знаков, имеющихся в буфере приема. Она позволяет знать, были ли получены знаки с последовательного порта.

COMOPEN

Синтаксис
COMOPEN

номерCOM, baudrate, wordsize, stopbits, parity

Аргументы
номерCOM

заданная константа. Номер последовательного порта. Могут задаваться следующие значения: от **COM1** до **COM8**

baudrate

скорость связи в бодах. Могут задаваться следующие значения: 2400, 4800, 9600, 19200, 38400, 57600, 115200

wordsize

размер слова данных. Могут задаваться следующие значения: 5,6,7,8

stopbits

стоповый бит. Могут задаваться следующие значения: 1, 2

parity

заданная константа. Паритет. Могут задаваться следующие значения: **NOPARITY**, **ODDPARITY** и **EVENPARITY**

Описание

Открывает последовательную линию. Должна выполняться до любой другой инструкции, управляющей последовательной линией. Если до COMOPEN выполняется любая другая инструкция, относящаяся к этой последовательной линии, генерируется ошибка системы. Передаваемые параметры должны быть значениями из указанных выше.

Канал связи с последовательной линией связан с задачей, выполнившей инструкцию COMOPEN. Если задача завершается, канал связи закрывается автоматически.

См. также [COMCLOSE](#), [COMREAD](#), [COMWRITE](#), [COMREADSTRING](#), [COMWRITESTRING](#).

Примечание

Число доступных последовательных линий зависит от аппаратной среды ЧПУ (см. документацию). В среде RTX доступны только COM1 и COM2.

COMREAD**Синтаксис****COMREAD****номерCOM, буфер, числознаковдлясчитывания, числосчитанныхзнаков [,таймаут]****Аргументы****номерCOM**заданная константа. Номер последовательного порта. Могут задаваться следующие значения: от **COM1** до **COM8****буфер**

вектор char. Это вектор, на котором записываются считанные знаки

числознаковдлясчитывания

число знаков, которые необходимо считать с последовательной линии

числосчитанныхзнаков

число фактически считанных знаков

таймаут

таймаут ожидания (в секундах)

Описание

Инструкция считывает знаки с последовательного порта **НомерCOM**. Считанные знаки записываются в переменную **буфер**. Поле **числознаковдлясчитывания** указывает число знаков, которые должна считать инструкция. Если в буфере приема последовательной линии оказывается меньше знаков и параметр **таймаут** не определен, инструкция завершится немедленно с указанием в параметре **числосчитанныхзнаков** число фактически считанных знаков. Если параметр **таймаут** указан, инструкция должна будет ожидать максимум количество секунд, указанное в переменной, в ожидании прибытия других знаков. При истечении **таймаута** инструкция выходит, все также указывая в параметре **числосчитанныхзнаков** число знаков, фактически скопированное в **буфер**.

COMREADSTRING**Синтаксис****COMREADSTRING****номерCOM, буфер, числосчитанныхзнаков [,терминатор [,таймаут]]****Аргументы****номерCOM**заданная константа. Номер последовательного порта. Могут задаваться следующие значения: от **COM1** до **COM8****буфер**

вектор char. Это вектор, на котором располагаются данные.

числосчитанныхзнаков

число фактически считанных знаков

терминатор

знак завершения передачи

таймаут

таймаут ожидания (в секундах)

Описание

Инструкция считывает знаки с последовательного порта **НомерCOM**. Отличается от [COMREAD](#) тем, что считывает с последовательной линии до тех пор, пока не натолкнется на символ-терминатор. Считанные знаки записываются в переменную **буфер**. Эта переменная должна быть типа вектора char. Поле **числосчитанныхзнаков** указывает число знаков, которые инструкция фактически считала с последовательной линии и, следовательно, скопировала в **буфер**. Параметр **терминатор** указывает символ, который будет служить терминатором передачи. Практически, инструкция должна будет считывать символы с последовательного порта до тех пор, пока не встретит такой же символ, как указанный в этом параметре. Этот параметр является дополнительным. Если его нет, считается, что символом-терминатором является ноль. Знак нуля не будет копироваться в буфер для параметра, а при наличии в инструкции символа-терминатора он будет копироваться в буфер. **Таймаут** - это еще один параметр, который указывает, сколько секунд инструкция должна дожидаться прибытия новых знаков после того, как она опорожнила буфер приема, не найдя символ-терминатор. Если параметр **таймаут** не указан, инструкция завершится сразу после опорожнения буфера приема.

Параметр **флажки** позволяет определять, как запрашиваемая информация должна использоваться Albatros. Допустимые значения и их следствия таковы:

значени	команда	описание
e		
\$0008H	CancelAfter	Информация будет удалена после считывания
\$0800H	UpdateFlags	Меняет состояние информации (считанная/ для считывания) без изменения данных
\$8000H	Delete	Удаляет информацию

Наконец, параметр **контейнер** - это переменная (или устройство), в которое будет записываться запрашиваемая информация. Этот параметр может опускаться, в этом случае запрашивается извещение события (может использоваться для синхронизации выполнения кода GPL на различных модулях).

Список **источников**, которыми управляет Albatros, и соответствующих команд:

"@List"

Позволяет управлять командами Моделирование и Уставка.

Допускаются следующие команды (параметр **идентификатор**):

- Sim,0,контейнер: запрашивает состояние кнопки Моделирование, которое записывается на флажковый выключатель Моделирование. Возвращаемая переменная **контейнер**, если не было выявлено ошибок, имеет значение 1, в противном случае она имеет значение 0.
- Setp,0,контейнер: запрашивает состояние кнопки Уставка, которое записывается на флажковый выключатель CmdSetP. Возвращаемая переменная **контейнер**, если не было выявлено ошибок, имеет значение 1, в противном случае она имеет значение 0.
- Esc,0,контейнер: запрашивает состояние кнопки Уставка, которое записывается на флажковый выключатель Исключен. Возвращаемая переменная **контейнер**, если не было выявлено ошибок, имеет значение 1, в противном случае она имеет значение 0.

"@Environ"

Позволяет получать информацию о состоянии системы: [уровень доступа](#) пользователя, модули, подключенные к супервизору, и т. д. Запрашиваемая информация записывается в параметр **контейнер**. Значения, допустимые для параметра **идентификатор**, и соответствующие ответы следующие:

- "AccessLevel" уровень доступа к системе 0=пользователь, 1=сервис, 2=изготовитель, 3=тра
- "MaskConfModules" маска сконфигурированных модулей
- "MaskActiveModules" маска подключенных модулей
- "CurrentModule" модуль, от которого поступает запрос
- "mod:NamePC" имя ПК, соответствующего модулю "mod". (mod находится в пределах от 0 до 15)
- "LocalDateTime":

Параметр **контейнер** получит дату и время ПК в формате, связанном с его типом:

- char: номер дня недели
- integer: число секунд с 1/1/1970
- float: число дней и долей дня с 1/1/1900
- double: число дней и долей дня с 1/1/1900
- string: текст "AAAA/MM/GG hh:mm:ss"

Маски подключенных и сконфигурированных модулей являются битовыми. Младший бит соответствует модулю 0. Если модуль подключен или сконфигурирован, бит каждого модуля будет 1. В случае "NamePC" номер модуля не обязателен, при его опускании принимается номер модуля, от которого поступает запрос.

"@Syn"

Связь между GPL и программой отображения обзорных экранов. Позволяет открывать и закрывать обзорные экраны, управление которыми осуществляется GPL, и запрашивать информацию из ячейки обзорного экрана. Допускаются следующие команды (параметр **идентификатор**):

- "Open:имяфайла" открытие обзорного экрана *имяфайла.xsyn*
- "Close:имяфайла" закрытие обзорного экрана *имяфайла.xsyn*
- "имячейки" ячейка, в которой считывается запрошенная информация

Можно иметь информацию об окне перемещения осей по требованиям, определенным также для приведенного ниже параметра **источник** ["@Devices"](#).

"@FileName"

Записывает в память ассоциацию между строковой константой и именем файла, составляемым строковыми переменными. При получении сообщения об ассоциации Albatros она заменяет все последующие имена файла именем, полученным этой инструкцией. Параметр **идентификатор** - это имя файла. Имя файла - это строковая переменная. Если в параметре идентификатора не

указан полный маршрут для сохранения файла, Albatros принимает маршрут, определенный в `tra.ini` в разделе `[tra]` в пункте `dirreport`. Значение параметра идентификатора записывается в `tra.ini` в разделе `[GPLFileName]` в пункте `Log` с тем, чтобы его можно было повторно использовать при последующих выполнениях Albatros. Для удаления ассоциации необходимо задать в качестве параметра идентификатора пустую строку. Определенная таким образом ассоциация действительна для всех модулей.

"@FileDelete"

Удаление файла. Параметр **идентификатор** - это имя файла, который будет удален (полный маршрут). Если в параметре идентификатора не указан полный маршрут для сохранения файла, Albatros принимает маршрут, определенный в `tra.ini` в разделе `[tra]` в пункте `dirreport`. Имя файла может определяться по правилам, описанным для параметра **источник** [@FileRead](#). Параметр **контейнер** будет содержать значение:

- 1, если файл был удален
- 0 в противном случае

"@FileRead"

Считывает содержание файла. Параметр **идентификатор** - это имя файла, который будет считан (полный маршрут). Если в параметре идентификатора не указан полный маршрут для сохранения файла, Albatros принимает маршрут, определенный в `tra.ini` в разделе `[tra]` в пункте `dirreport`. Если идентификатор начинается и заканчивается символом `%`, поиск строки внутри него выполняется в `tra.ini` в разделе `[tra]` и используется в качестве имени файла. В имя могут вводиться символы, которые будут заменяться при выполнении инструкции:

- `%n` номер модуля, который выполняет инструкцию `RECEIVE`
- `%h` текущий час (формат 00-23)
- `%d` текущий день (формат 01-31)
- `%m` текущий месяц (формат 01-12)
- `%y` текущий год (формат из 4 цифр)

Если параметр **контейнер** определен как переменная `char`, он будет содержать один байт, считанный из файла, если он определен как строка, он будет содержать целую строку текстового файла, если же он определен как `integer`, он будет содержать число байтов, не достигающих до конца файла (0 = конец файла).

Для позиционирования указателя файла в начало файла параметр **контейнер** должен опускаться.

"@FileExist"

Проверяет существование файла. Параметр **идентификатор** - это имя файла, который будет считан (полный маршрут). Если в параметре идентификатора не указан полный маршрут для сохранения файла, Albatros принимает маршрут, определенный в `tra.ini` в разделе `[tra]` в пункте `dirreport`. Имя файла может определяться по правилам, описанным для параметра **источник** [@FileRead](#). Параметр **контейнер** будет содержать значение:

- отлично от 0, если файл существует
- 0, если файл не существует

"@FileLastWrite"

Получает дату последнего изменения, выполненного на файле. Параметр **идентификатор** - это имя файла (полный маршрут). Если в параметре идентификатора не указан полный маршрут для сохранения файла, Albatros принимает маршрут, определенный в `tra.ini` в разделе `[tra]` в пункте `dirreport`. Имя файла может определяться по правилам, описанным для параметра **источник** [@FileRead](#). Параметр **контейнер** будет содержать дату последнего изменения файла в формате, связанном с типом параметра:

- `char`: номер дня недели
- `integer`: число секунд с 1 января 1970
- `float`: число дней и долей дня с 1 января 1900
- `double`: число дней и долей дня с 1 января 1900
- `string`: текст в формате "AAAA/MM/GG hh:mm:ss"

"@FileInfo"

Читает некоторую информацию из файла. Параметр **идентификатор** должен быть выражен в форме «свойство: имя файла», где **свойство** указывает имя свойства для чтения, а **имя файла** - это имя файла. Имя файла может быть установлено через «Имя». Параметр **контейнера** будет содержать данные, прочитанные из файла.

Список свойств:

- `"version:"`: возвращает данные типа `integer` в контейнере. Четыре числа, которые идентифицируют версию, находятся в 4 байтах переменной контейнера. Если возникает ошибка, значение переменной контейнера равно 0.
- `"size:"`: возвращает данные типа `integer` или `float` или `double` в контейнере. Данные - это размер файла. Если возникает ошибка, значение переменной контейнера равно -1.

"@Devices"

Запрос открытия или закрытия окна Диагностика применительно к направляющему модулю. Параметр идентификатор может принимать следующие значения:

- "Open" открытие Диагностики
- "Close" закрытие Диагностики

Параметр **идентификатор**, когда необходимо взаимодействовать с окном перемещения оси, может принимать следующие значения:

"MoveAX#имя_оси#HasFocus"	параметр контейнер будет содержать 1, если активно указанное окно перемещения оси, в противном случае будет содержать 0.
"MoveAX#имя_оси#Jog"	параметр контейнер будет содержать 1, если задано движение для смещений, управляемых runtime оператором, в противном случае будет содержать 0.
"MoveAX#имя_оси#Step"	параметр контейнер будет содержать 1, если задано движение для смещений на заданный шаг, в противном случае будет содержать 0.
"MoveAX#имя_оси#Absolute"	параметр контейнер будет содержать 1, если задано движение со смещением на заданную координату, в противном случае будет содержать 0.

где имя_оси является именем оси, отображенным в окне. Например, если необходимо проверить, активно ли окно перемещения оси X, параметр **идентификатор** будет "[@MoveAX#X#HasFocus](#)". Имя оси может быть в одной из следующих форм:

1. Имя_Группы.Имя_подгруппы.Имя_Оси или Имя_Группы.Имя_Оси: предоставляется полный маршрут оси.
2. Имя_Оси: для идентификации правильной оси по порядку выполняются следующие проверки:
 - если задача, от которой поступает команда, является функцией подгруппы, поиск оси осуществляется в этой подгруппе.
 - если задача, от которой поступает команда, является функцией главной подгруппы, поиск оси осуществляется во всей группе. При наличии более одной оси с этим именем поиск завершается неудачно.
 - если предыдущие проверки завершаются неудачно, поиск оси осуществляется во всех группах модуля. При наличии более одной оси с именем Имя_Оси поиск не имеет положительного исхода.

"@Vars"

Запрашивает обновление глобальной переменной GPL. Позволяет выполнять обновление данных технологических параметров и инструментов. Данные параметров обычно направляются на GPL при инициализации станка. Параметр **идентификатор** принимает значение имени глобальной переменной (станка или группы), для которой требуется обновление. Параметр **контейнер** будет содержать значение:

- 1, если переменная была обновлена правильно
- 0 в противном случае

"@Application"

Взаимодействие с Albatros. Позволяет отображать на экране "окно сообщений" и закрывать Albatros. Для параметра **идентификатор** допускаются следующие значения:

- "Quit" закрывает Albatros
- "IsLocked" Проверяет, не заблокирован ли выход из Albatros. Параметр **контейнер** будет содержать 1, если интерфейс заблокирован, 0, если можно выйти из Albatros.
- "MsgBox" считывает ответ окна сообщений, открытого ранее при помощи SEND

Параметр **контейнер** в случае окна сообщений позволяет знать, какая кнопка была нажата оператором:

- 1 кнопка "ОК"
- 2 кнопка "Отмена"
- 4 кнопка "Повторить"
- 6 кнопка "Да"
- 7 кнопка "Нет"

В случае команды "Бросить" параметр **контейнер** будет содержать значение:

- 1, если Albatros был закрыт правильно
- 0 в противном случае

"@Param"

Позволяет знать номер по порядку регистрации файлов параметров Partec.xpar и Partool.xpar. Запрашиваемая информация записывается в параметр **контейнер**. Для параметра **идентификатор** допускаются следующие значения:

- "partec" запрашивает номер по порядку регистрации partec.xpar
- "partool" запрашивает номер по порядку регистрации partool.xpar

"@Ini"

Считывает сочетание ключ=значение с файла tra.ini. Параметр **идентификатор** - это имя ключа для считывания в файле tra.ini в разделе [Tra]. Для считывания из определенного раздела к имени ключа необходимо добавить имя раздела в квадратных скобках.("[Раздел]Ключ").

"@ShellExecute"

Просит у операционной системы открыть файл с использованием программы, связанной с расширением файла. Можно также запустить исполняемый файл. Параметр **идентификатор** - это имя открываемого файла или имя запускаемой программы. Имя файла может объявляться полным маршрутом, в противном случае его поиск выполняется в текущей папке Albatros. Поиск имени файла выполняется также среди файлов, определенных при помощи "@FileName". Параметр **контейнер**, если при открытии файла не были выявлены ошибки, будет содержать значение 0, в противном случае он будет содержать код ошибки.

"@StartProg"

Выполняет программу, определенную в параметре **идентификатор**. Невозможно передать аргументы запускаемой программе. Имя программы должно содержать полный маршрут, в противном случае его поиск выполняется в текущей папке Albatros. Поиск имени программы выполняется также среди определенных при помощи "@FileName". Параметр **контейнер**, если запуск программы удался, будет содержать значение 0, в противном случае он будет содержать код ошибки. Если программа уже была запущена, код ошибки - 1056.

"@TermProg"

Завершает программу, определенную в параметре **идентификатор** запущенную при помощи "@StartProg". Имя программы должно содержать полный маршрут, в противном случае его поиск выполняется в текущей папке Albatros. Поиск имени программы выполняется также среди определенных при помощи "@FileName". Параметр **контейнер**, если запуск программы удался, будет содержать значение 0, в противном случае он будет содержать код ошибки. Если программа уже была запущена, код ошибки - 1056.

"@ProgRunning"

Проверяет, выполняется ли еще программа, запущенная при помощи "@StartProg". Имя программы определяется в параметре **идентификатор**. Имя программы должно содержать полный маршрут, в противном случае его поиск выполняется в текущей папке Albatros. Поиск имени программы выполняется также среди определенных при помощи "@FileName". Параметр **контейнер**, если программа еще выполняется, будет содержать значение 1, в противном случае будет содержать значение 0.

"@DialogFile"

Открывает диалоговое окно Файл Открыть или Файл Сохранить для обеспечения выбора имени файла. Для открытия окна Файл Открыть задать параметр **идентификатор** = "Open", для открытия окна Файл Сохранить задать параметр **идентификатор** = "Save". Имя выбранного файла записывается в параметр **контейнер**.

"@AxisCorrectors"

Заменяет таблицу корректоров линейности оси новой таблицей, загруженной из файла, которая, в любом случае, должна иметь такое же число корректоров и те же оси, что и для перекрестных корректоров.

Параметр **идентификатор** - это имя файла, которое обычно имеет расширение .csv и находится в папке ...\\Mod.n\\Config (имя файла 'нормализуется', как происходит, например, для "@FileExist"). Параметр **контейнер** определен как переменная INTEGER и будет содержать 1, если были направлены новые корректоры, или же 0.

"@Language"

Он получает переводимый текст, соответствующий сообщению группы, библиотеки или модуля, которое связано с инструкцией MESSAGE или инструкцией ERROR.

Допустимые значения для параметра **идентификатора**:

- "DEFMSG:число", где число - это последовательность цифр. Albatros записывает в **контейнер** текст сообщения модуля номера «номер».
- "DEFMSG:имя", где "имя" - имя, включая группу и библиотеку, DEFMSG. Albatros записывает в **контейнер** текст указанного сообщения. Если имя группы или библиотеки отсутствует, используется имя задачи, отправившей RECEIVE.
- "DEFMSG:*", Albatros записывает в **контейнер** текст сообщения группы или модуля, ранее указанного инструкцией SEND.

Пример
; В GPL

```

RECEIVE "@Param", "partec", 0, prog
RECEIVE "@Param", "partool", 0, prog

; в GPL
; считывает значение ключа Radix в разделе [Albatros] из файла tra.ini
RECEIVE "@INI", "[Albatros]Radix", 0, значение

; открывает окно файл Открыть и записывает имя файла в переменную
Имяфайла
RECEIVE "@DialogFile", "Открыть", 0, Имяфайла

; полное чтение файла
Function ReadProperties
PARAM file AS STRING
LOCAL version AS INTEGER
LOCAL size AS DOUBLE

SEND "@FileName" "theFile" 0 file
WAITRECEIVE "@FileInfo", "version:theFile", 0, version
WAITRECEIVE "@FileInfo", "size:theFile", 0, size

```

SEND

Синтаксис

SEND [получатель,] идентификатор, флажки [, информация]

Аргументы

получатель	константа строкового типа
идентификатор	константа строкового типа
флажки	константа типа integer
информация	имя устройства или константа или переменная (цифровая или строковая)

Описание

Эта инструкция, вместе с RECEIVE, используется для обмена информацией между модулями установки и ПК-супервизором. SEND используется для отправления информации, RECEIVE - для ее запроса. Информация может направляться на Albatros или на внешнюю программу (Server OLE Automation). Во втором случае, информация, в любом случае, будет получена Albatros, которая, в свою очередь, обеспечит ее отправку внешней программе.

Параметр **получатель** представляет собой строку, позволяющую указывать, кому направлять запрос информации. Существует три класса получателей:

- получатели, начинающиеся с символа "@" (см. список ниже). На самом деле получатель - Albatros или, точнее, одна из ее функций.
- получатели, не начинающиеся с символа "@". Они рассматриваются как Server OLE. При получении первой направленной им информации Albatros попытается направить их в исполнение и, следовательно - передать им информацию, полученную от модуля.
- неуказанный получатель (параметр, на самом деле, факультативен). В этом случае информация сохраняется Albatros в таблице и остается доступной для выполняющих ее запрос (другой модуль или внешняя программа).

Параметр **идентификатор** - это имя информации, не может опускаться. Принимает различные значения в зависимости от получателя:

- если получатель - Albatros, это будет команда, связанная с функцией, к которой выполняется доступ
- если получатель - сервер OLE, это будет свойство созданного объекта OLE.
- если получатель не указан, информация в хранимой Albatros таблице будет определяться ярлыком.

Параметр **флажки** позволяет определять, как информация должна использоваться Albatros. Допустимые значения и их следствия таковы:

значение	команда	описание
\$0001H	Broadcast	Нормальная отправка информации
\$0008H	CancelAfter	Информация будет удалена после считывания
\$0020H	ReadOnly	Информация может стираться только отправителем

\$1000H UpdateFlags Меняет состояние информации (считанная/ для считывания) без изменения данных

\$8000H Delete Удаляет информацию

Наконец, параметр **информация** - это направляемая информация. Она может опускаться и в этом случае отправка пустой информации приобретает значение извещения события (может использоваться для синхронизации выполнения кода GPL на различных модулях). В качестве параметра информации допускаются все устройства (за исключением осей), простые переменные GPL и строки.

Список **получателей**, которыми управляет Albatros, и соответствующих команд:

"@List"

Позволяет управлять командами Моделирование и Уставка

Допускаются следующие команды (параметр **идентификатор**):

- Sim: извещает изменение состояния флажкового выключателя Моделирование. В зависимости от состояния флажка отображается нажатой или отжатой кнопка, которая обозначает его на панели инструментов (1=выбрана, 0=не выбрана)
- Setp: извещает изменение состояния флажкового выключателя CmdSetp. В зависимости от состояния флажка отображается нажатой или отжатой кнопка, которая обозначает его на панели инструментов (1=выбран, 0=не выбран)
- Esc: извещает изменение состояния флажкового выключателя Исключен. В зависимости от состояния флажка отображается нажатой или отжатой кнопка, которая обозначает его (та же, что и для флажкового выключателя CmdSetp) на панели инструментов (1=выбран, 0=не выбран)
- End: завершает выполнение списка. Эта команда нажимает кнопки Пуск и Стоп и отключает пункты меню Пуск и Стоп
- Hold: нажимает кнопку Стоп и подключает пункт меню Стоп. Отжимает кнопку Пуск и отключает пункт меню Пуск

"@Syn"

Связь между GPL и программой отображения обзорных экранов. Позволяет открывать и закрывать обзорные экраны, управление которыми осуществляется GPL, и направлять информацию в ячейку обзорного экрана. Допускаются следующие команды (параметр **идентификатор**):

- "Open:имяфайла" открытие обзорного экрана *имяфайла.xsyn*
 - "Close:имяфайла" закрытие обзорного экрана *имяфайла.xsyn*
 - "Open" открытие обзорного экрана. Имя файла считывается переменной **информация**
 - "Close" закрытие обзорного экрана. Имя файла считывается переменной **информация**
 - "имяячейки" ячейка, в которой должна отображаться направленная информация
- Можно взаимодействовать с окном перемещения осей по требованиям, определенным также для приведенного ниже параметра **получатель** ["@Devices"](#).

"@File"

Запись в файл. Позволяет создавать персонализированные журналы регистрации для записи выполненных станком операций. Это текстовые файлы (ASCII). Параметр **идентификатор** - это имя файла, в котором будет выполняться запись.

Имя файла - это строковая константа. Если в параметре идентификатора не указан полный маршрут для сохранения файла, Albatros принимает маршрут, определенный в tra.ini в разделе [tra] в пункте dirreport.

Если идентификатор начинается и заканчивается символом %, поиск строки внутри него выполняется в tra.ini в разделе [tra] и используется в качестве имени файла. В имя могут вводиться символы, которые будут заменяться при выполнении инструкции:

- %n номер модуля, который выполняет инструкцию SEND
- %h текущий час (формат 00-23)
- %d текущий день (формат 01-31)
- %m текущий месяц (формат 01-12)
- %y текущий год (формат из 4 цифр)

См. пример.

Операции записи выполняются в режиме добавления (данные добавляются в конце файла). В файл могут направляться числовые данные (автоматически преобразовываемые в ASCII) или строки. Можно записывать строки в формате дата и время при помощи символов формата %d для даты и %t для времени. Для времени используется формат "HH:mm:ss" (т. е., часы, минуты и секунды, разделенные ":"), а для даты - формат, который определяется национальными особенностями. Можно использовать другой формат, задавая в tra.ini в разделе [Albatros] пункт "LogNoLocale=1" (умолчание - LogNoLocale=0, т. е. использование текущего формата). Можно также задать формат для даты и времени независимо от формата, заданного в Windows, все также определяя в tra.ini в разделе [Albatros] пункты "LogDateFormat=" и "LogTimeFormat=" и задавая строку символов по приведенной ниже схеме. Если эти пункты отсутствуют или незаполнены, используются форматы, заданные в Windows.

Формат времени

h	время в формате 12 часа без начальных нулей
---	---

hh	время в формате 12 часов с начальными нулями
H	время в формате 24 часа без начальных нулей
HH	время в формате 24 часов с начальными нулями
m	минуты без начальных нулей
mm	минуты с начальными нулями
s	секунды без начальных нулей
ss	секунды с начальными нулями
t	только один символ для указания метки времени, например, A или P
tt	несколько символов для указания метки времени, например, AM или PM

Примечание: форматы "t" и "tt" используют метку времени, указанную на панели управления текущего пользователя. Это не обязательно "AM" и "PM".

Пример: если сейчас 11:29 после обеда и строка имеет следующий состав "hh':'mm':'ss tt", будет написано "11:29:40 PM".

Формат дня

d	число месяца без начальных нулей, представленное цифрами
dd	число месяца с начальными нулями, представленное цифрами
ddd	день недели, представленный символами и сокращенный до трех букв
dddd	день недели, представленный символами с полным именем
M	месяц без начальных нулей, представленный цифрами
MM	месяц с начальными нулями, представленный цифрами
MMM	месяц, представленный символами и сокращенный до трех букв
MMMM	месяц, представленный символами с полным именем
y	год с двумя цифрами без начальных нулей для меньше 10 лет
yy	год с двумя цифрами с начальными нулями для меньше 10 лет
yyyy	год, представленный 4 или 5 цифрами в зависимости от используемого календаря
yyyyy	год, представленный 4 или 5 цифрами в зависимости от используемого календаря

Например, для среды 31 августа 1994 года строка имеет следующий вид "dddd,' MMM dd yy", будет написано "Срд, август 31 94"

Если эта информация опускается, к файлу добавляется "возврат в начало".

"@FileName"

Записывает в память ассоциацию между строковой константой и именем файла, составляемым строковыми переменными. При получении сообщения об ассоциации Albatros она заменяет все последующие имена файла именем, полученным этой инструкцией. Параметр **идентификатор** - это имя файла, в котором будет выполняться запись. Имя файла - это строковая переменная. Если в параметре идентификатора не указан полный маршрут для сохранения файла, Albatros принимает маршрут, определенный в tra.ini в разделе [tra] в пункте dirreport. Значение параметра идентификатора записывается в tra.ini в разделе [GPLFileName] в пункте Log с тем, чтобы его можно было повторно использовать при последующих выполнениях Albatros. Для удаления ассоциации необходимо задать в качестве параметра идентификатора пустую строку. Определенная таким образом ассоциация действительна для всех модулей.

"@FileDelete"

Удаление файла. Параметр **идентификатор** - это имя файла, который будет удален (полный маршрут). Если в параметре идентификатора не указан полный маршрут для сохранения файла, Albatros принимает маршрут, определенный в tra.ini в разделе [tra] в пункте dirreport. Имя файла может определяться по правилам, описанным для параметра **получатель @File**.

"@FileRead"

Позиционирует указатель файла в начало файла. Параметр **идентификатор** - это имя файла (полный маршрут). Если в параметре идентификатора не указан полный маршрут для сохранения файла, Albatros принимает маршрут, определенный в tra.ini в разделе [tra] в пункте dirreport. Имя файла может определяться по правилам, описанным для параметра **получатель @FileRead**.

"@Axis"

Взаимодействует с окном ручного перемещения осей по спецификациям, определенным также для указанного ниже параметра получателя "@Devices". Если окно, контролирующее перемещение указанной оси, уже открыто, эта команда воздействует на это окно независимо от того, открыто оно в обзорном, или же в диагностическом режиме. Если окно закрыто, команда пытается открыть его в диагностическом режиме или в одном из уже открытых обзорных экранов, содержащих эту ось.

"@Devices"

Запрос открытия или закрытия окна Диагностика применительно к направляющему информацию модулю. Выполнение команд в окне перемещения оси при диагностике. Параметр **идентификатор** может принимать следующие значения:

- "Open" открытие Диагностики
- "Close" закрытие Диагностики

Параметр **идентификатор**, когда необходимо взаимодействовать с окном перемещения оси, может принимать следующие значения:

"MoveAX#имя_оси#Open"	открытие окна перемещения оси.
"MoveAX#имя_оси#Close"	закрытие окна перемещения оси.
"MoveAX#имя_оси#Plus"	нажатие кнопки перемещения оси в положительном направлении
"MoveAX#имя_оси#Minus"	нажатие кнопки перемещения оси в отрицательном направлении
"MoveAX#имя_оси#Stop"	нажатие кнопки стоп движения
"MoveAX#имя_оси#Jog"	задает режим движения для смещений, управляемых в режиме runtime оператором
"MoveAX#имя_оси#Step"	задает режим движения для смещений на заданный шаг
"MoveAX#имя_оси#Absolute"	задает режим движения со смещением на установленную координату оси

где имя_оси является именем оси, отображенным в окне. Например, если необходимо открыть окно перемещения оси X, параметр **идентификатор** будет "@MoveAX#X#Open". Имя оси может быть в одной из следующих форм:

1. Имя_Группы.Имя_подгруппы.Имя_Оси или Имя_Группы.Имя_Оси: предоставляется полный маршрут оси.
2. Имя_Оси: для идентификации правильной оси по порядку выполняются следующие проверки:
 - если задача, от которой поступает команда, является функцией подгруппы, поиск оси осуществляется в этой подгруппе.
 - если задача, от которой поступает команда, является функцией главной подгруппы, поиск оси осуществляется во всей группе. При наличии более одной оси с этим именем поиск завершается неудачно.
 - если предыдущие проверки завершаются неудачно, поиск оси осуществляется во всех группах модуля. При наличии более одной оси с именем Имя_Оси поиск не имеет положительного исхода.

Можно запретить пользователю нажимать кнопки перемещения оси во всех окнах перемещения оси модуля в режиме диагностики, задавая параметр **идентификатор** следующим образом:

- "MoveAX#UIENABLE", если параметр **информация** задается на 0, перемещение осей Albatros отключается. При задании на 1 подключается перемещение осей Albatros.

Отключение перемещения осей Albatros рекомендуется при перемещении осей с кнопочного пульта станка.

"@Vars"

Запрашивает сохранение содержания глобальной переменной GPL в архиве технологических параметров или инструмента. Параметр **идентификатор** - это имя глобальной переменной (станка или группы или библиотеки), для которой требуется обновление.

"@Application"

Взаимодействие с Albatros. Позволяет закрывать Albatros или отображать на экране "окно сообщений", чтобы проинформировать пользователя или запросить согласие на последующие задачи. Для параметра **идентификатор** допускаются следующие значения:

"Quit"	закрывает Albatros
"Lock"	запрещает закрытие Albatros из Файл->Выход или сочетанием клавиш [ALT+F4] или кнопкой закрытия.
"UnLock"	восстанавливает возможность закрыть Albatros
"MsgBox:flags"	открывает окно сообщений
"	"

Поведение окон сообщений контролируется "флажками" строки **идентификатор**. Это может быть последовательность следующих символов (они могут быть как в верхнем, так и в нижнем регистре):

"O"	кнопка "ОК"
"C"	кнопка "Отмена"
"Y"	кнопка "Да"
"N"	кнопка "Нет"
"R"	кнопки "Повторить"
"S"	значок знака Стоп
"?"	информационный значок, состоящий из строчной буквы "i" внутри круга
"!"	значок с восклицательным знаком
"*"	информационный значок
"1"	первая кнопка - кнопка по умолчанию
"2"	вторая кнопка - кнопка по умолчанию
"3"	третья кнопка - кнопка по умолчанию

"4" четвертая кнопка - кнопка по умолчанию

Если не указано, кнопка по умолчанию - первая.

Например, "MsgBox:YN2" определяет окно сообщений с информационным значком, двумя кнопками "Да" и "Нет", из которых вторая кнопка - кнопка по умолчанию. Параметр **информация** может быть строкой, содержащей текст для отображения, или целым числом, которое интерпретируется как код сообщения модуля, управляемого TraLangs.exe, или ярлыком группового сообщения, определенного инструкцией [DEFMSG](#).

Что касается текста, если внутри него есть символ новой строки «\ u000A», текст будет разделен на две части, и первая часть будет отображаться как текст окна сообщения, а вторая часть будет отображаться как пояснение или деталь текста.

Кнопки отображаются на языке Windows.

"@Help"

Открытие файла справки. Позволяет управлять отображением файла справки с указанием отображаемой темы. Для параметра **идентификатор** допускаются следующие значения:

- "Open:имяфайла" открытие файла справки
- "Close:имяфайла" закрытие файла справки

Часть "имяфайла" строки указывает имя открываемого файла справки.

Параметр **информация** может быть строкой или числом и принимать, соответственно, значение ключа или контекстного номера (служит для определения страницы или темы справки для отображения).

"@Report"

Добавляет сигналы к файлу отчета Albatros (MONTH(№ месяца).TER). Параметр **идентификатор**:

- "Add"

Параметр **информация** может быть:

- строковая переменная или строковая константа: в этом случае в память записывается текст, содержащийся в строке
- переменная integer или числовое значение integer: в этом случае в память записывается текст, определенный в инструкции [DEFMSG](#).

"@Ini"

Записывает сочетание ключ=значение в файл tra.ini. Параметр **идентификатор** - это имя ключа, которое должно добавляться в файл tra.ini в разделе [Tra]. Для записи в определенный раздел к имени ключа необходимо добавить имя раздела в квадратных скобках.("[Раздел]Ключ").

Параметр **информация** может быть строковой или числовой переменной, строковой или числовой константой.

"@ShellExecute"

Просит у операционной системы открыть файл с использованием программы, связанной с расширением файла. Можно также запустить исполняемый файл. Параметр **идентификатор** - это имя открываемого файла или имя запускаемой программы. Имя файла может объявляться полным маршрутом, в противном случае его поиск выполняется в текущей папке Albatros. Поиск имени файла выполняется также среди файлов, определенных при помощи "@FileName".

"@StartProg"

Выполняет программу, определенную в параметре **идентификатор**. Невозможно передать аргументы запускаемой программе. Имя программы должно содержать полный маршрут, в противном случае его поиск выполняется в текущей папке Albatros. Поиск имени программы выполняется также среди определенных при помощи "@FileName".

"@TermProg"

Завершает программу, определенную в параметре **идентификатор** запущенную при помощи "@StartProg". Имя программы должно содержать полный маршрут, в противном случае его поиск выполняется в текущей папке Albatros. Поиск имени программы выполняется также среди определенных при помощи "@FileName".

"@DialogFile"

Позволяет задавать некоторые параметры диалогового окна Файл Открыть или Файл Сохранить.

Для параметра **идентификатор** допускаются следующие значения:

- "Extension" если пользователь не вводит расширение, используется расширение, определенное в параметре **информация** (переменная или строковая константа)
- "Filter" задает фильтр типов файлов для использования. Параметр **информация** может быть строковой переменной или строковой константой и в этом случае используется как фильтр текст, содержащийся в строке, или целой переменной или целым цифровым значением и в этом случае используется как фильтр текст, определенный в команде [DEFMSG](#).

"Flags"	задает флажки инициализации. Список значений, задаваемых в поле информация (переменная или константа целого типа), см. в официальной документации Microsoft по члену Флажки структуры OPENFILENAME.
"InitalDir"	задает исходную папку, определенную в поле информация (строковая переменная или константа)
"Title"	задает название окна. Параметр информация может быть строковой переменной или строковой константой и в этом случае в качестве заголовка используется текст, содержащийся строке, или целой переменной или целым цифровым значением и в этом случае в качестве заголовка используется текст, определенный в команде DEFMSG .
"@Language"	Устанавливает номер сообщения группы, модуля или библиотеки, который будет использоваться в следующем RECEIVE с тем же идентификатором. Допустимое значение для параметра идентификатора - "DEFMSG:*". Параметр информация может быть переменная целочисленного типа или целочисленная константа, и в этом случае он определяет номер отображаемого группового сообщения. Он может быть строкового типа или строковой константы, и в этом случае он определяет имя DEFMSG.

Пример

```
; Пример инструкции "send file" с именем, построенным при выполнении.
; Предположим, что дата выполнения инструкции
; - 31-01-2000
```

```
; в GPL
SEND"@File", "%Log%", 0, "Начало выполнения"
; добавляет "возврат в начало"
; в файле tra.ini в разделе [TPA] добавляется
SEND"@File", "%Log%", 0
Log=c:\Albatros\report\%y\Rep%m%d.txt
```

```
; Получаемое имя файла:
c:\Albatros\report\2000\Rep0131.txt
```

```
; Пример инструкции "send Vars"
; определяется переменная Var_SendVars
; as double в файле глобальных переменных
; в технологических параметрах вводится Var_SendVars
; в поле "Имя матрицы"
; в GPL
SETVAL 100.0,Var_SendVars
; направляет значение 100.0 в параметр технологических параметров
; связан с переменной Var_SendVars
SEND"@Vars", "Var_SendVars", 0
```

```
; Пример инструкции send INI
; в tra.ini записывается ключ Radix в раздел [Albatros]
; для задания числовой основы отображения десятичных чисел
SEND "@INI", "[Albatros]Radix", 0,1
```

```
; Пример задания ассоциации между постоянной строкой GPL
; и именем файла.
```

```
; объявление переменной строкой
имяфайла as string
; состав имени файла
setstring "c:\albatros\report\LogFile.txt", имяфайла
; ассоциация
SEND "@FileName", "LOG", 0, имяфайла
; все операции записи происходят с этого момента
; выполняются в файле, определенном переменной имяфайла
SEND "@File", "LOG", 0, "Запись в файл LOG"
```

SENDIPC

Синтаксис

SENDIPC	имяIPC, ожидание [, имяпер1 [, имяперN, ...]]
SENDIPC	имяIPC, ожидание, матрица[строка]
SENDIPC	имяIPC, ожидание, вектор
SENDIPC	имяIPC, ожидание, матрица

Аргументы

имяIPC	строковая константа Имя IPC
ожидание	заданная константа. Режим ожидания считывания команды Допускаются следующие значения: WAIT ожидает считывание команды NOWAIT не ожидает считывание команды
имяпер1[...имяперN]	константа или переменная. Имена переменных 1÷N
матрица[строка]	константа или переменная integer. Номер строки матрицы
вектор	Имя вектора
матрица	Имя матрицы

Описание

Направляет команду IPC в совместно используемую память "**имяIPC**". При первом исполнении инструкции SENDIPC выделяется совместно используемая память, размер которой вычисляется в зависимости от направляемых данных. Максимальный размер совместно используемой памяти составляет 64 кб. Могут определяться максимум 48 совместно используемые памяти, определяемые 48 однозначными именами. С совместно используемой памятью связан светофор, позволяющий синхронизировать выполнение получающих к ней доступ задач. Задача, записывающая данные, включает светофор по завершении записи, задача, считывающая данные, отключает его по завершении считывания. Если в качестве параметра **ожидание** был указан WAIT, задача, записывающая данные, перед продолжением выполнения дожидается, когда они будут считаны (светофор отключен). SENDIPC без данных ограничивается синхронизмом задач. В этом случае не выделяется совместно используемая память.

Интермодуль IPC

Два удаленных модуля могут обмениваться данными при помощи IPC. Эти IPC называются интермодулями IPC. Для определения интермодуля IPC необходимо записать **имяIPC** со следующим формализмом:
номер исходного модуля, "->", номер конечного модуля, ":", затем - другие символы имени IPC. Например, "0->1:Базовые параметры".

См. также [WAITIPC](#) и [TESTIPC](#).

WAITIPC

Синтаксис

WAITIPC	имяIPC [, имяпер1 [, имяперN, ...]]
WAITIPC	имяIPC, матрица[строка]
WAITIPC	имяIPC, вектор
WAITIPC	имяIPC, матрица

Аргументы

имяIPC	строковая константа. Имя IPC
имяпер1[...имяперN]	константа или переменная. Имена переменных 1÷N
матрица[строка]	константа или переменная integer. Номер строки матрицы
вектор	Имя вектора
матрица	Имя матрицы

Описание

Получает команду IPC из совместно используемой памяти "**имяIPC**". При первом исполнении инструкции WAITIPC выделяется совместно используемая память, размер которой вычисляется в зависимости от направляемых данных. Максимальный размер совместно используемой памяти составляет 64 кб. Могут определяться максимум 48 совместно используемые памяти, определяемые 48 однозначными именами. С совместно используемой памятью связан светофор, позволяющий синхронизировать выполнение получающих к ней доступ задач. Считывающая данные задача ждет, когда будет подключен светофор задачей, которая записывает данные, считывает данные и отключает светофор.

WAITIPC без данных ограничивается синхронизмом задач. В этом случае не выделяется совместно используемая память.
См. также [SENDIPC](#) и [TESTIPC](#).

WAITRECEIVE

Синтаксис

WAITRECEIVE [источник,] идентификатор, флажки [, контейнер]

Аргументы

источник	константа строкового типа
идентификатор	константа строкового типа
флажки	константа типа integer
контейнер	имя устройства или переменной (цифровой или строковой)

Описание

Ожидает, пока придет запрошенная информация (указанная **идентификатором**) перед продолжением выполнения программы GPL. По использованию см. документацию инструкции [RECEIVE](#).

10.3.10 Математика

ABS

Синтаксис

ABS операнд, результат

Аргументы

операнд	константа или переменная или имя устройства
результат	переменная или имя устройства

Описание

Выводит абсолютное значение **операнда** и помещает его в **результат**. Для преобразования данных в зависимости от типа объявленных данных см. главу [Преобразование данных](#).

Пример

```
SetVal -10,op ; задает -10 переменной op
Abs op,var
```

;значение, которое будет помещено в переменную var, будет 10

ADD

Синтаксис

ADD операнд1, операнд2, результат

Аргументы

операнд1	константа или переменная или имя устройства
операнд2	константа или переменная или имя устройства
результат	переменная или имя устройства

Описание

Выполняет операцию суммирования между **операндом1** и **операндом2** и помещает результат в **результат**. Для преобразования данных в зависимости от типа объявленных данных см. главу [Преобразование данных](#).

Пример

```
SetVal 5,op1 ; задает 5 переменной op1
Add op1,3,var
```

;значение, которое будет помещено в переменную var, будет 8

AND

Синтаксис

AND операнд1, операнд2, результат

Если операнды в выражении не все одного типа, выполняется автоматическое преобразование и тип результата операции - больший из двух в соответствии со следующим правилом:

- char < integer
- float < double
- char или integer < float или double.

После решения **выражения** результат преобразуется в тип **переменной**.

Разрешаются следующие операторы, сгруппированные по уровням превосходства:

()	скобки
-	оператор изменения знака
ABS	абсолютное значение операнда
ROUND	округление единиц измерений
TRUNC	усечение значения до целой части
LOG	натуральный логарифм
LOGDEC	десятичный логарифм
EXP	экспонента
SRQ	операция квадратного корня
SIN	операция синуса. Операнд выражен в градусах и может иметь дробную часть с сотыми долями (напр., 30° 15" = 30,25.)
COS	операция косинуса. Операнд выражен в градусах и может иметь дробную часть с сотыми долями (напр., 30° 15" = 30,25.)
TAN	операция тангенса. Аргумент выражен в градусах
ARCSIN	операция арксинуса. Результат выражен в градусах и его значение находится в интервале -90°ч+90°
ARCCOS	операция арккосинуса. Результат выражен в градусах и его значение находится в интервале 0°ч180°
ARCTAN	выполняет операцию арктангенса. См. ARCTAN
^	оператор степени

*	умножение
/	деление
%	остаток деления (модуль)
+	сумма
-	вычитание

Эта инструкция позволяет упростить запись кода GPL в случаях, когда должны выполняться математические расчеты, заменяя отдельные инструкции GPL, соответствующие операторам, перечисленным в таблице. Указанные инструкции остаются доступными для совместимости.

Пример

; вычисление расстояния между двумя точками

EXPR dist = SQR ((Xb - Xa) ^ 2 + (Yb - Ya) ^ 2)

; преобразование данных в EXPR

local op1 as double

local op2 as integer

local op3 as float

local результат as integer

setval 900,op1

setval 100,op2

setval 500.0,op3

EXPR результат = (op1 + op2) / op3

; вначале решается выражение в double

; затем выполняется преобразование в integer

; как тип данных результата

LOG

Синтаксис

LOG

операнд, результат

Аргументы

операнд константа или переменная или имя устройства
результат переменная или имя устройства

Описание

Вычисляет натуральный логарифм **операнда** и помещает значение в **результат**. Для преобразования данных в зависимости от типа объявленных данных см. главу [Преобразование данных](#).

Пример

```
SetVal 10,op ; задает 10 переменной op
Log op,var
;Значение, которое будет помещено в переменную var, будет 2.302585093
```

LOGDEC**Синтаксис**

LOGDEC **операнд, результат**

Аргументы

операнд константа или переменная или имя устройства
результат переменная или имя устройства

Описание

Вычисляет десятичный логарифм **операнда** и помещает значение в **результат**. Для преобразования данных в зависимости от типа объявленных данных см. главу [Преобразование данных](#).

Пример

```
SetVal 10,op ; задает 10 переменной op
Logdec op,var
;Значение, которое будет помещено в переменную var, будет 1
```

MOD**Синтаксис**

MOD **операнд1, операнд2, результат**

Аргументы

операнд1 константа или переменная integer или имя устройства
операнд2 константа или переменная integer или имя устройства
результат переменная integer или имя устройства

Описание

Выполняет операцию модуля между **операндом1** и **операндом2** и помещает результат в **результат**. Модуль - это остаток деления между первым и вторым операндом. Инструкция может генерировать ошибку системы, когда **операнд2** равен 0. Для преобразования данных в зависимости от типа объявленных данных см. главу [Преобразование данных](#).

Пример

```
SetVal 20,op1 ; задает 20 переменной op1
SetVal 3,op2 ; задает 3 переменной op2
Mod op1,op2,var
;Значение, которое будет помещено в переменную var, будет 2
```

MUL**Синтаксис**

MUL **операнд1, операнд2, результат**

Аргументы

операнд1 константа или переменная или имя устройства
операнд2 константа или переменная или имя устройства

результат переменная или имя устройства

Описание

Выполняет операцию умножения между **операндом1** и **операндом2** и помещает результат в **результат**. Для преобразования данных в зависимости от типа объявленных данных см. главу [Преобразование данных](#).

Пример

```
SetVal 5,op1 ; задает 5 переменной op1
SetVal 2,op2 ; задает 2 переменной op2
Mul op1,op2,var
```

;Значение, которое будет помещено в переменную var, будет 10

NOT

Синтаксис

NOT **операнд**

Аргументы

операнд переменная или имя устройства

Описание

Выполняет бинарную операцию NOT (*инвертируются отдельные биты*) значения, выраженного **операндом**. Результат записывается в **операнд**.

Пример

```
SetVal 5,var ; задает значение 5 "var"
Not var
```

```
; Результат будет var = -6
; Бинарная запись: 5 = 0000 0101,
; Бинарная запись:10 = 0000 1010
; Шестнадцатеричная запись 5 = 0000 0000 0000 0005
; Шестнадцатеричная запись 10 = 0000 0000 0000 000A
; при выполнении NOT на значении 5 получаем 0xFFFF FFFF FFFF FFFA = -6
```

OR

Синтаксис

OR **операнд1, операнд2, результат**

Аргументы

операнд1 константа или переменная или имя устройства
операнд2 константа или переменная или имя устройства
результат переменная или имя устройства

Описание

Выполняет бинарную операцию OR (из двух битов, дает 1, если по меньшей мере один из двух равен 1) между **операндом1** и **операндом2** и помещает результат в **результат**. Для преобразования данных в зависимости от типа объявленных данных см. главу [Преобразование данных](#).

Пример

```
;Значение, которое будет помещено в переменную var, будет 7
;(Бинарная запись: 5 = 0101, 3 = 0011, 7 = 0111 )
```

```
Or 5,3,var
```

RANDOM

Синтаксис

RANDOM **мин, макс, результат**

Выполняет операцию округления на **операнде** и помещает значение в **результат**. Для преобразования данных в зависимости от типа объявленных данных см. главу [Преобразование данных](#).

Пример

```
SetVal 5.7,op ;задает 5.7 переменной op
Round op,var

;Значение, которое будет помещено в переменную var, будет 6

SetVal 5.2,op ;задает 5.2 переменной op
Round op,var

;Значение, которое будет помещено в переменную var, будет 5
```

SETBIT

Синтаксис

```
SETBIT маскера, nbit
```

Аргументы

маска целая константа или переменная или номерсчетчика или имяпорта. Изменяемое значение (макс. 32 бит)

nbit целая константа или переменная или номерсчетчика. Номер изменяемого бита (1÷32)

Описание

Задаёт на 1 отдельный бит, указанный **nbit**, **переданной битовой маски**. Аргумент **маска** должен соответствовать целому значению максимум с 32 бит. Номер бита, **nbit**, находится в пределах от 1 до 32.

Пример

Состояние порта до выполнения кода



Состояние порта после выполнения кода



```
-----
;
; Пример для подключения одной строки флагового порта:
;
;
;-----
```

```
SetVal 2,nbit
Setbit FlagPort,nbit

; подключает строку 2 флагового порта
```

SHIFTL

Синтаксис

```
SHIFTL операнд1 [, операнд2]
```

Аргументы

операнд1 переменная (integer или char) или имя устройства

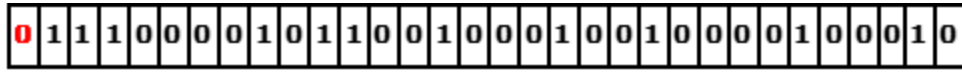
операнд2 переменная (integer или char) или имя устройства

Описание

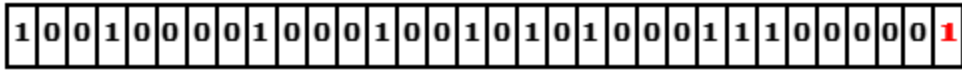
Если не указан **операнд2**, выполняет операцию прокрутки влево битов, образующих **операнд1**. Если указан и второй операнд, выполняет операцию ротации между битами **операнда2** и битами **операнда1**. По завершении операции **операнд2** будет считать перенос, то есть высокий бит **операнда 1**.

Пример

Вращение операндов типа integer (сдвиг влево с переносом)
До вращения



Операнд1



Операнд2

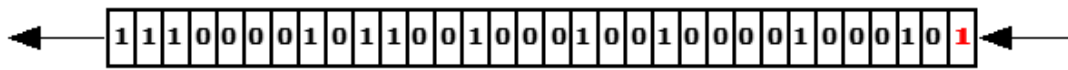
После вращения



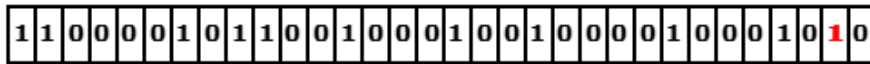
Операнд1

Операнд2

Прокрутка (сдвиг влево без переноса)



Операнд1



Операнд1 после
сдвига влево

SHIFTR

Синтаксис

SHIFTR

операнд1 [, операнд2]

Аргументы

operand1
operand2

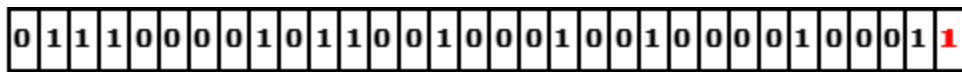
переменная (integer или char) или имя устройства
переменная (integer или char) или имя устройства

Описание

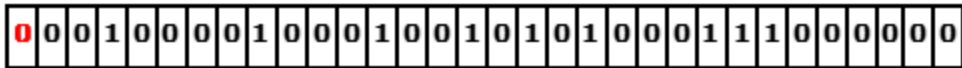
Если не указан **operand2**, выполняет операцию прокрутки вправо битов, образующих **operand1**. Если **operand1** объявлен типа char, высокий бит на входе всегда равен 0. Если **operand1** объявлен типа integer, бит 32 на входе является знаковым битом. Если указан и второй операнд, выполняет операцию ротации между **операндом2**, понимаемым как значение 0 или отличным от 0, и битами **операнда1**. По завершении операции **operand2** будет считать перенос операции и старший бит **операнда1** станет 0 или 1 в зависимости от исходного значения **операнда2** (0 или 1).

Пример:

Вращение операндов типа integer (сдвиг вправо с переносом)
До вращения



Операнд1



Операнд2

После вращения

Операнд2

Прокрутка вправо char (сдвиг вправо без переноса)

ENDREALTIMETASK

Синтаксис

ENDREALTIMETASK **имяфункции**

Аргументы

имяфункции имя функции

Описание

Завершает выполнение [задачи реального времени](#). См. также [STARTREALTIMETASK](#).

ENDTASK

Синтаксис

ENDTASK **[имязадачи]**

Аргументы

имязадачи имя задачи

Описание

Завершает выполнение задачи и всех подключенных ею задач (дочерние задачи). Кроме того, эта инструкция приостанавливает движение осей, отменяет незаконченные RECEIVE и закрывает возможные подключения на последовательных портах. Если переменная **имязадачи** опускается, она завершает выполнение текущей задачи.

GETPRIORITYLEVEL

Синтаксис

GETPRIORITYLEVEL **уровень [,имяфункции]**

Аргументы

уровень переменная. Уровень приоритета выполнения
имяфункции имя функции

Описание

Возвращает в переменной **уровень** значение приоритета задачи, определенной **имяфункцией**. Указанное значение является числом от 1 до 255, где 1 - самый высокий уровень приоритета, а 255 - самый низкий. Если **имяфункция** не определена, возвращается значение приоритета текущей задачи, т. е. функции, в которой выполняется инструкция GETPRIORITYLEVEL. См. также [SETPRIORITYLEVEL](#).

GETREALTIME

Синтаксис

GETREALTIME **имяпер**

Аргументы

имяпер переменная integer.

Описание

Возвращает в переменной **имяпер** время, прошедшее с начала последней задачи реального времени управления осями. Возвращаемое время выражается в микросекундах. См. также [GETREALTIMECOUNT](#).

GETREALTIMECOUNT

Синтаксис

GETREALTIMECOUNT **имяпер**

Аргументы

имяпер переменная integer.

Описание

Возвращает в переменной **имяпер** число задач реального времени управления осями, выполненных с момента последней инициализации ЧПУ. См. также [GETREALTIME](#).

HOLDTASK

Синтаксис

HOLDTASK [имязадачи]

Аргументы

имязадачи имя задачи

Описание

Приостанавливает выполнение задачи, определенной в **именизадачи**. Эта инструкция не приостанавливает перемещение осей, которые должны останавливаться инструкцией STOP. Если **имязадачи** опускается, приостанавливается выполнение текущей задачи.

RESUMETASK

Синтаксис

RESUMETASK [имязадачи]

Аргументы

имязадачи имя задачи

Описание

Возобновляет выполнение задач, определенных в **именизадачи**. Если **имязадачи** опускается, возобновляет выполнение текущей задачи. Если задача была приостановлена инструкцией STOPTASK, возобновляются также возможные движения осей.

SENDMAIL

Синтаксис

SENDMAIL mail, ожидание[, имяпер1 [,..имяперN]]
SENDMAIL mail, ожидание, матрица[строка]

Аргументы

mail константа или переменная integer. Номер почтового ящика (1ч256)
ожидание заданная константа. Режим ожидания считывания или выполнения команды. Константам ожидания могут присваиваться следующие значения:
 - **WAIT** ожидает считывание команды
 - **NOWAIT** не ожидает считывание команды
 - **WAITACK** ожидает выполнение команды
имяпер1[...имяперN] константа или переменная integer. Имена переменных 1ч20
матрица[строка] константа или переменная integer. Номер строки матрицы

Описание

Направляет сообщение (или команду) в ячейку **mail**. Сообщения могут использоваться для синхронизации и обмена информацией между двумя и более задачами.

Если ячейка **mail** не существует, т. е. еще не была выполнена инструкция [WAITMAIL](#) или [TESTMAIL](#), инструкция игнорируется.

Если принимающая задача не находится в режиме ожидания сообщения (инструкция [WAITMAIL](#)) или занята, переданные инструкцией данные (**имяпер** (1÷20) или строка матрицы указанная **матрицей[строка]**) записываются в очередь. В этом случае:

1. если аргумент ожидания - **NOWAIT**, выполнение продолжается со следующей инструкции;
2. если аргумент ожидания - **WAIT**, выполнение ожидает, пока сообщение будет считано получающей задачей;
3. если аргумент ожидания - **WAITACK**, выполнение ожидает, пока будет считано сообщение и подтверждено выполнение команды получающей задачей (при помощи инструкции [ENDMAIL](#) или новой [WAITMAIL](#)).

Очень важно, чтобы число переданных переменных и их типы совпадали с числом и типом переменных, использованных для создания почтового ящика инструкцией [WAITMAIL](#). Блок управления не позволяет использовать другие типы и не выполняет автоматических (cast) преобразований, как это обычно происходит.

SENDMAIL без дополнительных параметров (данных) становится простым механизмом синхронизации задач.

Пример

[Сервер перемещения осей](#)

SETPRIORITYLEVEL

Синтаксис

SETPRIORITYLEVEL **уровень [, имяфункции]**

Аргументы

уровень константа или переменная. Уровень приоритета выполнения
имяфункции имя функции

Описание

Задаёт значение приоритета, содержащееся в переменной **уровень**, задаче, определенной в **имяфункции**. Указанное значение является числом от 0 до 255, где 0 - самый высокий уровень приоритета, а 255 - самый низкий. Если имя задачи в переменной **имяфункции** не определено, меняется значение приоритета текущей задачи, т. е. уровень выполнения функции, в которой выполняется инструкция.

См. также [GETPRIORITYLEVEL](#).

STARTREALTIMETASK

Синтаксис

STARTREALTIMETASK **имяфункции**

Аргументы

имяфункции имя функции

Описание

Включает выполнение [задачи реального времени](#). Эта задача выполняется с такой же частотой реального времени, что и контроль осей. В отличие от обычных задач GPL выполняется каждое реальное время, от первой инструкции функции до первой инструкции FRET. См. также [ENDREALTIMETASK](#).

Примечание:

Локальные переменные, объявленные в задаче реального времени, инициализируются только при запуске задачи и затем сохраняют значение последнего выполнения.

STARTTASK

Синтаксис

STARTTASK **имязадачи [, параметры]**

Аргументы

имязадачи имя задачи
параметры параметры, которые могут потребоваться при выполнении задачи

Описание

Подключает выполнение [задачи](#) определенной в переменной **имязадачи**. Задачи могут передаваться **параметры**, используемые при выполнении. Число и тип передаваемых параметров должны соответствовать числу и типу, объявленным реализующей задачу функцией. Если задача уже выполняется, эта инструкция не имеет никакого эффекта.

Пример

[Последовательное / параллельное выполнение](#)

STOPTASK

Синтаксис

STOPTASK **имязадачи**

Аргументы

имязадачи имя функции

Описание

Останавливает выполнение [задачи](#) и всех остальных задач, которые запустила эта задача (дочерние задачи), останавливая перемещение осей (если выполняется такое действие).

Если **имязадачи** опускается, останавливает выполнение текущей задачи. Выполнение задачи и перемещение осей может подключаться снова инструкцией [RESUMETASK](#).

WAITMAIL

Синтаксис

WAITMAIL	mail [, имяпер1 [,.. имяперN]]
WAITMAIL	mail , матрица [строка]

Аргументы

mail	константа или переменная integer. Номер почтового ящика (1÷256)
имяпер1 [.. имяперN]	константа или переменная integer. Имена переменных 1÷20
матрица [строка]	константа или переменная integer. Номер строки матрицы

Описание

Получает сообщение из почтового ящика **mail**. С сообщением могут быть ассоциированы данные. Полученные вместе с сообщением данные записываются в указанные переменные **имяпер** (1÷20) или в строке матрицы, указанной **матрица**[**строка**].

Если в момент выполнения инструкции WAITMAIL уже не присутствуют ожидающие считывания сообщения, задача помещается в состояние HOLD, из которого выходит только когда другая задача направит в ящик с сообщением инструкцией [SENDMAIL](#) .

Соответствие переданных данных и данных, ожидаемых инструкцией, проверяется при выполнении инструкции.

WAITMAIL без дополнительных параметров становится простым механизмом синхронизации задач. См. также инструкции [SENDMAIL](#), [ENDMAIL](#) и [TESTMAIL](#)

Пример

[Сервер перемещения осей](#)

WAITTASK

Синтаксис

WAITTASK	имязадачи
-----------------	------------------

Аргументы

имязадачи	имя задачи
------------------	------------

Описание

Ожидает, пока завершит выполнение задача **имязадачи**.

Пример

[Последовательное / параллельное выполнение](#)

10.3.12 Управление потоком

CALL

Синтаксис

CALL	имяподпрограммы
-------------	------------------------

Аргументы

имяподпрограммы	имя подпрограммы, ярлык
------------------------	-------------------------

Описание

Выполняет подпрограмму, определенную ярлыком **имяподпрограммы**.

Каждая подпрограмма для возврата к инструкции, следующей за CALL, должна завершаться в точке выхода инструкцией [RET](#).

Примечание

Эта инструкция, вместе с RET, является типичным источником ошибок программирования. Рекомендуется быть внимательными при ее использовании, в частности, рекомендуется размещать подпроцедуры в конце тела функции (после инструкции FRET) для предупреждения случайного выполнения кода подпроцедуры так, как будто она является составной частью основного кода. Результатом этой ситуации, в лучшем случае, является ошибка системы, в других случаях наблюдаются аномальные действия станка, причину которых установить бывает сложно.

DELONFLAG

Синтаксис

DELONFLAG	имяфлажка
------------------	------------------

Аргументы

имяфлажка	имя флагового устройства
------------------	--------------------------

Описание

Отключает управление программного прерывания состоянием флагового бита или флажкового выключателя, которое ранее было подключено при помощи [ONFLAG](#).

DELONINPUT**Синтаксис**

DELONINPUT	имявхода
-------------------	-----------------

Аргументы

имявхода	имя входа
-----------------	-----------

Описание

Отключает управление программного прерывания состоянием входа, которое ранее было подключено при помощи инструкции [ONINPUT](#).

FCALL**Синтаксис**

[FCALL] имяфункции	имяфункции [, параметры] [параметры]
-------------------------------------	---

Аргументы

имяфункции параметры	имя вызываемой функции параметры, которые могут быть переданы функции
---------------------------------------	--

Описание

Выполняет вызов по функции, т. е. выполняется функция **имяфункции**.
Функции могут передаваться возможные **параметры**. Они должны соответствовать числу и типу параметров, объявленных в вызываемой функции.
Выполнение вызываемой функции (в которой выполняется FCALL) возобновляется по завершении выполнения вызываемой функции (которая определена параметром **имяфункции**).

Обратите внимание на разницу с инструкцией [STARTTASK](#), которая направляет на выполнение другую функцию параллельно с вызываемой (используемой для того, чтобы иметь несколько выполняемых задач одновременно).

Пример

[Последовательное / параллельное выполнение](#)

FOR/NEXT**Синтаксис**

FOR	индекс, начало, конец [, шаг]
инструкция	
инструкция	
...	
NEXT	

Аргументы

индекс	переменная или имясчетчика
начало	константа или переменная или имясчетчика Начальное значение
конец	константа или переменная или имясчетчика Конечное значение
шаг	константа или переменная или имясчетчика Шаг приращения или убывания

Описание

Циклически повторяет выполнение инструкций, заключенных между инструкцией FOR и инструкцией NEXT.

В ходе первого цикла переменная **индекс** инициализируется на значении переменной **начало**. На втором цикле переменная **индекс** будет иметь значение, равное (**начало+шаг**) и так далее до тех пор, пока переменная **индекс** не стане больше (или меньше в случае отрицательного значения переменной **шаг**) переменной **конец**. Если переменная **шаг** опускается, принимается значение по умолчанию, равное +1.

Инструкции, заключенные между FOR и NEXT, могут изменить число повторений, меняя **индекс**.

Когда повторения будут завершены, выполняется инструкция, следующая за NEXT.

Пример

```
Function Loop
local i As integer
local вектор[10] as integer

For i,1,10
Setval i, vettore[i] ; заполняет элементы вектора
; числами 1,2, .... 10
Next
Fret

Function loop2
local j As integer
local вектор[10] as integer

For j,1,10,2
Setval 27, вектор[j] ; вводит значение 27 в следующие
; элементы вектора: 1,3,5,7,9
Next
Fret
```

FRET

Синтаксис
FRET

Аргументы
никакого аргумента

Описание
Возврат от функции. Вызывает завершение выполнения функции и освобождение памяти, выделенной для локальных переменных. Если функция была направлена на исполнение при помощи FCALL, выполнение вызываемой функции возобновляется последующей инструкцией. Если ранее выполнялись WAITTASK с использованием текущей функции (в которой выполняется FRET) в качестве аргумента, ожидающие задачи разблокируются.

GOTO

Синтаксис
GOTO ярлык

Аргументы
ярлык

Описание
Выполняет безусловный переход к ярлыку, указанному параметром **ярлык**. Ярлык определяется ключевым словом, за которым сразу же следует символ ":". Ярлык должен находиться внутри тела функции, в которой выполняется инструкция GOTO.

Примечание
Тело функции находится между инструкцией FUNCTION, которая объявляет имя функции, и аналогичной инструкцией, определяющей последующую функцию (или концом файла). Из этого можно сделать вывод о возможности выполнения переходов с главного тела функции на возможные подпроцедуры (см. инструкции [CALL](#) и [RET](#)). Настоятельно не рекомендуется программировать подобным образом, т. к. это является источником многочисленных трудно выявляемых ошибок.

Пример

```
; функция, которая включает мигание флага
; (напр., тревожный индикатор на обзорном экране)
```

Function Loop

```

loop:
setflag    сигнал тревоги    ; подключает флажок
delay     1
resetflag  сигнал тревоги    ; отключает флажок
delay     1
goto      loop
fret

```

IF/IFVALUE/IF-THEN-ELSE**Синтаксис**

IF		имяпер, оператор сравнения, значение, GOTO ярлык
IF		имяпер, оператор сравнения, значение, CALL имяподпрограммы
IF		имяпер, оператор сравнения, значение, имяфункции
IF	инструкция	имяпер, оператор сравнения, значение THEN
	инструкция	
	...	
ENDIF		
IF		имяпер, оператор сравнения, значение THEN
	инструкция	
	инструкция	
	...	
ELSE		
	инструкция	
	инструкция	
	...	
ENDIF		

Аргументы

имяпер	константа или переменная или имяустройства
оператор сравнения	для выполнения сравнения могут использоваться следующие символы: < (меньше) = (равно) > (больше) =< (меньше или равно) >= (больше или равно) <> (не равно)
значение	константа или переменная или имяустройства
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Инструкции IF и IFVALUE являются синонимами. Рекомендуется использовать краткую форму. Эта инструкция позволяет выполнять сравнение между **имяпер** и **значение** и в зависимости от результата сравнения выполнять определенное действие.

В первых трех формах если сравнение положительно, выполняется переход к ярлыку (GOTO) или выполняется вызов подпрограммы (CALL) или вызов функции (имяфункции). По завершении выполнения вызванной функции или подпрограммы выполнение возобновляется со следующей строки. Если же сравнение отрицательно, выполнение программы продолжается.

Конструкция IF...THEN позволяет выполнять одну или несколько инструкций по условию. Инструкции, содержащиеся между ключевыми словами THEN и ENDIF, выполняются, если сравнение между **имяпер** и **значение** положительно.

Конструкция IF...THEN...ELSE позволяет определить два блока инструкций, из которых только один будет выполняться. Если сравнение между **имяпер** и **значением** положительно, выполняются инструкции, содержащиеся между ключевыми словами THEN и ELSE, если отрицательно - выполняются инструкции, содержащиеся между ключевыми словами ELSE и ENDIF. В обоих случаях выполнение затем продолжается инструкцией, следующей за ENDIF.

Примечание

IFVALUE сохраняется для обеспечения совместимости с предыдущими версиями GPL.

IFACC

Синтаксис

IFACC ось, **GOTO** ярлык
IFACC ось, **CALL** имяподпрограммы
IFACC ось, имяфункции

Аргументы

ось имя устройства типа оси
ярлык имя ярлыка, к которому выполнять переход
имяподпрограммы имя подпрограммы
имяфункции имя функции

Описание

Тестирует, находится ли ось, определенная переменной **ось**, в состоянии ускорения. Если это условие выполняется, выполняет переход к **ярлыку** или вызывает **имяподпрограммы** или **имяфункции**.

IFAND

Синтаксис

IFAND операнд1, операнд2, проверочноезначение, **GOTO** ярлык
IFAND операнд1, операнд2, проверочноезначение, **CALL** имяподпрограммы
IFAND операнд1, операнд2, проверочноезначение, имяфункции

IFAND операнд1, операнд2, проверочноезначение **THEN**

инструкция
инструкция

...

ENDIF

IFAND операнд1, операнд2, проверочноезначение **THEN**

инструкция
инструкция

...

ELSE

инструкция
инструкция

...

ENDIF

Аргументы

операнд1 константа или переменная или имяустройства
операнд2 константа или переменная или имяустройства
проверочноезначение константа. Значение, используемое для проверки результата операции. Может принимать следующие значения: **TRUE 1, FALSE 0**
ярлык имя ярлыка, к которому выполнять переход
имяподпрограммы имя подпрограммы
имяфункции имя функции

Описание

Выполняются два сравнения: первое - между **операндом1** и **операндом2**, второе - между результатом первого сравнения и **проверочнымзначением**.
Первое сравнение состоит в выполнении бинарного AND между **операндом1** и **операндом2**. Эти два операнда рассматриваются как битовые маски. Если результат бинарного AND имеет по меньшей мере один бит, отличный от 0, результат первого сравнения - TRUE. Затем он сравнивается с **проверочнымзначением**. Если два значения совпадают, выполняется переход к ярлыку или вызов функции или подпрограммы.
Более подробная информация приводится в конструкции [IF-THEN-ELSE](#).

IFBIT

Синтаксис

IFBIT маска, **nbit**, состояние, **GOTO** ярлык
IFBIT маска, **nbit**, состояние, **CALL** имяподпрограммы
IFBIT маска, **nbit**, состояние, имяфункции

IFBIT маска, **nbit**, состояние **THEN**

	инструкция	
	инструкция	
	...	
ENDIF		
IFBIT		маска, nbit , состояние THEN
	инструкция	
	инструкция	
	...	
ELSE		
	инструкция	
	инструкция	
	...	
ENDIF		

Аргументы

маска	целая константа или переменная или имя счетчика или имя порта. Проверяемое значение
nbit	целая константа или переменная или имя счетчика. Номер бита (1÷32)
состояние	заданная константа. Состояние, которое должно проверяться на маске. Допускаются следующие значения: ON бит задан на 1 OFF бит задан на 0
ярлык	ярлык перехода (GOTO)
имя подпрограммы	вызов подпрограммы (CALL)
имя функции	имя функции

Описание

Тест на отдельном бите переданной битовой **маски**. Аргумент **маска** должен соответствовать целому значению максимум с 32 бит. Значение, которое должно задаваться переменной **nbit** для определения проверяемого бита, находится в пределах от 1 до 32. Если выполняется условие, указанное в **состояние**, выполняет переход к **ярлыку** или вызывает **имя подпрограммы** или **имя функции**. Более подробная информация приводится в конструкции [IF-THEN-ELSE](#).

IFBLACKBOX**Синтаксис**

IFBLACKBOX	GOTO ярлык
IFBLACKBOX	CALL имя подпрограммы
IFBLACKBOX	имя функции

Аргументы

ярлык	имя ярлыка, к которому выполнять переход
имя подпрограммы	имя подпрограммы
имя функции	имя функции

Описание

При активной регистрации выполняет переход к **ярлыку** или вызывает **имя подпрограммы** или **имя функции**. См. также [STARTBLACKBOX](#), [PAUSEBLACKBOX](#) и [ENDBLACKBOX](#).

IFCHANGEVEL**Синтаксис**

IFCHANGEVEL	ось [, состояние], GOTO ярлык
IFCHANGEVEL	ось [, состояние], CALL имя подпрограммы
IFCHANGEVEL	ось [, состояние], имя функции

Аргументы

ось	имя устройства типа оси
состояние	тип изменения. Допускаются следующие значения: POSITIVE , NEGATIVE
ярлык	имя ярлыка, к которому выполнять переход
имя подпрограммы	имя подпрограммы
имя функции	имя функции

Описание

Проверяет, произошло ли изменение скорости оси. Если ось, определенная переменной **ось**, выполняет изменение скорости при смещении, происходит переход на **ярлык** или вызов **имя подпрограммы** или **имя функции**.

Параметр **состояние** указывает, увеличилась ли скорость (POSITIVE) или уменьшилась (NEGATIVE).

IFCOUNTER

Синтаксис

```

IFCOUNTER имясчетчика оператор сравнения значение, GOTO ярлык
IFCOUNTER имясчетчика оператор сравнения значение, CALL
имяподпрограммы
IFCOUNTER имясчетчика оператор сравнения значение, имяфункции

IFCOUNTER имяпер, оператор сравнения, значение THEN
инструкция
инструкция
...
ENDIF

IFCOUNTER имяпер, оператор сравнения, значение THEN
инструкция
инструкция
...
ELSE
инструкция
инструкция
...
ENDIF

```

Аргументы

имясчетчика	имя счетчика
оператор сравнения	для выполнения сравнения могут использоваться следующие символы: < (меньше) = (равно) > (больше) =< (меньше или равно) >= (больше или равно) <> (не равно)
значение	константа или переменная или имясчетчика
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Эта инструкция выполняет тест счетчика.

Если содержимое счетчика, определенного в переменной **имясчетчика**, определяет условие, указанное **оператором сравнения** со значением, выраженным переменной **значение**, выполняется переход к ярлыку, указанному **ярлыком**, или вызов подпрограммы, определенной **имяподпрограммы** или функции, определенной **имяфункции**.

Более подробная информация приводится в конструкции [IF-THEN-ELSE](#).

IFDEC

Синтаксис

```

IFDEC ось, GOTO ярлык
IFDEC ось, CALL имяподпрограммы
IFDEC ось, имяфункции

```

Аргументы

ось	имя устройства типа оси
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Тестирует, находится ли ось, определенная переменной **ось**, в состоянии замедления.

Если это условие выполняется, выполняет переход к **ярлыку** или вызывает **имяподпрограммы** или **имяфункции**.

IFDIR

Синтаксис

```

IFDIR ось, направление, GOTO ярлык
IFDIR ось, направление, CALL имяподпрограммы

```

IFDIR		ось, направление, имяфункции
IFDIR		ось, направление THEN
	инструкция	
	инструкция	
	...	
ENDIF		
IFDIR		ось, направление THEN
	инструкция	
	инструкция	
	...	
ELSE		
	инструкция	
	инструкция	
	...	
ENDIF		

Аргументы

ось	имя устройства типа оси
направление	направление оси. Допускаются следующие значения: POSITIVE положительное направление оси NEGATIVE отрицательное направление оси
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
ы	
имяфункции	имя функции

Описание

Тест текущего направления оси.

Если **ось** перемещается в направлении, определенном переменной **направление**, выполняется переход к **ярлыку** или вьзов **имяподпрограммы** или **имяфункции**.

Более подробная информация приводится в конструкции [IF-THEN-ELSE](#).

IFERRAN**Синтаксис**

IFERRAN		ось, оператор сравнения, значение, GOTO ярлык
IFERRAN		ось, оператор сравнения, значение, CALL имяподпрограммы
IFERRAN		ось, оператор сравнения, значение, имяфункции
IFERRAN		ось, оператор сравнения, значение THEN
	инструкция	
	инструкция	
	...	
ENDIF		
IFERRAN		ось, оператор сравнения, значение THEN
	инструкция	
	инструкция	
	...	
ELSE		
	инструкция	
	инструкция	
	...	
ENDIF		

Аргументы

ось	имя устройства типа оси
оператор сравнения	для выполнения сравнения могут использоваться следующие символы: < (меньше) = (равно) > (больше) =< (меньше или равно) >= (больше или равно) <> (не равно)
значение	константа или переменная или имясчетчика
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
ы	
имяфункции	имя функции

Описание

Тест значения погрешности преследования (погрешность контура) оси, определенной в переменной **ось**.

Если погрешность контура **оси** определяет условие, выраженное **оператором сравнения** со значением, выраженным **значением**, выполняется переход к **ярлыку** или вызов **имяподпрограммы** или **имяфункции**.

Более подробная информация приводится в конструкции [IF-THEN-ELSE](#).

IFERROR**Синтаксис**

IFERROR	число, IDпозиц, GOTO ярлык
IFERROR	число, IDпозиц, CALL ярлык
IFERROR	число, IDпозиц, имяфункции
IFERROR	имяустройства, состояние, IDпозиц, GOTO ярлык
IFERROR	имяустройства, состояние, IDпозиц, CALL ярлык
IFERROR	имяустройства, состояние, IDпозиц, имяфункции

Аргументы

число	DEFMSG или константа или переменная integer
имяустройства	имя устройства
состояние	заданная константа. Может принимать следующие значения: ON, OFF
IDпозиц	константа или переменная. Цифровое значение, используемое в обзорных экранах.
ярлык	имя ярлыка, к которому выполнять переход
имяфункции	имя функции

Описание

Тест погрешности активного цикла.

Если ошибка цикла, обозначаемая **число** и **IDпозиц** или **имяустройства, состояние** и **IDпозиц** активна, выполняется переход к **ярлыку** или вызов функции **имяфункции**,

Параметр **номер** может обозначать ошибку цикла модуля (т. е. целое числовое значение) или группы (в этом случае используется DEFMSG).

Параметр **имяустройства** - это имя устройства, а параметр **состояние** отражает состояние ON/OFF, в котором должно находиться устройство в момент генерирования ошибки.

Параметр **IDпозиц** - это дополнительный параметр, который указывает цифровое значение, используемое на обзорных экранах для сортировки ошибок цикла в различные ячейки. Должен соответствовать значению, указанному изготовителем обзорных экранов для этой конкретной ячейки отображения. Если не требуется указывать конкретную ячейку, присваивается заданная константа NOPLACE. Диапазон задаваемых значений составляет от 0 (NOPLACE) до 1023.

Если эта инструкция используется без подключения управления сигналами тревоги состояния, генерируется ошибка системы.

См. также инструкцию [ERROR](#).

IFFLAG**Синтаксис**

IFFLAG	имяфлажка, состояние, GOTO ярлык
IFFLAG	имяфлажка, состояние, CALL имяподпрограммы
IFFLAG	имяфлажка, состояние, имяфункции

IFFLAG	имяфлажка, состояние THEN
---------------	----------------------------------

**инструкция
инструкция**

...

ENDIF

IFFLAG	имяфлажка, состояние THEN
---------------	----------------------------------

**инструкция
инструкция**

...

ELSE

**инструкция
инструкция**

...

ENDIF

Аргументы

имяфлажка	имя флагового устройства
------------------	--------------------------

Если сообщение, определенное **число** и **IDпозиц**, активно, выполняется переход на **ярлык** или вызов функции **имяфункции**,
 Параметр **IDпозиц** - это дополнительный параметр, который указывает цифровое значение, используемое на обзорных экранах для сортировки ошибок цикла в различные ячейки. Должен соответствовать значению, указанному изготовителем обзорных экранов для этой конкретной ячейки отображения. Если не требуется указывать конкретную ячейку, присваивается заданная константа NOPLACE. Диапазон задаваемых значений составляет от 0 (NOPLACE) до 1023.
 Если эта инструкция используется без подключения управления сигналами тревоги состояния, генерируется ошибка системы.
 См. также инструкцию [MESSAGE](#).

IFOR

Синтаксис

IFOR		операнд1, операнд2, проверочноезначение, GOTO ярлык
IFOR		операнд1, операнд2, проверочноезначение, CALL имяподпрограммы
IFOR		операнд1, операнд2, проверочноезначение, имяфункции
IFOR		операнд1, операнд2, проверочноезначение THEN
	инструкция	
	инструкция	
	...	
ENDIF		
IFOR		операнд1, операнд2, проверочноезначение THEN
	инструкция	
	инструкция	
	...	
ELSE		
	инструкция	
	инструкция	
	...	
ENDIF		

Аргументы

операнд1	константа или переменная или имяустройства
операнд2	константа или переменная или имяустройства
проверочноезначение	константа. Это значение, используемое для проверки результата операции
ние	Может принимать следующие значения: TRUE 1 FALSE 0
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Выполняются два сравнения: первое - между **операндом1** и **операндом2**, второе - между результатом первого сравнения и **проверочнымзначением**.
 Первое сравнение состоит в выполнении бинарного OR между **операндом1** и **операндом2**. Эти два операнда рассматриваются как битовые маски. Если результат бинарного OR имеет по меньшей мере один бит, отличный от 0, результат первого сравнения - TRUE. Затем он сравнивается с **проверочнымзначением**. Если два значения совпадают, выполняется переход к ярлыку или вызов функции или подпрограммы.
 Более подробная информация приводится в конструкции [IF-THEN-ELSE](#).

IFOUTPUT

Синтаксис

IFOUTPUT		имявыхода, состояние, GOTO ярлык
IFOUTPUT		имявыхода, состояние, CALL имяподпрограммы
IFOUTPUT		имявыхода, состояние, имяфункции
IFOUTPUT		имявыхода, состояние THEN
	инструкция	
	инструкция	
	...	
ENDIF		

IFOUTPUT	инструкция инструкция ...	имявыхода, состояние THEN
ELSE	инструкция инструкция ...	
ENDIF		

Аргументы

имявыхода	имя выхода
состояние	заданная константа. Представляет состояние, которое должно проверяться на выходе. Допускаются следующие значения: ON активно OFF неактивно
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
ы	
имяфункции	имя функции

Описание

Тест логического состояния выхода.

Если выход, определенный переменной **имявыхода**, находится в указанном **состоянии**, выполняется переход на **ярлык** или вызов **имяподпрограммы** или **имяфункции**.

Более подробная информация приводится в конструкции [IF-THEN-ELSE](#).

IFQUOTER**Синтаксис**

IFQUOTER	ось, оператор сравнения, значение, GOTO ярлык
IFQUOTER	ось, оператор сравнения, значение, CALL имяподпрограммы
IFQUOTER	ось, оператор сравнения, значение, имяфункции
IFQUOTER	ось, оператор сравнения, значение THEN
	инструкция инструкция ...
ENDIF	
IFQUOTER	ось, оператор сравнения, значение THEN
	инструкция инструкция ...
ELSE	инструкция инструкция ...
ENDIF	

Аргументы

ось	имя устройства типа оси
оператор сравнения	для выполнения сравнения могут использоваться следующие символы: < (меньше) = (равно) > (больше) =< (меньше или равно) >= (больше или равно) <> (не равно)
значение	константа или переменная или имясчетчика
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Тест реальной координаты, выраженной переменной **ось**.

Если значение переменной **ось** определяет условие, выраженное **оператором сравнения** со значением, выраженным **значением**, выполняется переход к **ярлыку** или вызов **имяподпрограммы** или **имяфункции**.

Более подробная информация приводится в конструкции [IF-THEN-ELSE](#).

IFQUOTET

Синтаксис

IFQUOTET		ось, оператор сравнения, значение, GOTO ярлык
IFQUOTET		ось, оператор сравнения, значение, CALL имяподпрограммы
IFQUOTET		ось, оператор сравнения, значение, имяфункции
IFQUOTET	инструкция	ось, оператор сравнения, значение THEN
	инструкция	
	...	
ENDIF		
IFQUOTET	инструкция	ось, оператор сравнения, значение THEN
	инструкция	
	...	
ELSE	инструкция	
	инструкция	
	...	
ENDIF		

Аргументы

ось	имя устройства типа оси
оператор сравнения	для выполнения сравнения могут использоваться следующие символы: < (меньше) = (равно) > (больше) =< (меньше или равно) >= (больше или равно) <> (не равно)
значение	константа или переменная или имясчетчика
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Тест теоретической координаты, выраженной переменной **ось**.
Если значение переменной **ось** определяет условие, выраженное **оператором сравнения** со значением, выраженным **значение**, выполняется переход к **ярлыку** или вызов **имяподпрограммы** или **имяфункции**.
Более подробная информация приводится в конструкции [IF-THEN-ELSE](#).

IFRECEIVED

Синтаксис

IFRECEIVED	[источник,] идентификатор, GOTO ярлык
IFRECEIVED	[источник,] идентификатор, CALL имяподпрограммы
IFRECEIVED	[источник,] идентификатор, имяфункции

Аргументы

источник	строковая константа переменная.
идентификатор	строковая константа переменная.
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Проверяет, были ли удовлетворена инструкция [RECEIVE](#).
Если была удовлетворена какая-либо определенная предыдущая RECEIVE, выполняется переход к **ярлыку** или вызов **имяподпрограммы** или **имяфункции**.
См. также инструкции [RECEIVE](#), [WAITRECEIVE](#), [SEND](#).

IFREG

Синтаксис

IFREG	ось, GOTO ярлык
IFREG	ось, CALL имяподпрограммы
IFREG	ось, имяфункции

Аргументы

инструкция
...
ENDIF

Аргументы

строка1	строковая переменная. Это первая строка ASCII
оператор сравнения	для выполнения сравнения между строками могут использоваться следующие символы: < (меньше) = (равно) > (больше) =< (меньше или равно) >= (больше или равно) <> (не равно)
строка2	строковая переменная. Это вторая строка ASCII
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Тест на строках ASCII.
Если строка, определенная в **строка1**, определяет условие, выраженное **оператором сравнения** со строкой, содержащейся в **строка2**, выполняется переход к **ярлыку** или вызов **имяподпрограммы** или **имяфункции**.
Более подробная информация приводится в конструкции [IF-THEN-ELSE](#).

IFTARGET

Синтаксис

IFTARGET	ось, GOTO ярлык
IFTARGET	ось, CALL имяподпрограммы
IFTARGET	ось, имяфункции

Аргументы

ось	имя устройства типа оси
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Тестирует, достигла ли ось, определенная переменной **ось**, конечной запрограммированной координаты. Ось, даже если она достигла конечной координаты, не обязательно стоит, как правило, она должна компенсировать погрешность контура. Если это условие выполняется, выполняет переход к **ярлыку** или вызывает **имяподпрограммы** или **имяфункции**.
См. также [IFSTILL](#) и [IFWIN](#).

IFTASKHOLD

Синтаксис

IFTASKHOLD	имязадачи, GOTO ярлык
IFTASKHOLD	имязадачи, CALL ярлык
IFTASKHOLD	имязадачи, имяфункции

Аргументы

имязадачи	имя параллельной задачи
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Проверяет, приостановлена ли задача (состояние hold).
Если задача **имязадачи** приостановлена, выполняется переход к **ярлыку** или вызов **имяподпрограммы** или **имяфункции**.

IFTASKRUN

Синтаксис

IFTASKRUN	имязадачи, GOTO ярлык
IFTASKRUN	имязадачи, CALL имяподпрограммы
IFTASKRUN	имязадачи, имяфункции

Аргументы

имязадачи	имя параллельной задачи
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Проверяет, выполняется ли задача.

Если задача, определенная в **имязадачи**, выполняется, выполняется переход к **ярлыку** или вызов **имяподпрограммы** или **имяфункции**.

IFTIMER**Синтаксис**

IFTIMER		имятаймера, оператор сравнения, значение, GOTO ярлык
IFTIMER		имятаймера, оператор сравнения, значение, CALL
IFTIMER		имятаймера, оператор сравнения, значение, имяфункции
IFTIMER	инструкция	имятаймера, оператор сравнения, значение THEN
	инструкция	
	...	
ENDIF		
IFTIMER		имятаймера, оператор сравнения, значение THEN
	инструкция	
	инструкция	
	...	
ELSE		
	инструкция	
	инструкция	
	...	
ENDIF		

Аргументы

имятаймера	имя устройства таймера
оператор сравнения	для выполнения сравнения между строками могут использоваться следующие символы: < (меньше) = (равно) > (больше) =< (меньше или равно) >= (больше или равно) <> (не равно)
значение	константа или переменная или имятаймера. Это значение, на котором должно выполняться сравнение
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Тест таймера.

Если содержимое таймера **имятаймера** определяет условие, выраженное **оператором сравнения** со значением, выраженным **значение**, выполняется переход к **ярлыку** или вызов **имяподпрограммы** или **имяфункции**.

Более подробная информация приводится в конструкции [IF-THEN-ELSE](#).

IFVEL**Синтаксис**

IFVEL		ось, оператор сравнения, значение, GOTO ярлык
IFVEL		ось, оператор сравнения, значение, CALL имяподпрограммы
IFVEL		ось, оператор сравнения, значение, имяфункции
IFVEL	инструкция	ось, оператор сравнения, значение THEN
	инструкция	
	...	
ENDIF		
IFVEL	инструкция	ось, оператор сравнения, значение THEN

```

        инструкция
        ...
ELSE
        инструкция
        инструкция
        ...
ENDIF

```

Аргументы

ось	имя устройства типа оси
оператор сравнения	для выполнения сравнения могут использоваться следующие символы: < (меньше) = (равно) > (больше) =< (меньше или равно) >= (больше или равно) <> (не равно)
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Тест текущей скорости оси.

Если скорость **ось** определяет условие, выраженное **оператором сравнения** со значением, выраженным **значением**, выполняется переход к **ярлыку** или вызов **имяподпрограммы** или **имяфункции**.

Более подробная информация приводится в конструкции [IF-THEN-ELSE](#).

IFWIN

Синтаксис

```

IFWIN ось, GOTO ярлык
IFWIN ось, CALL имяподпрограммы
IFWIN ось, имяфункции

```

Аргументы

ось	имя устройства типа оси
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции

Описание

Тестирует, вошла ли ось, определенная переменной **ось**, в окно прибытия на координату (см. [Условности и терминология](#)).

Если это условие выполняется, выполняет переход к **ярлыку** или вызывает **имяподпрограммы** или **имяфункции**.

См. также [IFTARGET](#) и [IFSTILL](#).

IFXOR

Синтаксис

```

IFXOR операнд1, операнд2, проверочноезначение, GOTO ярлык
IFXOR операнд1, операнд2, проверочноезначение, CALL
      имяподпрограммы
IFXOR операнд1, операнд2, проверочноезначение, имяфункции

IFXOR операнд1, операнд2, проверочноезначение THEN
        инструкция
        инструкция
        ...
ENDIF

IFXOR операнд1, операнд2, проверочноезначение THEN
        инструкция
        инструкция
        ...
ELSE
        инструкция
        инструкция
        ...
ENDIF

```


Пример

```

; пример цикла, который перемещает ось
; между двумя координатами 10 раз
Function Цикл
  Repeat 10
    MovAbs ось, 100
    waitinput switch, ON
    Movabs ось, -100
    waitinput switch, OFF
  EndRep
Fret

```

RET**Синтаксис****RET****Аргументы**

никакого аргумента

Описание

Завершает выполнение подпрограммы с возвратом к инструкции, непосредственно после CALL вызванной.

См. также инструкцию [CALL](#)

Примечание

Эта инструкция вместе с CALL является типичным источником ошибок программирования. Рекомендуется быть внимательными при ее использовании, в частности, рекомендуется размещать подпроцедуры в конце тела функции (после инструкции FRET) для предупреждения случайного выполнения кода подпроцедуры так, как будто она является составной частью основного кода. Результатом этой ситуации, в лучшем случае, является ошибка системы, в других случаях наблюдаются аномальные действия станка, причину которых установить бывает сложно.

SELECT**Синтаксис**

```

SELECT имяпер
  CASE значение
    инструкция
  CASE значение1 TO значение2
    инструкция
  CASE IS < => значение
    инструкция
  CASE ELSE
    инструкция
ENDSELECT

```

инструкция должна быть заменена одним из следующих значений:

```

GOTO ярлык
CALL имяподпрограммы
[FCALL] имяфункции [параметр1,...параметрN]
[EXPR] переменная =
[EXPR] устройство = выражение

```

Аргументы

имяпер	константа или переменная integer или имясчетчика
значение, значение1, значение2	константы integer
ярлык	имя ярлыка, к которому выполнять переход
имяподпрограммы	имя подпрограммы
имяфункции	имя функции
параметр1...параметрN	параметр, переданный вызываемой функции
переменная	имя переменной
устройство	имя устройства
выражение	набор операторов

имяфункции

имя функции

Описание

Тест и прием сообщения

Первая TESTMAIL на почтовом ящике **mail** создает почтовый ящик.

Если сообщение присутствует в почтовом ящике **mail**, данные, направленные вместе с сообщением, записываются в переменные **имяпер** (1÷20) только если они указаны, или в строку матрицы, указанную при помощи **матрица[строка]**. Кроме того, выполняется переход на **ярлык** или вызов **имяподпрограммы** или **имяфункции**.

При выполнении проверяется соответствие переданных данных и данных, ожидаемых инструкцией.

См. также инструкции [SENDMAIL](#), [WAITMAIL](#) и [ENDMAIL](#).

10.3.13 Разные**CLEARERRORS****Синтаксис****CLEARERRORS****[IDпозиц]****Аргументы****IDпозиц**

константа или переменная. Это цифровое значение, используемое в обзорных экранах

Описание

Заставляет ПК-супервизор удалить все ошибки цикла, относящиеся к модулю, выполняющему инструкцию, направленные ранее инструкцией ERROR. Параметр **IDпозиц** - это дополнительный параметр, который указывает цифровое значение, используемое на обзорных экранах для сортировки ошибок цикла в различные ячейки. Должен соответствовать значению, указанному изготовителем обзорных экранов для этой конкретной ячейки отображения. Albatros использует этот идентификационный номер для управления ошибками цикла в разных очередях. Для каждого IDпозиц создается очередь. Диапазон задаваемых значений составляет от 0 (NOPLACE) до 1023. Если параметр **IDпозиц** не указан, выполняется стирание всех ошибок цикла как в очереди по умолчанию, так и в других возможных очередях.

См. также инструкции [ERROR](#) и [DELERROR](#).

CLEARMESSAGES**Синтаксис****CLEARMESSAGES****[IDпозиц]****Аргументы****IDпозиц**

константа или переменная. Это цифровое значение, используемое в обзорных экранах

Описание

Заставляет ПК-супервизор удалить все сообщения, относящиеся к модулю, выполняющему инструкцию, направленные ранее инструкцией MESSAGE. Параметр **IDпозиц** - это дополнительный параметр, который указывает цифровое значение, используемое на обзорных экранах для сортировки сообщений в различные ячейки. Должен соответствовать значению, указанному изготовителем обзорных экранов для этой конкретной ячейки отображения. Albatros использует этот идентификационный номер для управления сообщениями в разных очередях. Для каждого IDпозиц создается очередь. Диапазон задаваемых значений составляет от 0 (NOPLACE) до 1023. Если параметр **IDпозиц** не указан, выполняется стирание всех сообщений как в очереди по умолчанию, так и в других возможных очередях.

См. также инструкции [MESSAGE](#) и [DELMESSAGE](#).

DEFMSG**Синтаксис****DEFMSG**

ярлык [, префиксязыка1], "строкасообщения", ... , [, префиксязыкаN, "строкасообщения"]

Аргументы

ярлык
префиксязыка
строкасообщения

мнемоническое имя отображаемого сообщения
заданная константа. Язык, на котором написано сообщение
отображаемое сообщение. Должно быть заключено в двойные кавычки
(")

Описание

Присваивает **ярлык** сообщению. Инструкция DEFMSG должна объявляться до реализации функций. Определение сообщения может использоваться только внутри файла (или группы), в которой оно объявлено. Можно вводить сообщения на различных языках, используя заданную константу **префиксязыка** (список языковых префиксов см. в главе «[Импорт сообщений](#)»). В этом случае инструкция MESSAGE отображает сообщение на языке, соответствующем языку, используемому в настоящее время для Albatros. Сообщение, с которым не связан какой-либо префикс, используется, когда используемый в данный момент язык не соответствует ни одному из существующих префиксов. Ярлыки различных языков записываются или все на одной строке, или на нескольких строках, переходя на новую строку помощью символа "_" с пробелом перед ним. Инструкция DEFMSG может передаваться как параметр функции. При этом получающая ее функция может использовать ее как один из трех аргументов ERROR или MESSAGE. (См. пример 2). См. также инструкции [MESSAGE](#), [DELMESSAGE](#), [ERROR](#), [DELError](#).

Пример 1:

```
; задание строки сообщения ярлыку
; без выбора языка
DEFMSG MSG_GRU_1 "Сообщение группы 1"

; использование определения
MESSAGE MSG_GRU_1 ; отображает: "Сообщение группы 1"

; задание строки сообщения ярлыку
; с выбором языка
DEFMSG MSG_GRU_1 ITA "Messaggio gruppo 1"
ENG "Message group 1"

; использование определения, когда язык
; Albatros - ENG
MESSAGE MSG_GRU_1 ; отображает: "Message group 1"
```

Пример 2:

```
; В группе:
DEFMSG MSG_TEST "Ошибка выполнения"

FUNCTION ВызовТекста
    Текст MSG_TEST
FRET

В библиотеке:
DEFMSG MSG_BASE "Сигнализация ошибки: $1"
...
FUNCTION Test Public
    PARAM код AS integer
    ERROR MSG_BASE NOPLACE NOSTORE код
FRET
; Отображенная ошибка цикла: Сигнализация ошибки: Ошибка выполнения
```

DELAY**Синтаксис**

DELAY значение

Аргументы

значение константа или переменная. Продолжительность задержки, выраженная в секундах.

Описание

Ожидает, пока пройдет время, указанное **значением**. По истечении временного интервала выполняется следующая инструкция. Минимальное задаваемое значение - 4 мс (0,004 секунды).

состояние заданная константа. Может принимать следующие значения: **ON, OFF**
log заданная числовая константа или переменная integer. Может принимать следующие значения:
STORE ошибка, сохраненная в файле
NOSTORE ошибка, не сохраненная в файле
arg1, ..., arg3 константа или устройство или переменная.

Описание

Генерирует ошибку цикла. Ошибка обозначается параметром **номер** или именем устройства. Параметр **номер** может обозначать ошибку цикла модуля (т. е. целое числовое значение) или группы (в этом случае используется DEFMSG). При указании имени устройства вместо числа на ПК передается тип и логический адрес устройства. Ошибка цикла направляется на ПК-супервизор и отображается на строке ошибок Albatros.

Параметр **IDпозиц** используется на обзорных экранах для сортировки ошибок цикла в различные ячейки. Должен соответствовать значению, указанному изготовителем обзорных экранов для этой конкретной ячейки отображения. Albatros использует этот идентификационный номер для управления ошибками цикла в разных очередях. Для каждого IDпозиц создается очередь. Если IDпозиц не указывается или если используется заданная константа NOPLACE, ошибка цикла направляется в очередь по умолчанию (IDпозиц=0). Диапазон задаваемых значений составляет от 0 (NOPLACE) до 1023.

Задание параметра **log** на **STORE** обуславливает регистрацию ошибки цикла в файле отчета с ошибками за текущий месяц. Генерирование повышенного числа ошибок (или их удаление) может нарушить уровень рабочих характеристик удаленных модулей. При этом ПК-супервизор должен управлять всеми направленными ошибками (и их возможным удалением) и это может замедлить отправку важных для управления данных, особенно программ обработки. Дополнительные параметры **arg1, ..., arg3** позволяют определять сообщения об ошибке параметров. В строку определения сообщения об ошибке включаются маркеры, которые при генерировании ошибки будут заменяться значением или именем устройства или переменной, переданной как параметр. В строку вводятся следующие маркеры:

- \$1, ... \$2 заменяемые **именем** устройства или переменной (\$1 соответствует arg1 и т. д.)
- \$(1), ..., \$(3) заменяемые **значением** устройства или переменной

Для параметров arg1, ..., arg3 разрешены следующие типы данных:

- CHAR
- INTEGER
- FLOAT
- DOUBLE (которое автоматически преобразуется в FLOAT)
- номер сообщения (или ярлык DEFMSG)
- устройство
- глобальная или локальная переменная
- параметр функции. В качестве параметра функции можно использовать также ярлык, определенный инструкцией **DEFMSG**

В качестве параметров не могут использоваться строки, матрицы и вектора (отдельные элементы вектора или матрицы допускаются). Для локальных переменных можно декодировать только значение, но не имя.

В конце удаления сообщения инструкцией **DELERROR** параметры arg1, ...arg3 игнорируются.

Имеется два режима управления ошибками цикла, предусмотренные производителем станка:

Сигналы тревоги, управление которыми осуществляется как сигналами: направляются все ошибки цикла. Albatros сохраняет очередь последних 100 ошибок из указанной очереди и последних 100 ошибок очереди по умолчанию.

Сигналы тревоги, управление которыми осуществляется как состояниями: ошибка рассматривается как активная или неактивная. Если ошибка активна, любая дальнейшая отправка одного и того же сообщения (инструкцией **MESSAGE**) игнорируется.

См. также инструкции [DELERROR](#), [CLEARERRORS](#).

Пример 1

```
DEFMSG ERR_TOOL "Инструмент не присутствует"
DEFMSG ERR_TOOL_P "Загрузить инструмент $(1) в положение $(2)"

; тег для обзорных экранов
CONST TOOLCHANGE = 5

; ошибка отображается в строке ошибок или
; в неотмеченных ячейках обзорного экрана.
ERROR ERR_TOOL

; ошибка сохраняется в отчетах и отображается в ячейках
```

```

; обзорного экрана, отмеченных кодом 5
ERROR      ERR_TOOL, TOOLCHANGE, STORE

; ошибка сохраняется в отчетах, но не ассоциируется
; с ячейками обзорного экрана
ERROR      ERR_TOOL, NOPLACE, STORE

; ошибка с параметрами
ERROR      ERR_TOOL_P, NOPLACE, NOSTORE, МХИНСТРУМЕНТЫ[3].Cod, 5

```

Пример 2

```

; определяется в группе
DEFMSG     MSG_ERR_CARICO      "Загрузка инструмента не выполнена"

```

```

Function ПоказатьСообщение
  MsgTool MSG_ERR_CARICO МХИНСТРУМЕНТЫ[3].Cod
fret

```

```

Function ПоказатьОшибку
  ErrTool STORE MSG_ERR_CARICO МХИНСТРУМЕНТЫ[3].Cod
fret

```

```

; определяется в библиотеке
DEFMSG     MSG_ERR_TOOL      "Ошибка инструмента: $1 $(2)"

```

```

Function MsgTool Public
PARAM parameter1 as integer
PARAM parameter2 as integer

  MESSAGE MSG_ERR_TOOL NOPLACE parameter1 parameter2
fret

```

```

Function ErrTool Public
PARAM log as integer
PARAM argument1 as integer
PARAM argument2 as integer

  ERROR MSG_ERR_TOOL NOPLACE log argument1 argument2
fret

```

IFDEF/ELSEDEF/ENDIF*Синтаксис*

IFDEF	инструкция	константа
	...	
ENDIF		
IFDEF	инструкция	константа, оператор сравнения, значение
	...	
ENDIF		
IFDEF	инструкция	EXIST, имя группы
	...	
ENDIF		
IFDEF	инструкция	LINKED, имя устройства
	...	
ENDIF		
IFDEF		UNLINKED, имя устройства

	инструкция	
	...	
ENDDF		
IFDEF	инструкция	константа, оператор сравнения, значение
	...	
ELSEDEF	инструкция	
	...	
ENDDF		

Аргументы

константа	постоянная типа integer, char, double, строки
оператор сравнения	для выполнения сравнения могут использоваться следующие символы: < (меньше) = (равно) > (больше) =< (меньше или равно) >= (больше или равно) <> (не равно)
значение	константа или имя устройства
имя группы	строковая константа или имя группы
имя устройства	строковая константа или имя устройства

Описание

Условная компиляция позволяет проверить, какие части файла функции GPL должны компилироваться и затем выполняться. Компилятор проверяет, что удовлетворяется условие, требуемое в качестве аргумента инструкции IFDEF. В этом случае код, содержащийся между инструкцией IFDEF и инструкцией ENDDF или ELSEDEF, компилируется. Если присутствует инструкция ELSEDEF и условие не удовлетворяется, компилируется код, содержащийся между инструкцией ELSEDEF и инструкцией ENDDF.

Условие компиляции может выражаться различными способами:

- после инструкции IFDEF указывается имя [константы](#). В этом случае условие удовлетворяется при наличии глобальной константы или текущей группы с указанным именем.
- после инструкции IFDEF указывается отношение между двумя операторами и одним операндом. Первый операнд должен быть константой. В этом случае условие удовлетворяется, если отношение верно (напр., MAX_TOOLS = 100).
- после команды IFDEF указывается ключевое слово EXIST или NOTEXIST, за которым следует имя группы станка или строка с именем группы станка или именем библиотеки. В этом случае условие удовлетворяется, если в Конфигурации станка присутствует и не присутствует группа с таким же именем.
- после инструкции IFDEF указывается пароль LINKED или UNLINKED, за которым следует имя устройства. В этом случае это условие выполняется, если устройство подключено (LINKED) или не подключено (UNLINKED) в виртуальном-физическом пространстве. Имя устройства может выражаться в следующей форме: Имя_Группы.Имя_Подгруппы.Имя_Устройства или Имя_Группы.Имя_Устройства или Имя_Подгруппы.Имя_Устройства или Имя_Устройства. Если устройство не существует в конфигурации, оно считается не подключенным.

Можно встроить несколько инструкций IFDEF, учитывая, что каждой инструкции IFDEF должна соответствовать инструкция ENDDF.

Пример 1

```

; выполнение кода GPL меняется в зависимости
; от наличия в станке группы ФРЕЗА
Const ГруппаФреза= "Фреза"
IFDEF Exist ГруппаФреза
инструкция
инструкция
ELSEDEF
инструкция
инструкция
ENDDF

```

Пример 2

```

; выполнение кода GPL меняется
; в зависимости от модуля
IFDEF _ID_MODULE = 1 ; компилирует инструкции для модуля 1
инструкция
инструкция
ELSEDEF ; компилирует инструкции для других модулей

```

```

инструкция
инструкция
ENDDDEF

; компилирует код для версии 3.2.0 Albatros

IFDEF _VER_MAJOR = 3
  IFDEF _VER_MINOR = 2
    IFDEF _VER_REVISION = 0
      инструкция
      инструкция
    ENDDDEF
  ENDDDEF
ENDDDEF

; компилирует код для версии
; пакета обновления 10 Albatros
IFDEF _VER_SP = "Service Pack 10"
  инструкция
ENDDDEF

; компилирует код только если система
; сконфигурирована для удаленного модуля
IFDEF _REMOTE_MODULE = 1 ; 1 = удаленный модуль, в противном
случае 0 = локальный модуль
  инструкция
ENDDDEF

; компилирует код для версии
; 2.4 пакета обновления 10 Albatros
IFDEF _VER_FULL = $0002040AH
  инструкция
ENDDDEF

```

Пример 3

```

; выполнение кода GPL меняется
; в зависимости от того, подключено ли устройство в
виртуальном-физическом пространстве
IFDEF LINKED out1 ; если Out1 подключен, код выполняется
  инструкция
  инструкция
  инструкция
ENDIF

```

MESSAGE**Синтаксис**

MESSAGE **число** [, **IDпозиц** [, **arg1**, ..., **arg3**]]

Аргументы

число	DEFMSG или константа или переменная
IDпозиц	константа или переменная. Цифровое значение, используемое в обзорных экранах
arg1, ..., arg3	константа или устройство или переменная.

Описание

Генерирует сообщение для оператора. Сообщение обозначается параметром **номер**. Параметр **номер** может обозначать сообщение модуля (т. е. целое числовое значение) или группы (в этом случае используется DEFMSG). Дополнительно может передаваться аргумент, представленный **IDпозиц**. Указывает, в каком окне обзорного экрана отображать сообщение. Должен соответствовать значению, указанному изготовителем обзорных экранов для этой конкретной ячейки отображения. Albatros использует этот идентификационный номер для управления сообщениями в разных очередях. Для каждого IDпозиц создается очередь. Если IDпозиц не указывается, сообщение направляется в очередь по умолчанию (IDпозиц=0). Диапазон задаваемых значений составляет от 0 (NOPLACE) до 1023.

Дополнительные параметры **arg1, ..., arg3** позволяют определять параметрические сообщения. В строку определения сообщения включаются маркеры, которые при генерировании сообщения будут заменяться значением или именем устройства или переменной, переданной как параметр. В строку вводятся следующие маркеры:

- \$1, ... \$2 заменяемые **именем** устройства или переменной (\$1 соответствует arg1 и т. д.)
- \$(1), ..., \$(3) заменяемые **значением** устройства или переменной

Для параметров arg1, ..., arg3 разрешены следующие типы данных:

- CHAR
- INTEGER
- FLOAT
- DOUBLE (которое автоматически преобразуется в FLOAT)
- номер сообщения (или ярлык DEFMSG)
- устройство
- глобальная или локальная переменная
- параметр функции. В качестве параметра функции можно использовать также ярлык, определенный инструкцией [DEFMSG](#)

Имеется два режима управления сообщениями, предусмотренные производителем станка:

Сообщения, управление которыми осуществляется как сигналами: направляются все сообщения.

Albatros сохраняет очередь последних 100 сообщений из указанной очереди и последних 100 сообщений очереди по умолчанию. Когда очередь сообщений будет заполнена, запись осуществляется по самому старому сообщению. Если предыдущее сообщение очереди идентично тому, которое направляется, сообщение не направляется (одинаковая задача, одинаковый номер, одинаковый аргумент).

Сообщения, управление которыми осуществляется как состояниями: сообщение рассматривается как активное или неактивное. При подключении любая дальнейшая отправка одного и того же сообщения (инструкцией MESSAGE) игнорируется.

В качестве параметров не могут использоваться строки, матрицы и вектора (отдельные элементы вектора или матрицы допускаются). Для локальных переменных можно декодировать только значение, но не имя.

В конце удаления сообщения инструкцией DELMESSAGE параметры arg1, ...arg3 игнорируются.

См. также инструкции [DELMESSAGE](#) и [CLEARMESSAGES](#).

Пример 1

```
DEFMSG      MSG_TOOL      "Выполнить смену инструмента"
DEFMSG      MSG_TOOL_P    "Загружен инструмент № $(1)"

;тег для обзорных экранов
CONST      TOOLCHANGE = 7

;сообщение отображается в строке ошибок или
;в неотмеченных ячейках обзорного экрана.
MESSAGE    MSG_TOOL

;сообщение отображается в строке ошибок или
;в ячейках обзорного экрана, отмеченных кодом 7
MESSAGE    MSG_TOOL, TOOLCHANGE

; сообщение с параметрами
MESSAGE    MSG_TOOL_P, NOPLACE, Mхинструментов[3].Cod
```

Пример 2

```
; определяется в группе

DEFMSG      MSG_ЗАГРУЗКА      "Загрузка выполнена"

Function ПоказатьСообщение
  MsgTool MSG_ERR_ЗАГРУЗКА Mхинструментов[3].Cod
fret

; определяется в библиотеке
DEFMSG      MSG_TOOL      "инструмент: $(1) $2"

Function MsgTool public
PARAM parameter1 as integer
```

```
PARAM parameter2 as integer
MESSAGE MSG_TOOL NOPLACE parameter1 parameter2
fret
```

SYSFAULT

Синтаксис
SYSFAULT

Аргументы
никакого аргумента

Описание
Отключает сигнал SYSOK.
Этот сигнал отключается, указывая, что станок больше не находится в безопасном состоянии (напр., больше нет исполняемых задач GPL, управляющих аварийными ситуациями).
См. также инструкцию [SYSOK](#).

SYSOK

Синтаксис
SYSOK [имявыхода1 [, ... имявыхода8]]

Аргументы
имявыхода1 [...имявыхода8] имя устройства типа цифрового выхода

Описание
Указывает ЧПУ, какие выходы подключены к предохранительным сетям станка (может быть один выход, подключенный к предохранительному реле, контролирующему силовое питание станка). Выходы подключаются после того, как ЧПУ завершит инициализацию станка и подключит все задачи управления аварийными ситуациями. Таким образом, станок может считаться находящимся в условиях безопасности. Может быть определено в общей сложности до 8 цифровых выходов. На каждом удаленном блоке может быть подключен только один выход. Список выходов, объявленных в первом применении инструкции SYSOK, может меняться при возможных последующих вызовах sysok **только** после инициализации блока управления.
При выполнении команды без параметров восстанавливается сигнал SYSOK.

См. также инструкцию [SYSFAULT](#).

Примечание
Инструкция SYSOK может подключаться только:

- на всех удаленных модулях GreenBus v3.0 с цифровыми выходами.
- на удаленных модулях GreenBus v4.0 типа TRS-IO.
- на удаленных модулях TRS-CAT, только на базе, не на расширениях, имеющих встроенное ПО версии 1.2 или выше (ред. 1.00 удаленного модуля).

TYPEOF

Синтаксис
TYPEOF имя, результат

Аргументы
имя имя устройства, константа, имяфункции, переменная, вектор, матрица или строка матрицы
результат переменная integer. Тип первого аргумента

Описание
Возвращает в переменной **результат** тип аргумента **имя**.

WATCHDOG

Синтаксис
WATCHDOG состояние

Аргументы
состояние заданная константа. Может принимать следующие значения: **ON**, **OFF**

Описание

Эта инструкция активирует использование сторожевой схемы, подключенной к аппаратному компоненту TMSWD. Позволяет выявлять ситуации ошибки, которые могут возникнуть при выполнении кода GPL.

Первый раз, что выполняется параметром **состояние** на ВКЛ. инструкция подключит использование сторожевой схемы. Для выполнения обновления счетчика платы, необходимо задать на ВКЛ. параметр **состояние**.

Если обновление не выполняется, срабатывает сторожевая схема и модуль TMSWD отключает предохранительный выход станка.

Для завершения использования сторожевой схемы необходимо задать на ВыКЛ. параметр **состояние**.

Эта инструкция может использоваться только с платами TMSbus+, TMSCan+ и TMSCombo+ с версией FPGA 2.0 или выше и с установленным аппаратным модулем TMSWD.

Пример

Function TestWatchDog автозапуск

watchdog ON ; активирует управление сторожевой схемой

loop:

watchdog ON ; выполняет обновление счетчика платы

goto loop

fret

10.3.14 MECHATROLINK-II**MECCOMMAND****Синтаксис**

MECCOMMAND ось, команда, параметры, ответ, ошибка

Аргументы

ось	имя устройства типа цифровой оси.
команда	константа integer.
параметры	вектор integer.
ответ	вектор integer.
ошибка	переменная integer. Код ошибки

Описание

Направляет на привод указанной **оси команду** и ожидает его ответ. Данные, необходимые для выполнения команды, вводятся в вектор **параметры**, а данные, возвращенные после выполнения инструкции, записываются в память в векторе **ответ**. Векторы **параметры** и **ответ** должны иметь тот же размер и число элементов максимум 14. Рассматриваемое значение - младший байт отдельных целых чисел. Параметр **ошибка** содержит коды возможных ошибок, генерированных в ходе операции. Управление кодами ошибок должно выполняться Gpl как для ошибок цикла.

Возвращаются следующие коды **ошибки**:

Код ошибки	Сообщение
-40	Команда, не разрешенная в текущих условиях функционирования
-41	Ошибка таймаута при выполнении команды MECHATROLINK-II
-44	Ошибка таймаута при выполнении подкоманды MECHATROLINK-II
-45	Ошибка подключения привода

По значениям, которые должны присваиваться параметрам **команда**, **параметры**, **ответ** и **ошибка**, см. официальную документацию Yaskawa MECHATROLINK-II. В ней значения, присваиваемые для команды, описаны начиная с индекса 2 по индекс 15. Значения, присваиваемые для подкоманд, описаны начиная с индекса 18 по индекс 32.

Значения **команды** могут меняться, как указано ниже, по сравнению с официальной документацией.

Команды можно разделить на:

- команда. Имеют код от 0x00 до 0xFF. По мотивам безопасности они выполняются только с отключенным сервоприводом оси.
- сервопривод. Команды, понимаемые как подкоманды, должны суммировать к задокументированному значению код 0x100. Например, команда NOP имеет задокументированный код 0x00, понимаемый как подкоманда он - 0x100.

- процедура. Команды, понимаемые как процедуры, имеют код со значением начиная с 0x200. На данный момент предусматриваются следующие процедуры:
 - \$201H процедура подключения параметров в автономном режиме (для использования при отключенной оси)

Эта функция может использоваться только с платами AlbMech, DualMech, DualMech Mono. За дополнительной информацией по использованию этой функции обращайтесь в ТРА.

Примечание

Эта инструкция действует на поведение цифровых осей, поэтому должна использоваться в контролируемом контексте.

MECGETPARAM

Синтаксис

MECGETPARAM ось, параметр, размер, данные, ошибка

Аргументы

ось	имя устройства типа цифровой оси.
параметр	константа или переменная integer.
размер	константа или переменная integer.
данные	переменная integer.
ошибка	переменная integer. Код ошибки

Описание

Считывает параметр привода указанной **оси** и записывает его в переменную **данные**. Параметр **ошибка** содержит коды возможных ошибок, генерированных в ходе операции. Управление кодами ошибок должно выполняться Gpl как для ошибок цикла.

Возвращаются следующие коды **ошибки**:

Код ошибки	Сообщение
-40	Команда, не разрешенная в текущих условиях функционирования
-41	Ошибка таймаута при выполнении команды MECHATROLINK-II
-44	Ошибка таймаута при выполнении подкоманды MECHATROLINK-II
-45	Ошибка подключения привода

По значениям, присваиваемым переменным **параметр** и **размер**, см. официальную документацию Yaskawa MECHATROLINK-II.

Эта функция может использоваться только с платами AlbMech, DualMech, DualMech Mono. За дополнительной информацией по использованию этой функции обращайтесь в ТРА.

MECGETSTATUS

Синтаксис

MECGETSTATUS ось, состояние, вход-выход, ошибка

Аргументы

ось	имя устройства типа цифровой оси.
состояние	постоянная или переменная integer.
вход-выход	постоянная или переменная integer.
ошибка	переменная integer. Код ошибки

Описание

Считывает и записывает в память в переменной **состояние** значения STATUS и ALARM, а в переменную **вход-выход** - значение IO_MON для указанной **оси**. По значениям STATUS, ALARM, IO_MON см. официальную документацию Yaskawa MECHATROLINK-II.

Параметр **ошибка** содержит коды ошибок, которые могли быть генерированы при операции. Коды ошибок должны управляться Gpl как ошибки цикла.

Возвращенные коды **ошибки**:

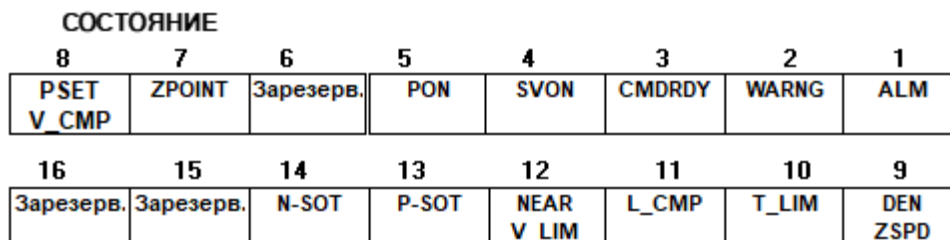
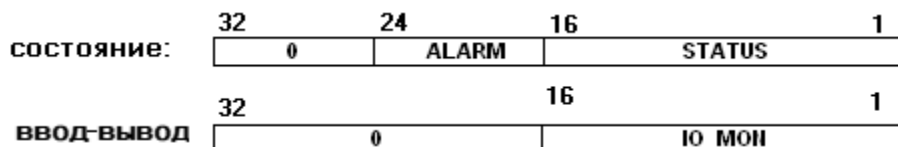
Код ошибки	Сообщение
-40	Команда, не разрешенная в текущих условиях функционирования
-41	Ошибка таймаута при выполнении команды MECHATROLINK-II
-44	Ошибка таймаута при выполнении подкоманды MECHATROLINK-II
-45	Ошибка подключения привода

Определяется серия категорий ошибок. Эти категории представляют более высокое значение полубайта ALARM.

Для выхода из сигналов тревоги, выходящих в одну из следующих категорий, 0x30,0x70,0xD0,0xF0, должна направляться команда CLEAR (0x06). Сигналы тревоги, входящие в одну из следующих

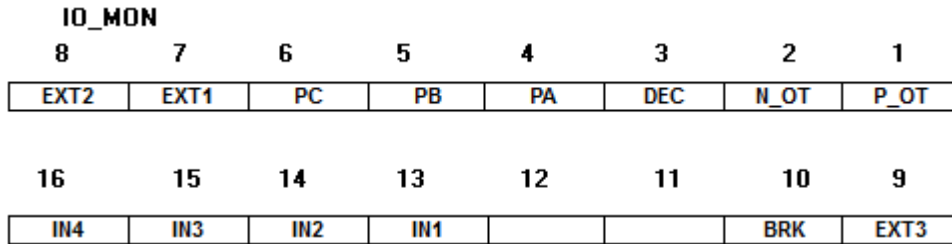
категорий 0x00,0x10,0x40,0xB0, не могут стираться никакой командой. Необходимо устранить проблему, вызвавшую подачу сигнала тревоги, выключить сервопривод и снова включить его.

Структура переменных **СОСТОЯНИЕ** и **ВВОД-ВЫВОД** - это битовая маска, организованная как показано ниже:



Значение битов СОСТОЯНИЕ

Бит	Команда	Физические выводы, которые могут подключаться виртуально-физически
1	ALM (Сигнал тревоги)	Цифровой вход
2	WARNG (Предупреждение)	Цифровой вход
3	CMDRDY (Команда готова)	
4	SVON (Сервопривод ВКЛ.)	Цифровой выход
5	PON (Главное питание ВКЛ.)	Цифровой вход
6	Зарезервировано	
7	ZPOINT (Нулевая точка)	
8	PSET (Полное положение) V_CMP (Соглашение о Скорости)	
9	DEN (Флажок завершения распределения команды) ZSPD (Нулевая Скорость)	
10	T_LIM (Предел момента)	
11	L_CMP (Защелка завершена)	
12	NEAR (Бесконтактный датчик положения) V_LIM (Предел скорости)	
13	P-SOT (Программный предел направления вперед)	
14	N-SOT (Программный предел направления назад)	
15	Зарезервировано	
16	Зарезервировано	



Значение битов IO_MON

Бит	Команда	Физические пин-коды, которые могут подключаться виртуально-физически
1	P_OT (Перебег вперед)	
2	N_OT (Перебег назад)	
3	DEC (Предельный выключатель замедления)	
4	PA (Фаза A)	
5	PB (Фаза B)	
6	PC (Фаза C)	Цифровой вход
7	EXT1 (Вход первой внешней защелки)	Цифровой вход
8	EXT2 (Вход второй внешней защелки)	Цифровой вход
9	EXT3 (Вход третьей внешней защелки)	
10	BRK (Выход тормоза)	
11		
12		
13	IN1 (Универсальный вход 1)	
14	IN2 (Универсальный вход 2)	
15	IN3 (Универсальный вход 3)	
16	IN4 (Универсальный вход 4)	

Эта инструкция используется только с платами AlbMech, DualMech и DualMech Mono. За дополнительной информацией по использованию этой инструкции обращайтесь в ТРА.

MECSETPARAM

Синтаксис

MECSETPARAM

ось, параметр, размер, данные, ошибка

Аргументы

ось
параметр
размер
данные
ошибка

имя устройства типа цифровой оси.
константа или переменная integer.
константа или переменная integer.
переменная integer.
переменная integer. Код ошибки

Описание

Записывает **данные** в **параметр** указанной **оси**.

По значениям, присваиваемым переменным **параметр** и **размер** см. официальную документацию Yaskawa MECHATROLINK-II. Параметр **ошибка** содержит коды возможных ошибок, генерированных в ходе операции. Управление кодами ошибок должно выполняться GpI как для ошибок цикла.

Возвращаются следующие коды **ошибки**:

Код **Сообщение**
ошибки

- 40 Команда, не разрешенная в текущих условиях функционирования
 - 41 Ошибка таймаута при выполнении команды MECHATROLINK-II
 - 44 Ошибка таймаута при выполнении подкоманды MECHATROLINK-II
 - 45 Ошибка подключения привода
- Эта функция может использоваться только с платами AlbMech, DualMech, DualMech Mono. За дополнительной информацией по использованию этой функции обращайтесь в ТРА

Примечание

Эта инструкция действует на поведение цифровых осей, поэтому должна использоваться в контролируемом контексте.
 Для записи данных в энергонезависимую память необходимо использовать инструкцию **MECCOMMAND**

10.3.15 Стандартная полевая шина**AХCONTROL****Синтаксис**

AХCONTROL ось, параметр

Аргументы

ось имя устройства типа оси
параметр переменная или константа целого типа. Задаёт Контрольное слово (ControlWord)

Описание

Задаёт **параметр** ControlWord, в соответствии с активированным режимом, согласно "CiA 402 CANopen device profile".

Таблица определения значений для EtherCAT

CONTROLWORD						
8	7	5	4	3	2	1
fr	oms	eo	qs	ev	so	
16		12	11	10	9	
	ms	r	oms	h		

Бит	Значение	Имя в виртуальном-физическом пространстве
1	so=Включение	CW1
2	ev=Подключить напряжение	EV
3	qs=Быстрый останов	СТОП
4	eo=Активировать работу	SVON
5	oms=Зависит от рабочего режима	CW5
6	oms=Зависит от рабочего режима	CW6
7	oms=Зависит от рабочего режима	CW7
8	fr=Сброс аварии	RESALM
9	h=Приостановка	CW9
10	oms=Зависит от рабочего режима	CW10
11	r=Зарезервировано	CW11
12	ms=Зависит от производителя	CW12
13	ms=Зависит от производителя	CW13

14	ms=Зависит от производителя	CW14
15	ms=Зависит от производителя	CW15
16	ms=Зависит от производителя	CW16

Таблица определения значений для S-CAN

CONTROLWORD

8	7	5	4	3	2	1
fr	oms	eo	qs	ev	so	

Бит	Значение	Имя в виртуальном-физическом пространстве
1	Ten_cmd=команда активации момента. 1:ось в моменте 0:свободная ось	SVON
2	Ien_cmd=команда активации движений. 1:движения активированы 0:ось в фиксаторе момента	ENMOVE
3	Stp_cmd=команда "стоп". 1:команда "стоп" активирована 0:команда "стоп" не активирована	СТОП
4	Alm_rst=состояние аварийных сигналов 1:команда сброса аварийных сигналов	RESALM
5	Ltc_rst:сброс бита 5 параметра StatusWord	CW5
6	oms=зависит от рабочего режима	CW6
7	oms=зависит от рабочего режима	CW7
8	oms=зависит от рабочего режима	CW8

AXSTATUS

Синтаксис

AXSTATUS

ось, значение

Аргументы

ось
значение

имя устройства типа оси
целая переменная.

Описание

Возвращает значение, содержащееся в параметре StatusWord, согласно "CiA 402 CANopen device profile".

Таблица определения значений для EtherCAT

STATUSWORD

8	7	5	4	3	2	1
fr	oms	f	oe	so	rtso	
16		12	11	10	9	
	ms	r	oms	h		

Бит	Значение	Имя в виртуальном-физическом пространстве
1	rtso=Готов к включению	RTSO
2	so=Включено	SW2
3	oe=Работа активирована	OE
4	f=Авария	ALM
5	ve=Подключить напряжение	VE
6	qs=Быстрый останов	QS
7	sod=Включение деактивировано	SOD
8	w=Предупреждение	WARNG
9	ms=Зависит от производителя	SW9
10	rm=Удаленный	SW10
11	tr=Уставка достигнута или зарезервирована	SW11
12	ila=Активен внутренний предел	SW12
13	oms=Зависит от рабочего режима	SW13
14	oms=Зависит от рабочего режима	SW14
15	ms=Зависит от производителя	SW15
16	ms=Зависит от производителя	SW16

Таблица определения значений для S-CAN

STATUSWORD							
8	7	6	5	4	3	2	1
w	sod	qs	ve	f	oe	so	rtso

Бит	Значение	Имя в виртуальном-физическом пространстве
1	Ten_st=состояние активации момента 1:ось в моменте 0:свободная ось	SW1
2	Ien_st=состояние активации движений 1:движения активированы 0:ось в фиксаторе момента	SW2
3	Stp_st=состояние остановки 1:выполняется постепенная остановка 0:остановка не активирована или процесс остановки завершен	SW3
4	Alm_rst=состояние аварийных сигналов 1:аварийный сигнал машины 0:не обнаружено аварийных сигналов	ALM
5	Ltc_st=состояние положения защелки 1:защелка положения выполнена, регистр готов для считывания 0:не обнаружено защелок положения	SW5
6	oms=зависит от рабочего режима	SW6

7	oms=зависит от рабочего режима	SW7
8	oms=зависит от рабочего режима	SW8

CNBYDEVICE

Синтаксис

CNBYDEVICE

устройство, плата, сп

Аргументы

устройство
плата
сп

имя устройства.
переменная integer. Возвращенный номер платы
переменная integer. Возвращенный номер узла

Описание

Возвращает номер платы и номер узла устройства, определенного в параметре **устройство**. Эта инструкция нужна для того, чтобы можно было использовать инструкции, которые не дают прямой ссылки на устройства, как, например, инструкции [READDICTIONARY](#) и [WRITEDICTIONARY](#).

Инструкция может использоваться для устройств, настроенных на шинах CAN, S-CAN, GreenBUS или EtherCAT.

Если возвращенный номер узла является отрицательным значением, это означает, что узел отключен.

READDICTIONARY

Синтаксис

READDICTIONARY

плата, сп, индекс, подиндекс, размпар, параметр, ошибка

Аргументы

плата
сп
индекс
подиндекс
размпар
параметр

ошибка

постоянная или переменная integer. Номер платы
постоянная или переменная integer. Номер узла
постоянная или переменная integer. Индекс объекта в словаре
постоянная или переменная integer. Подиндекс объекта в словаре
переменная integer. Величина считанного параметра.
переменная char, integer, float, double, вектор char, string. Переменная, получающая параметр
переменная integer. Код ошибки, возвращенный узлом

Описание

Считывает содержимое объекта словаря объектов, содержащихся в узле. Эта инструкция позволяет считывать при помощи протокола SDO все объекты, определенные согласно "CiA 402 CANopen device profile", помимо всех объектов, предоставленных производителем узла. По значению параметров **индекс**, **подиндекс** и **размпар** см. "CiA 402 CANopen device profile" или спецификацию производителя узла. Для устройств S-CAN параметр "подиндекс" всегда задается на ноль.

WRITEDICTIONARY

Синтаксис

WRITEDICTIONARY

плата, сп, индекс, подиндекс, размпар, параметр, ошибка

Аргументы

плата
сп
индекс
подиндекс
размпар
параметр

ошибка

постоянная или переменная integer. Номер платы
постоянная или переменная integer. Номер узла
постоянная или переменная integer. Индекс объекта в словаре
постоянная или переменная integer. Подиндекс объекта в словаре
постоянная или переменная integer. Размер параметра для записи
переменная char, integer, float, double, вектор char, string. Переменная, содержащая параметр
переменная integer. Код ошибки, возвращенный узлом

Описание

Записывает содержимое объекта словаря объектов, содержащихся в узле. Эта инструкция позволяет записывать при помощи протокола SDO все объекты согласно "CiA 402 CANopen device profile", помимо всех объектов, предоставленных производителем узла. По значению параметров **индекс**, **подиндекс** и **размпар** для записи см. "CiA 402 CANopen device profile" или спецификацию производителя узла. Для устройств S-CAN параметр "подиндекс" всегда задается на ноль.

10.3.16 EtherCAT

ACTIVATEMODE

Синтаксис

ACTIVATEMODE ось,параметр,ошибка

Аргументы

ось имя устройства типа оси
параметр константа или переменная integer. Рабочий режим
ошибка переменная integer. Код ошибки, возвращенный сервоустройством

Описание

Задаёт рабочий режим, определенный в переменной **параметре** в порядке, соответствующем "CiA 402 CANopen device profile". Рабочий режим оси при пуске соответствует значению **параметра**=9, т. е. "режим синхронной циклической скорости". Инструкция возвращает значение **ошибка**= 0, если команда завершилась успешно, или же возвращает код ошибки.
 Ниже приводится таблица со значениями, которые должны присваиваться **параметру** для выбора рабочего режима.

Значение	Определение
+6	режим homing
+9	режим синхронной циклической скорости

ECATGETREGISTER

Синтаксис

ECATGETREGISTER узел,адрес,раз,параметр,ошибка

Аргументы

узел константа или переменная integer. Положение, занятое ведомым модулем в цепи EtherCAT (начиная с 1)
адрес константа или переменная integer. Адрес регистра ESC (EtherCAT Slave Controller) для чтения (начиная с 0)
раз константа или переменная integer. Число байт для чтения (1, 2 или 4)
параметр переменная integer. Контейнер считанного параметра
ошибка переменная integer. Код ошибки.

Описание

Возвращает **[параметр]** содержимое регистра ESC (EtherCAT Slave Controller) указанного узла EtherCAT. Параметр **ошибка** будет содержать цифровой код ошибки, 0 если ошибок не было.

ECATSETREGISTER

Синтаксис

ECATSETREGISTER узел,адрес,раз,параметр,ошибка

Аргументы

узел константа или переменная integer. Положение, занятое ведомым модулем в цепи EtherCAT (начиная с 1)
адрес константа или переменная integer. Адрес регистра ESC (EtherCAT Slave Controller) для записи (начиная с 0)
раз константа или переменная integer. Число байт для записи (1, 2 или 4)
параметр константа или переменная integer. Параметр, который должен быть написан
ошибка переменная integer. Код ошибки.

Описание

Инструкция записывает **[параметр]** содержимое регистра ESC (EtherCAT Slave Controller) указанного узла EtherCAT. Параметр **ошибка** будет содержать цифровой код ошибки, 0 если ошибок не было.

GETPDO

Синтаксис

GETPDO плата,узел,nPDO,nObj,параметр,[ошибка]

Аргументы

плата константа или переменная integer. Номер платы
узел константа или переменная integer. Положение, занятое ведомым модулем в цепи EtherCAT (начиная с 1)

nPDO	константа или переменная integer. Идентификатор PDO (напр., \$1600h) или его положение в списке PDO, определенных в конфигурации аппаратного обеспечения для рассматриваемого узла (от 1 до 16)
nObj	константа или переменная integer. Идентификатор объекта (напр., \$6040h) или его положение в списке объектов, сконфигурированных в PDO (от 1 до 32)
параметр	переменная integer. Получает значение.
ошибка	переменная integer. Код ошибки.

Описание

Возвращает в **[параметр]** содержимое объекта, обмененного при помощи PDO, сконфигурированных для узла EtherCAT. Если переданные аргументы ошибочны и не был задан параметр **ошибка**, генерируется ошибка системы. Если был задан параметр **ошибка**, он будет содержать цифровой код соответствующей ошибки системы.

SETEOE**Синтаксис**

SETEOE **состояние,[ошибка]**

Аргументы

состояние	заданная константа. Допускаются эти значения: - ON подключает управление протоколом EoE - OFF отключает управление протоколом EoE
ошибка	переменная integer. Код ошибки.

Описание

Инструкция позволяет подключить или отключить управление протоколом EoE (Ethernet over EtherCAT). Эта функция использует вспомогательное программное обеспечение SnifferCE или SnifferRTX, управляющее связью EoE между числовым управлением и супервизором ПК. Переменная ошибка может принимать следующие значения:

- 0 если команда выполнена правильно
- 1 если команда не завершена в течение максимального предусмотренного времени 5 секунд
- 2 если произошла ошибка связи.

SETPDO**Синтаксис**

SETPDO **плата,узел,nPDO,nObj,параметр,[ошибка]**

Аргументы

плата	константа или переменная integer. Номер платы
узел	константа или переменная integer. Положение, занятое ведомым модулем в цепи EtherCAT (начиная с 1)
nPDO	константа или переменная integer. Идентификатор PDO (напр., \$1600h) или его положение в списке PDO, определенных в конфигурации аппаратного обеспечения для рассматриваемого узла (от 1 до 16)
nObj	константа или переменная integer. Идентификатор объекта (напр., \$6040h) или его положение в списке объектов, сконфигурированных в PDO (от 1 до 32)
параметр	целая переменная. Заданное значение.
ошибка	целая переменная. Код ошибки.

Описание

Задаёт содержимое **[параметр]** объекта, обмененного при помощи PDO, сконфигурированных для узла EtherCAT. Если переданные аргументы ошибочны и не был задан параметр **ошибка**, генерируется ошибка системы. Если был задан параметр **ошибка**, он будет содержать цифровой код соответствующей ошибки системы.

10.3.17 Платы TMSbus управлением CAN**GETCNSTATE****Синтаксис**

GETCNSTATE **плата, узел, состояние**

Аргументы

плата	константа или переменная integer. Номер платы
узел	константа или переменная integer. Номер узла
состояние	константа или переменная integer.

Описание

Возвращает состояние протокола NMT для **узла** указанной **платы**. Более подробная информация о значении этих параметров приводится в документации на отдельное устройство.

GETSDOERROR**Синтаксис**

GETSDOERROR **плата, ошибка**

Аргументы

плата	константа или переменная integer. Номер платы (от 1 до 4)
ошибка	переменная integer. Код ошибки

Описание

Возвращает последнюю выявленную **ошибку**, относящуюся к связи SDO для указанной **платы**. Более подробная информация о значении возвращаемого кода ошибки приводится в документации на отдельное устройство.

GETMNSTATE**Синтаксис**

GETMNSTATE **плата, состояние**

Аргументы

плата	константа или переменная integer. Номер платы (от 1 до 4)
состояние	константа или переменная integer.

Описание

Возвращает состояние протокола NMT для главного узла указанной **платы**. Более подробная информация о значении этих параметров приводится в документации на отдельное устройство.

RECEIVEPDO**Синтаксис**

RECEIVEPDO **плата, узел, номерPDO**

Аргументы

плата	константа или целая переменная. Номер платы (от 1 до 4)
узел	константа или целая переменная. Номер узла
номерPDO	константа или целая переменная. Номер PDO

Описание

Считывает содержание PDO, определенного **номеромPDO**, для указанного узла. Эта команда используется для считывания асинхронных PDO (т. е. PDO, которые в конфигурации аппаратного обеспечения включили опцию **Асинхронный**).
Считанный параметр копируется в соответствующие устройства подключенные в режиме виртуальное-физическое.
Эта команда может использоваться только с платами TMSCan и TMSCan+.

SENDPDO**Синтаксис**

SENDPDO **плата, узел, номерPDO**

Аргументы

плата	константа или целая переменная. Номер платы
узел	константа или целая переменная. Номер узла
номерPDO	константа или целая переменная. Номер PDO

Описание

ENABLE

Синтаксис

ENABLE ось1,[...ось6]

Аргументы

ось1...[...ось6] имена устройств типа оси

Описание

Подключает указанные оси. Оси на этапе инициализации всегда подключены. Эта инструкция вызывается только если оси были ранее отключены инструкцией [DISABLE](#).

Примечание

Оси шагового типа могут использоваться в этой инструкции только при их управлении удаленным TRS-AX.

ENABLEFORCEDINPUT

Синтаксис

ENABLEFORCEDINPUT

Аргументы

никакого аргумента

Описание

Подключает принудительное задание входов. Перед использованием инструкций для подключения или отключения устройств входа принудительным образом необходимо выполнить эту инструкцию. В противном случае инструкции принудительного задания входов не имеют никакого эффекта. См. также [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [RESETFORCEDINPUT](#), [SETFORCEDPORT](#), [SETFORCEDANALOG](#).

RESETFORCEDINPUT

Синтаксис

RESETFORCEDINPUT имявхода

Аргументы

имявхода имя цифрового входа

Описание

Принудительно задает на OFF вход, заданный **имявхода**. Для того, чтобы использовать эту инструкцию, необходимо предварительно подключить принудительное задание входов инструкцией [ENABLEFORCEDINPUT](#). См. также [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [SETFORCEDPORT](#), [SETFORCEDANALOG](#).

SETFORCEDANALOG

Синтаксис

SETFORCEDANALOG аналоговыйвход, значение

Аргументы

аналоговыйвход имя устройства аналогового входа
переменная константа или переменная integer или float или double

Описание

Принудительно задает **значение** в аналоговом входе, определенном **аналоговыйвход**. Для того, чтобы использовать эту инструкцию, необходимо предварительно подключить принудительное задание входов инструкцией [ENABLEFORCEDINPUT](#). См. также [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [RESETFORCEDINPUT](#), [SETFORCEDPORT](#).

SETFORCEDINPUT

Синтаксис

SETFORCEDINPUT имявхода

Аргументы

имявхода имя цифрового входа

Описание

STARTBLACKBOX

Синтаксис

STARTBLACKBOX [значение][,ошибка]

Аргументы

значение константа или целая переменная. Период регистрации
ошибка целая переменная. Код ошибки.

Описание

Активизирует функцию регистрации в файле всего действия локального или удаленного модуля. Под действием модуля подразумевается изменение во времени состояния логических устройств, за исключением устройств типа флажкового выключателя.

Период регистрации (**значение**) выражается в миллисекундах. Он не может быть меньше 10 и должен быть кратным периоду реального времени. В противном случае генерируется ошибка системы № 4399 (Параметр за пределами диапазона).

Если инструкция запускает регистрацию и **значение** опускается, значением по умолчанию считается 20.

Если инструкция возобновляет прерванную ранее регистрацию, не принимается во внимание никакого заданного **значения**.

Если оказалось невозможным запустить регистрацию, **ошибка** содержит значение, отличное от 0, в противном случае она содержит 0:

Код ошибки	Описание
0	Ошибок нет
1	Имеются различия между конфигурацией устройств в ЧПУ и в Albatros
2	Число регистрируемых устройств превышает максимально предусмотренное число для системы
3	В конфигурации не присутствуют устройства
4	Программное обеспечение связи в удаленном модуле не поддерживает функцию blackbox (только удаленные модули)
5	ЧПУ не позволяет запустить регистрацию
6	Ошибка загрузки библиотеки управления базой данных
7	Число столбцов в таблице превышает максимальное число столбцов, управляемых базой данных
8	Оказалось невозможным открыть базу данных на диске
9	Оказалось невозможным создать в базе данных таблицу регистраций
10	Ошибка в адресе IP для связи с удаленным модулем (только удаленные модули)
11	Оказалось невозможным создать гнездо связи для приема данных (только удаленные модули)
12	Оказалось невозможным ассоциировать локальный адрес с гнездом связи (только удаленные модули)
13	Оказалось невозможным подсоединиться к удаленному гнезду (только удаленные модули)
14	Оказалось невозможным получить доступ к области памяти, совместно используемой с ЧПУ
15	Конфигурация аппаратного обеспечения препятствует использованию функции "BlackBox"
16	Эта функция была отключена в tpa.ini

См. также [PAUSEBLACKBOX](#) и [ENDBLACKBOX](#).

10.3.20 ISO

ISOGO

Синтаксис

ISOGO ярлык, ось1 координата1, ось2, координата2, ось3, координата3, ось4, координата4, ось5, координата5, [значение]

Аргументы

ярлык константа или переменная integer. Ярлык, определяющий блок смещения. N в стандарте ISO

ось1 имя устройства типа оси. (X в стандарте ISO)

координата1 константа или переменная. Координата рабочего пространства оси1

ось2 имя устройства типа оси. (Y в стандарте ISO)

координата2 константа или переменная. Координата рабочего пространства оси2

ось3 имя устройства типа оси. (Z в стандарте ISO)

координата3 ось4	константа или переменная. Координата рабочего пространства оси3 имя устройства типа оси. (С в стандарте ISO)
коррдината4 ось5	константа или переменная. Координата пространства муфт оси4 имя устройства типа оси. (В в стандарте ISO)
координата5 значение	константа или переменная. Координата пространства муфт оси5 константа или переменная double. Представляет процент скорости подачи. (F в стандарте ISO)

Описание

Задаёт движение в быстром режиме. Управление участками движения в быстром режиме осуществляется в синхронизации. Определённые пользователем точки - это крайние точки отдельного проходимого участка смещения, чтобы оси между собой были синхронизированы. Это означает, что физические оси перемещаются независимо друг от друга, сохраняя, однако, одновременность пуска и прибытия, подобно тому, как происходит в инструкциях [MULTIABS](#) и [MULTIINC](#). Кончик инструмента не проходит линию в рабочем пространстве и его траектория не проверяется. Параметр **ярлык** используется вместе с инструкцией [SETLABELINTERP](#) для однозначного определения блока смещения. Первые три **координаты** определяют положение кончика в рабочем пространстве, а последующие две определяют значение осей, вращающихся в пространстве муфт. **Значение** FeedRate определяет процент снижения относительно максимальной возможной скорости (в ISO: F0 максимальная скорость, F100 FeedRate равно нулю, т. е. оси стоят).

Инструкция генерирует ошибку системы (4105- Инструкция не может быть выполнена на оси ИмяОси) при использовании на шаговых осях.

Невозможно использовать инструкцию [WAITCOLL](#), т. к. с момента столкновения вперед утратилась бы связь интерполяции с другими участвующими в движении осями с генерацией профиля, отличного от ожидаемого. При использовании генерируется ошибка системы "4101- Непоследовательное управление осью ИмяОси".

ISOG1

Синтаксис

ISOG1 **ярлык, ось1, координата1,ось2,координата2, ось3,координата3,ось4,координата4,ось5, координата5, [значение]**

Аргументы

ярлык	константа или переменная integer. Ярлык, определяющий блок смещения. (N в стандарте ISO)
ось1	имя устройства типа оси. (X в стандарте ISO)
координата1	константа или переменная. Координата рабочего пространства оси1
ось2	имя устройства типа оси. (Y в стандарте ISO)
координата2	константа или переменная. Координата рабочего пространства оси2
ось3	имя устройства типа оси. (Z в стандарте ISO)
координата3	константа или переменная. Координата рабочего пространства оси3
ось4	имя устройства типа оси. (C в стандарте ISO)
коррдината4	константа или переменная. Координата пространства муфт оси4
ось5	имя устройства типа оси. (B в стандарте ISO)
координата5	константа или переменная. Координата пространства муфт оси5
значение	константа или переменная double. Представляет значение скорости подачи. (F в стандарте ISO)

Описание

Определяет в рабочем пространстве точку, в которую должен прийти кончик инструмента в конце интерполяции блока. Параметр **ярлык** используется вместе с инструкцией [SETLABELINTERP](#) для однозначного определения блока смещения. Первые три **координаты** определяют положение кончика в рабочем пространстве, а последующие две определяют значение осей, вращающихся в пространстве муфт. **Значение** Feed определяет скорость кончика в плане единицы измерения (миллиметры или градусы) в минуту (при наличии инструкции [ISOG94](#)) обратно пропорционально продолжительности выполнения (при наличии инструкции [ISOG93](#)). Параметр **значение** обязателен для первой инструкции ISOG1 движения интерполяции.

Инструкция генерирует ошибку системы (4105- Инструкция не может быть выполнена на оси ИмяОси) при использовании на шаговых осях.

Невозможно использовать инструкцию [WAITCOLL](#), т. к. с момента столкновения вперед утратилась бы связь интерполяции с другими участвующими в движении осями с генерацией профиля, отличного от ожидаемого. При использовании генерируется ошибка системы "4101- Непоследовательное управление осью ИмяОси".

ISOG9

Синтаксис

ISOG9 **ось**

Аргументы

ось имя устройства типа оси

Описание

Позволяет осуществлять принудительную остановку движения. При наличии этой инструкции интерполяция или движение в быстром режиме останавливаются до перехода к следующему блоку. Однако это не блокирующая инструкция, каковой является инструкция [WAITSTILL](#). Блок управления информируется о принудительной остановке и процесс считывания блоков движения продолжается до заполнения буфера опережающего просмотра. Параметр **ось** определяет канал интерполяции 5 осей, которые должны останавливаться по завершении ранее вычисленного блока. В этом случае не существует различия, выполняется инструкция [ISOG1](#) или инструкция [ISOG0](#).

ISOG90**Синтаксис**

ISOG90 **ось**

Аргументы

ось имя устройства типа оси

Описание

Задаёт толкование координат как абсолютных координат. Параметр **ось** определяет канал интерполяции 5 осей, которые начиная с этой инструкции будут интерпретировать координаты осей как абсолютные координаты (условие по умолчанию). В этом случае не существует различия, выполняется инструкция [ISOG1](#) или инструкция [ISOG0](#).

ISOG91**Синтаксис**

ISOG91 **ось**

Аргументы

ось имя устройства типа оси

Описание

Задаёт толкование координат как относительных координат. Параметр **ось** определяет канал интерполяции 5 осей, которые начиная с этой инструкции будут интерпретировать координаты осей как относительные координаты. В этом случае не существует различия, выполняется инструкция [ISOG1](#) или инструкция [ISOG0](#).

ISOG93**Синтаксис**

ISOG93 **ось**

Аргументы

ось имя устройства типа оси

Описание

Задаёт интерпретацию скоростей обратно пропорционально продолжительности выполнения. Параметр **ось** определяет канал интерполяции 5 осей, который начиная с этой инструкции интерпретирует значение, переданное при помощи параметров F инструкции [ISOG1](#), обратно пропорционально продолжительности выполнения, выраженной в минутах. Благодаря этому блок управления может определять скорость, которая должна поддерживаться кончиком инструмента в блоках интерполяции.

ISOG94**Синтаксис**

ISOG94 **ось**

Аргументы

ось имя устройства типа оси

Описание

Задаёт интерпретацию скорости как единицу измерения в минуту. Параметр **ось** определяет канал

ISOG216

Синтаксис

ISOG216

имяМатрицыВращающихся, имяМатрицыДержателей, имяМатрицыИнструмента, маскаПодключения,ось1,ось2,ось3,ось4,ось5

Аргументы

имяМатрицыВращающихся

имя матрицы. Содержит данные, относящиеся к вращающимся осям

**имяМатрицыДержателей
имяМатрицыИнструмента
маскаПодключения**

имя матрицы. Содержит данные, относящиеся к держателю
имя матрицы. Содержит данные, относящиеся к инструментам
переменная или константа integer. Маска подключений осей С и В

ось1

имя устройства типа оси. (X в стандарте ISO)

ось2

имя устройства типа оси. (Y в стандарте ISO)

ось3

имя устройства типа оси. (Z в стандарте ISO)

ось4

имя устройства типа оси. (C в стандарте ISO)

ось5

имя устройства типа оси. (B в стандарте ISO)

Описание

Определяет три матрицы для определения параметров станка и пять образующих его устройств типа оси. Эта инструкция **должна** выполняться до любой другой инструкции ISO. Параметр **маскаПодключения** определяет, какие вращающиеся оси (С и/или В) подключать. Задаваемые значения см. в таблице ниже:

маскаПодключения	Описание
31	Подключение осей С и В
23	Подключение только оси В
15	Подключение только оси С
7	Отключение осей С и В

Примечание

Единицами измерения, в которых они выражаются, в конфигурации, для вращающихся осей должны быть градусы.

Связь между физическими осями и виртуальными осями ISO, заданная при помощи этой инструкции, прерывается в момент завершения задачи, в которой определена эта инструкция, или при помощи инструкции [ISOM2](#). Таким образом, оси могут использоваться для перемещений классического типа.

ISOG217

Синтаксис

ISOG217

ось1,ось2,ось3,ось4,ось5,виртуальнаяОсь1,виртуальнаяОсь2,виртуальнаяОсь3,виртуальнаяОсь4,виртуальнаяОсь5.

Аргументы

ось1

имя устройства типа оси

ось2

имя устройства типа оси

ось3

имя устройства типа оси

ось4

имя устройства типа оси

ось5

имя устройства типа оси

виртуальнаяОсь1

имя устройства типа виртуальной оси (X в стандарте ISO)

виртуальнаяОсь2

имя устройства типа виртуальной оси. (Y в стандарте ISO)

виртуальнаяОсь3

имя устройства типа виртуальной оси (Z в стандарте ISO)

виртуальнаяОсь4

имя устройства типа виртуальной оси (C в стандарте ISO)

виртуальнаяОсь5

имя устройства типа виртуальной оси (B в стандарте ISO)

Описание

Описывает образующие станок физические оси и виртуальные оси. Виртуальные оси описывают положение и ориентацию инструмента и должны объявляться виртуального типа в конфигурации Albatros. Первые пять указанных осей должны быть физическими осями, управление ими осуществляется интерполятором. Последующие пять могут быть виртуальными осями и используются в инструкциях [ISOG0](#) и [ISOG1](#).

Эта инструкция **должна** выполняться до любой другой инструкции ISO.

Формулы прямой и обратной кинематики для перехода из одного положения муфт (физических осей) в пространстве муфт в рабочее пространство (виртуальные оси) должны указываться инструкцией [KINEMATICEXPR](#) для каждой из десяти осей, определенных в инструкции ISOG217. Инструкция генерирует ошибку системы (4105- Инструкция не может быть выполнена на оси ИмяОси) при использовании на шаговых осях.

Примечание

Связь между физическими осями и виртуальными осями ISO, заданная при помощи этой инструкции, прерывается в момент завершения задачи, в которой определена эта инструкция, или при выполнении [ISOM2](#). Таким образом, оси могут использоваться для перемещений классического типа.

ISOM2

Синтаксис

ISOM2 ось

Аргументы

ось имя устройства типа оси

Описание

Освобождает оси от движения ISO, заданного при помощи инструкции [ISOG216](#) или инструкции [ISOG217](#)

ISOM6

Синтаксис

ISOM6 ось, индексСтрокаМатрицаВращающихся, индексСтрокаМатрицаДержателей, индексСтрокаМатрицаИнструмента

Аргументы

ось имя устройства типа оси
индексСтрокаМатрицаВращающихся константа или переменная integer. Индекс строки матрицы вращающихся осей
индексСтрокаМатрицаДержателей константа или переменная integer. Индекс строки матрицы держателей
индексМатрицыИнструмента константа или переменная integer. Индекс строки матрицы инструмента

Описание

Задаёт использование группы параметров, описывающих кинематику станка. **Индексы** относятся к трем матрицам с именем, определенным пользователем, объявленным в файле глобальных переменных Albatros. Параметр **ось** определяет соответствующий канал интерполяции. В таблицах ниже описывается, как должны объявляться три матрицы в файле глобальных переменных.

Поля матрицы	Матрица вращающихся осей
Сдвиг по X	Сдвиг по оси X между точкой опоры и точкой управления головкой
Сдвиг по Y	Сдвиг по оси Y между точкой опоры и точкой управления головкой
Сдвиг по Z	Сдвиг по оси Z между точкой опоры и точкой управления головкой
Смещение осей по X	Смещение по X между осью вращения и осью горизонтального вращения (когда положение оси C =0)
Смещение осей по Y	Смещение по Y между осью вращения и осью горизонтального вращения (когда положение оси C =0)
Смещение осей по Z	Расстояние выступ-точка опоры
Угол δ	Угол вокруг оси Z для правильного позиционирования головки относительно нуля станка
Угол γ	Угол между плоскостью вращения и осью горизонтального вращения

Поля матрицы	Матрица держателей
OffsetPU X	Сдвиг по оси X между точкой зацепления держателя с двигателем и точкой зацепления инструмента с держателем (когда положение оси C =0 и двигатель вертикален)

OffsetPU Y	Сдвиг по оси Y между точкой зацепления держателя с двигателем и точкой зацепления инструмента с держателем (когда положение оси C =0 и двигатель вертикален)
OffsetPU Z	Сдвиг по оси Z между точкой зацепления держателя с двигателем и точкой зацепления инструмента с держателем (когда положение оси C =0 и двигатель вертикален)
Угол α	Угол фазового сдвига между осью двигателя и осью инструмента (относительно Z)
Угол β	Угол фазового сдвига между осью двигателя и осью инструмента (относительно Y)

Поля матрицы	Матрица инструмента
Длина инструмента	Длина инструмента

ISOSETPARAM

Синтаксис

ISOSETPARAM

Идентификационный номер **Параметра**, **значение**

Аргументы

Идентификационный номер **Параметра** константа или переменная integer. Это номер, определяющий параметр

значение константа или переменная float. Это задаваемое значение.

Описание

Задаёт некоторые параметры, характеризующие плавность интерполированного движения ISO. В таблице ниже объясняется значение каждого **Идентификационный номер** **Параметра**, значения, в которых должна находиться переменная **значение**, и значения по умолчанию.

Идентификационный номер Параметра	Диапазон	Умолчание	Значение
0	0.0-100.0	50.0	Процент замедления на линейных осях при угловой точке. (0=никакого замедления, 100=максимальное замедление, разрешенное интерполятором)
1	0.0-100.0	50.0	Процент замедления на вращающихся осях при угловой точке. (0=никакого замедления, 100=максимальное замедление, разрешенное интерполятором)
2	0.5-1.0	0.9	Коэффициент снижения скорости на криволинейной абсциссе при угловой точке. (1=никакого замедления, 0.5=максимальное разрешенное замедление)
3	0.0-100.0	60.0	Процент замедления при приближенных разрывах. (0=никакого замедления, 100=максимальное замедление, разрешенное интерполятором)
4	0.0-100.0	10.0	Процент сглаживания траектории
5	(Выше 0.0)-1.0	0.2	Минимальный размер отрезка, проходимого только с линейными осями. Значение выражается в миллиметрах.
6	(Выше 0.0)-1.0	0.1	Минимальный размер отрезка, проходимого только с вращающимися осями. Значение выражается в градусах.
7	0.0-100.0	100.0	Нижний предел фильтра сглаживания

8	1.0.100.0	1.0	Множитель, примененный к ускорениям и замедлениям, определенным в конфигурации. Приращает максимальные ускорение и замедление только линейных осей. Значения за пределами интервала генерируют ошибку системы 4399 - Параметр за пределами диапазона.
9	1.0-100.0	1.0	Множитель, примененный к ускорениям и замедлениям, определенным в конфигурации. Приращает максимальные ускорение и замедление только поворотных осей. Значения за пределами интервала генерируют ошибку системы 4399 - Параметр за пределами диапазона.
10	0.0-1.0	0.0	Флажок для подключения (0.0) или отключения (1.0) снижения скорости между последовательными блоками в случае угловой точки. Отключение может также обеспечиваться при использовании инструкции со следующими параметрами: ISOSETPARAM 0 0.0 ISOSETPARAM 1 0.0 ISOSETPARAM 2 1.0

KINEMATICEXPR

Синтаксис

KINEMATICEXPR

ось = выражение

Аргументы

ось
выражение

имя устройства типа физической или виртуальной оси
совокупность операторов

Описание

Позволяет определять отдельные выражения прямой и обратной кинематики. Перед выполнением этой инструкции должна вызываться инструкция [ISOG217](#), описывающая образующие станок физические и виртуальные оси. Для каждой оси, определенной в [ISOG217](#), должна вызываться инструкция KINEMATICEXPR. Выражение кинематики оси в пространстве муфт (обратная кинематика) может быть функцией переменных, констант и координат осей в рабочем пространстве. Выражение кинематики оси в рабочем пространстве (прямая кинематика) может быть функцией переменных, констант и координат осей в пространстве муфт.

Синтаксис **выражения** - такой же, который описывается в инструкции [EXPR](#), с той разницей, что не могут использоваться локальные переменные. Кроме того, не могут использоваться оси того же типа, что и ось, объявленная в **оси** и не объявленная в инструкции [ISOG217](#). Например, при определении кинематики виртуальной оси, уже объявленной в инструкции [ISOG217](#), в выражении могут использоваться только пять физических осей, объявленных в [ISOG217](#).

Пример

```
ut as double ; номер инструмента
offsety as double ; сдвиг Y выступ точки опоры
offsetz as double ; сдвиг Z выступ точки опоры
```

Function ISO5Ax

```
setval 100,ut
setval 120.0,offsety
setval 60.0,offsetz
; ЯВНАЯ КИНЕМАТИКА
ISO217 Rx Ry Rz Rc Rb X Y Z C B
```

```

; ОПРЕДЕЛЕНИЕ ВЫРАЖЕНИЙ КИНЕМАТИКИ
; ЯВНАЯ ОБРАТНАЯ КИНЕМАТИКА физической ОСИ Rx
KinematicExpr Rx = X - 135 + ut * sin ( B ) * cos ( C )

; ЯВНАЯ ОБРАТНАЯ КИНЕМАТИКА физической ОСИ Ry
KinematicExpr Ry = Y + offsety + ut * sin ( B ) * sin ( C )

; ЯВНАЯ ОБРАТНАЯ КИНЕМАТИКА ОСИ Rz
KinematicExpr Rz = Z + offsetz + ut * cos ( B )

; ЯВНАЯ ОБРАТНАЯ КИНЕМАТИКА ОСИ Rc
KinematicExpr Rc = C

; ЯВНАЯ ОБРАТНАЯ КИНЕМАТИКА ОСИ Rb
KinematicExpr Rb = B

; ЯВНАЯ ПРЯМАЯ КИНЕМАТИКА ОСИ X
KinematicExpr X = Rx + 135 - ut * sin ( Rb ) * cos ( C )

; ЯВНАЯ ПРЯМАЯ КИНЕМАТИКА ОСИ Y
KinematicExpr Y = Ry - offsety - ut * sin ( Rb ) * sin ( C )

; ЯВНАЯ ПРЯМАЯ КИНЕМАТИКА ОСИ Z
KinematicExpr Z = Rz - offsetz - ut * cos ( Rb )

; ЯВНАЯ ПРЯМАЯ КИНЕМАТИКА ОСИ C
KinematicExpr C = Rc

; ЯВНАЯ ПРЯМАЯ КИНЕМАТИКА ОСИ B
KinematicExpr B = Rb

; ДВИЖЕНИЕ
ISOG0 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,0.0
ISOG1 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,1000.0
ISOG1 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,1000.0
ISOG1 1003,X 996.0,Y 600.0,Z 0.0,C 90.0,B 45.0,1000.0
WAITSTILL X
ISOM2 X

```

FRet

10.3.21 Больше недоступные инструкции

INPBCD	считывает серию цифровых полубайтов в формате BCD
OUTBCD	изменяет серию цифровых полубайтов в формате BCD
SETFORCEDBCD	принудительное задание серии полубайтов в формате BCD
CANOPENDRIVER	открытие канала связи CANopen
CANCLOSEDRIVER	закрытие канала связи CANopen
CANRESETBOARD	выполнение сброса платы CANopen
CANSETOBJECT	запись объекта CANopen
CANGETOBJECT	считывание объекта CANopen
SLMCOMMAND	выполнение команды SLM
SLMEEPROMDISABLE	выполнение команды отключения записи EEPROM
SLMEEPROMENABLE	выполнение команды подключения записи EEPROM
SLMGETEEPROM	считывание ячейки памяти EEPROM
SLMGETPARAM	считывание параметра SLM
SLMGETREGISTER	считывание регистра SLM
SLMGETSTATUS	считывание величины привода
SLMSEEEPROM	запись в ячейку памяти EEPROM
SLMSETPARAM	задание параметра SLM
SLMSETREGISTER	задание регистра SLM
HOMING	выполнение "поиска нуля"

SYNCROOPEN	открывает канал синхронизированного перемещения
SYNCROCLOSE	закрывает канал синхронизированного перемещения
SYNCROMOVE	задает точку синхронизированного перемещения
SYNCROSETACC	задает ускорение для синхронизированных перемещений
SYNCROSETDEC	задает замедление для синхронизированных перемещений
SYNCROSETVEL	задает скорость для синхронизированных перемещений
SYNCROSETFEED	масштабирует скорость осей в синхронизированном движении
SYNCROSTARTMOVE	начинает обработку синхронизированного движения
GETVF	считывает значение преобразователя напряжение/частота

10.3.22 Инструкции, которые не могут использоваться на событии

Следующие инструкции не могут использоваться в функциях, которые вызываются инструкциями [ONFLAG](#), [ONINPUT](#) и [ONERRSYS](#). Их использование запрещено также в [задачах реального времени](#).

Инструкции, которые, в свою очередь, выполняют вызов функции при прерывании:

- ONFLAG
- ONINPUT
- ONERRSYS

Инструкции, связанные с ожиданием:

- WAITINPUT
- WAITFLAG
- WAITACC
- WAITCOLL
- WAITDEC
- WAITREG
- WAITTARGET
- WAITWIN
- WAITSTILL
- WAITTASK
- WAITRECEIVE
- WAITPERSISTINPUT
- MULTIWAITFLAG
- MULTIWAITINPUT

Инструкции связи:

- SEND
- RECEIVE
- CLEARRECEIVE
- COMOPEN
- COMCLOSE
- COMREAD
- COMREADSTRING
- COMWRITE
- COMWRITESTRING
- COMGETERROR
- COMCLEARRXBUFFER
- COMGETRXCOUNT

Следующие инструкции, связанные с движением осей:

- MOVINC
- MOVABS
- LINEARINC
- LINEARABS
- CIRCLE
- CIRCINC
- CIRCABS
- HELICINC
- HELICABS
- COORDIN
- MULTIABS
- MULTIINC
- SETRIFLOC
- SETTOLERANCE
- RESRIFLOC
- SETPFLY
- SETPZERO
- SETINDEXINTERP

- STARTINTERP
- FASTREAD
- ENABLE
- DISABLE
- ENDMOV

Инструкции ISO:

- ISOG0
- ISOG1
- ISOG9
- ISOG90
- ISOG91
- ISOG93
- ISOG94
- ISOG216
- ISOG217
- ISOM2
- ISOM6
- ISOSETPARAM
- KINEMATICEXPR

Инструкции Blackbox:

- ENDBLACKBOX
- PAUSEBLACKBOX
- STARTBLACKBOX

Следующие инструкции, связанные с управлением EtherCAT

- READDICTIONARY
- WRITEDICTIONARY

Инструкции, связанные с многозадачным режимом:

- SENDMAIL
- WAITMAIL
- ENDMAIL
- SENDIPC
- WAITIPC
- TESTMAIL
- TESTIPC

Инструкции, связанные с длительным временем обработки:

- SORT
- FIND
- FINDB
- MOVEMAT

10.4 Примеры

10.4.1 Обнуление при прерывании

```

;-----
; Пример стандартной программы обнуления налету
;
; Функция выполняет следующие операции:
;
; 1) Задает ось, отключая программные пределы
;    и обнуляя координату
; 2) Проверяет, что датчик уже не находится в состоянии ON.
;    Если он находится в состоянии ON, перемещает ось и дожидается возврата
;    в состояние OFF. Если этого не происходит за 30 секунд,
;    генерирует сообщение об ошибке.
; 3) Задает скорость поиска датчика
; 4) Запускает движение оси и подключает обнуление
;    налету для указанной оси. При прерывании
;    координата оси обнуляется и автоматически запускается
;    движение на координату расцепления.

```

```

; 5) Ожидает прибытия оси на координату расцепления
; 6) Восстанавливает пределы оси
;
;
;
; © ТРА.
-----
Function Быстрое _обнуление

    ResLimPos    ось           ; Инициализация оси
    ResLimNeg    ось
    SetQuote     ось,0

    IfInput      FastInput,OFF,Goto Continue ; Тест датчика занят
    SetVel       ось,5          ; Задает скорость
                                           ; расцепления
    MovAbs       ось,30         ; перемещает ось
    WaitInput    FastInput,OFF,30,Call Error ; Контроль расцепления
                                           ; микровыключателя,
                                           ; ошибка после TimeOut=30

    EndMov       ось           ; Стоп оси
    WaitStill    ось           ; Ожидание остановленной
оси

Continue:
    SetVel       ось,10        ; Скорость поиска датчика обнуления
    MovAbs       ось,-1000    ; Движение поиска датчика в отрицательную
сторону
    SetPFly      ось,ON,10,0  ; Зацепление прерывания
                                           ; и задание координаты и скорости
                                           ; расцепления
    WaitStill    ось           ; Ожидание остановленной оси

    SetLimPos    ось           ; Восстановление пределов оси
    SetLimNeg    ось

    Fret

; подпроцедура отправки сообщения об ошибке
Error:
    Error        ERR_SETP     ; Сигнализация ошибки, невозможно продолжать
    Ret

```

10.4.2 Сервер перемещения осей

```

-----
; Пример сервера перемещения осей:
;
; Сервер перемещает оси станка
; для других задач.
;
; Задачи клиента направляют команды в форме
; сообщений (почты) на почтовый ящик.
;
; Сервер забирает команды из ящика и выполняет их.
;
; Запросы в ящике становятся в очередь, поэтому
; если запрос поступает, когда сервер уже
; занят, он не теряется, а будет исполнен как только это станет возможным.
;
; Сервер является единственной задачей, которая перемещает оси. Так можно
избежать
; конфликтов.

```

```

;
; Сервер реализован функцией Главные_оси.
;
; Пример клиента реализован функцией Check_flag.
; Эта функция периодически проверяет состояние
; флажка и когда находит его в состоянии ON, направляет на сервер
; команду выполнения обнуления осей.
; Флажок будет, предположительно, устанавливаться на ON вручную
; оператором с использованием, например, обзорного экрана.
;
;-----
;-----
; -- ГЛОБАЛЬНЫЕ КОНСТАНТЫ СТАНКА --
;-----
Const MBOX = 101      ; определяет почтовый ящик
; команд
Const SETP = 10      ; обнуление осей
Const CHG = 11      ; смена инструмента
Const ОТВЕРСТИЕ = 12 ; выполнение сверления

;-----
; --- ГРУППА ОСЕЙ ---
;-----

; определение сообщений об ошибке
Defmsg ERR_CMD "Команда группы осей неизвестна"

; --- Сервер---
Function Главные_оси autorun

    local cmd as integer          ; команда
    local координата_X as double ; координата X отверстия
    local координата_Y as double ; координата Y отверстия

Loop:
    waitmail MBOX, cmd, координата_X, координата_Y ; ожидание команды

    ; После получения команды мы ее идентифицируем и
    ; выполним соответствующее действие
    select cmd

    case SETP
        fcall обнуление_оси          ; обнуление осей
    case CHG
        fcall смена_инструмента      ; Выполняет смену инструмента
    case FORO
        fcall Сверление координата_X, координата_Y; сверление на
                                           ; указанных координатах
    case else
        call ошибка
    endselect

    endmail MBOX                    ; извещение об исполнении
команды
    goto Loop                       ; возвращается в ожидание
НОВОЙ
                                           ; команды

fret

```

```

; подпроцедура отправки сообщения об ошибке
ошибка:
  error ERR_CMD
  ret

;-----
; --- ОБЩАЯ ГРУППА ---
;-----

; --- Клиент ---
Function Check_flag

Loop:

  ifflag Setp_оси,OFF, goto loop          ; тест состояния флажка

  ; ОК, флажок на ON, отправка команды
  sendmail MBOX,WAITTASK,SETP,0.0,0.0

  resetflag Setp_оси                    ; сброс флажка

  goto loop                              ; возврат в ожидание

  fret

  ; ОБРАТИТЕ ВНИМАНИЕ, ЧТО:
  ; - после команды "SETP" необходимо указать
  ;   два параметра "координата_X" и "координата_Y" даже если это
  ;   не имеет смысла для операции обнуления.
  ; Однако сервер не может заранее знать, какая команда
  ; ему будет направлена, поэтому необходимо указать два значения
  ; того же типа, которые сервер ожидает получить
  ; в этом случае - два DOUBLE. Переданные значения - "0.0" и "0.0".
  ; - параметр "WAITTASK" обеспечивает, чтобы клиент оставался в ожидании
  ;   выполнения команды сервером.
  ; Клиент будет продолжать свое выполнение только когда сервер
  ; выполнит ENDMAIL или начнет обработку новой
  ; команды (WAITMAIL).

```

10.4.3 Цикл Главной функции с управлением ошибками

```

;-----
; Предполагаемая главная функция
; инициализирует станок и выполняет цикл контроля
;-----
Function Main
  OnErrSys GestErrSys                    ; подключает управление ошибками

  StartTask Аварийные ситуации           ; инициализирует
  StartTask Вычислительная машина
  ПодключитьОси

Loop:
  IfFlag Flag,OFF, Сброс аварийной ситуации
  .....
  Goto loop
Fret

```

```

;-----
; функция управления ошибками
;-----
Function GestErrSys
  Param nerrore as integer
  Param task as function
  Param типустройства as device

  EndTask          Вычислительная машина          ; Завершает задачу
                  ; вычислительной
  машины
  If               nerrore, >, 5, goto noerraxis    ; Первые 5 ошибок
                  ; относятся
                  ; к осям

  ResetFlag       Flag
  ОтключитьОси

noerraxis:
Fret

```

10.4.4 Операции на строках

```

;-----
; Пример манипулирования строками
;-----
Function пример
  Local строка1 as string
  Local строка2 as string
  Local строка3 as string
  Local длина as integer
  Local положение as integer

  SetString       "Строка", строка1                ; строка1 содержит
                  ; теперь "Строка"
  SetString       " проба", строка2
  AddString       строка1, строка2, строка3        ; строка3 содержит
                  ; "Пробная строка"
  Search          строка3, 't', положение          ; положение равно 2
  Search          строка3, 'Z', положение          ; положение равно -1
  Left           строка3, 7, строка1               ; строка1
                  ; содержит "Строка"
  Right          строка3, 2, строка2               ; строка2
                  ; содержит "va"
  Mid            строка3, 9, 2, строка3            ; строка3
                  ; содержит "из"
  ControlChar    65, строка1                       ; строка1
                  ; содержит "A"
  Len            строка3, длина                    ; длина равна 2
  Str            длина, строка3                    ; строка3
                  ; содержит "2"
  Val           положение, строка1                 ; строка1
                  ; содержит "-1"
  AddString      "Результат - ", строка1, строка2

```



```
; строка2 содержит "Результат равен -1"
```

```
Fret
```

10.4.5 Последовательное / параллельное выполнение

```
-----
; Пример стандартной программы, контролирующей обнуление
; станка с 3 осями, избегая возможных
; механических интерференций.
;
; Обнуления отдельных осей реализуются
; функциями, текст которых опущен.
; См. пример "Стандартная процедура обнуления".
;
; Вначале выполняется только обнуление
; оси Z (которое, как мы предполагаем, не может
; выполняться одновременно с другими),
; по завершении этого обнуления выполняются
; одновременно обнуления осей
; X и Y.
-----
; сообщение для оператора (переведенное на соответствующий язык)
DefMsg      MSG_SETP      ITA  "Выполняется обнуление осей ..."
              ENG  "Homing in progress ..."
```

Function ОбнулениеОсей

```
Message      MSG_SETP              ; информирует оператора
Fcall        ОбнулениеОсиZ         ; обнуление оси Z
; ОК обнуление оси Z завершено
StartTask    ОбнулениеОсиX         ; запускает обнуление X и Y
StartTask    ОбнулениеОсиY
WaitTask     ОбнулениеОсиX         ; ожидает завершения
WaitTask     ОбнулениеОсиY
DelMessage   MSG_SETP              ; удаляет сообщение
              ; для оператора
```

```
Fret
```

10.4.6 Стандартная процедура обнуления

```
-----
; Пример стандартной процедуры обнуления оси
;
; Функция выполняет следующие операции:
; 1) отключает программные пределы оси
; 2) задает скорость поиска выключателя
; 3) перемещает ось на инкрементальную координату, которая
; может обеспечить достижение выключателя
; 4) ожидает, пока ось вызовет срабатывание выключателя
; 5) останавливает ось и ожидает окончания движения
; 6) задает скорость (низкую) расцепления выключателя
; 7) отводит ось назад по необходимости
; для расцепления выключателя
; 8) ожидает расцепления выключателя
```

```

; 9) задает новую координату оси
; 10) восстанавливает скорость по умолчанию и программные пределы
;
; © ТРА.
;-----

```

Function Обнуление

```

ResLimPos ось ; отключает программные пределы
ResLimNeg ось

SetVel ось,10 ; задает скорость

MovInc ось,10000 ; перемещает ось

waitInput Switch,ON ; ожидает выключатель

EndMov ось ; останавливает ось
waitStill ось ; ожидает остановки оси

SetVel ось, 0.1 ; задает скорость расцепления

MovInc ось,-100 ; перемещает ось

waitInput Switch,OFF ; ожидает расцепления выключателя

SetQuote ось,0 ; задает новую координату

SetVel ось ; сброс скорости
SetLimpos ось ; восстановление программных пределов
SetLimNeg ось

```

Fret

10.4.7 Движения ISO

Пример движения ISO

```

;
; Генерируется профиль при помощи инструкций ISOG0 и ISOG1
;
; © ТРА.
;-----*
```

```

; Объявление матриц ISO
; Матрица вращающихся осей
MxRot[5] as double:off_X double:off_Y double:off_Z double:dis_X
double:dis_Y double:dis_Z double:delta double:gamma
; Матрица держателей
Mxдерж[1] as double:off_X double:off_Y double:off_Z double:alpha
double:beta
; Матрица инструмента
Mхинструменты[10] as double:ut double

```

Function Интерполяция ISO

```

; задание стандартных значений определения параметров станка
setval 90.0 MxRot[5].gamma
setval 260.3 Mхинструменты[10].ut
setval Mхинструменты[10].ut ut

; задание параметров алгоритма

```

```
IsosetParam 0 50
IsosetParam 1 50
IsosetParam 2 0.9
IsosetParam 3 60
IsosetParam 4 30

; задание станка: объявляет три матрицы,
; используемые для определения параметров станка, и
; физические оси, используемые в движениях ISO
isoG216 MxRot MxДерж MxИнструменты 31 X Y Z C B ; ЯВНАЯ КИНЕМАТИКА

; задание группы параметров, описывающих
; кинематику станка
isoM6 X 5 1 10 ;ЯВНАЯ КИНЕМАТИКА

; задание начальных значений
setquote x 500
setquote y 300
setquote z 0
setquote c 0
setquote b 0
setvel x
setvel y
setvel z
setvel c
setvel b
setveli x y z c b

; выполнение профиля
isoG0 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,50.0
isoG1 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,10000.0
isoG1 1003,X 996.0,Y 600.0,Z 0.0,C 90.0,B 45.0,10000.0
isoG1 1002,X 600.0,Y 600.0,Z 0.0,C 90.0,B 45.0,10000.0
isoG1 1004,X 599.131759111665,Y 599.924038765061,Z 0,C 100,B
45.0,10000.0
isoG1 1006,X 598.289899283372,Y 599.69846310393,Z 0,C 110,B 45.0,10000.0
isoG1 1005,X 597.5,Y 599.330127018922,Z 0,C 120,B 45.0,10000.0
isoG1 1003,X 596.786061951567,Y 598.830222215595,Z 0,C 130,B
45.0,10000.0
isoG1 1002,X 596.169777784405,Y 598.213938048433,Z 0,C 140,B
45.0,10000.0
isoG1 1012,X 595.669872981078,Y 597.5,Z 0,C 150,B 45.0,10000.0
isoG1 1011,X 595.301536896071,Y 596.710100716628,Z 0,C 160,B
45.0,10000.0
isoG1 1031,X 595.075961234939,Y 595.868240888335,Z 0,C 170,B
45.0,10000.0
isoG1 1102,X 595.0,Y 0.0,Z 0.0,C 180.0,B 45.0,10000.0
waitstill X Y Z C B
fret
```

Tecnologie e Prodotti per l'Automazione S.r.l

Via Carducci 221
I - 20099 Sesto S.Giovanni
(MI)
Tel. +39 02.36527550
Fax. +39 02.2481008
www.tpaspa.com