

BU 0000 – ru

NORD CON

Руководство по эксплуатации







## Оглавление

<b>1</b>	<b>Введение</b> .....	<b>9</b>
1.1	О программе NORD CON.....	9
1.2	Работа с NORD CON.....	9
<b>2</b>	<b>Окна и виды</b> .....	<b>12</b>
2.1	Структура окна .....	12
2.2	Структура главного меню .....	13
2.2.1	Категория «File».....	13
2.2.2	Категория «Edit» .....	14
2.2.3	Категория «Project» .....	16
2.2.4	Категория «Device».....	16
2.2.5	Категория «View».....	18
2.2.6	Категория «Extras» .....	19
2.2.7	Категория «Help» .....	19
2.3	Панели инструментов .....	20
2.3.1	Стандартная панель.....	20
2.3.2	Устройство .....	21
2.3.3	Категория «Start» .....	22
2.4	Вид «Project» .....	22
2.4.1	Структура контекстного меню .....	23
2.5	Вкладка «Messages» .....	25
2.6	Вкладка «Remote».....	26
2.7	Прикрепление и открепление окон .....	27
<b>3</b>	<b>Передача данных</b> .....	<b>33</b>
3.1	USS.....	33
3.1.1	Общие настройки.....	33
3.1.2	Busscan.....	34
3.2	Ethernet .....	36
3.2.1	General settings .....	36
3.2.2	Scan .....	37
<b>4</b>	<b>Параметризация</b> .....	<b>39</b>
4.1	Изменение параметров.....	39
4.2	Фильтр параметров.....	40
4.3	Офлайн-параметризация (без подключения к сети) .....	40
4.4	Вкладка параметров .....	41
4.5	Сравнительный отчет .....	42
4.6	Загрузка параметров с устройства .....	42
4.7	Передача параметров на устройство .....	43
<b>5</b>	<b>Управление</b> .....	<b>45</b>
5.1	Информация о функциях управления .....	45
5.2	Стандартное управление .....	46
5.3	Дополнительные функции управления.....	46
5.3.1	Обзор .....	46
5.3.2	Управление .....	47
5.3.3	Работа с расчетными и действительными значениями.....	48
5.3.4	Формат расчетного и действительного значения.....	49
5.3.5	Слово состояния.....	49
5.3.6	Слово управления .....	50
<b>6</b>	<b>Дистанционное управление</b> .....	<b>52</b>
6.1	Стандартная настройка .....	52
6.2	NORDAC SK 200 E .....	53
6.3	NORDAC SK 700/500/300 E .....	54
6.4	NORDAC vector mc.....	56
6.5	NORDAC vector ct.....	57
<b>7</b>	<b>Осциллограф</b> .....	<b>60</b>
7.1	Обзор .....	60

7.2	Вывод данных.....	60
7.3	Управление.....	61
7.4	Измерения .....	63
7.5	Печать, сохранение и загрузка серий измерений.....	64
<b>8</b>	<b>Редактор макросов .....</b>	<b>65</b>
8.1	Окна и панели пользовательского интерфейса.....	65
8.1.1	Окно переменных .....	65
8.1.2	Окно свойств .....	65
8.1.3	Окно журнала .....	68
8.2	Редактирование макроса.....	68
8.2.1	Создание нового макроса .....	68
8.2.2	Открытие макроса .....	68
8.2.3	Сохранение макроса .....	69
8.2.4	Вставка операторов.....	69
8.2.5	Копирование операторов .....	69
8.2.6	Вырезание операторов.....	69
8.2.7	Удаление операторов.....	69
8.2.8	Поиск и замена / Search and replace.....	69
8.2.9	Перемещение оператора вверх.....	69
8.2.10	Перемещение оператора вниз.....	69
8.2.11	Добавление нового оператора .....	70
8.3	Управление выполнением.....	71
8.3.1	Выполнение макроса.....	71
8.3.2	Прерывание выполнения .....	71
8.3.3	Выполнение следующего оператора .....	72
<b>9</b>	<b>Редактор кадров USS.....</b>	<b>73</b>
9.1	Master (order) .....	75
9.2	Device (response).....	76
<b>10</b>	<b>ПЛК.....</b>	<b>78</b>
10.1	Общая информация.....	78
10.1.1	Спецификация ПЛК .....	78
10.1.2	Конструкция ПЛК.....	79
10.1.2.1	Запоминающее устройство .....	79
10.1.2.2	Образ процесса .....	79
10.1.2.3	Программная задача .....	80
10.1.2.4	Обработка уставки .....	80
10.1.2.5	Обработка данных через накопительный регистр .....	81
10.1.3	Функционал .....	81
10.1.3.1	Библиотека управления движением Motion Control Lib .....	81
10.1.3.2	Электронные редукторы с летучей пилой .....	81
10.1.3.3	Визуализация .....	81
10.1.3.4	Регулятор технологического процесса .....	82
10.1.3.5	Передача данных через CANopen .....	82
10.2	Составление программы для ПЛК .....	83
10.2.1	Загрузка, сохранение и печать .....	83
10.2.2	Редактор.....	83
10.2.2.1	Переменные и описание функционального блока .....	84
10.2.2.2	Окно ввода .....	85
10.2.2.3	Окна отображения контрольной точки и точки прерывания (Watch- & Breakpoint) .....	85
10.2.2.4	Окно сообщений ПЛК .....	86
10.2.3	Передача программы ПЛК на прибор .....	86
10.2.4	Отладка .....	87
10.2.4.1	Контрольные точки (Watchpoints) .....	87
10.2.4.2	Точки прерывания (Breakpoints) .....	87
10.2.4.3	Пошаговое выполнение (Single Step) .....	87
10.2.5	Конфигурация ПЛК .....	88
10.3	Функциональные блоки.....	89
10.3.1	CANopen.....	89
10.3.1.1	Обзор .....	89
10.3.1.2	FB_NMT .....	90
10.3.1.3	FB_PDConfig .....	90
10.3.1.4	FB_PDORceive .....	92
10.3.1.5	FB_PDOSend .....	94
10.3.2	Электронные редукторы с летучей пилой* .....	96

10.3.2.1	Обзор	97
10.3.2.2	FB_FlyingSaw	97
10.3.2.3	FB_Gearing	98
10.3.3	Motion Control .....	99
10.3.3.1	MC_Control	100
10.3.3.2	MC_Control_MS	102
10.3.3.3	MC_Home	103
10.3.3.4	MC_MoveAbsolute	104
10.3.3.5	MC_MoveAdditive	105
10.3.3.6	MC_MoveRelative	106
10.3.3.7	MC_MoveVelocity	107
10.3.3.8	MC_Power	108
10.3.3.9	MC_ReadActualPos	109
10.3.3.10	MC_ReadParameter	110
10.3.3.11	MC_ReadStatus	111
10.3.3.12	MC_Reset	112
10.3.3.13	MC_Stop	112
10.3.3.14	MC_WriteParameter_16 / MC_WriteParameter_32	113
10.3.4	Standard .....	114
10.3.4.1	Вычитающий счетчик CTD	114
10.3.4.2	Суммирующий счетчик CTU	115
10.3.4.3	Суммирующий и вычитающий счетчик CTUD	116
10.3.4.4	R_TRIG и F_TRIG	117
10.3.4.5	RS Flip Flop	118
10.3.4.6	SR Flip Flop	119
10.3.4.7	Задержка выключения TOF	120
10.3.4.8	Задержка включения TON	120
10.3.4.9	Синхроимпульс TP	121
10.3.5	Доступ к областям памяти частотного преобразователя .....	122
10.3.5.1	FB_ReadTrace	123
10.3.5.2	FB_WriteTrace	124
10.3.6	Модуль визуализации ParameterBox .....	125
10.3.6.1	Обзор визуализации	126
10.3.6.2	FB_DINTToPBOX	126
10.3.6.3	FB_STRINGToPBOX	129
10.3.7	FB_Capture (Получение быстрых результатов) .....	131
10.3.8	FB_DinCounter .....	134
10.3.9	FB_FunctionCurve .....	135
10.3.10	FB_PIDT1 .....	136
10.3.11	FB_ResetPosition .....	138
10.3.12	FB_Weigh .....	139
10.4	Операторы .....	140
10.4.1	Арифметические операторы .....	140
10.4.1.1	ABS	141
10.4.1.2	ADD и ADD(	141
10.4.1.3	DIV и DIV(	142
10.4.1.4	LIMIT	142
10.4.1.5	MAX	143
10.4.1.6	MIN	143
10.4.1.7	MOD и MOD(	144
10.4.1.8	MUL и MUL(	144
10.4.1.9	MUX	145
10.4.1.10	SUB и SUB(	145
10.4.2	Расширенные математические операторы .....	146
10.4.2.1	COS, ACOS, SIN, ASIN, TAN, ATAN	146
10.4.2.2	EXP	146
10.4.2.3	LN	147
10.4.2.4	LOG	147
10.4.2.5	SQRT	148
10.4.3	Операции над битами .....	148
10.4.3.1	AND и AND(	148
10.4.3.2	ANDN и ANDN(	149
10.4.3.3	NOT	150
10.4.3.4	OR и OR(	150
10.4.3.5	ORN и ORN(	151
10.4.3.6	ROL	151
10.4.3.7	ROR	152
10.4.3.8	S и R	152

10.4.3.9	SHL	153
10.4.3.10	SHR	153
10.4.3.11	XOR и XOR(	154
10.4.3.12	XORN и XORN(	154
10.4.4	Операторы загрузки и сохранения	155
10.4.4.1	LD	155
10.4.4.2	LDN	155
10.4.4.3	ST	156
10.4.4.4	STN	156
10.4.5	Операторы сравнения	156
10.4.5.1	EQ	156
10.4.5.2	GE	157
10.4.5.3	GT	157
10.4.5.4	LE	158
10.4.5.5	LT	158
10.4.5.6	NE	159
10.5	Параметры процессов	159
10.5.1	Входы и выходы	159
10.5.2	Уставки и фактические значения ПЛК	165
10.5.3	Уставки и действительные значения шины	168
10.5.4	ControlBox и ParameterBox	170
10.5.5	Информационный параметр	171
10.5.6	Ошибки ПЛК	174
10.5.7	Параметры ПЛК	175
10.6	Языки	176
10.6.1	Список инструкций (AWL / IL)	176
10.6.1.1	Общие сведения	176
10.6.2	Структурированный текст (ST)	179
10.6.2.1	Общие сведения	179
10.6.2.2	Операторы	181
10.7	Переходы	184
10.7.1	JMP	184
10.7.2	JMPC	184
10.7.3	JMPCN	184
10.8	Конвертация типов	185
10.8.1	BOOL_TO_BYTE	185
10.8.2	BYTE_TO_BOOL	185
10.8.3	BYTE_TO_INT	186
10.8.4	DINT_TO_INT	186
10.8.5	INT_TO_BYTE	187
10.8.6	INT_TO_DINT	187
10.9	Сообщения о неисправностях ПЛК	187
<b>11</b>	<b>Режим проекта</b>	<b>189</b>
11.1	Общая информация	189
11.2	Интерфейс «человек-машина» (HMI)	190
11.3	Сохранение и восстановление	191
11.4	Загрузка проекта	192
<b>12</b>	<b>Встроенное программное обеспечение</b>	<b>194</b>
12.1	Обновление встроенного программного обеспечения	194
12.2	Программа для обновления встроенного ПО	196
12.3	Обновление встроенного ПО через системную шину	199
<b>13</b>	<b>Settings</b>	<b>202</b>
13.1	Окно программы	202
13.2	Device overview	203
13.3	Control	204
13.4	Project	205
13.5	Directories	206
13.6	Редактор макросов	207
13.7	Parameter	208
13.8	ПЛК	208
<b>14</b>	<b>Сообщения</b>	<b>209</b>

---

14.1	Ошибки и указания.....	209
<b>15</b>	<b>Getriebbau Nord.....</b>	<b>215</b>
15.1	История предприятия.....	216
15.2	Преобразователи частоты.....	218
15.2.1	SK 135E.....	218
15.2.2	SK 180E.....	219
15.2.3	SK 200E.....	220
15.2.4	SK 500E.....	222



## 1 Введение

### 1.1 О программе NORD CON

NORD CON — программа для работы с параметрами (4 "Параметризация") и управления (5.1 "Информация о функциях управления") преобразователями частоты и другим оборудованием Getriebebau NORD.

NORD CON может одновременно опрашивать до 31 преобразователей частоты через встроенный интерфейс RS485. Обмен данными с преобразователями осуществляется через последовательный интерфейс компьютера.

С помощью программы, установленной на ПК, можно также выполнять пробные запуски или ввод в эксплуатацию подключенных преобразователей и отслеживать состояние устройств. С помощью макросов (8 "Редактор макросов") можно запрограммировать некоторые технологические процессы или их этапы.

NORD CON позволяет создавать, документировать и сохранять значения параметров каждого преобразователя. а также считывать настройки параметров с преобразователя или передавать на преобразователь новые. Пользователь может создать базу данных параметров и работать с ними автономно, не подключаясь к преобразователю.

Кроме того, программа позволяет управлять подключенными преобразователями дистанционно. В режиме дистанционного управления (6 "Дистанционное управление") управление преобразователем выполняется с пульта управления, эмулируемого на ПК, что позволяет работать с устройствами, доступ к которым затруднен или которые не имеют пульта управления

### 1.2 Работа с NORD CON



#### Информация

#### Последовательный интерфейс

Для работы в NORD CON необходим ПК с последовательным интерфейсом.

#### 1. Установка

Запустите программу установки NORD CON с прилагаемого компакт-диска или загрузите ее из интернета.

["http://www2.nord.com/cms/de/documentation/software/software-overview.jsp"](http://www2.nord.com/cms/de/documentation/software/software-overview.jsp)

Укажите необходимую информацию и установите NORD CON в стандартную папку.

#### 2. Подключение

Если преобразователь частоты имеет порт RS232, его можно подключить к ПК напрямую. В этом случае потребуется только кабель последовательного интерфейса 1:1. Так можно подключить только один преобразователь.

В преобразователях NORD Частотный преобразователь имеется один встроенный порт RS485, доступный на управляющих клеммах. Через этот интерфейс можно подключить устройство к шине в режиме Master/Slave. В этом случае возможно объединение в сеть от 1 до 31 устройства. Для подключения NORD CON к шине потребуется переходник RS232 - RS485.

## **i** Информация

## Настройки USS

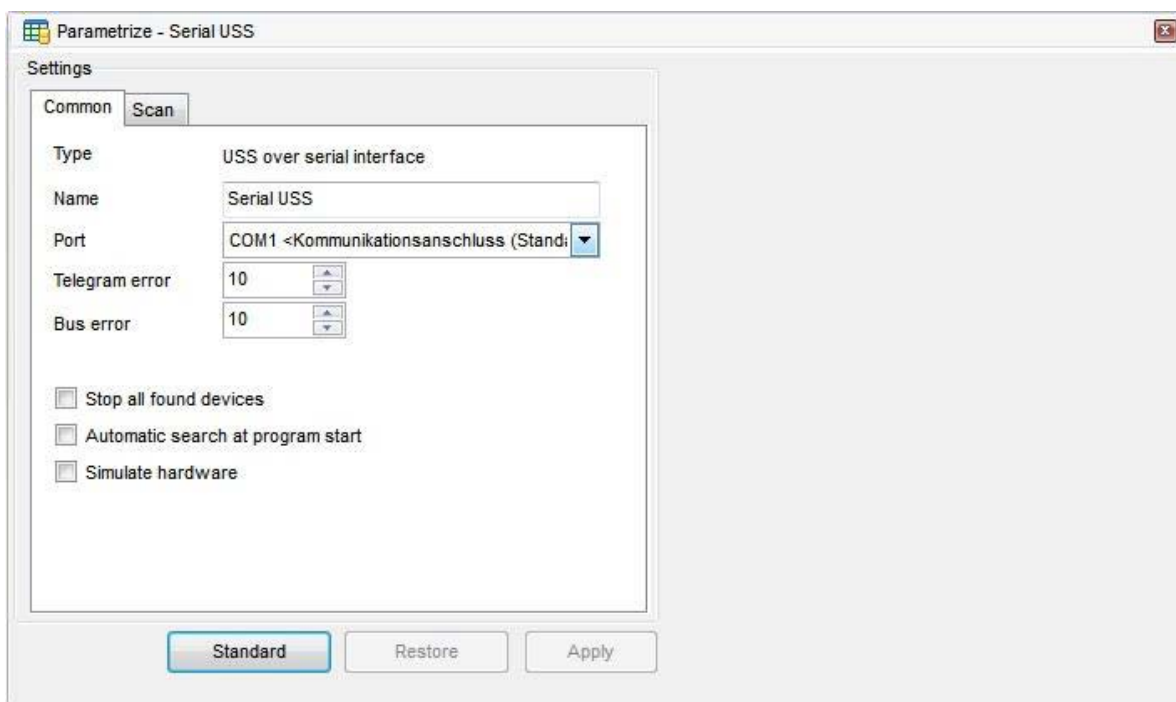
При одновременной эксплуатации нескольких устройств необходимо учитывать, что всем подключенным устройствам должны быть назначены уникальные адреса и одинаковые скорости передачи данных ( см. 3.1.2 "Busscan", а также руководство по эксплуатации, прилагаемое к соответствующему преобразователю).

### 3. Запуск NORD CON

Для запуска NORD CON откройте ссылку «NORD CON start» или нажмите «Start->Program->Nord->NORD CON 2.5->->NORD CON.

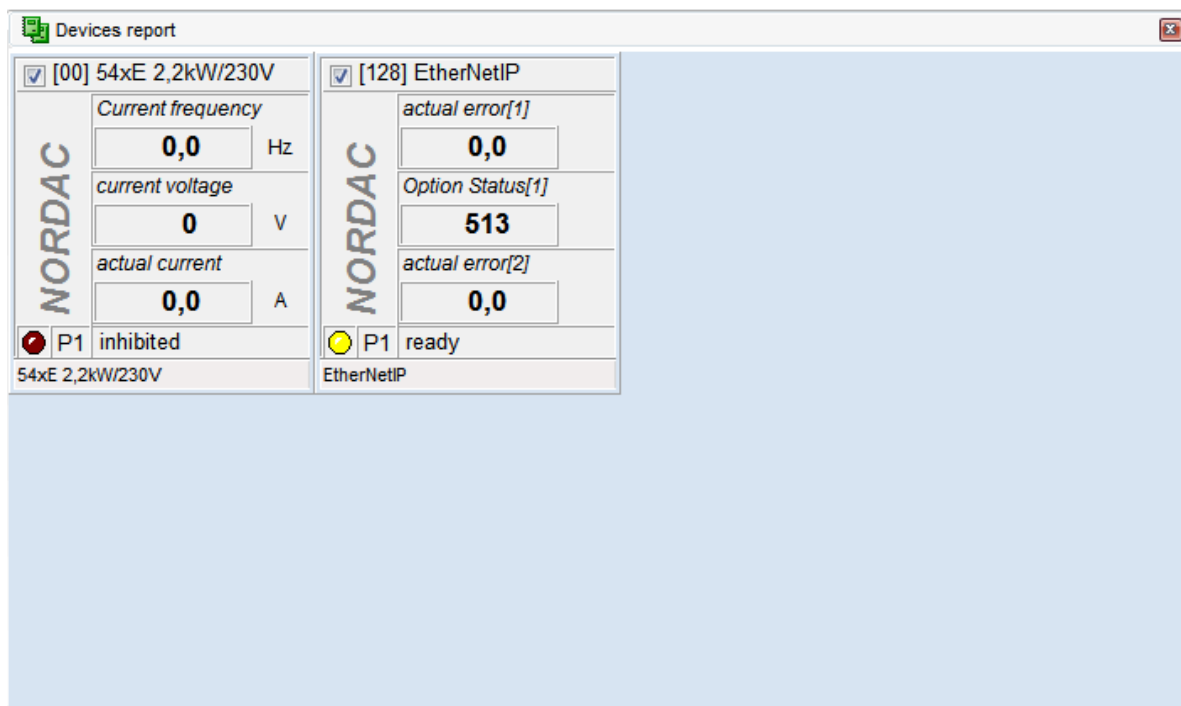
### 4. Настройка подключения

Отметьте в разделе проектов модуль, для которого нужно настроить сетевое подключение. Откройте окно параметров, открыв контекстное меню «Parameterise» или выбрав в меню «Device-> Parameterise». В поле «Port» введите правильный номер порта. Чтобы сохранить сделанные в этом окне изменения, нажмите кнопку «Apply». Другие настройки менять не нужно. Окно можно закрыть.



### 5. Сканирование шины

Функция сканирования шины позволяет искать подключенные к шине и готовые к эксплуатации устройства. Все обнаруженные устройства выводятся в окне устройств в виде дерева. Выделенным является первое устройство в списке, на нем можно выполнить все функции, которые оно поддерживает.



## 6. Работа с устройствами

Чтобы выбрать устройство, нажмите на название устройства в списке устройств или в 2.4 "Вид «Project»". Из контекстного меню в дереве устройств, из 2.3 "Панели инструментов" или в меню «Device» можно вызвать такие функции, как 5 "Управление" или 4 "Параметризация".

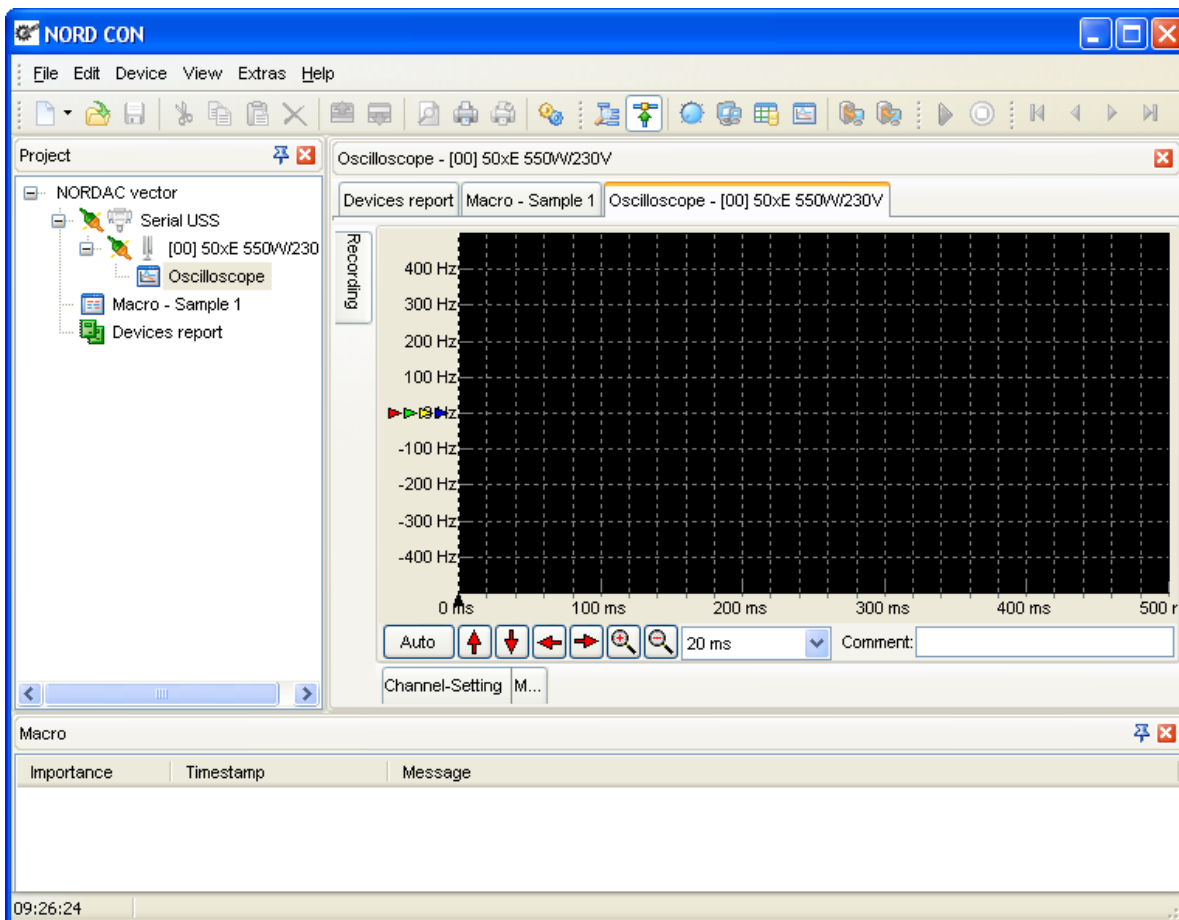
## 2 Окна и виды

### 2.1 Структура окна

Пользовательское окно содержит главное меню, панель инструментов, рабочую область и вспомогательные панели. В рабочей области выводятся разные окна (например, окно параметров или окно макросов), в которых можно редактировать содержимое полей. Вспомогательные панели можно расположить произвольным образом или прикрепить в боковой части рабочей области. Чтобы изменить положение прикрепленной панели, нажмите на строку заголовка и перетащите ее, удерживая кнопку мыши. Мышью укажите новое положение панели. Текущее положение и состояние панели можно определить по цветному прямоугольнику. Чтобы панель зафиксировалась в новом месте, отпустите левую кнопку мыши. Панель также можно прикрепить и открепить, дважды нажав мышью на строке заголовка. Структура окна запоминается при закрытии приложения и будет восстановлена при следующем запуске программы.

Окно программы разделено на следующие области:

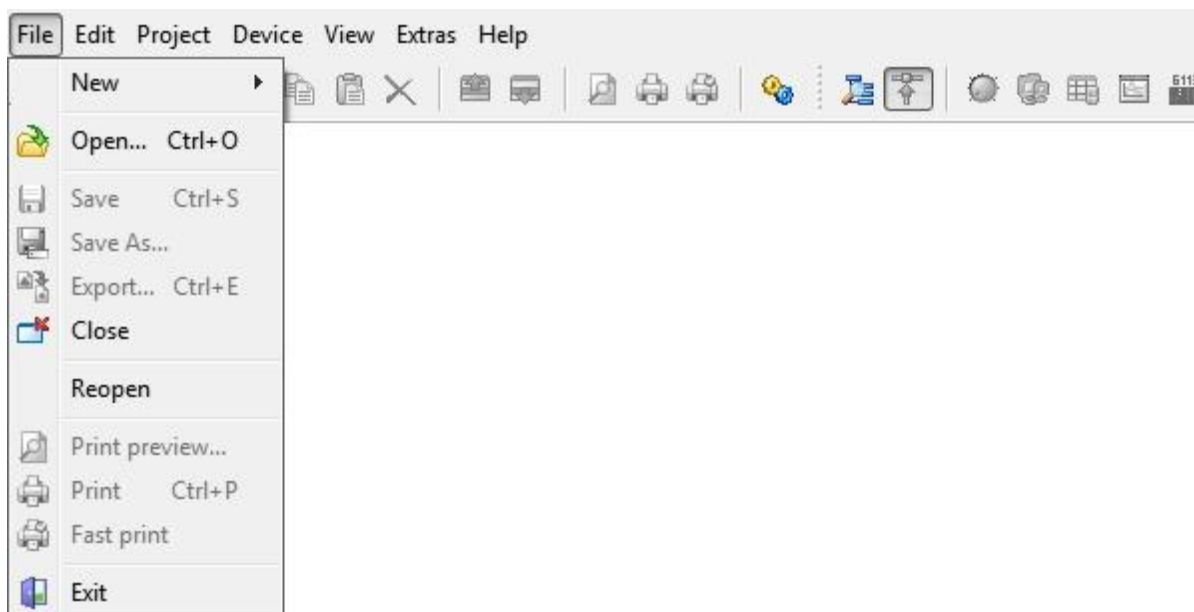
- 2.2 "Структура главного меню"
- 2.3 "Панели инструментов"
- Рабочая область
- 2.4 "Вид «Project»"
- 2.5 "Вкладка «Messages»"
- 2.6 "Вкладка «Remote»"












## 2.2 Структура главного меню

Главное меню содержит самые главные функции приложения. В редакторах в меню доступны функции, необходимые для редактирования. Все операции и функции в меню объединены в категории.

### 2.2.1 Категория «File»



Название операции	Комбинация клавиш	Значок	Описание
New dataset			Открывает окно параметров нового устройства. Прежде чем воспользоваться этой функцией, необходимо выбрать устройство.
New macro			Открывает редактор макросов с пустым макросом. Если в программе уже открыт редактор макросов, программа предложит сохранить открытый документ.  <b>Внимание:</b> В текущей версии можно открыть только одно окно редактора макросов!
PLC program			Открывает редактор ПЛК с пустым документом. Если в программе уже открыт редактор ПЛК, программа предложит сохранить открытый документ.
Open	Ctrl + O		Открывает диалоговое окно, которое позволяет выбрать и открыть сохраненный документ. Пользователь может ограничить число отображаемых документов, отсортировав документы по нужному типу. Поддерживаются следующие типы документов: <ul style="list-style-type: none"> <li>• файлы параметров (*.ndbx, *.db (V1.27))</li> <li>• файлы осциллографа (*.scox, *.sco (V1.27))</li> <li>• макросы (*.ncmx, *.ncm (V1.27))</li> <li>• файлы ПЛК (*.awlx, *.awl, *.nstx)</li> </ul>


Название операции	Комбинация клавиш	Значок	Описание
Save	Ctrl + S		Сохраняет текущий документ. Действие применяется к документу, открытому в окне редактора. Параметры операции зависят от типа редактора.
Save as...			Сохраняет текущий документ под новым именем. Действие применяется к документу, открытому в окне редактора. Параметры операции зависят от типа редактора.
Export	Ctrl + E		Экспортирует в файл данные из активного редактора. Действие применяется к документу, открытому в окне редактора. Параметры операции зависят от типа редактора.
Reopen			Содержит вложенное меню со списком документов, которые открывались последними. Журнал последних документов содержит 5 записей. При нажатии на любую из этих записей открывается соответствующий документ.
Print	Ctrl + P		Выводит на печать содержимое текущего редактора. Прежде чем отправить на печать, программа предлагает пользователю указать некоторые параметры печати.  <b>Примечание:</b> Параметры операции зависят от типа редактора. Эта операция недоступна, если она не поддерживается редактором или в программе нет открытых редакторов.
Print preview...			В активном редакторе открывает окно предварительного просмотра. В разных редакторах окно предварительного просмотра имеет разную структуру. Эта операция недоступна, если она не поддерживается редактором или в программе нет открытых редакторов.
Quit/Exit			Закрывает приложение.

## Информация

Этот пункт недоступен, если редактор не поддерживает операцию или нет открытых редакторов.

### 2.2.2 Категория «Edit»

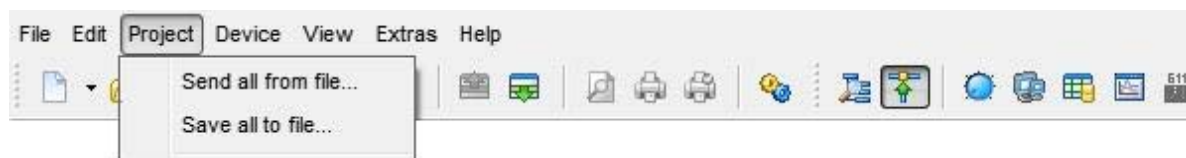




Название операции	Комбинация клавиш	Значок	Описание
Undo	Ctrl + Z		Позволяет отменить последнее выполненное действие. Действие применяется к документу, открытому в окне редактора. Параметры операции зависят от типа редактора.
Cut	Ctrl + X		Вырезает отмеченный объект и сохраняет его в буфере обмена Windows. Действие применяется к активному элементу управления. Параметры операции зависят от типа элемента управления.
Copy	Ctrl + C		Копирует отмеченный объект и сохраняет его в буфере обмена Windows. Действие применяется к активному элементу управления. Параметры операции зависят от типа элемента управления.
Paste	Ctrl + V		<p>Вставляет содержимое буфера обмена Windows в выбранное место. Действие применяется к активному элементу управления. Параметры операции зависят от типа элемента управления.</p> <p><b>Примечание.</b> Эта операция недоступна, если она не поддерживается текущим элементом управления или содержимое буфера обмена Windows нельзя вставить по каким-либо причинам.</p>
Delete	Ctrl + Del		Удаляет выбранный объект. Действие применяется к активному элементу управления. Параметры операции зависят от типа элемента управления.
Select all	Ctrl + A		Позволяет выбрать все объекты в активном элементе управления.
Replace...	Ctrl + H		Ищет указанный текст и заменяет найденный текст на другой. В диалоговом окне можно указать нужные настройки поиска и замены.
Up	Ctrl + U		Перемещает выбранный объект на одну позицию вверх.
Down	Ctrl + D		Перемещает выбранный объект на одну позицию вниз.

### Информация

Этот пункт недоступен, если редактор не поддерживает операцию или нет открытых редакторов.

### 2.2.3 Категория «Project»

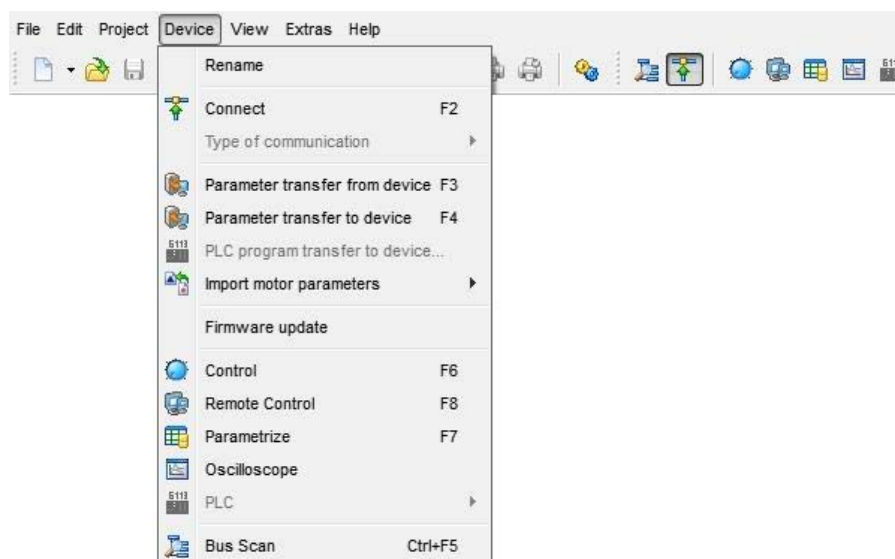


Название операции	Комбинация клавиш	Значок	Описание
Save all from file...			Загружает параметры всех обнаруженных устройств и сохраняет их в файле.
Send all from file			Открывает файл и передает сохраненные в нем параметры на устройства.

#### Информация










Этот пункт недоступен, если редактор не поддерживает операцию или нет открытых редакторов.

### 2.2.4 Категория «Device»



Название операции	Комбинация клавиш	Значок	Описание
Rename			Этот пункт меню позволяет переименовать устройство.
Connect	F2		Устанавливает или разрывает подключение с выбранным устройством.
Communication type/serial USS			Переключает модуль обмена данными в режим «serial USS». При изменении типа подключения из списка устройств будут удалены все устройства!
Communication			Переключает модуль обмена данными в режим

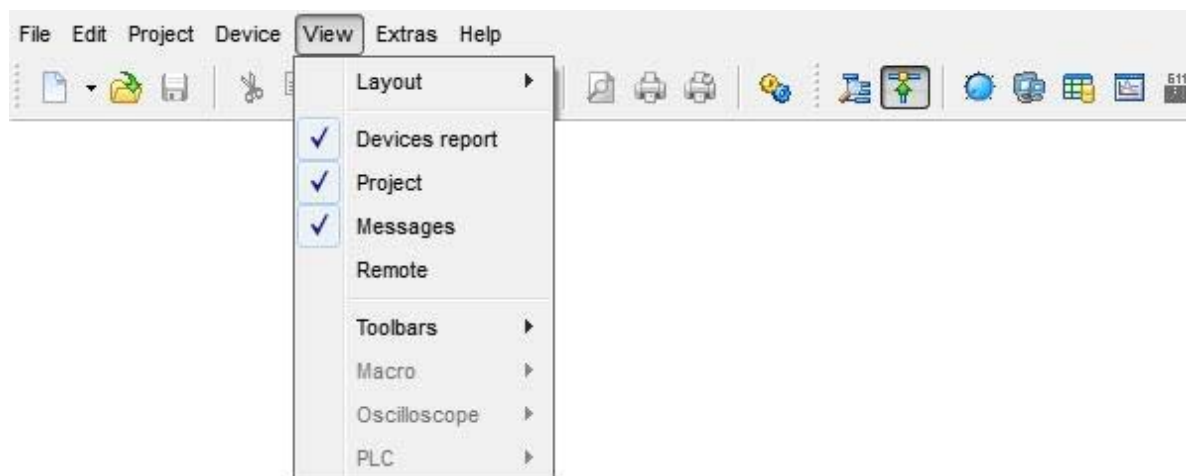


Название операции	Комбинация клавиш	Значок	Описание
type/Ethernet			«Ethernet». При изменении типа подключения из списка устройств будут удалены все устройства!
Parameter upload from frequency inverter	F3		Запускает загрузку параметров из устройства на ПК.
Parameter download to device	F4		Запускает передачу параметров из ПК на устройство.
Transfer PLC program to device			Передает сохраненную программу ПЛК на выбранное устройство.
Import motor parameters			Позволяет импортировать данные двигателя из внешнего источника. Если в окне выбора выбран файл с параметрами двигателя (*.csv), выводятся все двигатели, информация о которых содержится в файле. Пользователь выбирает в списке нужный набор данных и подтверждает действие, нажав ОК. После этого выполняется копирование всех параметров на выбранное устройство. Если импорт выполняется из окна параметров, значения импортируются в окно параметров и на устройство не передаются. Чтобы скопировать эти значения на устройство, необходимо запустить функцию передачи.
Update firmware			Запускает загрузку обновления для встроенной программы устройства.
Control	F6		Открывает в рабочей области окно «Control» для выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.
Remote control	F8		Открывает в окне «View and Control» окно «Remote Control» для выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.
Parameter setup	F7		Открывает в рабочей области окно «Parameters» с параметрами выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.
Oscilloscope			Открывает в рабочей области окно «Oscilloscope» с данными выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.
PLC			Открывает в рабочей области окно «PLC» с данными выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.
Bus scan	Ctrl + F5		Выполняет сканирование сети для выбранного режима подключения.  <b>Внимание.</b> При запуске этой функции все устройства будут удалены из списка и все окна, связанные с устройствами, будут закрыты!

### Информация

Этот пункт недоступен, если редактор не поддерживает операцию или нет открытых редакторов.

## 2.2.5 Категория «View»



Название операции	Комбинация клавиш	Описание
Layout -> Standard		Позволяет восстановить стандартный вид приложения во всех разделах. Положение окна редактора, однако, не меняется.
Layout -> Standard all windows		Позволяет восстановить стандартный вид приложения во всех окнах, в том числе в окнах редакторов.
Device report		Закрывает или открывает список устройств.
2.4 "Вид «Project»"		Закрывает или открывает вид «Project».
2.5 "Вкладка «Messages»"		Закрывает или открывает вид «Log».
2.6 "Вкладка «Remote»"		Закрывает или открывает вид «Remote control».
Toolbar->Standard		Закрывает или открывает панель инструментов «Standard».
Toolbar->Device		Закрывает или открывает панель инструментов «Device».
Toolbar ->Start		Закрывает или открывает панель инструментов «Start».
Macro		Открывает вложенное меню. В этом меню в категории «View» доступны функции, предназначенные для работы в редакторе макросов. Состояния, операции и действия применяются к активному редактору макросов. Функции недоступны, если

		редактор не открыт.
Oscilloscope		Открывает вложенное меню. В этом меню в категории «View» доступны функции для работы с осциллографом. Состояния, операции и действия применяются к активному окну осциллографа. Функции недоступны, если окно осциллографа не открыто.

### Информация

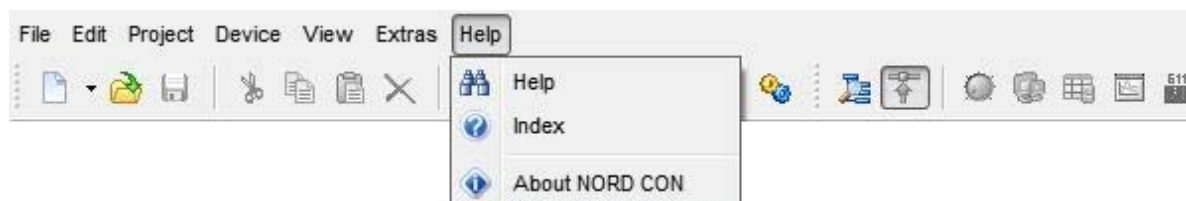
Этот пункт недоступен, если редактор не поддерживает операцию или нет открытых редакторов.

#### 2.2.6 Категория «Extras»



Название операции	Комбинация клавиш	Описание
Communication settings		Открывает редактор настроек для выбранного режима обмена данными.
Settings		Открывает окно, в котором можно изменить общие настройки программы.
Log		Открывает вложенное меню, в котором содержатся все функции окна «log».

#### 2.2.7 Категория «Help»










Название операции	Комбинация клавиш	Описание
Help	F1	Открывает раздел «Contents» интерактивной справки.
Index		Открывает раздел «Index» интерактивной справки.
About NORD CON		Открывает окно с информацией о программе.









## 2.3 Панели инструментов

На панели инструментов размещены значки наиболее часто используемых функций. Чтобы выполнить нужную функцию, нажмите мышью на соответствующий значок на панели инструментов.




**В программе имеются следующие панели инструментов:**







### 2.3.1 Стандартная панель

Название операции	Значок	Описание
New data set		Функция, которая открывает окно параметров нового устройства. Прежде чем воспользоваться этой функцией, необходимо выбрать устройство.
New Macro		Функция, которая открывает редактор макросов с пустым макросом. Если в программе уже открыт редактор макросов, программа предложит сохранить открытый документ.  <b>Внимание.</b> В текущей версии можно открыть только одно окно редактора макросов!
New PLC program		Функция, которая открывает редактор ПЛК с пустым документом. Если в программе уже открыт редактор ПЛК, программа предложит сохранить открытый документ.
Open		Открывает диалоговое окно, позволяющее выбрать и открыть сохраненный документ. Пользователь может ограничить число отображаемых документов, отсортировав документы по нужному типу. Поддерживаются следующие типы документов: <ul style="list-style-type: none"> <li>• файлы параметров V1.27 (*.db)</li> <li>• файлы параметров (*.ndbx)</li> <li>• файлы осциллографа V1.27 (*.sco)</li> <li>• файлы осциллографа V2.1 (*.scox)</li> <li>• макросы (*.ncmx)</li> <li>• макросы V1.27 (*.ncm)</li> <li>• программы ПЛК (*.awlх)</li> </ul>
Save		Действие применяется к документу, открытому в окне редактора. Параметры операции зависят от типа редактора. Эта операция недоступна, если она не поддерживается редактором или в программе нет открытых редакторов.
Cut		Действие применяется к активному элементу управления. Параметры операции зависят от типа элемента управления. Операция недоступна, если она не поддерживается активным окном.
Copy		Действие применяется к активному элементу управления. Параметры операции зависят от типа элемента управления. Операция недоступна, если она не поддерживается активным окном.

Название операции	Значок	Описание
Paste		Действие применяется к активному элементу управления. Параметры операции зависят от типа элемента управления. Операция недоступна, если она не поддерживается активным элементом управления.
Delete		Действие применяется к активному элементу управления. Параметры операции зависят от типа элемента управления. Операция недоступна, если она не поддерживается активным окном.
Up		Перемещает выбранную строчку или выбранное устройство на одну позицию вверх. Операция недоступна, если она не поддерживается активным окном.
Down		Действие применяется к активному элементу управления. Параметры операции зависят от типа элемента управления. Операция недоступна, если она не поддерживается активным окном.
Preview		Перемещает выбранную строчку или выбранное устройство на одну позицию вверх. Операция недоступна, если она не поддерживается активным окном.
Print		<p>Выводит на печать содержимое текущего редактора. Прежде чем отправить на печать, программа предлагает пользователю указать некоторые параметры печати.</p> <p><b>Примечание.</b> Параметры операции зависят от типа редактора. Эта операция недоступна, если она не поддерживается редактором или в программе нет открытых редакторов.</p>
Fast print		<p>Выводит на печать содержимое активного редактора, используя текущие настройки печати.</p> <p><b>Примечание.</b> Действие применяется к документу, открытому в активном окне редактора. Параметры операции зависят от типа редактора. Эта операция недоступна, если она не поддерживается редактором или в программе нет открытых редакторов.</p>
Settings		Функция «Settings» открывает диалоговое окно с общими настройками NORD CON

### 2.3.2 Устройство

Название операции	Значок	Описание
Bus scan		<p>Запускает функцию сканирования шины.</p> <p><b>Внимание.</b> При запуске этой функции все устройства будут удалены из списка и все окна, связанные с устройствами, будут закрыты!</p>
Connect		Устанавливает подключение с выбранным устройством.
Control		Открывает в рабочей области окно «Control» для выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.

Название операции	Значок	Описание
Remote		Открывает в окне «View and Control» окно «Remote» для выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.
Parameterise		Открывает в рабочей области окно «Parameter» с параметрами выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.  <b>Внимание.</b> В текущей версии можно открыть только одно окно параметров!
Oscilloscope		Открывает в рабочей области окно «Oscilloscope» с данными выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.
PLC		Открывает редактор ПЛК для выбранного устройства. Элемент недоступен, если выбранное устройство не поддерживает функции ПЛК.
Upload parameters from device		Запускает процесс передачи параметров с устройства на ПК.
Download parameters to device		Начинает копирование параметров с ПК на устройство.

### 2.3.3 Категория «Start»

Название операции	Комбинация клавиш	Значок	Описание
PLC settings			Открывает окно для конфигурирования ПЛК.
Compile	Shift + F7		Включает перевод программы ПЛК.
Programming	Shift + F8		Загружает программу ПЛК на устройство.
Run	F9		Запускает программу ПЛК или макрос. Действие применяется к документу, открытому в активном окне редактора. Параметры операции зависят от типа редактора.
Cancel	F11		Прекращает выполнение программы ПЛК или макроса. Действие применяется к документу, открытому в активном окне редактора. Параметры операции зависят от типа редактора.
Next	F12		Выполняет следующую команду. Действие применяется к документу, открытому в активном окне редактора. Параметры операции зависят от типа редактора.
Debug	Shift + F5		Запускает программу ПЛК в режиме отладки.

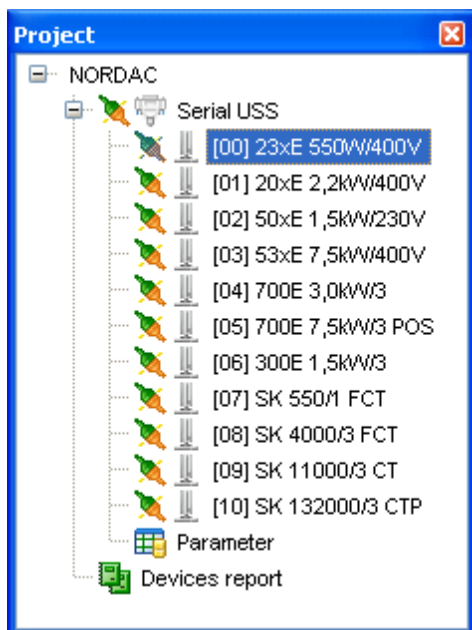
### Информация

Этот пункт недоступен, если редактор не поддерживает операцию или нет открытых редакторов.



## 2.4 Вид «Project»

В окне «Project» все устройства, принадлежащие одному проекту, выводятся в виде дерева. Окно можно открыть или закрыть, нажав в главном меню «View->Project». В дереве проекта отображаются все устройства, найденные в процессе сканирования. Переход между устройствами осуществляется с помощью мыши. Если окно позволяет вводить данные, устройство можно выбрать, используя также кнопки со стрелками «up» и «down». Если навести указатель мыши на устройство, отобразится окно подсказки с информацией о типе устройства и адресе шины. После выбора устройства пользователь может выполнить с ним любое действие, доступное в меню «Device», в панели инструментов или в контекстном меню. Если действие

недоступно, выбранное устройство не поддерживает соответствующую функцию. 2.4.1 "Структура контекстного меню" можно вызвать, нажав на окне вида правой кнопкой мыши.

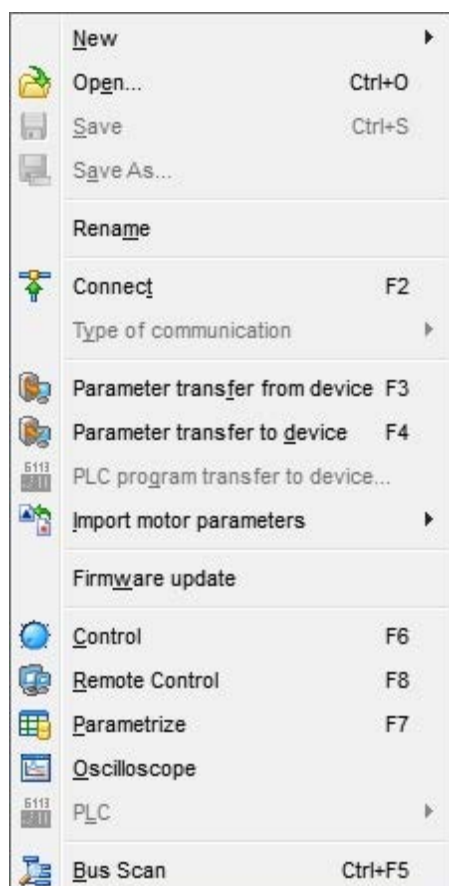


### Состояние устройства

-  Устройство подключено (онлайн)
-  Устройство не подключено (офлайн)

### 2.4.1 Структура контекстного меню

На иллюстрации изображено контекстное меню, открытое в окне проекта. Контекстное меню всегда относится к отмеченному элементу дерева проекта.







Название операции	Комбинация клавиш	Описание
New parameter set		Открывает окно параметров нового устройства. Прежде чем воспользоваться этой функцией, необходимо выбрать устройство.
New macro		Открывает редактор макросов с пустым макросом. Если в программе уже открыт редактор макросов, программа предложит сохранить открытый документ.  <b>Внимание:</b> В текущей версии можно открыть только одно окно редактора макросов!
New PLC program		Открывает редактор ПЛК с пустым документом. Если в программе уже открыт редактор ПЛК, программа предложит сохранить открытый документ.
Open	Ctrl + O	Открывает редактор ПЛК с пустым документом. Если в программе уже открыт редактор ПЛК, программа предложит сохранить открытый документ.
Save	Ctrl + S	Сохраняет текущий документ. Действие применяется к документу, открытому в окне редактора. Параметры операции зависят от типа редактора.
Save as ....		Сохраняет текущий документ под новым именем. Действие применяется к документу, открытому в окне редактора. Параметры операции зависят от типа редактора.
Rename		Открывает поле, в котором можно указать новое имя устройства.
Connect	F2	Устанавливает или разрывает подключение с выбранным







Название операции	Комбинация клавиш	Описание
		устройством.
Communication type/serial USS		Переключает модуль обмена данными в режим «serial USS». При изменении типа подключения из списка устройств будут удалены все устройства!
Communication type/Ethernet		Переключает модуль обмена данными в режим «Ethernet». При изменении типа подключения из списка устройств будут удалены все устройства!
Parameter upload from frequency inverter	F3	Запускает загрузку параметров из устройства на ПК.
Parameter download to device	F4	Запускает передачу параметров из ПК на устройство.
Update firmware		Загружает обновление для встроенного ПО устройства.
Control	F6	Открывает в рабочей области окно «Control» для выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.
Remote control	F8	Открывает в окне «View and Control» окно «Remote» для выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.
Parameter setup	F7	Открывает в рабочей области окно «Parameter» с параметрами выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.
Oscilloscope		Открывает в рабочей области окно «Oscilloscope» с данными выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.
PLC		Открывает в рабочей области окно «PLC» с данными выбранного устройства. Если окно уже открыто, оно выводится поверх других окон.
Bus scan	Ctrl + F5	Запускает функцию сканирования шины.  <b>Внимание.</b> При запуске этой функции все устройства будут удалены из списка и все окна, связанные с устройствами, будут закрыты!

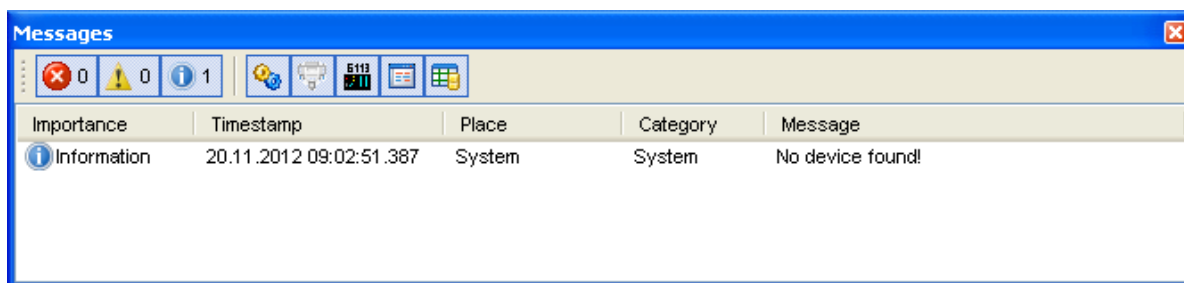
### 2.5 Вкладка «Messages»

Вкладка содержит список всех уведомлений программы «NORD CON». Уведомления по умолчанию упорядочены по дате. Их можно также отсортировать по другому полю, нажав на заголовок соответствующего столбца. Имеются следующие фильтры:

Фильтр	Значок	Описание
Error		Если включен этот фильтр, отображаются все сообщения об ошибках. На кнопке также выводится количество сообщений этого типа.
Warning		Если включен этот фильтр, отображаются все предупреждения. На кнопках также выводится количество сообщений этого типа.
Information		Если включен этот фильтр, отображаются все информационные сообщения. На кнопках также выводится количество сообщений этого типа.
System		Если включен этот фильтр, выводятся все системные сообщения (категории

		«System»).
Communication		Если включен этот фильтр, выводятся все сообщения категории «Communications»).
PLC		Если включен этот фильтр, выводятся все сообщения категории «PLC»).
Macro		Если включен этот фильтр, выводятся все сообщения категории «Macro»).
Parameter		Если включен этот фильтр, выводятся все сообщения категории «Parameter»).

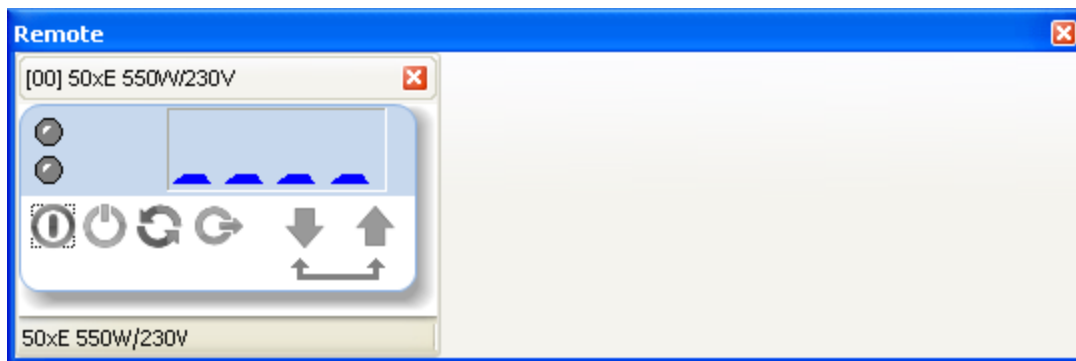
Сообщения можно удалить или сохранить, используя команды, доступные в контекстном меню (правая кнопка мыши). Эти действия можно также выполнить из главного меню («Extras/Messages»).



Название операции	Описание
Delete	Удаление сообщений.
Save	Сохранение сообщений в файле.

## 2.6 Вкладка «Remote»

Из панели «Remote» можно получить доступ ко всем окнам функции 2.6 "Вкладка «Remote»". Панель открывается автоматически при открытии первого окна и закрывается после выхода из последнего. Она, как и другие панели, может быть прикреплена к рабочей области. Если панель была закрыта пользователем, ее можно снова открыть, нажав «Remote» в окне соответствующего устройства. Новое окно всегда выводится в левой части текущего окна. Пользователь может прикрепить или открепить его с помощью мыши. Если панель в первый раз была открыта через меню «View->Remote», автоматически открывается окно «Remote» устройства, выбранного в списке.


**И** **Информация**
**Дистанционное управление**

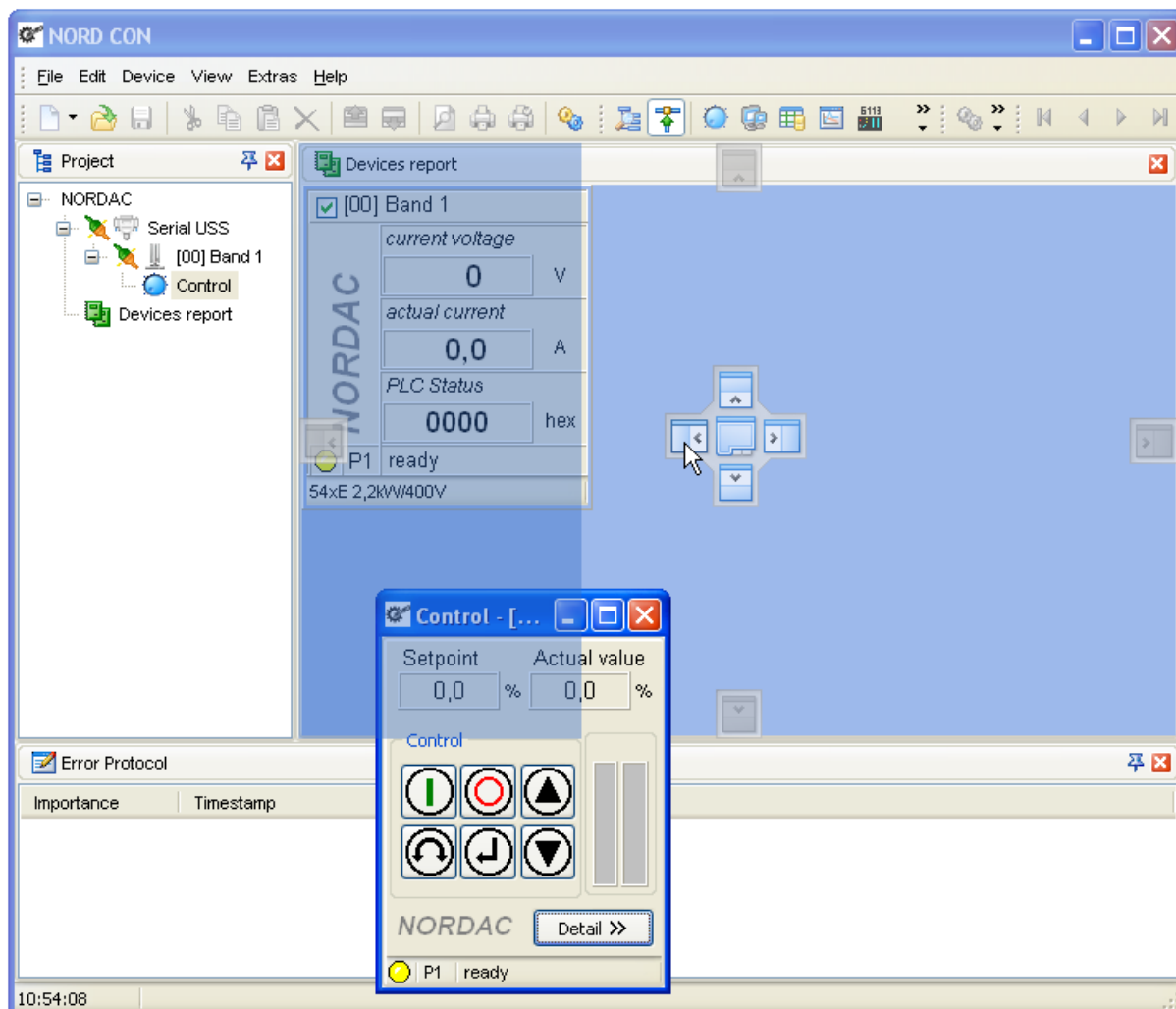
Окно «Remote» можно прикрепить к панели «Remote».

**2.7 Прикрепление и открепление окон**

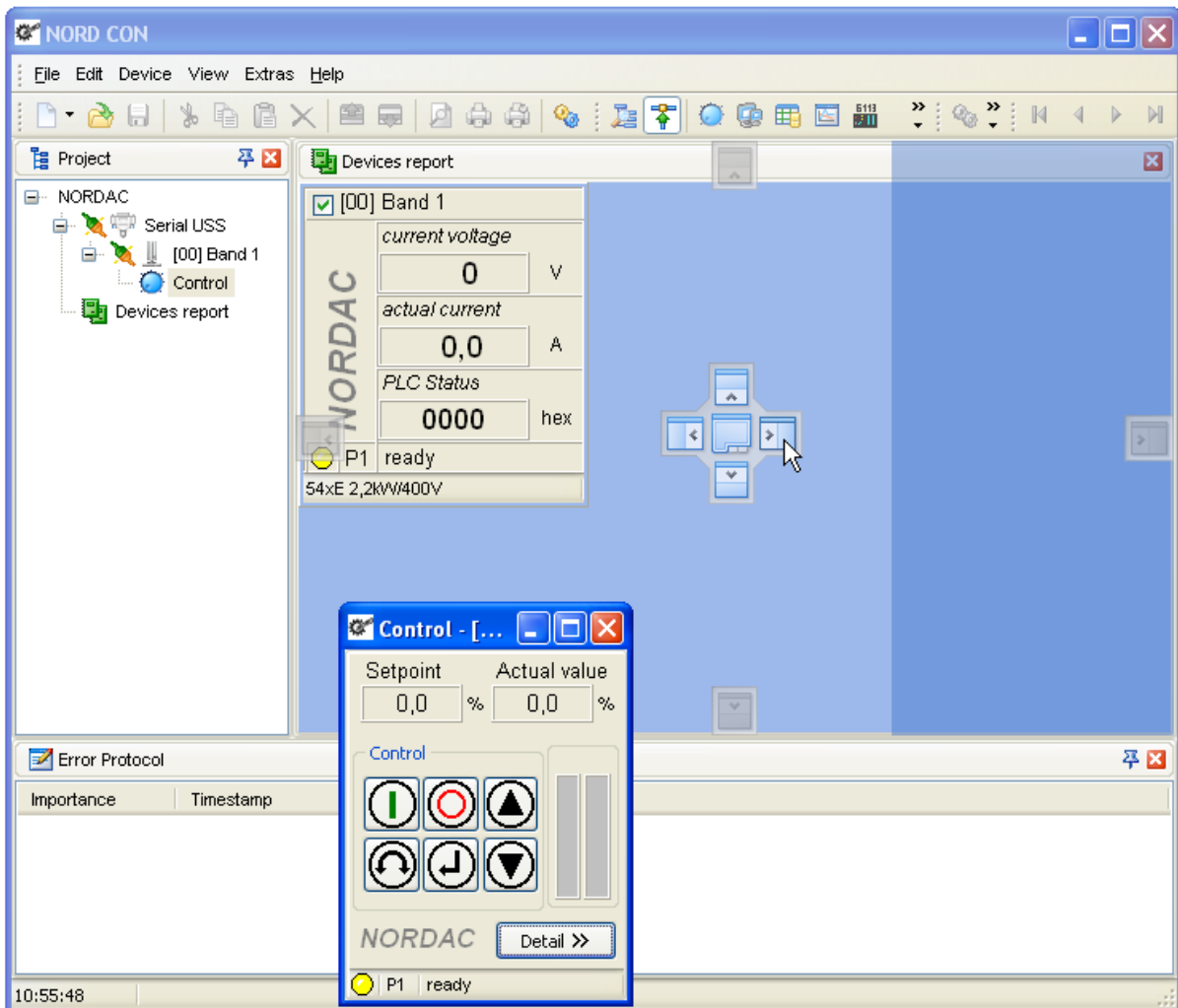
NORD CON имеет новый дизайн: теперь пользователи имеют возможность менять раскладку интерфейса в зависимости от своих предпочтений. Программа позволяет пользователю открепить панель или окно редактора и разместить их в любом месте экрана. Чтобы сделать это, необходимо нажать левой кнопкой мыши на строке заголовка соответствующего элемента и перетащить появившийся цветной прямоугольник в нужное место экрана. Элемент зафиксируется в новом положении после того, как будет отпущена кнопка мыши. Положение окна редактора можно изменить, используя контекстное меню, которое открывается при нажатии правой кнопки мыши на строке заголовка. Прикрепление выполняется аналогичным образом. Цветной прямоугольник указывает на текущее положение элемента на экране. Положение некоторых окон нежелательно менять, поэтому, как правило, применяются следующие правила

Тип окна	Правило расположения
Панель главного окна (например, панель проекта, журнала ошибок, управления и наблюдения)	Панели главного окна можно прикрепить только рядом с правым, левым или нижним краем рабочей области. Положение внутренней части окна можно менять произвольным образом.
Окно редактора (например, редактора макросов, параметров, осциллографа)	Окно редактора можно прикрепить только в рабочей области. Окно может располагаться сверху или снизу или выводиться в виде вкладки.
Панели окна макросов	Панели редактора макросов можно прикрепить только в окне редактора макросов. Они могут располагаться слева, справа или снизу окна. Внутренняя часть окна может располагаться произвольным образом.
Панели осциллографа	Панели редактора макросов можно прикрепить только в окне редактора макросов. Они могут располагаться слева, справа или снизу окна. Внутренняя часть окна может располагаться произвольным образом.
Окна «Remote»	Окна «Remote» можно прикрепить только в окне «Remote». Они, как правило, располагаются слева.

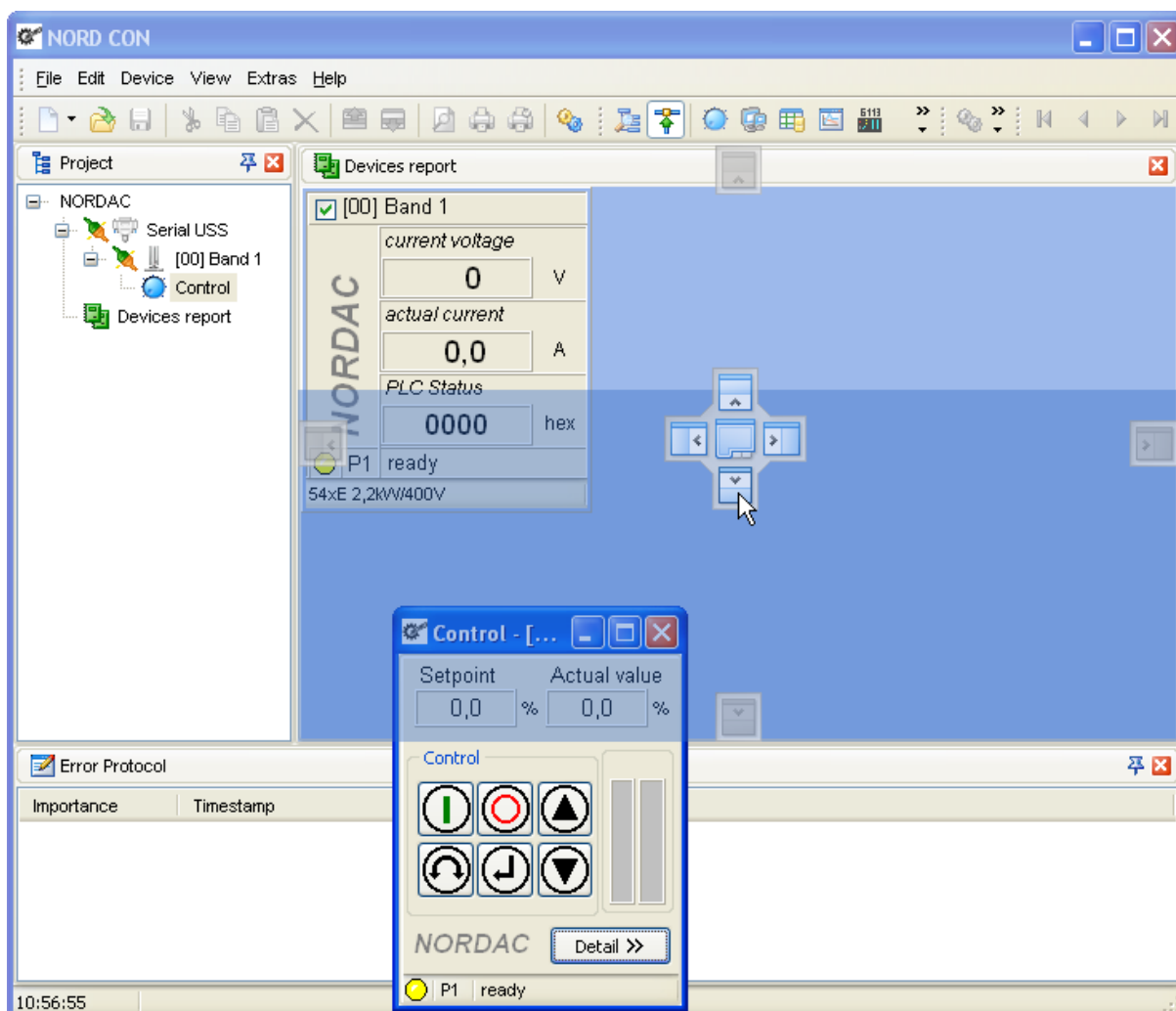
**Положение, в котором окно прикрепляется к левому краю рабочей области**



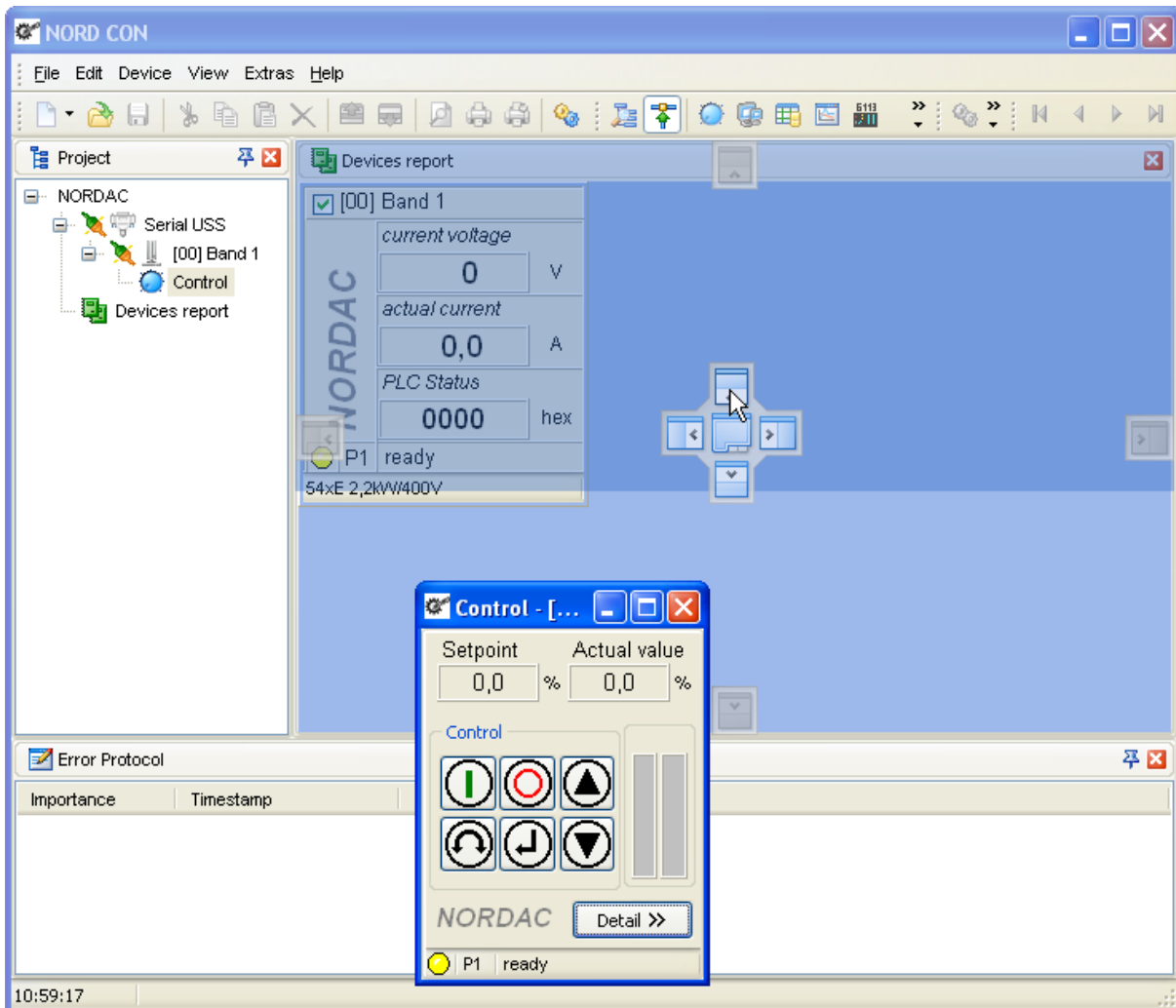
Положение, в котором окно прикрепляется к правому краю рабочей области



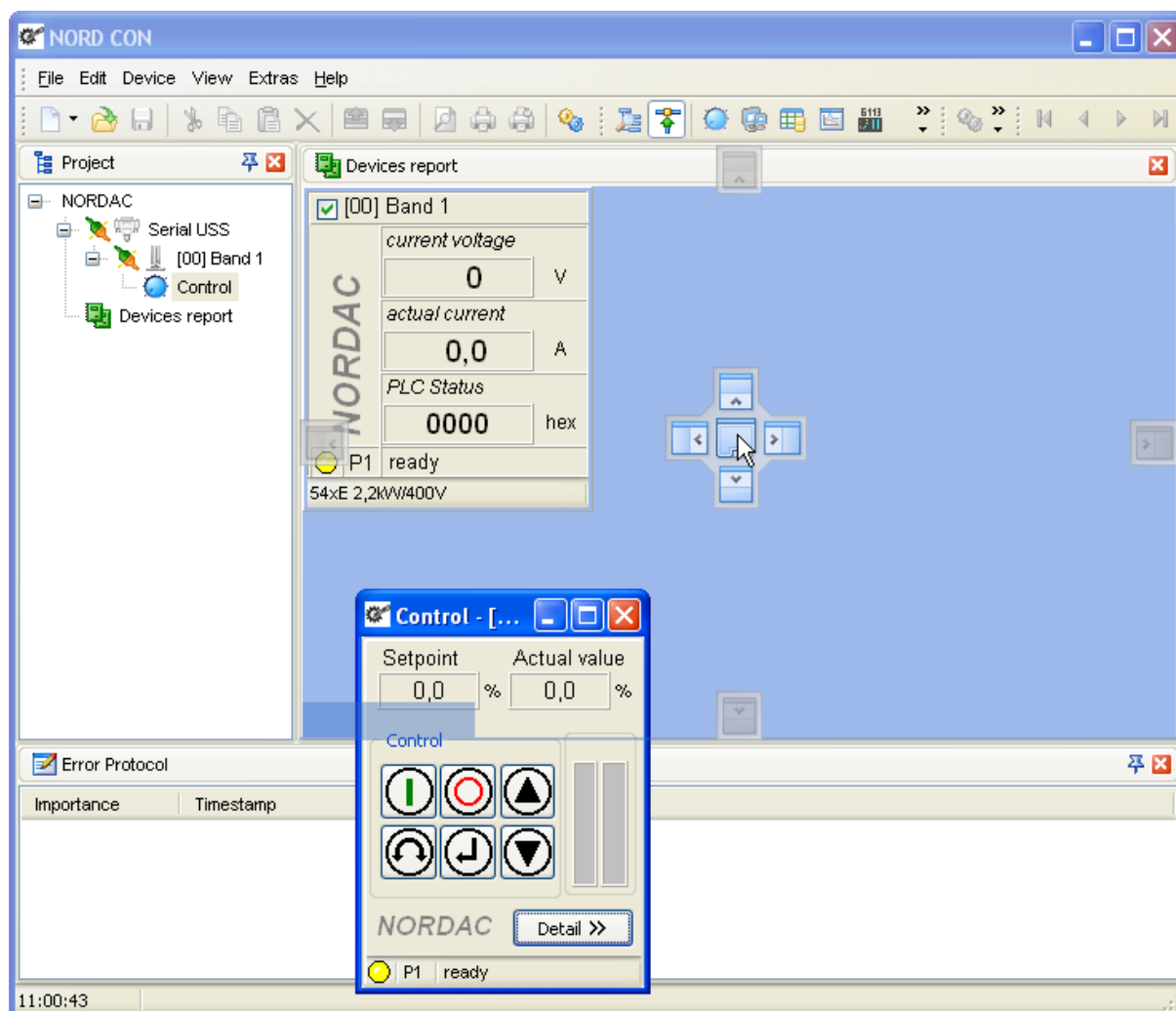
**Положение, в котором окно прикрепляется к нижнему краю рабочей области**



Положение, в котором окно прикрепляется к верхнему краю рабочей области



### Прикрепление окна в виде вкладки





## 3 Передача данных

Чтобы установить соединение с устройством, необходимо в проект добавить соответствующий модуль передачи данных. После установки в проект по умолчанию добавляется модуль «USS». Настройки модуля можно изменить, используя функцию «Parameterise».

**В настоящее время поддерживаются следующие типы модулей передачи данных:**

### 3.1 USS

#### 3.1.1 Общие настройки

##### **Name**

Поле, в котором можно указать имя модуля передачи данных.

##### **Port**

Список, в котором можно выбрать COM-порт, к которому подключен преобразователь частоты.

##### **Telegram error**

Поле, в котором пользователь может задать допустимое число ошибок при передаче телеграмм. Ошибка означает, что содержимое телеграммы является неверным, т.е. ответ не соответствует запросу параметров. Как правило, ответ на запрос поступает после двух телеграмм. Число допустимых ошибок при передаче телеграмм — это число попыток передачи до генерации сообщения об ошибке.

##### **Bus error**

Поле, в котором пользователь может задать допустимое число ошибок шины. Ошибка шины возникает, входящая или выходящая телеграмма является неверной. Поврежденная телеграмма не используется. В этом поле можно указать число поврежденных телеграмм, при достижении которого будет генерироваться ошибка. Это значение должно быть больше стандартного, если эксплуатация оборудования производится в условиях сильных помех.

##### **Stop all detected devices**

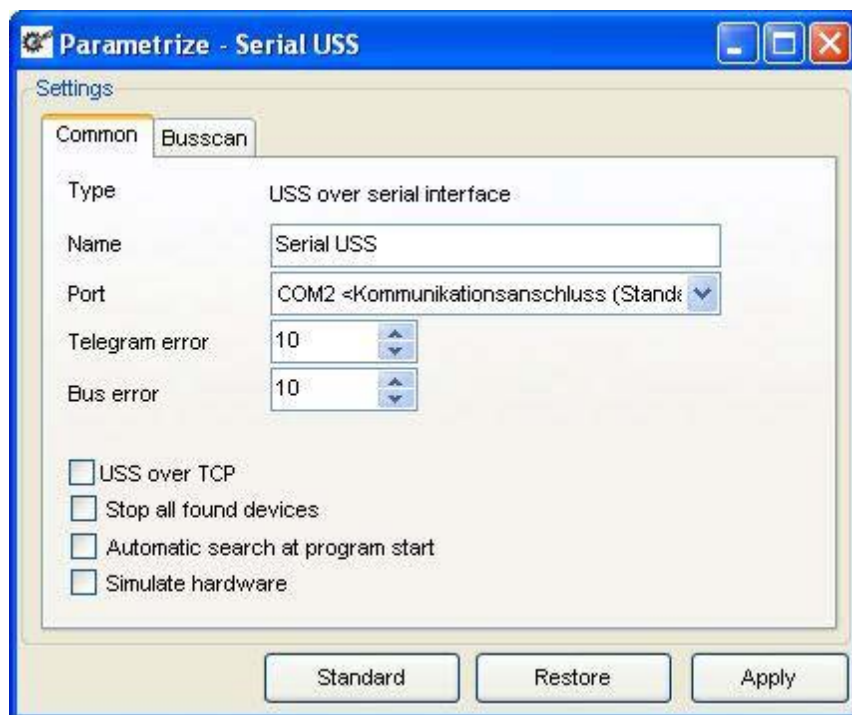
Если активен этот параметр, программа NORD CON, завершив поиск устройств, отправляет на все обнаруженные устройства команду «Disable». Останавливаются все устройства, управляемые через шину (P509).

##### **Automatic device search after program start**

Этот параметр включает (или отключает) автоматический поиск устройств после запуска программы. Если эта опция включена, при запуске NORD CON выполняется автоматический поиск устройств.

### Simulate hardware

Это параметр включает (или отключает) эмуляцию подключенных аппаратных средств.



### Информация

### Изменения

Изменения сохраняются в программе только после нажатия кнопки «Adopt». Кнопка «Restore» позволяет восстановить стандартные значения параметров.

### 3.1.2 Busscan

#### Baud rate

Список, в котором пользователь может выбрать скорость передачи данных через последовательный интерфейс. Соответствующее значение должно быть установлено также на преобразователе частоты. При работе с несколькими устройствами на всех устройствах должна быть задана одинаковая скорость передачи данных. Скорости передачи более 115 200 бит/с являются специальными и не поддерживаются некоторыми устройствами.

### Информация

### Проблемы подключения

Устаревшие последовательные интерфейсы, имеющиеся на некоторых ПК, не поддерживают определенные значения скорости передачи данных. В таких случаях невозможно установить соединение с устройством.

#### Bus-Scan with all baud rates

Этот параметр позволяет выполнять сканирование шины на разных скоростях передачи данных. Например, если скорость передачи данных устройства неизвестна, устройство можно найти, используя сканирование со всеми возможными скоростями.

#### Starting baud rate

В списке можно задать значение скорости, с которого будет начинаться сканирование шины.

#### Starting address

Здесь можно указать адрес USS, с которого NORD CON начнет поиск подключенных преобразователей частоты. В этом случае не будут найдены устройства, адреса которых меньше указанного значения.

#### End address

Здесь можно указать адрес USS, достигнув которого NORD CON прекратит поиск подключенных преобразователей частоты. В этом случае не будут найдены устройства, адреса которых больше этого значения.

#### Execute bus scan with all baud rates

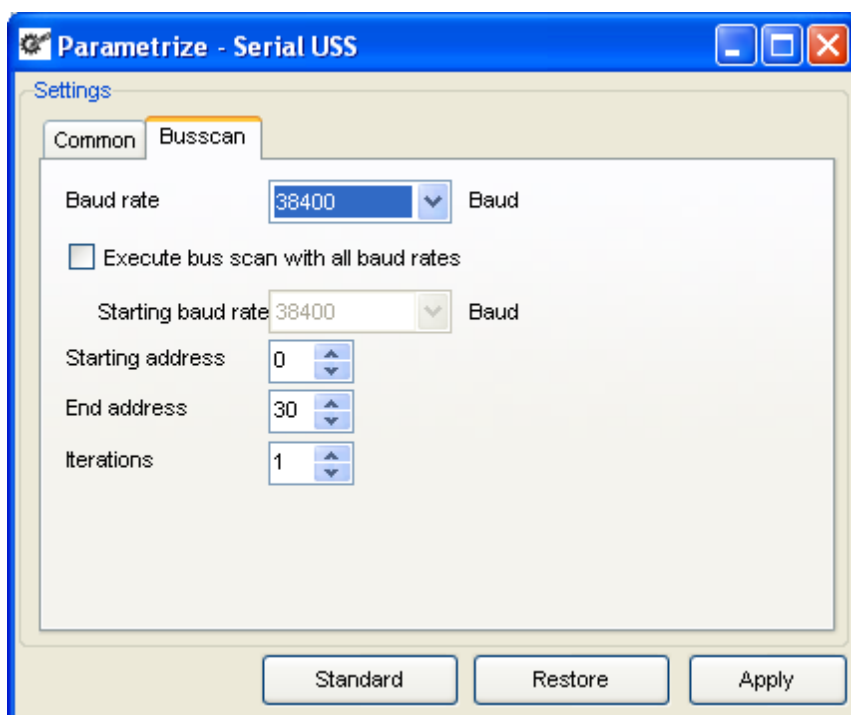
Этот параметр позволяет выполнять сканирование шины на разных скоростях передачи данных. Например, если скорость передачи данных устройства неизвестна, устройство можно найти, используя сканирование со всеми возможными скоростями.

#### Stop all connected devices

Эта функция, если активна, останавливает все найденные устройства (снятие сигнала разблокировки). Если опция активна, все подключенные к шине преобразователи частоты с активным сигналом разблокировки останавливаются при сканировании шины.

#### Automatic device search after start of program

Этот параметр включает (или отключает) автоматический поиск устройств после запуска программы. Если эта опция включена, NORD CON производит при запуске автоматический поиск устройств.

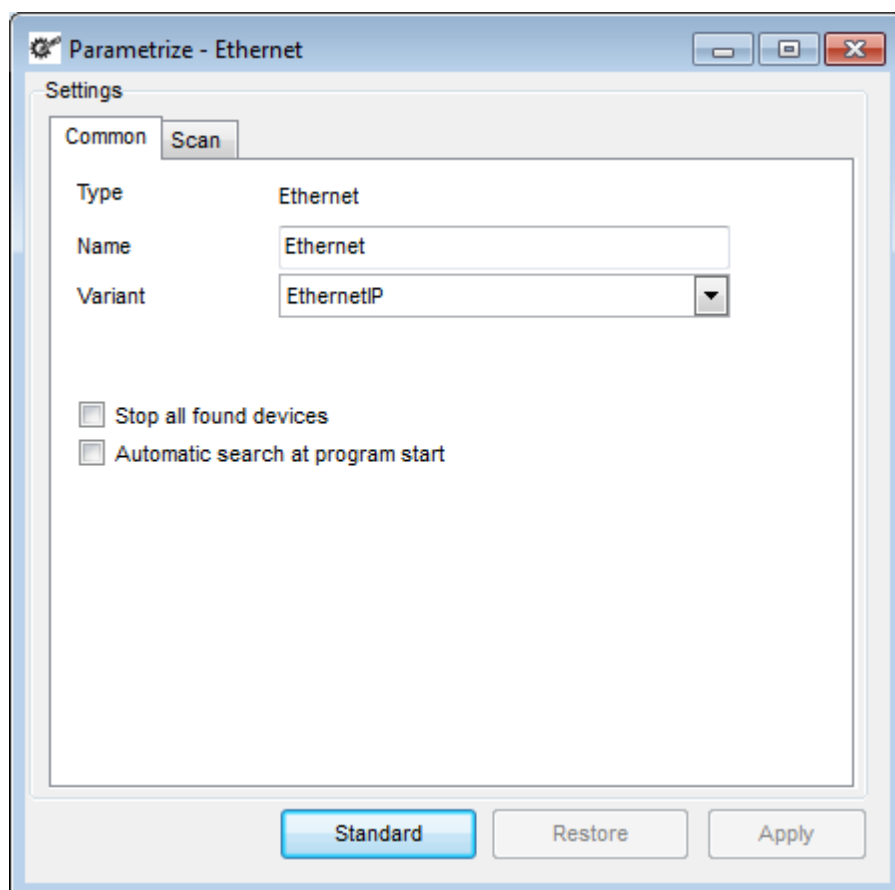


**Информация****Изменения**

Изменения сохраняются в программе только после нажатия кнопки «Adopt». Кнопка «Restore» позволяет восстановить стандартные значения параметров.

## 3.2 Ethernet

### 3.2.1 General settings

**Name**

Поле, в котором можно указать имя модуля передачи данных.

**Kind**

В раскрывающемся списке можно выбрать тип сети (ProfiNet, EthernetIP или EtherCAT).

**Stop all found devices**

Если активен этот параметр, программа NORD CON, завершив поиск устройств, отправляет на все обнаруженные устройства команду «Disable». Останавливаются все устройства, управляемые через шину (P509).

**Automatic search at program start**

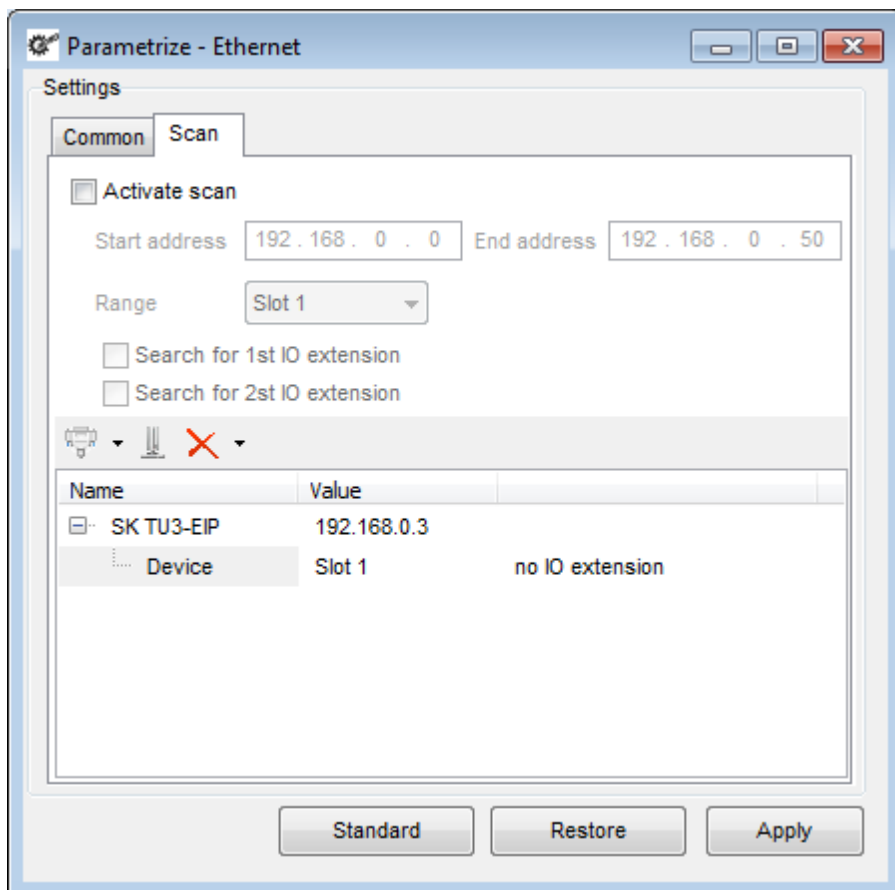
Этот параметр включает (или отключает) автоматический поиск устройств после запуска программы. Если эта опция включена, при запуске NORD CON выполняется автоматический поиск устройств.

#### **i** Информация

#### Изменения

Изменения сохраняются в программе только после нажатия кнопки «Adopt». Кнопка «Restore» позволяет восстановить стандартные значения параметров.

### 3.2.2 Scan



#### **Activate scan**

С помощью этой настройки включается и отключается поиск устройств. Если поиск устройств включен, выполняется поиск устройств в диапазоне IP-адресов, определенным начальным и конечным адресом. Если поиск устройств отключен, при сканировании шины используется следующая конфигурация.

#### **Starting address**

В этом поле пользователь может указать начальный адрес диапазона поиска.

#### **End address**

В этом поле пользователь может указать конечный адрес диапазона поиска.

### Add bus module

Кнопка позволяет добавить в список устройств новое оборудование.

### Add device

Добавляет новое устройство в список устройств.

### Delete

Удаляет отмеченное устройство из списка устройств.

### Value - bus module (Address):

В этот столбец нужно ввести IP-адрес подключенного оборудования.

### Value - device:

В этом столбце необходимо указать данные о разъеме устройства (см. таблицу ниже).

Разъем / слот 1	Разъем / слот 2	Разъем / слот 3	Разъем / слот 4	Разъем / слот 5	Разъем / слот 6	Разъем / слот 7	Разъем / слот 8
Адрес системной шины 32 или SK 5xxE через TU3	Адрес системной шины 34	Адрес системной шины 36	Адрес системной шины 38	Адрес системной шины 40	Адрес системной шины 42	Адрес системной шины 44	Адрес системной шины 46

### Пример:

### Additional - device:

В этот столбец нужно ввести конфигурацию модуля расширения.

Тип оборудования	Разъем / слот 1	Разъем / слот 2	Разъем / слот 3	Разъем / слот 4
SK TU3-EIP V1.2 SK TU3-PNT V1.2	SK 5xxE	недоступно	недоступно	недоступно
SK CU4-EIP V1.2 SK TU4-EIP V1.2 SK CU4-PNT V1.2	SK 5xxE SK 2xxE SK 19xE SK 1xxE	SK 5xxE SK 2xxE SK 19xE SK 1xxE	SK 5xxE SK 2xxE SK 19xE SK 1xxE	SK 5xxE SK 2xxE SK 19xE SK 1xxE

### Информация

### Права доступа

Для параметризации и управления оборудованием требуются соответствующие права доступа. Более подробно о правах доступа можно узнать из руководств, прилагаемых к используемым устройствам.

### Информация

### Изменения

Изменения сохраняются в программе только после нажатия кнопки «Adopt». Кнопка «Restore» позволяет восстановить стандартные значения параметров.

## 4 Параметризация

NORD CON позволяет считывать и менять все параметры, поддерживаемые преобразователем частоты. Эти параметры можно сохранить и снова передать на преобразователь частоты. Значения параметров можно также использовать для оформления документации.

### 4.1 Изменение параметров

Параметры преобразователей частоты хранятся в базе данных параметров. Базу данных параметров можно сохранить на внешнем запоминающем устройстве или вывести на печать; база данных также позволяет менять значения сохраненных в ней параметров. Эти действия можно выполнить из главного меню в окне параметризации. В окне параметризации также имеются кнопки, которые позволяют выполнять основные действия с параметрами.

#### Информация

#### Пункт меню «Parameterise»

Пункт меню «Parameterise» доступен, если выбрано окно параметров.

Изменение параметров в NORD CON может производиться следующим образом:

Действие	Путь	Описание
New	File -> New -> Dataset	Выполняется инициализация базы данных, текущие и новые значения параметров удаляются.
Open	File -> Open	Открытие сохраненной базы данных.
Save	File -> Save	Текущая база данных сохраняется под своим именем.
Save as...	File -> Save as...	Текущая база данных сохраняется под новым именем.
Print preview...	File -> Print preview...	Вывод на печать текущих значений параметров из базы данных.
Read all parameters or Read all	Parameterise -> Read -> All Parameter	Считывание всех параметров преобразователя частоты и сохранение их в базе данных.
Read actual menu group	Parameterise -> Read -> Actual menu group	Считывание всех параметров из выбранной в меню группы и сохранение их в базе данных.
Send new settings	Parameterise -> Send -> new Values	Новые значения параметров, введенные в поле «Neue Einstellungen», передаются на преобразователь частоты. Пользователь может выбрать, нужно ли выполнять это действие для всех параметров или только для параметров выбранной группы.
Send Factory settings	Parameterise -> Send -> Reset values	Выполняется передача стандартных значений всех параметров или параметров выбранной в меню группы.
Selection Enable	Parameterise -> Selection -> Release	Разблокировка всех параметров (или параметров выбранной в меню группы).

Действие	Путь	Описание
Selection Disable	Parameterise -> Selection -> no Release	Блокировка всех параметров (или параметров выбранной в меню группы).
Standard	Button "Standard"	Выбранному параметру присваивается стандартное значение.
Send	Button "Send"	На преобразователь передается значение «new setting» выбранного параметра.
Read	Button "Read"	С преобразователя считывается значение выбранного параметра, которое отображается в поле «Current Setting».

Если включена опция «Auto-Lesen», выбранный параметр будет считываться автоматически.

## 4.2 Фильтр параметров

NORD CON предлагает фильтры, которые упрощают работу с параметрами, позволяя выводить и передавать на устройства параметры определенного типа.

### Информация

Если используется фильтр, то все действия применяются только к параметрам, отображаемым на экране.

Чтобы скрыть параметр, необходимо убрать разблокировку. Это можно сделать, сняв флажок с параметра или через меню 4.1 "Изменение параметров".

В поле «Filter» доступны следующие настройки:

- **Selection only** Отображаются только разблокированные параметры (отмечен флажок перед параметром)
- **No standard** Отображаются только параметры, чьи значения отличаются от стандартных.
- **Info parameters**
  - **Yes** Отображаются информационные параметры.
  - **No** Информационные параметры не отображаются.
  - **Only** Отображаются только информационные параметры.

## 4.3 Офлайн-параметризация (без подключения к сети)

Офлайн-параметризация — это изменение значений параметров в базе данных, которая не подключена ни к одному устройству.

Для работы с параметрами нужно открыть в главном окне меню «Database».

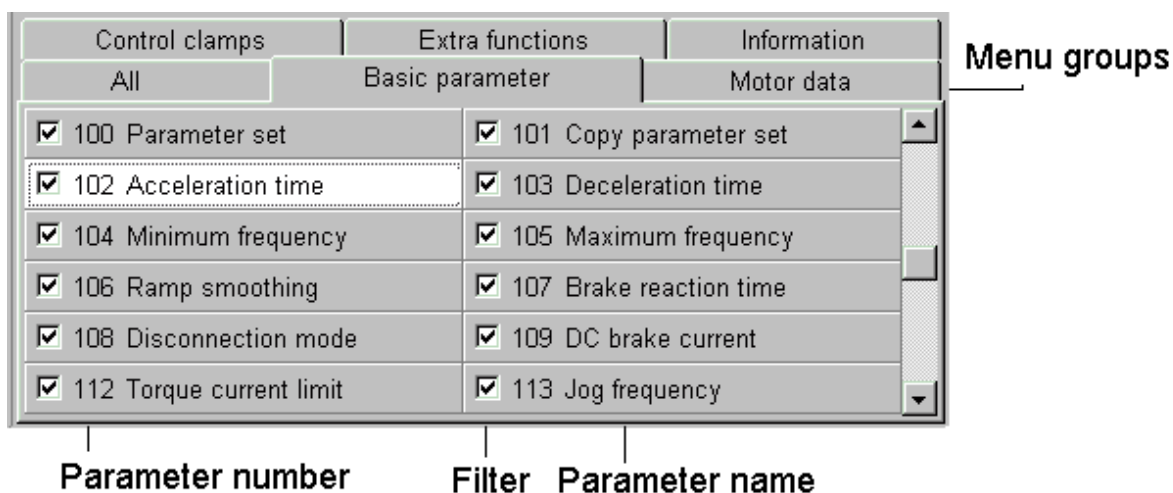
Меню «Database»



- **New** Создание новой базы данных. Новая база данных присваивается типу преобразователю, который нужно выбрать из списка.
- **Open** Сохраненная база данных, с которой можно работать без подключения к устройству.

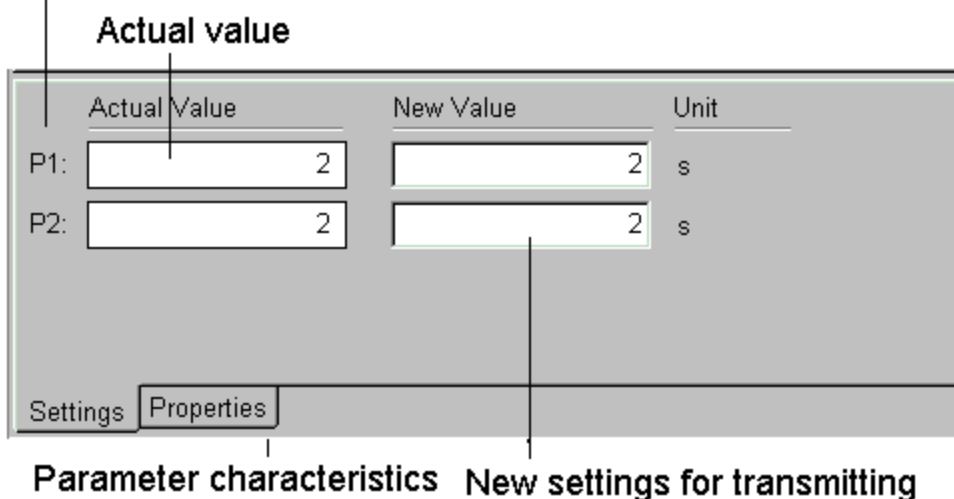
### 4.4 Вкладка параметров

Каждый параметр имеет имя и номер, по которому к нему можно обратиться. Параметры объединены в группы по типу.



У каждого параметра есть свойства и присвоенное ему значение.

### Parameter set



При выборе параметра отображаются все имеющиеся наборы значений соответствующего параметра.

## 4.5 Сравнительный отчет

Этот отчет позволяет сравнить два набора данных. Как правило, в отчете можно сравнить наборы данных, принадлежащие устройствам одного семейства. Параметры выводятся в виде списка. Строки, в которых параметры отличаются, выделены серым цветом. Дополнительно проверяется, есть ли параметры, чьи значения отличаются от стандартных. Такие параметры выделены красным цветом.

### Информация

### Сохранение набора данных

Набор данных нельзя сохранить после формирования отчета! Поэтому набор данных нужно сохранить до того, как будет сформирован отчет.

#### Сравнение онлайн- и офлайн-параметров

Сравнение возможно, если к NORD CON подключено устройство. Для сравнения необходимо открыть окно параметров соответствующего устройства. Рекомендуется выбрать все параметры устройства. Чтобы ограничить количество выводимых параметров, можно использовать фильтр. Для получения отчета необходимо нажать в меню «Parameter Setup -> Comparison». После запуска функции пользователь должен выбрать для сравнения сохраненный набор данных. Если полученные с устройства параметры будут использоваться в качестве резервной копии, пользователь должен сохранить текущий набор данных. После этого будет сформирован и отображен отчет.

### Информация

Сравнение параметров и стандартных значений производится по конфигурации устройства. Если выбран набор данных, не соответствующий конфигурации устройства, отсутствующие параметры будут пустыми или выделены как имеющие отличие.

#### Сравнение офлайн- и онлайн-параметров

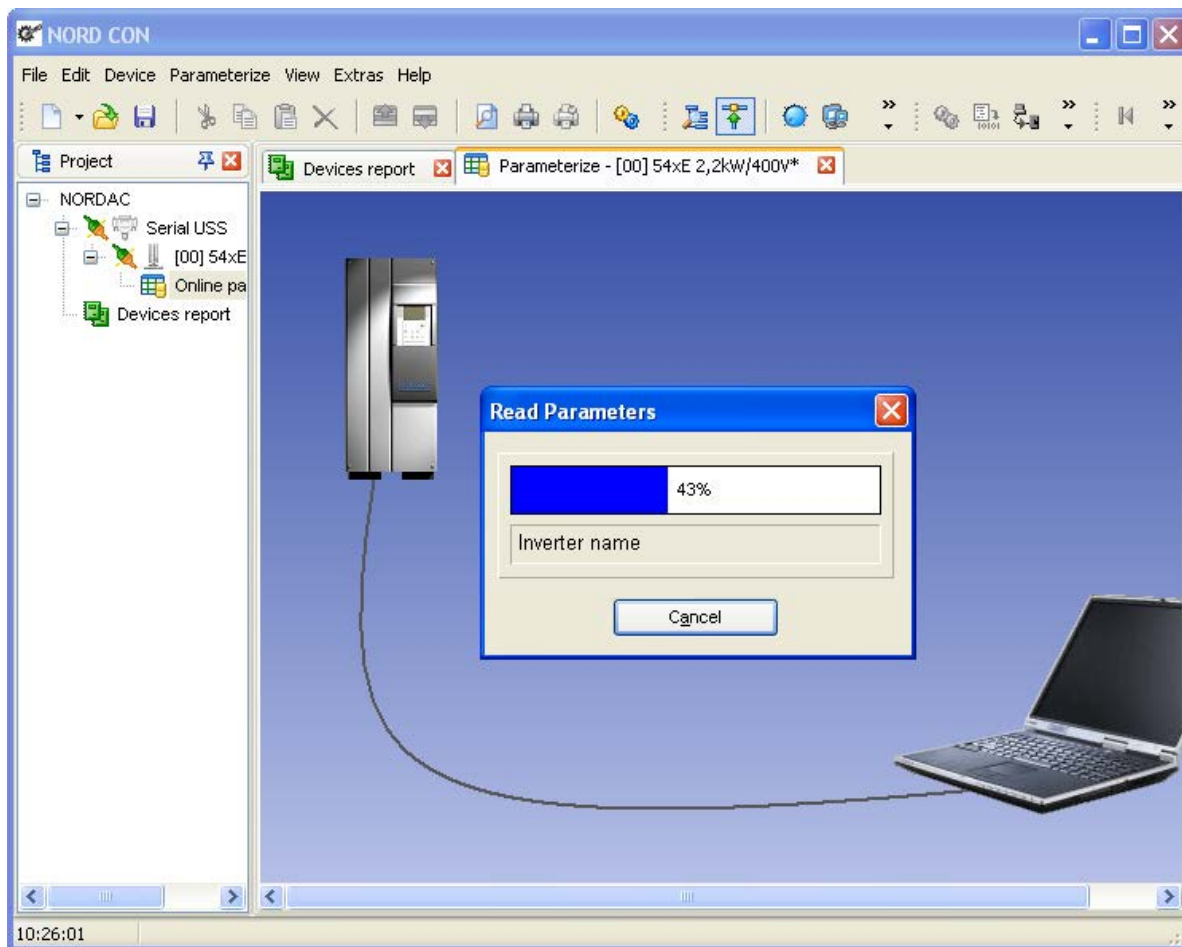
Для сравнения необходимо открыть сохраненный или создать новый набор данных. Чтобы ограничить количество выводимых параметров, можно использовать фильтр. Чтобы сформировать отчет, необходимо открыть меню «Parameter Setup -> Comparison». После запуска функции пользователь должен выбрать для сравнения сохраненный набор данных. После этого будет сформирован и отображен отчет.

### Информация

Сравнение параметров и стандартных значений производится по конфигурации устройства. Если выбран набор данных, не соответствующий конфигурации устройства, отсутствующие параметры будут пустыми или выделены как имеющие отличие.

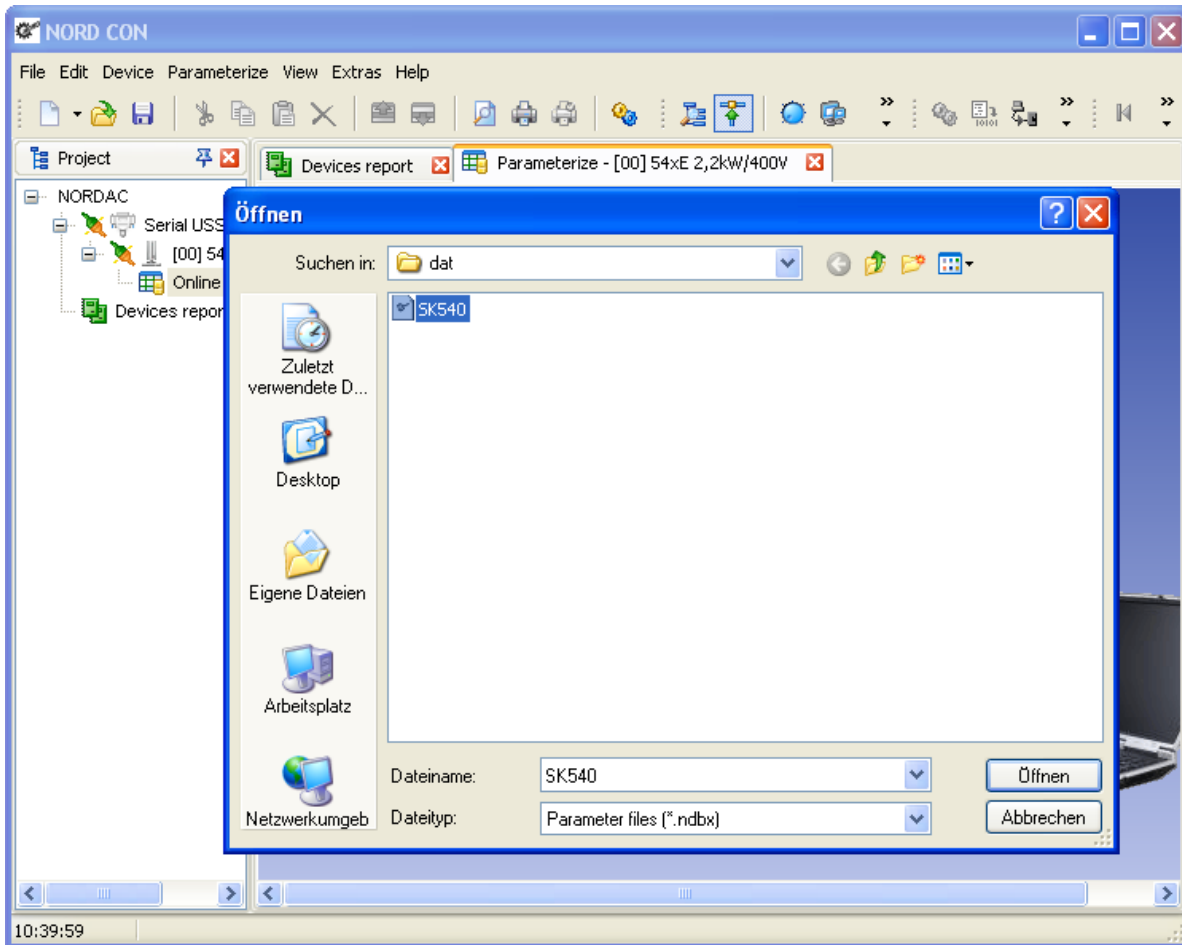
## 4.6 Загрузка параметров с устройства

Функция «Parameter upload from device» передает параметры устройства на ПК и сохраняет их в файле параметров. Функция может быть вызвана через панель инструментов «Device» или через пункт меню «Device/Parameter upload from device». После запуска функции открывается окно загрузки параметров, которое автоматически начинает загрузку параметров. Если во время передачи данных возникнет ошибка подключения, на экране будет выведено соответствующее сообщение. В конце передачи программа предложит пользователю указать имя файла, в котором будут сохранены полученные данные. Чтобы сохранить параметры, пользователь должен нажать «Save».



### 4.7 Передача параметров на устройство

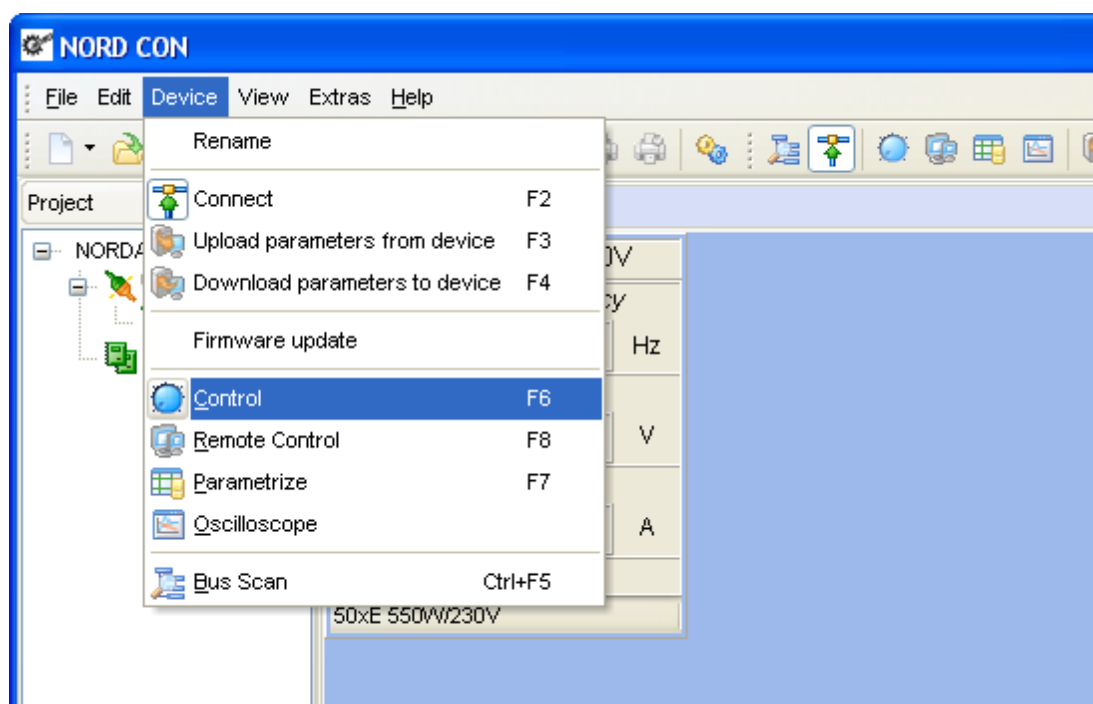
Функция «Parameter download to device» открывает файл параметров на ПК и передает значения из этого файла на устройство. Функция может быть вызвана через панель инструментов «Device» или через пункт меню «Device/Parameter download to device». После запуска функции открывается окно загрузки параметров, в котором пользователь должен указать файл с параметрами. Для подтверждения выбора необходимо нажать «Ок». После этого программа проверит, соответствуют ли параметры из файла параметрам устройства. Если параметры соответствуют, начинается передача данных на устройство.



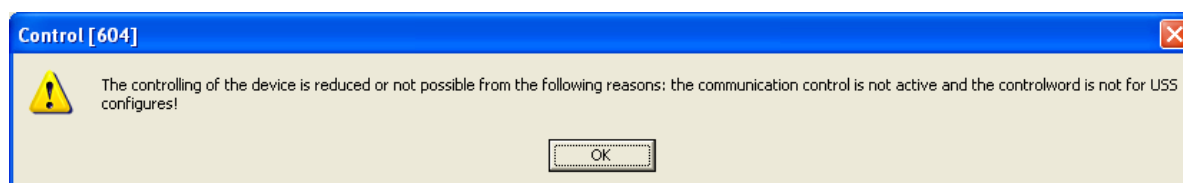
## 5 Управление

### 5.1 Информация о функциях управления

NORD CON позволяет управлять устройствами NORD Частотный преобразователь. Однако для этого необходимо, чтобы устройства были настроены правильным образом. Так как конфигурирование разных устройств может выполняться по-разному, необходимо руководствоваться информацией из документации, прилагаемой к конкретному устройству. Прежде чем использовать функции управления устройством, необходимо выполнить сканирование шины. После сканирования в главном окне отображается список всех подключенных и обнаруженных преобразователей. Выбор устройства производится щелчком левой кнопкой мыши по соответствующей строке в списке. После выбора устройства можно открыть окно «Control» из главного меню «Device/Control» (F6) или из контекстного меню, вызываемое правой кнопкой мыши.



В открытом окне для считывания и работы с управляющей конфигурацией используется стандартная настройка («setting/control/control configuration check»). Если функция «Control» недоступна или доступна с ограничениями, на экране выводится соответствующее сообщение.



В окне «Control» пользователю предлагается два действия:

5.2 "Стандартное управление" разблокировать преобразователь частоты, увеличить или уменьшить расчетное значение. Кроме того, здесь пользователь может изменить направление вращения и разблокировать сообщения об ошибках.

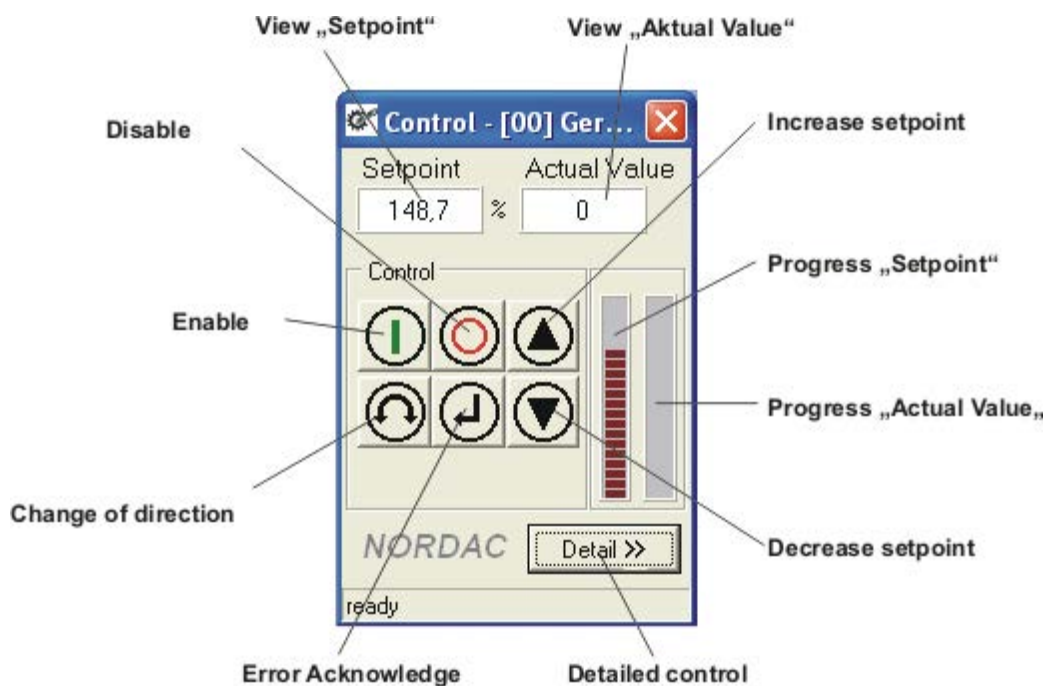
5.3.1 "Обзор" Это окно предоставляет доступ ко всем функциями управления.

## 5.2 Стандартное управление

Стандартное управление предлагает пользователю следующие функции:

- разблокировку устройства
- увеличение или уменьшение уставки
- изменение направления вращения
- разблокировка ошибок

Однако эти функции доступны, если устройство настроено для управления через шину. Описание необходимых параметров и значений приводится в руководстве, прилагаемом к соответствующему преобразователю частоты.



В панели «Standard» отображается только первое расчетное или действительное значение. В каждой конфигурации используется только один формат значений. Кнопка «Detail» позволяет отобразить дополнительные функции управления.

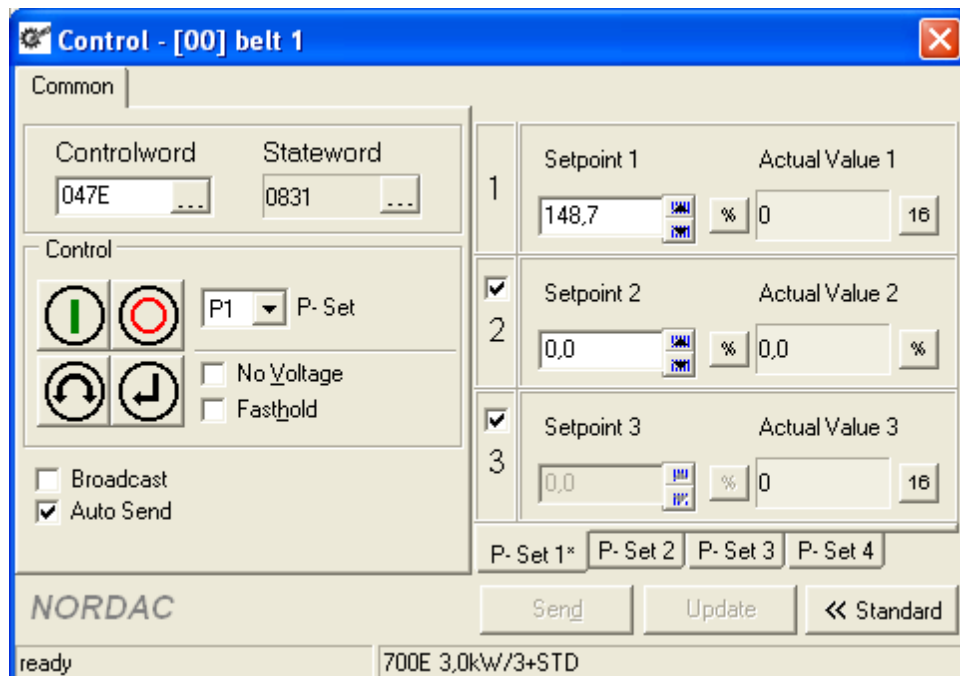
## 5.3 Дополнительные функции управления

### 5.3.1 Обзор

В режиме «Detailed control» пользователь может воспользоваться дополнительными функциями, такими как

- 5.3.2 "Управление"
- 5.3.3 "Работа с расчетными и действительными значениями"

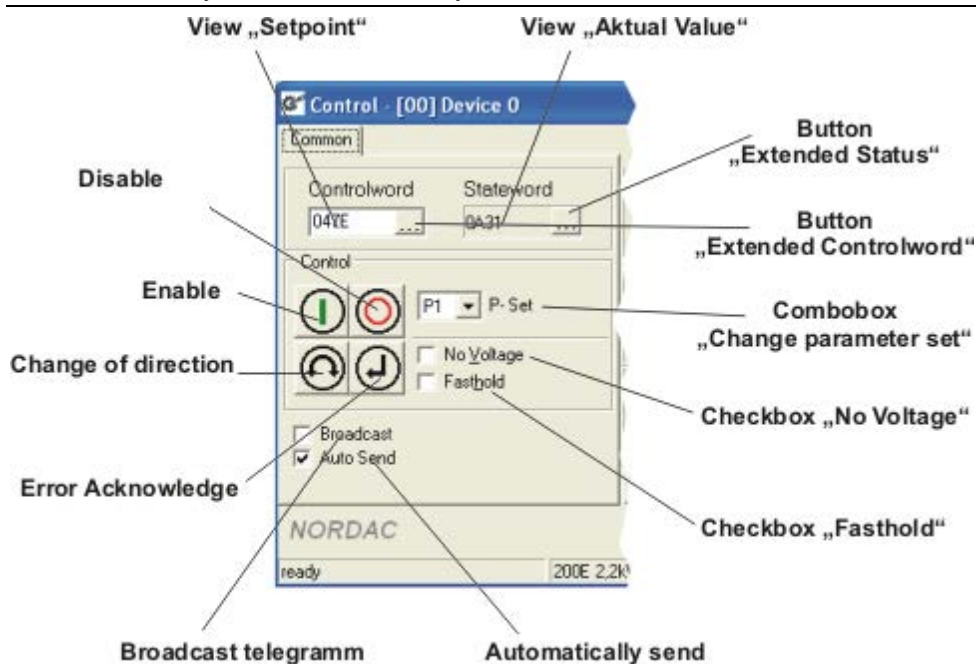
- отправка телеграмм через в режиме широкого вещания
- создание разных наборов параметров
- автоматическая отправка управляющего слова и расчетных значений



### 5.3.2 Управление

Управляющее слово отображается в поле «Control word» в шестнадцатеричном формате. Чтобы изменить управляющее слово, пользователь должен ввести в этом поле новое значение в шестнадцатеричном формате. Для побитового ввода значения управляющего слова нужно открыть дополнительные поля, нажав кнопку «Control word edit». Откроется окно, в котором управляющее слово можно ввести по битам.

Слово состояния выводится в шестнадцатеричном формате в панели «Status word». Кнопка «Bit orientated detail view» позволяет вывести слово состояния в битовом представлении. Слово состояния, выдаваемое машиной состояния преобразователя частоты, также выводится в виде текстовой строки в строке состояния.



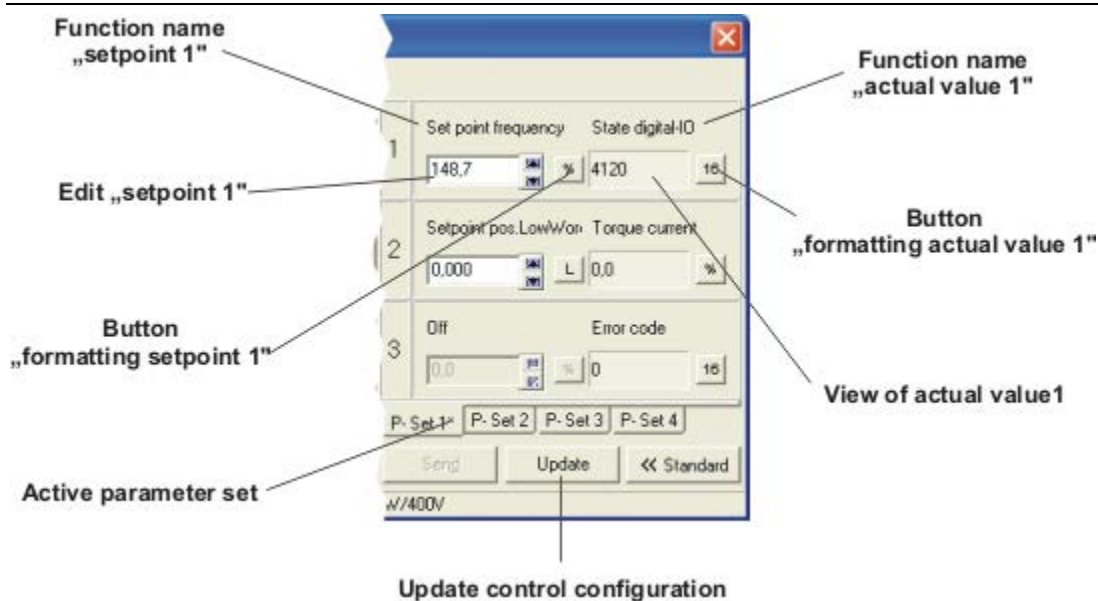
### 5.3.3 Работа с расчетными и действительными значениями

Для управления устройством пользователь может задать до 3 расчетных и действительных значений (см. описание устройства). Расчетные и действительные значения отображаются в заданном формате (**Fehler! Verweisquelle konnte nicht gefunden werden. "Fehler! Verweisquelle konnte nicht gefunden werden."**, кнопка «formatting setting value x»). Ввод расчетных значений производится в том же формате.

Пользователь может управлять расчетными значениями независимо от действительных, выбрав параметр «Setting/control/ parameter set individual management». В этом случае расчетные значения можно указать для каждого набора параметров. При активировании набора параметров значения параметров передаются на устройство. Это необходимо, чтобы для каждого набора параметров можно было задать различные расчетные или действительные значения. Активный набор параметров отмечен звездочкой.

Если не выбрано «Setting/control/configuration automatically checked», пользователь может считать конфигурацию, нажав кнопку «Update».






### 5.3.4 Формат расчетного и действительного значения

Обозначение	Наименование	Описание
"%"	16 Bit standardised values	Расчетное или действительное значение интерпретируется как нормализованное число 16 бит. Нормализация означает линейное изменение диапазона с коэффициентом от -200% до 199% относительно базового значения (например, номинальной частоты).
"16"	16 Bit not standardised	В этом случае расчетное или действительное значение отображается и передается в виде 16-битного числа без коэффициента.
"B"	DigInBits	Расчетное или фактическое значение интерпретируется как 8-битное число. Для каждого бита отображается его состояние в отдельном поле с флажком. Включая и отключая флажки можно менять отдельные биты расчетного значения.
"L"	32 Bit Low-Word	Расчетное или фактическое значение интерпретируется как младшее слово (16 бит) 32-битного числа. Если дополнительное расчетное или действительное значение определено в формате «32 Bit High-Word», оба расчетных (или действительных) значения выводятся в самом верхнем поле. Расчетное значение в таком случае можно указать в 32-битном формате.
"H"	32 Bit High-Word	Расчетное или фактическое значение интерпретируется как старшее слово (16 бит) 32-битного числа (см. «32 Bit Low-Word»).

### 5.3.5 Слово состояния

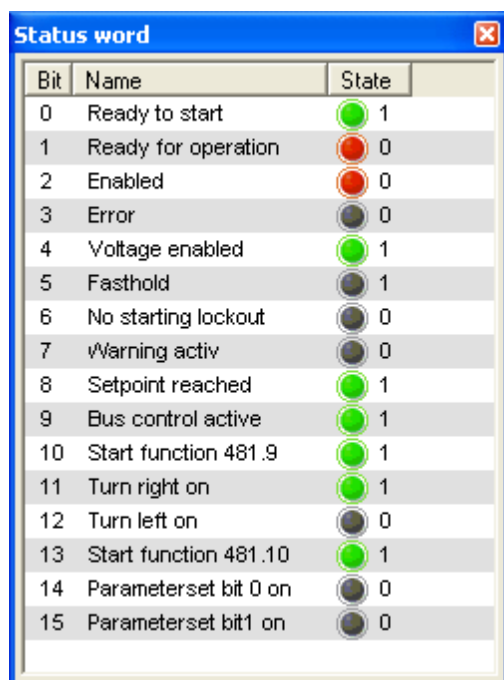
В окне «Status word» отображаются биты текущего слова состояния в виде таблицы. В таблице выводятся номер, название и состояние бита. Рядом с каждым битом отображается также индикатор, цвет которого зависит от величины и значения бита.

















**Значение индикаторов:**

Индикатор	Значение
	Бит используется и/или разблокирован.
	Ошибка или бит не разблокирован.
	Бит не используется.

По умолчанию слово состояния читается циклически, изменения слова отображаются в этом окне. Чтобы отключить периодическое считывание слова состояния, необходимо отключить опцию «Automatic» во всплывающем меню (правая кнопка мыши).

Это окно, как правило, расположено рядом с окном «Control». Чтобы разместить окно в другом месте рабочего стола, необходимо нажать во всплывающем меню на «Docking/no». Для экономии места на экране окно можно вставить в виде вкладки рядом с вкладкой «General». Для этого нужно перетащить окно в область над вкладкой «General», удерживая его левой кнопкой мыши. После того, как клавиша мыши будет отпущена, окно примет вид вкладки. Чтобы переключиться в режим окна, необходимо дважды щелкнуть мышью по вкладке.






Bit	Name	State
0	Ready to start	 1
1	Ready for operation	 0
2	Enabled	 0
3	Error	 0
4	Voltage enabled	 1
5	Fasthold	 1
6	No starting lockout	 0
7	Warning activ	 0
8	Setpoint reached	 1
9	Bus control active	 1
10	Start function 481.9	 1
11	Turn right on	 1
12	Turn left on	 0
13	Start function 481.10	 1
14	Parameterset bit 0 on	 0
15	Parameterset bit1 on	 0

**5.3.6 Слово управления**

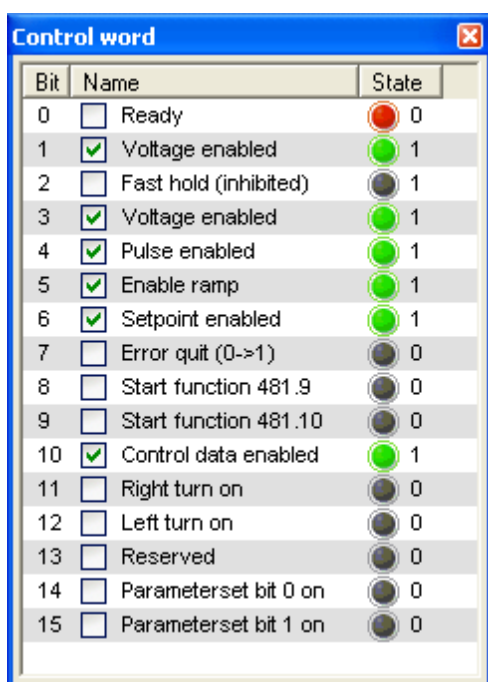
В окне «Control word» отображаются биты текущего слова управления в виде таблицы. В таблице выводятся номер, название и состояние бита. Рядом с каждым битом отображается также индикатор, цвет которого зависит от величины и значения бита. Если устройство настроено для управления через USS, пользователь может изменить значения битов, установив или сняв флажки рядом с каждым битом. При любом изменении управляющего слова новое значение сразу же отправляется на устройство («Automatic sending»).

















### Значение индикаторов:

Индикатор	Значение
	Бит используется и/или разблокирован.
	Ошибка или бит не разблокирован.
	Бит не используется.

По умолчанию слово состояния читается циклически, изменения слова отображаются в этом окне. Чтобы отключить периодическое считывание слова состояния, необходимо отключить опцию «Automatic» во всплывающем меню (правая кнопка мыши).

Это окно, как правило, расположено рядом с окном «Control». Чтобы разместить окно в другом месте рабочего стола, необходимо нажать во всплывающем меню на «Docking/no». Для экономии места на экране окно можно вставить в виде вкладки рядом с вкладкой «General». Для этого нужно перетащить окно в область над вкладкой «Control», удерживая его левой кнопкой мыши. После того, как клавиша мыши будет отпущена, окно примет вид вкладки. Чтобы переключиться в режим окна, необходимо дважды щелкнуть мышью по вкладке.



Bit	Name	State
0	<input type="checkbox"/> Ready	 0
1	<input checked="" type="checkbox"/> Voltage enabled	 1
2	<input type="checkbox"/> Fast hold (inhibited)	 1
3	<input checked="" type="checkbox"/> Voltage enabled	 1
4	<input checked="" type="checkbox"/> Pulse enabled	 1
5	<input checked="" type="checkbox"/> Enable ramp	 1
6	<input checked="" type="checkbox"/> Setpoint enabled	 1
7	<input type="checkbox"/> Error quit (0->1)	 0
8	<input type="checkbox"/> Start function 481.9	 0
9	<input type="checkbox"/> Start function 481.10	 0
10	<input checked="" type="checkbox"/> Control data enabled	 1
11	<input type="checkbox"/> Right turn on	 0
12	<input type="checkbox"/> Left turn on	 0
13	<input type="checkbox"/> Reserved	 0
14	<input type="checkbox"/> Parameterset bit 0 on	 0
15	<input type="checkbox"/> Parameterset bit 1 on	 0

## 6 Дистанционное управление

NORD CON может имитировать пульт управления преобразователя частоты. Для этого соответствующее устройство должно передать в NORD CON свои данные. На экране выводятся виртуальные кнопки, функции которых соответствуют функциями преобразователя. При нажатии кнопки на преобразователь передается соответствующая команда. Дистанционное управление из программы доступно, если устройство не было разблокировано ранее через управляющие клеммы или последовательный интерфейс (P509 = 0 и P510 = 0). Кроме того, в параметре «PotentiometerBox Function» (P549) не должна использоваться функция {4} «Frequency addition» или {5} «Frequency subtraction».

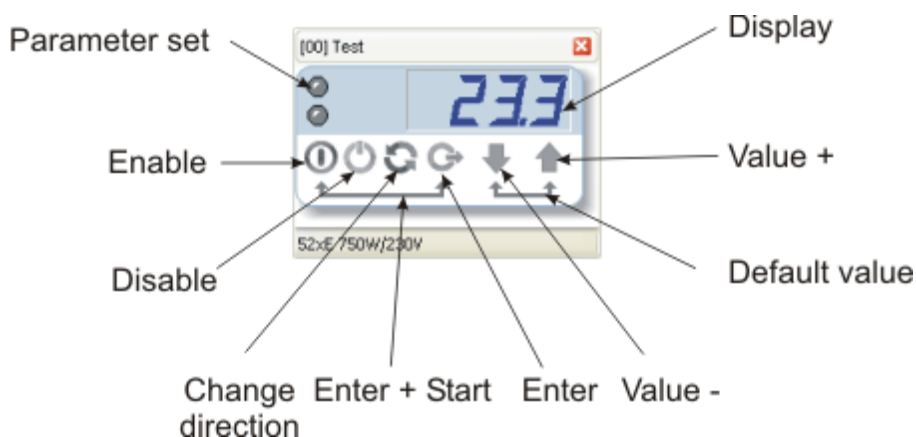
### **i** Информация




### Контроль превышения времени ожидания

Функциями NORD Частотный преобразователь (разблокировка, уставка +/-, направление вращения и т.д.) можно управлять с клавиатуры. В этом случае контроль за превышением времени ожидания не используется, т. е. при разрыве соединения между ПК и преобразователем частоты управление становится невозможным.

### 6.1 Стандартная настройка

Если не включена настройка 13.1 "Окно программы", функция « Remote» всегда открывает стандартное окно для всех устройств.



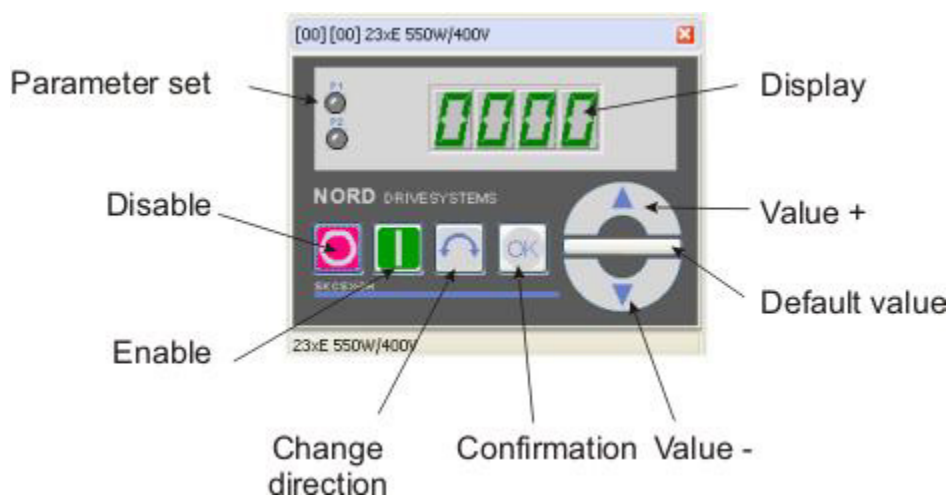
Название операции	Значок	Описание
Enable		Включение преобразователя частоты. Включение преобразователя частоты с установленной толчковой частотой (P113). Преобразователь обеспечивает по возможности заданную минимальную частоту (P104). Параметрам >Interface< (>Интерфейс<) P509 и P510 должно быть присвоено = 0.
Disable		Включение преобразователя частоты. Включение преобразователя частоты с установленной толчковой частотой (P113). Преобразователь обеспечивает по возможности заданную минимальную частоту (P104). Параметрам >Interface< (>Интерфейс<) P509 и P510 должно быть присвоено = 0.
Change dir		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b>







Название операции	Значок	Описание
		Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Up	↑	После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Down	↓	После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Enter	↻	Нажмите эту кнопку, чтобы сохранить измененное значение параметра или перейти к другим параметрам или значениям.  <b>Примечание.</b> Если сохранение измененного значения не требуется, для выхода из параметра без сохранения используйте кнопку «-».
Change Dir + Stop		Одновременное нажатие STOP и «Change direction key» приводит к инициализации быстрого останова.
Enter + Start		Если устройство разблокировано, при одновременном нажатии на кнопки «Вкл» и «ENTER» производится переключение в режим редактора.

На программном пульте управления можно выполнять все те же функции, что на пульте управления преобразователя (Control Box).

### 6.2 NORDAC SK 200 E

Окно дистанционного управления преобразователем частоты типа NORDAC SK 200 E содержит следующие элементы управления:

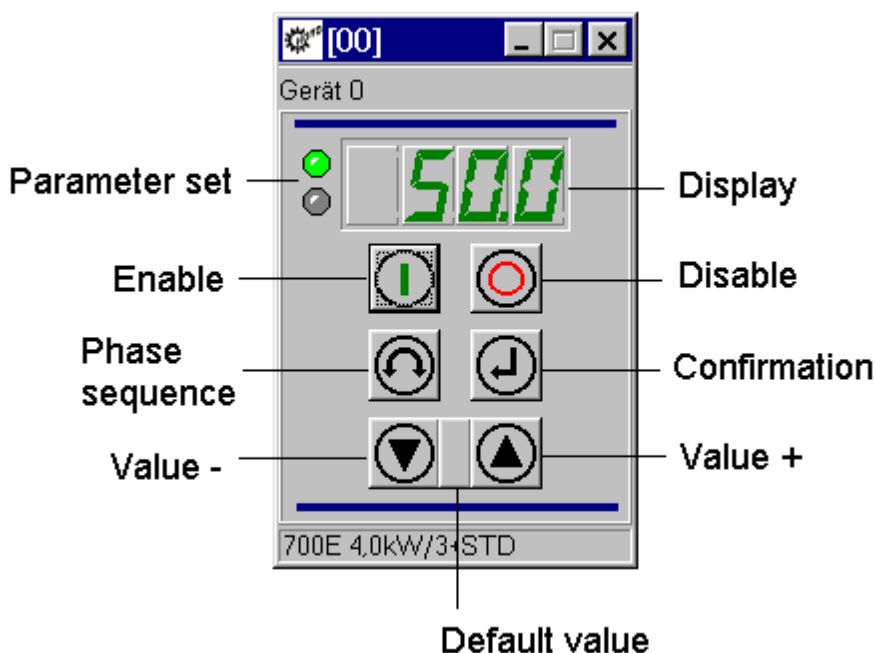







Название операции	Значок	Описание
Enable		Включение преобразователя частоты. Включение преобразователя частоты с установленной толчковой частотой (P113). Преобразователь обеспечивает по возможности заданную минимальную частоту (P104). Параметрам >Interface< (>Интерфейс<) P509 и P510 должно быть присвоено = 0.
Disable		Включение преобразователя частоты. Включение преобразователя частоты с установленной толчковой частотой (P113). Преобразователь обеспечивает по возможности заданную минимальную частоту (P104). Параметрам >Interface< (>Интерфейс<) P509 и P510 должно быть присвоено = 0.
Change dir		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Up		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Down		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Enter		Нажмите эту кнопку, чтобы сохранить измененное значение параметра или перейти к другим параметрам или значениям.  <b>Примечание.</b> Если сохранение измененного значения не требуется, для выхода из параметра без сохранения используйте кнопку «-».
Change Dir + Stop		Одновременное нажатие STOP и «Change direction key» приводит к инициализации быстрого останова.
Enter + On		Если устройство разблокировано, при одновременном нажатии на кнопки «Вкл» и «ENTER» производится переключение в режим редактора.


На программном пульте управления можно выполнять все те же функции, что на пульте управления преобразователя (Control Box).

### 6.3 NORDAC SK 700/500/300 E

Окно дистанционного управления преобразователем частоты типа NORDAC SK 700/500/300 E содержит следующие элементы управления:



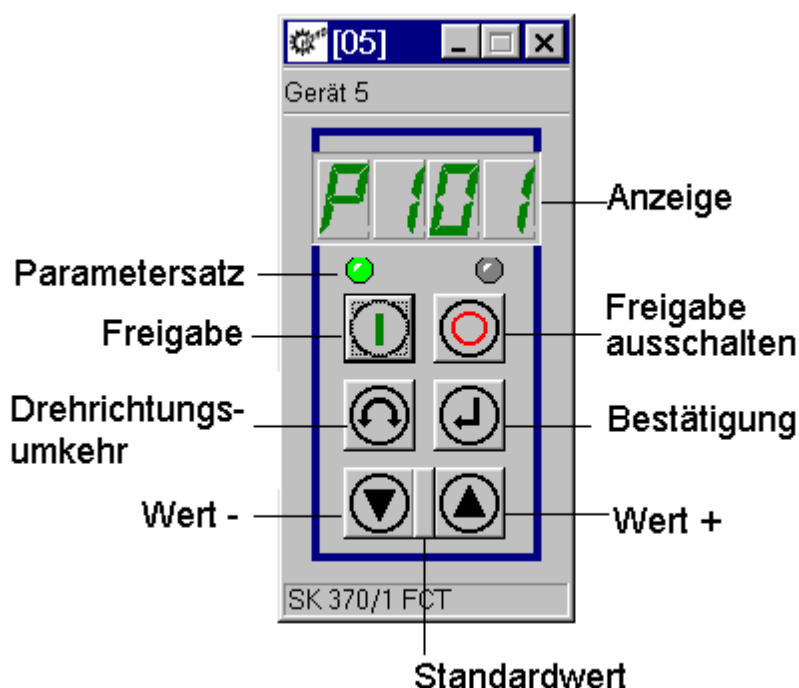
Название операции	Значок	Описание
Enable		Включение преобразователя частоты. Включение преобразователя частоты с установленной толчковой частотой (P113). Преобразователь обеспечивает по возможности заданную минимальную частоту (P104). Параметрам >Interface< (>Интерфейс<) P509 и P510 должно быть присвоено = 0.
Disable		Включение преобразователя частоты. Включение преобразователя частоты с установленной толчковой частотой (P113). Преобразователь обеспечивает по возможности заданную минимальную частоту (P104). Параметрам >Interface< (>Интерфейс<) P509 и P510 должно быть присвоено = 0.
Change dir		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Up		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Down		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.


Название операции	Значок	Описание
Enter		Нажмите эту кнопку, чтобы сохранить измененное значение параметра или перейти к другим параметрам или значениям.  <b>Примечание.</b> Если сохранение измененного значения не требуется, для выхода из параметра без сохранения используйте кнопку «-».
Change Dir + Stop		Одновременное нажатие STOP и «Change direction key» приводит к инициализации быстрого останова.
Enter + On		Если устройство разблокировано, при одновременном нажатии на кнопки «Вкл» и «ENTER» производится переключение в режим редактора.

На программном пульте управления можно выполнять все те же функции, что на пульте управления преобразователя (Control Box).






#### 6.4 NORDAC vector mc

Окно дистанционного управления преобразователем частоты типа NORDAC vector mc содержит следующие элементы управления:



Название операции	Значок	Описание
Enable		Включение преобразователя частоты. Включение преобразователя частоты с установленной толковой частотой (P113). Преобразователь обеспечивает по возможности заданную минимальную частоту (P104). Параметрам >Interface< (>Интерфейс<) P509 и P510 должно быть присвоено = 0.

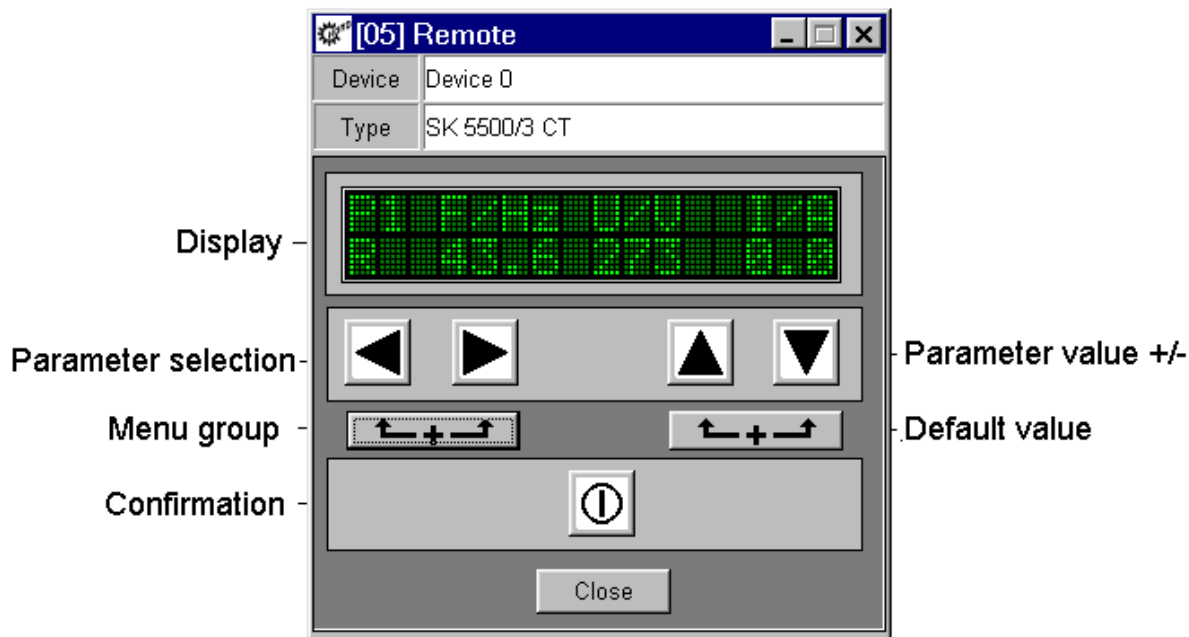








Название операции	Значок	Описание
Disable		Включение преобразователя частоты. Включение преобразователя частоты с установленной толчковой частотой (P113). Преобразователь обеспечивает по возможности заданную минимальную частоту (P104). Параметрам >Interface< (>Интерфейс<) P509 и P510 должно быть присвоено = 0.
Change dir		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Up		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Down		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Enter		Нажмите эту кнопку, чтобы сохранить измененное значение параметра или перейти к другим параметрам или значениям.  <b>Примечание.</b> Если сохранение измененного значения не требуется, для выхода из параметра без сохранения используйте кнопку «-».
Change Dir + Stop		Одновременное нажатие STOP и «Change direction key» приводит к инициализации быстрого останова.
Enter + On		Если устройство разблокировано, при одновременном нажатии на кнопки «Вкл» и «ENTER» производится переключение в режим редактора.

На программном пульте управления можно выполнять все те же функции, что на пульте управления преобразователя (Control Box).

### 6.5 NORDAC vector ct

Окно дистанционного управления преобразователем частоты типа NORDAC vector ct содержит следующие элементы управления:



Название операции	Значок	Описание
Enable		Включение преобразователя частоты. Включение преобразователя частоты с установленной толчковой частотой (P113). Преобразователь обеспечивает по возможности заданную минимальную частоту (P104). Параметрам >Interface< (>Интерфейс<) P509 и P510 должно быть присвоено = 0.
Disable		Включение преобразователя частоты. Включение преобразователя частоты с установленной толчковой частотой (P113). Преобразователь обеспечивает по возможности заданную минимальную частоту (P104). Параметрам >Interface< (>Интерфейс<) P509 и P510 должно быть присвоено = 0.
Change dir		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Up		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Down		После нажатия данной кнопки происходит изменение направления вращения двигателя. «Rotation to the left» — значение со знаком минус.  <b>Внимание.</b> Необходимо соблюдать осторожность при работе с насосами, шнековыми конвейерами, вентиляторами и т. п. Клавишу можно заблокировать при помощи параметра P540.
Enter		Нажмите эту кнопку, чтобы сохранить измененное значение параметра или перейти к другим параметрам или значениям.

Название операции	Значок	Описание
		<b>Примечание.</b> Если сохранение измененного значения не требуется, для выхода из параметра без сохранения используйте кнопку «-».
Change Dir + Stop		Одновременное нажатие STOP и «Change direction key» приводит к инициализации быстрого останова.
Enter + On		Если устройство разблокировано, при одновременном нажатии на кнопки «Вкл» и «Config» производится переключение в режим редактора.

На программном пульте управления можно выполнять все те же функции, что на пульте управления преобразователя (Control Box).

## 7 Осциллограф

### 7.1 Обзор

В NORD CON имеется функция осциллографа, позволяющая выводить внутренние технологические величины преобразователя NORD Частотный преобразователь в виде графиков.

#### **i** Информация

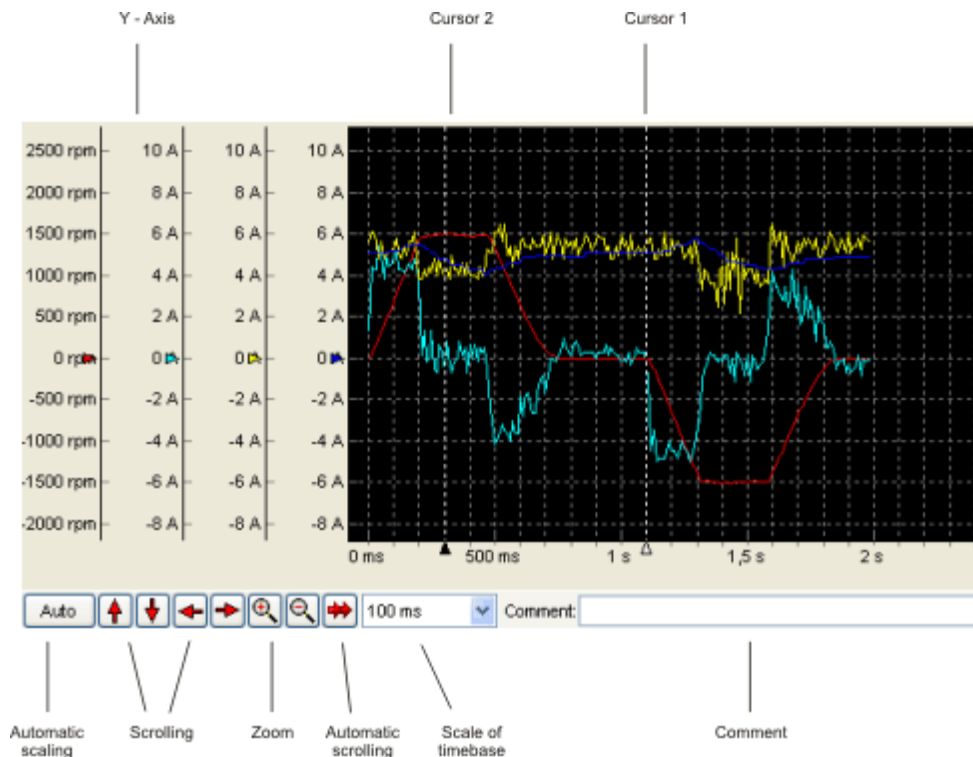
Эта функция доступна только для устройств серии NORDAC vector ct и NORDAC vector mc!

Особенности функции осциллографа:

- запись с 4 каналов
- поддержка разных триггеров (событий)
- масштабирование отдельных серий измерений
- определение среднего значения, эффективного значения и т.д.
- сохранение, печать и экспорт серий измерений

### 7.2 Вывод данных

Функция осциллографа позволяет производить измерения и выводить данные с 4 каналов.



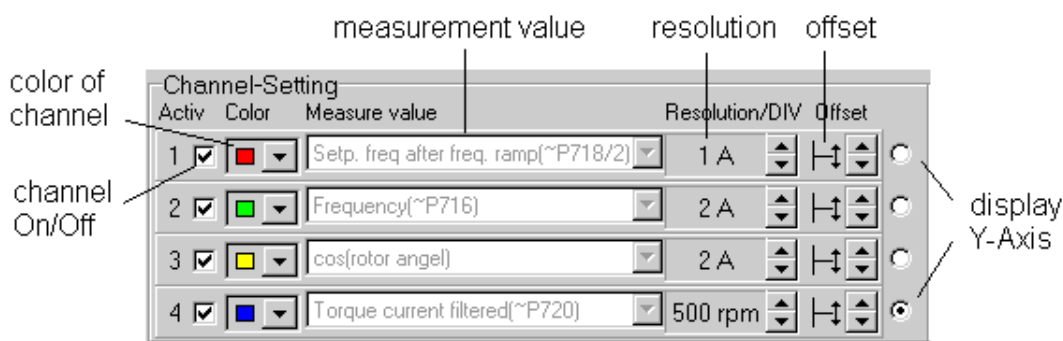
Доступны следующие настройки

Наименование	Описание
Auto	Автоматическое масштабирование всех выводимых серий измерений
Offset	Выбор отображаемого фрагмента (смещение всех серий измерений относительно оси x или y)
Zoom	Размер выводимых величин (отображение всех серий измерений с некоторым коэффициентом)  <b>Примечание.</b> Пользователь может переключаться между режимами «Move» и «Measurement». Для этого нужно переместить указатель мыши на поле и нажать правую кнопку мыши. В режиме «Move» выбор отображаемого фрагмента осуществляется с помощью мыши. Для этого навести указатель на поле, нажать и удерживать некоторое время левую кнопку мыши.
Auto scrolling	Если включен этот параметр, во время записи временная ось сдвигается автоматически к последней точке.
Resolution	В этом поле пользователь может изменить разрешение временной оси.
Comment	Дополнительное поле, в котором можно сохранить некоторую информацию о произведенных измерениях (не более 255 знаков).
Cursor	Выполнение измерения

### 7.3 Управление

Измерение производится следующим образом:

#### 1. Выбор канала

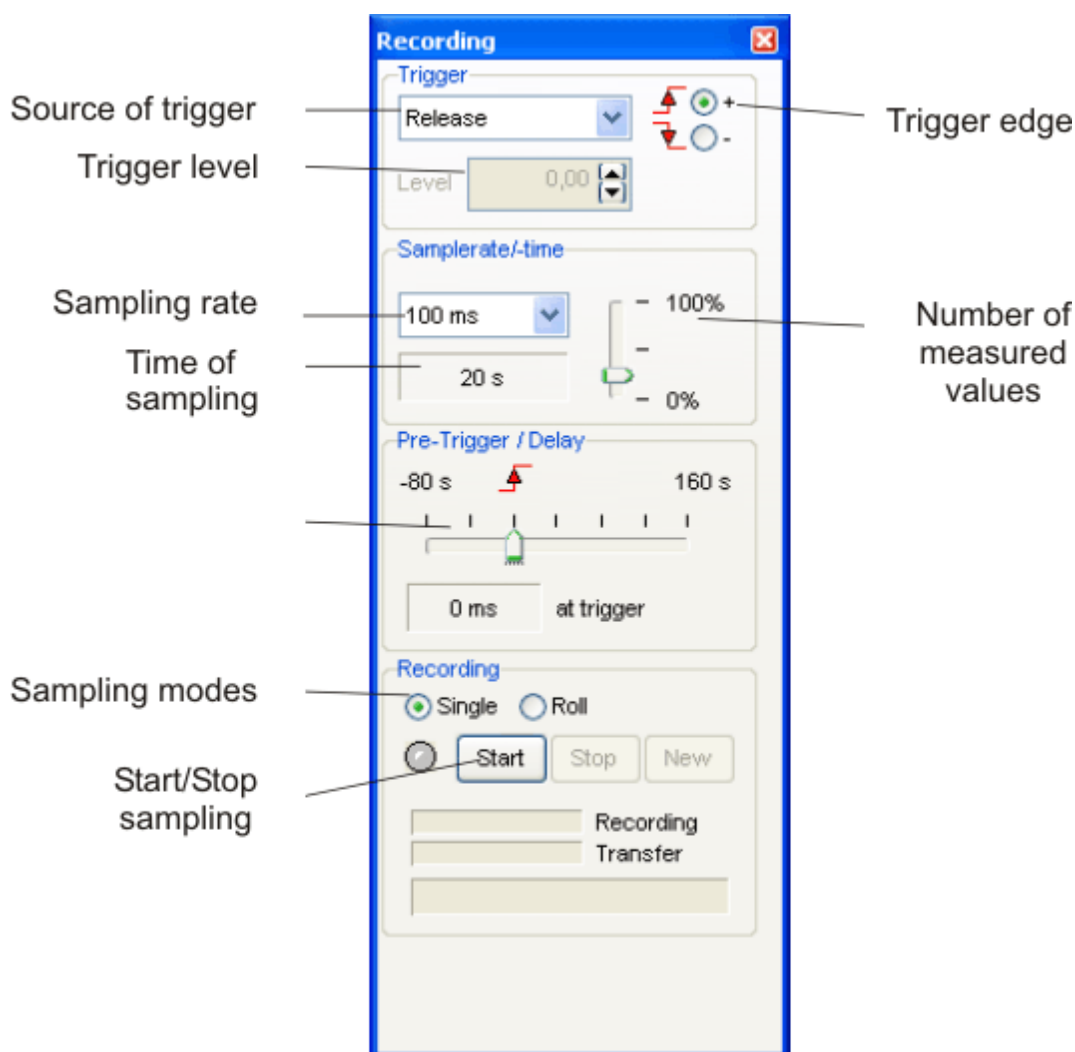


Пользователь может выбрать 4 измеряемые величины. Каждый канал представлен определенным цветом. С помощью флажков можно включить или отключить один или несколько каналов. Для каждого канала необходимо задать разрешение и смещение. При выводе измеряемой величины отображается ось y (ордината) соответствующего канала. Канал тоже можно выбрать в этом окне.

#### Значения функций измерения

Функция измерения	Описание
(=P[номер]) [имя]	Значение этой измерительной функции обновляется с периодом ок. 100 мс и соответствует величине указанного параметра.
[Имя]	Значение этой функции измерения обновляется с периодом ок. 100 мс.
(≈P[номер]) [имя]	Значение этой функции измерения обновляется с периодом ок. 50 мс.
(~P[номер]) [имя]	Значение этой функции измерения обновляется с периодом ок. 250 мкс.

## 2. Настройка триггера



Триггер — событие, которое запускает процесс измерения. Сначала нужно определить источник триггера. Источником триггера может быть некоторое значение измеряемой величины, сигнал на цифровом входе, состояние преобразователя и. д. По уровню триггера или фронту сигнала определяется точка начала измерения.

**i** **Информация****Уровень триггера**

В зависимости от источника уровень триггера может иметь разное распределение. Поэтому не всякие значения могут использоваться для задания уровня триггера. После начала записи достигается и регистрируется первое возможное значение.

Скорость развертки — это время между двумя результатами измерения. По числу развертки определяется длительность развертки.

Время триггера — время после инициирующего события, через которое начинается запись измерения.

**i** **Информация****Интервал развертки**

Эффективность настройки для интервала развертки зависит от динамики изменения измеряемой величины: Чем чаще меняется измеряемая величина, тем меньший интервал нужно задать. От числа развертки зависит длительность передачи результатов измерения между преобразователем и NORD CON.

**3. Выбор режима записи**

Осциллограф имеет 2 режима: «Single» и «Roll». Пользователь может переключаться между режимами, выбрав соответствующее поле. По умолчанию используется режим «Single». В этом режиме запись начинается при возникновении заданных условий (по триггеру). Длительность записи зависит от памяти осциллографа, но не может превышать 2000 с. Результаты регистрируются через заданные интервалы развертки.

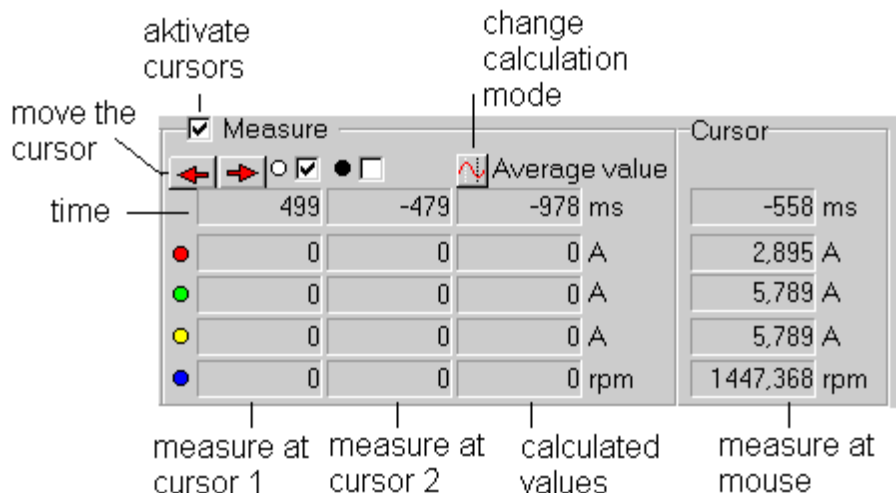
Режим «Roll» позволяет осуществлять запись в течение длительного времени. Зарегистрированные значения сразу же передаются на ПК. В этом режиме пользователь не может указать скорость развертки, так как она зависит от скорости передачи.



**4. Запуск процесса измерения**

Измерение запускает кнопкой «Start». Система следит за условиями, и в случае возникновения инициирующего события (триггера) начинает запись показателей преобразователя. Одновременно осуществляется передача результатов измерения NORD CON. Чтобы остановить процесс, нужно нажать кнопку «Stop». После завершения передачи можно повторить измерение с такими же настройками или запустить новый процесс, задав новые настройки (кнопка «New»).

**7.4 Измерения**

После завершения серии измерений с помощью курсоров осуществляется работа с отдельными измерениями серии.



Окно содержит два курсора. Перемещение курсора осуществляется кнопками . Выбор курсора производится с помощью флажков . Для переключения между режимами «Move» и «Measurement» необходимо навести указатель мыши на поле и нажать правую кнопку мыши. В режиме «Measurement» курсор устанавливается левой кнопкой мыши.

Значения измеренных кривых отображаются в окнах курсоров 1 и 2. Дополнительно по результатам измерений в разных положениях курсора делаются вспомогательные расчеты, например, вычисляется среднее значение. Переключение на другое вычисление производится щелчком на поле с вычисляемой величиной.

Результат измерения отображается в положении указателя мыши.

## 7.5 Печать, сохранение и загрузка серий измерений

Программа позволяет сохранять, экспортировать и выводить на печать полученные результаты серий.

### меню «File»

- **Open** - Позволяет выбрать и открыть файл с сохраненными результатами измерений и настройками. При открытии пользователь должен определить, нужно ли загружать только сохраненные настройки или настройки вместе с сериями измерений.
- **Save as** - Текущие серии измерений и настройки сохраняются в файле с другим именем
- **Export** - Экспорт серий измерений в файл графика или таблицы
- **Print** - Настройка печати и печать текущей серии измерений (на белом фоне)

### Scope Offline

В офлайн-режиме (в режиме, в котором преобразователь частоты не подключен) сохраненную серию измерений можно открыть в главном окне через меню File|Open.



## 8 Редактор макросов

Редактор макросов позволяет автоматизировать выполнение несложных процессов. Интерфейс пользователя для этих целей предлагает разные инструменты, доступные через контекстные меню, панели инструментов или окна макросов. Оператор можно вставить в макрос, перетаскивая мышью. Из контекстного меню доступны также стандартные функции, такие как сохранение или загрузка макроса. Макросы сохраняются в стандартном формате «XML». Макросы в устаревших форматах можно загрузить, выбрав в меню «Open» тип файла «Macro Files V1.26».

### 8.1 Окна и панели пользовательского интерфейса

Окно генератора макросов, помимо редактора, содержит дополнительные панели с инструментами для работы с макросами. Эти панели можно прикрепить у края главного окна. Чтобы отобразить или скрыть все панели, необходимо во всплывающем меню выбрать «View».

#### 8.1.1 Окно переменных

Панель «Variables» можно открыть (или закрыть), выбрав в меню «View->Macro->Variables». Эта панель предназначена для отладки. В окне после запуска макроса выводятся все переменные и объекты макроса вместе с их текущими значениями. Вывод значений настраивается в окне «Properties->Display format».

**Доступны следующие форматы выводимых данных:**

- десятичный
- шестнадцатеричный
- бинарный

#### 8.1.2 Окно свойств

Панель «Property» можно открыть (или закрыть), выбрав в меню «View -> Properties». В этом окне выводятся свойства текущего оператора. В зависимости от оператора пользователь может изменить тип и количество свойств.

Наименование	Описание
Result	С помощью этого свойства пользователь определяет объект, которому будет присвоено новое значение. Пользователь может выбрать только те объекты, которые могут принять некоторое значение (т.е. такие объекты как управляющее слово, параметр или переменная).
Operand	С помощью этого свойства пользователь выбирает объект, посредством которого можно определить отношение или операцию.
Operator	Свойство, которое задает тип операции, например, сложение.
Comment	Это свойство позволяет добавить к оператору комментарий.

В генераторе макросов переменные, управляющие слова, слова состояния, расчетные и действительные значения и параметры называются объектами. У каждого из этих объектов могут быть свои параметры.

Объект	Параметры	Описание
Variable	Name	Задаёт имя переменной или константы. В поле выбора отображаются все используемые переменные. Если задаётся новая переменная, необходимо дать ей уникальное имя. Редактор не различает прописные и строчные буквы.
	Display format	Параметр, посредством которого задаётся формат значений, выводимых в панели «Variables». Поддерживаются следующие форматы: <ul style="list-style-type: none"> <li>• десятичный</li> <li>• шестнадцатеричный</li> <li>• бинарный</li> </ul>
Constant	Value	Параметр, который задаёт значение константы.
	Display format	Параметр, посредством которого задаётся формат значений, выводимых в панели «Variables». Поддерживаются следующие форматы: <ul style="list-style-type: none"> <li>• десятичный</li> <li>• шестнадцатеричный</li> <li>• бинарный</li> </ul>
Control word, Status word	Node number	Параметр, назначающий конкретному устройству номер узла в сети USS.  Примечание. Если невозможно получить с устройства текущее значение управляющего слова, при запуске диспетчера управляющему слову присваивается 0.
	Display format	Параметр, посредством которого задаётся формат значений, выводимых в панели «Variables». Поддерживаются следующие форматы: <ul style="list-style-type: none"> <li>• десятичный</li> <li>• шестнадцатеричный</li> <li>• бинарный</li> </ul>
Setpoint and actual values	Node number	Параметр, назначающий конкретному устройству номер узла в сети USS.  Примечание. Если невозможно получить с устройства текущее расчетное значение, при запуске диспетчера значению присваивается 0.
	Type	Параметр, который задаёт тип значения. Пользователю доступны типы, перечисленные в таблице «Setpoint and actual value types».
	Format	Параметр, задающий формат расчетного и действительного значения. Возможные форматы перечислены в таблице «Setpoint and actual value formatting».
	Resolution	Параметр, задающий разрешение расчетного и действительного значения. Он используется для задания формата оператора в редакторе.
	Display format	Параметр, посредством которого задаётся формат значений, выводимых в панели «Variables». Поддерживаются следующие форматы: <ul style="list-style-type: none"> <li>• десятичный</li> <li>• шестнадцатеричный</li> <li>• бинарный</li> </ul>

Объект	Параметры	Описание
Parameter	Node number	Параметр, назначающий конкретному устройству номер узла в сети USS.
	Parameter number	Значение, которое определяет номер параметра (см. панель «Device catalogue»).
	Sub-index	Задаёт вложенный индекс параметра.
	Resolution	Значение, задающее разрешение расчетного и действительного значения. Оно используется для задания формата оператора в редакторе.
	Data type	Задаёт тип данных параметра. В последних устройствах используется только два типа данных – целое число 16 бит и длинное целое число 32 бит.
	Display format	Параметр, посредством которого задается формат значений, выводимых в панели «Variables». Поддерживаются следующие форматы: <ul style="list-style-type: none"> <li>• десятичный</li> <li>• шестнадцатеричный</li> <li>• бинарный</li> </ul>

### Типы расчетного и действительного значения

Тип	Описание
Value 1 (16-bit)	Используется первое, второе или третье расчетное или действительное значение.
Value 12 (32-bit)	Используется первое и второе расчетное или действительное значение (32 бит).  Примечание. Эта настройка может использоваться, если устройство сконфигурировано соответствующим образом (см. «Setpoint or actual value configuration»).
Value 13 (32-bit)	Используется первое и третье расчетное или действительное значение (32 бит).  Примечание. Эта настройка может использоваться, если устройство сконфигурировано соответствующим образом (см. «Setpoint or actual value formatting»).
Value 23 (32-bit)	Используется второе и третье расчетное или действительное значение (32 бит).  Примечание. Эта настройка может использоваться, если устройство сконфигурировано соответствующим образом (см. «Setpoint or actual value formatting»).

### Форматы расчетного и действительного значения

Формат	Описание
Standardised	Расчетное или действительное значение интерпретируется как нормализованное число 16 бит. Нормализация означает линейное изменение диапазона с коэффициентом от -200% до 199% относительно базового значения (например, номинальной частоты).
Not standardised	В этом случае расчетное или действительное значение отображается и передается в

Формат	Описание
	виде 16-битного числа без коэффициента.
Low word (32bit)	Первое значение является младшим словом, второе – старшим словом значения 12 (32 бит). Этот формат может быть выбран только для 32-битных типов.
High word (32bit)	Первое значение является старшим словом, второе – младшим словом значения 12 (32 бит). Этот формат может быть выбран только для 32-битных типов.

**Примечание.**

Выбранные настройки должны отвечать конфигурации устройства.

### 8.1.3 Окно журнала

Все события, возникшие в процессе управления, регистрируются в журнале. Чтобы вывести журнал, необходимо выбрать в меню «View -> Log» и открыть «Log». Эта панель является панелью инструментов, и она может быть прикреплена в боковой части главного окна. Здесь в виде списка выводятся все записи журнала. Последняя зарегистрированная запись отображается в начале списка.

#### Сохранение журнала

Журнал можно сохранить, выбрав во всплывающем меню пункт «Save as...». После этого выводится окно, в котором пользователь должен указать имя файла и путь для его сохранения. Список будет сохранен в текстовом файле после того, как пользователь подтвердит действие кнопкой «Save».

#### Удаление журнала

Чтобы удалить журнал, нужно выбрать во всплывающем меню пункт «Delete». После этого все записи журнала будут удалены без возможности восстановления.

#### Сортировка записей

С помощью фильтра можно отсортировать записи журнала по типу. Меню «Filter» позволяет определить тип записей, регистрируемых в журнале.

## 8.2 Редактирование макроса

### 8.2.1 Создание нового макроса

Для создания нового документа (макроса) необходимо в контекстном меню выбрать «New». Если до этого выполнялось редактирование другого макроса, генератор макроса предложит сохранить предыдущий документ. При нажатии «Cancel» новый документ не создается. В текущей версии ПО пользователь может редактировать только один документ.

### 8.2.2 Открытие макроса

Чтобы открыть макрос, нажмите в меню «Open» или воспользуйтесь комбинацией клавиш «Ctrl+O». Откроется диалоговое окно, в котором можно выбрать нужный макрос. Чтобы открыть макрос предыдущих версий, необходимо в этом окне также указать тип файла.

### 8.2.3 Сохранение макроса

Чтобы сохранить макрос, нажмите в меню «Save» или воспользуйтесь комбинацией клавиш «Ctrl+S». Эта операция применима только к уже сохраненным документам. Для сохранения новых документов нужно воспользоваться функцией «Save as...».

Чтобы использовать функцию «Save as...», выберите в меню пункт «Save as...». В открывшемся окне необходимо ввести название файла и указать путь, где он будет храниться. Подтвердите действие, нажав «Save». Макрос будет сохранен по указанному пути, и в строке заголовка появится название макроса.

### 8.2.4 Вставка операторов

Для использования функции «Insert» выберите в меню «Insert» или нажмите «Ctrl+V». Эта функция вставляет оператор из буфера обмена в строку, следующую за текущей строкой. Если буфер обмена пуст, пункт меню недоступен. В текущей версии оператор из буфера обмена можно вставить только один раз.

### 8.2.5 Копирование операторов

Чтобы воспользоваться функцией «Copy», выберите в меню «Copy» или нажмите комбинацию клавиш «Ctrl+C». Эта функция копирует выбранную строку в буфер обмена генератора макросов. В настоящей версии можно выбрать только одну строку. Соответственно, в буфер обмена можно скопировать только один оператор. Исключение – оператор Block. Это оператор копируется полностью.

### 8.2.6 Вырезание операторов

Чтобы воспользоваться функцией «Cut», выберите в меню «Cut» или нажмите комбинацию клавиш «Ctrl+X». Эта функция копирует выбранный оператор в буфер обмена генератора макросов. После вставки оператора исходный оператор удаляется из документа. За одну операцию можно вырезать только один оператор.

### 8.2.7 Удаление операторов

Чтобы воспользоваться функцией «Delete», выберите в меню «Delete» или нажмите клавишу «Ctrl + Del». Эта функция удаляет выбранный оператор из документа.

### 8.2.8 Поиск и замена / Search and replace

Для запуска функции «Search and replace» выберите в меню соответствующий пункт или воспользуйтесь комбинацией клавиш «Ctrl+H». Откроется диалоговое окно «Search and replace», в котором можно указать параметры поиска и замены и выполнить соответствующие изменения в документе.

### 8.2.9 Перемещение оператора вверх

Чтобы выполнить операцию «Shift up», выберите в меню «Shift up». Выделенный оператор будет перемещен на одну строку вверх. Если выделенная строка расположена в начале документа, операция не выполняется. Переместить оператор в нужную строчку можно также с помощью мыши.

### 8.2.10 Перемещение оператора вниз

Чтобы выполнить операцию «Down», выберите в меню «Down». Выделенный оператор будет перемещен на одну строку вниз. Если выделенная строка расположена в конце документа,

операция не выполняется. Переместить оператор в нужную строчку можно также с помощью мыши.

### 8.2.11 Добавление нового оператора

Чтобы добавить новый оператор, нужно выбрать в контекстном меню «Functions». Новый оператор всегда вставляется в строку, следующую за выделенной строкой. После вставки пользователь может изменить положение оператора в списке (см. «Up» или «Down»).

К добавленному оператору могут быть применены следующие функции:

Имя	Описание
Assignment	<p>Присваивает новое значение объекту в макросе. Новое значение – это значение, полученное из другого объекта, или константа, определенная пользователем. Как правило, добавляется строка, как описано в примере 1. Функция параметра может быть изменена в разделе «Properties».</p> <p><b>Пример:</b>  <code>Device 00 Controlword = 047F hex // присваивает управляющему слову значение 1151</code>  <code>Var1 = Device 00 Statusword // присваивает переменной значение слова состояния</code></p> <p><b>Примечание.</b>            Присвоение расчетных значений может производиться внутри оператора Block.</p>
Jump mark	<p>Определяет в макросе метку. С помощью команды «Goto» пользователь может перейти к метке, заданной в программе. По умолчанию строка добавляется, как описано в примере 1. Функция параметра может быть изменена в разделе «Properties». В каждом случае метке нужно дать уникальное имя, метки с одинаковыми именами не поддерживаются. Генератор переходит всегда на первую метку в макросе.</p> <p><b>Пример:</b>  <code>Label1: // Определение метки «Label1»</code></p> <p><b>или</b>  <code>Start: // Определение метки «Start»</code></p>
Sleep	<p>Команда, которая приостанавливает выполнение макроса. Необходимо указать время в «ms». Как правило, оператор добавляется, как описано в примере 1. Время можно изменить в разделе «Properties».</p> <p><b>Пример:</b>  <code>Sleep 1000 ms // Ожидание 1 с</code></p> <p><b>или</b>  <code>Sleep 500 ms // Ожидание 0,5 с</code></p>
Go to	<p>Команда перехода на другую строчку в макросе. При вызове команды генератор переходит на строку с указанной меткой. Если генератор не находит метку с указанным именем, строка игнорируется. Если в макросе не определены метки, меню недоступно. По умолчанию команда содержит первую метку. Название метки можно изменить в разделе «Properties».</p> <p><b>Пример:</b>  <code>Goto Start // Перейти к метке «Start»</code></p>
Condition	<p>Оператор, определяющий условный переход в макросе. Если условие верно, генератор переходит на строку с указанной в операторе меткой. Как правило, добавляется строка, как описано в примере 1. Параметры оператора можно изменить в разделе «Properties».</p> <p><b>Пример:</b></p>

Имя	Описание
	if Device 00 Controlword == 047F hex then // если управляющее слово имеет значение 1150, Goto Start // программа переходит к метке «Start»
Block	<p>Этот оператор позволяет включать в один оператор несколько операций присвоения. Эти операции могут применяться только к объектам «Control word» и «Setpoints» одного устройства. В зависимости от конфигурации и назначения устройства пользователь может выбрать «Control value with 1 setpoint», «Control value with 2 setpoints» или «Control value with 3 setpoints».</p> <p><b>Пример:</b>            Block // отправка управляющего слова и расчетных значений1 с протоколом USS            Device 00 Controlword = 1151 // Управляющему слову присваивается 1150            Device 00 Setpoint1 = 20,0 // Расчетному значению присваивается 20</p>
Mathematical and logical operations	<p>Эти операторы позволяют выполнять с объектами простые математические и логические операции. Результат присваивается некоторому объекту. Параметры оператора можно изменить в разделе «Properties».</p> <p><b>Пример:</b>            Var1 = Device 00 Controlword + 047F hex //Сложение            Var1 = Device 00 Statusword AND 047F hex //логическое «И»</p>

### 8.3 Управление выполнением

Планировщик («Scheduler») позволяет управлять порядком выполнения макроса. В нем имеется два параметра.

#### Auto

Если включен этот параметр (автоматический режим), после запуска планировщика обработка строк макроса выполняется последовательно и автоматически. Если эта опция отключена (пошаговый режим), каждый оператор должен выполняться вручную (пункт меню «Next» или комбинация клавиш «F12»).

#### Бесконечный цикл

Если включена эта опция, макрос повторяется бесконечно. После выполнения последнего оператора планировщик переходит в начало макроса.

#### 8.3.1 Выполнение макроса

Планировщик запускается пунктом меню «Start» или клавишей «F9». Если активен автоматический режим, обработка макроса производится строка за строкой. В пошаговом режиме после запуска выполняется только первая строка. Выполнение последующих строчек производится только после нажатия в меню «Next». Планировщик можно запустить повторно только после того, как завершится обработка макроса или его выполнение будет прервано пользователем. Во время работы планировщика изменить параметры операторов нельзя.

#### 8.3.2 Прерывание выполнения

Работу планировщика можно прервать, нажав в меню «Cancel» или клавишу «F11»

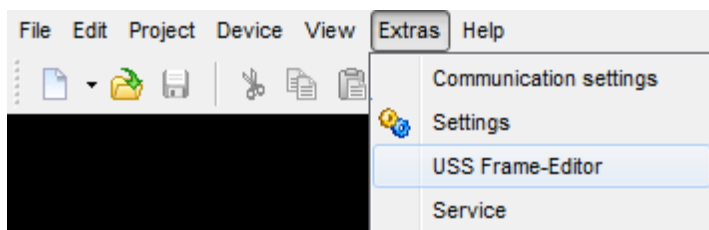
### **8.3.3 Выполнение следующего оператора**

Действие «Next» выполняется с помощью пункта меню «Next» или клавиши «F12». Оно доступно только в шаговом режиме и сообщает планировщику, что необходимо выполнить следующий оператор макроса. Если в макросе был выполнен последний оператор, планировщик автоматически завершает свою работу.



## 9 Редактор кадров USS

Протокол USS определяет распределяет доступ в сетях, построенных на основе последовательной шины, по принципу ведущего и ведомого устройства (Master/Slave). Одним из вариантов этого подключения является двухточечное соединение. К одной шине можно подключить одно ведущее устройство (Master) и до 31 ведомых устройств (Slave). Ведущее устройство обращается к ведомому по указанному в телеграмме адресу. Прямой обмен сообщениями между двумя ведомыми устройства невозможен. Обмен данными производится в полудуплексном режиме посредством USS-передач (телеграмм).



Для генерирования и анализа телеграмм используется редактор кадров USS. Он полностью интегрирован в интерфейс NORDCON, открыть его можно из меню «Extras/USS Frame-Editor». Для работы с телеграммами ведущих и ведомых устройств в редакторе имеется несколько разделов. Пользователь может переключаться между вкладками 9.1 "Master (order)" и 9.2 "Device (response)".

Объект	Описание																					
Telegramm type	<p>Объект, который задает размер и структуру телеграммы USS. Устройствами поддерживаются следующие типы телеграмм:</p> <table border="1" data-bbox="419 376 1396 1182"> <thead> <tr> <th data-bbox="419 376 528 443">Тип</th> <th data-bbox="528 376 742 443">Длина (LGE)</th> <th data-bbox="742 376 1396 443">Описание</th> </tr> </thead> <tbody> <tr> <td data-bbox="419 443 528 526">PPO 0</td> <td data-bbox="528 443 742 526">12</td> <td data-bbox="742 443 1396 526">Стандартная телеграмма для передачи технологических данных и значения параметра 16 бит</td> </tr> <tr> <td data-bbox="419 526 528 609">PPO 1</td> <td data-bbox="528 526 742 609">14</td> <td data-bbox="742 526 1396 609">Расширенная телеграмма для передачи технологических данных и значением параметра 32 бит</td> </tr> <tr> <td data-bbox="419 609 528 763">PPO 2</td> <td data-bbox="528 609 742 763">18</td> <td data-bbox="742 609 1396 763">Расширенная телеграмма для передачи технологических данных (главного и двух вспомогательных расчетных значений) и значения параметра 32 бит</td> </tr> <tr> <td data-bbox="419 763 528 846">PPO 3</td> <td data-bbox="528 763 742 846">6</td> <td data-bbox="742 763 1396 846">Телеграмма с технологическими данными для передачи главного расчетного значения без данных параметра</td> </tr> <tr> <td data-bbox="419 846 528 965">PPO 4</td> <td data-bbox="528 846 742 965">10</td> <td data-bbox="742 846 1396 965">Расширенная телеграмма с технологическими данными для передачи главного и вспомогательных расчетных значений без данных параметра</td> </tr> <tr> <td data-bbox="419 965 528 1182">PPO 6</td> <td data-bbox="528 965 742 1182">16</td> <td data-bbox="742 965 1396 1182">           Телеграмма с 5 расчетными/действительными значениями.   <b>Внимание.</b>            Этот тип телеграммы не поддерживается некоторыми устройствами.         </td> </tr> </tbody> </table>	Тип	Длина (LGE)	Описание	PPO 0	12	Стандартная телеграмма для передачи технологических данных и значения параметра 16 бит	PPO 1	14	Расширенная телеграмма для передачи технологических данных и значением параметра 32 бит	PPO 2	18	Расширенная телеграмма для передачи технологических данных (главного и двух вспомогательных расчетных значений) и значения параметра 32 бит	PPO 3	6	Телеграмма с технологическими данными для передачи главного расчетного значения без данных параметра	PPO 4	10	Расширенная телеграмма с технологическими данными для передачи главного и вспомогательных расчетных значений без данных параметра	PPO 6	16	Телеграмма с 5 расчетными/действительными значениями.  <b>Внимание.</b> Этот тип телеграммы не поддерживается некоторыми устройствами.
Тип	Длина (LGE)	Описание																				
PPO 0	12	Стандартная телеграмма для передачи технологических данных и значения параметра 16 бит																				
PPO 1	14	Расширенная телеграмма для передачи технологических данных и значением параметра 32 бит																				
PPO 2	18	Расширенная телеграмма для передачи технологических данных (главного и двух вспомогательных расчетных значений) и значения параметра 32 бит																				
PPO 3	6	Телеграмма с технологическими данными для передачи главного расчетного значения без данных параметра																				
PPO 4	10	Расширенная телеграмма с технологическими данными для передачи главного и вспомогательных расчетных значений без данных параметра																				
PPO 6	16	Телеграмма с 5 расчетными/действительными значениями.  <b>Внимание.</b> Этот тип телеграммы не поддерживается некоторыми устройствами.																				
Address	Объект, содержащий адрес запрашиваемого устройства.																					
5.3.5 "Слово состояния"	Объект, содержащий бит состояния устройства.																					
5.3.6 "Слово управления"	Объект, содержащий управляющий бит (например, для разблокировки или быстрого останова).																					
Setpoint/actual value 1-5	Расчетные/действительные значения — это значения 16 или 32 бита. В зависимости от конфигурации устройства могут передавать разные параметра (например, уставка частоты или расчетное значение положения).																					
Format	<p>Объект, содержащий формат расчетного значения. Поддерживаются следующие форматы:</p> <ul style="list-style-type: none"> <li data-bbox="419 1697 742 1753">• нормализованное число 16 бит</li> <li data-bbox="419 1861 742 1917">• ненормализованное число 16 бит</li> </ul> <p data-bbox="758 1697 1396 1850">Этот формат позволяет интерпретировать расчетное значение как нормализованное число размером 16 бит. Нормализация означает линейное изменение диапазона с коэффициентом от -200% до 199% относительно базового значения (например, номинальной частоты).</p> <p data-bbox="758 1861 1396 1917">В этом случае расчетное значение интерпретируется как 16-битное число и передается без коэффициента.</p>																					

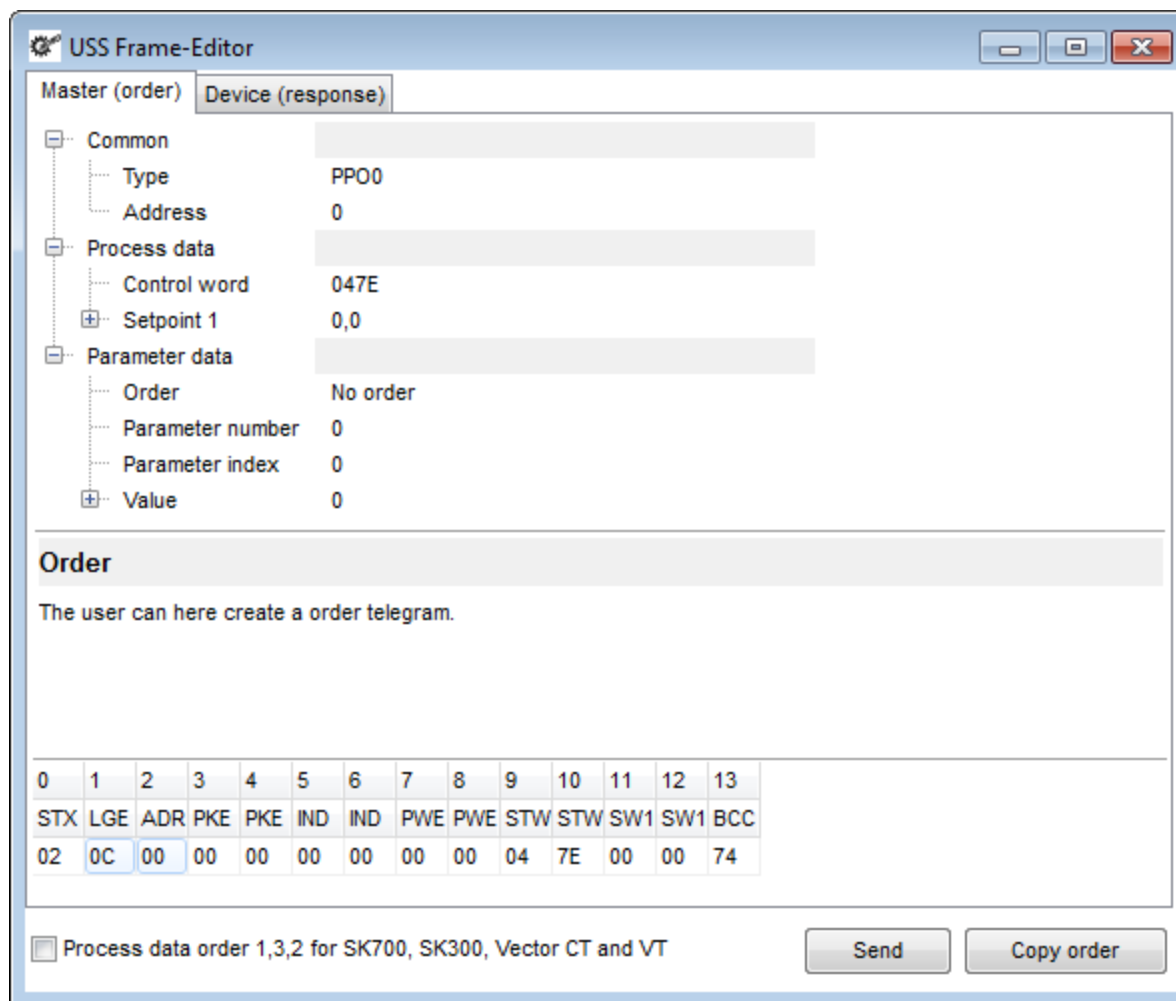
Parameter order	<p>Объект, содержащий запрос параметра. Определены следующие запросы:</p> <ul style="list-style-type: none"> <li>• Запрос значения параметра</li> <li>• Изменение значения параметра (16 бит)</li> <li>• Изменение значения параметра (32 бита)</li> <li>• Запрос значения параметра (массив)</li> <li>• Изменение значения параметра (массив, 16 бит)</li> <li>• Изменение значения параметра (массив, 32 бита)</li> <li>• Запрос числа элементов в массиве</li> <li>• Изменение значения параметра (двойное слово, массив) без сохранения в память EEPROM</li> <li>• Изменение значения параметра (слово, массив) без сохранения в память EEPROM</li> <li>• Изменение значения параметра (двойное слово) без сохранения в память EEPROM</li> <li>• Изменение значения параметра (слово) без сохранения в память EEPROM</li> </ul>
Parameter number	Объект, содержащий номер параметра.
Index	Объект, содержащий индекс параметра.
Value	Объект, содержащий значение параметра. В зависимости от типа телеграммы значение может иметь 16 или 32 бит. Вид значения зависит также от разрешения.
Resolution	Объект, содержащий разрешение параметра. При изменении разрешения меняется только вид значения. Величина разрешения указана в руководстве по эксплуатации конкретного устройства.

### Последовательность технологических значений 1,3,2 для SK700, SK300, Vector CT и VT

Эта опция позволяет менять последовательность для 2 и 3 технологических значений для устаревших устройств. Эта опция применима только к телеграммам типа PPO 2 и PPO 4. Последовательность технологических значений отображается в виде таблицы.

#### 9.1 Master (order)

Это окно имеет несколько разделов. В верхней части выводится телеграмма запроса в виде дерева. Структура дерева включает несколько категорий, в которых представлены разные части телеграммы. Пользователь может изменить любую запись, выводимую на белом фоне. Чтобы изменить запись, необходимо выделить ее с помощью мыши или клавиатуры. Нажав на запись второй раз, пользователь может открыть редактор значений. Внешний вид редактора отличается в зависимости от типа записи. Если поле имеет числовой тип, редактор можно открыть, нажав любое число на числовой клавиатуре. Чтобы сохранить новое значение и закрыть редактор, необходимо нажать на клавишу «Enter» или перейти к другой записи. Если новое значение не удастся сохранить, будет использоваться старое значение. Если редактор значений представляет из себя список для выбора, новое значение сохраняется, а редактор закрывается после выбора строки в списке. Если введенное значение не нужно сохранять, необходимо выйти из редактора, нажав клавишу «Esc». При выборе записи внизу дерева выводится описание этой записи. В нижней части выводится таблица с битами телеграммы. Отмеченные ячейки соответствуют записи, выбранной в дереве.



### Copy query

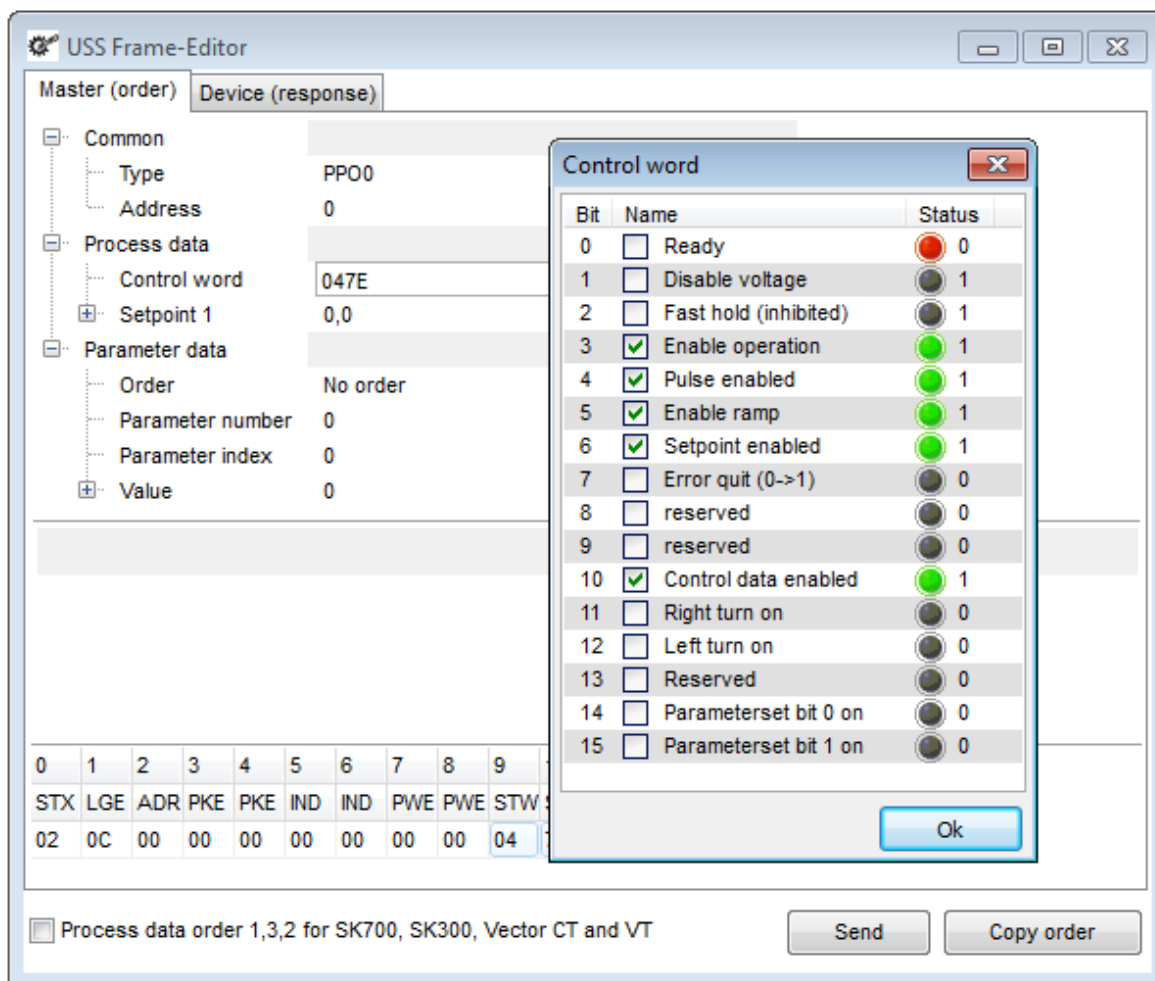
Преобразует телеграмму запроса в строку байтов в шестнадцатеричном формате и копирует строку в буфер обмена Windows.

## 9.2 Device (response)

Это окно имеет несколько разделов. В верхней части выводится таблица с байтами телеграммы ответа. В этой таблице пользователь может изменить значения отдельных байтов в ответной телеграмме. Изменить можно все байты вплоть до STX, LGE и BCC. Чтобы изменить, нужно выбрать строку и внести новое значение в таблицу. Изменение длины и структуры телеграммы выполняется из контекстного меню таблицы. Кроме того, в меню можно выбрать тип телеграммы.

Структура дерева обновляется после каждого изменения. Сама структура дерева служит для визуального представления разных частей телеграммы USS, поэтому ее нельзя изменить. Исключение: формат действительного значения и разрешение значения параметра. Эти данные не включены в телеграмму USS. Формат нужно задать в соответствии с настройками действительного значения. Разрешение также зависит от типа параметра. Эти значения указаны в руководстве по эксплуатации конкретного устройства.

Слово состояния представлено в дереве в шестнадцатеричном формате. Биты слова состояния отображаются в дополнительном окне. Чтобы открыть его, нужно выбрать слово состояния. Второй щелчок на записи открывает редактор значений в режиме чтения. После этого пользователь может открыть окно, нажав «...».



**Enter response**

Открывает окно для ввода телеграммы ответа. Телеграмма вводится в виде строки байтов в шестнадцатеричном формате.

## 10 ПЛК

### 10.1 Общая информация

Частотный преобразователь NORD серий SK 180E/SK 190E, SK 2xxE, SK 2xxE-FDS и SK 5xxE, а также пусковое устройство двигателя серии SK 155E-FDS/SK 175E-FDS имеют встроенную логическую систему обработки данных реализуемую на базе стандарта IEC61131-3, действующего для программируемых логических контроллеров (ПЛК/PLC). Вычислительная мощность и время отклика системы такого ПЛК позволяет ему выполнять простые вспомогательные задачи периферийного оборудования преобразователя. В частности, он может осуществлять контроль информации, поступающей через входы преобразователя или по полевой шине, анализировать ее и передавать на преобразователь уже обработанные расчетные значения. Совместно с другими приборами NORD он также позволяет оператору получать в графическом виде информацию о состояниях установки и вводить специальные параметры. Таким образом, обеспечивается экономия за счет отказа от использования внешних модулей ПЛК. Устройство поддерживает язык программирования AWL. AWL представляет собой машинный текстовый язык программирования, содержание и применение которого описываются стандартом IEC61131-3.

#### Информация

Программирование и установка на прибор осуществляются исключительно при помощи программного обеспечения NORD NORD CON.

#### 10.1.1 Спецификация ПЛК

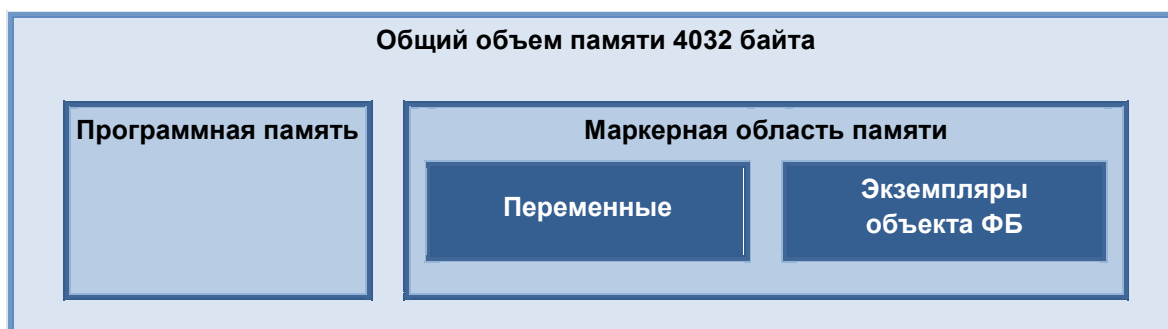
Функция	Спецификация		
Стандартная конфигурация	На базе IEC61131-3		
Язык	Instruction List ( IL ), strukturierter Text (ST)		
Задание	Циклическая задача, вызов программы каждые 5 мс		
Вычислительная мощность	Около 200 к.команд AWL за 1 мс		
Запоминающее устройство	SK 5xxE, SK 2xxE, SK 2x0E-FDS	SK 190E / SK 180E	SK 155E-FDS / SK 175E-FDS
	8128 байт для маркеров, функций и программы ПЛК	2032 байт для маркеров, функций и программы ПЛК	2028 байт для маркеров, функций и программы ПЛК
Максимальное количество команд	ок. 2580 команд	ок. 660 команд	ок. 660 команд
	<b>Примечание:</b> Данное значение является средним, использование большого числа маркеров, данных процессов и функций значительно уменьшает возможное количество строк, см. раздел "Ресурсы".		
Доступные буферы сообщений CAN Mailboxen	20		
Поддерживаемые приборы	SK 54xE SK 53xE / SK 52xE от V3.0		

Функция	Спецификация
	SK 2xxE от V2.0 SK 2x0E-FDS SK 180E / SK 190E SK 155E-FDS / SK 175E- FDS

## 10.1.2 Конструкция ПЛК

### 10.1.2.1 Запоминающее устройство

Запоминающее устройство ПЛК подразделяется на программную и идентификационную (маркерную) части. В маркерной части, помимо переменных, хранятся экземпляры объектов функциональных блоков (ФБ). Экземпляр объекта представляет собой область памяти, в которой хранятся все внутренние входные и выходные переменные функционального блока. Для описания каждого функционального блока требуется собственный экземпляр объекта. Граница между программной и маркерной областями памяти является динамической и зависит от размера маркерной области.



В маркерной памяти в области переменных сохраняются два различных класса:

#### [VAR]

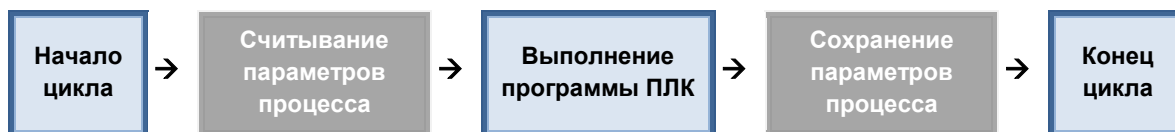
Переменные памяти для записи вспомогательной информации и состояний. Переменные этого типа заново устанавливаются при каждом запуске ПЛК. Содержимое памяти сохраняется во время выполнения цикла ПЛК.

#### [VAR\_ACCESS]

Служит для считывания и описания данных процессов (входы, выходы, уставки и т.д.) частотного преобразователя. Эти значения заново создаются при каждом цикле ПЛК.

### 10.1.2.2 Образ процесса

Работа прибора характеризуется множеством физических параметров, таких как вращающий момент, частота вращения, положение, входы, выходы и т.д. Эти параметры делятся на уставки и фактические значения. Образ процесса позволяет загружать и обрабатывать эти параметры. Требуемые параметры процессов определяются в списке переменных класса VAR\_ACCESS. Все заданные в списке переменных данные процессов преобразователя заново считываются при выполнении каждого цикла ПЛК. В конце каждого цикла ПЛК записываемые данные процессов снова передаются на преобразователь, см. рисунок далее.



Такой порядок выполнения определяет необходимость использования циклов при создании программы. Программирование циклов на ожидание определенного события (например, изменения уровня на входе) не дает желаемого результата. Функциональные блоки, которые обращаются к параметрам процессов, ведут себя иначе. Здесь параметры процессов считываются при вызове функционального блока и записываются сразу по окончании блока.

### **i** Информация

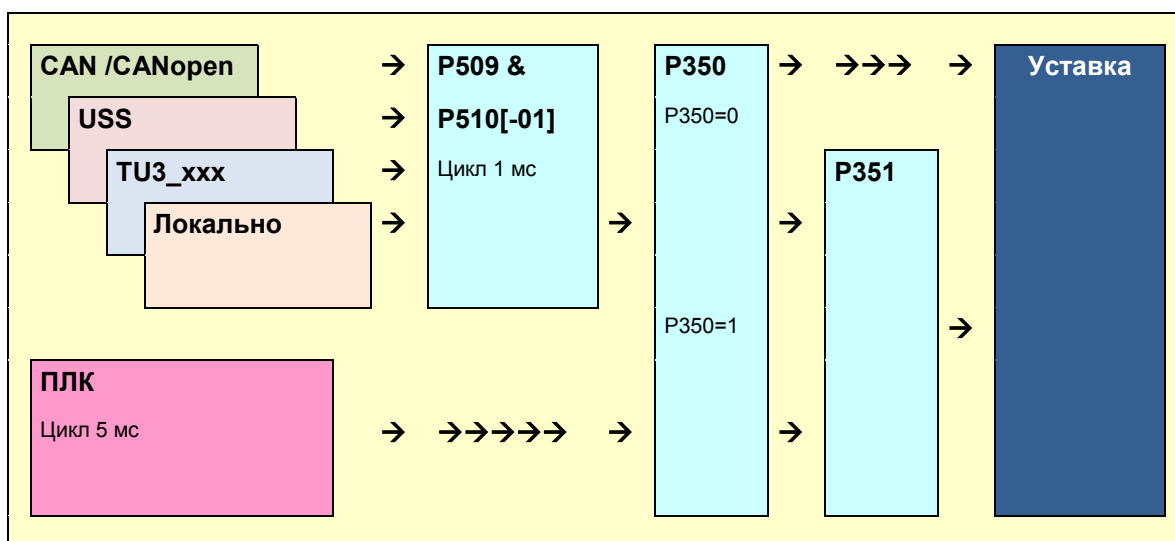
При использовании блоков управления движением (Motion) MC\_Power, MC\_Reset, MC\_MoveVelocity, MC\_Move, MC\_Home или MC\_Stop, не допускается использование параметра „PLC\_Control\_Word“ и параметров от „PLC\_Set\_Val1“ до „PLC\_Set\_Val5“. Иначе изменения функциональных блоков будут всегда перезаписываться значениями из списка переменных.

#### 10.1.2.3 Программная задача

Выполнение программы ПЛК происходит в рамках одной задачи. Обращение к задаче производится циклически каждые 5 мс, а максимальная длительность ее обработки не превышает 3 мс. Если большая программа не успевает обработаться за это время, то ее выполнение прерывается и возобновляется с началом следующего интервала в 5 мс для следующей задачи.

#### 10.1.2.4 Обработка уставки

Преобразователь получает уставки от множества источников, которые в конечном итоге соединяются друг с другом в единую результирующую частотного преобразователя посредством нескольких параметров.





При активированном ПЛК (P350=1) предустановка поступающих внешних уставок (главных уставок) осуществляется с помощью P509 & P510[-01]. Параметр P351 окончательно определяет, какие уставки будут использованы: от ПЛК или из входящих значений от P509/P510[-01]. Допускается смешанное использование уставок из обоих источников. Для вспомогательных уставок (P510[-02]) функция ПЛК не вызывает никаких изменений. Все источники вспомогательных уставок и ПЛК обладают равными правами при передаче своих вспомогательных уставок на частотный преобразователь.

#### **10.1.2.5 Обработка данных через накопительный регистр**

Накопительный регистр образует центральный вычислительный блок ПЛК. Почти все команды языка AWL выполняются исключительно совместно с накопительным регистром. ПЛК NORD имеет сразу три накопительных регистра: Akku1 и Akku2 размером 32 бита, а также текущий результат AE в формате BOOL. Переменная AE используется для всех булевых операций загрузки, сохранения и сравнения. При загрузке булевой величины она отображается при помощи AE. Операнды сравнения отправляют результат в переменную AE, и на основании ее значения выполняются условные переходы. Akku1 и Akku2 применяются для всех операндов в формате данных BYTE, INT и DINT. Akku1 представляет собой главный накопительный регистр, тогда как Akku2 выполняет вспомогательные функции. Все операнды загрузки и сохранения выполняются через Akku1. В Akku1 также сохраняются результаты всех арифметических операций. В Akku2 при выполнении каждой команды загрузки смещается содержимое Akku1. Следующий далее оператор может объединить оба регистра, либо произвести вычисления и снова сохранить результат в Akku1, далее в общем случае именуемый также „accumulator“.

#### **10.1.3 Функционал**

ПЛК поддерживает множество операторов, функций и стандартных функциональных блоков, описанных в стандарте IEC1131-3. Подробное описание представлено в следующих главах. Далее также описываются дополнительно поддерживаемые функциональные блоки.

##### **10.1.3.1 Библиотека управления движением Motion Control Lib**

Библиотека Motion Control Lib создана на базе спецификации PLCopen „Function blocks for motion control“. В ней преимущественно содержатся функциональные блоки, описывающие порядок работы привода. Дополнительно библиотека также предоставляет функциональные блоки для чтения и записи параметров оборудования.

##### **10.1.3.2 Электронные редукторы с летучей пилой**

Частотный преобразователь обладает функциями электронного редуктора (синхронизация движения в режиме позиционирования) и летучей пилы. Благодаря этим функциям преобразователь может перемещаться синхронно с другим приводом. Кроме того дополнительная функция летучей пилы обеспечивает возможность позиционно точной синхронизации с движущимся приводом. Запустить и отключить режим электронного редуктора можно в любой момент. Команды перемещения и функции редуктора могут комбинироваться с классическим контролем положения. Для работы редуктора на ведущем устройстве обязательно должен быть установлен частотный преобразователь NORD с внутренней шиной CAN-Bus.

##### **10.1.3.3 Визуализация**

С помощью модулей ControlBox или ParameterBox может осуществляться визуализация рабочего состояния и параметризация частотного преобразователя. В качестве альтернативы для отображения информации может также использоваться функционал ведущего устройства CANopen панели ПЛК с шиной CAN-Bus.

## ControlBox

Самым простым способом визуализации является использование модуля ControlBox. Доступ к 4-разрядному дисплею и состоянию клавиатуры может осуществляться посредством двух параметров. Это позволяет очень быстро создавать простые приложения человеко-машинного интерфейса (HMI). Чтобы ПЛК мог получать доступ к индикации для параметра P001 должно быть установлено значение „PLC-Controlbox Value“. Следует обратить внимание на то, что меню параметров больше не будет вызываться при помощи кнопок со стрелками. Вместо этого нужно будет одновременно нажимать кнопки „On“ и „Enter“.

## ParameterBox

В режиме визуализации посредством ПЛК на дисплее P-Box может использоваться каждый из 80 символов (4 строки по 20 символов). С их помощью могут отображаться как цифры, так и текст. Кроме того, ПЛК может передавать сигналы ввода с клавиатуры на P-Box. Это позволяет реализовывать сложные функции интерфейса HMI (индикация фактических значений, смена изображений, передача уставок и пр.). Доступ к индикации P-Box осуществляется посредством функциональных блоков в ПЛК. Для визуализации используются индикаторы рабочего состояния ParameterBox. Содержание индикаторов рабочего состояния настраивается при помощи параметра P1003 в P-Box. Этот параметр располагается в пункте основного меню „Display“. Для параметра P1003 должно быть установлено значение „PLC Display“. После этого можно снова осуществлять выбор индикаторов рабочих значений при помощи кнопок со стрелками "вправо" и "влево". Теперь здесь будет отображаться дисплей, управляемый ПЛК. После перезапуска оборудования настройки сохраняются.

### 10.1.3.4 Регулятор технологического процесса

Регулятор технологического процесса представляет собой ПИД регулятор типа PID-T1 с ограниченной выходной величиной. За счет данного функционального блока сложные регуляторы, управляющие различными процессами, например регулировкой давления, могут быть сформированы в ПЛК значительно более эффективным способом, чем с помощью часто используемых двухточечных регуляторов.

### 10.1.3.5 Передача данных через CANopen

Помимо стандартных имеющихся каналов передачи данных ПЛК предлагает также дополнительные способы коммуникации. Он позволяет устанавливать дополнительные коммуникационные связи с другими приборами через интерфейс CAN Bus частотного преобразователя, либо через системную шину. Для этого используется протокол CANopen. Коммуникационные связи при этом ограничиваются передачей сообщений PDO и командами NMT. Такая функция ПЛК не влияет на стандартный для частотного преобразователя обмен данными по протоколу CANopen с помощью SDO, PDO1, PDO2 и Broadcast.

## PDO (Process Data Objects)


Сообщения PDO позволяют управлять и контролировать работу других частотных преобразователей. К ПЛК могут подключаться также приборы других производителей. Это могут быть модули входов-выходов, датчики CANopen, панели и пр. Это позволяет увеличить количество входов/выходов преобразователя, а также использовать аналоговые выходы.

## NMT (Network Management Objects)

Все приборы CANopen должны быть переведены с помощью Busmaster в состояние „Operational“ для шины CANopen Bus. Обмен сообщениями PDO возможен только при таком состоянии шины. Если шина CANopen Bus не имеет опции Busmaster, то

соответствующую настройку следует установить при помощи ПЛК. Для этого предназначен функциональный блок FB\_NMT.

## 10.2 Составление программы для ПЛК

Составление программы для ПЛК происходит исключительно в программе для ПKNORD CON. Открыть редактор программы для ПЛК можно через пункт меню „File/New/PLC program“ или с помощью символа . Эта кнопка доступна только если в обзоре устройств выбран прибор с функциями ПЛК.

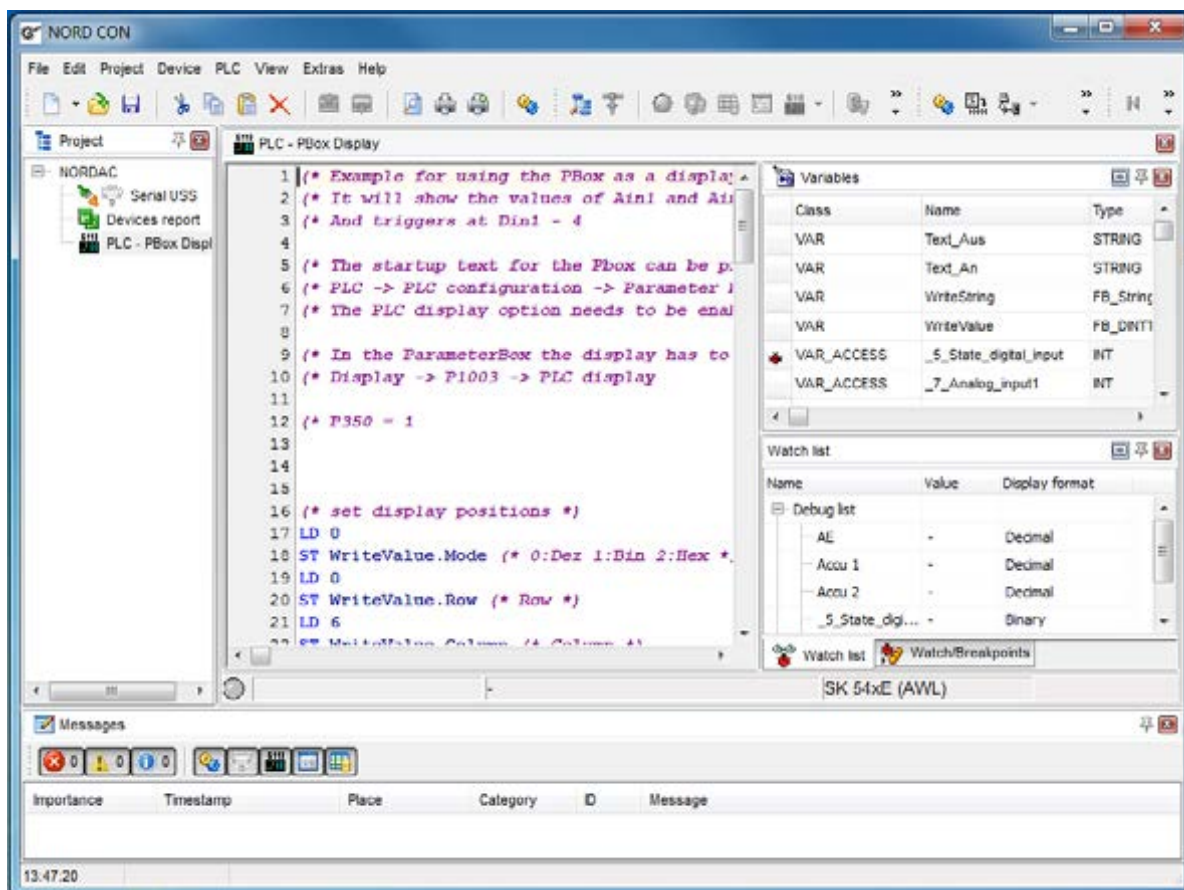
### 10.2.1 Загрузка, сохранение и печать

Функции загрузки, сохранения и печати реализуются при помощи соответствующих пунктов главного меню или символов. При открытии в диалоговом окне „Open“ рекомендуется установить тип файла „PLC program“ (\*.awlх, \*.nstx). При этом в окне будут показаны только те файлы, которые могут быть прочитаны редактором ПЛК. Для сохранения составленной программы для ПЛК должно быть активировано соответствующее окно редактора ПЛК. Сохранение выполняется при помощи команд „Save“ или „Save as“. Команда „Save as“ позволяет задать тип файла при сохранении (программа для ПЛК (\*.awlх\*.nstx)). Для вывода программы ПЛК на печать также следует активировать соответствующее окно. Для начала печати выбрать пункт „File/Print“ или нажать соответствующий символ.

Кроме того, программа для ПЛК может быть сохранена как защищенная программа для ПЛК. Для этого в меню выбора файла следует установить "Backed-up AWL files" или "Backed up ST files". В результате этого будет сохранена программа для ПЛК в зашифрованном формате (\*.awls или \*.nsts), а также обычная версия программы Version (\*.awlх, \*.nstx). Зашифрованная программа может быть передана на прибор (см. 2.2.4 "Категория «Device»").

### 10.2.2 Редактор

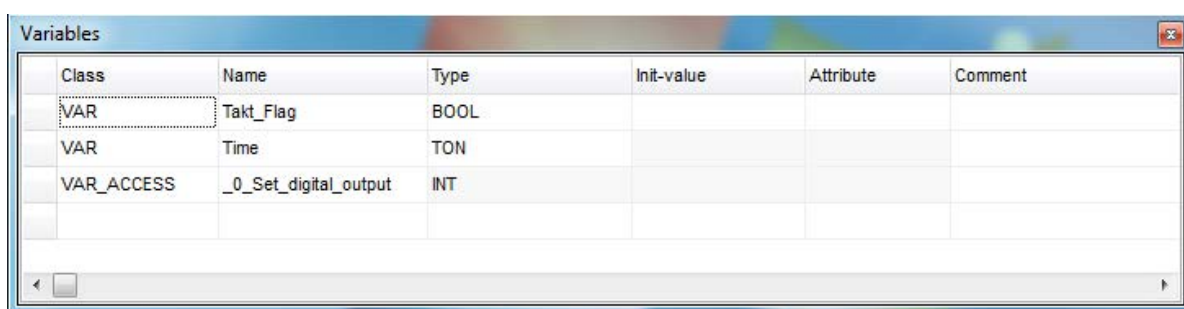
Редактор программы для ПЛК имеет четыре окна.



Отдельно о каждом окне рассказывается в следующих разделах руководства.

### 10.2.2.1 Переменные и описание функционального блока

В этом окне задаются все необходимые программе переменные, параметры процессов и функциональные блоки.



#### Переменные

Переменные создаются путем выбора „VAR“ в поле класса. Переменная может иметь любое имя. Для переменной следует выбрать один из типов: BOOL, BYTE, INT или DINT. Для переменной может быть задано начальное значение.

#### Параметры процессов

Для создания параметра в поле класса следует установить значение „VAR\_ACCESS“. Имя параметра не может быть задано произвольно, а поле начального значения для него недоступно.

### **Функциональные блоки**

В поле класса выбрать значение „VAR“. Для соответствующего экземпляра объекта ФБ может быть выбрано любое имя. Требуемый ФБ выбирается в поле типа. Начальное значение для ФБ не устанавливается.

Все пункты меню, связанные с окном переменных, вызываются через контекстное меню. С их помощью можно добавлять и удалять записи, а также активировать переменные и переменные процессов для отслеживания (функция Watchpoint) или отладки (Breakpoint).

#### **10.2.2.2 Окно ввода**

Окно ввода служит для ввода программы, а также представления программы на языке AWL. Оно обладает следующими функциями:

- Выделение синтаксиса
- Закладки
- Описание переменных
- Отладка

#### **Выделение синтаксиса**

Если редактор распознал команду и соответствующую ей переменную, то команда выделяется синим, а переменная черным цветом. Пока этого не произошло, то текст отображается тонким, наклонным, черным шрифтом.

#### **Закладка**

Так как программы в редакторе могут достигать значительных размеров, функция закладки позволяет отметить в тексте программы важные участки для быстрого перехода к ним. Чтобы отметить строку следует установить на нее курсор. Выбрать пункт меню „Switch bookmark“ (меню правой кнопки мыши). На строку будет установлена требуемая закладка. Для перехода к закладке использовать пункт меню „Go to bookmark“.

#### **Описание переменных**

Новые переменные можно описать с помощью редактора, используя меню редактора „Add Variable“ (меню правой кнопки мыши).

#### **Отладка**

Для функции отладки в редакторе устанавливаются позиции контрольных точек и точек прерывания. Для этого используются пункты меню „Switch breakpoint“ (точки прерывания) и „Switch monitoring point“ (контрольные точки). Задать положение точки прерывания можно также нажатием клавишей мышки на левой границе окна редактора. Переменные и параметры процессов, которые во время отладки требуется считывать с частотного преобразователя, должны быть выделены. Это можно сделать в редакторе с помощью пунктов меню „Debug variable“ и „Watch variable“. Нужные переменные следует выделить перед переходом в соответствующий пункт меню.

#### **10.2.2.3 Окна отображения контрольной точки и точки прерывания (Watch- & Breakpoint)**

В этом окне имеется две вкладки, описанных далее.

### Точки прерывания

В этом окне отображаются все установленные точки прерывания и контрольные точки. Их включение/отключение выполняется при помощи установки и снятия флажков в соответствующих ячейках. Для удаления используется „Delete key“. Соответствующее меню открывается нажатием правой кнопки мыши.

### Список отслеживания

Здесь отображаются все переменные, выбранные для отслеживания. В поле значения представлено текущее содержимое переменной. В поле индикации можно выбрать формат представления.

#### 10.2.2.4 Окно сообщений ПЛК

В этом окне вводятся все сообщения о состоянии и ошибках ПЛК. Если трансляция программы выполнена без ошибок в окне появляется сообщение „Translated without error“. На строку ниже представлена информация об использовании ресурсов. При наличии ошибок в программе ПЛК появляется сообщение „Error X“, значение X соответствует количеству ошибок. В следующих строках отображается конкретное сообщение об ошибке в формате:

[номер строки]: описание ошибки

### 10.2.3 Передача программы ПЛК на прибор

Существует несколько способов передачи программы ПЛК на прибор.

#### Передача программы ПЛК напрямую:

1. Выбрать прибор в дереве проекта.
2. Открыть контекстное меню (нажать правую клавишу мыши)
3. Выполнить функцию "Transfer PLC program to device"
4. Выбрать файл в окне выбора файла и нажать "Open"


#### Передача программы ПЛК при помощи редактора ПЛК (оффлайн):

1. Открыть программу ПЛК при помощи функции "Open" (Datei->Öffnen)
2. Соединить редактор ПЛК с прибором (PLC->Verbinden)
3. Транслировать программу ПЛК
4. Передать программу ПЛК на прибор

#### Передача программы ПЛК при помощи редактора ПЛК (онлайн):

1. Отметить прибор в дереве проекта.
2. Запустить редактор ПЛК
3. Открыть программу ПЛК
4. Импортировать программу ПЛК в онлайн-вид



5. Транслировать программу ПЛК
6. Передать программу ПЛК на прибор 

## Информация

### SK 1xxE-FDS - ограниченное количество циклов записи

В приборах SK 155E-FDS / SK 175E-FDS в качестве среды хранения используется флеш-накопитель. Количество циклов записи для флеш-накопителей строго ограничено. Поэтому стандартная загрузка программы выполняется только в оперативную память RAM. Она может быть запущена и протестирована. Если после этого требуется перезапустить ПЛК, то программу следует повторно загрузить на прибор, чтобы установить переменные ПЛК. Если требуется сохранить программу на приборе на длительное время, пользователь должен выполнить действие "Transfer and store program to device".

## 10.2.4 Отладка

Поскольку очень редко программы запускаются с первого раза, в ПЛК NORD предусмотрен ряд возможностей для обнаружения ошибок. Их можно разделить на два пункта, о которых далее будет рассказано подробно.

### 10.2.4.1 Контрольные точки (Watchpoints)


Функция Watchpoint является самым простым вариантом отладки. Она обеспечивает быструю проверку поведения отдельных переменных. Для этого в любом месте программы назначается контрольная точка. Когда ПЛК обрабатывает эту программную строку, функция сохраняет до 5 значений и отображает в списке отслеживания (окно „Observation list“). Выбрать 5 отслеживаемых значений можно в окне ввода или окне переменных через контекстное меню. Если контрольная точка назначена в месте без программного кода, NORD CON выполняет поиск предшествующей строки кода. При достижении этой строки кода в ходе выполнения программы происходит обновление значений. Если контрольная точка была пропущена в результате выполнения перехода (JMP, IF, Switch), то значения не обновляются.

## Информация

В текущей версии переменные функциональных блоков не могут быть добавлены в список отслеживания!

### 10.2.4.2 Точки прерывания (Breakpoints)


Точки прерывания позволяют остановить выполнение программы ПЛК на определенной строке. Когда ПЛК доходит до точки прерывания происходит считывания текущих значений Akku1 и Akku2, а также всех переменных, выбранных через пункт меню „Debug variable“ (контекстное меню). В программе ПЛК можно установить до 5 точек прерывания. Запуск функции


осуществляется при помощи символа . Программа выполняется до тех пор пока не сработает точка прерывания. При повторном нажатии кнопки с символом программа запускается далее и выполняется до следующей точки прерывания. Если требуется



продолжить выполнение программы без прерываний используется символ .

### 10.2.4.3 Пошаговое выполнение (Single Step)


Этот метод отладки позволяет обрабатывать программу для ПЛК строка за строкой. При выполнении каждого отдельного шага программы все выбранные переменные считываются с прибора- и отображаются в окне „Observation list“. Выбрать отслеживаемые значения можно в окне ввода или окне переменных с помощью меню правой кнопки мыши. Обязательным условием работы пошагового метода является назначение минимум одной точки прерывания

перед началом отладки. Для включения режима отладки используется символ . После достижения программой первой точки прерывания начинается отладка последующих строк

программы в пошаговом режиме при помощи символа . За некоторыми командными строками скрываются несколько отдельных команд. Из-за этого может произойти так, что два или более отдельных шага будут обработаны до того как в окне ввода произойдет переход к отображению следующего шага. Текущее положение указывается маленькой стрелкой в левом

окне редактора ПЛК. При нажатии на символ  выполнение программы продолжается до следующей точки прерывания. Если требуется продолжить выполнение программы без прерываний используется символ .

### 10.2.5 Конфигурация ПЛК

Диалоговое окно конфигурации ПЛК открывается при помощи символа . Здесь устанавливаются некоторые основные настройки ПЛК, о которых подробно рассказывается далее.

#### Контроль времени цикла

Функция контролирует максимальное время обработки для цикла ПЛК. Это позволяет отделять ошибочно запрограммированные непрерывные циклы в программе для ПЛК. В случае превышения допустимой продолжительности цикла на частотном преобразователе возникает ошибка E22.4.

#### Разрешить использование ФБ ParameterBox

Если программа ПЛК использует для визуализации модуль ParameterBox, следует активировать данную опцию. Иначе в соответствующих функциональных блоках при запуске частотного преобразователя будет возникать ошибка компилятора.

#### Недопустимые управляющие данные

ПЛК может анализировать команды управления, поступающие через возможные имеющиеся системы шин. При этом команды пропускаются только в том случае, если установлен бит „PZD valid“ (бит 10). Чтобы ПЛК мог анализировать также команды управления, не соответствующие протоколу USS, следует активировать данную опцию. Тогда бит 10 первой команды опрашиваться не будет.

#### "Теплый" запуск после ошибки

При запуске ПЛК все переменные всегда загружаются с начальным значением "0". При этом не важно, выполняется ли запуск после остановки, загрузки программы или ошибки ПЛК. При использовании опции теплого запуска содержимое переменных не изменяется. Теплый запуск выполняется после срабатывания команды остановки ПЛК или ошибки.

#### Не останавливать системное время в точке прерывания

В ходе отладки, если ПЛК находится в точке прерывания или в пошаговом режиме, системное время останавливается. Системное время служит основой для работы всех таймеров ПЛК.



Если требуется не останавливать системное время во время отладки, необходимо активировать данную функцию.

### 10.3 Функциональные блоки

Функциональные блоки - это небольшие программы, которые могут сохранять свои параметры состояния во внутренних переменных. По этой причине для каждого функционального блока в списке переменных должен быть создан свой экземпляр объекта NORD CON. Например, если таймер должен параллельно контролировать три времени, то и в списке переменных он должен быть задан три раза.

#### **i** Информация

#### Определение фронта сигнала

Чтобы последующие функциональные блоки могли распознавать фронт сигнала на входе нужно чтобы вызов функции был произведен на входе с различными состояниями.

#### 10.3.1 CANopen

С помощью функциональных блоков ПЛК может конфигурировать каналы сообщений PDO, управлять ими и отправлять на них сигналы. Сообщения PDO позволяют ПЛК отправлять или принимать до 8 бит информации о процессах. Обращение к каждому из таких сообщений PDO выполняется через собственный адрес (COB-ID). ПЛК может конфигурировать до 20 сообщений PDO. Для более эффективной работы эти сообщения не отправляются напрямую COB-ID. Вместо этого адрес прибора и номер PDO передаются в ФБ. Адрес COB-ID определяется в итоге на основании заранее определенных связей (CiA DS301). Таким образом для ПЛК могут быть получены нижеследующие COB-ID.

Отправка сообщений PDO		Контролируемые PDO	
PDO	COB-ID	PDO	COB-ID
PDO1	200h + адрес устройства	PDO1	180h + адрес устройства
PDO2	300h + адрес устройства	PDO2	280h + адрес устройства
PDO3	400h + адрес устройства	PDO3	380h + адрес устройства
PDO4	500h + адрес устройства	PDO4	480h + адрес устройства

NORD Частотный преобразователь используют PDO1 для передачи данных процессов, а PDO2 используется только для передачи уставок/фактических значений 4 и 5.

#### 10.3.1.1 Обзор

Функциональный блок	Описание
FB_PDConfig	Конфигурация PDO
FB_PDOSend	Отправка PDO
FB_PDORceive	Получение PDO
FB_NMT	Разблокировка и блокировка PDO

### 10.3.1.2 FB\_NMT

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

После выполнения *Power UP* все абоненты CAN находятся в состоянии шины Pre-Operational. В этом состоянии они не могут принимать или отправлять сообщения PDO. Чтобы ПЛК мог осуществлять обмен данными с другими абонентами они должны находиться в состоянии Operational. Как правило за это отвечает Busmaster. Если Busmaster отсутствует, то эту задачу может выполнять блок FB\_NMT. Входы **PRE**, **OPE** или **STOP** позволяют менять состояние всех абонентов, подключенных к шине. Положительный фронт переводит входы в состояние **EXECUTE**. Функция должна вызываться до тех пор, пока на выходе не будет получено значение 1 для **DONE** или **ERROR**.

Получение значения 1 на выходе **ERROR** означает либо отсутствие напряжения 24В на шине RJ45 CAN преобразователя, либо драйвер CAN преобразователя находится в состоянии *Bus off*. Отрицательный фронт **EXECUTE** возвращает 0 на все выходы.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
EXECUTE	Выполнение	BOOL	DONE	Отправка команды NMT	BOOL
PRE	Перевод всех абонентов в состояние Pre-Operational	BOOL	ERROR	Ошибка в ФБ	BOOL
OPE	Перевод всех абонентов в состояние Operational	BOOL			
STOP	Перевод всех абонентов в состояние Stopped	BOOL			

### 10.3.1.3 FB\_PDOConfig

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X		

Данный ФБ используется для конфигурации PDO. Экземпляр объекта данной функции позволяет выполнить настройку всех нужных PDO. Для каждого PDO требуется вызвать ФБ только один раз. При этом может быть настроено до 20 PDO. Каждое PDO обладает своим набором параметров. Назначение PDO другим ФБ CANopen производится при помощи номера окна сообщений. **TARGETID** содержит адрес устройства. У NORD Частотный преобразователь для настройки используется P515 или DIP-переключатель. Для PDO указывается требуемый номер окна сообщений (см. введение). **LENGTH** содержит отправляемую длину PDO. **DIR** определяет направление отправки/приема. При положительном фронте на **EXECUTE** происходит передача данных. Запрос на выход **DONE** может быть отправлен сразу после вызова ФБ. Если **DONE** имеет значение 1, выполняется конфигурирование канала PDO. Если **ERROR** = 1, значит возникла проблема, причина которой хранится в **ERRORID**. Отрицательный фронт **EXECUTE** возвращает 0 на все выходы.

Отправка сообщений PDO		Контролируемые PDO	
PDO	COB-ID	PDO	COB-ID
PDO1	200h + адрес устройства	PDO1	180h + адрес устройства
PDO2	300h + адрес устройства	PDO2	280h + адрес устройства
PDO3	400h + адрес устройства	PDO3	380h + адрес устройства
PDO4	500h + адрес устройства	PDO4	480h + адрес устройства
PDO5	180h + адрес устройства	PDO5	200h + адрес устройства
PDO6	280h + адрес устройства	PDO6	300h + адрес устройства
PDO7	380h + адрес устройства	PDO7	400h + адрес устройства
PDO8	480h + адрес устройства	PDO8	500h + адрес устройства

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>EXECUTE</b>	Выполнение	BOOL	<b>DONE</b>	Конфигурация PDO	BOOL
<b>NUMBER</b>	Номер окна сообщений Диапазон значений = от 0 до 19	BYTE	<b>ERROR</b>	Ошибка в ФБ	BOOL
<b>TARGETID</b>	Адрес прибора Диапазон значений = от 1 до 127	BYTE	<b>ERRORID</b>	Код ошибки	INT
<b>PDO</b>	PDO Диапазон значений = от 1 до 4	BYTE			
<b>LENGTH</b>	Длина PDO Диапазон значений = от 1 до 8	BYTE			
<b>DIR</b>	Отправка или прием Отправка = 1 / прием = 0	BOOL			
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1800h	Превышен диапазон значений				
1801h	Превышен диапазон значений TARGETID				
1802h	Превышен диапазон значений PDO				
1803h	Превышен диапазон значений LENGT				

**i** **Информация**

**Исключение двойного использования CAN ID**

Не допускается параметрирование CAN-ID, который уже используется устройством!

Связанные адреса приема:

- CAN ID = 0x180 + P515[-01] PDO1
- CAN ID = 0x180 + P515[-01]+1 CAN ID для абсолютного энкодера
- CAN ID = 0x280 + P515[-01] PDO2

Связанные адреса отправки:

- CAN ID = 0x200 + P515[-01] PDO1
- CAN ID = 0x300 + P515[-01] PDO2

**Пример на ST:**

```
(* Конфигуратор PDO*)
PDOConfig(
  Execute := TRUE,
  (* Конфигурация окна сообщения Messagebox 1 *)
  Number := 1,
  (* Установить номер узла CAN *)
  TargetID := 50,
  (* Выбор PDO (стандарт для команды управления PDO1, Sollwert1, Sollwert2, Sollwert3) *)
  PDO := 1,
  (* Определить длину данных (стандарт для PDO1 равен 8 *)
  LENGTH := 8,
  (* Отправка *)
  Dir := 1);
```

или

```
(* Конфигуратор PDO*)
PDOConfig(
  Execute := TRUE,
  (* Конфигурация окна сообщения Messagebox 1 *)
  Number := 2,
  (* Установить номер узла CAN *)
  TargetID := 50,
  (* Выбор PDO (стандарт для команды управления PDO2 Sollwert4, Sollwert SK540E) *)
  PDO := 2,
  (* Определить длину данных (стандарт для PDO2 равен 4 *)
  LENGTH := 4,
  (* Отправка *)
  Dir := 1);
```

или

```
(* Конфигуратор PDO*)
PDOConfig(
  Execute := TRUE,
  (* Конфигурация окна сообщения Messagebox 2 *)
  Number := 2,
  (* Установить номер узла CAN *)
  TargetID := 50,
  (* Выбор PDO (стандарт для команды управления PDO1, Istwert1, Istwert2, Istwert3) *)
  PDO := 1,
  (* Определить длину данных (стандарт для PDO1 равен 8 *)
  LENGTH := 8,
  (* Получение *)
  Dir := 0);
```

**10.3.1.4 FB\_PDORceive**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X		

Этот ФБ осуществляет контроль входящих сообщений для предварительно сконфигурированного канала PDO. Функция запускается когда входу **ENABLE** соответствует значение 1. После вызова функции выполняется проверка выхода **NEW**. Если сообщение поступило, то выходу будет соответствовать значение 1. Содержимое выхода **NEW** удаляется при следующем вызове функции. Переменные с **WORD1** по **WORD4** хранят полученные данные. Переменная **TIME** позволяет контролировать циклический прием сообщений через канал PDO. В **TIME** указывается значение от 1 до 32767 мс, которое соответствует временному промежутку приема сообщений. Иначе в ФБ переходит в состояние ошибки (**ERROR** = 1). Значение 0 отключает функцию Таймер контролирующей функции работает с интервалами 5 мс. В случае ошибки **ERROR** принимает значение 1. В этом случае значение **DONE** равно 0. **ERRORID** содержит соответствующий код ошибки. При отрицательном фронте на **ENABLE** производится сброс значений **DONE**, **ERROR** и **ERRORID**.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Выполнение	BOOL	<b>NEW</b>	Получение нового PDO	BOOL
<b>NUMBER</b>	Номер окна сообщений Диапазон значений = от 0 до 19	BYTE	<b>ERROR</b>	Ошибка в ФБ	BOOL
<b>TIME</b>	Функция Watchdog Диапазон значений = от 0 до 32767 0 = выкл. от 1 до 32767 = время контроля	INT	<b>ERRORID</b>	Код ошибки	INT
			<b>WORD1</b>	Принимаемые данные Слово данных 1	INT
			<b>WORD2</b>	Принимаемые данные Слово данных 2	INT
			<b>WORD3</b>	Принимаемые данные Слово данных 3	INT
			<b>WORD4</b>	Принимаемые данные Слово данных 4	INT
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1800h	Превышен диапазон значений				
1804h	Неверная конфигурация выбранного окна				
1805h	Отсутствует напряжение 24 В на драйвере шины или он находится в состоянии „Bus off“				
1807h	Timeout получения (функция Watchdog)				

## Информация

## Время цикла ПЛК

Цикл ПЛК составляет 5 мс, то есть при вызове функции в программе ПЛК каждые 5 мс может считываться только одно сообщение CAN. Если сообщения отправляются быстрее, то они могут перезаписываться.

### Пример на ST:

```
IF bFirstTime THEN
  (* Перевести устройства в состояние Pre-Operational *)
```

```

NMT(Execute := TRUE, OPE := TRUE);
IF not NMT.Done THEN
  RETURN;
END_IF;

(* Конфигуратор PDO*)
PDOConfig(
  Execute := TRUE,
  (* Конфигурация окна сообщения Messagebox 2 *)
  Number := 2,
  (* Установить номер узла CAN *)
  TargetID := 50,
  (* Выбор PDO (стандарт для команды управления PDO1, Istwert1, Istwert2, Istwert3) *)
  PDO := 1,
  (* Определить длину данных (стандарт для PDO1 равен 8 *)
  Length := 8,
  (* Получение *)
  Dir := 0);
END_IF;

(* Считывание состояний и факт.значений *)
PDOReceive(Enable := TRUE, Number := 2);
IF PDOReceive.New THEN
  State := PDOReceive.Word1;
  Sollwert1 := PDOReceive.Word2;
  Sollwert2 := PDOReceive.Word3;
  Sollwert3 := PDOReceive.Word4;
END_IF

```

### 10.3.1.5 FB\_PDOsend

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X		

Этот ФБ позволяет отправлять сообщения PDO на предварительно сконфигурированный канал. Сообщения могут отправляться однократно или циклически. Данные для отправки заносятся в переменные **WORD1** - **WORD4**. Отправка PDO может выполняться независимо от состояния CANopen частотного преобразователя. Выбор предварительно сконфигурированного канала производится при помощи **NUMBER**. В **WORD1** - **WORD4** заносятся данные для отправки. С помощью переменной **CYCLE** осуществляется выбор между однократной отправкой (значение=0) или циклической. При положительном фронте **EXECUTE** выполняется отправка PDO. Если **DONE** = 1, значит все данные были корректны и PDO успешно отправлены. Если **ERROR** = 1, значит произошла ошибка. Точная причина ошибки указывается в **ERRORID**. При отрицательном фронте **EXECUTE** значения всех выходов сбрасываются. Опорное время ПЛК составляет 5 мс, что также действительно для входа **CYCLE**. Циклы отправки могут выполняться только с кратностью 5 мс.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>EXECUTE</b>	Выполнение	<b>BOOL</b>	<b>DONE</b>	PDO отправлено = 1	<b>BOOL</b>
<b>NUMBER</b>	Номер окна сообщений Диапазон значений = от 0 до 19	<b>BYTE</b>	<b>ERROR</b>	Ошибка в ФБ	<b>BOOL</b>
<b>CYCLE</b>	Цикл отправки Диапазон значений =	<b>BYTE</b>	<b>ERRORID</b>	Код ошибки	<b>INT</b>

	от 0 до 255 0 = выкл. от 1 до 255 = время цикла в мс				
<b>WORD1</b>	Отправляемые данные Слово данных 1	<b>INT</b>			
<b>WORD2</b>	Отправляемые данные Слово данных 2	<b>INT</b>			
<b>WORD3</b>	Отправляемые данные Слово данных 3	<b>INT</b>			
<b>WORD4</b>	Отправляемые данные Слово данных 4	<b>INT</b>			
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1800h	Превышен диапазон значений				
1804h	Неверная конфигурация выбранного окна				
1805h	Отсутствует напряжение 24 В на драйвере шины или он находится в состоянии „Bus off“				

Когда **DONE** принимает значение 1, сообщение для отправки передается на модуль CAN, но не отправляется. Собственно отправка происходит фоново. Если ФБ предусматривает отправку нескольких сообщений одно за другим, может возникнуть ситуация, когда при новом вызове программы предыдущее сообщение еще не отправлено. Определить это можно по отсутствию значения 1 у сигналов **DONE** и **ERROR** после вызова **CAL**. В этом случае следует повторять вызов **CAL**, пока не будет получено значение 1 для одного из сигналов. Если один должен ФБ описывать несколько различных CAN-ID, это можно выполнить при помощи новой конфигурации ФБ. При этом это не должен быть тот же цикл ПЛК, что и у отправки. Иначе возникает опасность, что в процессе конфигурации предназначенное для отправки сообщение будет удалено функцией FB\_PDOConfig.

#### Пример на ST:

```

IF bFirstTime THEN
  (* Перевести устройства в состояние Pre-Operational *)
  NMT(Execute := TRUE, OPE := TRUE);
  IF not NMT.Done THEN
    RETURN;
  END_IF;

  (* Конфигуратор PDO*)
  PDOConfig(
    Execute := TRUE,
    (* Конфигурация окна сообщения Messagebox 2 *)
    Number := 2,
    (* Установить номер узла CAN *)
    TargetID := 50,
    (* Выбор PDO (стандарт для команды управления PDO1, Istwert1, Istwert2, Istwert3) *)
    PDO := 1,
    (* Определить длину данных (стандарт для PDO1 равен 8 *)
    Length := 8,
    (* Получение *)
    Dir := 0);
  END_IF;

  (* Считывание состояний и факт.значений *)
  PDOReceive(Enable := TRUE, Number := 2);
  IF PDOReceive.New THEN
    State := PDOReceive.Word1;
  
```

```
Sollwert1 := PDOReceive.Word2;
Sollwert2 := PDOReceive.Word3;
Sollwert3 := PDOReceive.Word4;
END_IF
```

### 10.3.2 Электронные редукторы с летучей пилой\*

Для функции *электронного редуктора* („синхронизация с вращением на один и тот же угол за один и тот же промежуток времени“) и вспомогательной функции *летучей пилы* предусмотрено два функциональных блока, обеспечивающих управление этими функциями. Кроме того, для выполнения обоих функциональных блоков на ведущем и ведомом частотных преобразователях могут быть настроены разные параметры. В качестве примера в нижеследующей таблице рассмотрена реализация для SK 540E.

Ведущий ЧП			Ведомый ЧП		
Параметр	Настройка	Значение	Параметр	Настройка	Значение
P502[-01]	20	Уставка частоты по линейному изменению	P509	10 *	Широкое вещание CANopen Broadcast*
P502[-02]	15	Текущая позиция в инкр. High – Word	P510[-01]	10	Широкое вещание CANopen Broadcast
P502[-03]	10	Текущая позиция в инкр. Low – Word	P510[-02]	10	Широкое вещание CANopen Broadcast
P503	3	CANopen	P505	0	0,0 Гц
P505	0	0,0 Гц	P515[-02]	P515[-03] <sub>Master</sub>	Адрес Broadcast Slave
P514	5	250 Кбод (мин. 100 Кбод)	P546[-01]	4	Сложение частот
P515[-03]	P515[-02] <sub>Slave</sub>	Адрес Broadcast Master	P546[-02]	24	Уст.полож. в инкр. High – Word
			P546[-03]	23	Уст.полож. в инкр. Low – Word
			P600	1,2	Контроль положения
			Только для ФБ FB_Gearing		
			P553[-01]	21	Полож. Уставка полож. Low Word
			P553[-02]	22	Полож. Уставка полож. High Word

\* (P509) не обязательно должен быть {10} „CANopen Broadcast“. (P502 [-01]) на ведущем устройстве для настройки {21} "Фактическая частота без проскальзывания".



#### Информация

#### Формат передачи фактического положения

Фактическое положение ведущего устройства должно передаваться в формате „Increments“ (Inc).



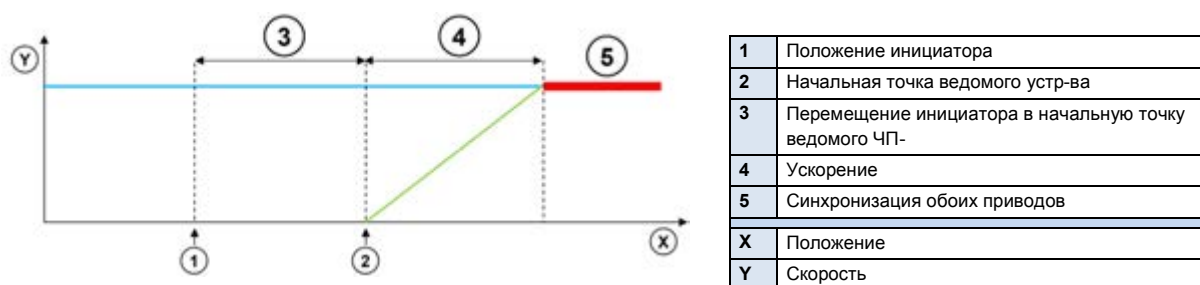
## 10.3.2.1 Обзор

Функциональный блок	Описание
FB_Gearing	ФБ для простой функции редуктора
FB_FlyingSaw	ФБ для функции редуктора с летучей пилой.

## 10.3.2.2 FB\_FlyingSaw

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	

Функция летучей пилы представляет собой дополнение к функции редуктора. Функция летучей пилы обеспечивает возможность позиционно точной синхронизации с движущимся приводом. Синхронизация выполняется противоположно относительно функции FB\_Gearing, то есть ведомая ось перемещается синхронно в позицию ведущего, занимаемую при запуске функции „Flying Saw“. Процесс синхронизации представлен на следующем рисунке.



При запуске функции ведомый частотный преобразователь ускоряется до скорости ведущей оси. Линейное изменение ускорения определяется посредством кривой **ACCELERATION**. При низкой скорости линейное изменение будет более плоским, а при более высокой скорости ведущего устройства линейное изменение для ведомого частотного преобразователя будет иметь более крутой вид. Кривая ускорения задается в оборотах (1000 = 1,000 rev), если параметр P553 задан в качестве установки положения. Если для установки положения P553 используется формат INC, то для описания кривой ускорения также используется инкремент.

Если пусковое устройство устанавливается перед положением ведомого приводного механизма на расстоянии, сохраненном в **ACCELERATION**, то ведомое устройство точно синхронизируется с ведущим приводным механизмом по положению срабатывания.

ФБ включается посредством входа **ENABLE**. Запуск функции осуществляется либо через цифровой вход (P420[-xx]=64, *Start Fliegende Säge*) или с помощью сигнала **EXECUTE**. Частотный преобразователь ускоряется до скорости ведущей оси. При достижении синхронизации с ведущей осью выход **DONE** принимает значение 1.

Отключение функции редуктора производится с помощью входа **STOP** или цифровой функции P420[-xx] = 77, *Fliegende Säge anhalten*, частотный преобразователь снижает скорость до 0 Гц и останавливается. Вход **HOME** используется для передачи преобразователю команды возвращения в абсолютное положение 0. По окончании выполнения команд **HOME** или **STOP** соответствующий назначенный выход остается активным. Повторное использование **EXECUTE** или цифрового входа позволяет снова запустить функцию редуктора. Цифровая функция входа (P420[-xx] = 63, *Gleichlauf ausschalten*) останавливает функцию привода и возвращает в абсолютное положение 0.

Если выполнение функции было прервано с помощью MC\_Stop, **ABORT** принимает значение 1. В случае ошибки **ERROR** принимает значение 1, а код ошибки заносится в **ERRORID**. Все три выхода сбрасываются, когда **ENABLE** переключается на 0.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Разблокировка	BOOL	<b>VALID</b>	Достижение заданной уставки частоты	BOOL
<b>EXECUTE</b>	Начало синхронизации	BOOL	<b>DONEHOME</b>	Окончен возврат в исходное положение	
<b>STOP</b>	Конец синхронизации	BOOL	<b>DONESTOP</b>	Выполнена команда остановки	
<b>HOME</b>	Откат в положение 0	BOOL	<b>ABORT</b>	Прерванная команда	BOOL
<b>ACCELERATION</b>	Путь разгона (1rev. = 1 000)	DINT	<b>ERROR</b>	Ошибка в ФБ	BOOL
			<b>ERRORID</b>	Код ошибки	INT
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1000ч	ЧП не разблокирован				
1200ч	Контроль положения не активирован				

### 10.3.2.3 FB\_Gearing

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	

Функциональный блок **FB\_Gearing** позволяет синхронизировать положение и скорость вращения частотного преобразователя относительно положения и скорости ведущего преобразователя. Ведомое устройство, использующее данную функцию, всегда следует за движениями ведущего преобразователя.

Выполняется абсолютная синхронизация, то есть положение ведущего и ведомого устройств всегда одинаковое.

#### **Информация**

Если при переходе в режим редуктора ведомое устройство находится в другом положении, чем ведущее, то ведомое устройство переводится в положение ведущего с максимальной частотой.

Если задано передаточное соотношение, то после повторного включения также будет принято новое положение.

Значение позиции, по которому будет выполняться синхронизация, а также скорость вращения, передаются через канал Broadcast. Вход **ENABLE** активирует функцию. Контроль положения и выходной каскад должны быть активированы. Разблокировать выходной каскад можно,

например, с помощью функции MC\_Power. Если **ENABLE** принимает значение 0, частотный преобразователь снижает скорость до 0Гц и останавливается. Преобразователь возвращается в режим контроля положения. При активации MC\_Stop частотный преобразователь запускает режим редуктора, а выход **ABORT** принимает значение 1. В случае ошибки в ФБ ERROR принимает значение 1, а причина неисправности заносится в ERRORID. Значение 0 на **ENABLE** сбрасывает **ERROR, ERRORID** и **ABORT**.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Синхронизация движения активирована	BOOL	<b>VALID</b>	Функция редуктора активирована	BOOL
<b>RELATIVE</b>	Относительный режим (от V2.1)	BOOL	<b>ABORT</b>	Прерванная команда	BOOL
			<b>ERROR</b>	Ошибка в ФБ	BOOL
			<b>ERRORID</b>	Код ошибки	INT
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1000h	ЧП не разблокирован				
1200h	Контроль положения не активирован				
1201h	Не задано верхнее положение High в уставках ПЛК				
1202h	Не задано нижнее положение Low в уставках ПЛК				

### 10.3.3 Motion Control

Библиотека Motion Control Lib создана на базе спецификации PLCopen „Function blocks for motion control“. Она содержит функциональные блоки для управления и перемещения частотного преобразователя, а также обеспечивает доступ к его параметрам. Для работы блоков управления движением следует выполнить ряд настроек в параметрах устройства.

Функциональный блок	Требуемые настройки
MC_MoveVelocity	<ul style="list-style-type: none"> <li>• P350 = ПЛК активен</li> <li>• P351 = Получение главной уставки от ПЛК</li> <li>• P553 [-xx] = Уставка частоты</li> <li>• P600 = Контроль положения (режим позиционирования) выключен</li> </ul>
MC_MoveAbsolute	<ul style="list-style-type: none"> <li>• P350 = ПЛК активен</li> <li>• P351 = Получение главной уставки от ПЛК</li> <li>• P600 = Контроль положения (режим позиционирования) включен</li> <li>• Верхнее положение High должно быть задано в P553 [-xx] ( PLC_Sollwerte )</li> <li>• Нижнее положение Low должно быть задано в P553 [-xx] ( PLC_Sollwerte )</li> <li>• Уставка частоты должна быть задана в P553 [-xx] ( PLC_Sollwerte )</li> </ul>
MC_MoveRelative	
MC_MoveAdditive	
MC_Home	
MC_Power	<ul style="list-style-type: none"> <li>• P350 = ПЛК активен</li> </ul>

MC_Reset	• P351 = Получение команды управления от ПЛК
MC_Stop	

### **i** Информация

PLC\_Sollwert от 1 до 5 и команда управления ПЛК также описываются переменными процессов. При использовании ФБ управления движением Motion Control не разрешается задавать соответствующие переменные процессов в таблице переменных, так как в этом случае данные Motion Control будут перезаписываться.

### **i** Информация

#### Определение фронта сигнала

Чтобы последующие функциональные блоки могли распознавать фронт сигнала на входе нужно чтобы вызов функции был произведен на входе с различными состояниями.

Функциональный блок	Описание
MC_ReadParameter	Доступ для чтения параметров устройства
MC_WriteParameter	Доступ для записи параметров устройства
MC_MoveVelocity	Команда перемещения в режиме скорости вращения
MC_MoveAbsolute	Команда перемещения с абсолютным указанием положения
MC_MoveRelative	Команда перемещения с относительным указанием положения
MC_MoveAdditive	Команда перемещения с аддитивным указанием положения
MC_Home	Запуск возврата в исходное положение
MC_Power	Включение/выключение напряжения двигателя
MC_ReadStatus	Состояние устройства
MC_ReadActualPos	Считывание текущего положения
MC_Reset	Сброс ошибки прибора
MC_Stop	Остановка активной команды перемещения

#### 10.3.3.1 MC\_Control

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	

Данный ФБ служит для управления частотным преобразователем и позволяет использовать команды управления ЧП более детализировано, подобно MC\_Power. Управление ЧП осуществляется посредством входов **ENABLE**, **DISABLEVOLTAGE** и **QUICKSTOP**, см. нижеследующую таблицу.

Блок входов			Поведение частотного преобразователя
ENABLE	QUICKSTOP	DISABLEVOLTAGE	
High	Low	Low	Включение частотного преобразователя.
Low	Low	Low	Частотный преобразователь останавливается до 0 Гц (P103) и отключает напряжение на двигателе.
X	X	High	Подача напряжения на частотный преобразователь сразу прекращается, двигатель отключается без торможения.
X	High	Low	Частотный преобразователь останавливается до 0 Гц (P103) и отключает напряжение на двигателе.

Вход **PARASET** позволяет настраивать активный набор параметров.

Если выход **STATUS = 1**, ЧП включен, на двигатель подается напряжение.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Разблокировка	BOOL	<b>STATUS</b>	Подача тока на двигатель	BOOL
<b>DISABLEVOLTAGE</b>	Без напряжения	BOOL	<b>ERROR</b>	Ошибка в ФБ	BOOL
<b>QUICKSTOP</b>	Быстрый останов	BOOL	<b>ERRORID</b>	Код ошибки	INT
<b>PARASET</b>	Активный набор параметров Диапазон значений: 0 - 3	BYTE			
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1001ч	Активна функция остановки				
1300h	Непредвиденное состояние ЧП				

**Пример на ST:**

```
(* Разблокировка прибора с помощью Dig3*)
Control.Enable := _5_State_digital_input.2;
(* Наборы параметров определяются посредством Dig1 и Dig2. *)
Control.ParaSet := INT_TO_BYTE(_5_State_digital_input and 2#11);
Control;
(* Прибор разблокирован? *)
if Control.Status then
  (* Требуется переместить в другое положение? *)
  if SaveBit3 <> _5_State_digital_input.3 then
    SaveBit3 := _5_State_digital_input.3;
    if SaveBit3 then
      Move.Position := 500000;
    else
      Move.Position := 0;
    end_if;

    Move(Execute := False);
  end_if;
end_if;

(* Перейти в положение когда прибор разблокирован. *)
Move(Execute := Control.Status);
```

**10.3.3.2 MC\_Control\_MS**

	SK 54xE	SK 53xE SK 52xE	SK 2xE	SK 2xE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>						X

Данный ФБ используется для управления пусковым устройством (ПУ).

Блок входов				Поведение частотного преобразователя
ENABLE_RIG HT	ENABLE_LEFT	QUICKSTOP	DISABLEVOLTAGE	
High	Low	Low	Low	Включение ПУ, вращение вправо
Low	High	Low	Low	Включение ПУ, вращение влево
High	High	Low	Low	Выключение ПУ
Low	Low	Low	Low	ПУ останавливается до 0 Гц (P103) и отключает напряжение на двигателе.
X	X	X	High	Подача напряжения на ПУ сразу прекращается, двигатель отключается без торможения.
X	X	High	Low	ПУ выполняет быстрый останов

				(P426)и отключает напряжение на двигателе.
--	--	--	--	--

(X = уровень на входе не важен)

Если выход **STATUS** = 1, ПУ включен, на двигатель подается напряжение.

Если **OPENBRAKE** принимает значение 1, тормоз открывается.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE_RIGHT</b>	Вправо пуск право	BOOL	<b>STATUS</b>	Подача тока на двигатель	BOOL
<b>ENABLE_LEFT</b>	Влево пуск влево	BOOL	<b>ERROR</b>	Ошибка в ФБ	BOOL
<b>DISABLEVOLTA GE</b>	Без напряжения	BOOL	<b>ERRORID</b>	Код ошибки	INT
<b>QUICKSTOP</b>	Быстрый останов	BOOL			
<b>OPENBRAKE</b>	Открыть тормоз	BOOL			
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1001ч	Активна функция остановки				
1300h	Непредвиденное состояние ПУ				

### 10.3.3.3 MC\_Home

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	

Дает команду частотному преобразователю начать перемещение в заданную точку, как только **EXECUTE** меняет значение с 0 на 1 (фронт). Перемещение ЧП выполняется согласно уставке частоты, заданной **VELOCITY**. При активации входа с сигналом контрольного положения (P420[-xx] = заданная точка), направление вращения меняется на обратное. При отрицательном фронте сигнала контрольного положения принимается значение, хранящееся в **POSITION**. После этого частотный преобразователь останавливается до 0Гц, сигнал **DONE** принимает значение 1. Во время выполнения перемещения в исходное положение **HOME** выход **BUSY** остается активным.

При необходимости прерывания данной процедуры (например, другим функциональным блоком MC), используется сигнал **COMMANDABORTED**.

В случае ошибки **ERROR** принимает значение 1. В этом случае значение **DONE** равно 0. **ERRORID** содержит соответствующий код ошибки.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>EXECUTE</b>	Разблокировка	BOOL	<b>DONE</b>	Достижение заданной уставки положения	BOOL
<b>POSITION</b>	Уставка положения	DINT	<b>COMMAND-ABORTED</b>	Прерванная команда	BOOL
<b>VELOCITY</b>	Уставка частоты	INT	<b>ERROR</b>	Ошибка в ФБ	BOOL
			<b>ERRORID</b>	Код ошибки	INT
			<b>BUSY</b>	Активирован возврат в исходное положение	BOOL
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1000h	ЧП не разблокирован				
1200h	Контроль положения не активирован				
1201h	Не задано верхнее положение High в уставках ПЛК (P553)				
1202h	Не задано нижнее положение Low в уставках ПЛК (P553)				
1D00h	Абсолютный энкодер не поддерживается				

#### 10.3.3.4 MC\_MoveAbsolute

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	

Записывает уставку положения и скорости для частотного преобразователя, если **EXECUTE** меняется с 0 на 1 (фронт). Уставка частоты **VELOCITY** передается после пересчета, описываемого в MC\_MoveVelocity.

#### **POSITION:**

**MODE** = False:

Уставка положения определяется по значению, переданному в **POSITION**.

**MODE** = True:

Значение, переданное в **POSITION**, соответствует индексу из параметра P613 увеличенному на 1. Положение, заданное в этом индексе параметра, соответствует уставке положения.

Пример:

Mode = True; Position = 12

ФБ переводит в положение, указанное в текущем наборе параметров P613[-13].

При достижении преобразователем уставки положения, **DONE** принимает значение 1. **DONE** удаляется при сбросе **EXECUTE**. Когда **EXECUTE** удаляется перед достижением требуемого



положения, **DONE** для цикла принимает значение 1. Сигнал **BUSY** остается активным во время перемещения к уставке положения. При необходимости прерывания данной процедуры (например, другим функциональным блоком MC), используется сигнал **COMMANDABORTED**. В случае ошибки **ERROR** принимает значение 1, а код ошибки заносится в **ERRORID**. В этом случае значение **DONE** равно 0. Отрицательный фронт **EXECUTE** возвращает 0 на все выходы.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>EXECUTE</b>	Разблокировка	BOOL	<b>DONE</b>	Достижение заданной уставки положения	BOOL
<b>POSITION</b>	Уставка положения	DINT	<b>BUSY</b>	Уставка частоты не достигнута	BOOL
<b>VELOCITY</b>	Уставка частоты	INT	<b>COMMAND-ABORTED</b>	Прерванная команда	BOOL
<b>MODE</b>	Режим источника уставки положения	BOOL	<b>ERROR</b>	Ошибка в ФБ	BOOL
			<b>ERRORID</b>	Код ошибки	INT
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
0x1000	ЧП не разблокирован				
0x1200	Контроль положения не активирован				
0x1201	Не задано верхнее положение High в уставках ПЛК (P553)				
0x1202	Не задано нижнее положение Low в уставках ПЛК (P553)				

#### Пример на ST:

```
(* Устройство разблокируется если DIG1 = TRUE *)
Power(Enable := _5_State_digital_input.0);
IF Power.Status THEN
  (* Устройство разблокировано и перемещается в положение 20000 с 50% макс. частоты.
  Датчик и контроль положения для двигателя должны быть активными для выполнения данного
  действия. *)
  MoveAbs(Execute := _5_State_digital_input.1, Velocity := 16#2000, Position := 20000);
END_IF
```

#### 10.3.3.5 MC\_MoveAdditive

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	

Соответствует MC\_MoveAbsolute по всем пунктам до входа **DISTANCE**. Уставка положения определяется путем добавления текущей уставки положения к значению, переданному **DISTANCE**.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>EXECUTE</b>	Разблокировка	BOOL	<b>DONE</b>	Достижение заданной уставки положения	BOOL
<b>DISTANCE</b>	Уставка положения	DINT	<b>COMMAND-ABORTED</b>	Прерванная команда	BOOL
<b>VELOCITY</b>	Уставка частоты	INT	<b>ERROR</b>	Ошибка в ФБ	BOOL
<b>MODE</b>	Режим источника уставки положения	BOOL	<b>ERRORID</b>	Код ошибки	INT
			<b>BUSY</b>	Уставка положения не достигнута	BOOL
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1000h	ЧП не разблокирован				
1200h	Контроль положения не активирован				
1201h	Не задано верхнее положение High в уставках ПЛК (P553)				
1202h	Не задано нижнее положение Low в уставках ПЛК (P553)				

### 10.3.3.6 MC\_MoveRelative

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	

Соответствует MC\_MoveAbsolute по всем пунктам до входа **DISTANCE**. Уставка положения определяется путем добавления текущего фактического положения к значению, переданному **DISTANCE**.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>EXECUTE</b>	Разблокировка	BOOL	<b>DONE</b>	Достижение заданной уставки положения	BOOL
<b>DISTANCE</b>	Уставка положения	DINT	<b>COMMAND-ABORTED</b>	Прерванная команда	BOOL
<b>VELOCITY</b>	Уставка частоты	INT	<b>ERROR</b>	Ошибка в ФБ	BOOL
<b>MODE</b>	Режим источника уставки положения	BOOL	<b>ERRORID</b>	Код ошибки	INT
			<b>BUSY</b>	Уставка положения не достигнута	BOOL

ERRORID	Описание
0	Отсутствие ошибки
1000h	ЧП не разблокирован
1200h	Контроль положения не активирован
1201h	Не задано верхнее положение High в уставках ПЛК (P553)
1202h	Не задано нижнее положение Low в уставках ПЛК (P553)

### 10.3.3.7 MC\_MoveVelocity

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	

Применяет уставку частоты для частотного преобразователя, если **EXECUTE** меняется с 0 на 1 (фронт). При достижении преобразователем уставки частоты, **INVELOCITY** принимает значение 1. Выход **BUSY** остается активным, пока ЧП ускоряется до уставки частоты. Если **EXECUTE** уже принял значение 0, то **INVELOCITY** принимает значение 1 только на один цикл. При необходимости прерывания данной процедуры (например, другим функциональным блоком MC), используется сигнал **COMMANDABORTED**.

Отрицательный фронт **EXECUTE** возвращает 0 на все выходы.

**VELOCITY** пересчитывается по следующей формуле:

$$\mathbf{VELOCITY} = (\text{уставка частоты (Гц)} \times 0x4000) / P105$$

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>EXECUTE</b>	Разблокировка	BOOL	<b>INVELOCITY</b>	Достижение заданной уставки частоты	BOOL
<b>VELOCITY</b>	Уставка частоты	INT	<b>BUSY</b>	Уставка частоты еще не достигнута	BOOL
			<b>COMMAND-ABORTED</b>	Прерванная команда	BOOL
			<b>ERROR</b>	Ошибка в ФБ	BOOL
			<b>ERRORID</b>	Код ошибки	INT
ERRORID	Описание				
0	Отсутствие ошибки				
1000ч	ЧП не разблокирован				
1100h	ЧП не находится в режиме частоты вращения (активирован контроль положения)				
1101h	Не настроен параметр уставки частоты (P553)				

### Пример на AWL:

```

CAL Power
CAL Move

LD TRUE
ST Power.Enable

(*Установить 20 Гц (макс. 50 Гц) *)
LD DINT#20
MUL 16#4000
DIV 50

DINT_TO_INT
ST Move.Velocity

LD Power.Status
ST Move.Execute

```

### Пример на ST:

```

(* Прибор готов к работе если установлен DIG1 *)
Power(Enable := _5_State_digital_input.0);
IF Power.Status THEN
  (* Прибор разблокируется с 50% от макс. частоты, если установлен DIG2 *)
  MoveVelocity(Execute := _5_State_digital_input.1, Velocity := 16#2000);
END_IF

```

#### 10.3.3.8 MC\_Power

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

Данная функция предназначена для включения и выключения выходного каскада. Если вход **ENABLE** принимает значение 1, происходит разблокировка выходного каскада. Обязательным условием для этого является нахождение прибора в состоянии „Switch-on Block“ или „Ready for Switch-on“. Если прибор находится в состоянии „Fault“ или „Fault reaction active“, то сначала необходимо устранить неисправность и обработать соответствующее сообщение. Только после этого может быть выполнена разблокировка с помощью данного блока. Если прибор находится в состоянии „Not Ready for Switch-on“, то включение также не выполняется. Во всех вышеперечисленных случаях ФБ переходит в состояние ошибки, поэтому **ENABLE** следует сначала присвоить 0, чтобы обработать ошибку.

Если вход **ENABLE** принимает значение 0, происходит выключение прибора. Если двигатель при этом работает, то посредством линейного изменения, установленного с помощью параметра P103, двигатель сначала переводится на 0 Гц.

При включенном выходном каскаде прибора выходу **STATUS** соответствует значение 1, при выключенном - 0.

Значения **ERROR** и **ERRORID** сбрасываются, когда **ENABLE** переключается на 0.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Разблокировка	BOOL	<b>STATUS</b>	Подача тока на двигатель	BOOL
			<b>ERROR</b>	Ошибка в ФБ	BOOL
			<b>ERRORID</b>	Код ошибки	INT
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1001ч	Активна функция остановки				
1300h	Устройство не приведено в состояние „Ready for Switch-on“ или „Switch-on Block“				

#### Пример на AWL:

```

CAL Power
CAL Move

LD TRUE
ST Power.Enable

(*Установить 20 Гц (макс. 50 Гц) *)
LD DINT#20
MUL 16#4000
DIV 50

DINT_TO_INT
ST Move.Velocity

LD Power.Status
ST Move.Execute
    
```

#### Пример на ST:

```

(* Активировать блок питания Power Block *)
Power(Enable := TRUE);
IF Power.Status THEN
    (* Устройство готово к включению *)
END_IF
    
```

#### 10.3.3.9 MC\_ReadActualPos

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	

Непрерывно сообщает текущее фактическое положение частотного преобразователя, если **ENABLE** соответствует значению 1. При получении действующего фактического положения на выходе, **VALID** переводится в действующее значение. В случае ошибки **ERROR** принимает значение 1, а **VALID** значение 0.

Пересчет положения: 1 оборот вала двигателя = 1000

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Разблокировка	BOOL	<b>VALID</b>	Действительный выход	BOOL
			<b>ERROR</b>	Ошибка в ФБ	BOOL
			<b>POSITION</b>	Текущее факт.положение ЧП	DINT

**Пример на ST:**

```

ReadActualPos(Enable := TRUE);
IF ReadActualPos.Valid THEN
    Pos := ReadActualPos.Position;
END_IF

```

**10.3.3.10 MC\_ReadParameter**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

Выполняет циклическое считывание параметра с устройства, пока **ENABLE** равен 1. Считанный параметр заносится в Value и действует, если **DONE** равно 1. Во время процедуры чтения выход **BUSY** имеет значение 1. Пока **ENABLE** равен 1, параметр считывается циклически непрерывно. Номер параметра и индекс могут быть изменены в любое время при активном **ENABLE**. Однако при этом трудно определить, когда считано новое значение, так как сигнал **DONE** все это время равен 1. В этом случае рекомендуется установить для сигнала **ENABLE** значение 0, так как сигнал **DONE** в этом случае сбрасывается. Индекс параметра определяется по индекс в документации минус 1. Например, P700 индекс 3 („Reason for switch-on block“) соответствует индексу параметра 2. В случае ошибки **ERROR** принимает значение 1. **DONE** в этом случае равен 0, а **ERRORID** содержит код ошибки. Если сигнал **ENABLE** принимает значение 0, то все сигналы и **ERRORID** удаляются.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Разблокировка	BOOL	<b>DONE</b>	Действительное значение Value	BOOL
<b>PARAMETERNUMBER</b>	Номер параметра	INT	<b>ERROR</b>	Ошибка процедуры чтения	BOOL
<b>PARAMETERINDEX</b>	Индекс параметра	INT	<b>BUSY</b>	Процедура не завершена	BOOL
			<b>ERRORID</b>	Код ошибки	INT
			<b>VALUE</b>	Считанный параметр	DINT
<b>ERRORID</b>	<b>Описание</b>				
0	недопустимый номер параметра				
3	ошибка индекса параметра				
4	нет массива				

201	Недействительный элемент в последнем полученном задании
202	Невозможно отобразить внутренний код отклика

**Пример на ST:**

```
(* Блок движения FB_ReadParameter *)
ReadParam(Enable := TRUE, ParameterNumber := 102, ParameterIndex := 0);
IF ReadParam.Done THEN
  Value := ReadParam.Value;
  ReadParam(Enable := FALSE);
END_IF
```

**10.3.3.11 MC\_ReadStatus**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

Считывает состояние устройства. Определение состояния осуществляется на основании спецификации PLCopen „Function blocks for motion control“. Если сигналу **ENABLE** соответствует 1, происходит считывание состояния.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Разблокировка	BOOL	<b>VALID</b>	Действительный выход	BOOL
			<b>ERROR</b>	Ошибка в ФБ	BOOL
			<b>ERRORSTOP</b>	Ошибка устройства	BOOL
			<b>DISABLED</b>	Выходной каскад устройства выключен.	BOOL
			<b>STOPPING</b>	Активирована команда остановки.	BOOL
			<b>DISCRETEMOTION</b>	Активирован один из трех ФБ позиционирования.	BOOL
			<b>CONTINUOUSMOTION</b>	Активировано MC_Velocity	BOOL
			<b>HOMING</b>	Активировано MC_Home	BOOL
			<b>STANDSTILL</b>	Активные команды перемещения на устройстве отсутствуют Установлено число оборотов 0 об/мин, выходной каскад включен.	BOOL

**Пример на ST:**

```
ReadStatus(Enable := TRUE);
IF ReadStatus.Valid THEN
  fError := ReadStatus.ErrorStop;
  fDisable := ReadStatus.Disabled;
  fStopping := ReadStatus.Stopping;
```

```
fInMotion := ReadStatus.DiscreteMotion;
fInVelocity := ReadStatus.ContinuousMotion;
fInHome := ReadStatus.Homing;
fStandStill := ReadStatus.StandStill;
end_if
```

### 10.3.3.12 MC\_Reset

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

Сброс ошибки на устройстве (обработка неисправности), при возрастающем фронте сигнала **EXECUTE**. В случае ошибки **ERROR** принимает значение 1, а причина неисправности заносится в **ERRORID**. При отрицательном фронте **EXECUTE** выполняется сброс всех ошибок.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>EXECUTE</b>	Запуск	BOOL	<b>DONE</b>	Сброс ошибок устройства	BOOL
			<b>ERROR</b>	Ошибка в ФБ	BOOL
			<b>ERRORID</b>	Код ошибки	INT
			<b>BUSY</b>	Процесс сброса еще активен	BOOL
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1001ч	Активна функция остановки				
1700h	Сброс ошибки не может быть выполнен, причина не устранена.				

#### Пример на ST:

```
Reset(Execute := TRUE);
IF Reset.Done THEN
  (* Сброс ошибки выполнен *)
  Reset(Execute := FALSE);
ELSIF Reset.Error THEN
  (* Сброс ошибки не может быть выполнен, причина не устранена *)
  Reset(Execute := FALSE);
END_IF
```

### 10.3.3.13 MC\_Stop

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

При повышении значения фронта (с 0 на 1) устройство переводится в состояние **STANDINGSTILL**. Все активные функции движения прерываются. Выполняется торможение на 0 Гц, выходной каскад отключается. Выполнение всех остальных ФБ перемещения блокируется, пока команда остановки активна (**EXECUTE** = 1). Выход **BUSY** активируется при растущем фронте на **EXECUTE** и остается активным до получения падающего фронта на **EXECUTE**.



VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
EXECUTE	Запуск	BOOL	DONE	Команда выполнена	BOOL
			BUSY	Команда активирована	BOOL

### 10.3.3.14 MC\_WriteParameter\_16 / MC\_WriteParameter\_32

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X

Записывает параметр 16/32 бита на устройство, когда **EXECUTE** меняется с 0 на 1 (фронт). Когда **DONE** принимает значение 1 - параметр записан. Во время процедуры чтения выход **BUSY** имеет значение 1. В случае ошибки **ERROR** принимает значение 1, а код ошибки заносится в **ERRORID**. Сигналы **DONE**, **ERROR**, **ERRORID** сохраняют свои значения до тех пор, пока **EXECUTE** снова не изменит значение на 0. При изменении значения **EXECUTE** на 0 процесс записи не прерывается. Только сигнал **DONE** сохраняет значение на 1 цикл ПЛК.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
EXECUTE	Разблокировка	BOOL	DONE	Действительное значение Value	BOOL
PARAMETERNUMBER	Номер параметра	INT	BUSY	Активирована процедура записи	BOOL
PARAMETERINDEX	Индекс параметра	INT	ERROR	Ошибка процедуры чтения	BOOL
VALUE	Записываемое значение	INT	ERRORID	Код ошибки	INT
RAMONLY	Хранить значение в RAM (версия V2.1)	BOOL			
<b>ERRORID</b>	<b>Описание</b>				
0	недопустимый номер параметра				
1	Изменение значения параметра невозможно				
2	выход за пределы нижней или верхней границы диапазона значений				
3	ошибка индекса параметра				
4	нет массива				
5	недопустимый тип данных				
6	только сброс (разрешена запись только значения 0)				
7	Изменение элемента описания невозможно				

201	Недействительный элемент в последнем полученном задании
202	Невозможно отобразить внутренний код отклика

**Пример на ST:**

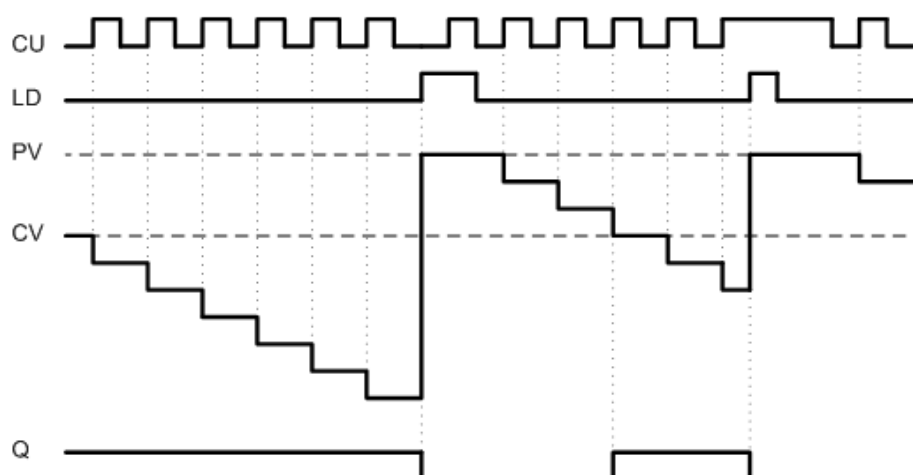
```
WriteParam16(Execute := TRUE, ParameterNumber := 102, ParameterIndex := 0, Value := 300);
IF WriteParam16.Done THEN
  WriteParam16(Execute := FALSE);
END_IF;
```

**10.3.4 Standard**

**10.3.4.1 Вычитающий счетчик CTD**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

При растущем фронте на **CD** счетчик функционального блока **CV** уменьшается на единицу, пока значение CV больше -32768. Если **CV** меньше или равно 0, выход **Q** сохраняет значение TRUE. С помощью **LD** счетчику **CV** может быть присвоено значение, хранящееся в **PV**.



VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>CD</b>	Вход счетчика	<b>BOOL</b>	<b>Q</b>	TRUE, если CV ≤ 0	<b>BOOL</b>
<b>LD</b>	Загрузка начального значения	<b>BOOL</b>	<b>CV</b>	Текущее состояние счетчика	<b>INT</b>
<b>PV</b>	Начальное значение	<b>INT</b>			

**Пример на AWL:**

```
LD VarBOOL1
ST CTDInst.CD
LD VarBOOL2
ST CTDInst.LD
LD VarINT1
ST CTDInst.PV
CAL CTDInst
LD CTDInst.Q
ST VarBOOL3
LD CTDInst.CV
ST VarINT2
```

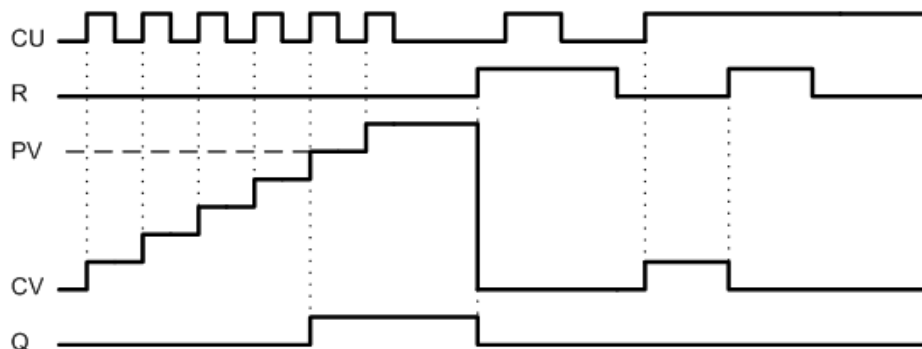
**Пример на ST:**

```
CTDInst(CD := VarBOOL1, LD := VarBOOL2, PV := VarINT1);
VarBOOL3 := CTDInst.Q;
VarINT2 := CTDInst.CV;
```

**10.3.4.2 Суммирующий счетчик CTU**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

При растущем фронте на **CU** счетчик функционального блока **CV** увеличивается на единицу, пока **CV** не достигнет значения 32767. Если **CV** больше или равно **PV**, выход **Q** сохраняет значение TRUE. С помощью **R** может быть выполнен сброс счетчика **CV** на ноль.



VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>CU</b>	Вход счетчика	<b>BOOL</b>	<b>Q</b>	TRUE, если CV >= 0	<b>BOOL</b>
<b>R</b>	Сброс счетчика	<b>BOOL</b>	<b>CV</b>	Текущее состояние счетчика	<b>INT</b>
<b>PV</b>	Начальное значение	<b>INT</b>			

**Пример на AWL:**

```
LD VarBOOL1
ST CTUInst.CU
LD VarBOOL2
ST CTUInst.R
LD VarINT1
ST CTUInst.PV
CAL CTUInst(CU := VarBOOL1, R := VarBOOL2, PV := VarINT1)
LD CTUInst.Q
ST VarBOOL3
LD CTUInst.CV
ST VarINT2
```

**Пример на ST:**

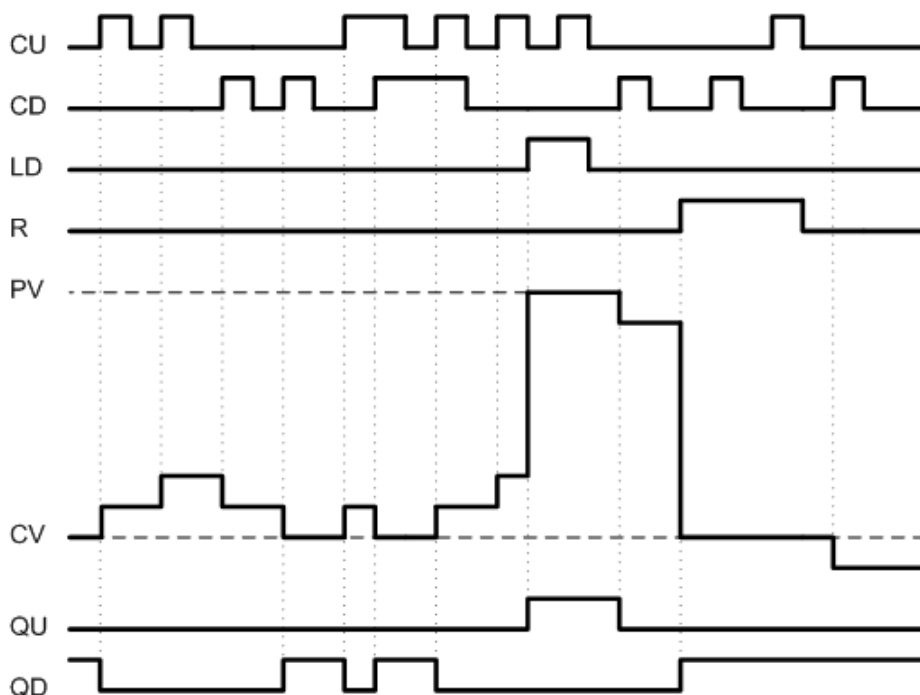
```
CTUInst(CU := VarBOOL1, R := VarBOOL2, PV := VarINT1);
VarBOOL3 := CTUInst.Q;
VarINT2 := CTUInst.CV;
```

**10.3.4.3 Суммирующий и вычитающий счетчик CTUD**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

При растущем фронте на **CU** счетчик **CV** увеличивается на единицу, пока значение **CV** меньше 32767. При растущем фронте на **CD** счетчик **CV** уменьшается на единицу, пока значение **CV** больше -32768. С помощью **R** может быть выполнен сброс счетчика **CV** на ноль. С помощью **LD** хранящееся в **PV** значение копируется в **CV**.

**R** имеет преимущество перед **LD**, **CU** и **CV**. **PV** можно изменить в любое время, **QU** всегда относится к текущему установленному значению.



VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>CU</b>	Суммирующий счетчик	BOOL	<b>QU</b>	TRUE, если CV >= PV	BOOL
<b>CD</b>	Вычитающий счетчик	BOOL	<b>QD</b>	TRUE, если CV <= 0	BOOL
<b>R</b>	Сброс счетчика	BOOL	<b>CV</b>	Текущее состояние счетчика	INT
<b>LD</b>	Загрузка начального значения	BOOL			
<b>PV</b>	Начальное значение	INT			

**Пример на AWL:**

```
LD VarBOOL1
ST CTUDInst.CU
LD VarBOOL3
ST CTUDInst.R
LD VarBool4
ST CTUDInst.LD
LD VarINT1
ST CTUInst.PV
CAL CTUDInst
LD CTUDInst.QU
ST VarBOOL5
LD CTUDInst.QD
ST VarBOOL5
LD CTUInst.CV
ST VarINT2
```

**Пример на ST:**

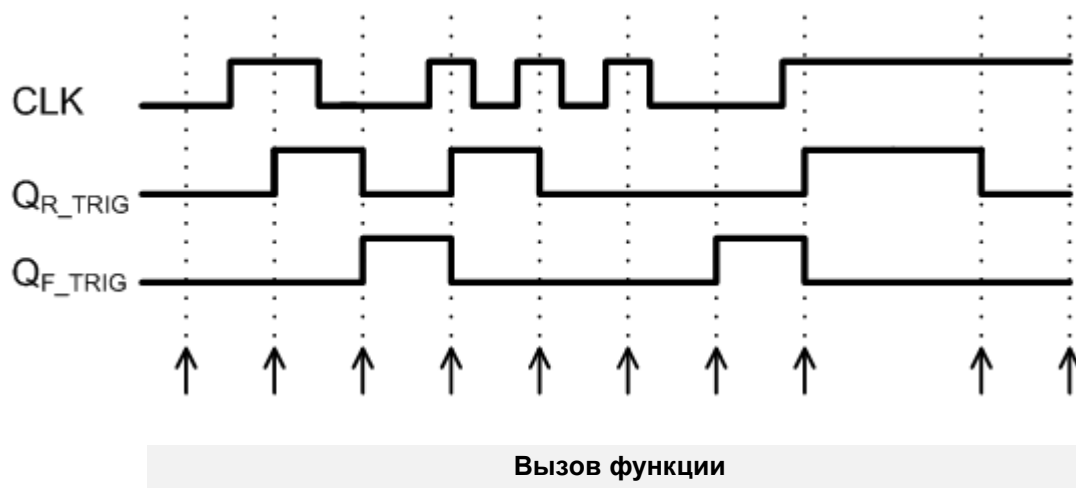
```
CTUDInst(CU:=VarBOOL1, R:=VarBOOL3, LD:=VarBOOL4, PV:=VarINT1);
VarBOOL5 := CTUDInst.QU;
VarBOOL5 := CTUDInst.QD;
VarINT2 := CTUDInst.CV;
```

**10.3.4.4 R\_TRIG и F\_TRIG**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

Обе функции служат для распознавания фронта. При обнаружении фронта на **CLK**, **Q** принимает значение TRUE до следующего вызова функции, а после этого снова значение FALSE. Только после получения нового фронта **Q** может принять значение TRUE на цикл.

- R\_TRIG = растущий фронт
- F\_TRIG = падающий фронт



VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
CLK	Установка	BOOL	Q	Выход	BOOL

**Пример на AWL:**

```
LD VarBOOL1
ST RTRIGInst.CLK
CAL RTRIGInst
LD RTRIGInst.Q
ST VarBOOL2
```

**Пример на ST:**

```
RTRIGInst(CLK:= VarBOOL1);
VarBOOL2 := RTRIGInst.Q;
```

**Информация**

Выходной сигнал функции изменяется только при вызове функции. По этой причине целесообразно непрерывно вызывать определение фронта с помощью цикла ПЛК.

**10.3.4.5 RS Flip Flop**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

Бистабильная функция, **S** устанавливает значение выхода **Q1**, а **R1** снова сбрасывает его. Если **R1** и **S** одновременно принимают значение TRUE, то приоритет имеет **R1**.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
S	Установка	BOOL	Q1	Выход	BOOL
R1	Сброс	BOOL			

**Пример на AWL:**

```
LD VarBOOL1
ST RSInst.S
LD VarBOOL2
ST RSInst.R1
CAL RSInst
LD RSInst.Q1
ST VarBOOL3
```

**Пример на ST:**

```
RSInst(S:= VarBOOL1, R1:=VarBOOL2);
VarBOOL3 := RSInst.Q1;
```

**10.3.4.6 SR Flip Flop**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X

Бистабильная функция, **S1** устанавливает значение выхода **Q1**, а **R** снова сбрасывает его. Если **R1** и **S** одновременно принимают значение TRUE, то приоритет имеет **S1**.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
S1	Установка	BOOL	Q1	Выход	BOOL
R	Сброс	BOOL			

**Пример на AWL:**

```
LD VarBOOL1
ST SRInst.S1
LD VarBOOL2
ST SRInst.R
CAL SRInst
LD SRInst.Q1
ST VarBOOL3
```

**Пример на ST:**

```
SRInst(S1:= VarBOOL1, R:=VarBOOL2);
VarBOOL3 := SRInst.Q1;
```

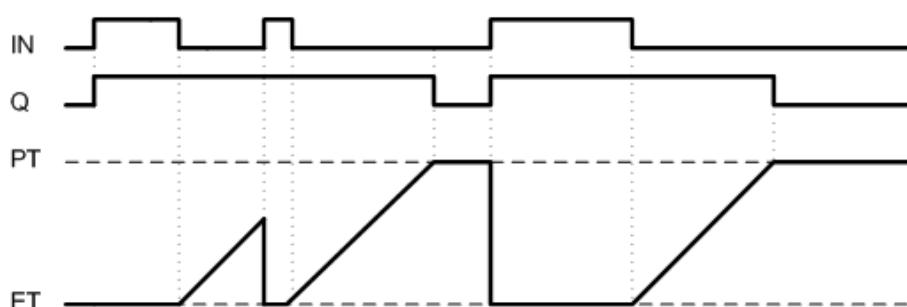
### 10.3.4.7 Задержка выключения TOF

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X

Если **IN** = TRUE, то **Q** принимает значение TRUE. Если **IN** = FALSE, таймер увеличивается. Пока таймер работает (**ET** < **PT**), **Q** имеет значение TRUE. Если (**ET** = **PT**) - таймер останавливается, **Q** принимает значение FALSE. При новом растущем фронте в **IN**, таймер **ET** снова обнуляется.

Для упрощения ввода здесь могут использоваться литералы, например:

- LD TIME#50s20ms = 50,020 секунд
- LD TIME#1d30m = 1 день и 30 минут



VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
IN	Таймер активирован	BOOL	Q	TRUE & ( ET < PT )	BOOL
PT	Количество времени	DINT	ET	Текущее состояние таймера	DINT

#### Пример на AWL:

```
LD VarBOOL1
ST TOFInst.IN
LD DINT#5000
ST TOFInst.PT
CAL TOFInst
LD TOFInst.Q
ST VarBOOL2
```

#### Пример на ST:

```
TOFInst(IN := VarBOOL1, PT:= T#5s);
VarBOOL2 := TOFInst.Q;
```

## Информация

## Таймер ET

Время ET идет независимо от цикла ПЛК. Запуск таймера с IN и назначение выхода Q производятся только при вызове функции „CAL“. Вызов функции осуществляется в цикле ПЛК, но в длинных программах для ПЛК он может быть больше 5 мс, что может привести к возникновению фазовой флуктуации.

### 10.3.4.8 Задержка включения TON

	SK 54xE	SK 53xE	SK 2xxE	SK 2xxE-FDS	SK 180E	SK 155E-FDS

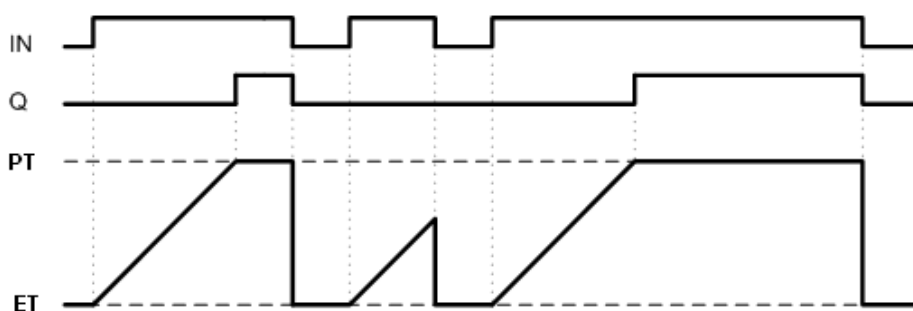


		<b>SK 52xE</b>			<b>SK 190E</b>	<b>SK 175E-FDS</b>
<b>Наличие</b>	X	X	X	X	X	X

Если **IN** = TRUE, таймер увеличивается. Если **ET** = **PT**, **Q** принимает значение TRUE, а таймер останавливается. **Q** сохраняет значение TRUE пока **IN** также имеет значение TRUE. При новом растущем фронте на **IN**, таймер запускается снова с нуля. **PT** может изменяться во время работы таймера. Продолжительность указывается в **PT** в миллисекундах. Допускается задержка по времени в диапазоне от 5мс до 24,8 дней. Минимальная задержка по времени составляет 5 мс, так как опорное время ПЛК также составляет 5 мс.

Для упрощения ввода здесь могут использоваться литералы, например:

- LD TIME#50s20ms = 50,020 секунд
- LD TIME#1d30m = 1 день и 30 минут



VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>IN</b>	Таймер активирован	BOOL	<b>Q</b>	TRUE & ( IN=TRUE & ET=PT )	BOOL
<b>PT</b>	Количество времени	DINT	<b>ET</b>	Текущее состояние таймера	DINT

#### Пример на AWL:

```
LD VarBOOL1
ST TONInst.IN
LD DINT#5000
ST TONInst.PT
CAL TONInst
LD TONInst.Q
ST VarBOOL2
```

#### Пример на ST:

```
TONInst(IN := VarBOOL1, PT:= T#5s);
VarBOOL2 := TONInst.Q;
```

### **i** Информация

### Таймер ET

Время **ET** идет независимо от цикла ПЛК. Запуск таймера с **IN** и назначение выхода **Q** производятся только при вызове функции „CAL“. Вызов функции осуществляется в цикле ПЛК, но в длинных программах для ПЛК он может быть больше 5 мс, что может привести к возникновению фазовой флуктуации.

#### 10.3.4.9 Синхроимпульс TP

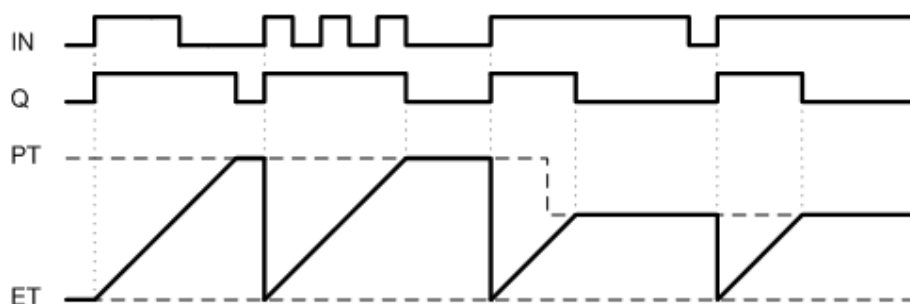
	<b>SK 54xE</b>	<b>SK 53xE</b> <b>SK 52xE</b>	<b>SK 2xxE</b>	<b>SK 2xxE-FDS</b>	<b>SK 180E</b> <b>SK 190E</b>	<b>SK 155E-FDS</b> <b>SK 175E-FDS</b>
--	----------------	----------------------------------	----------------	--------------------	----------------------------------	--

Наличие	X	X	X	X	X	X
---------	---	---	---	---	---	---

При положительном фронте на **IN** таймер запускается со значения 0. Таймер отсчитывает время до достижения значения, указанного в **PT** и останавливается. Эта процедура не может быть прервана. Во время приращения **PT** может изменяться. Выход **Q** равен TRUE, пока таймер **ET** меньше **PT**. Если **ET = PT** и на **IN** зафиксирован растущий фронт, таймер снова запускается с нуля.

Для упрощения ввода здесь могут использоваться литералы, например:

- LD TIME#50s20ms = 50,020 секунд
- LD TIME#1d30m = 1 день и 30 минут



VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
IN	Таймер активирован	BOOL	Q	TRUE & ( ET < PT )	BOOL
PT	Количество времени	DINT	ET	Текущее состояние таймера	DINT

#### Пример на AWL:

```
LD VarBOOL1
ST TPInst.IN
LD DINT#5000
ST TPInst.PT
CAL TPInst
LD TPInst.Q
ST VarBOOL2
```

#### Пример на ST:

```
TPInst (IN := VarBOOL1, PT:= T#5s);
VarBOOL2 := TPInst.Q;
```

### Информация

### Таймер ET

Время **ET** идет независимо от цикла ПЛК. Запуск таймера с **IN** и назначение выхода **Q** производятся только при вызове функции „CAL“. Вызов функции осуществляется в цикле ПЛК, но в длинных программах для ПЛК он может быть больше 5 мс, что может привести к возникновению фазовой флуктуации.

#### 10.3.5 Доступ к областям памяти частотного преобразователя

При необходимости промежуточного хранения большого объема данных, их передачи или получения с других устройств следует использовать функциональные блоки **FB\_WriteTrace** und **FB\_ReadTrace**.



VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Выполнение	BOOL	<b>VALID</b>	Процедура чтения выполнена успешно	BOOL
<b>SIZE</b>	Формат сохранения	BOOL	<b>READY</b>	Считан весь объем памяти	BOOL
<b>MEMORY</b>	Выбор области памяти	BYTE	<b>ERROR</b>	Ошибка ФБ	BOOL
<b>STARTINDEX</b>	Указывает на описываемую ячейку памяти	INT	<b>ERRORID</b>	Код ошибки	INT
			<b>ACTINDEX</b>	Текущий индекс памяти, из которого будет выполнено чтение в следующем цикле	INT
			<b>VALUE</b>	Считанное значение	DINT
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1A00h	Превышен диапазон значений STARTINDEX				
1A01h	Превышен диапазон значений MEMORY				

### 10.3.5.2 FB\_WriteTrace

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X		


С помощью данного ФБ осуществляется промежуточное хранение отдельных значений или большого объема данных на частотном преобразователе. Хранение не является длительным, то есть при перезапуске ЧП все данные теряются.

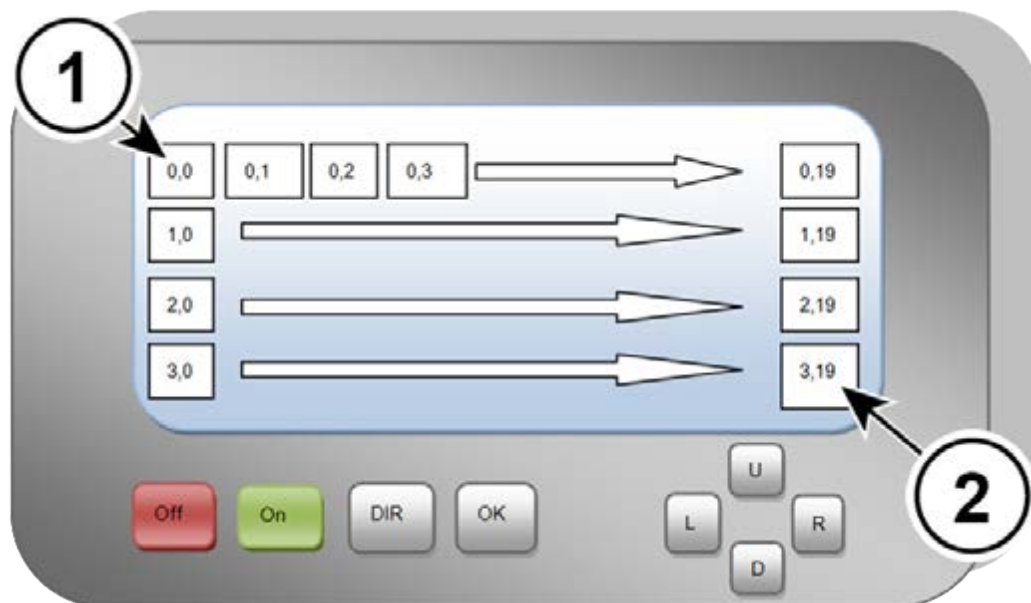
Когда ФБ определяет положительный фронт на входе **ENABLE**, происходит передача всех параметров, связанных с данным входом. Значение, сохраненное в **VALUE**, записывается в ячейки памяти, отмеченные с помощью **STARTINDEX** и **MEMORY**. В случае успешного выполнения процедуры записи выход **VALID** принимает значение 1.

Если ФБ вызывается многократно, а на входе **ENABLE** сохраняется значение 1, то при каждом вызове ФБ происходит считывание и сохранение входа **VALUE**, а адрес памяти увеличивается на 1. Текущий индекс памяти для следующего доступа может быть считан на выходе **ACTINDEX**. При достижении конца памяти выход **FULL** принимает значение 1, а процедура сохранения завершается. Однако если вход **OVERWRITE** равен 1, то индекс памяти снова будет равен **STARTINDEX**, а ранее сохраненные данные будут перезаписаны.

Для сохранения доступны форматы данных INT или DINT. Для значений в формате INT на входе **VALUE** обрабатывается только нижняя часть Low. Для назначения используется вход **SIZE**: 0 соответствует формату INT, а 1 - формату DINT.



конфигурирования ПЛК (кнопка ) , пункт “Allow ParameterBox function modules“. Параметр „Parameterbox\_key\_state“ позволяет дополнительно запрашивать состояние клавиатуры модуля. За счет этого может выполняться ввод в программу ПЛК. Нижеследующее изображение описывает вид дисплея и положение задействованных кнопок для модуля ParameterBox.



1	Первый символ	(0,0 → строка = 0, поле = 0)
2	Последний символ	(3,19 → строка = 3, поле = 19)

### 10.3.6.1 Обзор визуализации

Функциональный блок	Описание
FB_STRINGToPBox	Копирует строку в P-Box
FB_DINTToPBox	Копирует значение DINT в P-Box

### 10.3.6.2 FB\_DINTToPBOX

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X

Данный функциональный блок выполняет конвертацию значения в формате DINT в строку кодировки ASCII и копирует его в ParameterBox. Вывод может осуществляться в десятичном, двоичном или шестнадцатеричном формате, выбор формата осуществляется при помощи **MODE**. **ROW** и **COLUMN** используются для установки начального положения строки на дисплее модуля P-Box. Параметр **LENGTH** определяет длину строки в символах. Если **MODE** определен десятичный формат, то параметр **POINT** устанавливает запятую в отображаемом числе. В параметре **POINT** указывается сколько знаков стоит справа от запятой. Если настройка параметра равна 0, функция **POINT** отключена. Если число должно содержать больше символов, чем позволяет длина, а запятая не установлена, то превышение отображается символом „#“. Если число содержит запятую, то, при необходимости, все цифры после запятой

могут быть отброшены. В шестнадцатеричном и двоичном форматах, заданных MODE, всегда будет отображаться младший разряд, если установленная длина слишком мала. Пока **ENABLE** равен 1, все изменения на входах сразу считываются. Если **VALID** равно 1, значит строка передана корректно. В случае ошибки **ERROR** принимает значение 1. В этом случае значение **VALID** равно 0. **ERRORID** содержит соответствующий код ошибки. При отрицательном фронте на **ENABLE** производится сброс значений **VALID**, **ERROR** и **ERRORID**.

**Примеры:**

Настройка	Отображаемое число	Индикация P-Box
Длина = 5 Точка = 0	12345	12345
Длина = 5 Точка = 0	-12345	#####
Длина = 10 Точка = 3	123456789	123456,789
Длина = 8 Точка = 3	123456789	123456,7

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Передача строки	BOOL	<b>VALID</b>	Передать строку	BOOL
<b>MODE</b>	Формат представления 0 = десятичный 1 = двоичный 2 = шестнадцатеричный Диапазон значений = от 0 до 2	BYTE	<b>ERROR</b>	Ошибка в ФБ	BOOL
<b>ROW</b>	Строка дисплея Диапазон значений = от 0 до 3	BYTE	<b>ERRORID</b>	Код ошибки	INT
<b>COLUMN</b>	Поле дисплея Диапазон значений = от 0 до 19	BYTE			
<b>POINT</b>	Положение запятой Диапазон значений = от 0 до 10 0 = функция выключена	BYTE			
<b>LENGTH</b>	Длина вывода Диапазон значений = от 1 до 11	BYTE			
<b>VALUE</b>	Выводимое число	DINT			
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1500h	Строка перезаписывает область памяти массива P-Box				
1501h	Превышен диапазон значений на входе LINE				
1502h	Превышен диапазон значений на входе ROW				
1504h	Превышен диапазон значений на входе POINT				
1505h	Превышен диапазон значений на входе LENGTH				
1506h	Превышен диапазон значений на входе MODE				



**Пример на ST:**

```

(* Инициализация *)
if FirstTime then
    StringToPBox.ROW := 1;
    StringToPBox.Column := 16;
    FirstTime := False;
end_if;

(* Запрос текущего положения *)
ActPos(Enable := TRUE);
if ActPos.Valid then
    (* Отображение положения в PBox (PBox P1003 = Индикация ПЛК) *)
    DintToPBox.Value := ActPos.Position;
    DintToPBox.Column := 9;
    DintToPBox.LENGTH := 10;
    DintToPBox(Enable := True);
end_if;

(* Включение или выключение устройства через DIG1 *)
Power(Enable := _5_State_digital_input.0);
if OldState <> Power.Status then
    OldState := Power.Status;
    (* Устройство включено? *)
    if Power.Status then
        StringToPBox(Enable := False, Text := TextOn);
    else
        StringToPBox(Enable := False, Text := TextOff);
    end_if;

    StringToPBox(Enable := TRUE);
else
    StringToPBox;
end_if;
    
```

**10.3.6.3 FB\_STRINGTOPBOX**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

Данный функциональный блок выполняет копирование строки (последовательности символов) в массив памяти P-Box. **ROW** и **COLUMN** используются для установки начального положения строки на дисплее модуля P-Box. Параметр **TEXT** передает требуемую строку в функциональный блок, имя строки может быть взято из таблицы переменных. Пока **ENABLE** равен 1, все изменения на входах сразу считываются. При сигнале на входе **CLEAR** все содержимое дисплея перезаписывается пробелами перед записью выбранной строки. Если **VALID** равно 1, значит строка передана корректно. В случае ошибки **ERROR** принимает значение 1. В этом случае значение **VALID** равно 0. **ERRORID** содержит соответствующий код ошибки. При отрицательном фронте на **ENABLE** производится сброс значений **VALID**, **ERROR** и **ERRORID**.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Передача строки	BOOL	<b>VALID</b>	Передать строку	BOOL
<b>CLEAR</b>	Очистить дисплей	BOOL	<b>ERROR</b>	Ошибка в ФБ	BOOL
<b>ROW</b>	Строка дисплея Диапазон значений = от 0 до 3	BYTE	<b>ERRORID</b>	Код ошибки	INT
<b>COLUMN</b>	Поле дисплея Диапазон значений = от 0 до 19	BYTE			
<b>TEXT</b>	отображаемый текст	STRING			
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1500h	Строка перезаписывает область памяти массива P-Box				
1501h	Превышен диапазон значений на входе ROW				
1502h	Превышен диапазон значений на входе COLUMN				
1503h	Выбранный номер строки не существует				
1506h	В конфигурации ПЛК не активирована опция „Allow ParameterBox function modules“.				

**Пример на ST:**

```
(* Инициализация *)
if FirstTime then
    StringToPBox.ROW := 1;
    StringToPBox.Column := 16;
    FirstTime := False;
end_if;

(* Запрос текущего положения *)
ActPos(Enable := TRUE);
if ActPos.Valid then
    (* Отображение положения в PBox (PBox P1003 = Индикация ПЛК) *)
    DintToPBox.Value := ActPos.Position;
    DintToPBox.Column := 9;
    DintToPBox.LENGTH := 10;
    DintToPBox(Enable := True);
end_if;

(* Включение или выключение устройства через DIG1 *)
Power(Enable := _5_State_digital_input.0);
if OldState <> Power.Status then
    OldState := Power.Status;
    (* Устройство включено? *)
    if Power.Status then
        StringToPBox(Enable := False, Text := TextOn);
    else
        StringToPBox(Enable := False, Text := TextOff);
    end_if;

    StringToPBox(Enable := TRUE);
else
    StringToPBox;
end_if;
```

**10.3.7 FB\_Capture (Получение быстрых результатов)**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X		

Время цикла ПЛК составляет 5 мс, если обработка внешних результатов производится очень быстро, такая длина цикла слишком велика. ФБ Capture позволяет считывать определенные физические величины по фронтам сигналов на входах ЧП. Контроль входов производится циклами продолжительностью 1 мс. Сохраненные таким образом значения потом могут считываться ПЛК.

При положительном фронте на **EXECUTE** производится считывание на всех входах и включение функции Capture. Контролируемый вход ЧП выбирается через вход **INPUT**. С помощью **EDGE** выбирается тип фронта и характер поведения функционального блока.

**EDGE = 0** При первом положительном фронте выбранное значение сохраняется в **OUTPUT1**, а **DONE1** принимает значение 1. При следующем положительном фронте значение сохраняется в **OUTPUT2**, **DONE2** принимает значение 1. После этого функциональный блок отключается.

**EDGE = 1** Поведение аналогично **EDGE = 0**, однако операция производится по отрицательному фронту.

**EDGE = 2** При первом положительном фронте выбранное значение сохраняется в **OUTPUT1**, а **DONE1** принимает значение 1. При следующем отрицательном фронте выполняется сохранение в **OUTPUT2**, **DONE2** принимает значение 1. После этого

функциональный блок отключается.

**EDGE = 3** Поведение аналогично **EDGE = 2**, но операция выполняется сначала при отрицательном, а затем при положительном фронте.

Если вход **CONTINUOUS** имеет значение 1, функция **EDGE** используется только с настройками 0 и 1. Функциональный блок непрерывно работает и сохраняет последнее событие всегда в **OUTPUT1**. Начиная с первого события **DONE1** остается активным. **DONE2** и **OUTPUT2** не используются.

Выход **BUSY** остается активным, пока не наступят оба события Capture (**DONE1** и **DONE2**).

Работа функционального блока может быть прекращена в любое время посредством отрицательного фронта на **EXECUTE**. При этом все выходы сохраняют свои значения. Положительный фронт на **EXECUTE** сначала удаляет всю информацию на всех выходах, а затем запускает функциональный блок.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>EXECUTE</b>	Выполнение	BOOL	<b>DONE1</b>	Действительное значение на OUTPUT1	BOOL
<b>CONTINUOUS</b>	Однократное выполнение или непрерывный режим.	BOOL	<b>DONE2</b>	Действительное значение на OUTPUT2	BOOL
<b>INPUT</b>	<b>SK54xE</b> Контролируемый вход 0 = вход 1 ---- 7 = вход 8  <b>SK52xE, SK53xE, SK2xxE, SK2xx-EFDS</b> Контролируемый вход 0 = вход 1 ---- 3 = вход 4	BYTE	<b>BUSY</b>	Функциональный блок ожидает событие Capture	BOOL
<b>EDGE</b>	Иницирующий фронт	BYTE	<b>ERROR</b>	Ошибка ФБ	BOOL
<b>SOURCE</b>	Сохраняемая величина 0 = положение в оборотах 1 = факт. частота 2 = момент	BYTE	<b>ERRORID</b>	Код ошибки	INT
			<b>OUTPUT1</b>	Значение для 1го события Capture	DINT
			<b>OUTPUT2</b>	Значение для 2го события Capture	DINT
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1900h	Превышен диапазон значений INPUT.				
1901h	Превышен диапазон значений EDGE.				
1902h	Превышен диапазон значений SOURCE.				
1903h	Активно более двух экземпляров объекта				

**Пример на ST:**

```

Power(ENABLE := TRUE);
IF Power.STATUS THEN
  Move(EXECUTE := TRUE, POSITION := Pos, VELOCITY := 16#2000);
  (* Capture ожидает сигнала High на DIG1. При его обнаружении функциональный блок сохраняет в
  памяти текущее положение. Значение опрашивается с помощью свойства "OUTPUT1". *)
  Capture(EXECUTE := TRUE, INPUT := 0);

  IF Capture.DONE1 THEN
    Pos := Capture.OUTPUT1;
    Move(EXECUTE := FALSE);
  END_IF;
END_IF;

```

** Информация**

Этот функциональный блок допускает несколько экземпляров объекта в программе ПЛК. Но одновременно могут быть активными только два из них!

**10.3.8 FB\_DinCounter**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	от V2.3	от V3.1	от V2.1	X	от V1.1	

Данный ФБ служит для подсчета импульсов через цифровые входы. Учитывается каждый фронт (Low – High и High – Low). Минимальная ширина импульса составляет 250мкс.

Для активации ФБ используется ENABLE. При положительном фронте происходит передача входов PV, UD, DIN и MODE, а все выходы сбрасываются.

**UD** определяет направление счета

- 0 = больший подсчет
- 1 = меньший подсчет

Значение счетчика может заноситься в PV. В зависимости от настройки входа MODE он действует по-разному.

**MODE**

- 0 = переполнение, счетчик работает как непрерывный счетчик. Переполнение может быть в положительном и отрицательном направлении. При запуске функции назначается CV = PV. В этом режиме BUSY всегда равен 1, а Q всегда 0.
- 1 = без переполнения
  - Счетчик прямого действия à CV запускается при 0, BUSY = 1, выполняется до CV=>PV. BUSY принимает значение 0, а Q равен 1. Процедура счета останавливается.
  - счетчик обратного действия à CV запускается с PV и выполняется до CV<=0. Все это время BUSY = 1 и принимает значение 0, когда достигается конец подсчета. Q, наоборот, принимает значение 1.
  - Новый запуск счетчика производится при появлении нового фронта на входе ENABLE

**DIN** определяет вход измерений. Количество входов зависит от соответствующего ЧП.

- Вход 1 = 0
- Вход 2 = 1

и т.д.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Разблокировка	BOOL	<b>Q</b>	Подсчет окончен	BOOL
<b>UD</b>	Направление подсчета  0 = больший подсчет 1 = меньший подсчет	BOOL	<b>BUSY</b>	Счетчик работает	BOOL
<b>PV</b>	Значение счетчика	INT	<b>ERROR</b>	Ошибка ФБ	BOOL
<b>MODE</b>	Режим	BYTE	<b>ERRORID</b>	Код ошибки	INT
<b>DIN</b>	Измерительный вход	BYTE	<b>CV</b>	Значение счетчика	INT
			<b>CF</b>	Частота счета (разрешение 0,1)	INT
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
0x1E00	Цифровой вход уже используется другим счетчиком				
0x1E01	Цифровой вход не существует				
0x1E02	Превышен диапазон значений MODE				

### 10.3.9 FB\_FunctionCurve

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	

Функциональный блок обеспечивает управление характеристиками. На него могут передаваться определенные точки, по которым он эмулирует функцию. Поведение выхода определяется соответствующей характеристикой. По отдельным опорным точкам выполняется линейная интерполяция. Опорные точки определяются значениями X и Y. Значения X при этом всегда имеют тип **INT**, а значения Y могут иметь тип **INT** или **DINT**, в зависимости от величины наибольшей опорной точки. При использовании типа **DINT** он занимает больше памяти. Опорные точки указываются в окне переменных в разделе "Init Value". При получении на входе **ENABLE** значения TRUE, выполняется расчет соответствующего выходного значения **OUTVALUE** на основании входного значения **INVALUE**. Значение TRUE на **VALID** сигнализирует о том, что получено действительное выходное значение **OUTVALUE**. Пока **VALID** имеет значение FALSE, выходу **OUTVALUE** соответствует значение 0. Если входное значение **INVALUE** выходит за пределы верхнего или нижнего конца характеристики, на выходе сохраняется первое или последнее выходное значение характеристики, пока **INVALUE** не вернется снова в диапазон характеристики. При выходе за верхний или нижний пределы характеристики соответствующий выход **MINLIMIT** или **MAXLIMIT** принимает значение TRUE. **ERROR** принимает значение TRUE, если абсцисса (значение X) характеристики не увеличивается, либо если таблица не инициализирована. Соответствующая ошибка также указывается посредством **ERRORID**, а выходное значение равно 0. Ошибка сбрасывается когда **ENABLE** = FALSE.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Выполнение	BOOL	<b>VALID</b>	Действительное выходное значение	BOOL
<b>INVALUE</b>	Входное значение ( x )	INT	<b>ERROR</b>	Ошибка в ФБ	BOOL
			<b>ERRORID</b>	Код ошибки	INT
			<b>MAXLIMIT</b>	Достигнут макс.предел	BOOL
			<b>MINLIMIT</b>	Достигнут мин.предел	BOOL
			<b>OUTVALUE</b>	Выходное значение ( y )	DINT
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1400h	Абсцисса (значение X) характеристики не возрастает				
1401h	Характеристика не инициализирована				

### 10.3.10 FB\_PIDT1

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

P-I-DT1 представляет собой свободно параметризуемый дискретный регулятор. Если отдельные составляющие, P, I или DT1 - составляющие, не используются, то данный параметр описывается значением 0. Составляющая T1 работает только вместе с составляющей D. То есть регулятор PT1 не доступен для параметрирования. Внутренние ограничения памяти ограничивают параметры регулирования следующими диапазонами.

<b>Допустимые диапазоны значений для параметров регулирования.</b>			
Параметр	Диапазон значений	Пересчет	полученный диапазон значений
<b>P (Kp)</b>	0 – 32767	1/100	0,00 – 32,767
<b>I (Ki)</b>	0 – 10240	1/100	0,00 – 10,240
<b>D (Kd)</b>	0 – 32767	1/1000	0,000 – 3,2767
<b>T1 (ms)</b>	0 – 32767	1/1000	0,000 – 3,2767
<b>Max</b>	-32768 – 32767		
<b>Min</b>	-32768 – 32767		



Регулятор запускается когда **ENABLE** принимает значение TRUE. Передача параметров регулирования производится только при растущем фронте **ENABLE**. Пока **ENABLE** равен TRUE, изменения параметров регулирования не имеют действия. Если **ENABLE** принимает значение FALSE, на выходе сохраняется последнее значение.

Выходной бит **VALID** используется, пока выходное значение Q изменяется в пределах диапазона минимального и максимального значений, а вход **ENABLE** равен TRUE.

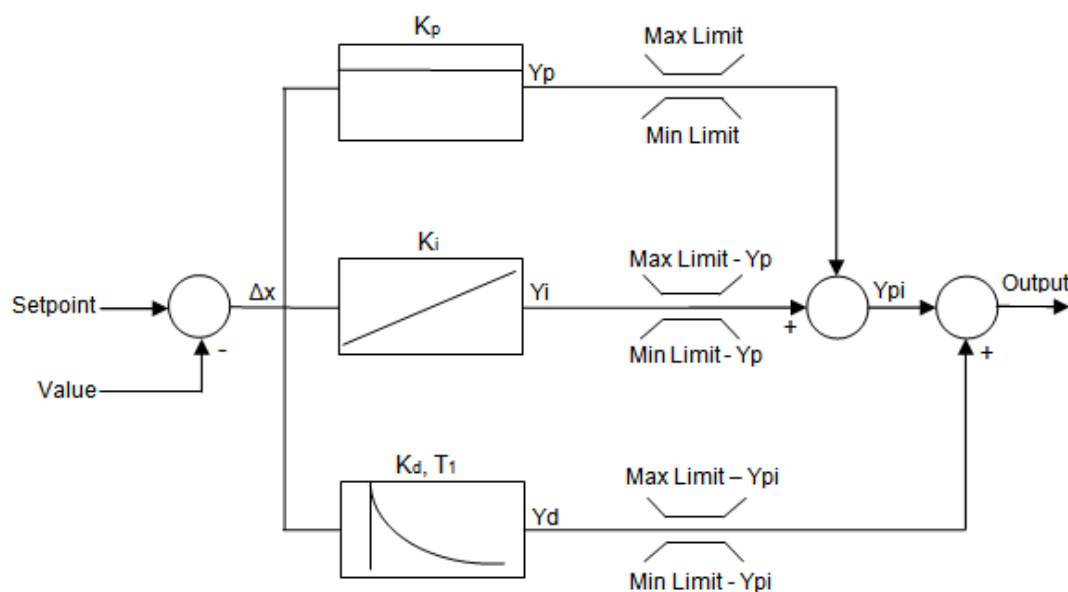
При появлении ошибки срабатывает **ERROR**. Тогда бит **VALID** принимает значение FALSE, а причина ошибки указывается посредством **ERRORID** (см. таблицу внизу).

Если бит **RESET** принимает TRUE, содержимое интегратора и дифференциатора меняется на 0. Если вход **ENABLE** принимает значение FALSE, то выход **OUTPUT** принимает значение 0. Если вход **ENABLE** принимает значение TRUE, работает только P-составляющая выхода **OUTPUT**.

Если значение на выходе **OUTPUT** больше максимального или меньше минимального выходного значения, используется соответствующий бит **MAXLIMIT** или **MINLIMIT**, а бит **VALID** принимает значение FALSE.

### Информация

Если программа не может быть обработана в рамках одного цикла, регулятор рассчитывает выходное значение второй раз со старым результатом считывания. За счет этого достигается постоянная скорость считывания. По этой причине необходимо, чтобы команда CAL выполнялась для регулятора PIDT1 в каждом цикле ПЛК и только в конце программы ПЛК!



VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>ENABLE</b>	Выполнение	BOOL	<b>VALID</b>	Действительное выходное значение	BOOL
<b>RESET</b>	Восстановить выходное значение	BOOL	<b>ERROR</b>	Ошибка в ФБ	BOOL
<b>P</b>	P-составляющая (Kp)	INT	<b>ERRORID</b>	Код ошибки	INT
<b>I</b>	I-составляющая (Ki)	INT	<b>MAXLIMIT</b>	Достигнут макс.предел	BOOL
<b>D</b>	D-составляющая (Kd)	INT	<b>MINLIMIT</b>	Достигнут мин.предел	BOOL
<b>T1</b>	T1-составляющая в мс	INT	<b>OUTPUT</b>	Выходное значение	INT
<b>MAX</b>	Макс.выходное значение	INT			
<b>MIN</b>	Мин.выходное значение	INT			
<b>SETPOINT</b>	Уставка	INT			
<b>VALUE</b>	Фактическое значение	INT			
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
1600h	P-составляющая выходит за пределы диапазона значений				
1601h	I-составляющая выходит за пределы диапазона значений				
1602h	D-составляющая выходит за пределы диапазона значений				
1603h	T1-составляющая выходит за пределы диапазона значений				

### 10.3.11 FB\_ResetPosition

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	от V2.3	от V3.1	от V2.1	X	от V1.2	

При фронте на входе **EXECUTE**, текущее положение принимает заданное значение. Для абсолютных энкодеров текущее положение может сбрасываться только на значение 0. Значение положения не используется.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
<b>EXECUTE</b>	Выполнение	BOOL			
<b>Position</b>	Положение	DINT			

10.3.12 FB\_Weigh

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	от V2.3	от V3.1	от V2.1	X	от V1.2	

Функциональный блок служит для определения среднего момента вращения при перемещении с постоянным количеством оборотов. На основании данного значения могут определяться другие физические величины, например, перемещаемый вес.

ФБ запускается при положительном фронте на **EXECUTE**. Фронт запускает передачу всех входов в ФБ. ЧП движется со скоростью вращения, заданной параметром **SPEED**. Измерение начинается при достижении времени, установленного параметром **STARTTIME**. Продолжительность измерения определяется **MEASURETIME**. ЧП останавливается после окончания времени измерения. Если вход **REVERSE** = 1, то процедура измерения запускается заново, но с инвертированным числом оборотов. В противном случае измерение прекращается, выход **DONE** принимает значение 1, а результат измерений заносится в **VALUE**.

**BUSY** активен на всем протяжении процедуры измерений.

Пересчет результатов измерений в **VALUE** выполняется по формуле  $1 = 0,01\%$  от номинального момента вращения двигателя.

Вызов другого ФБ управления движением останавливает выполнение функции измерения, а выход **ABORT** принимает значение 1.

При новом положительном фронте на **EXECUTE** все остальные выходы ФБ сбрасываются.

VAR_INPUT			VAR_OUTPUT		
Вход	Описание	Тип	Выход	Описание	Тип
EXECUTE	Выполнение	BOOL	DONE	Окончание измерений	BOOL
REVERSE	Изменение направления вращения	BOOL	BUSY	Выполнение измерений	BOOL
STARTTIME	Время до начала измерений в мс	INT	ABORT	Прерывание измерений	BOOL
MEASURETIME	Время измерений в мс.	INT	ERROR	Ошибка ФБ	BOOL
SPEED	Скорость измерений в % (нормированная по максимальной частоте, 16#4000 соотв. 100%)	INT	ERRORID	Код ошибки	INT
			VALUE	Результат измерений	INT
<b>ERRORID</b>	<b>Описание</b>				
0	Отсутствие ошибки				
0x1000	Частотный преобразователь не включен				
0x1101	Уставка частоты не задана в качестве уставки параметра (P553)				
0x1C00	Превышен диапазон значений STARTTIME				
0x1C01	Превышен диапазон значений MEASURETIME				
0x1C02	Погрешность измеренных значений относительно друг друга больше 1/8				

### Пример на ST:

```

(* Разблокировать прибор *)
Power(Enable := TRUE);
(* Прибор разблокирован? *)
if Power.Status then
  (* Задать время запуска 2000 мс *)
  Weigh.STARTTIME := 2000;
  (* Задать время измерений 1000 мс *)
  Weigh.MEASURETIME := 1000;
  (* Задать скорость 25% от максимальной *)
  Weigh.SPEED := 16#1000;
end_if;

Weigh(EXECUTE := Power.Status);
(* Взвешивание окончено? *)
if Weigh.done then
  Value := Weigh.Value;
end_if;

```

## Информация

Этот ФБ допускает только один экземпляр объекта в программе ПЛК!

## 10.4 Операторы

### 10.4.1 Арифметические операторы

## **i** Информация

Некоторые из нижеследующих операторов могут содержать дополнительные команды. Они указываются в скобках после оператора. При этом следует обратить внимание на то, что после открытой скобки должен стоять пробел. Закрывающая скобка устанавливается в отдельную программную строку.

```
LD Var1
ADD( Var2
SUB Var3
)
```

### 10.4.1.1 ABS

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных			X	X		

Создает абсолютное значение из Akku.

#### Пример на AWL:

```
LD -10 (* Загружает значение -10 *)
ABS (* Akku = 10 *)
ST Value1 (* Сохраняет значение 10 в Value1 *)
```

#### Пример на ST:

```
Value1 := ABS(-10); (* Результат равен 10 *)
```

### 10.4.1.2 ADD и ADD(

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

Складывает переменные и константы с учетом знака. Первое значение находится в Akku, второе загружается по команде ADD, либо находится внутри скобок. Команда ADD может применяться к нескольким переменным или константам. При сложении со скобками Akku прибавляется к результату выражения в скобках. Допускается использование до 6 уровней скобок. Слагаемые величины должны относиться к одному типу переменных.

#### Пример на AWL:

```
LD 10
ADD 204 (* Сложение двух констант *)
ST Value
LD 170 (* Сложение одной константы и 2 переменных. *)
ADD Var1, Var2 (* 170dez + Var1 + Var2 *)
ST Value
LD Var1
ADD( Var2
SUB Var3 (* Var1 + ( Var2 - Var3 ) *)
)
ST Value
```

**Пример на ST:**

```
Результат := 10 + 30; (* Результат равен 40 *)
Результат := 10 + Var1 + Var2;
```

**10.4.1.3 DIV и DIV(**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

Делит значение в Акки на операнд. При делении на ноль в Акки заносится максимальный возможный результат, например при делении на значения типа INT это будет значение 0x7FFF, а если делитель отрицательный, то значение равно 0x8000. При делении со скобками Акки делится на результат выражения в скобках. Допускается использование до 6 уровней скобок. Делимое и делитель должны относиться к одному типу переменных.

**Пример на AWL:**

```
LD 10
DIV 3 (* Деление двух констант *)
ST iValue (* Результат равен 9 *)
LD 170 (* Деление одной константы и 2 переменных. *)
DIV Var1, Var2 (* (170dez : Var1) : Var2 *)
ST Value
LD Var1 (* Деление Var1 на содержимое скобок*)
DIV( Var2
SUB Var3
) (* Var1 : ( Var2 - Var3 ) *)
ST Value
```

**Пример на ST:**

```
Результат := 30 / 10; (* Результат равен 3 *)
Результат := 30 / Var1 / Var2;
```

**10.4.1.4 LIMIT**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

Команда ограничивает хранящееся в Акки значение полученными минимальным и максимальным значениями. Значения. Если значение больше максимального в Акки заносится максимальное, если меньше минимального - минимальное. Если значение не выходит за границы диапазонов, оно не изменяется.

**Пример на AWL:**

```
LD 10 (* Загружает значение 10 в регистр Акки *)
LIMIT 20, 30 (* Значение сравнивается с границами 20 и 30. *)
(* Значение в Акки меньше, перезаписывается значение 20 в Акки *)
ST iValue (* Сохраняет значение 20 в Value1 *)
```

**Пример на ST:**

```
Результат := Limit(10, 20, 30); (* Результат равен 20 *)
```

**10.4.1.5 MAX**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X
	<b>BOOL</b>	<b>BYTE</b>	<b>INT</b>	<b>DINT</b>		
<b>Тип данных</b>		X	X	X		

Определяет максимальную из двух переменных или констант. При этом текущее содержимое накопительного регистра сравнивается со значением отправленным команде MAX. После выполнения команды в регистре сохраняется максимальное из двух значений. Обе величины должны относиться к одному типу переменных.

**Пример на AWL:**

```
LD 100 (* Загрузка значения 100 в Akku *)
MAX 200 (* Сравнение со значением 200 *)
ST iValue (* Сохранение 200 в Value2 (т.к. это значение больше) *)
```

**Пример на ST:**

```
Результат := Max(100, 200); (* Результат равен 200 *)
```

**10.4.1.6 MIN**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X
	<b>BOOL</b>	<b>BYTE</b>	<b>INT</b>	<b>DINT</b>		
<b>Тип данных</b>		X	X	X		

Определяет минимальную из двух переменных или констант. При этом текущее содержимое накопительного регистра сравнивается со значением отправленным команде MIN. После выполнения команды в регистре сохраняется минимальное из двух значений. Обе величины должны относиться к одному типу переменных.

**Пример на AWL:**

```
LD 100 (* Загрузка значения 100 в Akku *)
MIN 200 (* Сравнение со значением 200 *)
ST Value2 (* Сохранение 100 в Value2 (т.к. это значение меньше) *)
```

**Пример на ST:**

```
Результат := Min(100, 200); (* Сохранение 100 в Value2 (т.к. это значение меньше) *)
```

### 10.4.1.7 MOD и MOD(

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

Выполняется деление Акки на одну или несколько переменных или констант, остаток от деления сохраняется в качестве результата в Акки. При использовании скобок модуля содержимое Акки делится на результат выражения в скобках и берется по модулю. Допускается использование до 6 уровней скобок.

#### Пример на AWL:

```
LD 25 (* Загрузка делимого *)
MOD 20 (* Деление 25/20 по модулю = 5 *)
ST Var1 (* Сохранение результата 5 в Var1 *)
LD 25 (* Загрузка делимого *)
MOD( Var1 (* Результат = 25/(Var1 + 10) по модулю в Акки *)
ADD 10
)
ST Var3 (* Сохранение результата 10 в Var3 *)
```

#### Пример на ST:

```
Результат := 25 MOD 20; (* Сохранение результата 5 в Var1 *)
Результат := 25 MOD (Var1 + 10); (* Результат = 25/(Var1 + 10) по модулю в Акки *)
```

### 10.4.1.8 MUL и MUL(

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

Умножает содержимое Акки на одну или несколько переменных или констант. При умножении со скобками Акки умножается на результат выражения в скобках. Допускается использование до 6 уровней скобок. Обе величины должны относиться к одному типу переменных.

#### Пример на AWL:

```
LD 25 (* Загрузка множителя *)
MUL Var1, Var2 (* 25 * Var1 * Var2 *)
ST Var2 (* Сохранение результата *)
LD 25 (* Загрузка множителя *)
MUL( Var1 (* Результат = 25*(Var1 + Var2) *)
ADD Var2
)
ST Var3 (* Сохранение результата в Var3 *)
```

#### Пример на ST:

```
Результат := 25 * Var1 * Var2;
Результат := 25 * (Var1 + Var2);
```



## 10.4.1.9 MUX

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

С помощью одного индекса, расположенного в Акку перед командой, может осуществляться выбор различных констант или переменных. Первому значению соответствует Индекс 0. Выбранное значение загружается в Акку. Количество значений ограничивается только размером программной памяти.

## Пример на AWL:

```
LD 1 (* Выбор нужного элемента *)
MUX 10,20,30,40,Value1 (* Команда MUX для 4 констант и одной переменной *)
ST Value (* Сохраняемый Результат = 20 *)
```

## Пример на ST:

```
Результат := Mux(1, 10, 20, 30, 40, Value1) (* Сохраняемый Результат = 20 *)
```

## 10.4.1.10 SUB и SUB(

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

Вычитает одну или несколько переменных или констант из значения в Акку. При вычитании со скобками из Акку вычитается результат выражения в скобках. Допускается использование до 6 уровней скобок. Вычитаемые и уменьшаемые величины должны относиться к одному типу переменных.

## Пример на AWL:

```
LD 10
SUB Var1 (* Результат = 10 - Var1 *)
Результат ST
LD 20
SUB Var1, Var2, 30 (* Результат = 20 - Var1 - Var2 - 30 *)
Результат ST
LD 20
SUB( 6 (* Вычитание содержимого скобок из 20 *)
AND 2
) (* Результат = 20 - (6 AND 2) *)
Результат ST (* Результат = 18 *)
```

## Пример на ST:

```
Результат := 10 - Value1;
```

## 10.4.2 Расширенные математические операторы

### **i** Информация

Описанные далее операторы требуют большого объема вычислений. Их использование может привести к существенному увеличению времени выполнения программы для ПЛК.

#### 10.4.2.1 COS, ACOS, SIN, ASIN, TAN, ATAN

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X		

	BOOL	BYTE	INT	DINT
Тип данных				X

Вычисление соответствующей математической функции. Рассчитываемая величина должна находиться в Акки в дуговых минутах. Пересчет выполняется по формуле  $1 = 1000$ .

Пересчет: Угол в дуговом измерении = (угол в градусах \* PI / 180)\*1000 например, угол 90° пересчитывается следующим образом:  $90^\circ * 3.14 / 180) * 1000 = 1571$

$$AE = \sin\left(\frac{AE}{1000}\right) \cdot 1000 \quad AE = \cos\left(\frac{AE}{1000}\right) \cdot 1000 \quad AE = \tan\left(\frac{AE}{1000}\right) \cdot 1000$$

#### Пример на AWL:

```
LD 1234
SIN
Результат ST (* Результат = 943 *)
```

#### Пример на ST:

```
Результат := COS(1234); (* Результат = 330 *)
Результат := ACOS(330); (* Результат = 1234 *)
Результат := SIN(1234); (* Результат = 943 *)
Результат := ASIN(943); (* Результат = 1231 *)
Результат := TAN(999); (* Результат = 1553 *)
Результат := ATAN(1553); (* Результат = 998 *)
```

#### 10.4.2.2 EXP

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X		

	BOOL	BYTE	INT	DINT
Тип данных				X

Создает из Акки экспоненциальную функцию числа Эйлера (2,718). После запятой может быть указано 3 знака, то есть число 1,002 вводится как 1002.

$$AE = e^{\left(\frac{AE}{1000}\right)} \cdot 1000$$

**Пример на AWL:**

```
LD 1000
EXP
Результат ST (* Результат = 2718 *)
```

**Пример на ST:**

```
Результат := EXP(1000); (* Результат = 2718 *)
```

**10.4.2.3 LN**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X		
	BOOL	BYTE	INT	DINT		
Тип данных				X		

Логарифм по основанию e (2,718). После запятой может быть указано 3 знака, то есть число 1,000 вводится как 1000.

$$AE = \ln \left( \frac{AE}{1000} \right) \cdot 1000$$

**Пример на AWL:**

```
LD 1234
LN
Результат ST
```

**Пример на ST:**

```
Результат := LN(1234); (* Результат = 210 *)
```

**10.4.2.4 LOG**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X		
	BOOL	BYTE	INT	DINT		
Тип данных				X		

Создает из Акки логарифм по основанию 10. После запятой может быть указано 3 знака, то есть число 1,000 вводится как 1000.

$$AE = \log_{10} \left( \frac{AE}{1000} \right) \cdot 1000$$

**Пример на AWL:**

```
LD 1234
LOG
Результат ST (* Результат = 91 *)
```

**Пример на ST:**

```
Результат := LOG(1234); (* Результат = 91 *)
```

**10.4.2.5 SQRT**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X		
	BOOL	BYTE	INT	DINT		
Тип данных				X		

Вычисляет квадратный корень из Акку. После запятой может быть указано 3 знака, то есть число 1,000 вводится как 1000.

$$AE = \sqrt{\left(\frac{AE}{1000}\right)} \cdot 1000$$

**Пример на AWL:**

```
LD 1234
SQRT
Результат ST (* Результат = 1110 *)
```

**Пример на ST:**

```
Результат := SQRT(1234); (* Результат = 1110 *)
```

**10.4.3 Операции над битами**

**10.4.3.1 AND и AND(**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных	X	X	X	X		

Побитовая связь И значения AE/Акку с одной или двумя переменными или константами. Побитовая связь "И" (...) содержимого AE/Акку и содержимого AE/Акку, предварительно полученного в скобках. Допускается использование до 6 уровней скобок. Все величины должны относиться к одному типу переменных.

**Пример на AWL:**

```
LD 170
AND 204 (* AND связь между двумя константами *)
```

```
(* Akku = 136 (см. пример под таблицей) *)
LD 170 (* Связь между одной константой и 2 переменными.*)
AND Var1, Var2 (* Akku = 170dez AND Var1 AND Var2 *)

LD Var1
AND ( Var2 (* AE/Akku = Var1 AND ( Var2 OR Var3 ) *)
OR Var3
)
```

**Пример на ST:**

```
Результат := 170 AND 204; (* Результат = 136dez *)
```

Var2	Var1	Результат
0	0	0
0	1	0
1	0	0
1	1	1

Пример: 170dez (1010 1010bin) AND 204dez (1100 1100bin) = (1000 1000bin) 136dez

**10.4.3.2 ANDN и ANDN(**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X
	<b>BOOL</b>	<b>BYTE</b>	<b>INT</b>	<b>DINT</b>		
<b>Тип данных</b>	X	X	X	X		

Побитовая связь И значения AE/Akku с инвертированным операндом. Побитовая связь "И" (...) с AE/Akku и инвертированным результатом в скобках. Допускается использование до 6 уровней скобок. Связываемые величины должны относиться к одному типу переменных.

**Пример на AWL:**

```
LD 2#0000_1111
ANDN 2#0011_1010 (* ANDN связь между двумя константами *)
(* Akku = 2#1111_0101 *)

LD 170 (* Связь между одной константой и 2 переменными *)
ANDN Var1, Var2 (* Akku = 170d ANDN Var1 ANDN Var2 *)

LD Var1
ANDN ( Var2 (* AE/Akku = Var1 ANDN ( Var2 OR Var3 ) *)
OR Var3
)
```

Var2	Var1	Результат
0	0	1
0	1	1
1	0	1

1	1	0
---	---	---

Пример: 170dez (1010 1010bin) AND 204dez (1100 1100bin) = (1000 1000bin) 136dez

### 10.4.3.3 NOT

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных	X	X	X	X		

Побитовая инверсия Акку.

#### Пример на AWL:

```
LD BYTE#10 (* Загрузка десятичного значения в АККУ в формате Byte *)
NOT (* Значение переводится на уровень битов (0000 1010), *)
(* Выполняется побитовая инверсия (1111 0101) и обратная конвертация в десятичное значение*)
(* результат преобразования = 245дес.*)
ST Var3 (* Сохранение результата в Var3 *)
```

#### Пример на ST:

```
Результат := not BYTE#10; (* Результат = 245дес. *)
```

### 10.4.3.4 OR и OR(

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных	X	X	X	X		

Побитовая связь "ИЛИ" АЕ/Акку с одной или двумя переменными или константами. Побитовая связь "ИЛИ" (...) содержимого АЕ/Акку и содержимого АЕ/Акку, предварительно полученного в скобках. Допускается использование до 6 уровней скобок. Все величины должны относиться к одному типу переменных.

#### Пример на AWL:

```
LD 170
OR 204 (* OR связь между двумя константами *)

LD 170 (* Связь между одной константой и 2 переменными *)
OR Var1, Var2 (* Akku = 170d OR Var1OR Var2 *)

LD Var1
OR ( Var2 (* АЕ/Акку = Var1 OR ( Var2 AND Var3 ) *)
AND Var3
)
```

#### Пример на ST:

```
Результат := 170 or 204; (* Результат = 238 *)
```

Var2	Var1	Результат
0	0	0
0	1	1
1	0	1
1	1	1

#### 10.4.3.5 ORN и ORN(

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных	X	X	X	X		

Побитовая связь ИЛИ значения АЕ/Акку с инвертированным операндом. Побитовая связь "ИЛИ" (...) с АЕ/Акку и инвертированным результатом в скобках. Допускается использование до 6 уровней скобок. Обе величины должны относиться к одному типу переменных.

#### Пример на AWL:

```
LD 2#0000_1111
ORN 2#0011_1010 (* ORN связь между двумя константами *)
(* Akku = 2#1100_0000 *)

LD 170 (* Связь между одной константой и 2 переменными *)
ORN Var1, Var2 (* Akku = 170d ORN Var1 ORN Var2 *)

LD Var1
ORN ( Var2 (* АЕ/Акку = Var1 ORN ( Var2 OR Var3 ) *)
OR Var3
)
```

#### Пример на ST:

```
Результат := 2#0000_1111 ORN 2#0011_1010; (* Результат = 2#1100_0000 *)
```

Var2	Var1	Результат
0	0	1
0	1	0
1	0	0
1	1	0

#### 10.4.3.6 ROL

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS

<b>Наличие</b>	X	X	X	X	X	X
	<b>BOOL</b>	<b>BYTE</b>	<b>INT</b>	<b>DINT</b>		
<b>Тип данных</b>		X	X	X		

Побитовый циклический сдвиг содержимого Акку влево. При этом содержимое Акку сдвигается влево на n раз, левый бит перемещается на место правого.

**Пример на AWL:**

```
LD 175      (* Загружает значение 1010_1111 *)
ROL 2      (* Содержимое Акку 2 раза прокручивается влево *)
ST Value1  (* Сохраняет значение 1011_1110 *)
```

**Пример на ST:**

```
Результат := ROL(BYTE#175, 2); (* Результат = 2#1011_1110 *)
Результат := ROL(INT#175, 2); (* Результат = 16#C02B *)
```

**10.4.3.7 ROR**

	<b>SK 54xE</b>	<b>SK 53xE SK 52xE</b>	<b>SK 2xxE</b>	<b>SK 2xxE-FDS</b>	<b>SK 180E SK 190E</b>	<b>SK 155E-FDS SK 175E-FDS</b>
<b>Наличие</b>	X	X	X	X	X	X
	<b>BOOL</b>	<b>BYTE</b>	<b>INT</b>	<b>DINT</b>		
<b>Тип данных</b>		X	X	X		

Побитовый циклический сдвиг содержимого Акку вправо. При этом содержимое Акку сдвигается вправо на n раз, правый бит перемещается на место левого.

**Пример на AWL:**

```
LD 175      (* Загружает значение 1010_1111 *)
ROR 2      (* Содержимое Акку 2 раза сдвигается вправо *)
ST Value1  (* Сохраняет значение 1110_1011 *)
```

**Пример на ST:**

```
Результат := ROR(BYTE#175, 2); (* Результат = 2#1110_1011 *)
```

**10.4.3.8 S и R**

	<b>SK 54xE</b>	<b>SK 53xE SK 52xE</b>	<b>SK 2xxE</b>	<b>SK 2xxE-FDS</b>	<b>SK 180E SK 190E</b>	<b>SK 155E-FDS SK 175E-FDS</b>
<b>Наличие</b>	X	X	X	X	X	X
	<b>BOOL</b>	<b>BYTE</b>	<b>INT</b>	<b>DINT</b>		
<b>Тип данных</b>	X					

Назначение и сброс булевой переменной, если предыдущий результат связи (AE) имел значение TRUE.

**Пример на AWL:**



```
LD TRUE (* Загружает в АЕ значение TRUE *)
S Var1 (* VAR1 присваивается TRUE*)
R Var1 (* VAR1 присваивается FALSE*)
```

**Пример на ST:**

```
Результат := TRUE;
Результат := FALSE;
```

**10.4.3.9 SHL**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

Побитовый сдвиг влево Акки. При этом содержимое Акки сдвигается влево на n раз, выдвигаемые биты теряются.

**Пример на AWL:**

```
LD 175 (* Загружает значение 1010_1111 *)
SHL 2 (* Содержимое Акки 2 раза сдвигается влево *)
ST Value1 (* Сохраняет значение 1011_1100 *)
```

**Пример на ST:**

```
Результат := SHL(BYTE#175, 2); (* Результат = 2#1011_1100 *)
Результат := SHL(INT#175, 2); (* Результат = 16#2BC *)
```

**10.4.3.10 SHR**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

Побитовый сдвиг Акки вправо. При этом содержимое Акки сдвигается вправо на n раз, выдвигаемые биты теряются.

**Пример на AWL:**

```
LD 175 (* Загружает значение 1010_1111 *)
SHR 2 (* Содержимое Акки 2 раза сдвигается вправо *)
ST Value1 (* Сохраняет значение 0010_1011 *)
```

**Пример на ST:**

```
Результат := SHR(BYTE#175, 2); (* Результат = 2#0010_1011 *)
```

### 10.4.3.11 XOR и XOR(

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных	X					

Побитовая связь “exclusive OR” между АЕ/Акку и с одной или двумя переменными или константами. Первое значение находится в АЕ/Акку, второе загружается по команде, либо находится внутри скобок. Связываемые величины должны относиться к одному типу переменных.

#### Пример на AWL:

```
LD 2#0000_1111
XOR 2#0011_1010 (* XOR связь между двумя константами *)
(* Akku = 2#0011_0101 *)

LD 170 (* Связь между одной константой и 2 переменными *)
XOR Var1, Var2 (* Akku = 170d XOR Var1 XOR Var2 *)

LD Var1
XOR ( Var2 (* АЕ/Акку = Var1 XOR ( Var2 OR Var3 ) *)
OR Var3
)
```

#### Пример на ST:

```
Результат := 2#0000_1111 XOR 2#0011_1010; (* результат = 2#0011_0101 *)
```

Var2	Var1	Результат
0	0	0
0	1	1
1	0	1
1	1	0

### 10.4.3.12 XORN и XORN(

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных	X					

Побитовая связь "исключающее ИЛИ" значения АЕ/Акку с инвертированным операндом. Побитовая связь "исключающее ИЛИ" (...) с АЕ/Акку и инвертированным результатом в скобках. Допускается использование до 6 уровней скобок. Связываемые величины должны относиться к одному типу переменных.

#### Пример на AWL:

```
LD 2#0000_1111
XORN 2#0011_1010 (* XORN связь между двумя константами *)
(* Akku = 2#1100_1010 *)
```

```
LD 170 (* Связь между одной константой и 2 переменными *)
XORN Var1, Var2 (* Akku = 170d XORN Var1 XORN Var2 *)

LD Var1
XORN ( Var2 (* AE/Akku = Var1 XORN ( Var2 OR Var3 ) *)
OR Var3
)
```

**Пример на ST:**

```
Результат := 2#0000_1111 XORN 2#0011_1010; (* Результат = 2#1100_1010 *)
```

Var2	Var1	Результат
0	0	1
0	1	0
1	0	0
1	1	1

**10.4.4 Операторы загрузки и сохранения**

**10.4.4.1 LD**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X
	<b>BOOL</b>	<b>BYTE</b>	<b>INT</b>	<b>DINT</b>		
<b>Тип данных</b>	X	X	X	X		

Загружает константы или переменные в текущее значение AE или в накопительный регистр Akku.

**Пример на AWL:**

```
LD 10 (* Загружает 10 типа BYTE *)
LD -1000 (* Загружает -1000 типа INT *)
LD Value1 (* Загружает переменную Value1 *)
```

**10.4.4.2 LDN**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X
	<b>BOOL</b>	<b>BYTE</b>	<b>INT</b>	<b>DINT</b>		
<b>Тип данных</b>	X					

Загружает инвертированную булеву переменную в AE.

**Пример на AWL:**

```
LDN Value1 (* Value1 = TRUE à AE = FALSE *)
```

ST Value2 (\* Сохраняет в Value2 = FALSE \*)

#### 10.4.4.3 ST

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных	X	X	X	X		

Сохраняет содержимое АЕ/Акку в переменную. Сохраняемая переменная должна соответствовать предварительно загруженному и обработанному типу данных.

#### Пример на AWL:

LD 100 (\* Загружает значение 1010\_1111 \*)  
ST Value1 (\* Содержимое Акку 100 сохраняется в Value1 \*)

#### 10.4.4.4 STN

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных	X					

Сохраняет содержимое АЕ в переменную и инвертирует ее. Сохраняемая переменная должна соответствовать предварительно загруженному и обработанному типу данных.

#### Пример на AWL:

LD Value1 (\* Value1 = TRUE à AE = TRUE \*)  
STN Value2 (\* Сохраняет в Value2 = FALSE \*)

### 10.4.5 Операторы сравнения

#### 10.4.5.1 EQ

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

Сравнивает содержание Акку с переменной или константой. Если значения равны, АЕ присваивается TRUE.

#### Пример на AWL:

LD Value1 (\* Value1 = 5 \*)

```
EQ 10 (* AE = 5 равно 10 ? *)
JMPC NextStep (* AE = FALSE - программа не выполняет переход*)
ADD 1
NextStep:
ST Value1
```

#### Пример на ST:

```
(* Value = 10? *)
if Value = 10 then
    Value2 := 5;
end_if;
```

#### 10.4.5.2 GE

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

Сравнивает содержание Акки с переменной или константой. Если значение в Акки больше или равно переменной или константе, AE присваивается значение TRUE.

#### Пример на AWL:

```
LD Value1 (* Value1 = 5 *)
GE 10 (* 5 больше или равно 10?*) *)
JMPC NextStep (* AE = FALSE - программа не выполняет переход*)
ADD 1

NextStep:
ST Value1
```

#### Пример на ST:

```
(* 5 больше или равно 10?*) *)
if Value >= 10 then
    Value := Value - 1
end_if;
```

#### 10.4.5.3 GT

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

Сравнивает содержание Акки с переменной или константой. Если значение в Акки больше переменной или константы, AE присваивается значение TRUE.

#### Пример на AWL:

```
LD Value1(* Value1 = 12 *)
GT 8 (* 12 больше 8? *) *)
JMPC NextStep (* AE = TRUE - программа выполняет переход*)
ADD 1
NextStep:
```

ST Value1

**Пример на ST:**

```
(* 12 больше 8? *) *)
if Value > 8 then
  Value := 0;
end_if;
```

**10.4.5.4 LE**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X
	<b>BOOL</b>	<b>BYTE</b>	<b>INT</b>	<b>DINT</b>		
<b>Тип данных</b>		X	X	X		

Сравнивает содержание Akku с переменной или константой. Если значение в Akku меньше или равно переменной или константе, AE присваивается значение TRUE.

**Пример на AWL:**

```
LD Value1 (* Value1 = 5 *)
LE 10 (* 5 меньше или равно 10?*) *)
JMPC NextStep:
ST Value1
```

**Пример на ST:**

```
(* Значение меньше или равно 10?*)
if Value <= 10 then
  Value := 11;
end_if;
```

**10.4.5.5 LT**

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X
	<b>BOOL</b>	<b>BYTE</b>	<b>INT</b>	<b>DINT</b>		
<b>Тип данных</b>		X	X	X		

Сравнивает содержание Akku с переменной или константой. Если значение в Akku меньше переменной или константе, AE присваивается значение TRUE.

**Пример на AWL:**

```
LD Value1 (* Value1 = 12 *)
LT 8 (* 12 меньше 8 ? *)
JMPC NextStep (* AE = FALSE à программа не выполняет переход*)
ADD 1
NextStep:
ST Value1
```

**Пример на ST:**

```
(* Значение меньше 0? *)
```

```

if Value < 0 then
    Value := 0;
end_if;
    
```

#### 10.4.5.6 NE

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X	X	X		

Сравнивает содержание Akku с переменной или константой. Если значение в Akku не равно переменной или константе, AE присваивается значение TRUE.

#### Пример на AWL:

```

LD Value1 (* Value1 = 5 *)
NE 10 (* 5 равно 10 ?
JMPC NextStep (* AE = TRUE à программа выполняет переход*)
ADD 1
NextStep:
ST Value1
    
```

#### Пример на ST:

```

if Value <> 5 then
    Value := 5;
end_if;
    
```

## 10.5 Параметры процессов

Все аналоговые и цифровые входы и выходы, а также уставки и фактические значения шин, могут считываться и обрабатываться ПЛК, либо назначаться им (для выходных значений). Доступ к отдельным значениям осуществляется с помощью описанных далее параметров процессов. Для выходных значений выход (например, цифровой выход или уставка ПЛК) должен быть запрограммирован так, чтобы в качестве источника события выступал ПЛК. Все данные процессов при каждом новом выполнении цикла сначала считываются ПЛК с устройства, и только в конце программы для ПЛК записываются на устройство! В нижеследующей таблице представлены все параметры, к которым может напрямую обращаться функция ПЛК. Доступ к другим параметрам осуществляется через функциональные блоки MC\_ReadParameter или MC\_WriteParameter.

### 10.5.1 Входы и выходы

Здесь объединены все параметры процессов, описывающие интерфейсы входов и выходов прибора.

Имя	Функция	Нормирование	Тип	Доступ	Устройство
<code>_0_Set_digital_output</code>	Установка цифровых выходов	Бит 0: Mfr1 Бит 1: Mfr2 Бит 2: Dout1 Бит 3: Dout2 Бит 4: цифр. Функция	UINT	R/W	SK 54xE

Имя	Функция	Нормирование	Тип	Доступ	Устройство
		Aout Бит 5: Dout3 (Din7) Бит 6: Слово состояния Бит 8 Бит 7: Слово состояния Бит 9 Бит 8: BusIO Bit0 Бит 9: BusIO Bit1 Бит 10: BusIO Bit2 Бит 11: BusIO Bit3 Бит 12: BusIO Bit4 Бит 13: BusIO Bit5 Бит 14: BusIO Bit6 Бит 15: BusIO Bit7			
<code>_0_Set_digital_output</code>	Установка цифровых выходов	Бит 0: Реле 1 Бит 1: Relais 2 Бит 2: DOUT1 Бит 3: DOUT2 Бит 4: Цифр. аналог. вых. Бит 5: свободн. Бит 6: Bus PZD Бит 10 Бит 7: Bus PZD Бит 13 Бит 8: BusIO Bit0 Бит 9: BusIO Bit1 Бит 10: BusIO Bit2 Бит 11: BusIO Bit3 Бит 12: BusIO Bit4 Бит 13: BusIO Bit5 Бит 14: BusIO Bit6 Бит 15: BusIO Bit7	UINT	R/W	SK 52xE SK 53xE
<code>_0_Set_digital_output</code>	Установка цифровых выходов	Бит 0: DOUT1 Бит 1: BusIO Bit0 Бит 2: BusIO Bit1 Бит 3: BusIO Bit2 Бит 4: BusIO Bit3 Бит 5: BusIO Bit4 Бит 6: BusIO Bit5 Бит 7: BusIO Bit6 Бит 8: BusIO Bit7 Бит 9: Bus PZD Bit 10 Бит 10: Bus PZD Bit 13 Бит 11: DOUT2	UINT	R/W	SK 2xxE SK 2xxE-FDS
<code>_0_Set_digital_output</code>	Установка цифровых выходов	Бит 0: DOUT1 Бит 1: DOUT2 Бит 2: BusIO Bit0 Бит 3: BusIO Bit1 Бит 4: BusIO Bit2 Бит 5: BusIO Bit3 Бит 6: BusIO Bit4	UINT	R/W	SK 180E SK 190E



Имя	Функция	Нормирование	Тип	Доступ	Устройство
		Бит 7: BusIO Bit5 Бит 8: BusIO Bit6 Бит 9: BusIO Bit7 Бит 10: Bus PZD Bit 10 Бит 11: Bus PZD Bit 13			
<code>_0_Set_digital_output</code>	Установка цифровых выходов	Бит 0: DOUT1 Бит 1: DOUT2 Бит 2: DOUT_BRAKE Бит 3: DOUT_BUS1 Бит 4: DOUT_BUS2	UINT	R/W	SK 155E-FDS SK 175E-FDS
<code>_1_Set_analog_output</code>	Установка аналоговых выходов ЧП	10,0В = 100	BYTE	R/W	SK 54xE SK 53xE SK 52xE
<code>_2_Set_external_analog_out1</code>	Установка аналог. выхода 1 IOE	10,0В = 100	BYTE	R/W	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_3_Set_external_analog_out2</code>	Установка аналог. выхода 2 IOE	10,0В = 100	BYTE	R/W	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_4_State_digital_output</code>	Состояние цифровых выходов	Бит 0: Mfr1 Бит 1: Mfr2 Бит 2: Dout1 Бит 3: Dout2 Бит 4: цифр. Функция Aout Бит 5: Dout3 (Din7) Бит 6: Шина Слово состояния Бит8 Бит 7: Слово состояния Бит 9 Бит 8: BusIO Bit0 Бит 9: BusIO Bit1 Бит 10: BusIO Bit2 Бит 11: BusIO Bit3 Бит 12: BusIO Bit4 Бит 13: BusIO Bit5 Бит 14: BusIO Bit6 Бит 15: BusIO Bit7	INT	R	SK 54xE
<code>_4_State_digital_output</code>	Состояние цифровых выходов	P711	BYTE	R	SK 52xE SK 53xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E

Имя	Функция	Нормирование	Тип	Доступ	Устройство
<code>_4_State_digital_output</code>	Состояние цифровых выходов	Бит 0: DOUT1 Бит 1: DOUT2 Бит 2: DOUT_BRAKE Бит 3: DOUT_BUS1 Бит 4: DOUT_BUS2	BYTE	R	SK 155E-FDS SK 175E-FDS
<code>_5_State_Digital_input</code>	Состояние цифровых входов	Бит 0: DIN1 Бит 1: DIN2 Бит 2: DIN3 Бит 3: DIN4 Бит 4: DIN5 Бит 5: DIN6 Бит 6: DIN7 Бит 7: Цифровая функция AIN1 Бит 8: Цифровая функция AIN2	INT	R	SK 54xE
<code>_5_State_Digital_input</code>	Состояние цифровых входов	Бит 0: DIN1 Бит 1: DIN2 Бит 2: DIN3 Бит 3: DIN4 Бит 4: DIN5 Бит 5: DIN6 Бит 6: DIN7	INT	R	SK 52xE SK 53xE
<code>_5_State_Digital_input</code>	Состояние цифровых входов	Бит 0: DIN1 Бит 1: DIN2 Бит 2: DIN3 Бит 3: DIN4/AIN1 Бит 4: AIN2 Бит 5: Позистор Бит 6: свободн. Бит 7: свободн. Бит 8: DIN1 IOE 1 Бит 9: DIN2 IOE 1 Бит 10: DIN3 IOE 1 Бит 11: DIN4 IOE 1 Бит 12: DIN1 IOE 2 Бит 13: DIN2 IOE 2 Бит 14: DIN3 IOE 2 Бит 15: DIN4 IOE 2	INT	R	SK 2xxE SK 180E SK 190E
<code>_5_State_Digital_input</code>	Состояние цифровых входов	Бит 0: DIN1 Бит 1: DIN2 Бит 2: DIN3 Бит 3: TF (позистор) Бит 4: DIN-BUS1 (ASi1) Бит 5: DIN-BUS2 (ASi2) Бит 6: DIN-BUS3 (ASi3) Бит 7: DIN-BUS4 (ASi4) Бит 8: STO Бит 9: BDDI1 (ASIO3) Бит 10: BDDI2 (ASIO4)	INT	R	SK 155E-FDS SK 175E-FDS

Имя	Функция	Нормирование	Тип	Доступ	Устройство
<code>_5_State_Digital_inp ut</code>	Состояние цифровых входов	Бит 0: DIN1 Бит 1: DIN2 Бит 2: DIN3 Бит 3: DIN4 Бит 4: DIN5 Бит 5: DIN6/AIN1 Бит 6: DIN7/AIN2 Бит 7: Позистор Бит 8: DIN1 IOE 1 Бит 9: DIN2 IOE 1 Бит 10: DIN3 IOE 1 Бит 11: DIN4 IOE 1 Бит 12: DIN1 IOE 2 Бит 13: DIN2 IOE 2 Бит 14: DIN3 IOE 2 Бит 15: DIN4 IOE 2	INT	R	SK 2xxE-FDS
<code>_6_Delay_digital_inp uts</code>	Состояние цифровых входов после P475	Бит 0: DIN1 Бит 1: DIN2 Бит 2: DIN3 Бит 3: DIN4 Бит 4: DIN5 Бит 5: DIN6 Бит 6: DIN7 Бит 7: Цифровая функция AIN1 Бит 8: Цифровая функция AIN2	INT	R	SK 54xE
<code>_6_Delay_digital_inp uts</code>	Состояние цифровых входов после P475	Бит 0: DIN1 Бит 1: DIN2 Бит 2: DIN3 Бит 3: DIN4 Бит 4: DIN5 Бит 5: DIN6 Бит 6: DIN7	INT	R	SK 52xE SK 53xE
<code>_6_Delay_digital_inp uts</code>	Состояние цифровых входов после P475	Бит 0: DIN1 Бит 1: DIN2 Бит 2: DIN3 Бит 3: AIN1 Бит 4: AIN2 Бит 5: Позистор Бит 6: свободн. Бит 7: свободн. Бит 8: DIN1 IOE 1 Бит 9: DIN2 IOE 1 Бит 10: DIN3 IOE 1 Бит 11: DIN4 IOE 1 Бит 12: DIN1 IOE 2 Бит 13: DIN2 IOE 2 Бит 14: DIN3 IOE 2 Бит 15: DIN4 IOE 2	INT	R	SK 2xxE SK 180E SK 190E

Имя	Функция	Нормирование	Тип	Доступ	Устройство
<code>_6_Delay_digital_inputs</code>	Состояние цифровых входов после P475	Бит 0: DIN1 Бит 1: DIN2 Бит 2: DIN3 Бит 3: DIN4 Бит 4: DIN5 Бит 5: DIN6/AIN1 Бит 6: DIN7/AIN2 Бит 7: Позистор Бит 8: DIN1 IOE 1 Бит 9: DIN2 IOE 1 Бит 10: DIN3 IOE 1 Бит 11: DIN4 IOE 1 Бит 12: DIN1 IOE 2 Бит 13: DIN2 IOE 2 Бит 14: DIN3 IOE 2 Бит 15: DIN4 IOE 2	INT	R	SK 2xxE-FDS
<code>_7_Analog_input1</code>	Значение аналогового входа 1 (AIN1)	10,00B = 1000	INT	R	все
<code>_8_Analog_input2</code>	Значение аналогового входа 2 (AIN2)	10,00B = 1000	INT	R	все
<code>_9_Analog_input3</code>	Значение аналоговой функции DIN2	10,00B = 1000	INT	R	SK 54xE SK 155E-FDS SK 175E-FDS
<code>_10_Analog_input4</code>	Значение аналоговой функции DIN3	10,00B = 1000	INT	R	SK 54xE SK 155E-FDS SK 175E-FDS
<code>_11_External_analog_input1</code>	Значение аналогового входа 1 (1.IOE)	10,00B = 1000	INT	R	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_12_External_analog_input2</code>	Значение аналогового входа 2 (1.IOE)	10,00B = 1000	INT	R	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_13_External_analog_input3</code>	Значение аналогового входа 1 (2.IOE)	10,00B = 1000	INT	R	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_14_External_analog_input4</code>	Значение аналогового входа 2 (2.IOE)	10,00B = 1000	INT	R	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_15_State_analog_out</code>	Состояние аналогового	10,0B = 100	BYTE	R	SK 54xE

Имя	Функция	Нормирование	Тип	Доступ	Устройство
<code>put</code>	выхода				
<code>_16_State_ext_analog_out1</code>	Состояние аналогового выхода (1 IOE)	10,00В = 1000	INT	R	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_17_State_ext_analog_out2</code>	Состояние аналогового выхода (2 IOE)	10,00В = 1000	INT	R	SK 54xE SK 2xxE SK 180E SK 190E
<code>_18_Dip_switch_state</code>	Состояние DIP переключателя	Бит 0: DIP1 Бит 1: DIP2 Бит 2: DIP3 Бит 3: DIP4 Бит 4: DIP_I1 Бит 5: DIP_I2 Бит 6: DIP_I3 Бит 7: DIP_I4	INT	R	SK 155E-FDS SK 175E-FDS

### 10.5.2 Уставки и фактические значения ПЛК

Представленные здесь параметры процессов формируют интерфейс между ПЛК и устройством. Функция уставок ПЛК определяется параметром (P553).

#### Информация

Параметр `PLC_control_word` перезаписывает функциональный блок `MC_Power`. Уставки ПЛК перезаписывают функциональные блоки `MC_Move....` и `MC_Home`.

Имя	Функция	Нормирование	Тип	Доступ	Устройство
<code>_20_PLC_control_word</code>	Команда управления ПЛК	Соответствует профилю USS	INT	R/W	все
<code>_21_PLC_set_val1</code>	Заданное значение ПЛК 1	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_22_PLC_set_val2</code>	Заданное значение ПЛК 2	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_23_PLC_set_val3</code>	Заданное значение	100% = 4000h	INT	R/W	SK 54xE

Имя	Функция	Нормирование	Тип	Доступ	Устройство
	ПЛК 3				SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_24_PLC_set_val4</code>	Заданное значение ПЛК 4	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS
<code>_25_PLC_set_val5</code>	Заданное значение ПЛК 5	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS
<code>_26_PLC_additional_control_word1</code>	Доп. команда управления ПЛК 1	Соответствует профилю USS	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_27_PLC_additional_control_word2</code>	Доп. команда управления ПЛК 2	Соответствует профилю USS	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_28_PLC_status_word</code>	Слово состояния ПЛК	Соответствует профилю USS	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_29_PLC_act_val1</code>	Факт. значение ПЛК 1	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_30_PLC_act_val2</code>	Факт. значение ПЛК 2	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS

Имя	Функция	Нормирование	Тип	Доступ	Устройство
					SK 180E SK 190E
<code>_31_PLC_act_val3</code>	Факт. значение ПЛК 3	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_32_PLC_act_val4</code>	Факт. значение ПЛК 4	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS
<code>_33_PLC_act_val5</code>	Факт. значение ПЛК 5	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS
<code>_34_PLC_Busmaster_Control_word</code>	Команда управления ведущей функции (функция ведущей шины) через ПЛК	Соответствует профилю USS	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_35_PLC_32Bit_set_val1</code>	Уставка ПЛК 32бит - P553[1] = Low Part значения 32бит - P553[2] = High Part значения 32бит	—	LONG	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_36_PLC_32Bit_act_val1</code>	Факт. значение ПЛК 32бит - Факт. значение ПЛК 1 = Low Part значения 32бит - Факт. значение ПЛК 2 = High Part значения 32бит	—	LONG	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_37_PLC_status_bits</code>	Виртуальные выходы состояния ПЛК	Бит 0: PLC-DOUT1 Бит 1: PLC-DOUT2	INT	R/W	SK 155E-FDS SK 175E-FDS
<code>_38_PLC_control_bits</code>	Виртуальные управляющие выходы ПЛК	Бит 0: PLC-DIN1 Бит 1: PLC-DIN2 Бит 2: PLC-DIN3 Бит 3: PLC-DIN4 Бит 4: PLC-DIN5 Бит 5: PLC-DIN6	INT	R/W	SK 155E-FDS SK 175E-FDS

Имя	Функция	Нормирование	Тип	Доступ	Устройство
		Бит 6: PLC-DIN7 Бит 7: PLC-DIN8			

### 10.5.3 Уставки и действительные значения шины

Эти параметры процессов отражают все уставки и действительные значения, передаваемые на устройство по различным системам шин.

Имя	Функция	Нормирование	Тип	Доступ	Устройства
_40_Inverter_status	Слово состояния ЧП	Соответствует профилю USS	INT	R	все
_41_Inverter_act_val1	Тек.знач. 1 ПЛК	100% = 4000h	INT	R	все
_42_Inverter_act_val2	Тек.знач. 2 ПЛК	100% = 4000h	INT	R	все
_43_Inverter_act_val3	Тек.знач. 3 ПЛК	100% = 4000h	INT	R	все
_44_Inverter_act_val4	Тек.знач. 4 ПЛК	100% = 4000h	INT	R	SK 54xE
_45_Inverter_act_val5	Тек.знач. 5 ПЛК	100% = 4000h	INT	R	SK 54xE
_46_Inverter_lead_val1	Широковещательная функция ведущего устройства: ведущее значение 1	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_47_Inverter_lead_val2	Широковещательная функция ведущего устройства: ведущее значение 2	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_48_Inverter_lead_val3	Широковещательная функция ведущего устройства: ведущее значение 3	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_49_Inverter_lead_val4	Широковещательная функция ведущего устройства: ведущее значение 4	100% = 4000h	INT	R	SK 54xE
_50_Inverter_lead_val5	Широковещательная функция ведущего устройства: ведущее значение 5	100% = 4000h	INT	R	SK 54xE



Имя	Функция	Нормирование	Тип	Доступ	Устройства
_51_Inverter_control_word	Результирующее управляющее слово шины	Соответствует профилю USS	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_52_Inverter_set_val1	Результирующая главная уставка 1 шины	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_53_Inverter_set_val2	Результирующая главная уставка 2 шины	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_54_Inverter_set_val3	Результирующая главная уставка 3 шины	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_55_Inverter_set_val4	Результирующая главная уставка 4 шины	100% = 4000h	INT	R	SK 54xE
_56_Inverter_set_val5	Результирующая главная уставка 5 шины	100% = 4000h	INT	R	SK 54xE
_57_Broadcast_set_val1	Широковещательный режим, ведомое устройство: вспом. уставка 1	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_58_Broadcast_set_val2	Широковещательный режим, ведомое устройство: Вспом. уставка 2	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_59_Broadcast_set_val3	Широковещательный режим, ведомое устройство: Вспом. уставка 3	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE

Имя	Функция	Нормирование	Тип	Доступ	Устройства
					SK 2xxE SK 180E SK 190E
_60_Broadcast_set_val4	Широковещательный режим, ведомое устройство: вспом. уставка 4	100% = 4000h	INT	R	SK 54xE
_61_Broadcast_set_val5	Широковещательный режим, ведомое устройство: вспом. уставка 5	100% = 4000h	INT	R	SK 54xE
_62_Inverter_32Bit_set_val1	Результирующая главная уставка 1 шины 32 бит	- Low Part в P546[1] - High Part в P546[2]	LONG	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_63_Inverter_32Bit_act_val1	Тек.знач. 1 ЧП 32 бит	- Low Part в P543[1] - High Part в P543[2]	LONG	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_64_Inverter_32Bit_lead_val1	Ведущее значение 1 32 бит	- Low Part в P502[1] - High Part в P502[2]	LONG	R	SK 54xE SK 2xxE SK 180E SK 190E
_65_Broadcast_32Bit_set_val1	Вспом. уставка 1 ведомого устройства 32 бит, широковещ. режим	- Low Part в P543[1] - High Part в P543[2]	LONG	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_66_BusIO_input_bits	Входящие данные шины ввода-вывода	- Бит 0 – 7 = Bus I/O In бит 0 – 7 - Бит 8 = Метка 1 - Бит 9 = Метка 2 - Бит 10 = Бит 8 из упр. слова шины - Бит 11 = Бит 9 из упр. слова шины	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E

#### 10.5.4 ControlBox и ParameterBox

С помощью представленных здесь параметров процессов может осуществляться доступ к блокам управления. Это позволяет реализовывать простые задачи интерфейса HMI.

Имя	Функция	Нормирование	Тип	Доступ	Устройство
<code>_70_Set_controlbox_s how_val</code>	Отображаемое значение для ControlBox	Отображаемое значение = бит 29 – бит 0 Положение запятой = бит 31 – бит 30	DINT	R/W	все
<code>_71_Controlbox_key_s tate</code>	Состояние клавиатуры ControlBox	Бит 0: ON Бит 1: OFF Бит 2: DIR Бит 3: UP Бит 4: DOWN Бит 5: Enter	BYTE	R	все
<code>_72_Parameterbox_key _state</code>	Состояние клавиатуры ParameterBox	Бит 0: ON Бит 1: OFF Бит 2: DIR Бит 3: UP Бит 4: DOWN Бит 5: Enter Бит 6: Right Бит 7: Left	BYTE	R	все

### 10.5.5 Информационный параметр

Здесь приводятся важнейшие фактические значения прибора.

Имя	Функция	Нормирование	Тип	Доступ	Устройство
<code>_80_Current_fault</code>	текущий номер неисправности	Ошибка 10.0 = 100	BYTE	R	все
<code>_81_Current_warning</code>	текущее предупреждение	Предупреждение 10.0 = 100	BYTE	R	все
<code>_82_Current_reason_F I_blocked</code>	текущая причина для состояния "Einschaltsperr"	Проблема 10.0 = 100	BYTE	R	все
<code>_83_Input_voltage</code>	текущее сетевое напряжение	100В = 100	INT	R	все
<code>_84_Current_frequenz</code>	текущая частота	10Гц = 100	INT	R	все
<code>_85_Current_set_poin t_frequency1</code>	текущая уставка частоты, полученная из источника уставки	10Гц = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_86_Current_set_poin t_frequency2</code>	текущая уставка частоты преобразователь	10Гц = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS

Имя	Функция	Нормирование	Тип	Доступ	Устройство
					SK 180E SK 190E
<code>_87_Current_set_point_frequency3</code>	текущая уставка частоты по линейному изменению	10Гц = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_88_Current_Speed</code>	текущее рассчитанное значение частоты вращения	100об/мин = 100	INT	R	все
<code>_89_Actual_current</code>	текущее значение выходного тока	10,0A = 100	INT	R	все
<code>_90_Actual_torque_current</code>	действительный ток крутящего момента	10,0A = 100	INT	R	все
<code>_91_Current_voltage</code>	текущее значение напряжения	100В = 100	INT	R	все
<code>_92_Dc_link_voltage</code>	текущее значение напряжения в промежуточной цепи	100В = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_93_Actual_field_current</code>	действительный ток намагничивания	10,0A = 100	INT	R	все
<code>_94_Voltage_d</code>	текущее значение составляющей напряжения ось d	100В = 100	INT	R	все
<code>_95_Voltage_q</code>	текущее значение составляющей напряжения ось q	100В = 100	INT	R	все
<code>_96_Current_cos_phi</code>	текущее значение Cos(phi)	0.80 = 80	BYTE	R	все
<code>_97_Torque</code>	текущее значение крутящего момента	100% = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_98_Field</code>	текущее поле	100% = 100	BYTE	R	SK 54xE SK 53xE SK 52xE

Имя	Функция	Нормирование	Тип	Доступ	Устройство
					SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_99_Apparent_power</code>	текущее значение потребляемой мощности	1,00кВт = 100	INT	R	все
<code>_100_Mechanical_power</code>	текущая механическая мощность	1,00кВт = 100	INT	R	все
<code>_101_Speed_encoder</code>	текущее измеренное значение частоты вращения	100об/мин = 100	INT	R	SK 54xE SK 53xE SK 52xE
<code>_102_Usage_rate_motor</code>	текущий коэффициент нагрузки двигателя (мгновенное значение)	100% = 100	INT	R	все
<code>_103_Usage_rate_motor_I2t</code>	текущий коэффициент нагрузки двигателя I2t	100% = 100	INT	R	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_104_Usage_rate_brake_resistor</code>	текущий коэффициент нагрузки тормозного резистора	100% = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_105_Head_sink_temp</code>	текущая температура радиатора	100°C = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_106_Inside_temp</code>	текущая температура внутри устройства	100°C = 100	INT	R	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_107_Motor_temp</code>	текущая температура двигателя	100°C = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
<code>_108_Actual_net_frequency</code>	текущая частота сети	10Гц = 100	INT	R	SK 155E-FDS SK 175E-FDS

Имя	Функция	Нормирование	Тип	Доступ	Устройство
<code>_109_Mains_phase_sequence</code>	текущая последовательность фаз сети	0=CW, 1=CCW	BYTE	R	SK 155E-FDS SK 175E-FDS
<code>_141_Pos_Sensor_Inc</code>	Положение инкрементного датчика	0.001 оборота	DINT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
<code>_142_Pos_Sensor_Abs</code>	Положение абсолютного энкодера	0.001 оборота	DINT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
<code>_143_Pos_Sensor_Uni</code>	Положение универсального энкодера	0.001 оборота	DINT	R	SK 54xE
<code>_144_Pos_Sensor_HTL</code>	Положение HTL-датчика	0.001 оборота	DINT	R	SK 54xE
<code>_145_Actual_pos</code>	Текущее положение	0.001 оборота	DINT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
<code>_146_Actual_ref_pos</code>	Текущая уставка положения	0.001 оборота	DINT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
<code>_147_Actual_pos_diff</code>	Расхождение положений между уставкой и факт.знач.	0.001 оборота	DINT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E

### 10.5.6 Ошибки ПЛК

Метки ошибок пользователя User Error Flags могут описывать ошибки устройства из программы ПЛК от E23.0 до E23.2.

Имя	Функция	Нормирование	Тип	Доступ	Устройство
<code>_110_ErrorFlags</code>	Ошибка пользователя	Бит 0: E 23.0 бит 1: E 23.1	BYTE	R/W	все

	на устройстве	бит 2: E 23.2 бит 3: E 23.3 бит 4: E 23.4 бит 5: E 23.5 бит 6: E 23.6 бит 7: E 23.7			
<code>_111_ErrorFlags_ext</code>	Ошибка пользователя на устройстве	Бит 0: E 24.0 бит 1: E 24.1 бит 2: E 24.2 бит 3: E 24.3 бит 4: E 24.4 бит 5: E 24.5 бит 6: E 24.6 бит 7: E 24.7	BYTE	R/W	все

### 10.5.7 Параметры ПЛК

Эта группа данных процессов обеспечивает прямой доступ к параметрам ПЛК P355, P356 и P360.

Имя	Функция	Нормирование	Тип	Доступ	Устройство
<code>_115_PLC_P355_1</code>	ПЛК INT параметр P355 [-01]	-	INT	R	все
<code>_116_PLC_P355_2</code>	ПЛК INT параметр P355 [-02]	-	INT	R	все
<code>_117_PLC_P355_3</code>	ПЛК INT параметр P355 [-03]	-	INT	R	все
<code>_118_PLC_P355_4</code>	ПЛК INT параметр P355 [-04]	-	INT	R	все
<code>_119_PLC_P355_5</code>	ПЛК INT параметр P355 [-05]	-	INT	R	все
<code>_120_PLC_P355_6</code>	ПЛК INT параметр P355 [-06]	-	INT	R	все
<code>_121_PLC_P355_7</code>	ПЛК INT параметр P355 [-07]	-	INT	R	все
<code>_122_PLC_P355_8</code>	ПЛК INT параметр P355 [-08]	-	INT	R	все
<code>_123_PLC_P355_9</code>	ПЛК INT параметр P355 [-09]	-	INT	R	все
<code>_124_PLC_P355_10</code>	ПЛК INT параметр P355 [-10]	-	INT	R	все
<code>_125_PLC_P356_1</code>	ПЛК LONG параметр P355 [-01]	-	DINT	R	все
<code>_126_PLC_P356_2</code>	ПЛК LONG параметр P355 [-02]	-	DINT	R	все
<code>_127_PLC_P356_3</code>	ПЛК LONG параметр P355 [-03]	-	DINT	R	все

Имя	Функция	Нормирование	Тип	Доступ	Устройство
_128_PLC_P356_4	ПЛК LONG параметр P355 [-04]	-	DINT	R	все
_129_PLC_P356_5	ПЛК LONG параметр P355 [-05]	-	DINT	R	все
_130_PLC_P360_1	ПЛК параметр индикации P355 [-01]	-	DINT	R/W	все
_131_PLC_P360_2	ПЛК параметр индикации P355 [-02]	-	DINT	R/W	все
_132_PLC_P360_3	ПЛК параметр индикации P355 [-03]	-	DINT	R/W	все
_133_PLC_P360_4	ПЛК параметр индикации P355 [-04]	-	DINT	R/W	все
_134_PLC_P360_5	ПЛК параметр индикации P355 [-05]	-	DINT	R/W	все
_135_PLC_Scope_Int_1	ПЛК Scope Отображаемое значение 1	-	INT	R/W	все
_136_PLC_Scope_Int_2	ПЛК Scope Отображаемое значение 2	-	INT	R/W	все
_137_PLC_Scope_Int_3	ПЛК Scope Отображаемое значение 3	-	INT	R/W	все
_138_PLC_Scope_Int_4	ПЛК Scope Отображаемое значение 4	-	INT	R/W	все
_139_PLC_Scope_Bool_1	ПЛК Scope Отображаемое значение 5	-	INT	R/W	все
_140_PLC_Scope_Bool_2	ПЛК Scope Отображаемое значение 6	-	INT	R/W	все

## 10.6 Языки

### 10.6.1 Список инструкций (AWL / IL)

#### 10.6.1.1 Общие сведения

##### Тип данных

ПЛК поддерживает приведенные ниже типы данных.

Имя	Занимаемая память	Диапазон значений
BOOL	1 бит	от 0 до 1



BYTE	1 байт	от 0 до 255
INT	2 байта	от -32768 до 32767
DINT	4 байта	от -2 147 483 648 до 2 147 483 647
LABEL_ADDRESS	2 байта	Метка перехода

### Литералы

Для лучшей наглядности константы всех типов данных могут быть заданы в различных формах представления. В нижеследующей таблице приводятся все возможные варианты.

Литерал	Пример	Число в десятичном выражении
<b>Bool</b>	FALSE	0
	TRUE	1
	BOOL#0	0
	BOOL#1	1
<b>Двоичное (основа 2)</b>	2#01011111	95
	2#0011_0011	51
	BYTE#2#00001111	15
	BYTE#2#0001_1111	31
<b>Восьмеричное (основа 8)</b>	8#0571	377
	8#05_71	377
	BYTE#8#10	8
	BYTE#8#111	73
	BYTE#8#1_11	73
<b>Шестнадцатеричное (основа 16)</b>	16#FFFF	-1
	16#0001_FFFF	131071
	INT#16#1000	4096
	DINT#16#0010_2030	1056816
<b>Целочисленное (основа 10)</b>	10	10
	-10	-10
	10_000	10000
	INT#12	12
	DINT#-100000	-100000
<b>Время</b>	TIME#10s50ms	10,050 секунд
	T#5s500ms	5,5 секунд

	TIME#5.2s	5,2 секунд
	TIME#5D10H15M	5дней+10часов+15минут
	T#1D2H30M20S	1день+2часа+30минут+20секунд

## Комментарии

Для лучшего понимания содержания программы для ПЛК в последующем рекомендуется снабжать ее разделы комментариями. В пользовательской программе такие комментарии выделяются символами „(\*“ в начале и символами „\*)“ в конце, согласно нижеследующему примеру.

```
(* Комментарий к блоку программы *)
LD 100 (* Комментарий после команды *)
ADD 20
```

## Метка перехода

Операторы JMP, JMPC или JMPCN позволяют перепрыгивать целые разделы программы. В качестве целевой точки таких переходов служит метка перехода. Метка может содержать любые буквы, кроме „ß“ и умлаутов, цифры от 0 до 9 и нижнее подчеркивание. Использование других символов не допускается. В конце метки ставится двоеточие. Она может стоять сама по себе. Также после нее, в той же строке, может находиться другая команда.

Возможные варианты представлены далее:

### Пример:

```
Метка перехода:
LD 20

Das_Ist_eine_Sprungmarke:
ADD 10

MainLoop: LD 1000
```

Другой вариант: передача метки перехода в качестве переменной. Для этого переменная должна быть описана в таблице переменных с типом LABEL\_ADDRESS, чтобы в эту переменную можно было загружать метки перехода. Этот способ позволяет легко создавать машины состояний, см. ниже.

### Пример:

```
LD FirstTime
JMPC AfterFirstTime
(* Перед началом необходимо инициализировать адрес метки. *)
LD Address_1
ST Address_Var
LD TRUE
ST FirstTime
AfterFirstTime:
JMP Address_Var
Address_1:
LD Address_2
ST Address_Var
JMP Ende
Address_2:
LD Address_3
ST Address_Var
JMP Ende
Address_3:
LD Address_1
ST Address_Var
```

Окончание :

### Вызов функции

Редактор поддерживает одну форму вызова функции. В представленных далее вариантах производится вызов функции CTD с помощью экземпляра объекта I\_CTD. Результат сохраняется в переменных. Значение использованных функций описывается в руководстве далее.

#### Пример:

```
LD 10000
ST I_CTD.PV
LD LoadNewVar
ST I_CTD.LD
LD TRUE
ST I_CTD.CD
CAL I_CTD
LD I_CTD.Q
ST ResultVar
LD I_CTD.CV
ST CurrentCountVar
```

### Побитовый доступ к переменным

Для доступа к одному биту из переменной или переменной процесса может быть использован упрощенный способ записи.

Команда	Значение
LD Var1.0	загружает бит 0 из Var1 в AE
ST Var1.7	сохраняет AE в бит 7 Var1
EQ Var1.4	сравнивает AE с бит4 из Var1

## 10.6.2 Структурированный текст (ST)

Структурированный текст состоит из ряда операторов, выполняемых по условию ("IF..THEN..ELSE) или в цикле (WHILE..DO), аналогично языкам высокого уровня.

#### Пример:

```
IF value < 7 THEN
  WHILE value < 8 DO
    value := value + 1;
  END_WHILE;
END_IF;
```

### 10.6.2.1 Общие сведения

#### Типы данных в ST

ПЛК поддерживает приведенные ниже типы данных.

Имя	Занимаемая память	Диапазон значений
BOOL	1 бит	от 0 до 1
BYTE	1 байт	от 0 до 255
INT	2 байт	от -32768 до 32767
DINT	4 байт	от -2 147 483 648 до 2 147 483 647

### Операторы присваивания

В левой части оператора присваивания находится операнд (переменная, адрес), которому будет присвоено значение выражения из правой части при помощи оператора ":=".

#### Пример:

```
Var1 := Var2 * 10;
```

После выполнения данной строки переменная Var1 принимает значение Var2, умноженное на десять.

### Вызов функциональных блоков в ST

Вызов функционального блока на языке ST производится путем указания имени экземпляра объекта ФБ и требуемого значения параметра в скобках. В следующем примере производится вызов таймера с назначением IN и PT для его параметров. Затем переменная результата Q назначается переменной A.

Обращение к переменной результата, также как на языке AWL, производится с помощью имени ФБ, соединительной точки и имени переменной.

#### Пример:

```
Timer(IN := TRUE, PT := 300);  
A := Timer.Q;
```

### Обработка выражений

Обработка выражения производится путем обработки операторов согласно определенным правилам связывания. Сначала выполняется обработка операторов с сильным связыванием, а затем оператора менее сильным связыванием и т.д., пока не будут обработаны все операторы. Операторы с одинаковой силой связывания обрабатываются слева направо.

Далее представлена таблица операторов языка ST в порядке силы связывания.

Операция	Символ	Сила связывания
Заключение в скобки	(Выражение)	Сильное связывание
Вызов функции	Имя функции (список параметров)	
Инвертированное дополнение	NOT	
Умножение	*	

Деление Модуль AND	/ MOD AND	
Сложение Вычитание OR XOR	+ - OR XOR	
Сравнение Равенство Неравенство	<,>,<=,>= = <>	Слабое связывание

### 10.6.2.2 Операторы

#### Return

Оператор RETURN может использоваться для перехода к концу программы, например, при выполнении какого-либо условия.

#### IF

Оператор IF позволяет проверять выполнение условия и в зависимости от этого выполнять другие операторы.

#### Синтаксис:

```
IF <Булево_выражение1> THEN
  <IF_оператор>
ELSIF <Булево_выражение2> THEN
  <ELSIF_Оператор1>
ELSIF <Булево_выражение n> THEN
  <ELSIF_Оператор n-1>
ELSE
  <ELSE_Оператор>}
END_IF;
```

Часть в фигурных скобках {} является опциональной.

Если результатом <Булево\_выражение1> является TRUE, то выполняется <IF\_оператор>, а все последующие операторы пропускаются. В противном случае производится проверка по порядку все булевых выражений, начиная с <Булево\_выражение2>, пока одно из них не даст результат TRUE. После этого проверяются только операторы после такого булево выражения и до следующего ELSE или ELSIF. Если ни одно булево выражение не принимает значение TRUE, то обрабатываться будут только <ELSE\_операторы>.

#### Пример:

```
IF temp < 17 THEN
  Boo11 := TRUE;
ELSE
  Boo12 := FALSE;
END_IF;
```

#### CASE

Оператор CASE позволяет объединить несколько условных операторов для одной и той же условной переменной в одну логическую структуру.

**Синтаксис:**

```
CASE <Var1> OF
  <Значение 1>: <Оператор 1>
  <Значение 2>: <Оператор 2>
  <Значение3, Значение4, Значение5: <Оператор 3>
  <Значение6 .. Значение10 : <Оператор 4>
  ...
  <Значение n>: <Оператор n>
ELSE <ELSE-Оператор>
END_CASE;
```

Оператор CASE обрабатывается по следующей схеме:

- Если переменная <Var1> принимает значение <Значение i>, то выполняется оператор <Оператор i>
- Если <Var 1> не принимает одно из заданных значений, выполняется <ELSE-оператор>.
- Если при нескольких значениях переменных выполняется один и тот же оператор, то можно записать несколько значений через запятую, объединив их в общее условие для оператора.
- Если данный оператор выполняется для определенного диапазона значений переменной, то начальное и конечное значение следует указать друг за другом через двоеточие, объединив их в общее условие для оператора.

**Пример:**

```
CASE INT1 OF
  1, 5:
    BOOL1 := TRUE;
    BOOL3 := FALSE;
  2:
    BOOL2 := FALSE;
    BOOL3 := TRUE;
  10..20:
    BOOL1 := TRUE;
    BOOL3:= TRUE;
  ELSE
    BOOL1 := NOT BOOL1;
    BOOL2 := BOOL1 OR BOOL2;
END_CASE;
```

**Цикл FOR**

Цикл FOR позволяет запрограммировать повторяющиеся процедуры.

**Синтаксис:**

```
FOR <INT_Var> := <INIT_ЗНАЧ> TO <END_ЗНАЧ> {BY <размер шага>} DO
  <Операторы>
END_FOR;
```

Часть в фигурных скобках {} является опциональной. <Операторы> выполняются до тех пор, пока счетчик <INT\_Var> не превысит конечное значение <END\_ЗНАЧ>. Условие проверяется перед выполнением <Операторов>, поэтому <Операторы> никогда не будут выполнены, если начальное значение <INIT\_ЗНАЧ> больше конечного <END\_ЗНАЧ>. Всегда, когда <Операторы> выполняются, значение <INT\_Var> увеличивается на <размер шага>. Размер шага может быть любым целым значением. Если он не задан, то по умолчанию принимает значение 1. При этом цикл нужно будет прервать, так как <INT\_Var> будет только увеличиваться.

**Пример:**

```
FOR Счетчик :=1 TO 5 BY 1 DO
  Var1 := Var1 * 2;
END_FOR;
```

## Цикл REPEAT

Цикл REPEAT отличается от цикла WHILE тем, что условие выхода из цикла проверяется только после выполнения цикла. В результате этого цикл будет выполнен как минимум один раз, независимо от условия выхода.

### Синтаксис:

```
REPEAT
  <Операторы>
UNTIL <Булево выражение>
END_REPEAT;
```

<Операторы выполняются до тех пор, пока <Булево выражение> не примет значение TRUE. Если <Булево выражение> принимает значение TRUE уже при первой проверке, то <Операторы> также выполняются ровно один раз. Если <Булево \_выражение> никогда не принимает значение TRUE, то <Операторы> повторяются бесконечно, что приведет к ошибке периода выполнения.

---

## Информация

Программист должен самостоятельно следить за тем, чтобы циклы не были бесконечными, когда он изменяет условие в разделе операторов цикла, например, увеличении или уменьшении счетчика.

---

### Пример:

```
REPEAT
  Var1 := Var1 * 2;
  Счетчик := Счетчик - 1;
UNTIL
  Счетчик = 0
END_REPEAT
```

## Цикл WHILE

Цикл WHILE может использоваться как и цикл FOR, с тем отличием, что условием его прерывания может быть любое булево выражение. То есть задается условие, при наступлении которого цикл выполняется.

### Синтаксис:

```
WHILE <Булево выражение> DO
  <Операторы>
END_WHILE;
```

<Операторы выполняются до тех пор, пока <Булево выражение> не примет значение FALSE. Если <Булево выражение> принимает значение FALSE уже при первом выполнении, то <Операторы> также выполняются ровно один раз. Если <Булево \_выражение> никогда не принимает значение FALSE, то <Операторы> повторяются бесконечно, что приведет к ошибке периода выполнения.

## **i** Информация

Программист должен самостоятельно следить за тем, чтобы циклы не были бесконечными, когда он изменяет условие в разделе операторов цикла, например, увеличении или уменьшении счетчика.

### Пример:

```
WHILE Счетчик >0 DO
  Var1 := Var1 * 2;
  Счетчик := Счетчик - 1;
END_WHILE
```

### Exit

Когда команда EXIT срабатывает при выполнении цикла FOR, WHILE или REPEAT, выполнение самого внутреннего цикла заканчивается, независимо от условия прерывания.

## 10.7 Переходы

### 10.7.1 JMP

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

Безусловный переход к метке перехода.

### Пример на AWL:

```
JMP NextStep (* Безусловный переход к NextStep *)
ADD 1

NextStep:
ST Value1
```

### 10.7.2 JMPC

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
<b>Наличие</b>	X	X	X	X	X	X

Условный переход (Jump Conditional) к метке перехода. Если AE = TRUE то команда JMPC выполняет переход к заданной метке.

### Пример на AWL:

```
LD 10
JMPC NextStep (* AE = TRUE - программа выполняет переход*)
ADD 1

NextStep:
ST Value1
```

### 10.7.3 JMPCN

	SK 54xE	SK 53xE	SK 2xxE	SK 2xxE-FDS	SK 180E	SK 155E-FDS
--	---------	---------	---------	-------------	---------	-------------



		SK 52xE			SK 190E	SK 175E-FDS
Наличие	X	X	X	X	X	X

Условный переход (Jump Conditional) к метке перехода. JMPCN выполняет переход если регистр AE = FALSE. В противном случае программа продолжает выполнение последующих выражений.

**Пример на AWL:**

```
LD 10
JMPCN NextStep (* AE = TRUE à программа не выполняет переход*)
ADD 1

NextStep:
ST Value1
```

## 10.8 Конвертация типов

### 10.8.1 BOOL\_TO\_BYTE

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X

	BOOL	BYTE	INT	DINT
Тип данных	X			

Конвертирует тип данных AE из BOOL в BYTE. Если AE равна FALSE, Akku принимает значение 0. Если AE равна TRUE, Akku принимает значение 1.

**Пример на AWL:**

```
LD TRUE
BOOL_TO_BYTE (* AE = 1 *)
```

**Пример на ST:**

```
Результат := BOOL_TO_BYTE(TRUE); (* Результат = 1 *)
```

### 10.8.2 BYTE\_TO\_BOOL

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X

	BOOL	BYTE	INT	DINT
Тип данных		X		

Конвертирует тип данных из BYTE в BOOL. Пока BYTE не равно нулю, результатом конвертирования всегда будет TRUE.

**Пример на AWL:**

```
LD 10
BYTE_TO_BOOL (* AE = TRUE *)
```

**Пример на ST:**

```
Результат := BYTE_TO_BOOL(10); (* Результат = TRUE *)
```

### 10.8.3 BYTE\_TO\_INT

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных		X				

Конвертирует тип данных из BYTE в INT. Переменная BYTE копируется в нижнюю часть переменной INT, а в верхнюю часть INT записывается 0.

**Пример на AWL:**

```
LD 10
BYTE_TO_INT (* Akku = 10 *)
```

**Пример на ST:**

```
Результат := BYTE_TO_INT(10); (* Результат = 10 *)
```

### 10.8.4 DINT\_TO\_INT

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
Тип данных				X		

Конвертирует тип данных из DINT в INT. При этом верхняя часть значения переменной DINT не переносится.

**Пример на AWL:**

```
LD 200000
DINT_TO_INT (* Akku = 3392 *)

LD DINT# -5000
DINT_TO_INT (* Akku = -5000 *)

LD DINT# -50010
DINT_TO_INT (* Akku = 15526 *)
```

**Пример на ST:**

```
Результат := DINT_TO_INT(200000); (* Результат = 3392 *)
Результат := DINT_TO_INT(-5000); (* Результат = -5000 *)
Результат := DINT_TO_INT(-50010); (* Результат = 15526 *)
```

### 10.8.5 INT\_TO\_BYTE

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X

	BOOL	BYTE	INT	DINT
Тип данных			X	

Конвертирует тип данных из INT в BYTE. При этом верхняя часть значения переменной INT не переносится. Знак теряется, так как тип BYTE не имеет знака.

#### Пример на AWL:

```
LD 16#5008
INT_TO_BYTE (* Akku = 8 *)
```

#### Пример на ST:

```
Результат := INT_TO_BYTE(16#5008); (* Результат = 8 *)
```

### 10.8.6 INT\_TO\_DINT

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
Наличие	X	X	X	X	X	X

	BOOL	BYTE	INT	DINT
Тип данных			X	

Конвертирует тип данных из INT в DINT. Переменная INT копируется в нижнюю часть переменной DINT, а в верхнюю часть DINT записывается 0.

#### Пример на AWL:

```
LD 10
INT_TO_DINT (* Akku = 10 *)
```

#### Пример на ST:

```
Результат := INT_TO_DINT(10); (* Результат = 10 *)
```

## 10.9 Сообщения о неисправностях ПЛК

Чтобы не допустить повреждения, при возникновении ошибки устройство отключается. При появлении сообщения о неисправности ПЛК его работа останавливается и ПЛК переводится в состояние „PLC-Error“. В случае появления других сообщений о неисправностях работа ПЛК продолжается. После обработки ошибки работа ПЛК возобновляется автоматически.

**При возникновении ошибки ПЛК PLC User Fault 23.X работа ПЛК продолжается!**

Вывод на дисплее SimpleBox		Неисправность Текстовое сообщение в Parameter Box	Причина Способ устранения
Группа	Описание в P700 [-01] / P701		
E022	22.0	Отсутствует программа для ПЛК	ПЛК запущен, но на ЧП отсутствует программа для ПЛК. - Загрузить программу для ПЛК на устройство
	22.1	Ошибка программы для ПЛК	В результате проверки контрольных сумм программой ПЛК произошла ошибка. - Заново запустить устройство (Power ON) и повторить попытку - или заново загрузить программу для ПЛК
	22.2	Неверный адрес перехода	Ошибка программы, аналогично ошибке 22.1
	22.3	Переполнен стек	При выполнении программы было открыто более 6 уровней скобок. - Проверить программу на наличие ошибок периода выполнения
	22.4	Превышено макс. время цикла ПЛК	Превышена заданная максимальная продолжительность цикла программы ПЛК. - Привести в соответствие продолжительность цикла или проверить программу
	22.5	Неизвестный код команды	Невозможно выполнить один из кодов команды, указанный в программе, так как такой код неизвестен. - Программная ошибка, аналогично ошибке 22.1, - Версия ПЛК не соответствует версии NORD CON
	22.6	Доступ записи ПЛК	В ходе выполнения программы для ПЛК ее содержание было изменено.
	22.9	Суммарная ошибка ПЛК	Невозможно точно установить причину ошибки. - Аналогично ошибке 22.1
E023	23.0	PLC User Fault 1	Ошибка может быть вызвана программой ПЛК, чтобы не подавать внешний сигнал о проблеме в ходе выполнения программы для ПЛК. Устранение выполняется посредством описания переменной процесса „ErrorFlags“.
	23.1	PLC User Fault 2	
	23.2	PLC User Fault 3	

## 11 Режим проекта

### 11.1 Общая информация

Режим проекта — это расширение нормального режима. Как правило, режим проекта отключен, и для работы в нем необходимо выполнить ряд настроек. Режим проекта позволяет управлять проектом, загружать и сохранять проекты. В проект включены устройства вместе с их данными (параметрами и программами ПЛК), ссылки на внешние файлы параметров или на программы ПЛК, а также раскладки окон приложений. После запуска NORD CON загружается последний сохраненный проект. Если программе не удастся обнаружить такой проект, создается новый. Режим проектов позволяет выполнять следующие действия:

- 11.2 "Интерфейс «человек-машина» (HMI)"
- 11.3 "Сохранение и восстановление"

Категория	Имя	Описание
File	New project	Создает пустой проект.
	Open project...	Открывает окно, в котором пользователь может выбрать файл проекта (*.ncrx).
	Save project	Открывает окно для сохранения файла. Пользователь должен указать новое имя для файла проекта (*.ncrx), после этого проект будет сохранен под этим именем.
	Save all	Сохраняет проект под текущим именем.
Project	Send all data	Передает все параметры и программу ПЛК на устройства.
	Read all data	Загружает все параметры с устройства и сохраняет в файле проекта. Дополнительно производится сравнение программы ПЛК на устройстве с данными в проекте. При обнаружении отличий выводится соответствующее предупреждение.
	Remove parameter	Удаляет из проекта параметры отмеченного устройства.
	Remove PLC program	Удаляет из проекта программу ПЛК отмеченного устройства.
	Add PLC program	Добавляет в выбранное устройство сохраненную программу ПЛК.
	Export parameters	Сохраняет все параметры выбранного устройства в файле.
	Export PLC program	Сохраняет программу ПЛК выбранного устройства в файле.
PLC	Save	Сохраняет ПЛК в файле проекта.
	Save as ...	Открывает окно, в котором пользователь может выбрать имя файла. После этого программа ПЛК будет сохранена в файле с этим именем.
Parameter setup	Save	Сохраняет параметры в файле проекта.
	Save as ...	Открывает окно, в котором пользователь может выбрать имя файла.

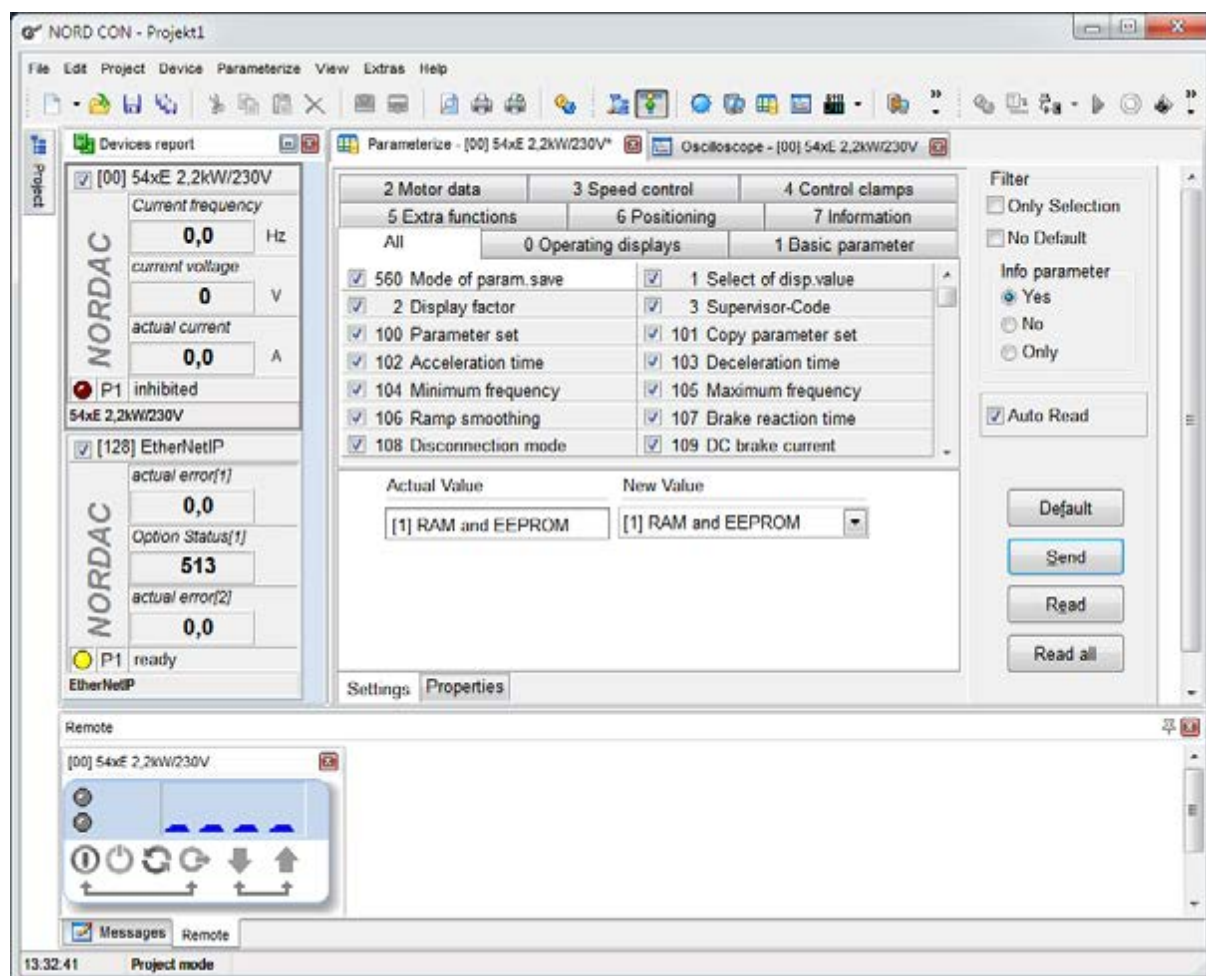
Категория	Имя	Описание
		После этого параметры сохраняются в файле с этим именем.

## 11.2 Интерфейс «человек-машина» (HMI)

Режим проекта предназначен прежде всего для визуализации данных и процессов. Пользователь должен подключить ПК к системе и запустить поиск устройства (функцию сканирования шины, Ctrl F5). После обнаружения устройств пользователь может вложить в рабочую область окна для каждого устройства требуемые элементы отображения, такие как окно параметров, осциллограф или окно управления. После этого следует сохранить проект. При открытии проекта в окне отображаются сохраненные устройства и восстанавливается предшествующая структура окна. То есть, пользователь всегда работает в одном и том же интерфейсе.

### Информация

При загрузке проекта следует убедиться, что к системе подключены устройства, указанные в проекте. Если к шине подключены другие устройства, возможен вывод ошибок подключения. При использовании системной шины необходимо учитывать, что сетевое подключение должно всегда устанавливаться с одними и теми же устройствами.



### 11.3 Сохранение и восстановление

Одна из функций, которую выполняет режим проекта, является сохранение и восстановление параметров и программ ПЛК. Пользователь может ограничить список используемых устройств, полученный после сканирования шины. Чтобы исключить устройства из процессов сохранения и восстановления, необходимо отключить их в списке устройств.

Выполнение некоторых процедур может занимать некоторое время, особенно если установка оснащена несколькими устройствами. Ход выполнения отображается в отдельном окне. При выполнении некоторых процедур пользователь не может работать в NORD CON.

#### **Информация**

При загрузке проекта следует убедиться, что к системе подключены устройства, указанные в проекте. Если к шине подключены другие устройства, возможен вывод ошибок подключения. При использовании системной шины необходимо учитывать, что сетевое подключение должно всегда устанавливаться с одними и теми же устройствами.

#### **Сохранение**

Функция «Read all Data» позволяет прочитать параметры всех найденных в процессе сканирования шины устройств. Параметры сначала сохраняются в NORD CON, после чего их нужно сохранить вручную в файле проекта («Сохранить все»). Функция «Read all Data» имеет три параметра, которые можно включить или отключить в окне настроек.

Опция	Описание
Delete all data records on cancellation	Если включен этот параметр, при отмене операции «Read all Data» из проекта удаляются все наборы параметров. В противном случае после отмены операции считывания данных данные могут сохраниться не полностью.
Delete incomplete data records	Если включен этот параметр, при возникновении ошибки во время выполнения операции «Read all Data» набор данных устройства удаляется.
Delete data records from devices not ready for communication	Если включен этот параметр, при отсутствии ответа со стороны устройства во время выполнения операции «Read all Data» набор данных устройства удаляется.

Текущая версия NORD CON не может считывать программы ПЛК. По этой причине при выполнении операции «Read all Data» производится сравнение программ на устройстве и в файле проекта. Если программы различаются, NORD CON выводит предупреждение. Если программа ПЛК устройства не сохранена, сравнение не производится.

Устройства, у которых в файле проекта сохранены параметры, отмечены в дереве проекта специальным значком. Значком также выделяются устройства, имеющие программы ПЛК. Наличие значка на самом деле не свидетельствует о полноте и точности сохраненных данных. После считывания параметр можно изменить в редакторе параметров. Для этого необходимо выбрать устройство в дереве проекта и открыть редактор параметров (F7). В редакторе можно получить с устройства и при необходимости изменить параметры. Функция «Save» сохраняет параметры выбранного устройства в проекте и на жестком диске. Чтобы сохранить параметры в другом файле, следует нажать «Save as».

#### **Информация**

Если при выполнении операции «Read all Data» возникает ошибка, информация об этом сохраняется в журнале и в дальнейшем работа продолжается с резервной копией данных. Параметры, зарегистрированные в журнале ошибок, не сохраняются в файле проекта. Рекомендуется устранить причину ошибки и использовать резервную копию устройства.

#### **Восстановление**

Эта функция доступна после открытия проекта в главном меню программы. Она управляет на устройство параметры, сохраненные в файле проекта. Как правило, на устройство отправляются все параметры. Однако в большинстве случаев в этом нет необходимости. Кроме того, передача всех параметров может занимать значительное время. Чтобы уменьшить число параметров, необходимо включить опцию «Only transfer enabled parameters» и в редакторе параметров активировать нужные параметры.

На втором этапе из проекта загружаются программы ПЛК, которые затем компилируются и отправляются на устройство. Программу ПЛК устройства можно отредактировать в редакторе ПЛК. При открытии редактора программа ПЛК автоматически загружается из файла проекта. После редактирования программу можно сохранить в файле проекта, нажав «Save». Чтобы сохранить программу ПЛК в другом файле, следует нажать «Save as».

### Информация

Если при выполнении этих операций возникнет ошибка, запись об ошибке сохраняется в журнале, процесс не прерывается. При этом на устройство не передаются параметры, зарегистрированные в журнале ошибок. Обработка программ ПЛК производится аналогичным образом. Рекомендуется устранить причину ошибки и повторить операцию.

## 11.4 Загрузка проекта

Автоматическая загрузка проекта с помощью командного файла загружать параметры и программу ПЛК на одно или несколько устройств. Результат передачи сохраняется в файле протокола и затем может быть проанализирован. Параметры и программы ПЛК необходимо сохранять до проекта. Для этого необходимо включить режим проекта в NORD CON. После поиска устройства в каталоге проекта отражаются все найденные устройства. Теперь параметры и программу ПЛК можно закрепить за любым из выбранных устройств.

### Информация

Программы ПЛК можно закреплять только за устройствами с функцией ПЛК!

Для более быстрой загрузки проекта можно передавать только нужные параметры. С этой целью в редакторе параметров можно закрепить значения только за этими параметрами.

Для настройки конфигурации загрузки проекта в каталоге установок NORD CON сохранен командный файл. Этот файл необходимо скопировать и соответствующим образом скорректировать. Для обеспечения функционирования предусмотрены следующие параметры передачи:

Параметры передачи	Описание
AUTODOWNLOAD=[project file] (АВТОМАТИЧЕСКАЯ ЗАГРУЗКА=[файл проекта])	Этот параметр передачи включает загрузку проекта. После знака «равно» необходимо ввести адрес файла проекта.  Пример: „AUTODOWNLOAD=c:\Projekt_Starter.ncpx“ (АВТОМАТИЧЕСКАЯ ЗАГРУЗКА=c:\Projekt_Starter.ncpx)
CONNECTIONSTRING=[ID=1, PORTNR=[COMx (x=serial port number)], BAUDRATE=[baud rate]] (СТРОКА ПОДКЛЮЧЕНИЙ=[ID=1, PORTNR=[COMx (x=номер	Этот параметр передачи устанавливает параметры коммуникации. Если этот параметр не передается, используются настройки из проекта.  Пример: "CONNECTIONSTRING=ID=1,PORTNR=COM1,BAUDRATE=38400" ("СТРОКА ПОДКЛЮЧЕНИЙ=ID=1,НОМЕР ПОРТА=COM1, СКОРОСТЬ



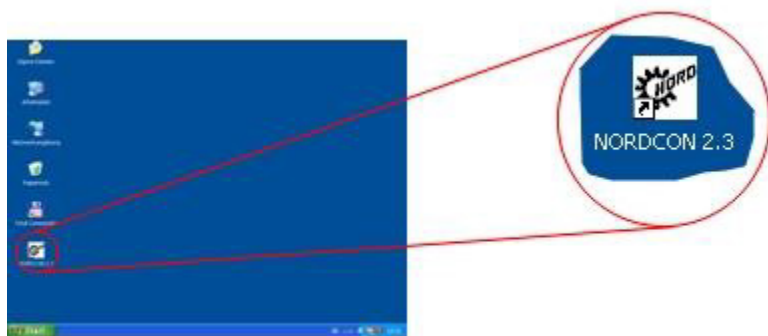
серийного порта], BAUDRATE=[скорость передачи данных]]) (по заказу)	ПЕРЕДАЧИ ДАННЫХ=38400»)
AUTOLOG=[log file] (АВТОМ.ЖУРНАЛ=[системный журнал])	Этот параметр передачи устанавливает адрес протокольного файла. Если .этот параметр не передается, файл протокола не создается.

## 12 Встроенное программное обеспечение

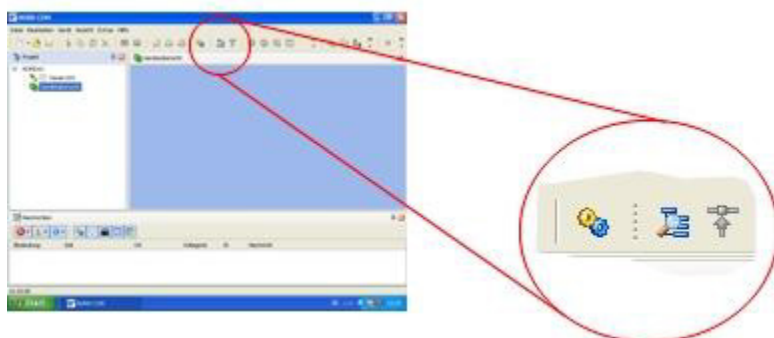
### 12.1 Обновление встроенного программного обеспечения

Чтобы обновить встроенное ПО, необходимо выполнить следующее:

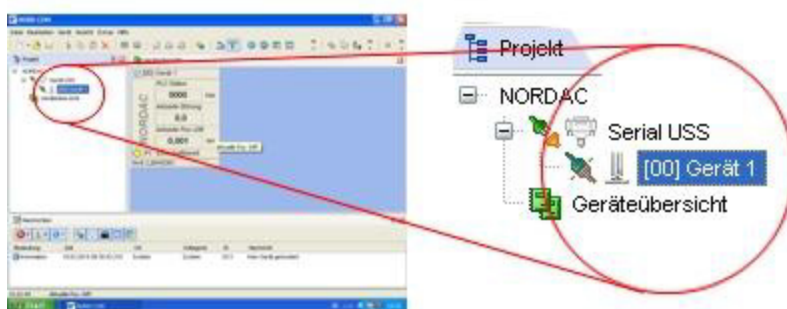
1. Запустить NORD CON



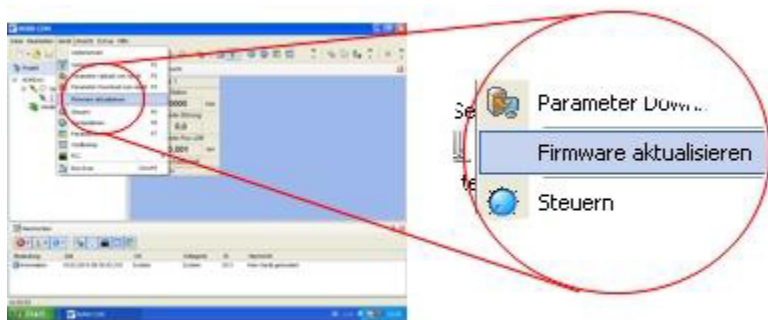
2. Выполнить поиск устройств



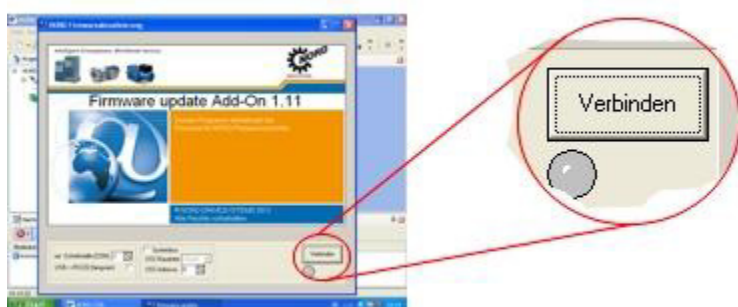
3. Отметить нужное устройство в дереве проекта



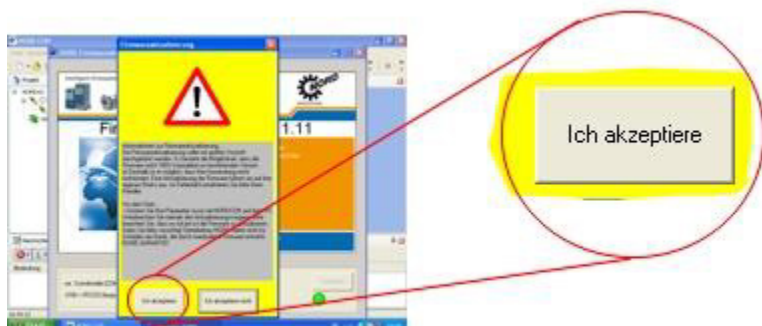
4. Запустить приложение для обновления программного обеспечения в меню «Device -> Update firmware»



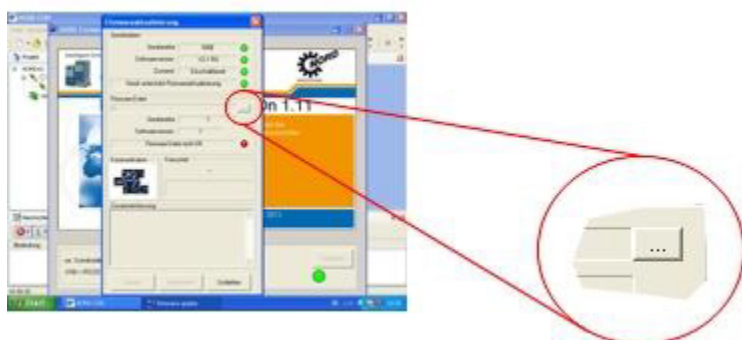
5. Нажать на подключение



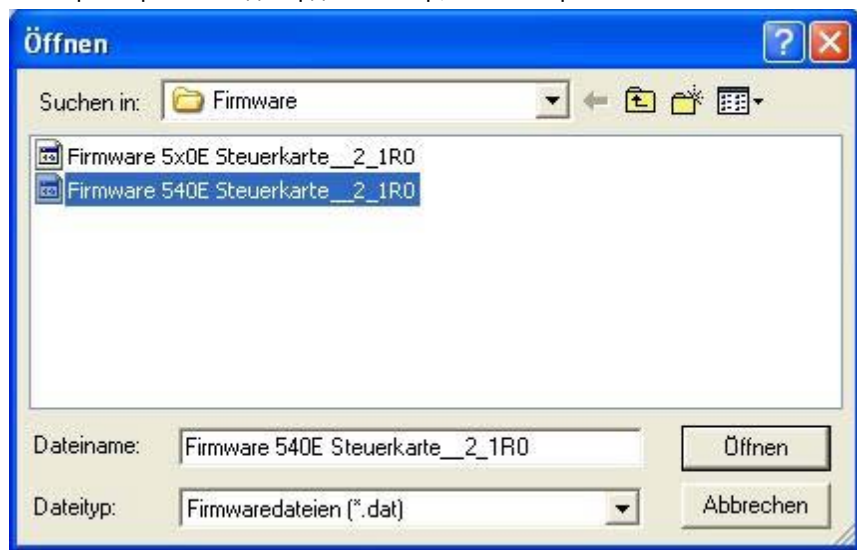
6. Внимательно прочитать предупреждение и подтвердить действие, нажав «I accept»



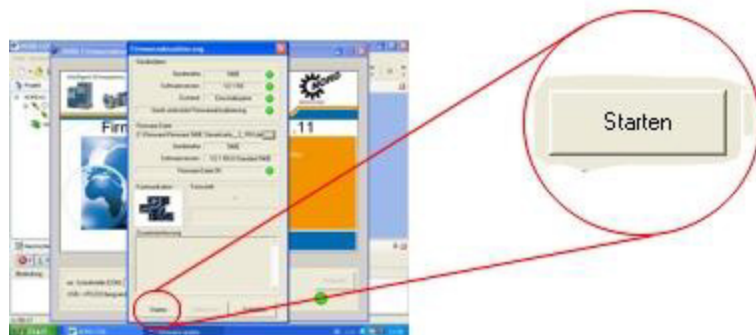
7. Найти файл встроенного программного обеспечения, нажав «...»



8. Выбрать файл и подтвердить выбор, нажав «Open»



9. Запустить передачу ПО устройства, нажав «Start»



### **i** Информация

Обновление программного обеспечения возможно только на устройствах с адресом 0 и скоростью передачи данных 38400 бит/с.

### **i** Информация

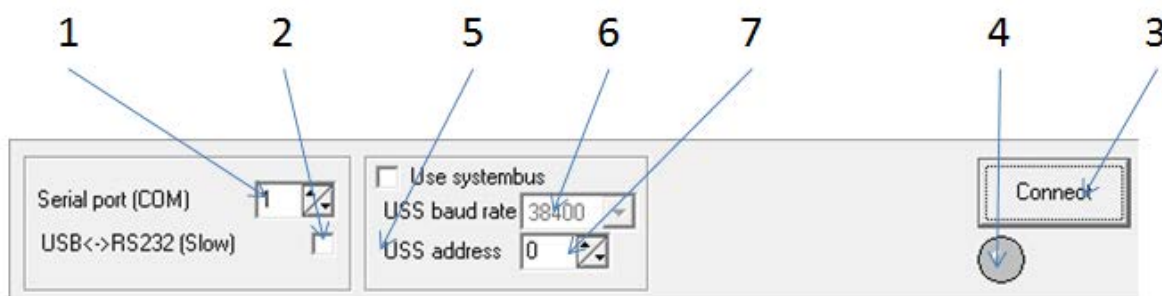
При прерывании передачи данных в процессе обновления ПО или возникновении ошибки необходимо перезапустить устройство. Если после этого устройство не удается найти в сети с помощью функции сканирования шины, обновление встроенной программы (файл FirmwareUpd.exe) можно запустить вручную. Программа находится в главном каталоге NORD CON.

## 12.2 Программа для обновления встроенного ПО

### 1. Настройки

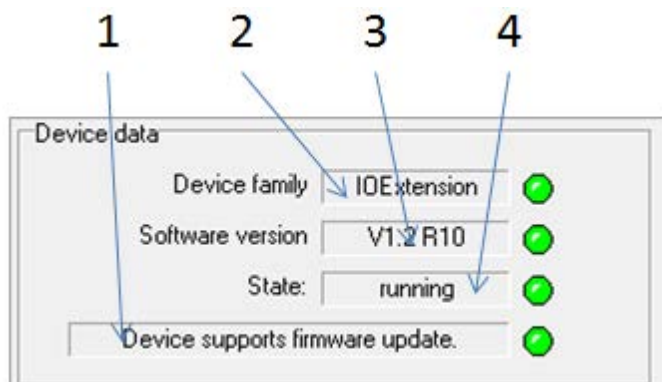
№	Описание
1	Список, в котором можно выбрать COM-порт, к которому подключено устройства. Если программа вызывается из NORD CON, этот параметр не следует менять.
2	При использовании преобразователей USB -> RS232 некоторых типов эта настройка может обеспечить стабильную связь между устройствами. Рекомендуется выбирать эту настройку

№	Описание
	только при наличии проблем при передаче данных.
3	Кнопка «Connect» устанавливает подключение к подсоединенному устройству. Если устройство обнаружено, индикатор (4) горит зеленым светом и открывается окно загрузки встроенного ПО.
4	Этот индикатор указывает на состояние подключения. Серый: подключение отсутствует. Зеленый: программа подключена к устройству. Красный: устройства не обнаружены.
5	Включено обновление встроенного ПО через системную шину.
6	Поле выбора, в котором пользователь может выбрать скорость передачи данных между ПК и подключенным устройством.
7	Поле ввода для задания адреса USS (преобразованного).



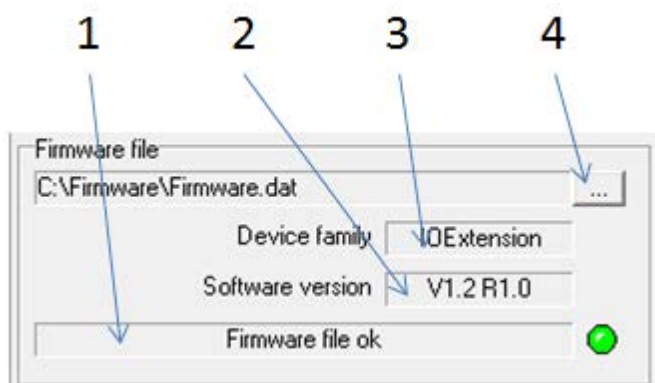
### 2. Характеристики устройства

№	Описание
1	В этом поле отображается, поддерживает ли подключенное устройство обновление встроенного ПО. Если устройство не поддерживает, индикатор рядом с полем имеет красный цвет.
2	В этом поле отображается семейство, к которому принадлежит подключенное устройство. Если устройства не обнаружены, индикатор рядом с полем имеет красный цвет. Обновление встроенного ПО в этом случае невозможно!
3	В этом поле отображается номер версии подключенного устройства.
4	В этом поле отображается состояние подключенного устройства. Если устройство разблокировано, индикатор рядом с полем имеет красный цвет. В этом случае обновление встроенного ПО невозможно!



### 3. Выбор файла встроенного ПО

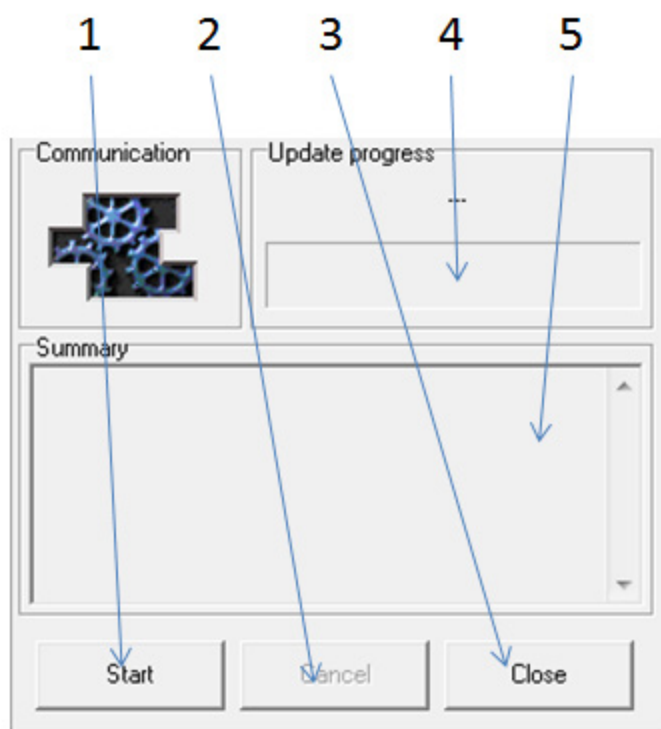
№	Описание
1	В этом поле отображается состояние загруженного ПО. Если файл встроенного ПО не загружен или не соответствует подключенному устройству, индикатор рядом с полем имеет красный цвет. Обновление встроенного ПО в этом случае невозможно.
2	В этом поле отображается номер версии загруженного встроенного ПО.
3	В этом поле отображаются семейства устройств, поддерживаемые загруженным встроенным ПО.
4	Нажатие на кнопку «...» позволяет открыть диалог выбора файла, в котором пользователь может указать нужный файл. Имя файла принимается после нажатия «Open», файл сохраняется в файле конфигурации программы.



### 4. Обновление встроенного ПО

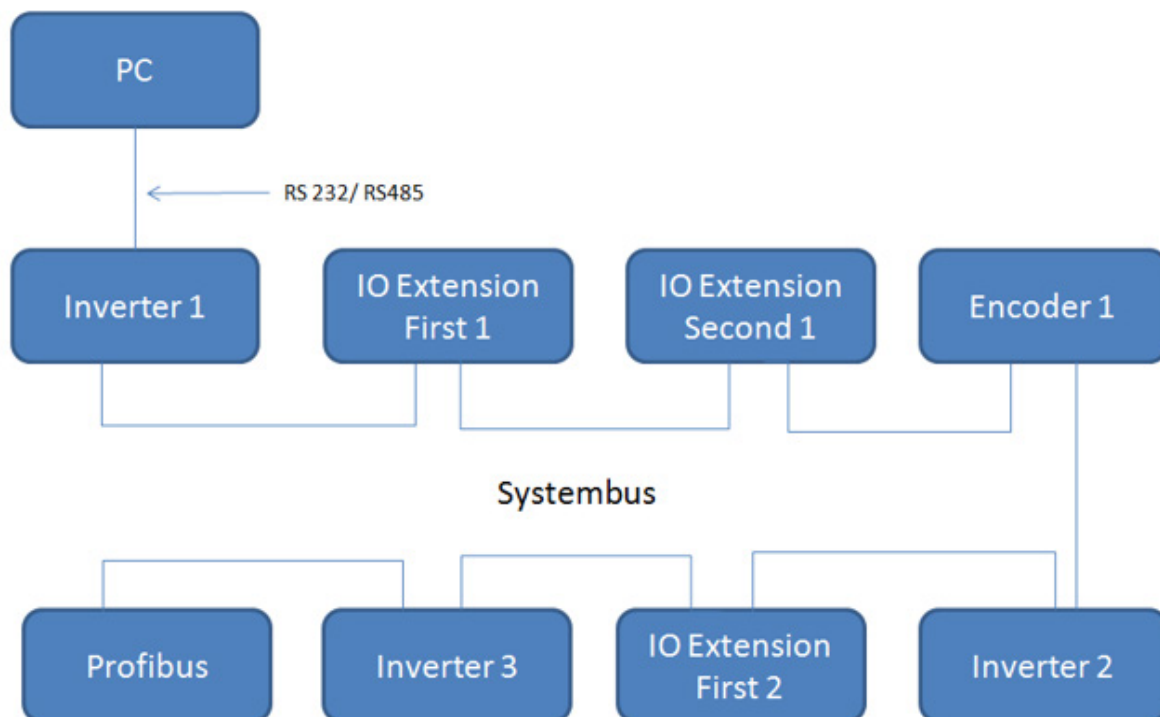
№	Описание
1	Чтобы запустить обновление встроенного ПО, нажмите кнопку «Start». Если кнопка недоступна, выбранное ПО нельзя загрузить в устройство.
2	Кнопка «Cancel» прерывает и отменяет запущенное обновление. Отмена возможна только в процессе инициализации.

№	Описание
3	Окно загрузки нельзя закрыть во время обновления. Пользователь может завершить обновление до загрузки или после, нажав кнопку «Close».
4	В панели хода выполнения отображается процесс обновления и его текущее состояние.
5	В поле «Summary» отображается результат обновления.



### 12.3 Обновление встроенного ПО через системную шину

Системная шина — шина на базе CAN, разработанная NORD. Шина поддерживается всеми устройствами SK2xxE и SK5xxE со встроенным интерфейсом CAN, а также разными видами дополнительного оборудования. Шина может одновременно обслуживать четыре преобразователя частоты, оснащенных двумя дополнительными устройствами и одним энкодером CANopen, и одно независимое устройство. Таким образом, к одной системной шине может быть подключено 17 устройств. Системная шина использует протокол, соответствующий CANopen. Каждое устройство имеет в системе CAN фиксированный адрес, который нельзя изменить произвольным образом.



При наличии абонента с интерфейсом RS 232/RS485 данные устройств NORD, подключенных к системной шине, могут выводиться и меняться через интерфейс NORD CON. Таким образом производится туннелирование запросов через устройство, подключенное к NORD CON, или встроенное ПО. Для туннелирования команд используется следующий порядок преобразования:

Адрес USS	Модуль
0	Этот адрес необходимо задать устройству, подключенному к NORD CON.
1	Преобразователь частоты 1 (CAN-ID: 32)
2	Преобразователь частоты 2 (CAN-ID: 34)
3	Преобразователь частоты 3 (CAN-ID: 36)
4	Преобразователь частоты 4 (CAN-ID: 38)
10	Дополнительное устройство 1 для преобразователя частоты 1 (модуль расширения)
11	Дополнительное устройство 1 для преобразователя частоты 2 (модуль расширения)
12	Дополнительное устройство 1 для преобразователя частоты 3 (модуль расширения)
13	Дополнительное устройство 1 для преобразователя частоты 4 (модуль расширения)
19	Устройство, у которого было прервано обновление встроенного ПО
20	Дополнительное устройство 2 для преобразователя частоты 1 (модуль расширения)



Адрес USS	Модуль
21	Дополнительное устройство 2 для преобразователя частоты 2 (модуль расширения)
22	Дополнительное устройство 2 для преобразователя частоты 3 (модуль расширения)
23	Дополнительное устройство 2 для преобразователя частоты 4 (модуль расширения)
30	Тип оборудования

Следующие устройства поддерживают туннелирование обновления для следующих устройств:

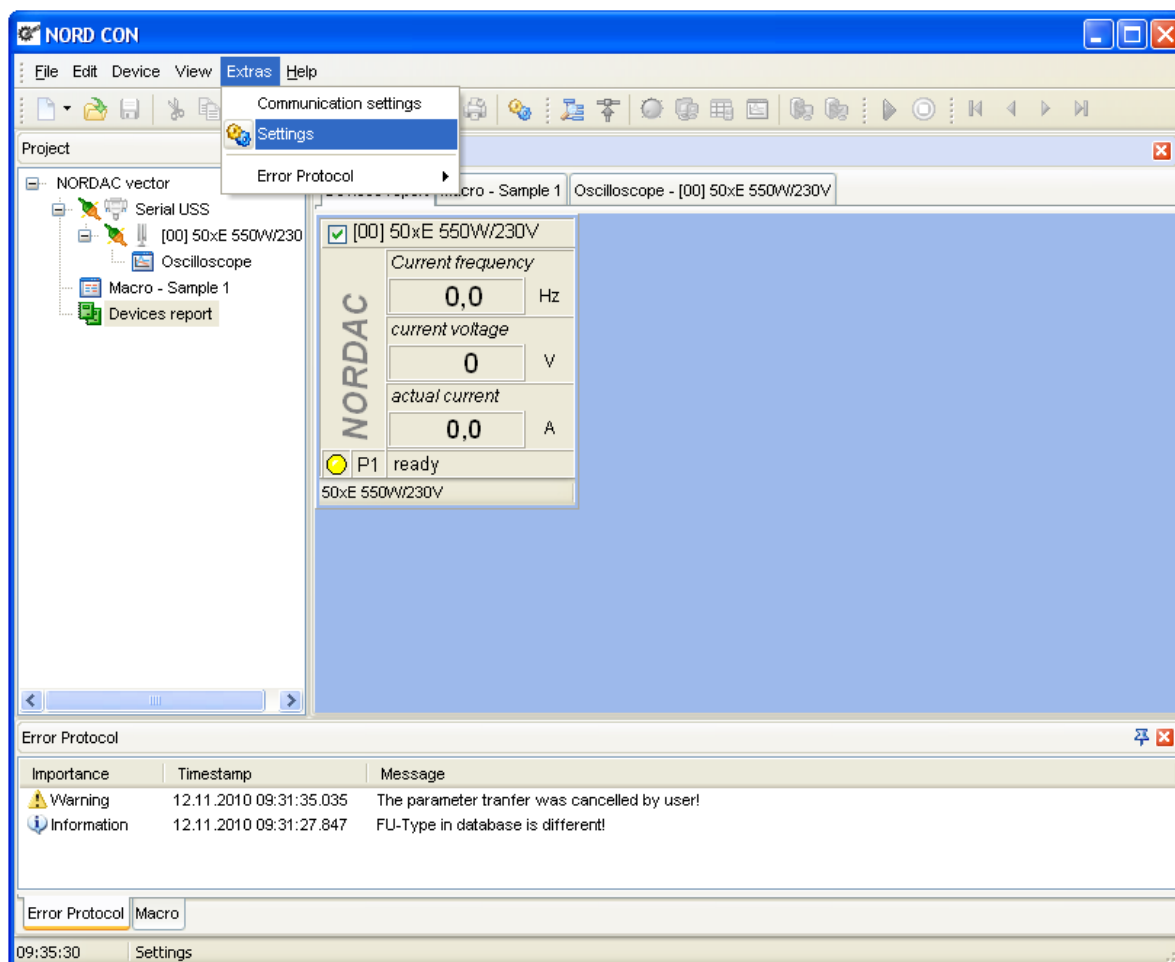
Устройство	Версия
SK 1xxE	все
SK 2xxE	с версии V1.3
SK 5xxE	с версии V2.0
SK 540E	с версии V2.0
SK TU4-DEV	с версии V1.4
SK TU4-CAO	с версии V2.2
SK TU4-PBR	с версии V1.2
SK TU4-POL	все
SK TU4-PNT	все
SK TU4-IOE	с версии V1.2
SK TU4-EIP	все

Программное обеспечение следующих устройств может обновляться через системную шину:

Устройство	Версия
SK 1xxE	все
SK 540E	с версии V2.0
SK TU4-DEV, SK CU4-DEV	с версии V1.4
SK TU4-CAO, SK CU4-CAO	с версии V2.2
SK TU4-PBR, SK CU4-PBR	с версии V1.2
SK TU4-POL, SK CU4-POL	все
SK TU4-PNT, SK CU4-PNT	все
SK TU4-IOE, SK CU4-IOE	с версии V1.2
SK TU4-EIP, SK CU4-EIP	все

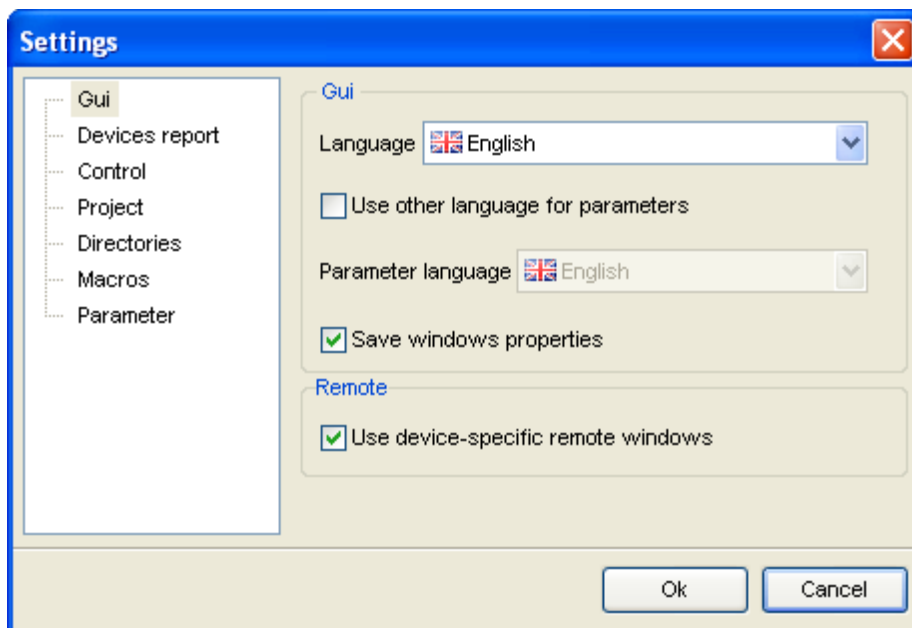
## 13 Settings

С помощью параметра «Extras->Settings» производится настройка программы <%PROGRAMNAME%>. Настройки разделены по следующим категориям:



### 13.1 Окно программы

В этом разделе пользователь может изменить настройки интерфейса.



### Language

Выбор языка интерфейса.

### Use other language for parameter setting

Если включена эта опция, пользователь может выбрать в списке «Parameter language» язык, который будет использоваться только для работы с именами параметров в окне «Parameterisation».

### Parameter language

Эта опция позволяет выбрать язык, используемый для названий параметров в окне «Parameterisation». Язык можно выбрать, если включена опция «Use other language for parameter setting». Если опция отключена, используется язык интерфейса.

### Save window setting

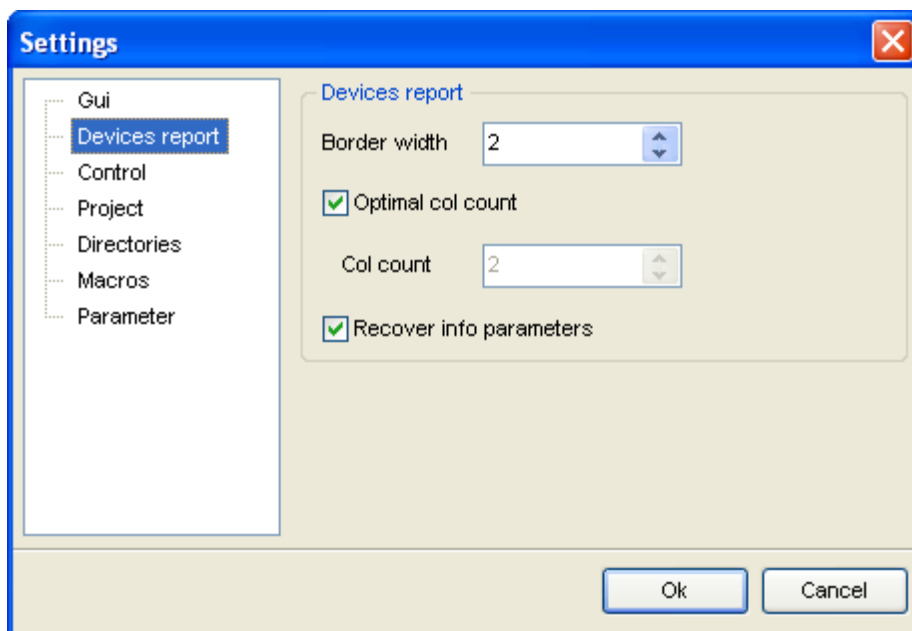
Если параметр включен, настройки окна (положение или размер) сохраняются при выходе и используются при следующем открытии.

### Use device-specific remote windows

Если включен этот параметр, для каждого типа устройства выводится специальное окно «Remote». Если параметр выключен, выводится стандартное окно.

## 13.2 Device overview

В этом разделе можно изменить настройки окна «Device overview».



### Border width

Этот параметр позволяет изменить ширину рамки в окне устройства. Допустимые значения: от 0 до 10 пикселей. Если вводится значение меньше минимального или больше максимального, автоматически используется минимальное или максимальное значение.

### Optimal number of columns

Если выбрана эта опция, приложение рассчитывает оптимальное число столбцов в зависимости от ширины окна и числа устройств.

### Number of columns

В случае использования этого параметра число столбцов остается всегда неизменным. Допустимое значение: от 1 до 10. Если вводится значение меньше минимального или больше максимального, автоматически используется минимальное или максимальное значение.

### Внимание:

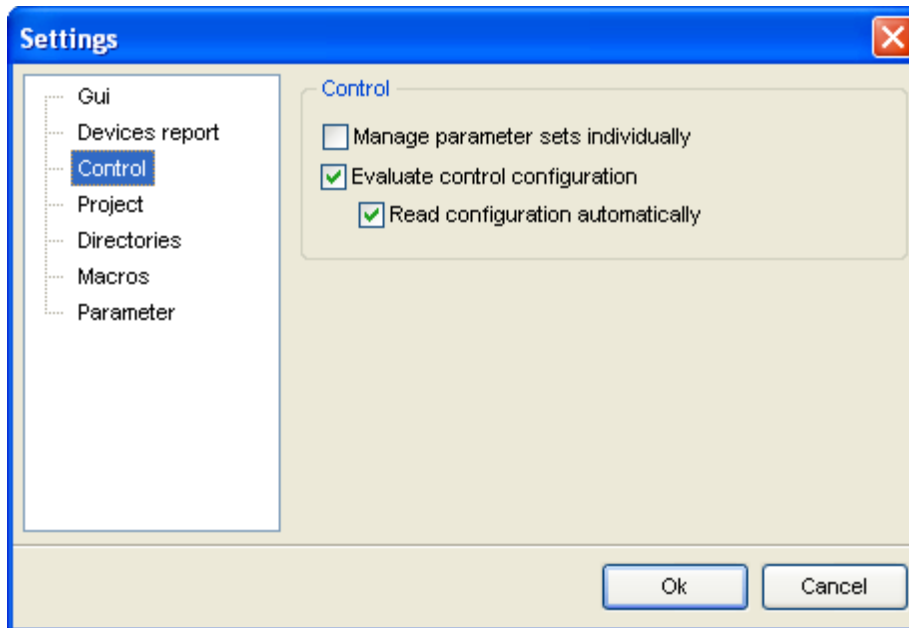
Этот параметр доступен, если не используется параметр «Optimal number of columns».

### Recover info parameters

Если выбрана эта опция, заданный параметр, представляющий некоторую информацию с устройства, сохраняется и восстанавливается после сканирования сети или перезапуска приложения.

## 13.3 Control

В этом разделе можно изменить специальные настройки формы 5 "Управление".



### Manage parameter sets individually

Если выбрана эта настройка, управление расчетным значением в окне «Control» производится независимо от действительного.

### Evaluate control configuration

Эта настройка включает (или отключает) анализ управляющей конфигурации. При выборе этой настройки после прочтения конфигурации некоторые функции активируются или отключаются. Кроме того, названия функций для параметризации расчетного или действительного значений отображаются в виде текстовой строки.

### Read configuration automatically

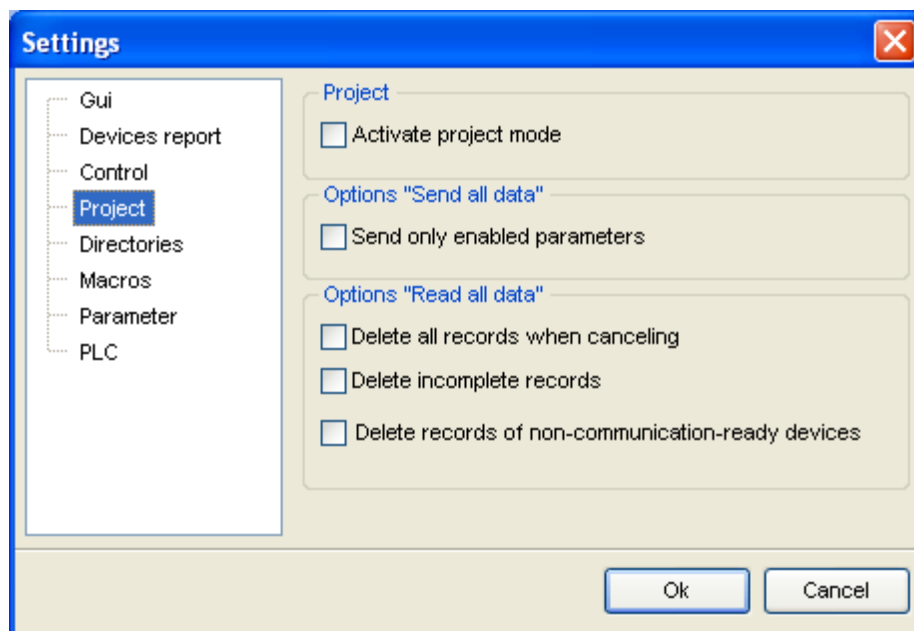
Эта настройка включает (или отключает) автоматическое считывание конфигурации. Если выбрана эта настройка, после фокусирования окна производится считывание и анализ управляющей конфигурации.

### Примечание:

Функция «Evaluate control configuration» не поддерживается в некоторых устройствах!

## 13.4 Project

В этом разделе можно указать путь к файлу проекта. В файле проекта сохраняются такие настройки, как используемые интерфейсы, настройки сканирования шины, названия устройств и т.д. После выбора файла загружаются все сохраненные в нем настройки.



### Activate project mode

Эта опция включает и отключает режим проекта. В режиме проекта пользователь может указать тип и число устройств, подключенных к шине, а также сохранить параметры устройства и настройки приложения в файле проекта.

### Only transfer enabled parameters

Если активна настройка «Send all Data», на устройство передаются только параметры, которые были разблокированы пользователем. По умолчанию на устройство всегда передаются все параметры. Разблокировка параметров производится в редакторе параметров.

### Delete all data records on cancellation

Если включен этот параметр, при отмене операции «Read all Data» из проекта удаляются все наборы параметров.

### Delete incomplete data records

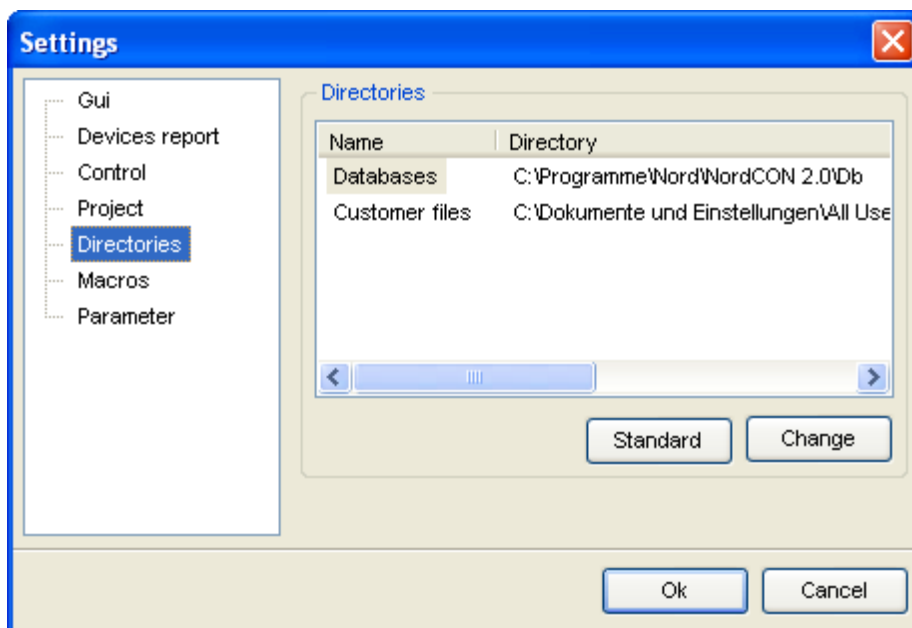
Если включен этот параметр, при возникновении ошибки во время выполнения операции «Read all Data» набор данных устройства удаляется.

### Delete data records from devices not ready for communication

Если при выполнении операции «Read all Data» не удалось получить данные с устройства, набор данных устройства будет удален.

## 13.5 Directories

В этом разделе можно указать каталоги, в которых хранятся базы данных параметров, файлы конфигураций и макросов, а также внутренние базы данных. Чтобы изменить путь, необходимо выделить нужный каталог в списке, после этого нажать на кнопку «Change» и задать новый путь. С помощью кнопки «Standard» можно указать для каждой категории путь по умолчанию.



### Customer files

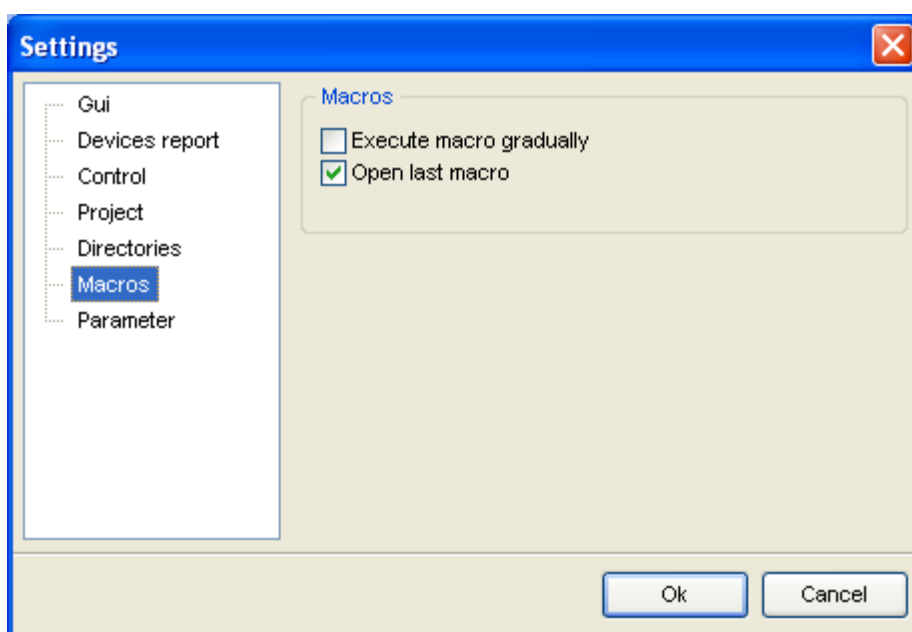
В этом каталоге хранятся пользовательские файлы, например, файлы макросов или параметров.

### Internal databases

Эти базы данных используются при выполнении программой внутренних процессов. В них хранится структура параметров разных типов оборудования.

## 13.6 Редактор макросов

В этом разделе можно изменить настройки редактора 8 "Редактор макросов".



### Macro execution step by step

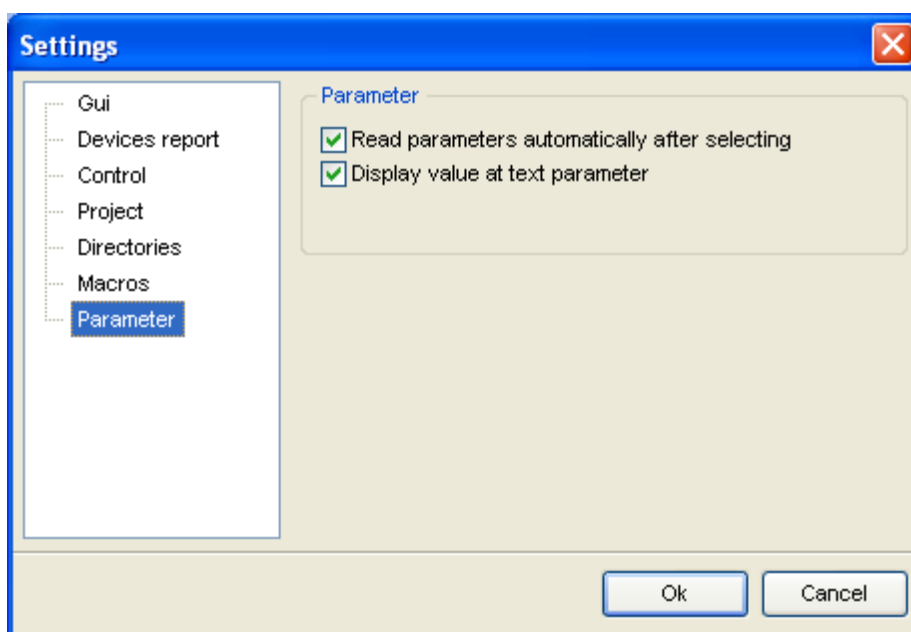
Эта настройка включает (или отключает) пошаговое выполнение макроса. Если настройка включена, каждый оператор макроса должен запускаться отдельно («Ablauf/Start»).

### Open last macro

Если включена эта опция, при открытии генератора макроса загружается макрос, который использовался в последний раз.

## 13.7 Parameter

В этом разделе можно изменить настройки окна 4 "Параметризация".



### Read parameter automatically after selection

Если настройка включена, при выборе параметра производится автоматическое считывание его значений.

### Show also the value with text parameter

Если настройка включена, в окне «Parameter» вместе с текстовой информацией выводится числовое значение параметра.

## 13.8 ПЛК

### Delete old log entries before compiling

Если включена эта настройка, в журнале перед каждой компиляцией удаляются старые записи.

### Jump to current breakpoint (debug mode)

Если включена эта настройка, область просмотра перемещается в текущую точку останова.



## 14 Сообщения

### 14.1 Ошибки и указания

Сообщения об ошибках и указаниях включают описание (текст) и номер ошибки.

Возможен вывод следующих сообщений:

Номер	Описание
100	Недопустимый № параметра
101	Изменение значения параметра невозможно
102	Нарушение предельных значений параметров
103	Недопустимый индекс SUB
104	Отсутствие параметров массива
105	Изменение описания невозможно
106	Описание отсутствует
107	Время ожидания приема
108	Время ожидания передачи
109	Принятые данные содержат ошибки
110	Ответ и задание различаются
200	Ошибка: Открытие серийного интерфейса!
201	Ошибка: Закрытие серийного интерфейса!
202	Сначала закрыть старый интерфейс!!
203	Интерфейс не открыт!
204	Не удалось задать настройки режима коммуникации. Проверьте, поддерживается ли текущая скорость передачи данных.
205	Ошибка буферной памяти!
206	Ошибка настройки времени ожидания!
207	Коммуникация невозможна!
208	Внутренняя ошибка объекта!
210	Ошибка записи файла!
211	Не удалось создать телеграмму!
212	Не удалось найти таймер высокого разрешения!
213	Устройство не найдено!
214	Возможно только при значении уставки 16 бит!

Номер	Описание
215	ПЧ в работе. Закрыть окно?
216	Обновление программного обеспечения возможно только на устройствах с адресом 0.
217	Не удалось запустить программу для обновления программного прошивки! Установите заново NORD CON, чтобы устранить проблему.
218	Сначала добавьте модуль коммуникации!
219	Импортировать файл для просмотра в режиме онлайн?
220	Здесь добавить устройство нельзя!
221	Найдено более 1 устройства, подключенного к шине. Обновление может стать причиной возникновения проблем. Все равно продолжить?
222	Если одновременно использовать макропрограммы и окно управления, может возникнуть несовместимость данных. Закройте все окна управления или редактор макропрограмм.
223	Невозможно начать передачу данных, т.к. открыт редактор параметров! Закройте редактор и заново запустите функцию.
224	Не удалось найти онлайн-справку! Установите заново NORD CON, чтобы устранить проблему.
225	Отсоединить устройство нельзя, т.к. Открыто, по меньшей мере, еще одно окно.
226	Невозможно открыть файл. Неизвестный формат файла.
227	Невозможно прочитать файл!
228	Неизвестный формат файла!
229	Файл изменен пользователем!
230	Действие невозможно выполнить, т.к. устройство не подсоединено!
231	Настройки были изменены. Сохранить изменения?
232	Ваш компьютер не совместим с китайским алфавитом, поэтому могут возникнуть ошибки отображения!
233	Невозможно преобразовать значение в INT16!
234	Текущая версия устройства не поддерживает обновление программной прошивки через системную шину!
235	Текущая версия устройства технологического модуля не поддерживает обновление программной прошивки через системную шину!
236	Устройство по адресу 0 не поддерживает обновление программной прошивки через системную шину!
237	Этот ПЛК не зарегистрирован! Обратитесь в службу техподдержки (+49 (0)180 500 61 84).
238	Неверный код регистрации! Обратитесь в службу техподдержки (+49 (0)180 500 61 84).
239	Загрузка встроенного ПО может выполняться только при скорости 38400 бод!
240	Невозможно распечатать отчет, т.к. принтер не установлен!
241	Не удалось найти файл в вашей системе!
242	Текущая версия устройства технологического модуля TU3 не поддерживает обновление

Номер	Описание
	встроенной прошивки ПО!
243	Добавить еще одно устройство невозможно!
244	Проект изменился! Сохранить проект?
245	Невозможно установить связь с устройством %s!
246	Не удалось найти программу ПЛК для устройства %s!
247	Не удалось найти параметры для устройства %s!
248	Передача проекта прекращена пользователем!
249	Во время передачи проекта возникла, по меньшей мере, одна ошибка!
250	Во время передачи проекта возникло, по меньшей мере, одно предупреждение!
251	Введенный IP-адрес недействителен!
252	Добавить еще одно устройство невозможно!
253	Файл поврежден или подвержен манипуляциям!
254	Обновление встроенного ПО в режиме "USS через TCP" невозможно!
255	Изменения требуют сканирования шины! Применить изменения?
256	Не удалось найти файл проекта!
257	Сначала вставьте шинный расширенный модуль!
258	На выбранный шинный расширенный модуль могут быть переданы не все настройки! Продолжить?
259	В процессе записи возникла ошибка!
260	Программа ПЛК не совместима с устройством %s!
261	Файл проекта еще не создан! Сохранить проект прямо сейчас?
262	Введенный IP-адрес уже используется!
263	Не удалось найти каталог!
264	Не удалось преобразовать текст в массив байтов!
265	Неверная телеграмма USS!
266	Программу ПЛК можно загрузить только к конкретному устройству!
267	Перезаписать существующую программу ПЛК?
268	Программа ПЛК защищена и ее можно загрузить только к конкретному устройству!
269	Необходимо изменить вид коммуникации. Все устройства из списка будут заменены! Продолжить?
300	Необходимо исправить путь к внутренней базе!
301	Неправильный путь к внутренней базе данных. NORD CON завершается
302	Ошибка при открытии баз данных!

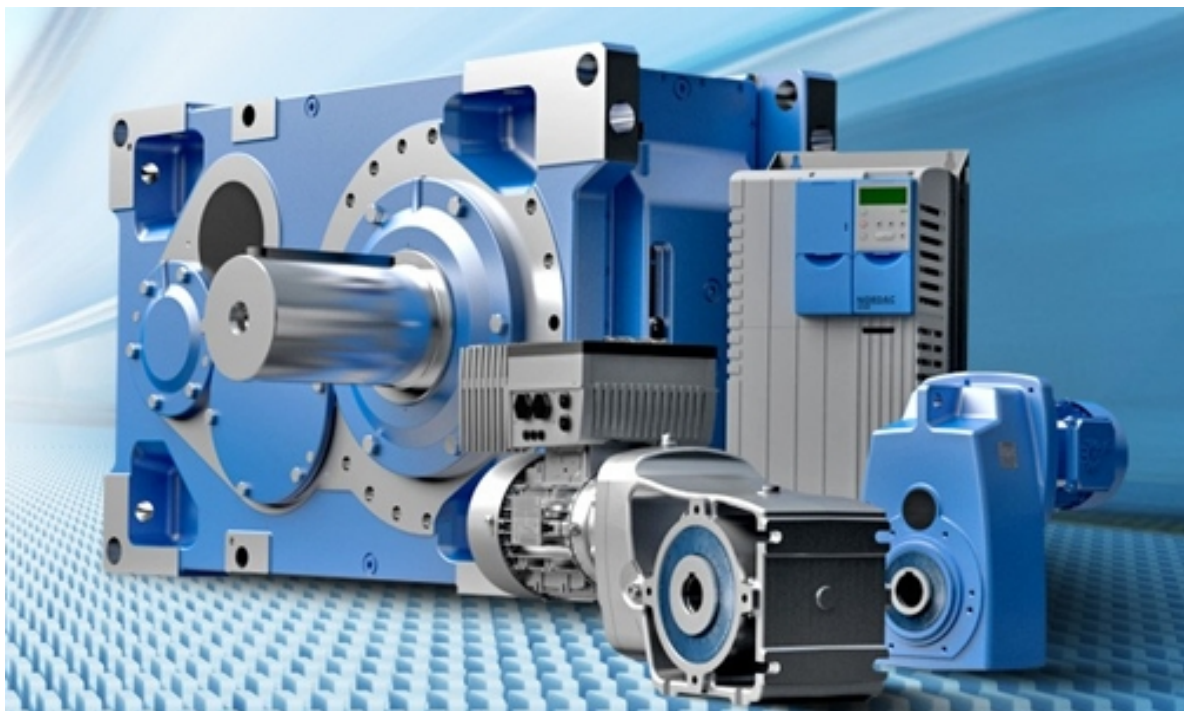
Номер	Описание
303	Несовместимый тип ПЧ в базе данных!
304	В базе данных другой тип ПЧ!
305	Сохранить текущую базу данных?
306	Невозможно открыть базу данных!
307	Недопустимый путь!
308	Невозможно сохранить базу данных!
309	Вывести сразу все параметры?
310	Обновите NORD CON! Безупречное параметрирование не гарантируется.
311	Неправильно установлен принтер!
312	В настоящее время разрешено только 1 окно параметров. Показать открытое окно?
313	Для выхода из программы необходимо закрыть окно параметров!
314	Для выполнения сканирования шины необходимо закрыть окно параметров!
315	Сравнение параметров может быть сохранено только в формате PDF.
316	Параметры еще не сохранены на устройстве на постоянной основе. Все равно закрыть?
317	Начальный адрес не должен быть длиннее конечного!
318	Не все i-параметры актуальны. Выполните команду «Alles lesen» (Прочитать все).
319	Еще не все измененные значения переданы!
320	Не все i-параметры выбраны. Измените фильтр!
400	Не удалось загрузить файл, т.к. версия файла неизвестна!
401	Не удалось загрузить файл, т.к. формат файла неизвестен!
402	Файл изменен пользователем!
403	Ошибка при открытии файла!
405	Файл с макрокомандами отсутствует!
406	Список макрокоманд пуст!
407	Выполняется список макрокоманд!
408	Цель перехода не найдена!
409	Функцию невозможно выполнить, т.к. запущен планировщик!
410	Сохранить изменения в макрокоманде?
411	Файл изменен пользователем! Открыть файл?
500	Загрузить только настройки?
501	Типы устройств отличаются? Открыть файл?
502	Не удалось открыть файл, т.к. версия формата файла неизвестна!
503	Не удалось открыть файл, т.к. формат файла неизвестен!

Номер	Описание
504	Файл изменен пользователем! Открыть файл?
600	Управление устройством ограничено или невозможно по следующим причинам: в конфигурации не задано управляющее слово (P509) для USS!
601	Управление устройством ограничено или невозможно по следующим причинам: в конфигурации не задан источник уставок 1 (P510.0) для USS!
602	Управление устройством ограничено или невозможно по следующим причинам: в конфигурации не задан источник уставок 2 (P510.1) для USS!
603	Управление устройством ограничено или невозможно по следующим причинам: в конфигурации не задано управляющее слово (P509) и источник уставок 1 (P510.0) для USS!
604	Управление устройством ограничено или невозможно по следующим причинам: в конфигурации не задано управляющее слово (P509) и источник уставок 2 (P510.1) для USS!
605	Управление устройством ограничено или невозможно по следующим причинам: в конфигурации не задан управляющее слово (P509) и источник уставок 2 (P510.1) для источник уставок 1 (P510.0) и 2 (P510.1) для USS!
606	Управление устройством ограничено или невозможно по следующим причинам: в конфигурации не задано управляющее слово (P509), источник уставок 1 (P510.0) и 2 (P510.1) для USS!
700	Действие не может быть выполнено, т.к. нарушена связь с устройством!
701	Действие невозможно выполнить, т.к. доступ заблокирован!
800	Действие "Передать параметры" успешно выполнено.
801	Во время выполнения действия "Передать параметры" возникла ошибка!
802	Действие "Передать параметры" отменено пользователем!
803	Во время выполнения действия "Передать параметры" возникла ошибка! Сохранить?
804	Действие "Передать параметры" отменено пользователем! Сохранить?
805	Обнаружены различия! Просмотреть отчет?
806	Создание отчета отменено пользователем!
807	Идет повторная установка связи с устройством! Продолжить?
808	Один параметр отсутствует!
809	Значение параметра выше максимально допустимого!
810	Значение параметра ниже минимально допустимого!
811	Во время импорта данных электродвигателя возникла ошибка!
900	В список отслеживания можно внести не более 5 переменных!
901	Перед тем, как файл можно будет переводить, его следует сохранить. Создать новый файл?
902	Не удалось открыть файл, т.к. формат файла неизвестен!
903	Невозможно прочитать файл!
904	Файл изменен пользователем! Открыть файл?
905	Функция не реализована!

Номер	Описание
906	Перед программированием необходимо сохранить программу ПЛК!
907	Программа ПЛК изменена! Сохранить?
908	Настройки изменились! Сохранить?
909	Формат ПЛК 1.0 не поддерживается.
910	Не удалось сохранить программу ПЛК!
1100	Все наборы данных удалены (отмена пользователем)!
1101	Неполный набор данных для устройства удален!
1102	Неполный набор данных для устройства сохранен.
1103	Набор данных устройства, не готового к коммуникации, удален.
1104	Параметры для устройства не сохранены.

## 15 Getriebebau Nord

Getriebebau NORD предлагает широкий спектр механического и электрического оборудования для приводной техники по принципу «все из одних рук». Имея представительства в 48 странах, мы можем предложить приводную технику следующих видов: цилиндрические, червячные, конические, плоские и планетарные редукторы в диапазоне мощностей от 12 кВт до 200 кВт с моментами вращения от 10 Нм до 60 000 Нм.



### Полный спектр услуг

Наше главное производство расположилось в г. Баргтехайде под Гамбургом, однако компания имеет обширную сеть сборочных производств и сервисных предприятий, которая охватывает более 60 стран. В компании работает более 3100 человек, усилиями которым мы обеспечиваем кратчайшие сроки поставок и максимальное качество обслуживания во всех уголках земного шара. Мы готовы помочь в любой ситуации и в любое время.

### Мы — источник движения

С помощью нашей мощной приводной техники мы заставляем работать «голиафов» мира техники — огромные портовые краны, механизмы раздвижных крыш современных стадионов, багажные ленты в аэропортах, горнолыжные фуникулеры и канатные дороги. NORD — это источник движения, заставляющий работать технику в любых условиях.

### Двигатель прогресса

Наши изделия — это синтез компактных механических конструкций и «умной» электроники. Мы проектируем, производим и продаем полный ассортимент механического и электронного приводного оборудования с оптимальным соотношением «цена-качество». Мы предлагаем



высококачественные редукторы, мотор-редукторы, преобразователи частоты, серворегуляторы и периферийную технику.

### Приводить все в движение совместными усилиями

Наши высокие стандарты качества и обслуживания — это результат непрерывного развития компании в направлении, которое диктуется интересами клиентов. Мы разрабатываем нашу продукцию в тесном взаимодействии со специалистами, использующими нашу технику. Перед нами не стоит цель найти новаторские или творческие идеи. Взаимодействуя с клиентами не только в конце цепочки создания ценности, но и в самом ее начале, мы разрабатываем приводные решения, которые действительно нужны.

## 15.1 История предприятия

С момента основания в 1965 году компания NORD следует политике, ориентированной на клиента.



- 1977 Строительство современного производства зубчатых колес
- 1979 Создание дочерних предприятий по всему миру
- 1980 Реализация концепции модульной конструкции во всей линейке редукторов
- 1983 Открытие собственного производства электродвигателей
- 1985 Открытие производства преобразователей частоты
- 1992 Открытие сталелитейного производства в г. Гадебуш



- 1997 Открытие завода по изготовлению двигателей в Италии
- 1998 Запуск сборочного производства во Франции



- 2000 Запуск сборочных производств в Великобритании и Австрии
- 2001 Открытие сборочного производства в Китае
- 2002 Завершение третьей очереди завода в г. Гадебуш (ок. 7200 м²)
- 2003 Открытие дочернего предприятия в России
- 2004 Строительство завода по выпуску двигателей в Италии
- 2005 40 лет компании Getriebebau NORD. Ввод в эксплуатацию высотного склада в г. Баргтехайде. Строительство нового завода по выпуску двигателей в Сучжоу (Китай)
- 2006 Открытие нового производства электронного оборудования в г. Аурих (Германия)
- 2007 Открытие сборочных центров в Индии и Чехословакии
- 2008 Обновление здания в г. Баргтехайде:  
- сооружение многоэтажного парковочного центра



- Обновление здания в г. Баргтехайде:  
2009 - сооружение монтажного центра для промышленных редукторов  
- увеличение площадей высотного склада в два раза  
NORD DRIVESYSTEMS вводит четвертую очередь строящегося завода в г. Гадебуш и
- 2011 празднует 25 годовщину компании NORD GEAR Ltd. (г. Брамтон, Канада). Компания NORD открывает в Китае второй завод (г. Тяньцзинь, примерно в 100 км к юго-востоку от Пекина); в Австралии в г. Дарвине открыта дочерняя компания NORD Australia.  
В настоящее время компанию NORD DRIVESYSTEMS в разных странах мира представляют 35 дочерних компаний. Глобальная сеть партнеров NORD по сбыту и сервисному обслуживанию теперь охватывает более чем 60 стран. Благодаря команде высокомотивированных сотрудников и широчайшему ассортименту первоклассной приводной техники, отвечающей самым высоким технологическим стандартам, компания готова ответить на любой вызов, который готовит нам будущее.
- 2012
- 2013 Создание дополнительных производственных мощностей в г. Сучжоу
- 2014 Модернизация системы обслуживания. Обновление лакокрасочного производства в г. Баргтехайде.



- 2015
- 50-летний юбилей компании
  - Открытие нового административного здания в г. Баргтехайде

## 15.2 Преобразователи частоты

### 15.2.1 SK 135E



**SK 135E – пускатель двигателя с мощностью от 0,25 до 7,7 кВт для децентрализованных систем**

Электронный пуск и останов приводов востребован во многих областях применения, в т.ч. в подъемно-транспортном оборудовании. Для этих целей могут использоваться универсальные

пускатели электродвигателей SK 135E. Благодаря функции плавного пуска и возможности реверса они находят широчайшее применения в системах управления приводом. Пускатели также выполняют разные функции контроля, защищая, например, от перегрева. При использовании отключения по характеристике I<sup>2</sup>t допускается эксплуатация двигателя без дополнительного теплового реле. Благодаря наличию встроенного фильтра пускатель SK 135E, установленный на двигатель, обеспечивает электромагнитную совместимость, отвечающую самым строгим стандартам.

#### Свойства и характеристики

- Функция плавного пуска
- Функция реверса
- Установка на двигатель или стену
- IP55 (возможность увеличения до IP66)
- Диапазон мощностей: от 0,25 до 4,0 кВт для устройств 3~ 200 ... 240 В, от 0,25 до 7,5 кВт для устройств 3~ 380 ... 500 В
- Управление и подсоединение электромеханического тормоза
- Встроенный сетевой фильтр (класс ЭМС C1/C2)
- 2 цифровых входа
- 2 цифровых выхода
- Вход для датчика температуры (TF+/TF-)
- Интерфейс RS232 (разъем RJ12)
- Дополнительно: допуск АTEX для зоны 22 3D (разрабатывается)

Подробнее о [пускателях электродвигателей SK 135E можно узнать здесь](#).

#### 15.2.2 SK 180E



SK 180E – экономичные преобразователи частоты мощностью от 0,25 до 2,2 кВт для децентрализованных систем

SK 180E — оптимальный выбор в системах низкой мощности, предъявляющих высокие требования к регулированию частоты вращения. Благодаря векторному режиму управления, частотные преобразователи NORD широко применяются в таких системах. О высоком качестве

устройств типа SK 180E свидетельствует высочайший среди преобразователей этого типа класс ЭМС. Благодаря встроенному сетевому фильтру преобразователи этой серии, установленные на двигателе, могут использоваться даже в жилых помещениях (класс С1).

- Свойства и характеристики
- Эффективный преобразователь частоты для простых децентрализованных систем с регулированием частоты вращения
- Установка на двигатель или стену
- IP55 (возможность увеличения до IP66)
- Доступный диапазон мощностей (TP I): от 0,25 до 0,55 кВт для устройств 1~ / 3~ 200 ... 240 В, от 0,25 до 1,1 кВт для устройств 3~ 380 ... 480 В
- Встроенный сетевой фильтр (класс ЭМС C1/C2)
- 2 аналоговых входа
- 3 цифровых входа
- 2 цифровых выхода
- Вход для датчика температуры (TF+/TF-)
- Интерфейс RS232/RS485 через разъем RJ12
- Разрешен для бытового применения (исполнение для установки на двигатель)
- Возможность подключения к бытовой розетке
- Дополнительно: допуск АTEX для зоны 22 3D

### 15.2.3 SK 200E



#### **Универсальные преобразователи частоты мощностью от 0,25 до 22,0 кВт для децентрализованных систем**

Руководствуясь многолетним опытом в сфере преобразователей частоты, устанавливаемых на двигатель, компания NORD выпустила на рынок устройства серии SK 200E, которые позволяют решать широкий круг задач в приводном оборудовании. Эти прочные, надежные и экономичные в эксплуатации устройства подходят для самых разных установок, в том числе и для конвейерных линий, и являются оптимальным выбором в ситуациях, когда цена является немаловажным фактором. Как и серия SK 500E, предназначенная для распределительных шкафов, SK 200E включает устройства с разной производительностью и одинаковым набором базовых функций. Преобразователи SK 200 E позволяют быстро создавать распределенные приводные системы в производственных условиях, так как обладают всеми преимуществами, характерными для систем с децентрализованной архитектурой: надежность, возможность подключения к имеющимся системным разъемам, быстрая замена, дооснащение модулями расширения и обмена данными.

#### **Комплектация преобразователей частоты SK 200E:**

1~ 115 В 0,25 - 0,75 кВт

1~ 230 В 0,25 - 1,1 кВт

3~ 230 В 0,25 - 11,0 кВт

3~ 480 В 0,55 - 22,0 кВт

- С креплением на стену
- Дополнительные модули (в том числе с функцией шлюза)

**В стандартной комплектации устройства имеют класс защиты IP55. Дополнительное оснащение:**

- TP1 - 3: IP66 (устройства с маркировкой «С»= coated/с покрытием)
- TP: устройства с маркировкой «С»= coated/с покрытием с сохранением класса защиты IP55
- TP1 - 3: взрывоопасные среды (ATEX зона 22, 3D) или неблагоприятные условия

#### **Базовая конфигурация преобразователей SK 200E / SK 205E**

- Высокое качество регулирования (бездатчиковое регулирования вектора тока ISD)
- Встроенный сетевой фильтр 24 В (SK 200E)
- Внешний источник питания 24В для дополнительных модулей (SK 200E TP4 и SK 205E)
- 4 управляющих входа, возможность параметризации для использования разных цифровых функций
- 1 или 2 аналоговых входа (SK 200E)
- Встроенный механизм управления торможением для механических стояночных тормозов (SK 200E TP4 и SK 205E)
- Внешние индикаторы состояния (состояние сигнала на управляющих входах) (SK 200E TP4 и SK 205E)
- 2 потенциометра расчетных значения в внешних регуляторах (SK 200E TP4 и SK 205E)
- Съёмный модуль памяти (EEPROM)
- Автоматическая идентификация параметров двигателя
- 4 набора параметров, переключаемых при наличии сети
- Анализ данных инкрементного энкодера (HTL)
- Работа в 4 квадрантах при наличии дополнительного тормозного резистора
- ПИ-регулятор и процессный регулятор
- Диагностический интерфейс RS 232 (RJ12)
- Функция потенциометра двигателя
- Система управления позиционированием POSICON

#### **Преобразователи частоты SK 210E / SK 215E**

- Базовое оснащение аналогично моделям SK 200E / SK 205E
- Защитная функция «Безопасный останов» в соответствии с EN 954-1 (EN 13849-1) до кат.4, категория останова 0 и 1

#### **Преобразователи частоты SK 220E / SK 225E**

- Базовое оснащение аналогично моделям SK 200E / SK 205E
- Встроенный интерфейс AS

#### **Преобразователи частоты SK 230E / SK 235E**

- Базовое оснащение аналогично моделям SK 200E / SK 205E
- Защитная функция «Безопасный останов» в соответствии с EN 954-1 (EN 13849-1) до кат.4, категория останова 0 и 1
- Встроенный интерфейс AS



Подробное описание устройств: <http://www.sk200e.de>

#### 15.2.4 SK 500E



Мощность: 0,25 ... 2,2 кВт  
1/3 AC 200 ... 240 В  
3,0 ... 18,5 кВт  
3 AC 200 ... 240 В 0,55 ... 90 кВт  
3 AC 380 ... 480 В 0,25 ... 160 кВт

Выходная частота: 0 ... 400 Гц

Руководства Руководство по эксплуатации SK 5xxE  
Руководство по эксплуатации SK 54xE

## Предметный указатель

<b>B</b>		Messfunktion.....	61
Bus error .....	33	<b>N</b>	
<b>C</b>		Name.....	33
Connect .....	16, 83	New .....	13
Control .....	16, 83, 202, 204	NORD CON .....	9
Copy .....	14	<b>O</b>	
Cut .....	14	Open .....	13
<b>D</b>		<b>P</b>	
Delete .....	14	Parameter .....	39, 40, 202, 208
Device.....	16, 83	Parameter download to device .....	16, 83
Device overview .....	203	Parameter setup .....	16, 83
Directories.....	206	Parameter upload from device.....	42
Directories.....	202	Parameter upload from device.....	16
Down.....	14	Parameter upload from device.....	83
<b>E</b>		Paste .....	14
Edit.....	14	PLC	
Einstellungen .....	205	FB_FlyingSaw .....	97
Exit/Quit .....	13	Метка перехода .....	178
Export .....	13	Port.....	33
Extras.....	19	Print.....	13
<b>F</b>		Project.....	11, 12, 18, 22
File .....	13	Projekt.....	205
Filter.....	40	<b>R</b>	
<b>G</b>		Read configuration automatically.....	204
Gerät.....	21	Remote .....	12, 18, 26
<b>H</b>		Remote Control.....	16, 83
Help .....	19	Rename .....	16, 83
<b>I</b>		Replace .....	14
Import.....	13	<b>S</b>	
<b>L</b>		Save .....	13
Language.....	52, 202	Save as .....	13
Log.....	18	Save window setting .....	52, 202
<b>M</b>		Select all .....	14
Macro editor.....	202	Settings .....	19, 52, 202
Manage parameter sets.....	204	Simulate hardware .....	33
Master (запрос USS) .....	73	Start.....	22
		Steuerungskonfiguration auswerten .....	204

<b>T</b>	Обзор..... 9, 45
Telegramm error.....33	Обзор осциллографа ..... 60
Toolbar..... 18	Обновление встроенного ПО через системную шину..... 199
<b>U</b>	Обновление встроенного программного обеспечения ..... 194
Undo ..... 14	Окна и виды ..... 12
Up ..... 14	Окно журнала ..... 68
<b>V</b>	Окно переменных..... 65
View ..... 18	Окно программы ..... 202
View Remote .....12, 18, 26	Окно проекта ..... 11, 12, 18, 22
<b>В</b>	Окно свойств..... 65
Введение .....9	Осциллограф..... 60, 61, 63, 64
Вкладка .....41	загрузка данных..... 64
Восстановление параметров.....43	печать ..... 64
Вставка операторов.....69	сохранение..... 64
Вывод данных с помощью осциллографа.60	экспорт данных ..... 64
Выгрузка параметров .....43	Ответ USS..... 73, 76
Выполнение.....71	открепление..... 27
Выполнение макроса.....71	Открытие макроса..... 68
Выполнение следующего оператора .....72	Офлайн-параметризация ..... 40
Вырезание операторов .....69	Ошибки..... 209
<b>Г</b>	<b>П</b>
Генератор макросов .....9, 65, 207	Панель быстрого запуска ... 11, 12, 20, 21, 22
Главное меню..... 12, 13, 14, 16, 18, 19, 83	Панель инструментов ..... 11, 12, 20, 21, 22
Главное окно ..... 12	Панель управления..... 11, 12, 20, 21, 22
<b>Д</b>	Параметры..... 40, 41
Добавление нового оператора .....70	Первые шаги..... 9
Дополнительные функции .....46, 47	Передача данных..... 202
<b>З</b>	Перемещение оператора вверх..... 69
Запрос USS .....73, 75	Перемещение оператора вниз..... 69
<b>И</b>	Планировщик..... 71
Измерение с помощью осциллографа.....63	ПЛК ..... 78, 208
Интерфейс..... 189, 190	ABS ..... 141
История предприятия .....216	ACOS ..... 146
<b>К</b>	ADD ..... 141
Копирование операторов .....69	ADD(..... 141
<b>М</b>	AND ..... 148
Макрос .....9, 65, 207	AND(..... 148
<b>О</b>	ANDN ..... 149
О программе NORD CON.....9	ANDN(..... 149



ASIN.....	146	JMPCN .....	184
ATAN.....	146	LD.....	155
BOOL_TO_BYTE.....	185	LDN .....	155
BYTE_TO_BOOL.....	185	LE .....	158
BYTE_TO_INT.....	186	LIMIT .....	142
CASE .....	181	LN.....	147
ControlBox .....	82	LOG.....	147
ControlBox и ParameterBox .....	170	LT .....	158
COS .....	146	MAX.....	143
CTD.....	114	MC_MoveAbsolute .....	104
CTU.....	115	MC_WriteParameter_16 .....	113
CTUD .....	116	MC_WriteParameter_32 .....	113
DINT_TO_INT .....	186	MC_Control .....	100
DIV .....	142	MC_Control_MS.....	102
DIV(.....	142	MC_Home .....	103
EQ.....	156	MC_MoveAdditive .....	105
Exit .....	184	MC_MoveRelative .....	106
EXP .....	146	MC_MoveVelocity .....	107
F_TRIG .....	117	MC_Power.....	108
FB_FunctionCurve .....	135	MC_ReadActualPos .....	109
FB_PIDT1 .....	136	MC_ReadParameter .....	110
FB_ResetPosition .....	138	MC_ReadStatus .....	111
FB_Capture.....	131	MC_Reset .....	112
FB_DinCounter .....	134	MC_Stop .....	112
FB_DINTToPBOX.....	126	MIN.....	143
FB_Gearing.....	98	MOD .....	144
FB_NMT.....	90	MOD(.....	144
FB_PDOConfig .....	90	Motion Control Lib .....	81
FB_PDOReceive.....	92	MUL.....	144
FB_PDOSend .....	94	MUL(.....	144
FB_ReadTrace.....	123	MUX.....	145
FB_STRINGToPBOX.....	129	NE.....	159
FB_Weigh .....	139	NOT .....	150
FB_WriteTrace.....	124	OR .....	150
GE.....	157	OR(.....	150
GT .....	157	ORN.....	151
IF .....	181	ORN( .....	151
INT_TO_BYTE.....	187	ParameterBox .....	82
INT_TO_DINT .....	187	R .....	152
JMP .....	184	R_TRIG .....	117
JMPC .....	184	Return.....	181

ROL.....	151	окна отображения контрольной точки и точки прерывания (Watch- & Breakpoint) .....	85
ROR .....	152	.....	85
RS Flip Flop .....	118	окно ввода .....	85
S .....	152	Окно сообщений ПЛК .....	86
SHL .....	153	Операторы .....	140
SHR.....	153	операторы загрузки и сохранения .....	155
SIN .....	146	Операторы присваивания.....	180
SQRT .....	148	операторы сравнения .....	156
SR Flip Flop .....	119	операции над битами .....	148
ST.....	156	отладка .....	87
STN .....	156	ошибки .....	174
SUB.....	145	параметры.....	175
SUB(.....	145	Параметры процессов .....	159
TAN .....	146	передача данных через CANopen.....	82
TOF .....	120	передача программы ПЛК на прибор ....	86
TON.....	120	переменные и описание функционального блока .....	84
TP.....	121	Переходы .....	184
XOR .....	154	Побитовый доступ к переменным .....	179
XOR(.....	154	пошаговое выполнение.....	87
XORN.....	154	программная задача.....	80
XORN(.....	154	расширенные математические операторы .....	146
арифметические операторы .....	140	Регулятор технологического процесса..	82
визуализация .....	81	редактор .....	83
входы и выходы.....	159	Сообщения о неисправностях.....	187
Вызов функции .....	179	спецификация .....	78
Вызов функциональных блоков в ST ...	180	список инструкций (AWL / IL).....	176
загрузка, сохранение и печать .....	83	стандартные функциональные блоки..	114
запоминающее устройство.....	79	Структурированный текст (ST).....	179
Информационный параметр .....	171	тип данных .....	176
комментарии .....	178	Типы данных в ST.....	179
Конвертация типов.....	185	точки прерывания.....	87
контрольные точки .....	87	Уставки и действительные значения шины .....	168
конфигурация .....	88	Уставки и фактические значения .....	165
Литералы .....	177	функционал .....	81
Модуль визуализации ParameterBox....	125	функциональные блоки.....	89
Обзор визуализации .....	126	Цикл FOR.....	182
Обработка выражений.....	180	Цикл REPEAT .....	183
обработка данных через накопительный регистр.....	81	Цикл WHILE.....	183
обработка уставки .....	80		
образ процесса.....	79		



Электронные редукторы с летучей пиллой* .....81, 96	Сохранение макроса..... 69
языки .....176	Сохранение макроса под другим именем. 69
Поиск и замена.....69	Сохранение параметров..... 42
Прерывание выполнения .....71	Сравнение параметров ..... 42
прикрепление .....27	Сравнительный отчет ..... 42
Программа для обновления встроенного ПО196	Стандартная панель ..... 20
<b>Р</b>	Стандартное ..... 46
Работа с NORD CON .....9	<b>У</b>
Рабочая область..... 12	Удаление операторов ..... 69
Редактор макросов .....207	Указания..... 209
<b>С</b>	управление ..... 46
Создание нового макроса .....68	Управление..... 9, 45, 46, 47, 61
Сохранение и восстановление .....189, 191	<b>Ф</b>
	Файл проекта ..... 205

## **NORD DRIVESYSTEMS Group**

**Headquarters and Technology Centre**  
in Bargteheide, close to Hamburg

**Innovative drive solutions**  
for more than 100 branches of industry

**Mechanical products**  
parallel shaft, helical gear, bevel gear and worm gear units

**Electrical products**  
IE2/IE3/IE4 motors

**Electronic products**  
centralised and decentralised frequency inverters,  
motor starters and field distribution systems

**7 state-of-the-art production plants**  
for all drive components

**Subsidiaries and sales partners**  
**in 89 countries on 5 continents**  
provide local stocks, assembly, production,  
technical support and customer service

**More than 3,300 employees throughout the world**  
create customer oriented solutions

[www.nord.com/locator](http://www.nord.com/locator)

### **Headquarters:**

**Getriebebau NORD GmbH & Co. KG**

Getriebebau-Nord-Straße 1  
22941 Bargteheide, Germany

T: +49 (0) 4532 / 289-0

F: +49 (0) 4532 / 289-22 53

[info@nord.com](mailto:info@nord.com), [www.nord.com](http://www.nord.com)

**Member of the NORD DRIVESYSTEMS Group**

