

CN
BU 0000
NORD CON

操作说明



目录

1	介绍	7
1.1	关于NORD CON	7
1.2	如何使用NORD CON	7
2	图形用户界面	10
2.1	程序界面结构	10
2.2	主菜单	11
2.2.1	“文件” 目录	11
2.2.2	“编辑” 目录	12
2.2.3	“项目” 目录	13
2.2.4	“设备” 目录	13
2.2.5	“视图” 目录	15
2.2.6	“附加” 目录	16
2.2.7	“帮助” 目录	17
2.3	工具栏	17
2.3.1	标准	17
2.3.2	设备	18
2.3.3	启动	19
2.4	“项目” 视图	19
2.4.1	上下文菜单结构	20
2.5	“消息” 视图	22
2.6	“远程” 视图	23
2.7	停靠和取消停靠	24
3	通信	30
3.1	USS	30
3.1.1	一般设置	30
3.1.2	总线扫描	31
3.2	以太网	32
3.2.1	一般设置	33
3.2.2	扫描	34
4	参数化	36
4.1	如何操控参数	36
4.2	选择性参数化	37
4.3	离线参数化	37
4.4	参数查看	38
4.5	比较报告	38
4.6	从设备上传参数	39
4.7	下载参数到设备	40
5	控制	42
5.1	概述	42
5.2	标准控制	43
5.3	详细控制	43
5.3.1	概述	43
5.3.2	控制	44
5.3.3	管理设置值和实际值	45
5.3.4	格式化设定值和/或实际值	46
5.3.5	状态字	46
5.3.6	控制字	47
6	远程	49
6.1	标准	49
6.2	NORDAC SK 200 E	50
6.3	NORDAC SK 700/500/300 E	51
6.4	NORDAC vector mc	53
6.5	NORDAC vector ct	54
7	示波器	56

7.1	概述	56
7.2	显示	56
7.3	处理	57
7.4	测量	59
7.5	保存和打印	60
8	宏指令编辑器	61
8.1	用户界面和视图	61
8.1.1	“变量”窗口	61
8.1.2	属性窗口	61
8.1.3	日志窗口	64
8.2	使用宏指令	64
8.2.1	创建新宏指令	64
8.2.2	打开宏指令	64
8.2.3	保存宏指令	65
8.2.4	插入指令	65
8.2.5	复制指令	65
8.2.6	剪切指令	65
8.2.7	从指令中删除	65
8.2.8	搜索和替换	65
8.2.9	上移指令	65
8.2.10	下移指令	65
8.2.11	创建新指令	66
8.3	调度器	67
8.3.1	启动序列	67
8.3.2	取消宏指令	67
8.3.3	执行下一个指令	67
9	USS帧编辑器	68
9.1	主机（指令）	70
9.2	设备（响应）	70
10	PLC	72
10.1	一般信息	72
10.1.1	PLC规范	72
10.1.2	PLC结构	73
10.1.2.1	存储器	73
10.1.2.2	过程映像	73
10.1.2.3	程序任务	74
10.1.2.4	处理设定值	74
10.1.2.5	通过累加器进行数据处理	74
10.1.3	功能范围	75
10.1.3.1	运动控制库	75
10.1.3.2	带飞剪的电子齿轮	75
10.1.3.3	可视化	75
10.1.3.4	过程控制器	75
10.1.3.5	CANopen通信	76
10.2	创建PLC程序	76
10.2.1	加载，保存和打印	76
10.2.2	编辑器	76
10.2.2.1	变量和FB声明	77
10.2.2.2	输入窗口	78
10.2.2.3	监视和断点显示窗口	78
10.2.2.4	PLC消息窗口	79
10.2.3	将PLC程序传输到设备	79
10.2.4	调试	80
10.2.4.1	观测点（监视点）	80
10.2.4.2	待检点（断点）	80
10.2.4.3	单步	80
10.2.5	PLC配置	81
10.3	功能块	81
10.3.1	CANopen	81
10.3.1.1	概述	82
10.3.1.2	FB_NMT	82
10.3.1.3	FB_PDOConfig	83
10.3.1.4	FB_PDOReceive	85
10.3.1.5	FB_PDOSend	87

10.3.2	带飞剪的电子齿轮	89
10.3.2.1	概述	90
10.3.2.2	FB_FlyingSaw	90
10.3.2.3	FB_Gearing	91
10.3.3	运动控制	92
10.3.3.1	MC_Control	93
10.3.3.2	MC_Control_MS	94
10.3.3.3	MC_Home	96
10.3.3.4	MC_MoveAbsolute	97
10.3.3.5	MC_MoveAdditive	98
10.3.3.6	MC_MoveRelative	99
10.3.3.7	MC_MoveVelocity	100
10.3.3.8	MC_Power	101
10.3.3.9	MC_ReadActualPos	102
10.3.3.10	MC_ReadParameter	102
10.3.3.11	MC_ReadStatus	103
10.3.3.12	MC_Reset	104
10.3.3.13	MC_Stop	105
10.3.3.14	MC_WriteParameter_16 / MC_WriteParameter_32	105
10.3.4	标准	106
10.3.4.1	CTD向下计数器	106
10.3.4.2	CTU向上计数器	107
10.3.4.3	CTUD向上和向下计数器	108
10.3.4.4	R_TRIG和F_TRIG	110
10.3.4.5	RS触发器	111
10.3.4.6	SR触发器	111
10.3.4.7	TOF关闭延时继电器	112
10.3.4.8	TON打开延时继电器	113
10.3.4.9	TP时间脉冲	114
10.3.5	访问变频器的存储区域	115
10.3.5.1	FB_ReadTrace	115
10.3.5.2	FB_WriteTrace	116
10.3.6	参数盒可视化	117
10.3.6.1	可视化概述	118
10.3.6.2	FB_DINTToPBOX	118
10.3.6.3	FB_STRINGToPBOX	121
10.3.7	FB_Capture (检测快速事件)	123
10.3.8	FB_DinCounter	125
10.3.9	FB_FunctionCurve	126
10.3.10	FB_PIDT1	127
10.3.11	FB_ResetPostion	129
10.3.12	FB_Weigh	130
10.4	运算符	131
10.4.1	算术运算符	131
10.4.1.1	ABS	131
10.4.1.2	ADD和ADD (.....	132
10.4.1.3	DIV和DIV (.....	132
10.4.1.4	极限	133
10.4.1.5	最大	133
10.4.1.6	最小	133
10.4.1.7	MOD和MOD (.....	134
10.4.1.8	MUL和MUL (.....	134
10.4.1.9	MUX	135
10.4.1.10	SUB和SUB (.....	135
10.4.2	扩展数学运算符	136
10.4.2.1	COS, ACOS, SIN, ASIN, TAN, ATAN	136
10.4.2.2	EXP	137
10.4.2.3	LN	137
10.4.2.4	LOG	138
10.4.2.5	SQRT	138
10.4.3	位运算符	139
10.4.3.1	AND和AND (.....	139
10.4.3.2	ANDN和ANDN (.....	139
10.4.3.3	NOT	140
10.4.3.4	OR和OR (.....	140
10.4.3.5	ORN和ORN (.....	141
10.4.3.6	ROL	142
10.4.3.7	ROR	142

10.4.3.8 S和R	143
10.4.3.9 SHL	143
10.4.3.10 SHR	143
10.4.3.11 XOR和XOR (144
10.4.3.12 XORN和XORN (144
10.4.4 加载和存储运算符 (AWL)	145
10.4.4.1 LD	145
10.4.4.2 LDN	146
10.4.4.3 ST	146
10.4.4.4 STN	146
10.4.5 比较运算符	146
10.4.5.1 EQ	147
10.4.5.2 GE	147
10.4.5.3 GT	147
10.4.5.4 LE	148
10.4.5.5 LT	148
10.4.5.6 NE	149
10.5 处理值	149
10.5.1 输入和输出	150
10.5.2 PLC设定值和实际值	155
10.5.3 总线设定值和实际值	158
10.5.4 控制盒和参数盒	160
10.5.5 信息参数	161
10.5.6 PLC错误	164
10.5.7 PLC参数	164
10.6 语言	166
10.6.1 指令列表 (AWL/IL)	166
10.6.1.1 一般信息	166
10.6.2 结构化文本 (ST)	169
10.6.2.1 常用	169
10.6.2.2 过程	171
10.7 跳跃	173
10.7.1 JMP	173
10.7.2 JMPC	174
10.7.3 JMPCN	174
10.8 类型转换	174
10.8.1 BOOL_TO_BYTE	174
10.8.2 BYTE_TO_BOOL	175
10.8.3 BYTE_TO_INT	175
10.8.4 DINT_TO_INT	176
10.8.5 INT_TO_BYTE	176
10.8.6 INT_TO_DINT	177
10.9 PLC故障信息	177
11 项目模式	179
11.1 一般信息	179
11.2 HMI	180
11.3 保存和恢复	180
12 固件	183
12.1 如何更新固件	183
12.2 固件更新程序	185
12.3 通过系统总线进行固件更新	188
13 设置	191
13.1 用户界面	191
13.2 设备报告	192
13.3 控制	193
13.4 项目	194
13.5 目录	195
13.6 宏指令编辑器	196
13.7 参数	197
13.8 PLC	197
14 消息	198
14.1 错误和信息	198

15	诺德驱动系统	203
15.1	诺德驱动系统企业历史	204
15.2	变频器	206
15.2.1	SK 135E	206
15.2.2	SK 180E	207
15.2.3	SK 200E	207
15.2.4	SK 500E	209

1 介绍

1.1 关于NORD CON

NORD CON是PC程序，旨在4“参数化”和5.1“概述”Getriebebau诺德生产的变频器和选件模块。

使用NORD CON，可通过集成RS485接口同时控制多达31个变频器。与变频器的通信由PC串行接口处理。

为了启用试运行或系统启动，可通过PC控制连接的变频器。该程序还可以在这些操作进行时持续监测变频器的电流状态。可以使用宏指令开发完整的过程序列。

使用NORD CON，可以执行、记录和保存变频器的参数设置，这些参数设置将由变频器读取或分别传送给变频器。可以离线创建或操控参数数据库，即不连接变频器。

该程序还可以对连接的变频器进行远程控制。对于要远程控制的变频器，在PC上模拟所述型号的操作单元。对于难以进行操作或自身没有操作单元的设备，这是一种便捷的操作方式。

8. “宏指令编辑器”

6. “远程”

1.2 如何使用NORD CON

信息

串行接口

为使用NORD CON对设备进行参数化和控制，您的PC需要串行接口。

1. 安装

请在所附CD上启动NORD CON的安装程序，或从下列网站加载安装程序（“<http://www2.nord.com/cms/de/documentation/software/software-overview.jsp>”）。输入所有必要的信息并将NORD CON安装到标准目录中。

2. 连接

如果变频器配备有RS232可选接口，则可通过串行1-1电缆直接连接到PC。在这种情况下，只能连接一个变频器。每个诺德变频器都具有集成RS485接口，该接口可通过控制端子激活。该接口允许对最多31个设备的主机/从机总线系统进行配置。要将NORD CON连接到此类总线，则需要RS232-RS485转换器。

i 信息**USS设置**

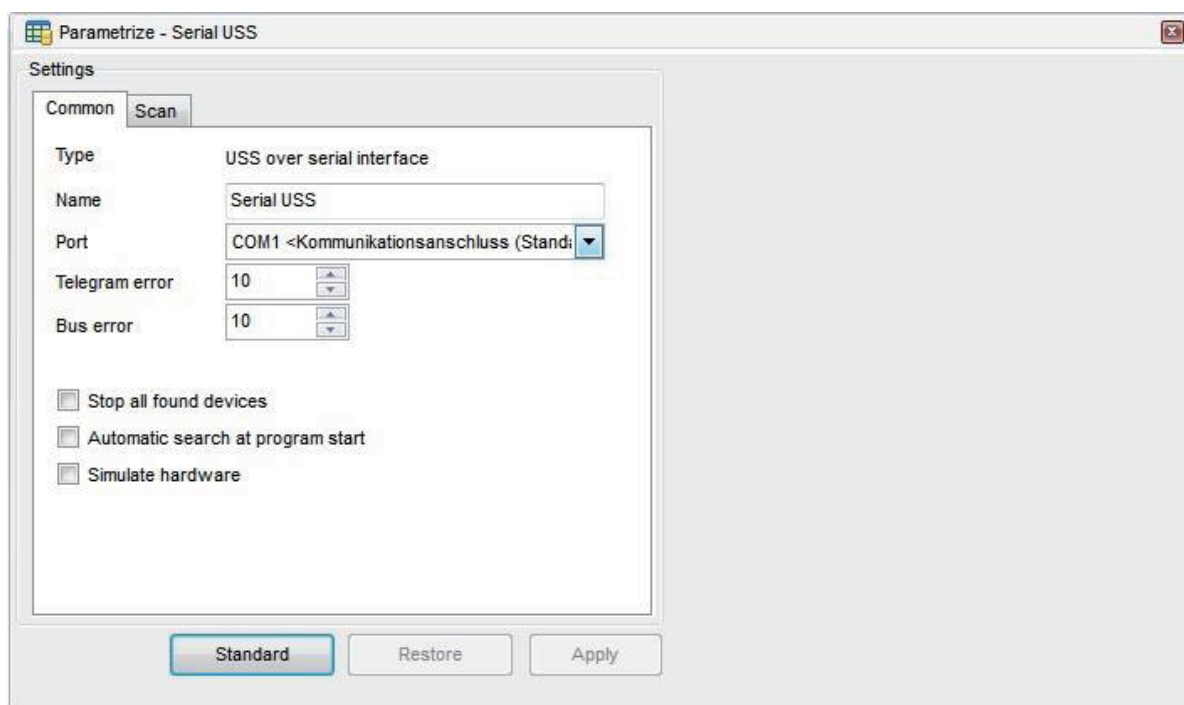
如果同时控制多个设备，请确保为每个连接的设备分配一个唯一的USS地址，并且所有设备都具有相同的3.1.2“总线扫描”设置（另请参阅相关变频器的操作说明）。

3. 运行NORD CON

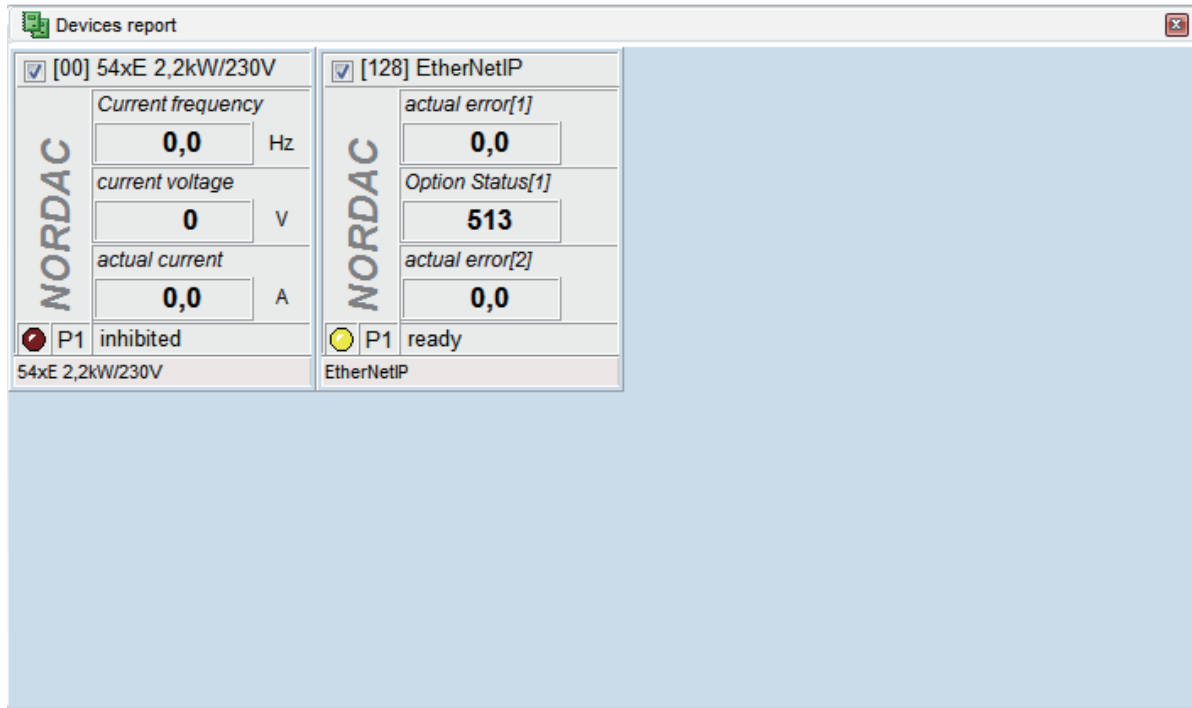
为启动NORD CON，请使用快捷键“NORD CON启动”或“启动->程序->->诺德->NORD CON 2.5->NORD CON”。

4. 设置通信模块

为了设置通信参数，必须在项目视图中选择适当的模块。通过菜单条目“设备->参数化”，可以打开模块的参数对话框。在编辑字段中，“端口”必须插入正确的COM端口号。之后，必须按“应用”按钮。第一个应用程序不需要其它设置，窗口可以关闭。

**5. 总线扫描**

总线扫描开始后，将搜索所有就绪和连接的设备。所有找到的设备都显示在项目树和设备概述中。随后，标记列表中的第一个设备，并且用户可以使用所有该设备特定的功能。



6. 使用设备

用户现在可以通过单击设备概述或项目树中的设备来选择设备。在项目树、工具栏或主菜单的弹出菜单中，可以使用控制或参数化等功能。

2.4 “项目”视图”

2.3 “工具栏”

5 “控制”

4 “参数化”

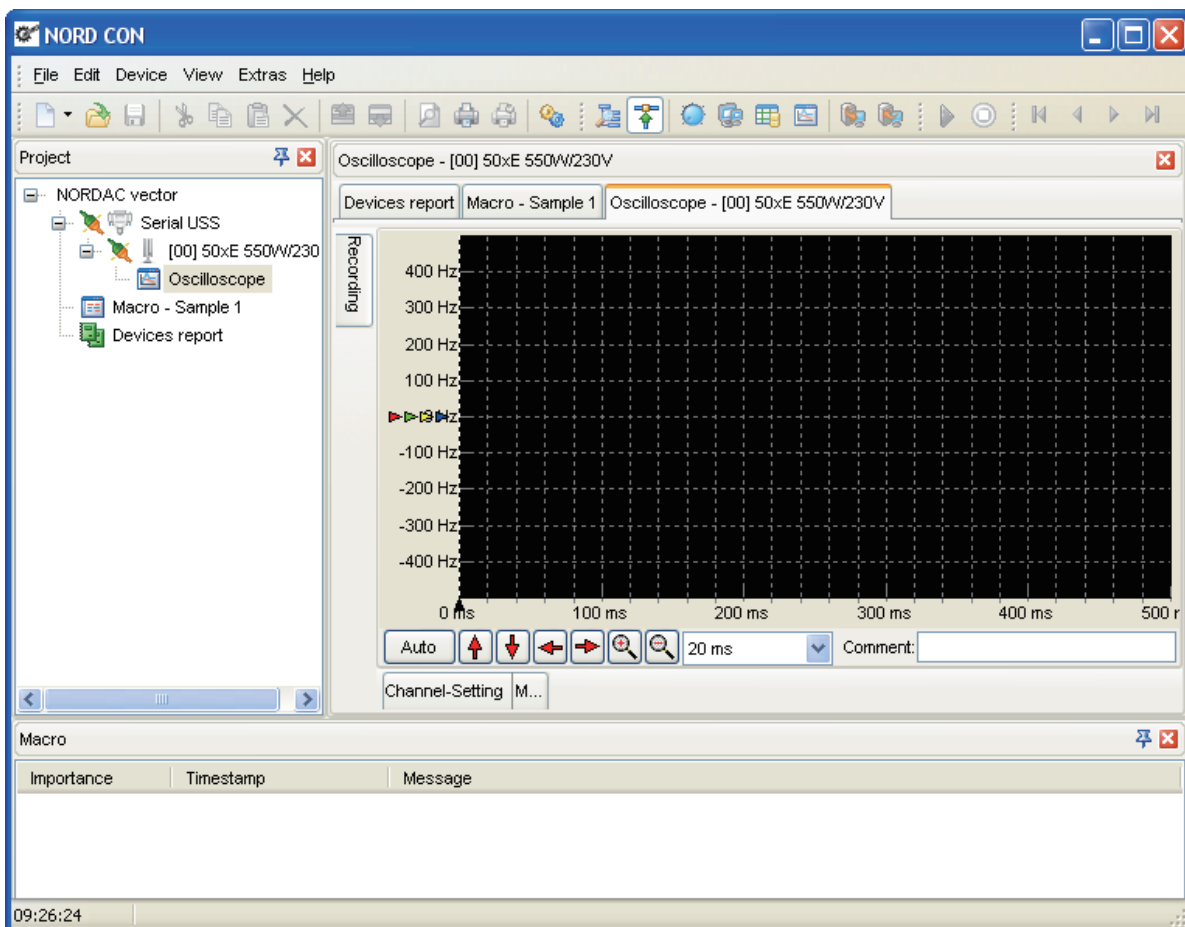
2 图形用户界面

2.1 程序界面结构

主窗口由主菜单、工具栏、工作区和不同的视图组成。在工作区中，将显示不同的编辑器窗口，如参数窗口或宏指令。窗口可以自由放置，也可以停靠在工作区的侧边。要更改停靠窗口的位置，请单击窗口的标题栏并按住鼠标按钮。随后，可以使用鼠标指针指定新的位置。彩色矩形框显示实时位置和停靠条件。释放鼠标左键后，执行实际操作。此外，用户可以通过单击标题栏停靠或取消停靠窗口。关闭应用程序时存储布局，并在重启时恢复布局。

界面分为以下几个区域：

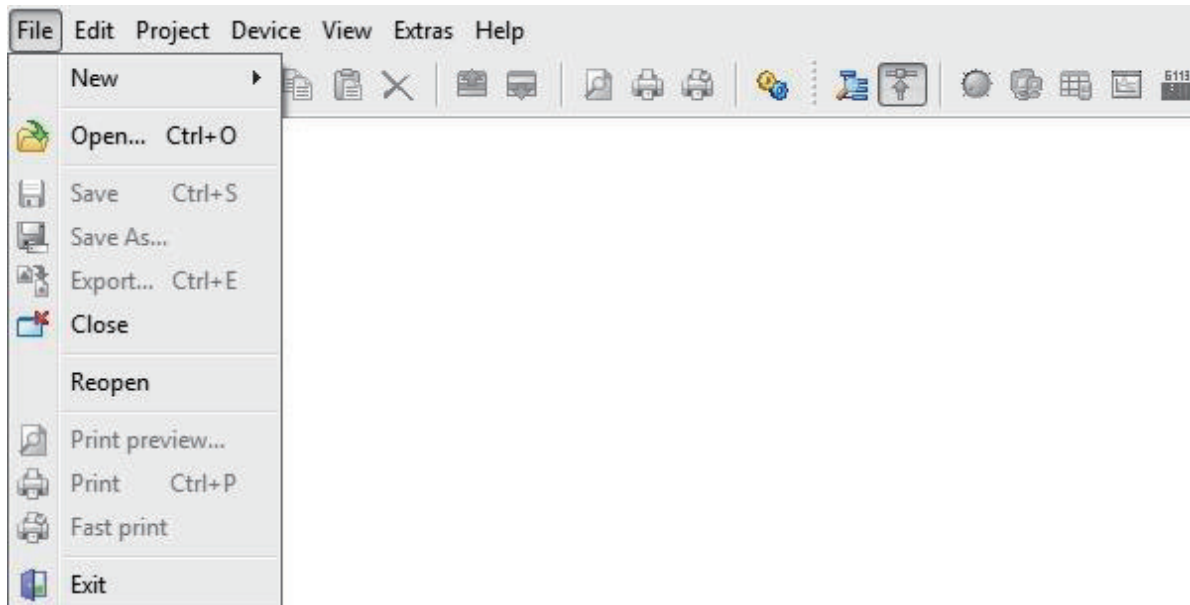
- 2.2 “主菜单”
- 2.3 “工具栏”
- 工作区
- 2.4 ““项目”视图”
- 2.5 ““消息”视图”
- 2.6 ““远程”视图”












2.2 主菜单

主菜单是所有应用程序操作的中心位置。所有编辑器窗口都在那里注册其窗口特定的操作。这些操作按目录划分。

2.2.1 “文件” 目录



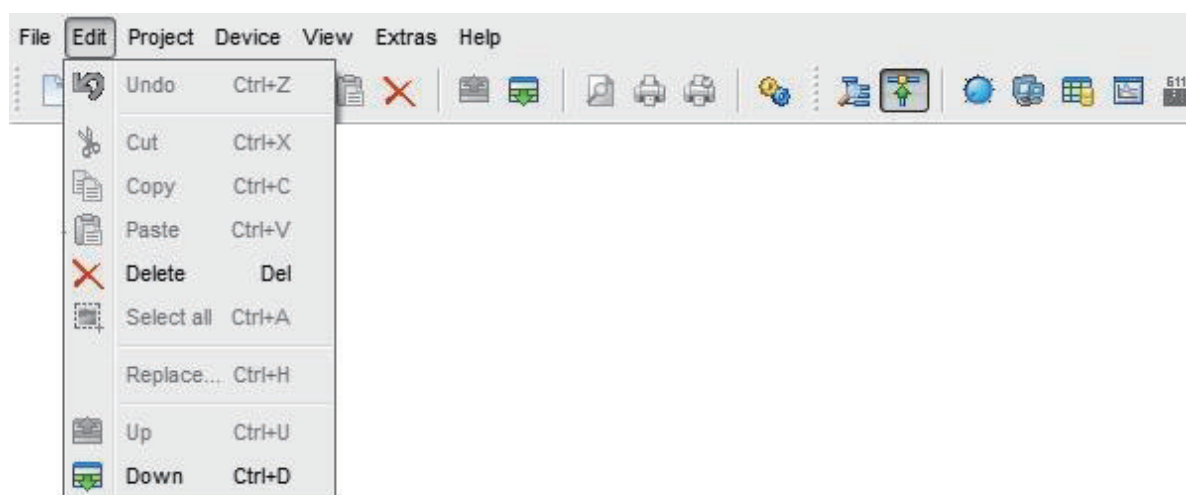
操作名称	快捷键	图标	描述
新数据集			该操作作为一个新设备打开参数窗口。用户必须在上一个窗口中选择所需的设备。
新宏指令			该操作用一个新文档打开宏指令编辑器。如果宏指令窗口已打开，则用户可以存储当前文档。 注意： 在当前版本中，只能打开一个宏指令窗口！
PLC程序			该操作用一个空文档打开PLC编辑器。如果PLC程序已打开，则用户可以存储当前文档。
打开	Ctrl + O		该操作打开文件选择对话框，以打开一个存储的文档。用户使用文件筛选器选择一种文档类型，然后选择文件。支持以下类型文档： <ul style="list-style-type: none"> 参数文件 (*.ndbx, *.db (V1.27)) 示波器记录文件 (*.scox, *.sco (V1.27)) 宏指令文件 (*.ncmx, *.ncm (V1.27)) PLC文件 (*.awlx, *.awl, *.nstx)
保存	Ctrl + S		该操作存储当前文档。该操作传递到活动编辑器窗口，并在那里实现。根据编辑器的类型，可以实现不同的操作。
另存为...			该操作用一个新名称存储当前文档。该操作传递到活动编辑器窗口，并在那里实现。根据编辑器的类型，可以实现不同的操作。



操作名称	快捷键	图标	描述
导出	Ctrl + E		该操作将数据活动编辑器窗口导出到一个文件中。该操作传递到活动编辑器窗口，并在那里实现。根据编辑器的类型，可以实现不同的操作。
重新打开			该操作包含一个子菜单，其中列出了打开的最后一个文档。历史记录仅限于5个。单击其中一个文件时，会再次打开该文件。
打印	Ctrl + P		该操作传递到活动编辑器窗口，并在那里实现。根据编辑器的类型，可以实现不同的操作。如果未打开编辑器窗口或编辑器不支持该操作，则会停用该操作。
打印预览...			该操作打开活动编辑器的一个打印预览。根据编辑器的不同，打印预览可以进行不同的开发。如果未打开编辑器窗口或编辑器不支持该操作，则会停用该操作。
退出			该操作关闭应用程序。







信息

如果未打开编辑器窗口或编辑器不支持该操作，则会停用操作。

2.2.2 “编辑” 目录



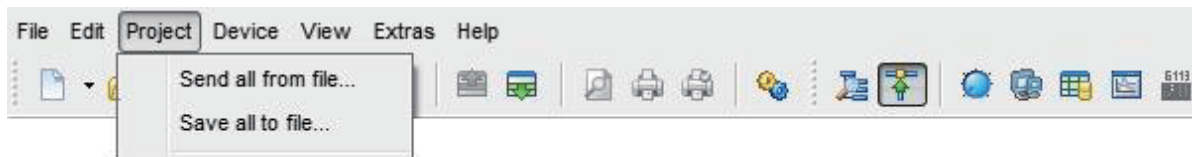
操作名称	快捷键	图标	描述
取消	Ctrl + Z		该操作取消最后的操作。该操作传递到活动编辑器窗口，并在那里实现。根据编辑器的类型，可以实现不同的操作。
剪切	Ctrl + X		该操作剪切所选的对象，并将其复制到剪贴板中。该操作传递到活动控制成员，并在那里实现。根据编辑器的类型，可以实现不同的操作。



操作名称	快捷键	图标	描述
复制	Ctrl + C		该操作将所选的对象复制到剪贴板中。该操作传递到活动控制成员，并在那里实现。根据编辑器的类型，可以实现不同的操作。
粘贴	Ctrl + V		该操作将剪贴板的内容粘贴到选定的位置。该操作传递到活动控制成员，并在那里实现。根据编辑器的类型，可以实现不同的操作。 说明： 如果当前控制元素不支持该操作，或无法插入剪贴板的内容，则停用该操作。
删除	Ctrl + Del		该操作删除所选的对象。该操作传递到活动控制，并在那里实现。根据编辑器的类型，可以实现不同的操作。
全选	Ctrl + A		该操作选择活动控制的所有对象。
替换...	Ctrl + H		该操作搜索指示的文本，然后将其替换为其它文本。在对话框中，可以调整适当的选项。
向上	Ctrl + U		该操作将所选的对象向上移动一个位置。
向下	Ctrl + D		该操作条目将所选的对象向下移动一个位置。

信息

如果未打开编辑器窗口或编辑器不支持该操作，则会停用操作。

2.2.3 “项目” 目录

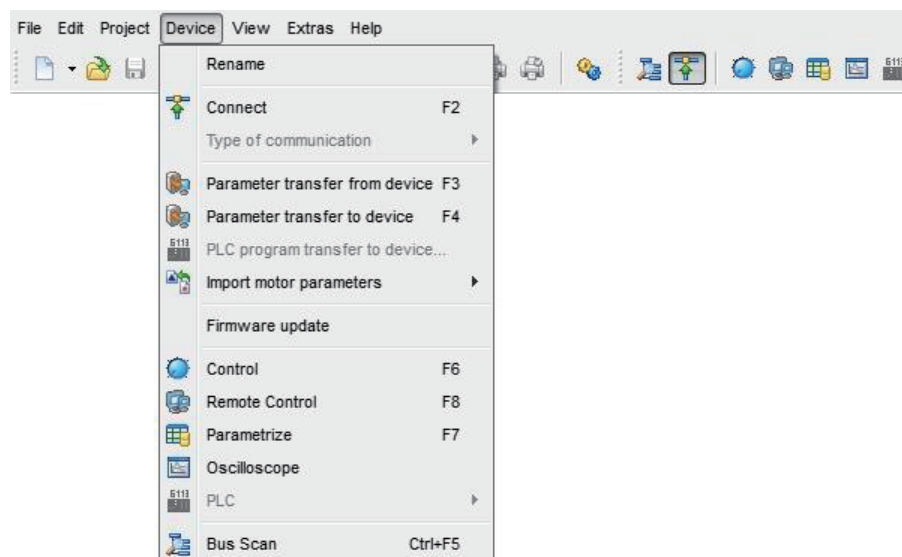


操作名称	快捷键	图标	描述
将所有内容保存到文件...			该操作读取设备的所有参数，并将其保存到一个文件。
从文件发送文件...			该操作打开一个文件，将所存的参数发送到设备。





信息

如果未打开编辑器窗口或编辑器不支持该操作，则会停用操作。

2.2.4 “设备” 目录



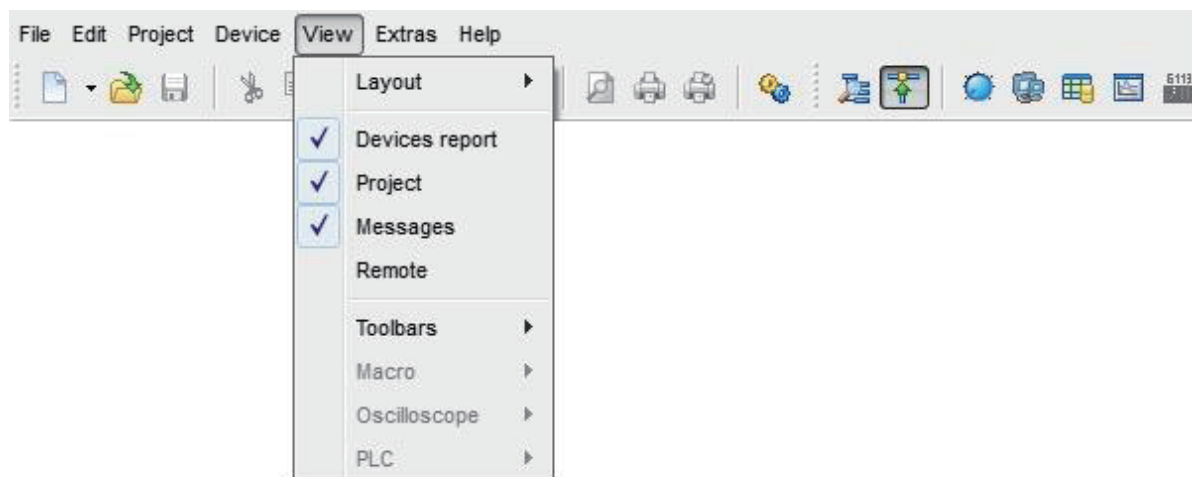
操作名称	组合键	图像	描述
重命名			借助此菜单项，用户可以更改突出显示的变频器的名称。
连接	F2		该菜单项连接或断开与突出显示的变频器的连接。
通信类型/串行USS			该操作存储当前文档。该操作传递到活动编辑器窗口，并在那里实现。根据编辑器的类型，可以实现不同的操作。
通信类型/以太网			该菜单项将通信模块设置为“以太网”。如果通信类型更改，将删除变频器列表！
从变频器上传参数	F3		该菜单项开始将参数从变频器上传到PC。
下载参数到设备	F4		该菜单项开始将参数从PC上传到变频器。
将PLC程序传输到设备			该菜单项将保存的PLC程序传输到所选的变频器。
导入电机参数			该功能可以从外部源导入电机参数。如果用户在文件选择对话框中选择了电机参数文件 (*.csv)，则列出该文件包含的所有电机。从列表中选择数据记录，并单击OK以确认。然后将参数传输到所选的变频器。如果参数窗口打开，则会导入参数窗口中的值，但不会传输到变频器。参数的传输必须单独进行。
更新固件			该菜单项启动固件上传程序。
控制	F6		该菜单项打开工作区中突出显示的变频器的“控制”窗口。如果窗口已经打开，则窗口会到前台。
远程控制	F8		该菜单项打开“查看和控制”窗口中突出显示的“远程控制”窗口。如果窗口已经打开，则窗口会到前台。

操作名称	组合键	图像	描述
参数设置	F7		该菜单项打开工作区中突出显示的变频器的“参数”窗口。如果窗口已经打开，则窗口会到前台。
示波器			该菜单项打开工作区中突出显示的变频器的“示波器”窗口。如果窗口已经打开，则窗口会到前台。
PLC			该菜单项打开工作区中突出显示的变频器的“PLC”窗口。如果窗口已经打开，则窗口会到前台。
总线扫描	Ctrl + F5		该菜单项为所选的通信模块进行一次网络扫描。 注意： 在网络扫描期间，所有变频器都从变频器列表中移除，并且所有设备特定的窗口都将关闭。

信息

如果未打开编辑器窗口或编辑器不支持该操作，则会停用操作。

2.2.5 “视图” 目录



操作名称	快捷键	描述
布局 -> 标准		该操作返回所有视图的标准布局。编辑器窗口的位置不变。
布局 -> 标准所有窗口		该操作返回包括工作区在内的所有窗口的标准应用程序布局。
设备报告		该操作关闭或打开设备报告。
2.4 “项目”视图		该操作关闭或打开“项目”视图。
2.5 “消息”视图		该操作关闭或打开“日志”视图。
2.6 “远程”视图		该操作关闭或打开“远程控制”视图。
工具栏->标准		该操作关闭或打开“标准”工具栏。
工具栏->设备		该操作关闭或打开“设备”工具栏。
工具栏->启动		该操作关闭或打开“启动”工具栏。
宏指令		该操作打开一个子菜单。在该子菜单中，列出宏指令编辑器的所有特殊操作。操作的状态和执行由活动宏指令编辑器进行管理。若无窗口处于活动状态，则停用所有操作。
示波器		该操作打开一个子菜单。在该子菜单中，列出示波器的所有特殊操作。操作的状态和执行由活动示波器进行管理。若无窗口处于活动状态，则停用所有操作。

信息

如果未打开编辑器窗口或编辑器不支持该操作，则会停用操作。

2.2.6 “附加”目录



操作名称	快捷键	描述
通信设置		该操作打开一个窗口，编辑所选通信模块的通信设置。
设置		该操作打开一个窗口，编辑程序的整体设置。
日志		该操作打开一个子菜单。在该子菜单中，列出“日志”视图的所有特殊操作。操作的状态和执行由该视图进行管理。

2.2.7 “帮助” 目录







操作名称	快捷键	描述
帮助	F1	该操作打开在线帮助，并选择寄存器映射“目录”。
索引		该操作打开在线帮助，并选择寄存器映射“索引”。
关于NORD CON		该操作打开与程序信息的一个对话框。

2.3 工具栏

在工具栏中，最常用的操作可用于快速访问。通过用鼠标单击工具栏中的相应符号，可以指定所需的操作。



可以使用以下工具栏：








2.3.1 标准

操作名称	图标	描述
新数据集		该操作作为一个新设备打开参数窗口。在用户打开对话框之前，必须选择设备。
新宏指令		该操作作用一个空文档打开宏指令编辑器。如果宏指令已打开，则用户可以存储当前文档。 注意： 在当前版本中，只能打开一个宏指令窗口！
新PLC程序		该操作作用一个空文档打开PLC编辑器。如果程序已打开，则用户可以存储当前文档。
打开		该操作打开文件对话框，以打开一个存储的文档。用户使用文件筛选器选择一种文档类型，然后选择文件。支持以下类型文档： <ul style="list-style-type: none"> 参数数据集V1.27 (*.db) 参数数据集 (*.ndbx) 示波器记录文件 (*.sco) 范围文件V2.1 (*.scox) 宏指令 (*.ncmx) 宏指令V1.27 (*.ncm) PLC程序 (*.awl)








操作名称	图标	描述
保存		该操作存储当前文档。该操作传递到活动编辑器窗口，并在那里实现。根据编辑器的类型，可以实现不同的操作。
剪切		该操作剪切所选的对象，并将其复制到剪贴板中。该操作传递到活动控制元素，并在那里实现。根据编辑器的类型，可以实现不同的操作。
复制		该操作将所选的对象复制到剪贴板中。该操作传递到活动控制元素，并在那里实现。根据编辑器的类型，可以实现不同的操作。
粘贴		该操作将剪贴板的内容粘贴到选定的位置。该操作传递到活动控制成员，并在那里实现。根据编辑器的类型，可以实现不同的操作。 说明： 如果当前控制元素不支持该操作，或无法插入剪贴板的内容，则停用该操作。
删除		该操作删除所选的对象。该操作传递到活动控制成员，并在那里实现。根据编辑器的类型，可以实现不同的操作。
向上		该操作将所选的对象向上移动一个位置。
向下		该操作将所选的对象向下移动一个位置。
预览		该操作打开活动编辑器的一个打印预览。根据编辑器的不同，打印预览可以进行不同的开发。如果未打开编辑器窗口或编辑器不支持该操作，则会停用该操作。
打印		该操作打印活动编辑器的内容。如果未打开编辑器窗口或编辑器不支持该操作，则会停用该操作。
快速打印		该操作在无打印对话框的情况下打印活动编辑器的内容。如果未打开编辑器窗口或编辑器不支持该操作，则会停用该操作。
设置		该操作打开一个窗口，编辑程序的整体设置。

2.3.2 设备

操作名称	图标	描述
总线扫描		该操作为所选的通信模块进行一次网络扫描。 说明： 使用网络扫描，所有设备都从设备列表中移除，并且所有设备特定的窗口都将关闭！
连接		该操作连接或断开与所选设备的连接。

操作名称	图标	描述
控制		该操作打开工作区中所选设备的“控制”窗口。如果窗口已经打开，则窗口会到前台。
远程		该操作打开所选设备的“远程”窗口。如果窗口已经打开，则窗口会到前台。
参数化		该操作打开工作区中所选设备的“参数”窗口。如果窗口已经打开，则窗口会到前台。
示波器		该操作打开工作区中所选设备的“示波器”。如果窗口已经打开，则窗口会到前台。
PLC		该操作打开工作区中所选设备的PLC编辑器。如果窗口已经打开，则窗口会到前台。
从设备上传参数		该操作将参数从设备上传到PC。
下载参数到设备		该操作将参数从PC下载到设备。

2.3.3 启动

操作名称	快捷键	图标	描述
PLC设置			该操作打开PLC的设置。
编译	Shift + F7		该操作启动PLC程序的翻译。
编程	Shift F8		该操作将PLC程序加载到设备。
运行	F9		该操作打开工作区中所选设备的“参数”窗口。如果窗口已经打开，则窗口会到前台。
取消	F11		该操作终止运行PLC程序或宏指令。该操作传递到活动编辑器窗口，并在那里实现。根据编辑器的类型，可以实现不同的操作。
下一步	F12		该操作执行下一步指令。该操作传递到活动编辑器窗口，并在那里实现。根据编辑器的类型，可以实现不同的操作。
调试	Shift + F5		该操作使用调试模式运行PLC程序。

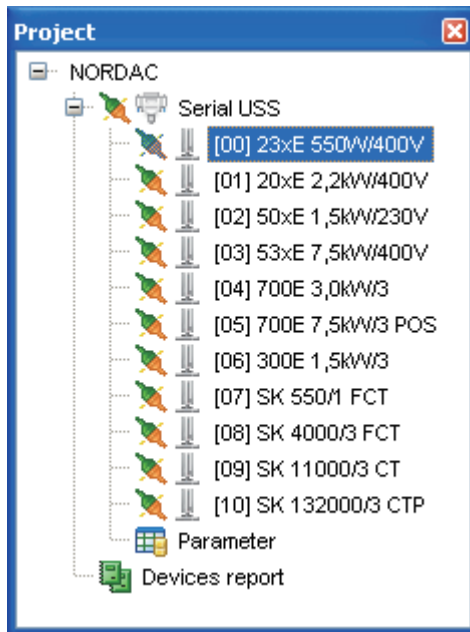
信息

如果未打开编辑器窗口或编辑器不支持该操作，则会停用操作。



2.4 “项目”视图

在“项目”视图中，项目的所有设备都显示在树形结构中。可以使用主菜单选项“视图->项目”关闭或打开项目。借助鼠标，可以在各个设备之间导航。如果选择了视图，则可以另外选择带有“向上”和“向下”箭头键的设备。如果鼠标指针位于设备条目上，则会显示有关设备型号和现场总线地址的参考。选择设备后，用户可以使用工具栏以及弹出菜单执行所有操作。如果操作以灰色阴影显示，则所选设备不支持。

通过在视图中单击鼠标右键打开2.4.1 “上下文菜单结构”。

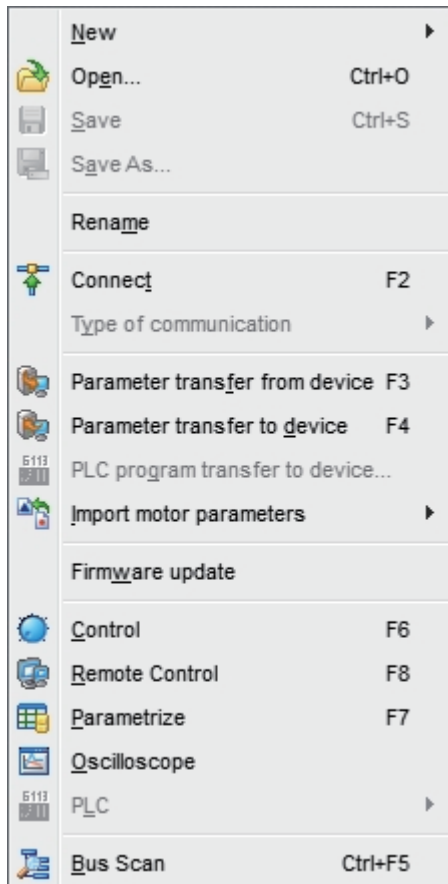


设备状态

-  设备连接在线
-  设备连接离线

2.4.1 上下文菜单结构

显示屏显示项目视图的上下文菜单。菜单始终与项目树中标记的节点相关。











操作名称	组合键	描述
新参数集		该菜单项打开新设备参数窗口。用户必须预先在对话框中选择所需的设备。
新宏指令		该菜单项用一个空文档打开宏指令编辑器。如果宏指令编辑器已打开，则要求用户保存当前文档。 注意： 当前版本只能打开一个宏指令编辑器！
新PLC程序		该菜单项用一个空文档打开PLC编辑器。如果窗口已打开，则用户可以保存当前文档。
打开	CTRL + O	该菜单项用一个空文档打开PLC编辑器。如果窗口已打开，则用户可以保存当前文档。
保存	CTRL + S	该菜单项保存当前文档。该操作传递到活动编辑器窗口，并在那里执行。根据编辑器的类型，可在那里执行不同的操作。
另存为...		该菜单项以新名称保存当前文档。该操作传递到活动编辑器窗口，并在那里执行。根据编辑器的类型，可在那里执行不同的操作。
重命名		该操作打开用以更改设备名称的输入字段。
连接	F2	该操作连接或断开与突出显示的设备的连接。

操作名称	组合键	描述
通信类型/串行USS		该菜单项将通信模块设置为“串行USS”。 如果通信类型更改，将删除设备列表！
通信类型/以太网		该菜单项将通信模块设置为“以太网”。 如果通信类型更改，将删除设备列表！
从变频器上传参数	F3	该操作开始将参数从设备上传到PC。
下载参数到设备	F4	该菜单项开始将参数从PC下载到设备。
更新固件		该操作启动上传固件的程序。
控制	F6	该操作打开工作区中突出显示的设备的“控制”窗口。如果窗口已经打开，则窗口会到前台。
远程控制	F8	该操作打开“查看和控制”窗口中突出显示的设备的“远程控制”窗口。如果窗口已经打开，则窗口会到前台。
参数设置	F7	该操作打开工作区中突出显示的设备的“参数”窗口。如果窗口已经打开，则窗口会到前台。
示波器		该操作打开工作区中突出显示的设备的“示波器”窗口。如果窗口已经打开，则窗口会到前台。
PLC		该操作打开工作区中突出显示的设备的“PLC”窗口。如果窗口已经打开，则窗口会到前台。
总线扫描	CTRL + F5	该操作触发一次新的总线扫描。 注意： 在总线扫描期间，所有设备都从设备列表中移除，并且所有设备特定的窗口都将关闭！

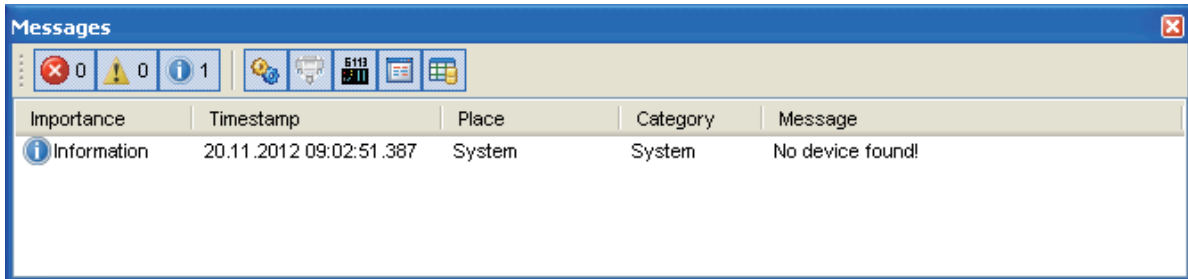
2.5 “消息”视图

该视图包含所有“NORD CON”消息的列表。默认情况下，条目以时间升序显示。可以通过单击列标题调整顺序。以下筛选器可用于筛选：

筛选器	图标	描述
错误		启用该筛选器时，将显示所有错误。此外，在按钮标题中显示错误数量。
警告		启用该筛选器时，将显示所有警告。在按钮标题中显示警告数量。
信息		启用该筛选器时，将显示所有信息。在按钮标题中显示信息项数量。
系统		启用该筛选器时，将显示“系统”目录的所有消息。
通信		启用该筛选器时，将显示“通信”目录的所有消息。
PLC		启用该筛选器时，将显示“PLC”目录的所有消息。

宏指令		启用该筛选器时，将显示“宏指令”目录的所有消息。
参数		启用该筛选器时，将显示“参数”目录的所有消息。

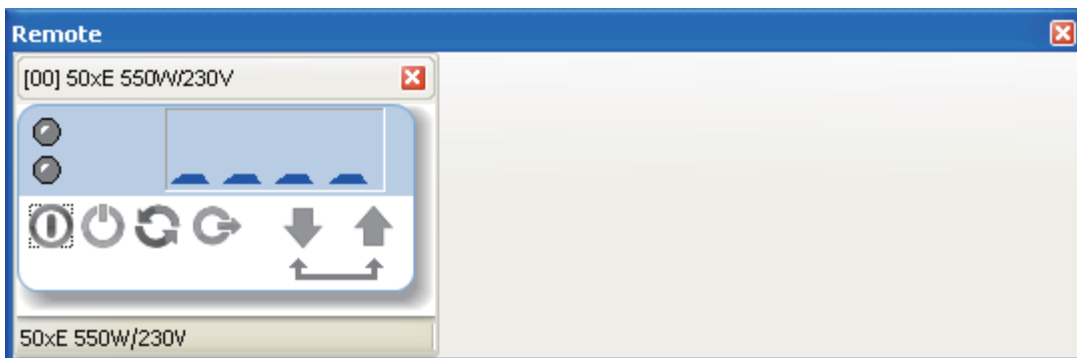
可以通过弹出菜单（鼠标右键）保存或删除消息。这些操作可以通过主菜单（“附加/消息”）执行。



操作名称	描述
删除	该操作删除列表。
保存	该操作将条目存储到文件。

2.6 “远程”视图

“远程”视图包含2.6 ““远程”视图”功能的所有窗口。视图在打开第一个窗口时自动打开，在关闭最新窗口时关闭。视图可以像所有视图一样停靠或取消停靠工作区。如果用户关闭视图，则可以通过再次“远程”操作打开。新窗口始终停靠在最后一个窗口的左侧。借助于鼠标，可以再次停靠或取消停靠该窗口。如果第一次显示视图（视图->远程），则会自动打开所有远程窗口。



信息

远程

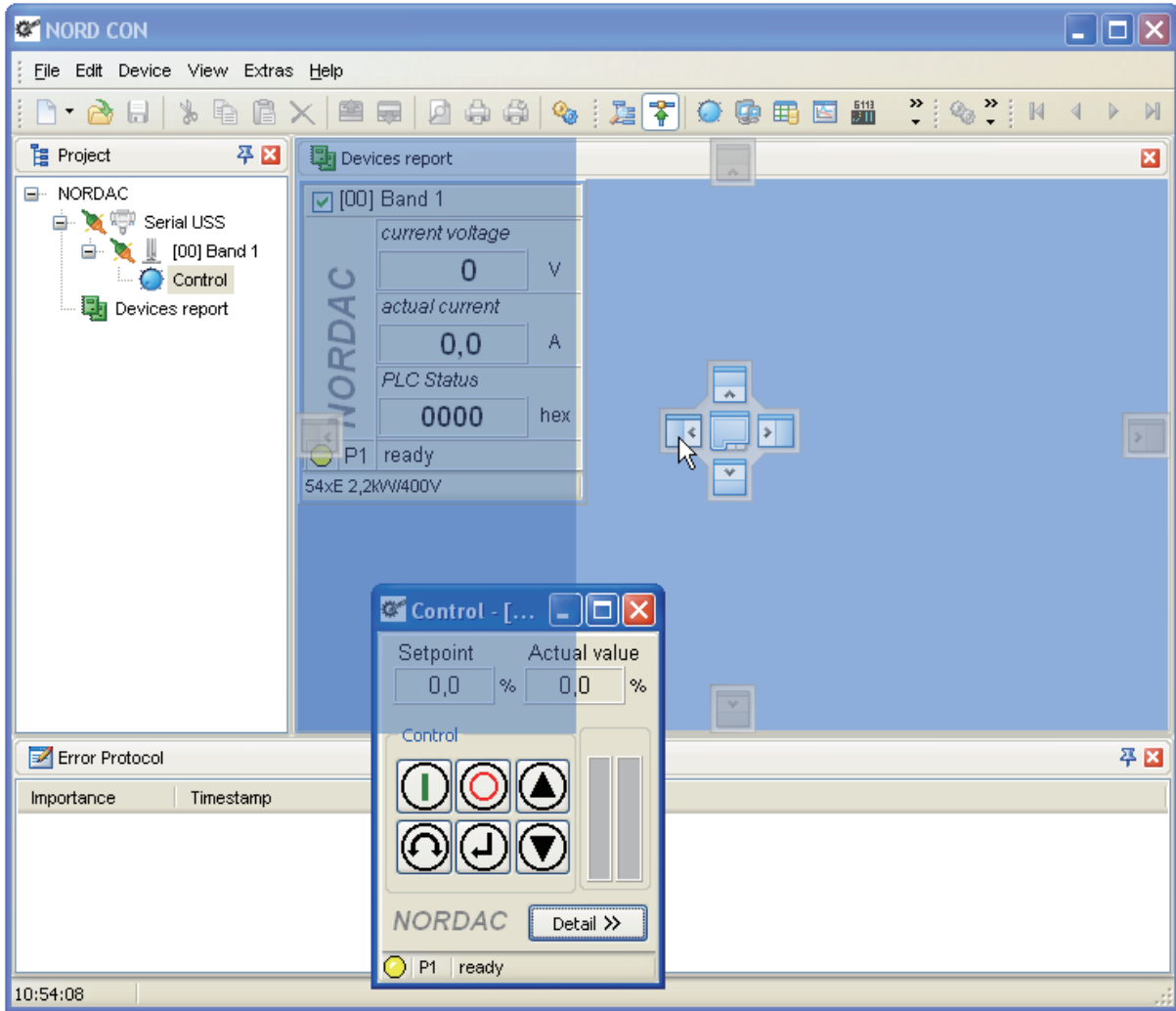
“远程”类型的窗口只能停靠在“远程”视图中。

2.7 停靠和取消停靠

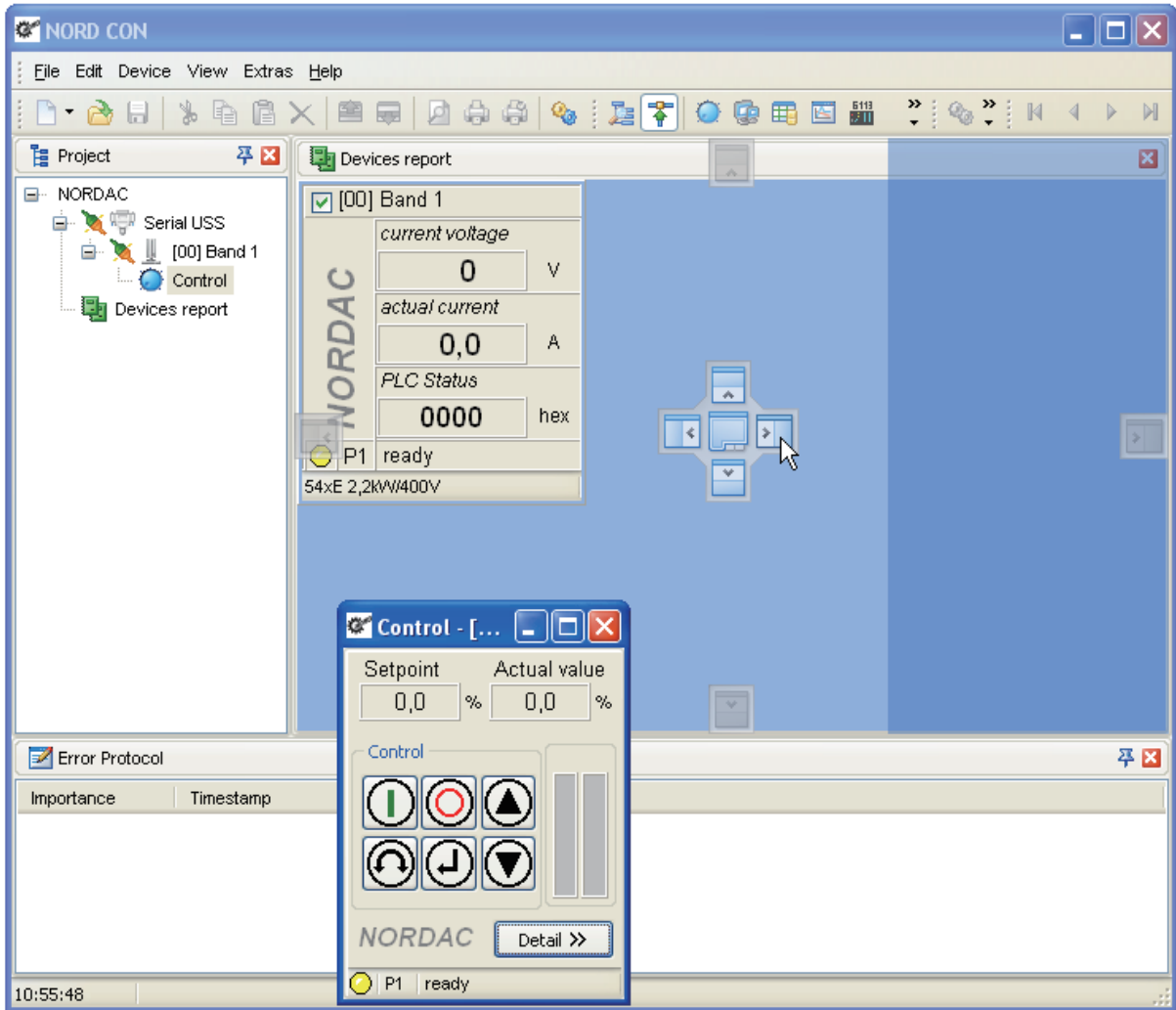
使用NORD CON的新设计，用户可以根据自己的要求调整窗口的布局。原则上，可以取消停靠每个视图和编辑器窗口，并将它们在屏幕上自由放置。为此，用户必须在标题边框上按鼠标左键，并将彩色矩形框移动到所需位置。释放鼠标按钮后，视图或编辑器窗口将作为独立窗口保留在这些位置上。使用编辑器窗口，还有另外一种可能性-弹出菜单，当在标题边框上单击鼠标右键时该弹出菜单打开。停靠功能类似于取消停靠功能。彩色矩形框表明每种情况下的实时停靠位置。

窗口类型	规则
主窗口视图（例如项目、日志、远程窗口）	主窗口的视图只能停靠在工作区的左、右和/或下边缘。在这些窗口内，没有规则，用户可以选择位置。
编辑器窗口（例如宏指令编辑器、参数窗口、示波器）	编辑器窗口只能停靠在工作区内。但是，调整固定在下和/或上，或作为寄存器映射。
宏指令编辑器视图	宏指令编辑器视图只能停靠在宏指令窗口中。此处的调整固定在左、右或下。在视图内，未定义任何规则。
示波器视图	示波器窗口视图只能停靠在示波器窗口中。此处的调整固定在左、右或下。在视图内，未定义任何规则。
“远程”窗口	“远程”窗口只能停靠在“远程”视图中。此处的调整固定在左。

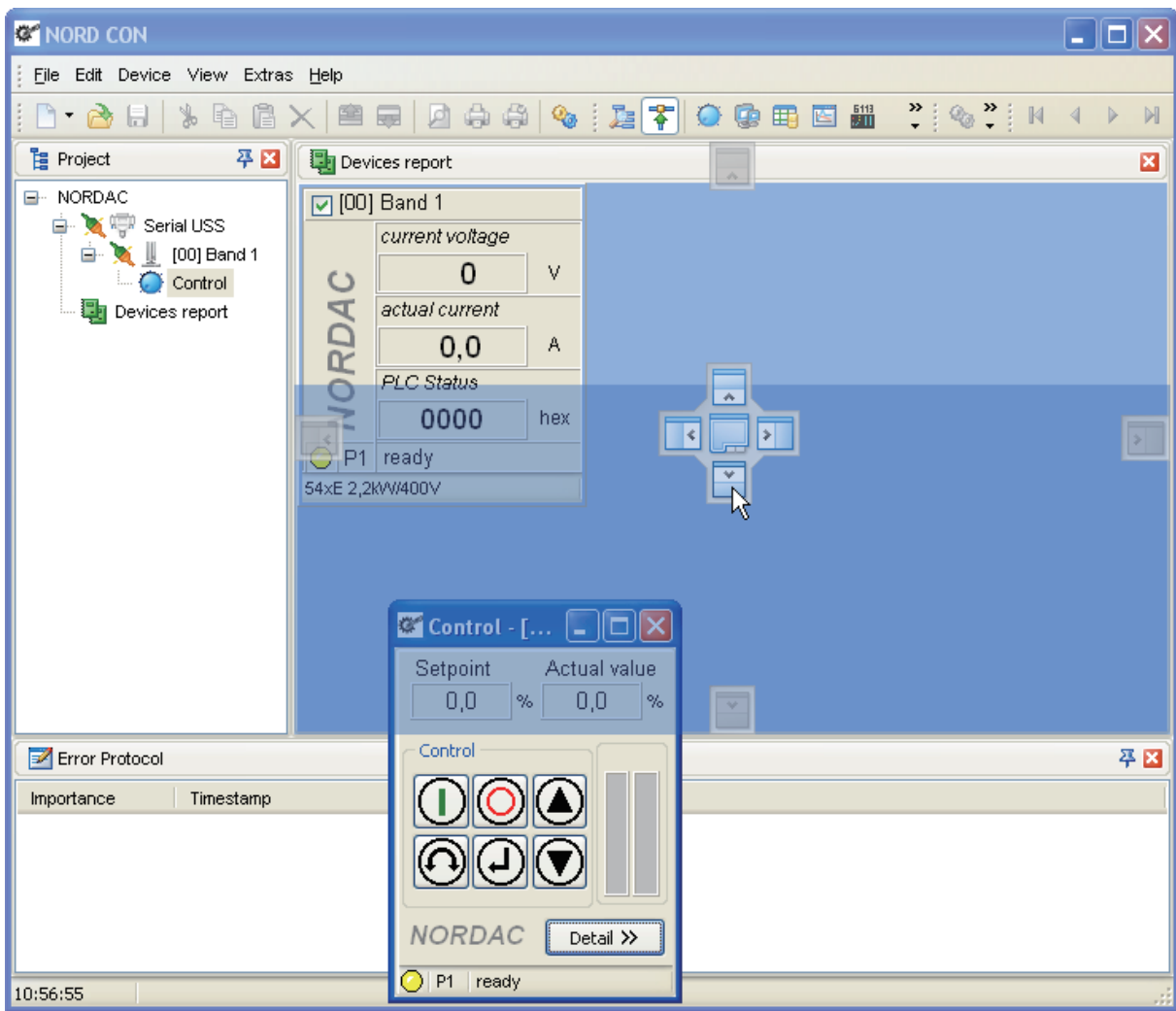
停靠位置左



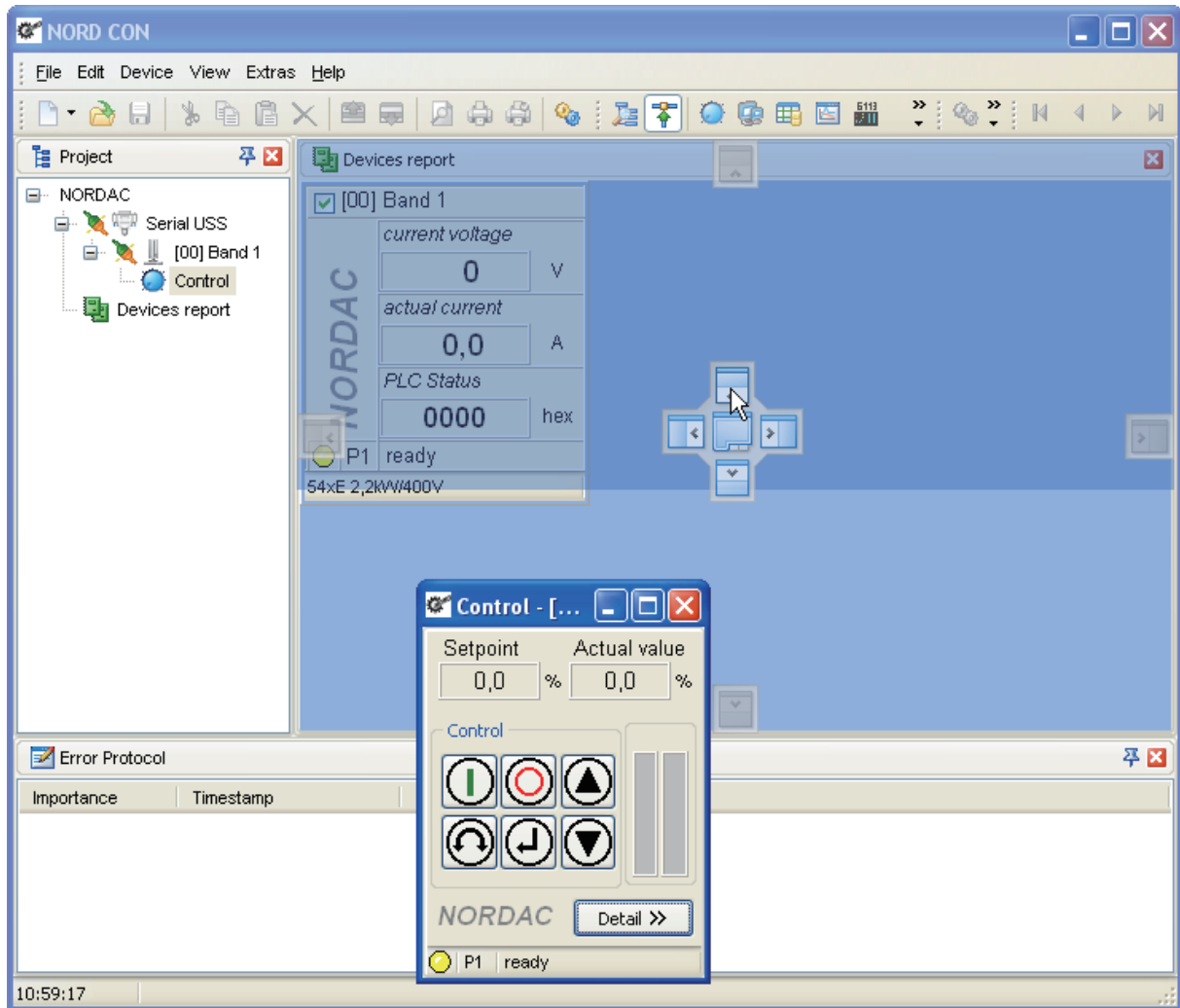
停靠位置右



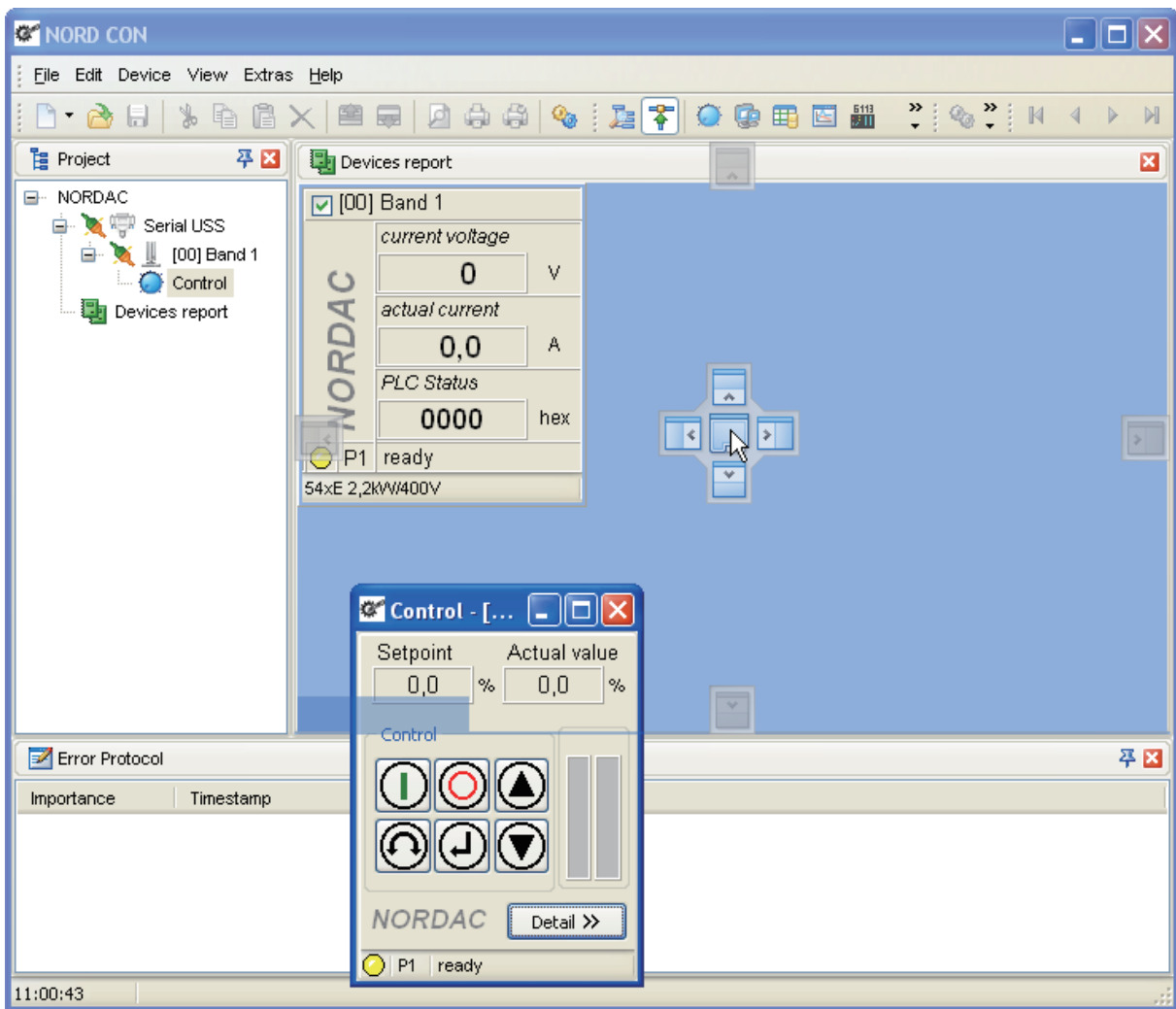
停靠位置下



停靠位置上



停靠位置选项卡



3 通信

要与设备开始连接，必须在项目中插入适当的通信模块。安装后，将配置USS通信模块。通过“参数化”操作，用户可以修改模块的参数。

目前支持以下通信模块：

3.1 USS

3.1.1 一般设置

名称

用户可以在输入字段中为通信模块分配名称。

端口

在选择框中，用户指定与变频器连接的PC的COM端口。

电报错误

在输入字段中，用户指定允许的电报错误数量。如果电报内容不正确，即响应与参数条目不对应，则会发生电报错误。通常，每次指令的2个电报后都会给出响应。允许的电报错误数量指定了在显示故障信息之前允许尝试的次数。

总线错误

在输入字段中，用户指定允许的总线错误数量。如果接收或传输电报有错误，则发生总线错误。将删除不正确的电报。在此，可以设置产生错误的电报的允许数量。对于存在干扰的环境，应将容错设置为适当的较高值。

停止所有检测到的设备

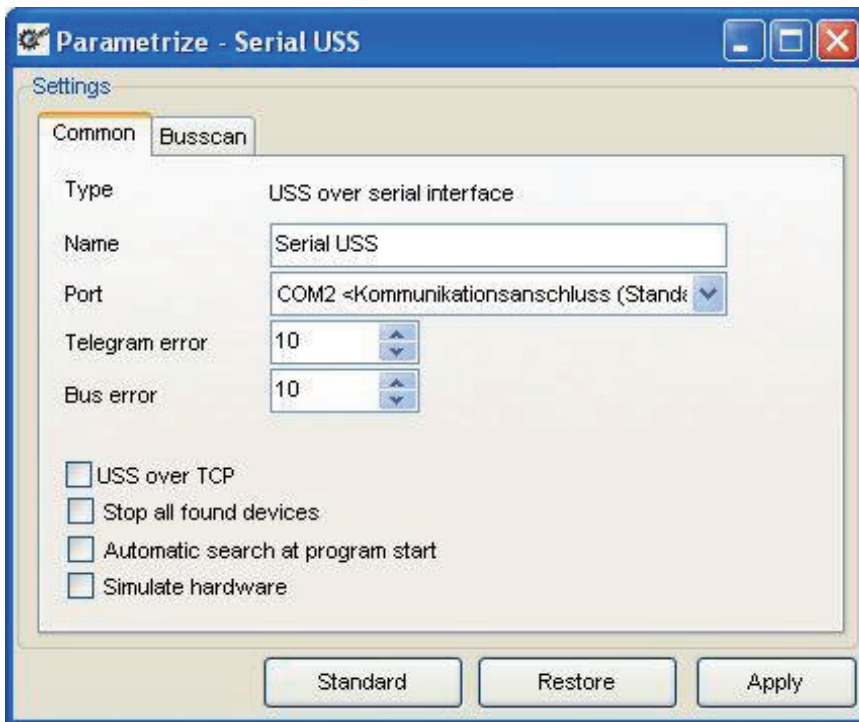
如果启用该选项，则在设备搜索后，NORD CON会向检测到的每个设备发送“禁用”命令。如果设备通过总线（P509）控制，则该设备停止。

程序启动后自动设备搜索

该选项启用或禁用程序启动时的自动搜索。如果启用该选项，则在启动NORD CON时自动启动设备搜索。

模拟硬件

使用该选项，用户可以启用或禁用所连接硬件的模拟。



信息

更改

所有更改仅在按下“采用”按钮之后生效。可以使用“恢复”按钮恢复实时有效的设置。

3.1.2 总线扫描

波特率

在选择框中，用户可以选择串行接口的通信速度。变频器上必须选择相同的值。当使用多个变频器，所有连接设备上的设备必须相同。超过115200位/秒的波特率是用户特定的波特率，并非所有设备都支持。

信息

连接问题

较旧的串行PC接口有时无法对齐用户特定的准确波特率。因此，无法与设备进行连接。

所有波特率的总线扫描

通过该操作，用户可以激活或停用所有波特率的总线扫描。如果所连接设备的波特率未知，则所有波特率的搜索可以找到正确的波特率，以开始通信。

起始波特率

在选择框中，用户可以定义波特率以开始波特率搜索。

起始地址

在此字段中，可以定义USS地址，NORD CON的搜索运行从该地址开始查找连接的设备。NORD CON无法找到所有地址较低的变频器。

结束地址

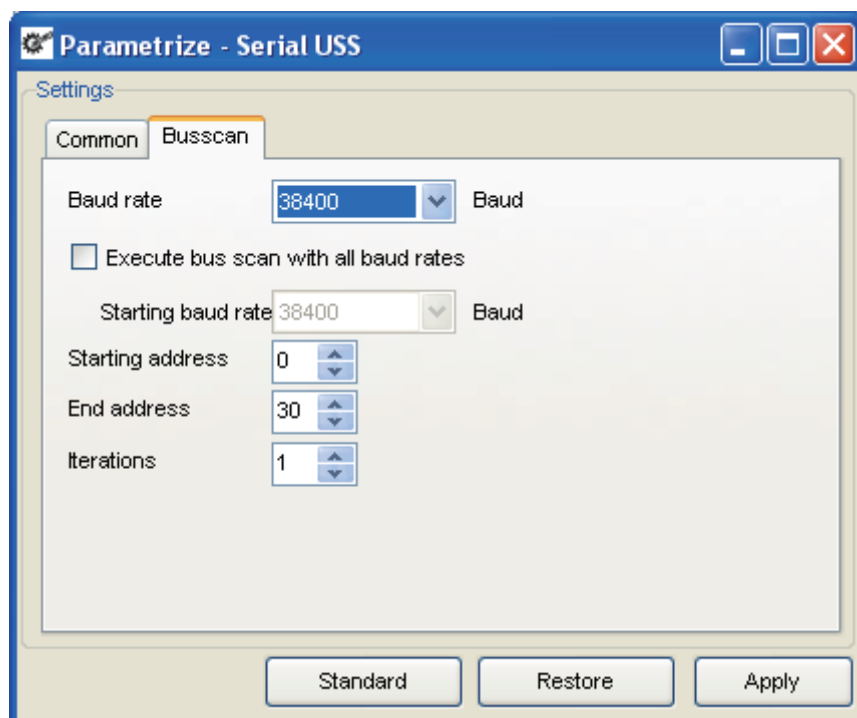
在此字段中，用户可以定义USS地址，以结束对连接设备的搜索。NORD CON无法找到所有地址编号较高的变频器。

停止所有连接的设备

通过该操作，用户可以激活或停用连接设备的停止（禁用）。当此功能激活时，如果设备接口编程为“总线”，则所有启用的设备将停止。

程序启动后自动设备搜索

通过该操作，用户可以在启动程序后激活或停用自动设备搜索。当此功能激活时，启动程序后，NORD CON自动启动总线扫描。



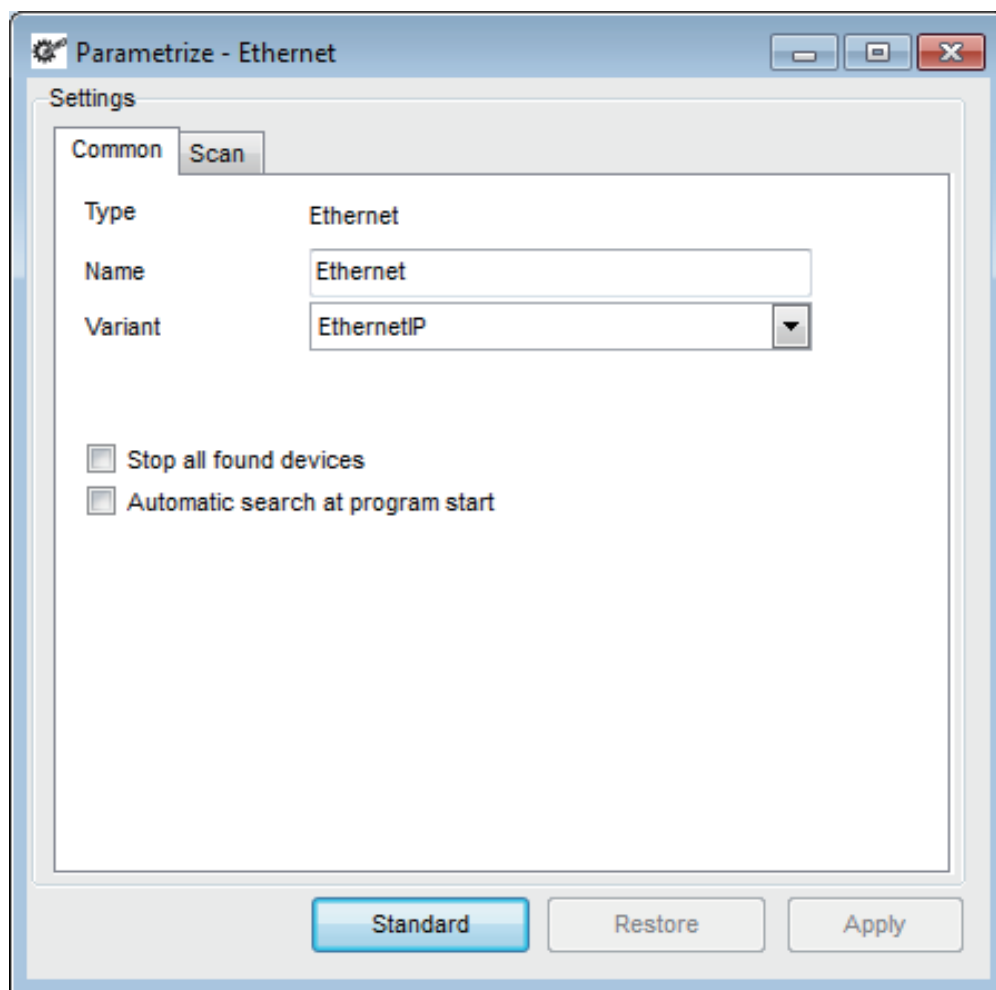
信息

更改

所有更改仅在按下“采用”按钮之后生效。可以使用“恢复”按钮恢复实时有效的设置。

3.2 以太网

3.2.1 一般设置



名称

在编辑字段中，用户可以为通信模块分配一个名称。

类型

在组合框中，用户可以定义通信的类型（PROFINET，EtherCAT或以太网IP）。

停止所有找到的设备

如果启用此选项，则搜索后将向每个检测到的设备发送“禁用”命令，前提是设备可以通过总线控制。

程序启动时自动搜索

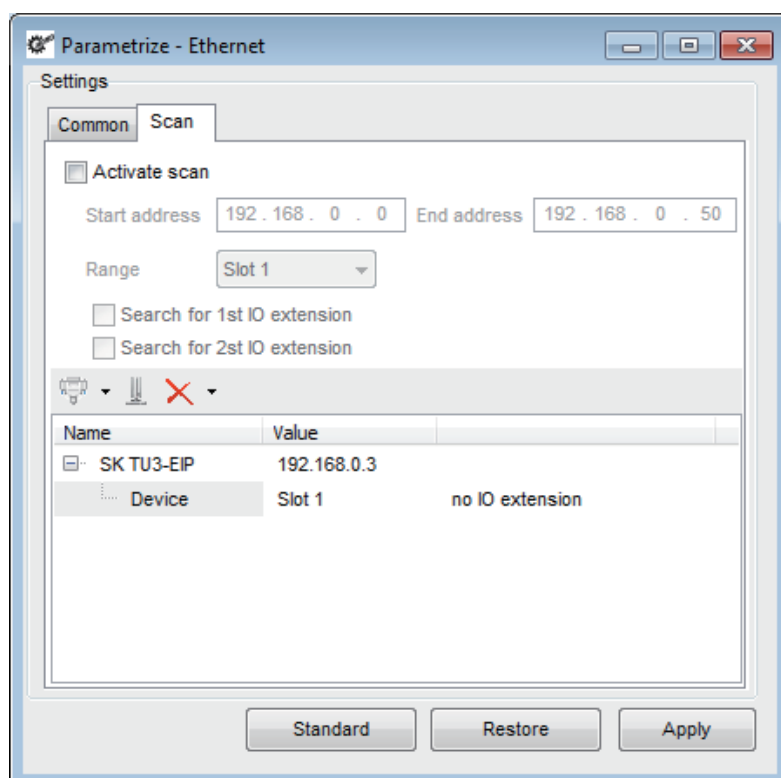
如果启用该选项，则在程序启动后启动设备搜索。

信息

更改

所有更改仅在按下“采用”按钮之后生效。可以使用“恢复”按钮恢复实时有效的设置。

3.2.2 扫描



激活扫描

该选项定义是否启用设备搜索。如果激活搜索，则从起始地址到结束地址搜索所有IP地址的设备。如果在总线扫描中禁用搜索，则使用以下配置。

起始地址

在此编辑框中，必须插入设备搜索的起始地址。

结束地址

在此编辑框中，必须插入设备搜索的结束地址。

添加总线模块

该按钮在设备列表中添加一个新的总线模块。

添加设备

该按钮在设备列表中添加一个新设备。

删除

该按钮去除设备列表中突出显示的条目。

值-总线模块（地址）：

在该列中，必须输入总线模块的IP地址。

值-设备：

在该列中，必须在设备上输入插槽（请参见下表）。

插槽1	插槽2	插槽3	插槽4	插槽5	插槽6	插槽7	插槽8
系统总线地址32或SK 5xxE TU3以上	系统总线地址34	系统总线地址36	系统总线地址38	系统总线地址40	系统总线地址42	系统总线地址44	系统总线地址46

示例：

附加-设备：

在该列中，用户指定IO扩展的数量。

总线模块	插槽1	插槽2	插槽3	插槽4
SK TU3-EIP V1.2 SK TU3-PNT V1.2	SK 5xxE	不适用	不适用	不适用
SK CU4-EIP V1.2 SK TU4-EIP V1.2 SK CU4-PNT V1.2	SK 5xxE SK 2xxE SK 19xE SK 18xE	SK 5xxE SK 2xxE SK 19xE SK 18xE	SK 5xxE SK 2xxE SK 19xE SK 18xE	SK 5xxE SK 2xxE SK 19xE SK 18xE

信息

访问权限

请注意，您需要通过总线模块进行参数更改或控制的访问权限。要进行相应的设置，请检查现场总线模块的数据表。

信息

更改

所有更改仅在按下“采用”按钮之后生效。可以使用“恢复”按钮恢复实时有效的设置。

4 参数化

变频器所有可以更改的参数也可以由NORD CON进行更改。所有参数可以存储和重新传输到变频器。已读取的参数可以打印以存档。

4.1 如何操控参数

读取所选的参数并将值传输到“实时设置”框。数据库确保变频器参数的管理。这些数据库可以在以后存储、打印或再次操控。

信息

“参数化”菜单

只有标记了参数窗口才能显示“参数化”菜单。

NORD CON具有两种参数操控方式：

操作	位置	描述
新建	文件 -> 新建 -> 数据集	当前数据库将重新初始化，也就是说，将删除当前设置和新设置。
打开	文件 -> 打开	任何保存的数据库可以重新打开。
保存	文件 -> 保存	当前数据库以当前名称保存。
另存为...	文件 -> 另存为...	当前数据库以新名称保存。
打印预览...	文件 -> 打印预览...	打印当前参数设置。
读取所有参数或全部读取	参数化 -> 读取 -> 所有参数	读取变频器的所有参数，并将所有参数输入数据库。
读取实际菜单组	参数化 -> 读取 -> 实际菜单组	读取所选菜单组的参数，并将参数输入数据库。
发送新设置	参数化 -> 发送 -> 新值	在“新设置”框中输入新值的所有参数将传输到变频器。可以选择是对所有参数还是仅对属于当前菜单组的参数执行此操作。
发送出厂设置	参数化 -> 发送 -> 复位值	传输的设置将分别是所有参数或当前菜单组参数的默认设置
选择启用	参数化 -> 选择 -> 释放	启用所有参数（或分别包含在当前菜单组中的参数）。
选择禁用	参数化 -> 选择 -> 不释放	不启用任何参数（或属于当前菜单组的参数）。
标准	“标准”按钮	默认值分配给当前选择的参数。

操作	位置	描述
发送	“发送”按钮	当前参数的“新设置”值发送到变频器。
读取	“读取”按钮	读取所选的参数并将值传输到“实时设置”框。

使用自动读取选项，将自动读取所选的参数。

4.2 选择性参数化

NORD CON允许遮蔽某些参数或其它，该特性有助于操控或可用于将参数读取或传输限定为仍未遮蔽或已被筛除的参数。

信息

当激活筛选器时，仅对显示的参数执行所有操作。

在遮蔽任何参数之前，必须禁用启用命令。可以使用参数前面的复选框，或通过4.1“如何操控参数”菜单来完成此操作。

筛选框提供以下设置选项：

- **仅限选择** 仅显示启用的参数（即参数前面的复选框被单击一次的参数）。
- **无标准** 仅显示其值与标准设置不同的参数。
- **信息参数**
 - **是** 显示信息参数。
 - **否** 不显示信息参数。
- **仅限** 仅显示信息参数。

4.3 离线参数化

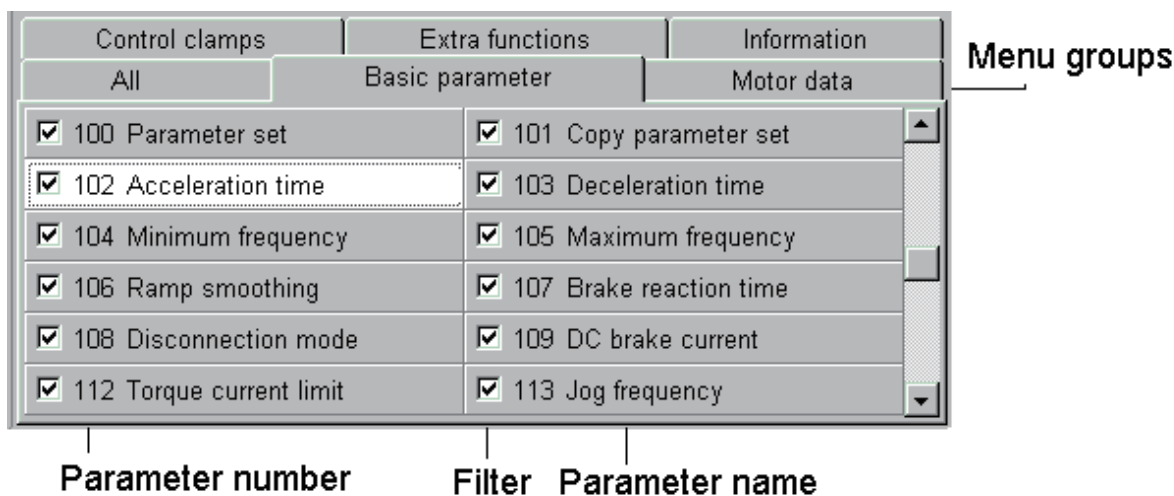
离线参数化意味着操控未分配给任何连接变频器的数据库。

通过主窗口中的数据库菜单启动离线参数化。

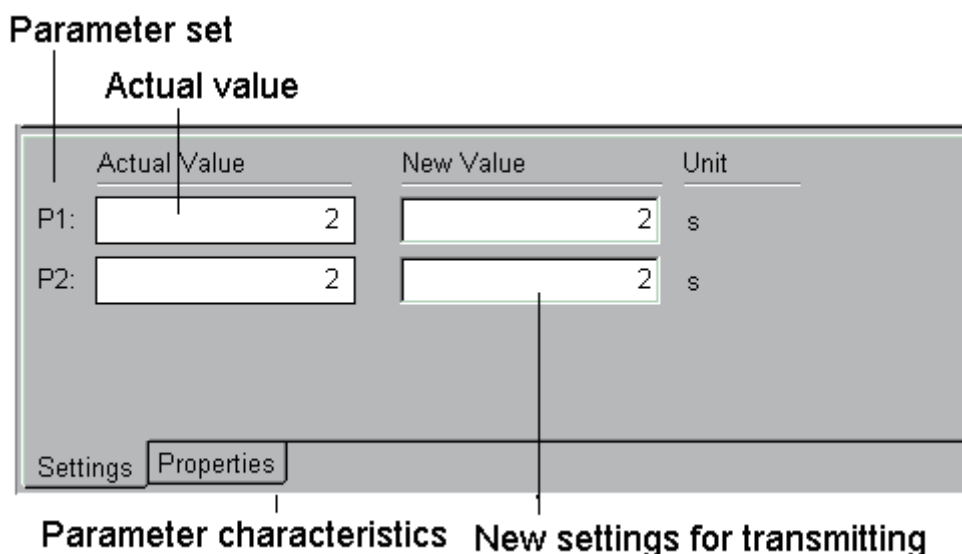
名称	描述
新建	可以创建一个新数据库。新数据库将分配给使用选择框设置的变频器。
打开	可以打开和操控任何读入存储器的数据库。

4.4 参数查看

每个参数都有一个参数名和一个唯一的参数号，通过该参数号可以直接访问该参数。参数分为菜单组。



每个参数都有一个参数值和参数特性：



选择一个参数后，如果可以在参数集中进行不同设置，则将显示所有参数集的值。

4.5 比较报告

该报告显示一个窗口中的两个数据记录之间的异同。基本上，只能比较一个设备系列中的数据记录。参数以列表的形式显示。如果两个参数不同，则该行用灰色条标记。还将检查一个值是否不同于标准值。如果是这种情况，则该值显示为红色。

i 信息

保存数据集

生成报告后，无法再保存数据记录！因此，建议预先保存数据记录。

在线/离线比较

要进行比较，必须连接具有NORD CON的设备。然后必须打开设备的参数窗口，建议读取所有参数。使用筛选器可以进一步限制参数选择。然后可以使用“参数设置->比较”菜单项生成报告。调用该功能后，用户必须选择存储的数据记录进行比较。如果要读取的参数用作备份，则用户必须随后保存当前数据记录。然后生成并显示报告。

i 信息

开放参数集的配置用作参数和标准值的参考。如果选择与设备配置不对应的数据记录，则任何不存在的参数显示为空，并标记为不同。

离线/离线比较

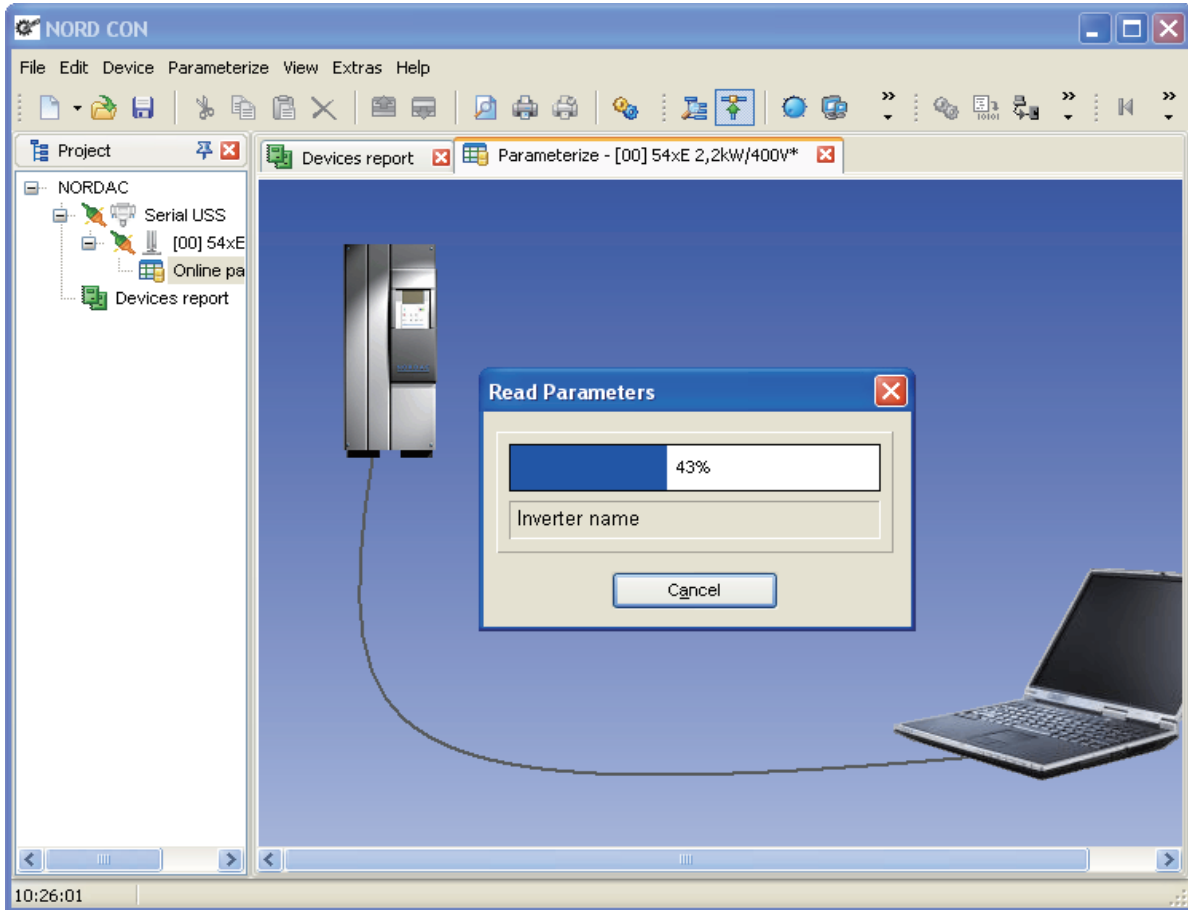
要进行比较，必须打开保存的或新的数据记录。使用筛选器可以进一步限制参数选择。然后可以使用“参数设置->比较”菜单项生成报告。调用该功能后，用户必须选择存储的数据记录进行比较。然后生成并显示报告。

i 信息

开放参数集的配置用作参数和标准值的参考。如果选择与设备配置不对应的数据记录，则任何不存在的参数显示为空，并标记为不同。

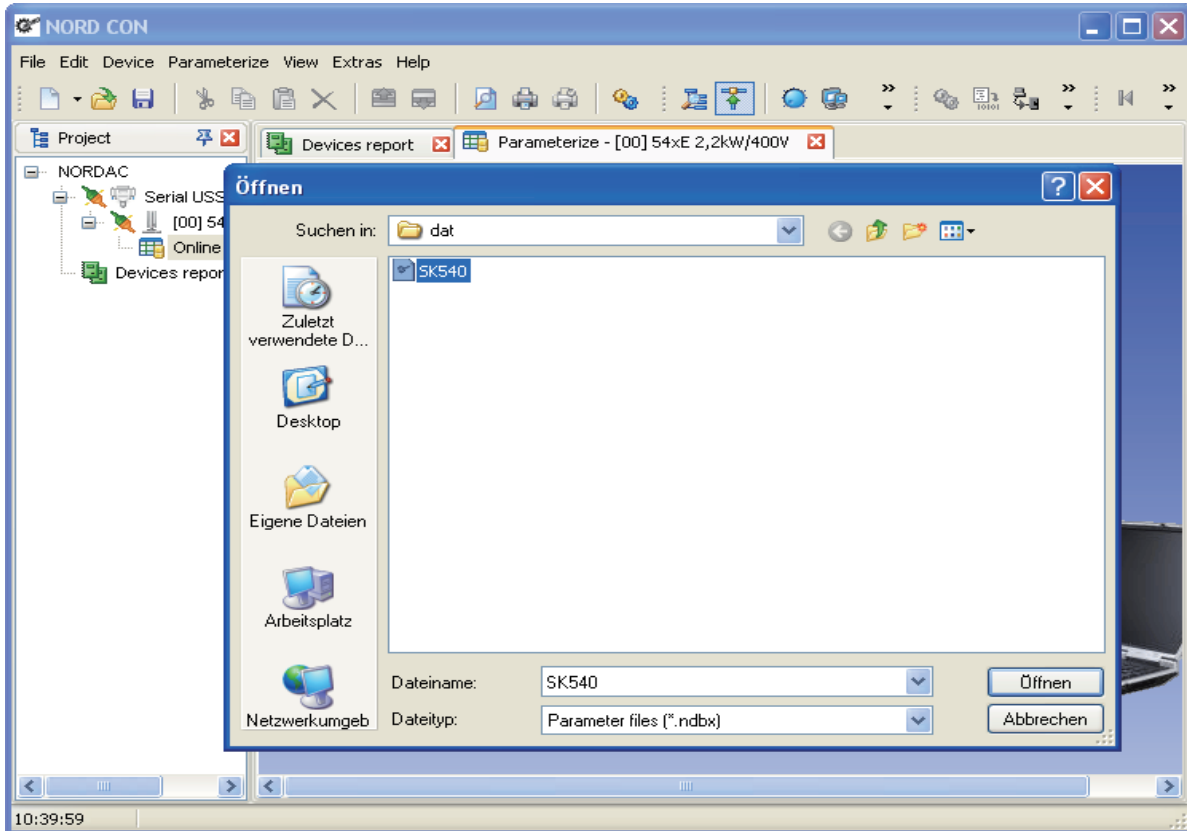
4.6 从设备上传参数

“从设备上传参数”功能将设备参数上传到PC上，然后将值保存在一个参数文件中。可以使用“设备”工具栏或通过“从设备上传设备/参数”菜单项调用此操作。执行该功能后，将打开以下窗口，并自动开始上传参数。如果传输过程中发生通信错误，这些错误将显示在消息窗口中。传输结束时，要求用户输入文件的一个文件名。如果用户通过“保存”确认，则参数被保存。



4.7 下载参数到设备

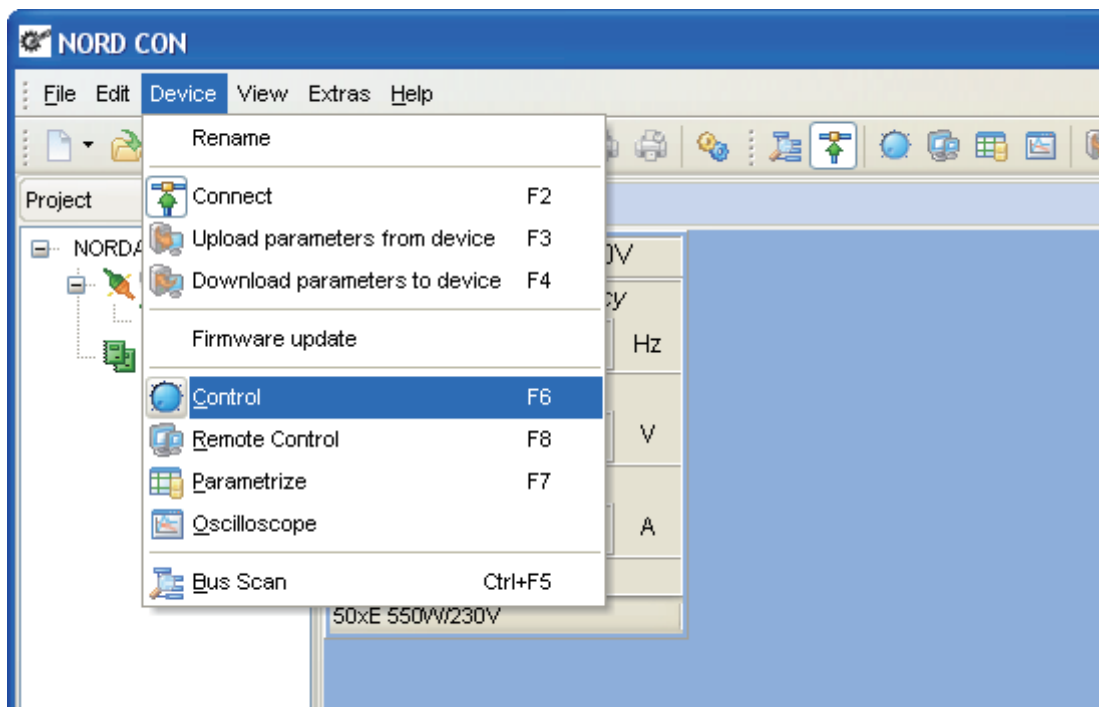
“下载参数到设备”功能可以在PC上打开一个参数文件，并将所有值传输到设备。可以使用“设备”工具栏或通过“设备/下载参数到设备”菜单项调用此操作。执行该功能后，将打开下列窗口和一个文件选择对话框。用户在该对话框中选择一个参数文件，并通过“打开”确认。然后，检查这个参数文件是否适合所选的设备。如果适合，则开始下载。



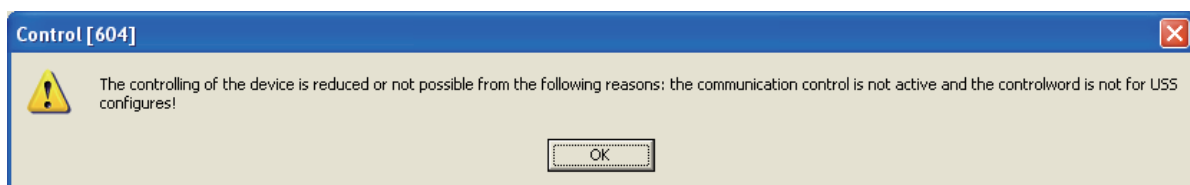
5 控制

5.1 概述

NORD CON程序可用于控制诺德变频器。要使用此功能，变频器必须以正确的方式参数化。由于不同变频器型号的设置不同，用户必须查看使用手册以找到正确的设置。在可以控制变频器之前，必须先完成总线扫描。扫描过程完成后，所有连接的变频器将显示在主窗口中。现在，可以通过单击鼠标来选择要控制的变频器。可以通过主菜单中的“设备/控制 (F6)”或通过弹出菜单（单击鼠标右键）打开“控制”窗口。



现在将读取变频器的控制配置，并通过标准设置进行检查（设置/控制/控制配置检查）。如果变频器的“控制”受到限制或无法进行操作，则屏幕上将出现警告提示。



在“控制”窗口中，有两个版本：

5.2 “标准控制” 可以释放变频器，并且可以增加或减少设置值。

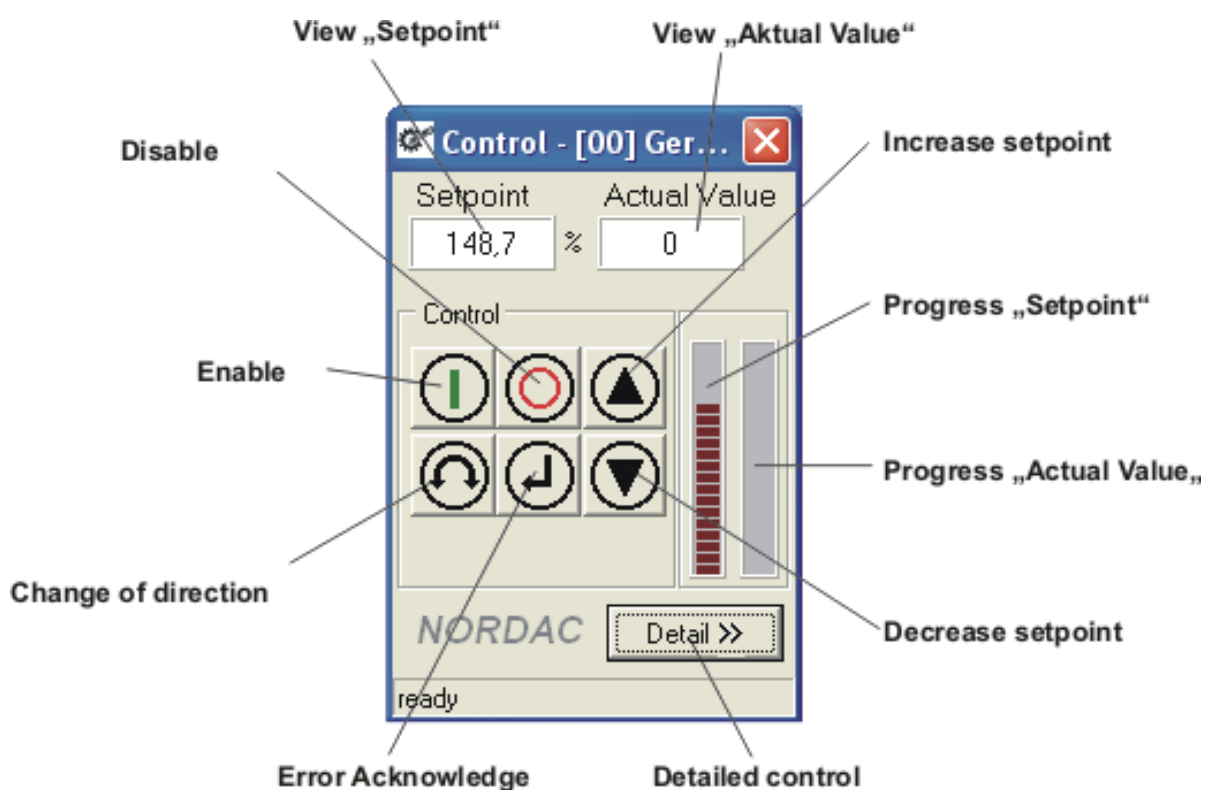
5.3.1 “概述” 所有控制均可与窗口一起使用。

5.2 标准控制

使用标准控制，可以使用以下功能：

- 启用变频器
- 增加或减少设置值
- 更改方向
- 确认错误

要使用此功能，必须对变频器进行编程，以通过总线进行控制。可以在每种型号变频器的使用手册中找到所需的参数和设置。



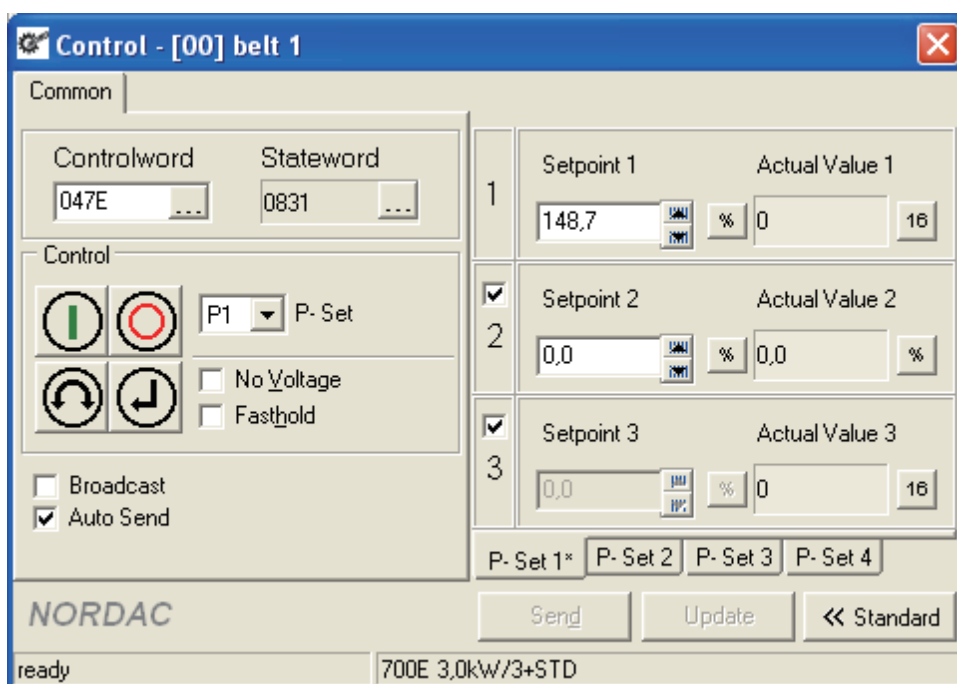
在“标准”显示器上，仅显示第一个设置值和第一个实际值。每种配置的值的形式都是固定的。通过按“详细”按钮，可以切换到扩展控制功能。

5.3 详细控制

5.3.1 概述

在“详细控制”模式下，可以使用一些附加功能：

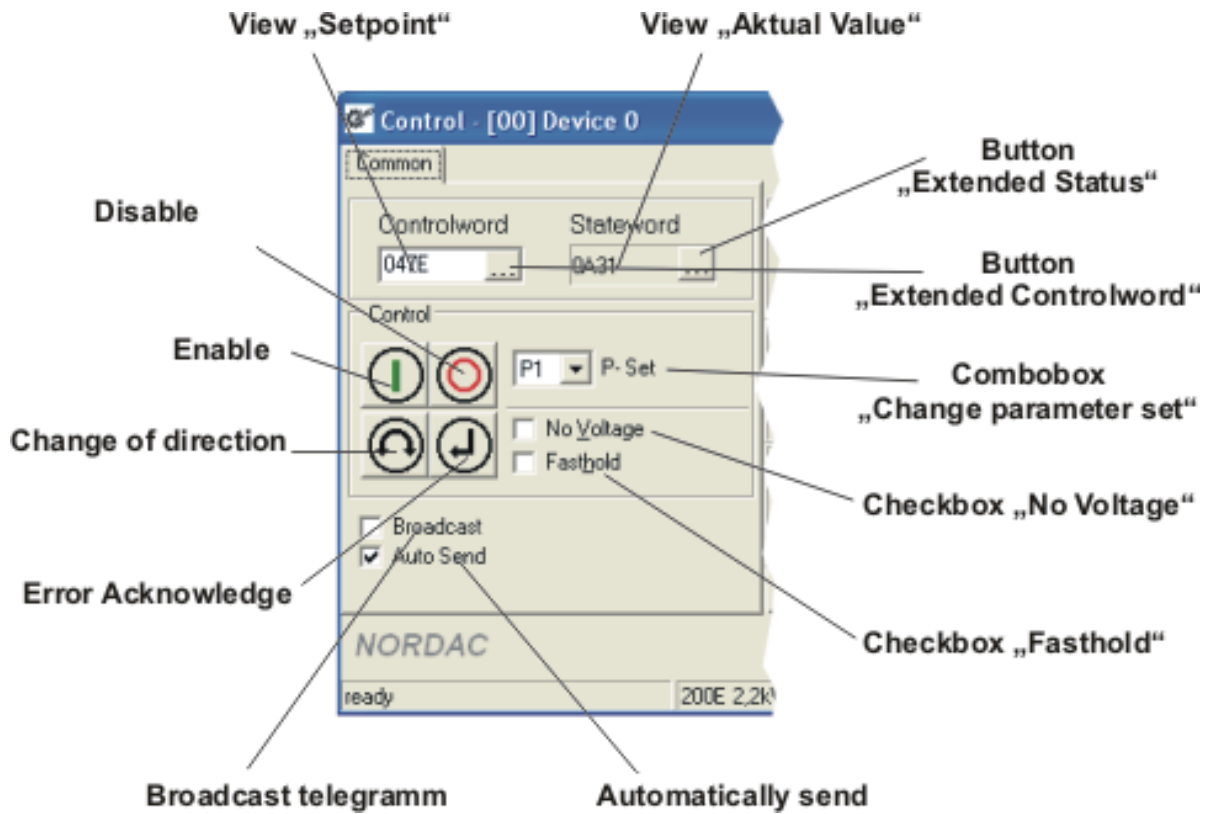
- 5.3.2 “控制”
- 5.3.3 “管理设置值和实际值”
- 发送广播电报
- 选择不同参数集
- 自动发送控制字和设置值



5.3.2 控制

控制字在“控制字”字段中显示为十六进制值。通过输入一个新值（十六进制），用户可以更改控制字。对于控制字的位编码设置，用户可以通过按“控制字编辑”按钮来打开一个新编辑窗口。在该窗口中，控制字以位显示。

状态字在“状态字”屏幕中显示为十六进制值。要以位分辨率显示状态字，可以选择“位定向详细视图”。状态显示以明文形式显示在状态机的状态行中。

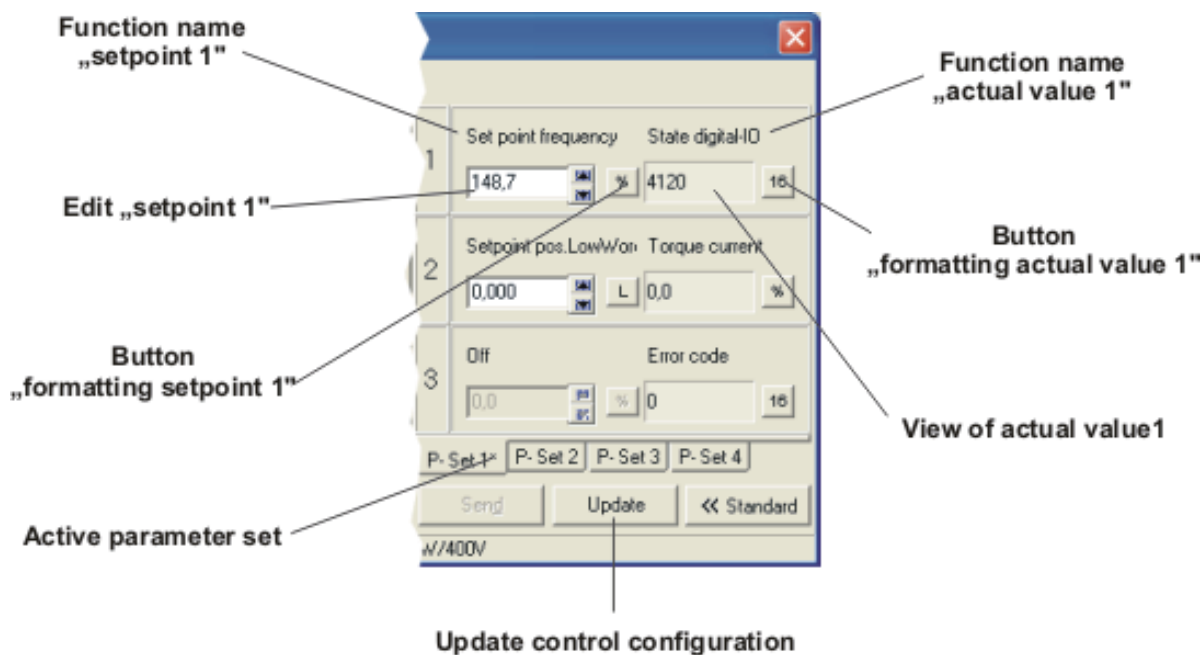


5.3.3 管理设置值和实际值

为了控制变频器，用户可以定义多达3个不同的设定值和实际值（请参见使用手册）。设定值和实际值根据**错误显示！找不到参考源。**“**错误！找不到参考源。**”（按钮“格式化设置值X”）。设定值的输入可以以相同的方式进行。

通过“设置/控制/参数集单独管理”选项，可以单独管理设置值和实际值。因此可以为每个参数集设置设置值。但一个参数集激活时，其设置值将传输到变频器。这是必要的，因为对于每个参数集，可以单独定义设置值和实际值。激活的参数集标有星号。

如果未激活“设置/控制/配置自动检查”选项，用户可以通过按“更新”按钮来传输新配置。



5.3.4 格式化设定值和/或实际值




字符	名称	描述
"%"	16位标准化值	该标准化将设定值/实际值转换为16位标准化值。标准化表示值范围的缩放，在基本值（例如额定频率）的-200%和199%之间。
"16"	16位未标准化	通过这种格式化，设定值和实际值将转换为16位值，并传输到变频器，不带任何缩放进行显示。
"B"	数字输入位	通过这种格式化，设定值和实际值将转换为8位值。位状态单独显示在复选框中。在这些复选框中，可以更改设置值的每一位。
"L"	32位低位字	通过这种格式化，设定值和实际值将作为32位字的低位字（16位）。如果通过格式化“32位高位字”参数化另一个设定值或实际值，则这两个值将在顶部显示中组合。设置值可以为32位值。
"H"	32位高位字	通过这种格式化，设定值和实际值将作为32位字的高位字（16位）。（请参见“32位低位字”）。

5.3.5 状态字

通过“状态字”窗口中的每一位显示当前状态字。表中列出了所有位，包括位号，名称和状态。根据位值和功能，显示彩色LED。

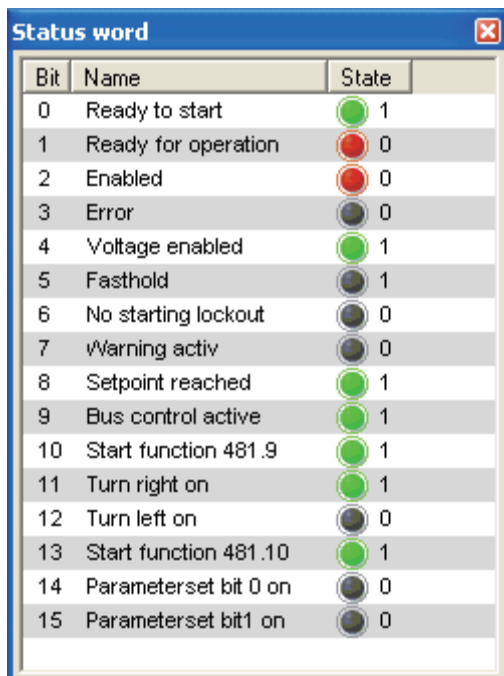
LED的重要性：

















LED	重要性
-----	-----

	设置位和/或启用变频器。
	激活一个错误或丢失一个启用信号。
	未设置位。

通过标准设置，状态字会循环读取，更改会显示在窗口中。要禁用循环读取，请关闭菜单中的“自动”功能（单击鼠标右键）。

窗口停靠在“控制”窗口的左侧。如果窗口要在桌面自由放置，用户应选择弹出菜单“停靠/否”。为了节省空间，窗口可以作为索引卡添加到索引卡“一般信息”旁边。为此，必须将窗口移到索引卡“一般信息”上（按下鼠标左键）。释放按钮后，窗口显示为索引卡。双击索引卡（鼠标左键），用户将返回到窗口模式。






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 控制字

通过“控制字”窗口中的每一位显示当前控制字。表中列出了所有位，包括位号，名称和状态。根据位值和功能，显示彩色LED。如果变频器编程为USS控制，则位由控制按钮设置。控制字的每一个更改将立即发送到变频器（请参见“自动发送”）。

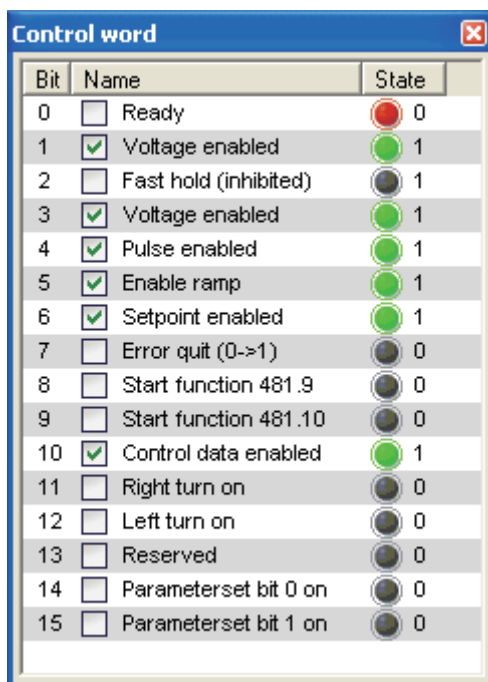
LED的重要性：

LED	重要性
	设置位和/或启用变频器。

	激活一个错误或丢失一个启用信号。
	未设置位。

通过标准设置，控制字会循环读取，更改会显示在窗口中。要禁用循环读取，请关闭菜单中的“自动”功能（单击鼠标右键）。

窗口停靠在“控制”窗口的左侧。如果窗口要在桌面自由放置，应选择弹出菜单“停靠/否”。为了节省空间，窗口可以作为索引卡添加到索引卡“一般信息”旁边。为此，必须将窗口移到索引卡“一般信息”上（按下鼠标左键）。释放按钮后，窗口显示为索引卡。双击索引卡（鼠标左键），用户将返回到窗口模式。



6 远程

NORD CON可以模拟各自变频器的控制装置。为此，变频器将其显示内容传输到NORD CON。关键功能在PC上进行模拟，并传输到变频器。

如果之前未通过控制端子或通过串行接口（P509=0和P510=0）启用变频器，则只能通过远程控制变频器。为此，参数“电位计盒功能”（P549）不得设置为功能{4}“频率添加”或功能{5}“频率减去”。

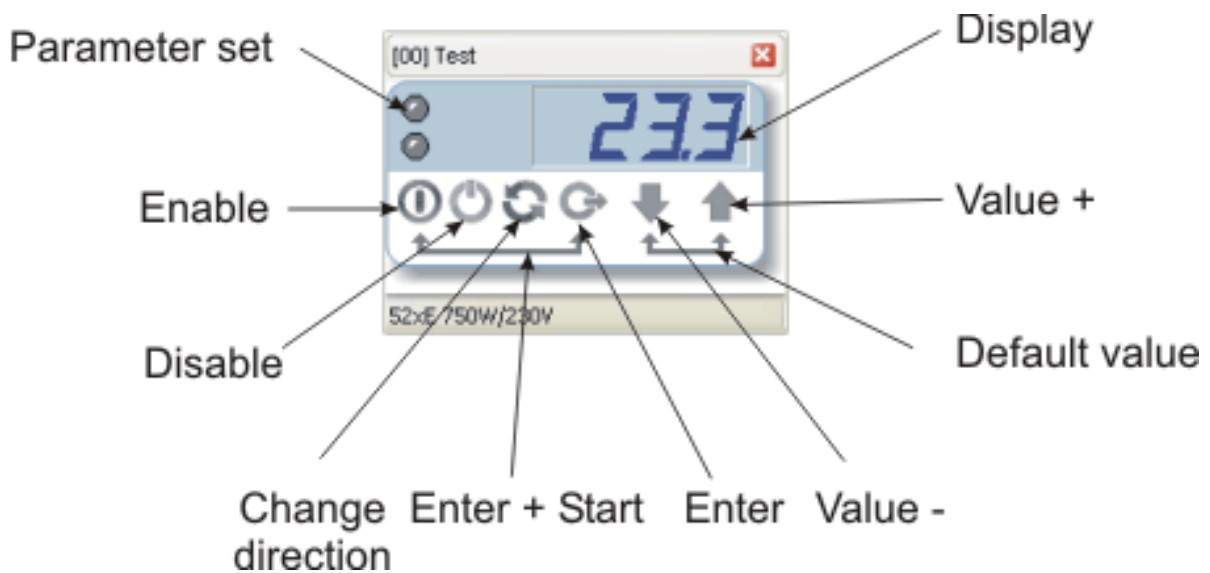
信息




超时监测功能

诺德变频器可以通过键盘控制变频器（启用，设定值+/-，旋转方向等）。在这种情况下，超时监测功能未激活，因此无法取消PC与变频器之间的连接。

6.1 标准

如果未激活“13.1“用户界面””选项，则“远程”功能的标准窗口将用于所有设备。



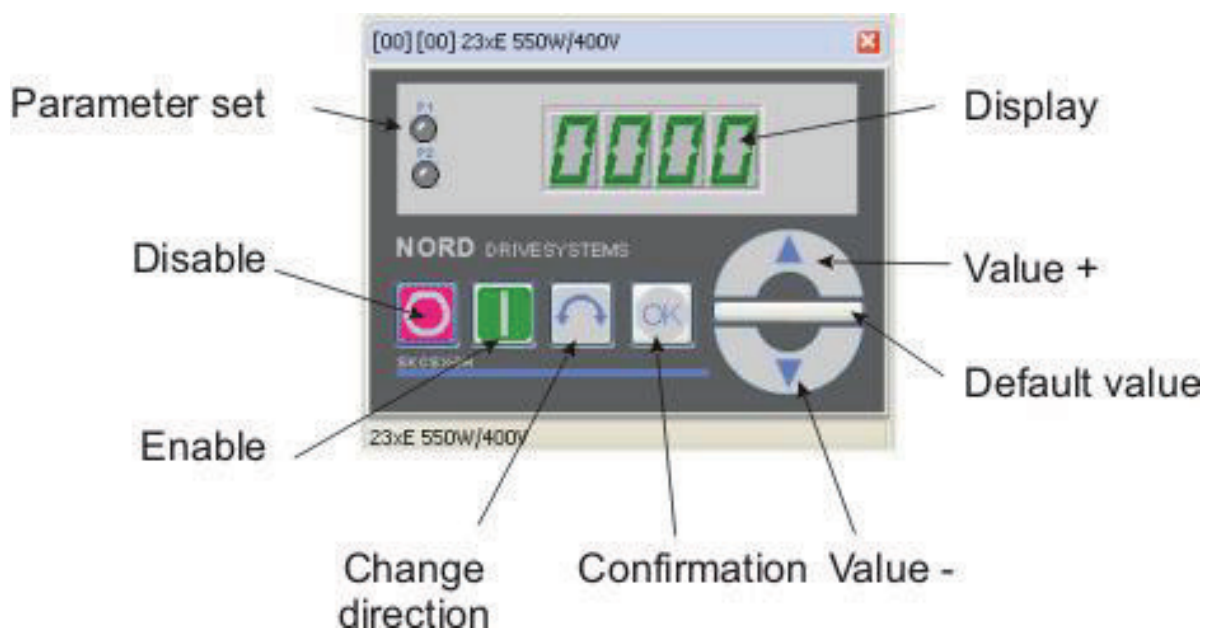
名称	图标	描述
启用		打开变频器。现在，通过设置的点动频率（P113）启用变频器。必须至少提供预设最小频率（P104）。参数 >接口< P509和P510必须 = 0。
禁用		关闭变频器。输出频率降低到绝对最小频率（P505），变频器关闭。
更改方向		按下此键时，电机旋转方向将更改。“向左旋转”用减号表示。 注意：

名称	图标	描述
		在操作泵，螺旋输送机，通风机等时要小心。用参数P540锁住键。
向上	↑	按键增加频率。在参数化过程中，参数号或参数值增加。
向下	↓	按键减小频率。在参数化过程中，参数号或参数值减小。
回车	↻	按“回车键”存储一个更改的参数值，或在参数号或参数值之间切换。 说明： 如果不存储更改的值，则可以使用该键退出参数而不存储更改。
更改方向 + 停止		通过同时按下停止键和“更改方向键”，可以启动快速停机。
回车 + 开始		如果通过“打开”键启动变频器，则可以通过同时按下打开键和回车键进入参数化模式。






可以执行变频器操作单元（控制盒）可用的所有功能。

6.2 NORDAC SK 200 E

NORDAC SK 200E系列变频器的远程控制窗口如下所示：



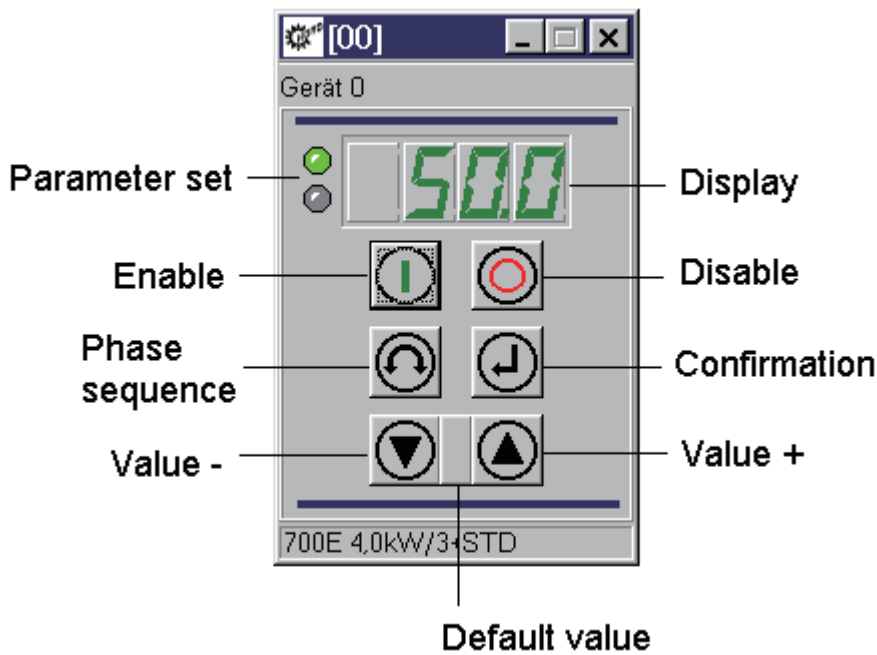
名称	图标	描述
启用	⏻	打开变频器。现在，通过设置的点动频率（P113）启用变频器。必须至少提供预设最小频率（P104）。参数 >接口< P509和P510必须 = 0。







名称	图标	描述
禁用		关闭变频器。输出频率降低到绝对最小频率（P505），变频器关闭。
更改方向		按下此键时，电机旋转方向将更改。“向左旋转”用减号表示。 注意： 在操作泵，螺旋输送机，通风机等时要小心。用参数P540锁住键。
向上		按键增加频率。在参数化过程中，参数号或参数值增加。
向下		按键减小频率。在参数化过程中，参数号或参数值减小。
回车		按“回车键”存储一个更改的参数值，或在参数号或参数值之间切换。 说明： 如果不存储更改的值，则可以使用该键退出参数而不存储更改。
更改方向 + 停止		通过同时按下停止键和“更改方向键”，可以启动快速停机。
回车 + 打开		如果通过“打开”键启动变频器，则可以通过同时按下打开键和回车键进入参数化模式。

可以执行变频器操作单元（控制盒）可用的所有功能。

6.3 NORDAC SK 700/500/300 E

NORDAC SK 700/500/300E系列变频器的远程控制窗口如下所示：

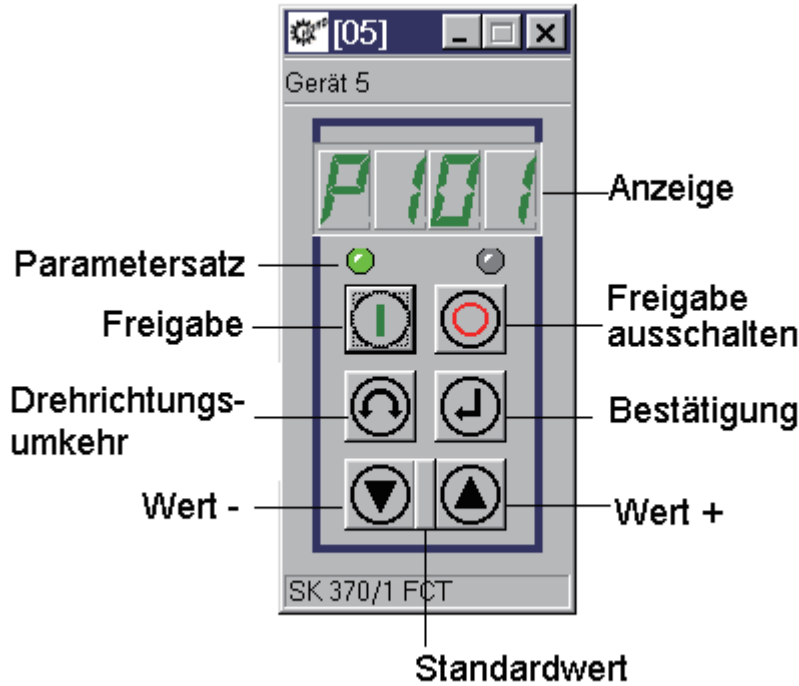








名称	图标	描述
启用		打开变频器。现在，通过设置的点动频率（P113）启用变频器。必须至少提供预设最小频率（P104）。参数 >接口< P509和P510必须 = 0。
禁用		关闭变频器。输出频率降低到绝对最小频率（P505），变频器关闭。
更改方向		按下此键时，电机旋转方向将更改。“向左旋转”用减号表示。 注意： 在操作泵，螺旋输送机，通风机等时要小心。用参数P540锁住键。
向上		按键增加频率。在参数化过程中，参数号或参数值增加。
向下		按键减小频率。在参数化过程中，参数号或参数值减小。
回车		按“回车键”存储一个更改的参数值，或在参数号或参数值之间切换。 说明： 如果不存储更改的值，则可以使用该键退出参数而不存储更改。
更改方向 + 停止		通过同时按下停止键和“更改方向键”，可以启动快速停机。
回车 + 打开		如果通过“打开”键启动变频器，则可以通过同时按下打开键和回车键进入参数化模式。

可以执行变频器操作单元（控制盒）可用的所有功能。

6.4 NORDAC vector mc

NORDAC vector mc系列变频器的远程控制窗口如下所示：



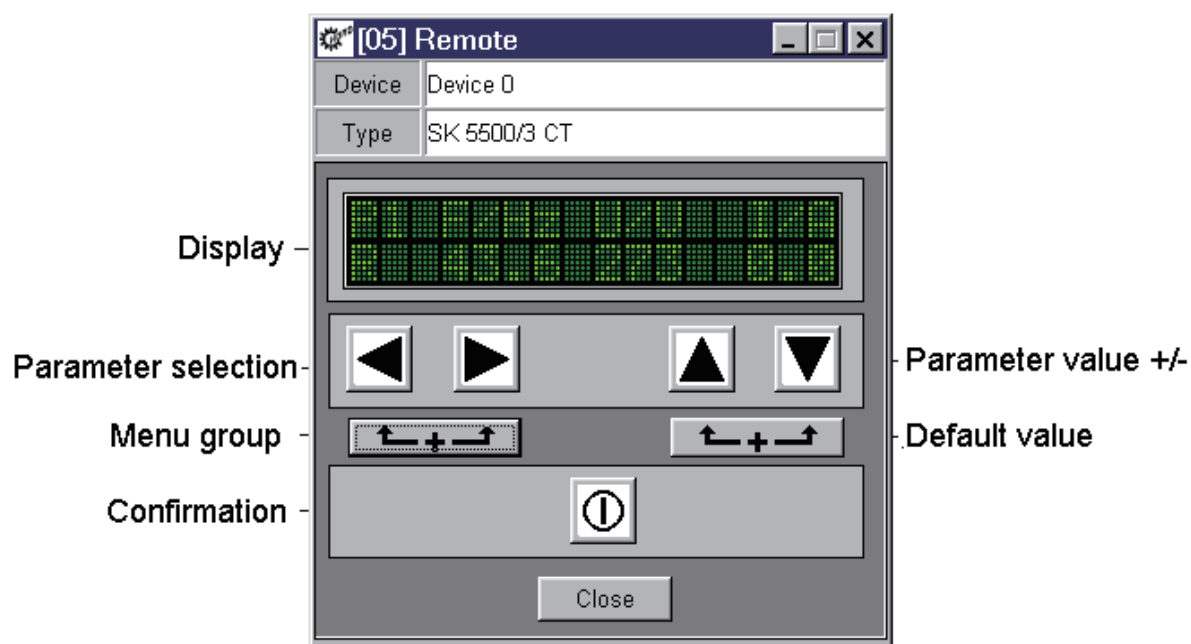
名称	图标	描述
启用		打开变频器。现在，通过设置的点动频率（P113）启用变频器。至少提供预设最小频率（P104）。参数 >接口< P509和P510必须 = 0。
禁用		关闭变频器。输出频率降低到绝对最小频率（P505），变频器关闭。
更改方向		按下此键时，电机旋转方向将更改。“向左旋转”用减号表示。 注意： 在操作泵，螺旋输送机，通风机等时要小心。用参数P540锁住键。
向上		按键增加频率。在参数化过程中，参数号或参数值增加。
向下		按键减小频率。在参数化过程中，参数号或参数值减小。
回车		按“回车键”存储一个更改的参数值，或在参数号或参数值之间切换。




名称	图标	描述
		说明: 如果不存储更改的值，则可以使用该键退出参数而不存储更改。
更改方向 + 停止		通过同时按下停止键和“更改方向键”，可以启动快速停机。
回车 + 打开		如果通过“打开”键启动变频器，则可以通过同时按下打开键和回车键进入参数化模式。




可以执行变频器操作单元（控制盒）可用的所有功能。

6.5 NORDAC vector ct

NORDAC vector ct系列的远程控制窗口具有以下外观：



操作名称	图像	描述
启用		打开变频器。现在，通过设置的点动频率（P113）启用变频器。至少提供预设最小频率（P104）。参数 >接口< P509和P510必须 = 0。
关闭启用		打开变频器。现在，通过设置的点动频率（P113）启用变频器。至少复位最小频率（P104）。参数 >接口< P509和P510必须 = 0。
更改旋转方向		按下此键时，电机旋转方向将更改。“向左旋转”用减号表示。 注意: 在操作泵，螺旋输送机，通风机等时要小心。用参数P540锁住键。

操作名称	图像	描述
增加		按下此键时，电机旋转方向将更改。“向左旋转”用减号表示。 注意： 在操作泵，螺旋输送机，通风机等时要小心。用参数P540锁住键。
减小		按下此键时，电机旋转方向将更改。“向左旋转”用减号表示。 注意： 在操作泵，螺旋输送机，通风机等时要小心。用参数P540锁住键。
确认		按下此键存储一个更改的参数值，或在参数号和参数值之间切换。 说明： 如果不存储更改的值，则可以使用该键退出参数而不保存更改。
旋转方向 + 关闭启用		通过同时按下停止键和“更改方向键”，可以启动快速停机。
确认 + 启用		同时按下打开键和“确认”键可切换到启用设备的编辑模式。

可以执行控制装置（控制盒）可以实现的所有功能。

7 示波器

7.1 概述

集成在NORD CON的示波器效果可以将诺德变频器的过程数据显示为算术图。

i 信息

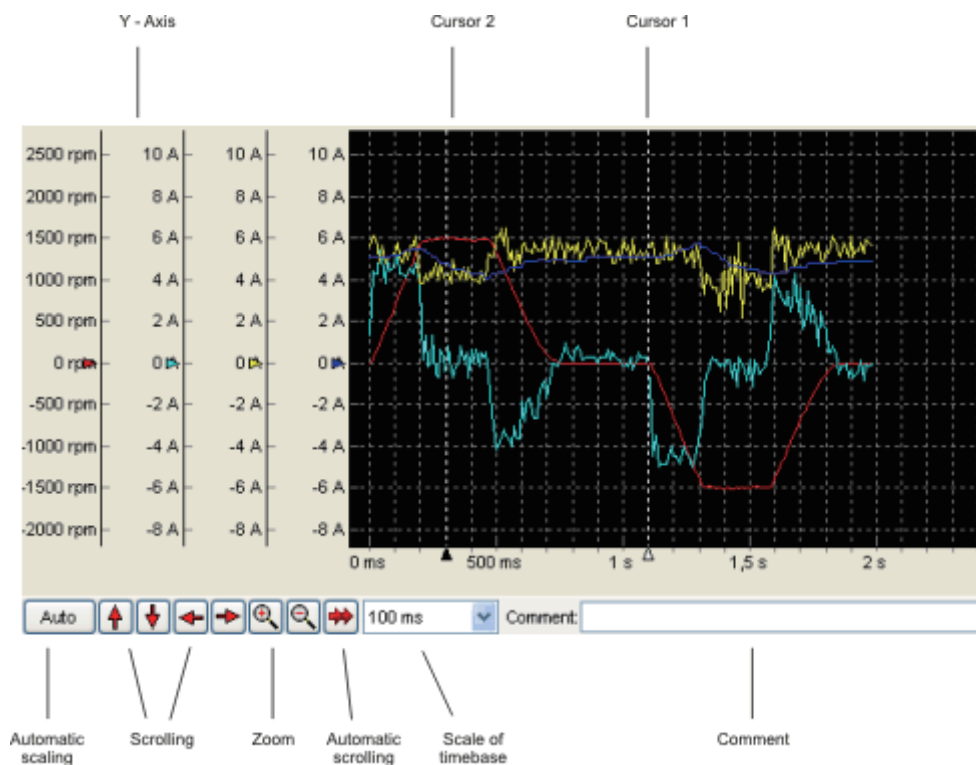
该选项在NORDAC vector ct和NORDAC vector mc系列中不可用！

示波器效果的特性是：

- 监测多达4个通道
- 多种不同的触发方式
- 每次测量的缩放
- 平均值，有效值等的计算
- 保存，打印和导出测量数据

7.2 显示

示波器效果最多可以测量和显示4个通道：



可以进行如下设置：

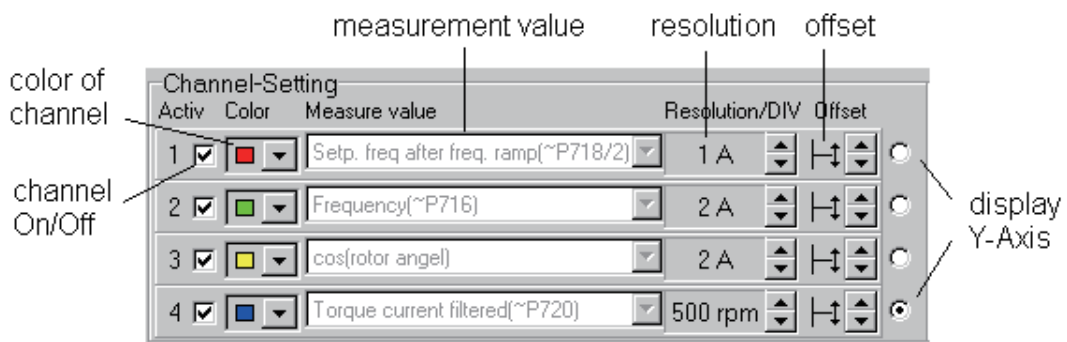
名称	描述
----	----

自动	自动缩放所有测量数据
偏移	选择显示细节（所有数据在x或y方向上的位移）
缩放	显示大小（所有数据的缩放） 说明： 如果鼠标指针在显示器上，则使用鼠标右键，可以在“移动”和“测量”模式之间进行选择。在“移动”模式下，鼠标指针在显示器上移动时可以通过按鼠标左键来选择显示的细节。
自动滚动	使用该选项，在记录期间，时间轴将自动滚动到最后一点。
分辨率	在该组合字段中，用户可以更改时间轴的缩放比例。
注释	附加信息字段，可在其中存储测量系列的更多信息。
光标	执行测量

7.3 处理

按照以下步骤执行测量：

1. 通道选择



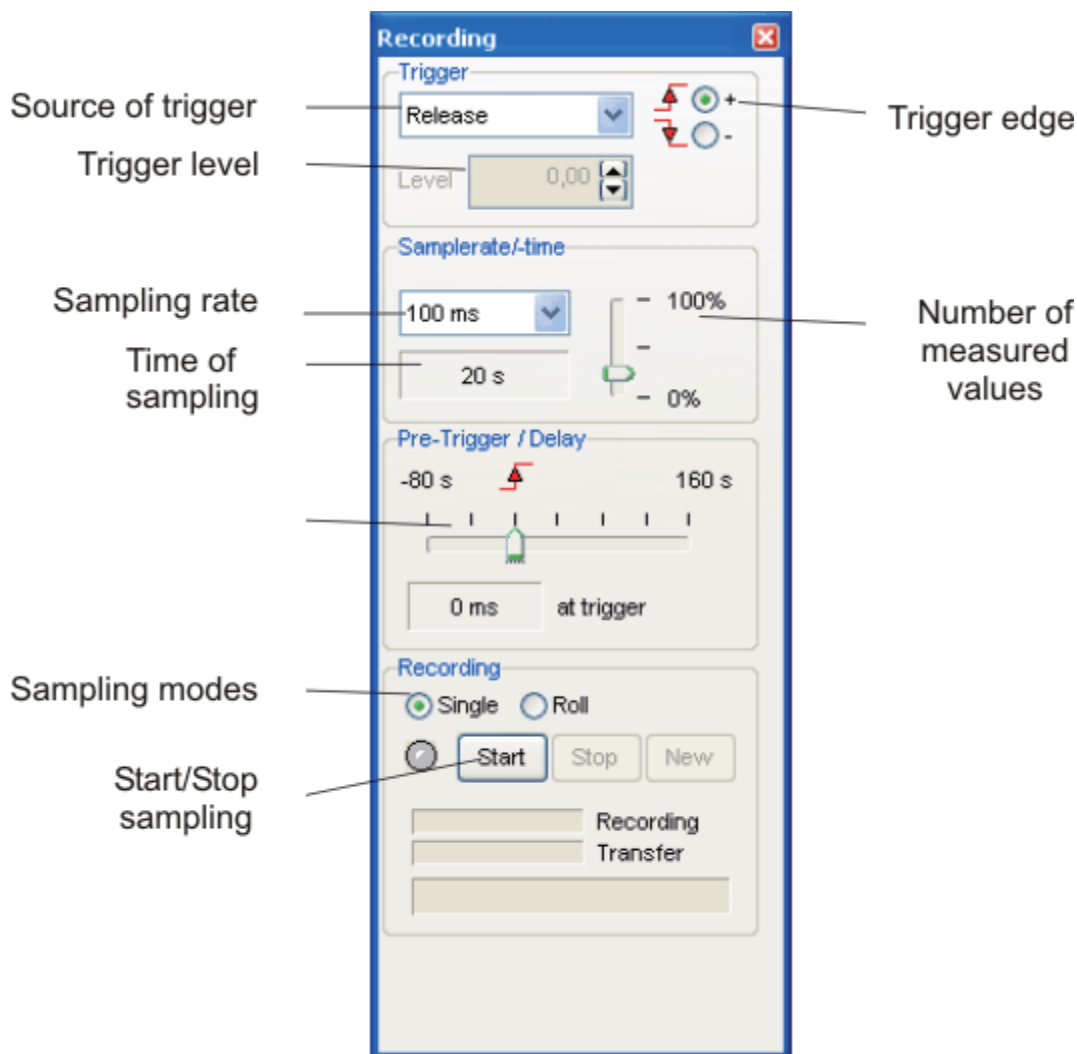
弹出菜单可用于选择4个通道。每个通道有一种颜色。每个通道都可以通过复选框打开和关闭。可以分别为每个通道选择分辨率和偏移量。在显示测量结果时，可以选择和指示每个通道的垂直轴的值。

测量值的重要性

测量值	描述
(=P[数字]) [名称]	此测量功能的值在时隙模式下更新约100 ms，并与参数指示的值相对应。
[名称]	此测量功能的值在时隙模式下更新约100 ms。

(≈P[数字]) [名称]	此测量功能的值在时隙模式下更新约50 ms。
(~P[数字]) [名称]	此测量功能的值在时隙模式下更新约250 μs。

2. 触发设置



触发启动测量。首先选择触发源。触发源可以是测量值、数字输入、变频器状态等。启动条件分别由触发级别和触发边进行定义。

信息

触发级别

触发级别的增量因触发源而不同。因此，并非每个值都能设置。开始记录后，将计算和设置最接近的可能值。

两个测量值之间的时间由采样率设置。测量值的数量和采样率定义了采样时间。预触发/延迟可设置与触发事件相关测量的开始。

信息

测量值

测量值的动态性定义了最佳采样率：快速变化值需要低采样率。测量值的数量定义了将值从变频器发送到NORD CON的时间。

3. 采样模式

示波器有2个不同的模式。用户可以选择“单一”和“滚动”模式。“单一”模式是标准模式。在该模式中，记录从当前触发设置开始。记录时间取决于设备的示波器存储器，最多约2000秒。这些值记录在调整后的采样率中。

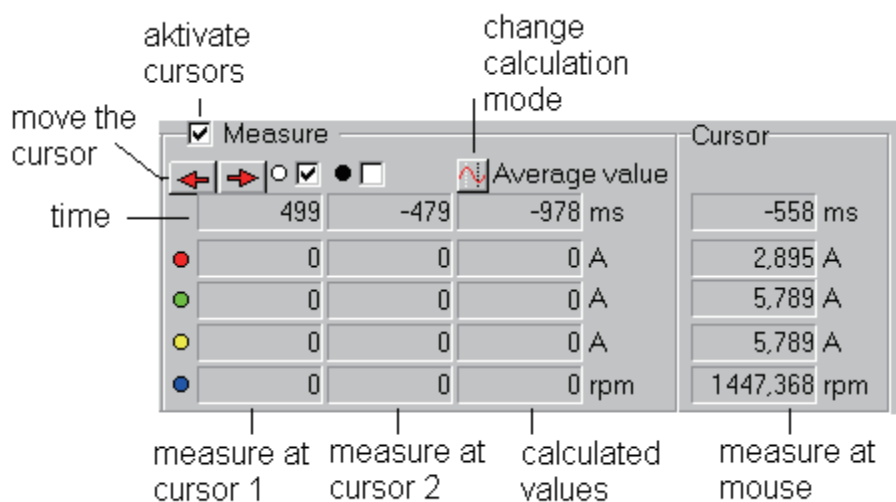
滚动模式使记录时间更长。记录的值会立即传输到PC。因此，用户无法更改采样率。这取决于传输速度。



4. 启动测量

开始按钮激活测量。可以检测到触发事件。事件出现时，在变频器中开始记录。数据传输到NORD CON也在同一时刻开始。这可以通过停止按钮取消。传输所有数据后，可以通过按下“新建”按钮启动新测量或进行新设置。

7.4 测量

完全记录测量后，可以使用光标对结果进行测量。



有两个可用的光标。可以  移动光标。可以通过  选择光标。要通过鼠标右键选择“移动”模式和“测量”，指针必须在显示屏上。在测量模式中，可以通过鼠标左键设置光标。测量线1和2的值显示在光标1和光标2上。另外，进行如平均值的计算。按下“计算”按钮启动计算移位。

7.5 保存和打印

可以保存、导出或打印记录的测量系列。

“文件”菜单项

名称	描述
打开	可以选择和加载存储的测量数据文件。加载期间，可以选择是只加载设置还是加载所有测量数据。
另存为	当前测量数据和设置以一个新的文件名保存。
导出	数据可以导出为图形文件或数据表。
打印	测量线以当前设置（背景色：白色）打印。

示波器记录离线

离线模式下（未连接变频器），可以通过“文件/打开”菜单项加载保存的一个测量文件。

8 宏指令编辑器

宏指令编辑器用于创建简单的过程序列。用户界面提供了使用上下文菜单、工具栏或工具窗口创建和调整宏指令的功能。可以使用拖放在视图中移动各个指令。标准功能，例如保存和加载宏指令，也集成在上下文菜单中。宏指令以标准格式“XML”存储。可以使用“打开”菜单项，文件类型“宏指令文件 V1.26”导入上一版本的格式。

8.1 用户界面和视图

除了编辑器窗口之外，处理宏指令生成器还需要其它视图。这些视图可用作工具窗口。这些窗口可以停靠和取消停靠在主窗口的边缘。可以使用弹出菜单中的“视图”菜单项显示和关闭所有视图。

8.1.1 “变量”窗口

可通过“视图->宏指令->变量”菜单选项来打开和关闭“变量”视图。这用于调试。启动宏指令后，在此窗口中显示具有当前额定值的所有变量和对象的宏指令。可以在“属性->显示格式”视图中停止值的消耗。

有以下格式：

- 十进制
- 十六进制
- 二进制

8.1.2 属性窗口

可以通过“视图->属性”菜单项打开和关闭“属性”视图。当前指示的所有属性都显示在该窗口中。根据指令，可以更改属性的数量和类型。

名称	描述
结果	使用该属性可以更改要为其分配一个新值的对象。只能选择可以分配一个新值的对象（例如控制字，参数或变量）。
操作数	使用该属性，用户可以选择要与分配或操作一起使用的对象。
运算符	该属性定义了操作的类型（例如加法）。
注释	用户可以使用该属性为任何指令分配注释。

变量、控制或状态字、设定值或实际值或参数可以在宏指令生成器中指定为对象。每个对象有不同的参数。

对象	参数	描述
变量	名称	参数定义了变量或常量的名称。所有已使用的变量将显示在选择框中。如果要创建一个新的变量，则必须输入还未使用的一个名称。大写和小写之间没有区别。
	显示格式	该参数定义“变量”视图中的显示格式。可选择以下显示之一： <ul style="list-style-type: none"> • 十进制 • 十六进制 • 二进制
常量	值	该参数定义了常量的值。
	显示格式	该参数定义“变量”视图中的显示格式。可选择以下显示之一： <ul style="list-style-type: none"> • 十进制 • 十六进制 • 二进制
控制字， 状态字	节点数	该参数定义了所需设备的USS节点数。 说明： 由于当前控制字不能从设备读取，因此当启动调度器时，控制字设置为0。
	显示格式	该参数定义“变量”视图中的显示格式。可选择以下显示之一： <ul style="list-style-type: none"> • 十进制 • 十六进制 • 二进制
设定值和 实际值	节点数	该参数定义了所需设备的USS节点数。 说明： 由于当前设定值不能从设备读取，因此当启动调度器时，值设置为0。
	型号	该参数定义了值的类型。“设定值和实际值类型”表中列出的类型可供用户使用。
	格式	该参数定义了设定值和实际值的格式化。可能的格式显示在“设定值和实际值格式化”表中。
	分辨率	该参数定义了设定值和实际值的分辨率。用于在编辑器中格式化指令。
	显示格式	该参数定义“变量”视图中的显示格式。可选择以下显示之一： <ul style="list-style-type: none"> • 十进制 • 十六进制 • 二进制
参数	节点数	该参数定义了所需设备的USS节点数。

对象	参数	描述
	参数号	该值定义了参数的数目（请参见“设备目录”视图）。
	子索引	该值定义了参数的子索引。
	分辨率	该值定义了设定值和实际值的分辨率。用于在编辑器中格式化指令。
	数据类型	该值定义了参数的数据类型。当前设备只能使用2种数据类型（16位整数和32位整数）。
	显示格式	该参数定义“变量”视图中的显示格式。可以选择以下显示之一： <ul style="list-style-type: none"> • 十进制 • 十六进制 • 二进制

设定值和实际值类型

型号	描述
值1（16位）	应该使用第一、第二或第三设定值或实际值。
值12（32位）	第一和第二设定值或实际值应该用作32位值。 说明： 必须为此配置对设备进行适当配置（请参见“设定值或实际值配置”）。
值13（32位）	第一和第三设定值或实际值应该用作32位值。 说明： 必须为此配置对设备进行适当配置（请参见“设定值或实际值格式化”）。
值23（32位）	第二和第三设定值或实际值应该用作32位值。 说明： 必须为此配置对设备进行适当配置（请参见“设定值或实际值格式化”）。

设定值和实际值格式化

格式化	描述
标准化	该格式化将设定值/实际值解释为16位标准化值。标准化表示值范围的缩放，在基本值（例如额定频率）的-200%和199%之间。
未标准化	在该格式化中，将设定值或实际值解释为16位值，该值传输到设备并在不缩放的情况下显示。

格式化	描述
低位字 (32位)	该格式化定义了第一字为低位字，第二值为高位字值 (32位)。只能为32位类型选择该值。
高位字 (32位)	该格式化定义了第一字为高位字，第二值为低位字值 (32位)。只能为32位类型选择该值。

说明：

请确保设备的配置与设置相对应。

8.1.3 日志窗口

序列控制中的所有事件都保存在日志中。要显示日志，需要使用“视图->日志”菜单项打开“日志”视图。值也是工具窗口，可以停靠和取消停靠在主窗口的边缘。所有日志条目都显示在窗口的排序列表中。在这种情况下，最后一个条目位于列表的开头。

保存日志

可以使用弹出菜单中的“另存为...”菜单项保存日志。然后打开文件选择对话框，用户必须规定用于存储日志文件的名称和路径。如果用户以“保存”确认，则当前列表保存在文本文件中。

删除日志

可以使用弹出菜单中的“删除”菜单项删除日志。然后，所有条目将被删除且不可挽回。

筛选条目

用户可以使用筛选功能根据条目类型筛选日志条目。可以使用“筛选”菜单项定义要在日志中输入的条目类型。

8.2 使用宏指令

8.2.1 创建新宏指令

新文档（宏指令）由上下文菜单中的“新建”菜单项生成。根据之前是否打开过文档，宏指令编辑器会提供对旧文档的存储。如果用户未以“取消”进行确认，则将生成一个新文档。当前版本在同一时间只能打开一个文档。

8.2.2 打开宏指令

可在“打开”菜单项或使用“Ctrl+O”组合键打开宏指令。随后，将打开文件选择对话框，用户可在对话框中选择所需的宏指令。如果用户想要打开宏指令的以前版本，则必须在文件选择对话框中对数据类型进行相应的更改。

8.2.3 保存宏指令

可在“保存”菜单项或使用“Ctrl+S”组合键存储宏指令。但是该功能仅可用于之前生成的文档。对于所有新文档，必须使用“另存为...”来执行该功能。

在“另存为...”菜单项中执行该功能。随后，将打开文件选择对话框，用户必须在对话框中选择文件名以及路径。以“保存”进行确认后，将存储宏指令。过程完成后，新命名的宏指令显示在标题栏中。

8.2.4 插入指令

使用“插入”菜单项或“Ctrl+V”组合键激活“插入”功能。可在文档的当前位置下插入之前复制或剪切的指令。如果预先未复制或剪切任何指令，则停用该菜单项。在当前版本中，每个复制或剪切的指令只能插入一次。

8.2.5 复制指令

使用“复制”菜单项或“Ctrl+C”组合键激活“复制”功能。可将所选的行复制到生成器的剪贴板中。当前版本只能选择一行。这意味着只能复制一条指令。块指令是个例外。只能全部复制。

8.2.6 剪切指令

使用“剪切”菜单项或“Ctrl+X”组合键激活“剪切”功能。可将所选的指令复制到生成器的剪贴板中。当插入剪切的指令时，旧指令从文档中删除。只能剪切一条指令的限制也适用此功能。

8.2.7 从指令中删除

可在“删除”菜单项或使用“Ctrl+Del”组合键执行该功能。从文档中删除标记的指令。

8.2.8 搜索和替换

可在“搜索和替换”菜单项或使用“Ctrl+H”组合键执行“搜索和替换”功能，此时将打开“搜索和替换”对话框。这允许用户插入搜索和替换词汇并启动更改过程。

8.2.9 上移指令

在“上移”菜单项中执行该功能。将标记的指令上移一行。如果文档的顶行作了标记，则不执行任何操作。还可以通过拖放鼠标移动指令。

8.2.10 下移指令

在“下移”菜单项中执行该功能。将标记的指令下移一行。如果文档的最后一行作了标记，则不执行任何操作。还可以通过拖放鼠标移动指令。

8.2.11 创建新指令

使用上下文菜单中的“功能”菜单项创建新指令。始终在所选行的下面插入新指令。然后，用户可以更改新指令的位置（请参见“向上”和“向下”）。

用户可以使用此版本中的下列功能：

名称	描述
分配	<p>指令将一个值分配给宏指令对象。新值可以从另一对象中读取，或用户定义常量。默认情况下，如示例1所示插入行。可以在“属性”视图中调整参数功能。</p> <p>示例： 设备00控制字 = 047F hex // 将1151值分配给控制字 变量1 = 设备00状态字 // 将状态字值分配给变量</p> <p>说明： 设定值只能在块指令内进行分配。</p>
跳转标记	<p>指令定义了宏指令中的跳转点。用户可以使用“转到”功能跳转到跳转标记的位置。默认情况下，如示例1所示插入行。可以在“属性”视图中调整参数功能。必须更改跳转点的名称，因为不支持重复的名称。生成器始终跳转到宏指令的第一个跳转标记。</p> <p>示例： 标签1: // 定义“标签1”跳转标记</p> <p>或 启动: // 定义“启动”跳转标记</p>
睡眠	<p>指令在执行宏指令过程中生成暂停。以“ms”为单位指定时间。默认情况下，如示例1所示插入行。可以在“属性”视图中调整时间。</p> <p>示例： 睡眠1000 ms // 等待1s</p> <p>或 睡眠500 ms // 等待0.5s</p>
转到	<p>指令生成宏指令中的跳转。执行了指令后，生成器以相关名称跳转到跳转标记行。如果生成器无法以该名称找到跳转标记，则忽略该行。如果尚未在宏指令中定义跳转标记，则将停用菜单项。默认情况下，始终输入第一个跳转标记。可以在“属性”视图中调整跳转标记的名称。</p> <p>示例： 转到启动 // 转到跳转标记“启动”</p>
条件	<p>指令生成宏指令中的条件跳转。如果条件是真实的，则生成器以相关名称跳转到跳转标记行。默认情况下，如示例1所示插入行。可以在“属性”视图中调整指令的参数。</p> <p>示例： 如果设备00控制字 == 047F hex则 // 如果控制字有1150值 转到启动 // 转到跳转标记“启动”</p>
块	<p>该指令允许用户在一个指令中执行多个指令。</p>

名称	描述
	<p>这些分配仅限于“控制字”和“设定值”对象。根据设备的配置和使用目的，用户可在“1个设定值的控制值”，“2个设定值的控制值”或“3个设定值的控制值”之间进行选择。</p> <p>示例： 块 // 使用1 USS协议发送控制字和设定值1 设备00控制字 = 1151 // 将1150值分配到控制字 设备00设定值1 = 20.0 // 将20值分配到设定值1</p>
数学和逻辑运算	<p>这些指令使得用户可以对对象进行简单的数学和逻辑运算。然后将新计算的分配给对象。可以在“属性”视图中调整指令的参数。</p> <p>示例： 变量1 = 设备00控制字 + 047F hex // 添加 变量1 = 设备00控制字 + 047F hex // 添加</p>

8.3 调度器

自动

激活该选项（自动模式），启动调度器后，将进行逐行处理。如果停用（单步模式）（“下一步”菜单条目或“F12”组合键），则用户必须手动运行每个指令。

环路

激活该选项后，宏指令在连续环路中执行。这意味着在完成最后一条指令后，调度器将跳回到宏指令的开头。

8.3.1 启动序列

使用“启动”菜单项或“F9”组合键启动调度器。如果自动模式激活，则进行逐行处理。在单步模式下，启动后仅执行第一行。用户必须为每个后续行调用“下一步”操作。调度器仅在完成宏指令或用户终止运行后才能再次启动。调度器运行时，无法编辑指令参数。

8.3.2 取消宏指令

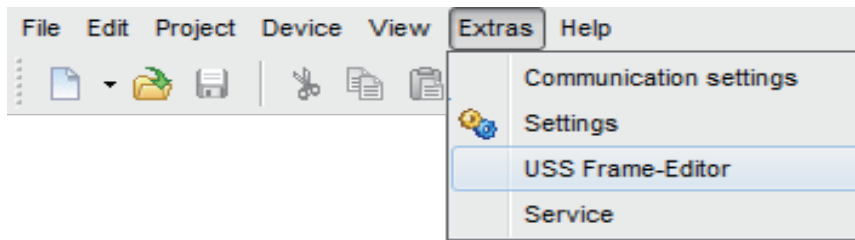
可在“取消”菜单项或使用“F11”组合键终止调度器。

8.3.3 执行下一个指令

可在“下一步”菜单项或使用“F12”键找到该功能。该功能仅在单步模式下可用，并指示调度器执行宏指令中的下一条指令。如果执行了最后一条指令，则调度器将自动终止。

9 USS帧编辑器

USS协议根据主机/从机原理定义访问过程，以通过串行总线进行通信。其子集还包括点对点连接。一台主机和最多31台从机可以连接到总线。主机通过电报中地址字符访问单个从机。单个从机之间无法直接进行消息交换。在半双工模式下，使用USS电报进行通信。



USS帧编辑器用于生成和分析USS电报。完全集成在NORDCON用户界面中，并通过“附加/USS帧编辑器”菜单项打开。编辑器以多个视图显示主机和从机电报。通过选项卡，用户可以在9.1“主机（指令）”从机-电报（USS Antwort）之间切换。

对象	描述																					
电报类型	<p>该对象指定了USS电报的大小和结构。变频器支持以下类型：</p> <table border="1"> <thead> <tr> <th>型号</th> <th>长度 (LGE)</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>PPO 0</td> <td>12</td> <td>带过程数据和16位参数值的标准电报</td> </tr> <tr> <td>PPO 1</td> <td>14</td> <td>带32位参数值和过程数据的扩展参数数据电报</td> </tr> <tr> <td>PPO 2</td> <td>18</td> <td>带扩展过程数据（主设定值和两个辅助设定值）和32位参数值的电报</td> </tr> <tr> <td>PPO 3</td> <td>6</td> <td>带主设定值但不带参数数据的过程数据电报</td> </tr> <tr> <td>PPO 4</td> <td>10</td> <td>带主设定值和辅助设定值但不带参数数据的扩展过程数据电报</td> </tr> <tr> <td>PPO 6</td> <td>16</td> <td>带5个设定值/实际值的电报</td> </tr> </tbody> </table> <p>注意： 所有变频器都不支持该电报类型。</p>	型号	长度 (LGE)	描述	PPO 0	12	带过程数据和16位参数值的标准电报	PPO 1	14	带32位参数值和过程数据的扩展参数数据电报	PPO 2	18	带扩展过程数据（主设定值和两个辅助设定值）和32位参数值的电报	PPO 3	6	带主设定值但不带参数数据的过程数据电报	PPO 4	10	带主设定值和辅助设定值但不带参数数据的扩展过程数据电报	PPO 6	16	带5个设定值/实际值的电报
型号	长度 (LGE)	描述																				
PPO 0	12	带过程数据和16位参数值的标准电报																				
PPO 1	14	带32位参数值和过程数据的扩展参数数据电报																				
PPO 2	18	带扩展过程数据（主设定值和两个辅助设定值）和32位参数值的电报																				
PPO 3	6	带主设定值但不带参数数据的过程数据电报																				
PPO 4	10	带主设定值和辅助设定值但不带参数数据的扩展过程数据电报																				
PPO 6	16	带5个设定值/实际值的电报																				
地址	该对象包含访问的变频器的地址。																					

9.2 “设备（响应）” Zustand swort</dg_ref_source_inline>	该对象包含变频器的状态位。
5.3.5 “状态字” Steuerwort</dg_ref_source_inline>	该对象包含控制位（例如启用或快速停机）。
设定值/实际值1-5	设定值/实际值为16位或32位值。根据变频器的参数化，这些代表不同的值（例如频率设定值或位置设定值）。
格式	<p>该对象包含设定值的格式。支持以下格式：</p> <ul style="list-style-type: none"> 16位标准值 该格式化将设定值解释为16位标准化值。标准化表示值范围的缩放，在基本值（例如额定频率）的-200%和199%之间。 16位非标准化 在该格式中，将设定值解释为16位值，该值传输到变频器并在不缩放的情况下显示。
参数顺序	<p>该对象包含参数顺序。定义了以下顺序：</p> <ul style="list-style-type: none"> • 要求参数值 • 更改参数值（16位） • 更改参数值（32位） • 要求参数值（数组） • 更改参数值（数组16位） • 更改参数值（数组32位） • 要求数组元素的数量 • 更改参数值（数组双字）无需写入EEPROM • 更改参数值（数组字）无需写入EEPROM • 更改参数值（双字）无需写入EEPROM • 更改参数值（字）无需写入EEPROM
参数号	该对象包含参数号。
索引	该对象包含参数索引。
值	该对象包含参数值。根据电报类型，为16或32位值。值的显示仍取决于值的分辨率。
分辨率	该对象包含参数的分辨率。如果更改分辨率，则仅更改参数值的显示。有关分辨率值，请参阅变频器说明。

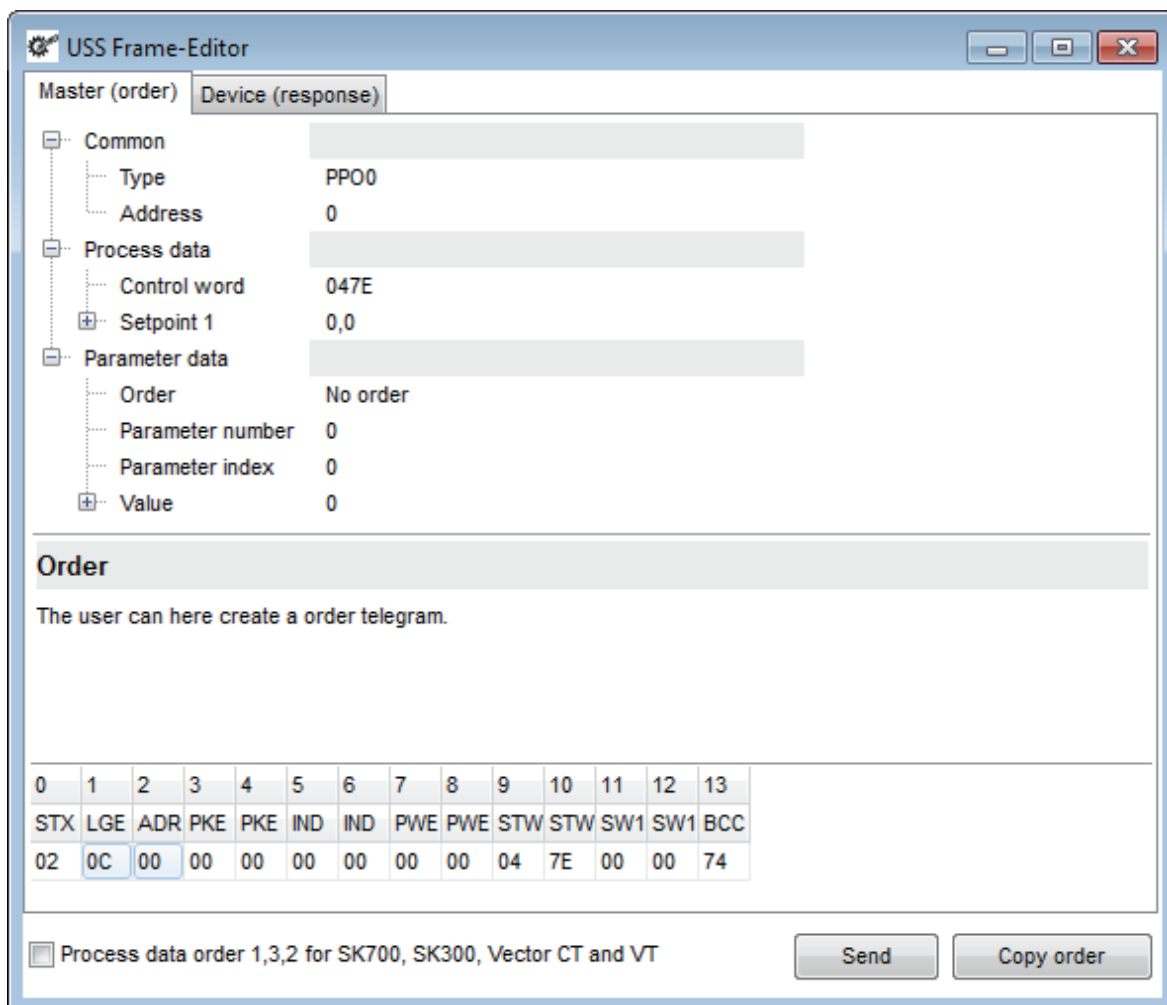
SK700, SK300, Vector CT和VT的过程值序列1, 3, 2

使用该选项，2和3过程值的序列可以为旧的变频器更改。该选项仅影响PPO2和PPO4电报类型。过程值的序列显示在表格视图中。

5.3.6 “控制字”

9.1 主机（指令）

该视图分为几个部分。在上部，顺序电报以树形结构显示。电报的每个分量按主题在树形结构中列出。用户可以更改所有带白色背景的条目。为此，必须用鼠标或键盘将条目突出显示。再次单击该条目将打开输入编辑器。根据条目，输入编辑器可能会有所不同。对于数值，也可以通过按数字键打开输入编辑器。采用新值的输入，并通过按“回车”键或通过突出显示新条目关闭输入编辑器。如果无法采用值，则仍使用旧值。如果输入编辑器是选择列表，则可以通过选择条目采用新值，并关闭输入编辑器。如果不采用更改，则用户必须通过按“Esc”键从输入编辑器退出。在树形结构下方显示每个突出显示条目的说明。在下部，顺序电报在表中按字节显示。突出显示的单元格对应于在树形结构中突出显示的条目。



复制查询

该操作将顺序电报转换为十六进制字节字符串，并将字符串复制到Windows剪贴板。

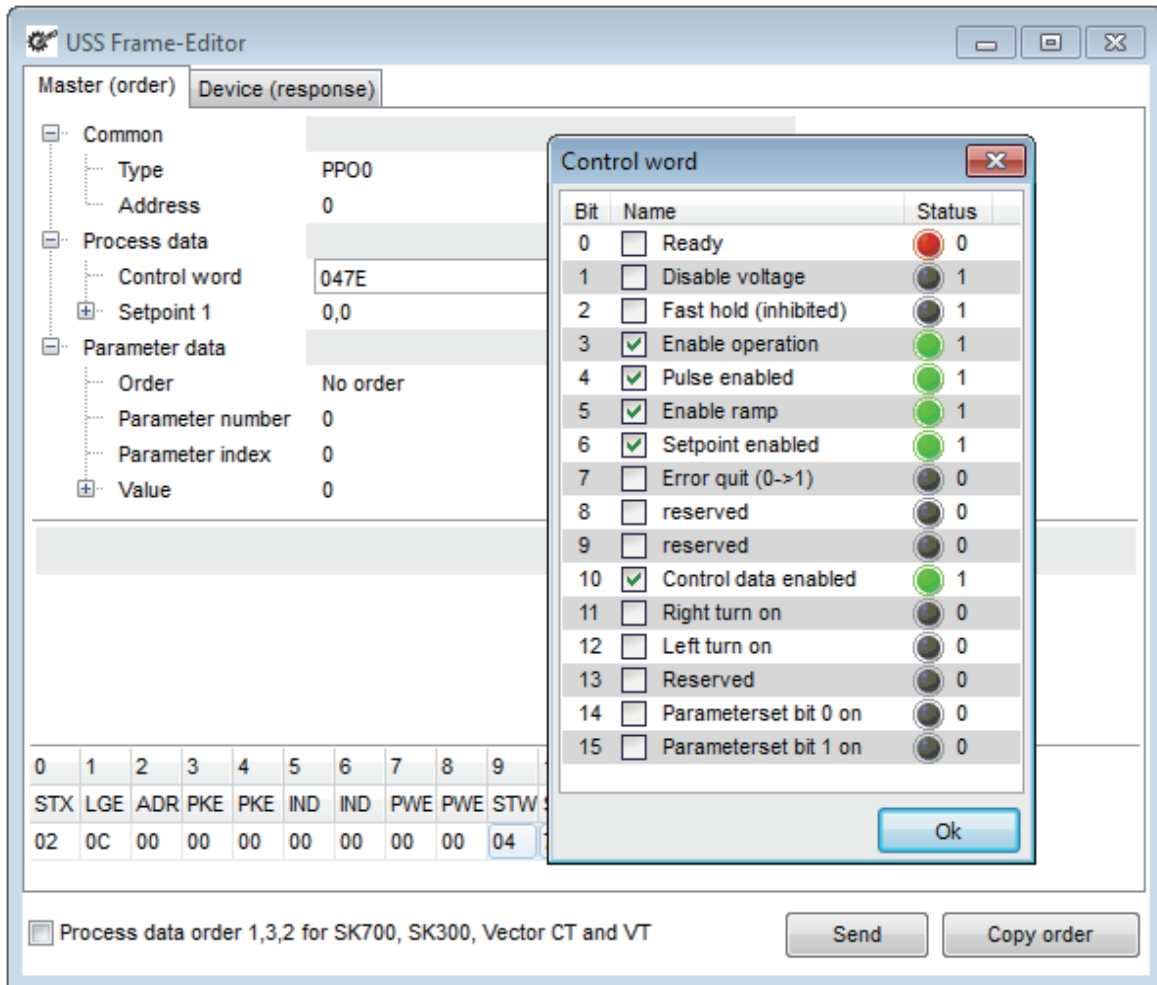
9.2 设备（响应）

该视图分为几个部分。在下部，响应电报在表中按字节显示。用户可以在该表中更改响应电报。

除了STX, LGE和BCC之外的所有字节都可以更改。用户在表中选择一个单元格，并输入一个新值。如果要更改电报的长度和结构，则必须打开表格的上下文菜单。之后，在菜单中选择一个新的电报类型。

每个更改之后，更新树形结构。树形结构仅用于可视化USS电报的分量，无法编辑。一个例外是实际值的格式化和参数值的分辨率。USS电报中不含该信息。格式化必须根据实际值的设置进行更改。分辨率也必须根据参数进行选择。请参阅特定变频器的说明以获取该值。

状态字在树形结构中显示为十六进制值。执行另一个视图以可视化各个位。必须突出显示状态字才能打开视图。再次单击该条目将以写保护模式打开输入编辑器。用户可以使用“...”按钮打开视图。



输入响应

该操作打开响应电报的输入对话框。用户可以将电报作为十六进制编码的字节字符串输入。

10 PLC

10.1 一般信息

诺德变频器SK 180E/SK 190E, SK 2xxE, SK 2xxE-FDS和SK 5xxE系列以及电机启动器SK 155E-FDS/SK 175E-FDS系列包含与当前IEC61131-3标准相似的逻辑处理, 用于存储器可编程控制装置 (SPS/PLC)。该PLC的反应速度或计算能力适合在变频器区域执行较小的任务。可以监测、评估来自连接现场总线的变频器输入或信息, 并进一步将其处理为变频器的适当设定值。与其它诺德设备结合使用, 也可以进行系统状态的可视化或特殊客户参数的输入。因此, 在有限的范围内, 通过消除之前的外部PC解决方案可以节省成本。支持作为编程语言的AWL。AWL是面向机器的, 基于文本的编程语言, 其范围和应用在IEC61131-3中规定。

信息

仅能通过诺德软件NORD CON才能进行编程和下载至设备。

10.1.1 PLC规范

功能	说明		
标准	符合IEC61131-3		
语言	指令列表 (IL) , 结构化文本 (ST)		
任务	循环任务, 每5 ms调用一次程序		
计算机性能	每1 ms大约200条AWL命令		
程序存储器	SK 5xxE, SK 2xxE, SK 2x0E-FDS	SK 190E / SK 180E	SK 155E FDS / SK 175E-FDS
	8128字节, 用于标志、功能和PLC程序	2032字节, 用于标志、功能和PLC程序	2028字节, 用于标志、功能和PLC程序
命令的最多可能数量	大约2580条命令	大约660条命令	大约660条命令
	说明: 这是平均值。大量使用标志、过程数据和功能会大大减少可能的行数; 请参阅“资源”章节。		
可自由访问的CAN邮箱	20		
支持的设备	SK 54xE SK 53xE / SK 52xE ab V3.0 SK 2xxE ab V2.0 SK 2x0E-FDS SK 180E / SK 190E SK 155E-FDS / SK 175E-FDS		

10.1.2 PLC结构

10.1.2.1 存储器

PLC存储器分为程序存储器和标志存储器。除了变量，功能块的实例也保存在标志存储器区域中。FB实例是个存储器区域，其中保存着功能命令的所有内部输入和输出变量。每个功能命令声明都需要一个单独的实例。程序存储器和标志存储器之间的边界是动态确定的，取决于标志区域的大小。



在标志存储器中，变量部分存储了两类不同的变量：

[VAR]

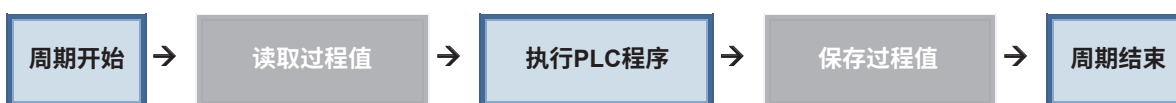
存储器变量用于保存辅助信息和状态。该类型的变量在每次PLC启动时初始化。在PLC循环序列期间保留存储器内容。

[VAR_ACCESS]

用于读取和描述变频器的过程数据（输入、输出、设定值等）。这些值在每个PLC周期重新生成。

10.1.2.2 过程映像

设备可以使用几个物理尺寸，例如扭矩、速度、位置、输入、输出等。这些尺寸分为实际值和设定值。它们可以加载到PLC的过程映像中，并受其影响。必须在VAR_ACCESS类中的变量列表中定义所需的过程。在每个PLC周期中，新读入所有在变量列表中定义的变频器过程数据。在每个PLC周期结束时，可写过程数据被传回到变频器。请见下图。



由于该序列，对循环程序进行编程很重要。为等待某个事件（例如输入电平的改变）而进行的编程环路不会产生所需的结果。如果是访问过程值的功能块，则该行为是不同的。这里，在调用功能块时读取过程值，在块终止时立即写入过程值。

i 信息

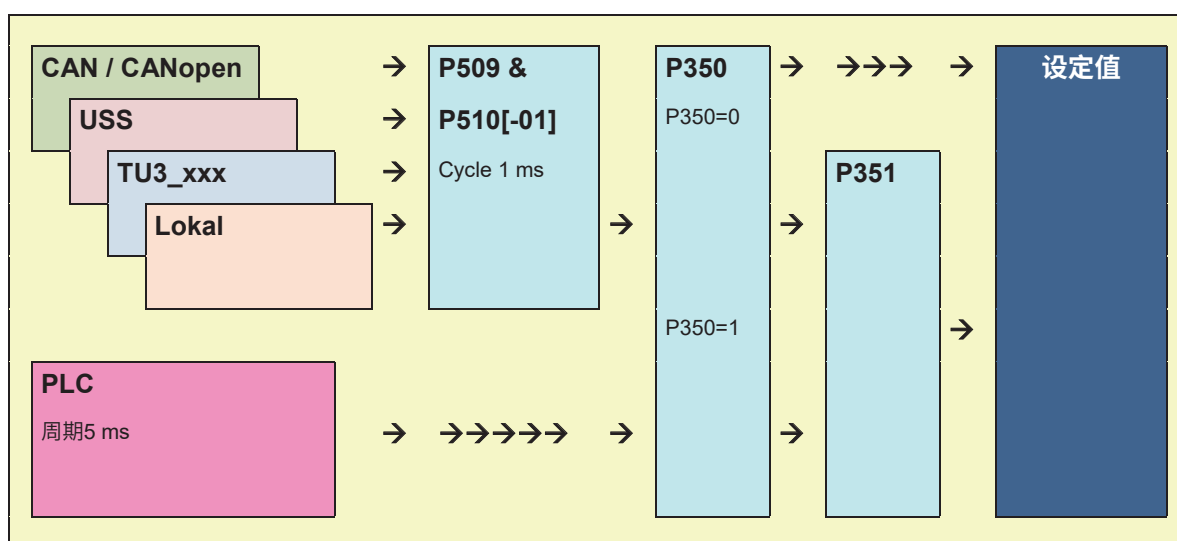
如果使用移动块MC_Power、MC_Reset、MC_MoveVelocity、MC_Move、MC_Home或MC_Stop，则不能使用“PLC_Control_Word”和“PLC_Set_Val1”至“PLC_Set_Val5”。否则，在变量列表中的值始终重写对功能块的更改。

10.1.2.3 程序任务

在PLC中执行程序作为一个单独的任务执行。该任务每5 ms周期性调用一次，最长持续时间为3 ms。如果此时无法执行较长的程序，则该程序将被中断，并在接下来的5 ms任务中继续。

10.1.2.4 处理设定值

变频器有多种设定值源，这些设定值源最终通过几个参数连接，形成变频器设定值。



如果激活PLC（P350=1），则通过P509和通过P351的P510[-01]从外部源（主设定值）进行预选设定值，最终决定是使用PLC的设定值还是使用通过P509/510[-01]输入的值。也可能两者结合使用。辅助设定值（P510[-02]）的更改与PLC功能无关。所有辅助设定值源和PLC以同等优先级将其辅助设定值传输到变频器。

10.1.2.5 通过累加器进行数据处理

累加器构成PLC的中央计算单元。几乎所有的AWL命令都只能与累加器结合使用。PLC有三个累加器。为23位累加器1和累加器2以及BOOL格式的AE。AE用于所有布尔加载、保存和比较操作。如果加载布尔值，则在AE比较操作中进行描述，将结果传输到AE，并由AE触发条件跳转。累加器1和累加器2用于数据格式BYTE、INT和DINT的所有操作数。累加器1是主累加器，累加器2仅用于辅助功能。所有加载和存储操作数均由累加器1处理。所有算术操作数将其结果保存在累加器1中。使用每个加载命令，将累加器1的内容移到累加器2中。后续的运算符将两个累加器连接在一起或对它们进行评估，并将结果保存在累加器1中，在下文中，该累加器通常称为“累加器”。

10.1.3 功能范围

PLC支持范围广泛的运算符、功能和标准功能模块，这些模块在IEC61131-3中定义。在以下章节中有详细说明。此外，还说明了所支持的功能块。

10.1.3.1 运动控制库

运动控制库基于PLCopen规范“运动控制功能块”。主要包含用于移动驱动器的功能块。此外，还提供了设备参数的读取和写入功能块。

10.1.3.2 带飞剪的电子齿轮

变频器配备有电子齿轮（定位模式下的同步操作）和飞剪功能。通过这些功能，变频器可以跟随另一个具有角度同步的驱动单元。除此之外，使用附加功能飞剪，可以与移动驱动单元的精确位置同步。操作模式电子齿轮可以随时启动和停止。这使得传统的位置控制与其移动命令和减速机功能相结合。对于齿轮功能，主轴上需要带内部CAN总线的NORDAC矢量。

10.1.3.3 可视化

借助控制盒或参数盒，可以可视化操作状态和参数化变频器。或者，可以使用PLC CAN总线面板的CANopen主机功能来显示信息。

控制盒

可视化的最简单版本是控制盒。可以通过两个过程值访问4位数字显示和键盘状态。这使简单的HMI应用程序可以非常快速地执行。P001必须设置为“PLC-控制盒值”，以便PLC可以访问显示。另一个特殊特性是不再可以通过箭头键访问参数菜单。而是必须同时按下“打开”和“回车”键。

参数盒

在可视化模式下，可以通过PLC设置参数盒显示中80个字符的每一个（4行，每行20个字符）。可以同时传输数字和文本。另外，PLC可以处理参数盒上的键盘条目。这样可以执行更复杂的HMI功能（显示实际值，更改窗口，传输或设定值等）。可通过PLC中的功能块访问参数盒显示。可视化是通过参数盒的操作值显示来实现的。通过参数盒参数P1003设置操作值显示的内容。该参数可以在“显示”主菜单项下找到。P1003必须设置为值“PLC显示”。此后，可以通过左右箭头键再次选择操作值显示。然后显示由PLC控制的显示。即使再次打开之后，该设置仍然有效。

10.1.3.4 过程控制器

过程控制器是具有有限输出大小的PID-T1控制器。借助于PLC中的该功能模块，可以简单地设置复杂的控制功能，通过该功能，各种过程（例如压力调节）可以以比常用的两点控制器更为简洁的方式执行。

10.1.3.5 CANopen通信

除了标准通信通道之外，PLC还提供了通信的更多可能性。通过变频器的CAN总线接口，可以与其它设备建立附加的通信。用于此处的协议为CANopen。通信仅限于PDO数据传输和NMT命令。通过SDO，PDO1，PDO2和广播的标准CANopen变频器通信不受该PLC功能的影响。


PDO（过程数据对象）

可以通过PDO控制和监测其它变频器。但是，也可以将其它制造商的设备连接到PLC。这些可能是IO模块，CANopen编码器，面板等。使用该功能，频率编码器的输入/输出数量可以根据需要进行扩展；然后可以进行模拟输出。

NMT（网络管理对象）

所有CANopen设备必须由总线主机设置为CANopen主线状态“运行”。PDO通信只能在此总线状态下执行。如果CANopen总线中没有总线主机，则必须由PLC执行。为此，可使用功能模块FB_NMT。

10.2 创建PLC程序

PLC程序的创建只能通过PC程序NORD CON执行。PLC编辑器可通过“文件/新建/PLC程序”菜单项或通过符号打开。 仅在具有PLC功能的设备构成设备概述的中心时，才能激活该按钮。

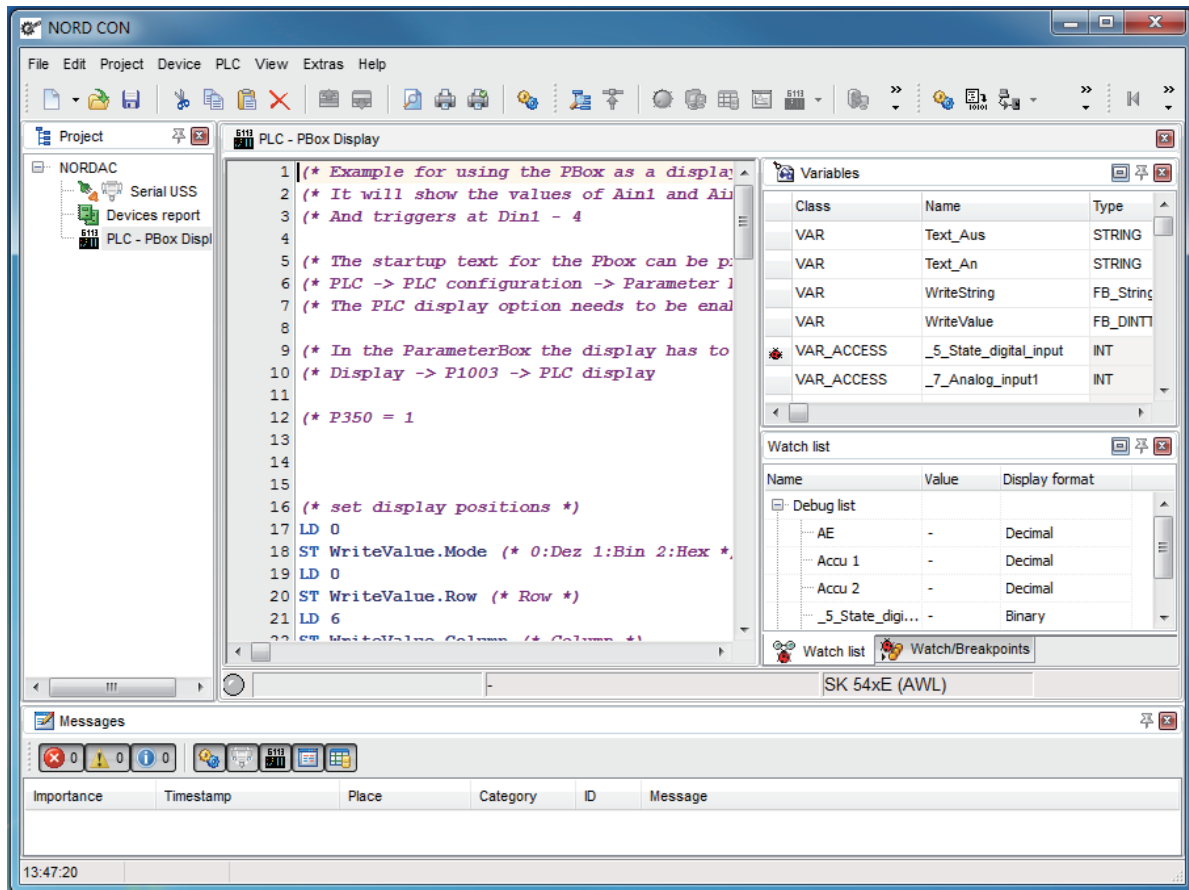
10.2.1 加载，保存和打印

通过主菜单或符号栏中的相应条目执行加载、保存和打印功能。当打开时，建议在“打开”对话框中将文件类型设置为“PLC程序”（*.awl）。这样，仅显示可由PLC编辑器读取的文件。如果要保存已创建的PLC程序，则必须激活PLC编辑器窗口。通过执行“保存”或“另存为”来保存PLC程序。通过“另存为”操作，也可以从文件类型的条目（程序PLC（*.awl））中检测到。要打印PLC程序，必须激活相应的PLC窗口。然后通过“文件/打印”或相应符号启动打印输出。

PLC程序也可作为备份PLC程序进行保存。为此，用户必须在文件选择对话框中将文件类型设置为“备份AWL文件”或“备份ST文件”。然后，PLC程序以加密（*.awls或*.nsts）和普通（*.awl，*.nstx）版本保存。加密的PLC程序仅能传输到设备（请参见2.2.4““设备”目录”）。

10.2.2 编辑器

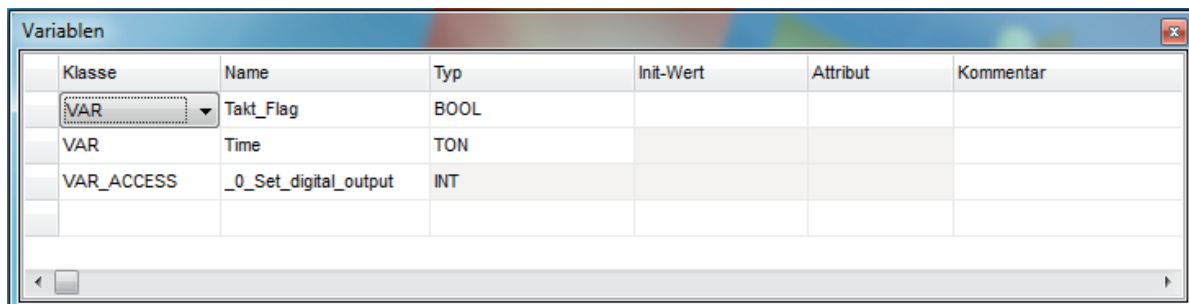
PLC编辑器分为四个不同的窗口。



以下章节将更详细地描述各个窗口。

10.2.2.1 变量和FB声明

程序所需的所有变量、过程值和功能块都在此窗口中声明。



变量

变量通过设置类别“VAR”创建。变量的名称可以自由选择。在类型字段中，可以在BOOL，BYTE，INT和DINT之间进行选择。可以在“初始值”下输入开始初始化。

过程值

通过在选择类别下的“VAR_ACCESS”条目来创建。名称不能自由选择，并且此类型禁止使用“初始值”字段。

功能模块

在类别下选择条目“VAR”。功能模块（FB）相关实例的名称可以自由选择。在类型下选择所需的FB。不能为功能模块设置初始值。

所有与变量窗口相关的菜单项可通过上下文菜单调用。这样可以添加和删除条目。可激活变量和用于监测（监视功能）或调试（断点）的过程变量。

10.2.2.2 输入窗口

输入窗口用于输入程序和显示AWL程序。由以下功能提供：

- 突出显示句法
- 书签
- 变量声明
- 调试

句法突出显示

如果编辑器识别了命令和分配变量，则命令以蓝色显示，变量以黑色显示。只要不是这种情况，则以细黑色斜体显示。

书签

由于编辑器中的程序可能相当长，因此可以使用功能书签标记程序中的要点，并直接跳转到这些要点。光标必须位于相关行中才能进行标记。通过“切换书签”菜单项（鼠标右键菜单），将该行标记为所需的书签。书签可通过“转到书签”菜单项进行访问。

声明变量

通过编辑器菜单“添加变量”（鼠标右键）可以使用编辑器声明新变量。

调试

对于调试功能，断点位置和监视点在编辑器中指定。这可以通过“切换监视点”（断点）和“切换监测点”（监视点）菜单项执行。也可以通过单击编辑器窗口的左边框来指定断点的位置。必须标记在调试期间从变频器读取的变量和过程值。这可以通过“调试变量”和“监视变量”菜单项在编辑器中执行。为此，在选择所需菜单项之前必须标记相关变量。

10.2.2.3 监视和断点显示窗口

该窗口有两个选项卡，如下所述。

待检点

该窗口显示了所有已设置的断点和监视点。这些可以通过复选框打开和关闭，并用“删除键”删除。用鼠标右键可调用相应菜单。

观测列表

这显示了所有已选择用于观测的变量。当前内容显示在“值”列中。可以使用“显示”列选择显示格式。

10.2.2.4 PLC消息窗口


所有PLC状态和故障信息在该窗口中显示。如果程序翻译正确，则显示“翻译无误”消息。资源的使用显示在该行的下一行。如果PLC程序有错误，则显示“错误X”消息。错误数量以X显示。以下行以格式显示特定的故障信息：

[行数]: 错误描述


10.2.3 将PLC程序传输到设备

有几种方式将PLC程序传输到设备。


直接传输PLC程序：

1. 在项目树中选择设备
2. 打开弹出菜单（按鼠标右键）
3. 执行功能“将PLC程序传输到设备” 
4. 在文件选择对话框中选择文件，并按“打开”

使用PLC编辑器传输PLC程序（离线）：

1. 打开PLC程序（文件->打开）
2. 将PLC编辑器与设备连接（PLC->连接）
3. 翻译PLC程序
4. 将PLC程序传输到设备 

使用PLC编辑器传输PLC程序（在线）：

1. 在项目树中选择设备
2. 打开PLC编辑器 
3. 打开PLC程序

4. 将文件导入在线视图
5. 翻译PLC程序
6. 将PLC程序传输到设备

信息

SK 1xxE-FDS – 写入周期的有限数量

在设备SK 155E-FDS/SK 175E-FDS中，闪存用作存储介质。闪存的写入周期数量非常有限。默认情况下，程序加载到RAM中。然后可以进行启动和测试。如果随后重新启动PLC，则必须将程序重新加载到设备中以初始化PLC变量。如果要永久存储在设备中，则必须执行功能“将程序传输并存储到设备”。

10.2.4 调试

由于程序很少在第一次运行时起作用，因此PLC提供了多种查找故障的可能性。这些可能性可大致分为两类，具体描述如下。



10.2.4.1 观测点（监视点）

最简单的调试版本是监视点功能。这提供了几个变量行为的快速概述。为此，在程序中的任意点设置观测点。当PLC处理该行时，在观测列表（“观测列表”窗口）中保存和显示多达5个值。使用上下文菜单在条目窗口或变量窗口选择要观测的5个值。



信息



当前版本中，功能的变量无法添加到监视列表中！

10.2.4.2 待检点（断点）


通过待检点，可以在程序的特定行故意停止PLC程序。如果PLC进入断点，则将读取AE累加器1和累加器2，以及所有已通过“调试变量”菜单项选择的变量。在PLC  程序中可设置多达5个断点。该功能可以通过  符号启动。程序现在运行，直到触发一个待检点。进一步激活符号栏可以使程序继续运行，直至到达下一个待检点。如果程序要继续运行，则启动符号。

10.2.4.3 单步

通过该调试方法，可以逐行执行PLC程序。在每个单独的步骤中，所有选择的变量从设备的PLC中读取，并在“观测列表”窗口显示。可通过鼠标右键菜单在输入窗口或变量窗口选择要观测的值。单步调试的条件是在开始调试之前至少已设置一个断点。通过  启动符号打开调试模式。只有当程序运行到第一个断点时，才能通过  符号调试后续行。一些命令行包含几个

单独的命令。因此，在条目窗口中步骤指示器向前跳转之前，可以处理两个或多个单独的步骤。实际位置由PLC编辑器左侧窗口中的小箭头显示。当  启动符号时，程序继续运行，直到下一个待检点。如果程序要继续运行，则  启动符号。

10.2.5 PLC配置

PLC配置对话框可通过  符号打开。这里，可以对PLC进行基本设置，具体描述如下。

周期时间监测

该功能监测PLC周期的最大处理时间。使用该功能，可以捕捉到PLC程序中意外的连续程序环路。如果超过该时间，则变频器将触发错误E22.4。

允许参数盒功能模块

如果在PLC程序中通过参数盒执行可视化，则必须启用该选项。否则，当启动变频器时，相应的功能块会产生编译器错误。

无效控制数据

PLC可以评估从可能的总线系统接收的控制字。但是，只有设置了位“PZD有效”（位10），控制字才能通过评估。如果PLC要评估不符合USS协议的控制字，则必须激活该选项。这样就不再查询第一个字的位10。

不得在待检点暂停系统时间

如果PLC处于待检点或单步模式，则调试期间将暂停系统时间。系统时间构成PLC中所有计时器的基础。如果要在调试期间继续运行系统时间，则必须激活该功能。

10.3 功能块

功能块是小程序，可以将其状态值保存在内部变量中。因此，必须在NORD CON变量列表中为每个功能块创建一个单独的实例。例如，如果计时器要并行监测3次，则必须在变量列表中设置3次。

信息

检测信号边缘

为了使以下功能块在输入处检测到边缘，必须在输入处以不同状态执行两次功能调用。

10.3.1 CANopen

PLC可以通过功能块在PDO通道进行配置、监测和传输。PDO可以通过PDO传输或接收多达8字节过程数据。可以通过一个单独地址（COB-ID）访问每个PDO。在PLC中可配置多达20个PDO。对于较简单的操作，COB-ID不是直接输入。而是将设备地址和PDO号通信

到FB。生成的COB-ID是基于预定义的连接装置（CiADS301）确定的。这将为PLC产生以下可能的COB-ID。

传输PDO		接收PDO	
PDO	COB-ID	PDO	COB-ID
PDO1	200h + 设备地址	PDO1	180h + 设备地址
PDO2	300h + 设备地址	PDO2	280h + 设备地址
PDO3	400h + 设备地址	PDO3	380h + 设备地址
PDO4	500h + 设备地址	PDO4	480h + 设备地址

诺德变频器使用PDO1与过程数据通信。PDO2仅用于设定值/实际值4和5。

10.3.1.1 概述

功能模块	描述
FB_PDConfig	PDO配置
FB_PDOSend	传输PDO
FB_PDOReceive	接收PDO
FB_NMT	启用和禁用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
可用性	X	X	X	X	X	X

加电后，所有的CAN参与者都处于总线状态“运行前”。在这种状态下，既不能传输也不能接收PDO。为了使PLC能够与CAN总线上的其它参与者通信，必须将它们设置为运行状态。这通常由总线主机执行。如果没有总线主机，则由FB_NMT执行该任务。可以通过输入**PRE**，**OPE**或**STOP**控制连接到总线的所有参与者的状态。通过**EXECUTE**上正侧面采用输入。必须调用该功能，直到将输出**DONE**或**ERROR**已设置为1。

如果**ERROR**设置为1，则变频器的RJ45 CAN插口没有24V电源，或者变频器的CAN驱动器处于总线断开状态。通过**EXECUTE**上负侧面，将所有输出复位为0。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	执行	BOOL	DONE	NMT命令已传输	BOOL

PRE	将所有参与者设置为“运行前”状态	BOOL	ERROR	FB错误	BOOL
OPE	将所有参与者设置为“运行”状态	BOOL			
STOP	将所有参与者设置为“停止”状态	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
可用性	X	X	X	X		

使用该FB配置PDO。通过该功能的一个实例，可以配置所有所需的PDO。每个PDO只能调用一次FB。可设置多达20个PDO。每个PDO有其自己的参数化。通过消息框编号对其它CANopen FB的PDO进行分配。**TARGETID**表示设备的地址。对于诺德变频器，可在P55中或通过DIP开关进行设置。在PDO下输入所需的消息框编号（请参见介绍）。**LENGTH**指定PDO的传输长度。**DIR**指定传输/接收方向。通过**EXECUTE**输入上的正侧面采用数据。调用FB之后，可以立即查询**DONE**输出。如果**DONE**设置为1，则已配置PDO通道。如果**ERROR=1**，则存在一个问题，问题的确切原因保存在**ERRORID**中。通过**EXECUTE**上负侧面，将所有输出复位为0。

传输PDO		接收PDO	
PDO	COB-ID	PDO	COB-ID
PDO1	200h + 设备地址	PDO1	180h + 设备地址
PDO2	300h + 设备地址	PDO2	280h + 设备地址
PDO3	400h + 设备地址	PDO3	380h + 设备地址
PDO4	500h + 设备地址	PDO4	480h + 设备地址
PDO5	180h + 设备地址	PDO5	200h + 设备地址
PDO6	280h + 设备地址	PDO6	300h + 设备地址
PDO7	380h + 设备地址	PDO7	400h + 设备地址
PDO8	480h + 设备地址	PDO8	500h + 设备地址

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	执行	BOOL	DONE	已配置PDO	BOOL
NUMBER	消息框数值范围 = 0至19	字节	ERROR	FB错误	BOOL
TARGETID	设备地址值范围 = 1至127	字节	ERRORID	错误代码	INT
PDO	PDO值范围 = 1至4	字节			
LENGTH	PDO长度值范围 = 1至8	字节			
DIR	传输或接收 传输 = 1 / 接收 = 0	BOOL			
ERRORID	描述				
0	无错误				
1800h	超过数值范围				
1801h	超过TARGETID值范围				
1802h	超过PDO值范围				
1803h	超过LENGTH值范围				

信息

无法重复使用CAN ID

正由设备使用的CAN-ID不能参数化!

相关接收地址:

- CAN ID = $0x180 + P515[-01]$ PDO1
- CAN ID = $0x180 + P515[-01]+1$ 用于绝对值编码器的CAN ID
- CAN ID = $0x280 + P515[-01]$ PDO2

相关传输地址:

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

ST中的示例：

```
(* Configure PDO *)
PDOConfig(
  Execute := TRUE,
  (* Configure Message box 1 *)
  Number := 1,
  (* Set CAN node number *)
  TargetID := 50,
  (* Select (Standard for PDO1 control word, setpoint1, setpoint2, setpoint3) *)
  PDO := 1,
  (* Specify length of data (Standard for PDO1 is 8 *)
  LENGTH := 8,
  (* Transmit *)
  Dir := 1);
```

```
(* Configure PDO *)
PDOConfig(
  Execute := TRUE,
  (* Configure message box 1 *)
  Number := 2,
  (* Set CAN node number *)
  TargetID := 50,
  (* Select PDO (Standard for PDO2 setpoint4, setpoint5 SK540E) *)
  PDO := 2,
  (* Specify length of data (Standard for PDO2 is 4 *)
  LENGTH := 4,
  (* Transmit *)
  Dir := 1);
```

```
(* Configure PDO *)
PDOConfig(
  Execute := TRUE,
  (* Configure message box 2 *)
  Number := 2,
  (* Set CAN node number *)
  TargetID := 50,
  (* Select PDO (Standard for PDO1 status word, actual value1, actual value2, actual
  value3) *)
  PDO := 1,
  (* Specify length of data (Standard for PDO1 is 8 *)
  LENGTH := 8,
  (* Receive *)
  Dir := 0);
```

10.3.1.4 FB_PDOReceive

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

该FB监测先前配置的用于传入消息的PDO通道。如果“ENABLE”输入设置为1，则开始监测。调用功能后，必须检查NEW输出。如果更改为1，则会收到一条新消息。使用下一个调用功能删除NEW输出。已接收的数据在WORD1至WORD4中。可通过TIME监测PDO通道的周期性接收。如果在TIME中输入1和32767 ms之间的值，则在该期间必须接收到一条消息。否则，FB进入错误状态（ERROR=1）。可以使用值0禁用该功能。监测计时器以5 ms步长运行。如果错误ERROR设置为1。在这种情况下，DONE为0。则在ERRORID中，相关错误代码有效。通过ENABLE负侧面，复位DONE、ERROR和ERRORID。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	执行	BOOL	NEW	传输PDO = 1	BOOL
NUMBER	消息框数值范围 = 0至19	字节	ERROR	FB错误	BOOL
TIME	监视功能值范围 = 0至32767 0 = 禁用 1至32767 = 监测时间t	INT	ERRORID	错误代码	INT
			WORD1	接收数据字1	INT
			WORD2	接收数据字2	INT
			WORD3	接收数据字3	INT
			WORD4	接收数据字4	INT
ERRORID	描述				
0	无错误				
1800h	超过数值范围				
1804h	所选框配置不正确				
1805h	总线驱动器无24V电源，或总线驱动器处于“总线关闭”状态				
1807h	接收超时（监视功能）				

信息

PLC周期时间

PLC周期约为5 ms，即在PLC程序调用一次功能时，每5 ms只能读取一条CAN消息。如果快速连续发送几条消息，则消息可能会被覆盖。

ST中的示例：

```

IF bFirstTime THEN
    (* Set the device to the status Pre - Operational *)
    NMT(Execute := TRUE, OPE := TRUE);
    IF not NMT.Done THEN
        RETURN;
    END_IF;

    (* Configure PDO *)
    PDOConfig(
        Execute := TRUE,
        (* Configure Messagebox 2 *)
        Number := 2,
        (* Set CAN node number *)
        TargetID := 50,
        (* Select PDO (Standard for PDO1 status word, actual value1, actual value2, actual
value3) *)
        PDO := 1,
        (* Specify length of data (Standard for PDO1 is 8 *)
        Length := 8,
        (* Receive *)
        Dir := 0) ;
    END_IF;

    (* Read out status and actual values *)
    PDOReceive(Enable := TRUE, Number := 2);
    IF PDOReceive.New THEN
    
```

```

State := PDORceive.Word1;
Sollwert1 := PDORceive.Word2;
Sollwert2 := PDORceive.Word3;
Sollwert3 := PDORceive.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
可用性	X	X	X	X		

通过该FB，可以在先前配置的通道上传输PDO。这可以是一次性传输，也可以是周期性传输。要传输的数据在**WORD1**至**WORD4**中输入。无论变频器的CANOpen状态如何，都可以传输PDO。可通过**NUMBER**选择先前配置的PDO通道。要传输的数据在**WORD1**至**WORD4**中输入。通过**CYCLE**，可选择一次性传输（设置=0）或周期性传输。通过**EXECUTE**上正侧面发送PDO。如果**DONE=1**，则所有条目都正确，并发送PDO。如果**ERROR=1**，则存在问题。确切原因保存在**ERRORID**中。通过**EXECUTE**上负侧面复位所有输出。PLC的时基为5 ms，这也适用于**CYCLE**输入。只能执行5 ms倍数的传输周期。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	执行	BOOL	DONE	传输PDO = 1	BOOL
NUMBER	消息框数值范围 = 0至19	字节	ERROR	FB错误	BOOL
CYCLE	传输周期值范围 = 0至255 0 = 禁用 1至255 = 以ms为单位的传输周期	字节	ERRORID	错误代码	INT
WORD1	传输数据字1	INT			
WORD2	传输数据字2	INT			
WORD3	传输数据字3	INT			
WORD4	传输数据字4	INT			
ERRORID	描述				
0	无错误				
1800h	超过数值范围				
1804h	所选框配置不正确				
1805h	总线驱动器无24V电源，或总线驱动器处于“总线关闭”状态				

如果**DONE**更改为1，则CAN模块接受要传输的消息，但该消息尚未传输。实际传输在背景中并行运行。如果现在通过FB直接按序列发送几个消息，则可能是在新调用中，先前的信息尚未发送。这可以通过以下事实来识别：CAL调用后，**DONE**和**ERROR**信号均未设置为1。可以重复调用CAL，直到两个信号之一更改为1。如果要通过单个FB写入几个不同的CAN ID，则可以通过FB的新配置来执行。但是，不得在与传输相同的PLC周期中执行此操作。否则，存在要传输的消息被FB_PDOConfig删除的危险。

ST中的示例：

```

IF bFirstTime THEN
  (* Set the device to the status Pre-Operational *)
  NMT(Execute := TRUE, OPE := TRUE);
  IF not NMT.Done THEN
    RETURN;
  END_IF;

  (* Configure PDO*)
  PDOConfig(
    Execute := TRUE,
    (* Configure Messagebox 1 *)
    Number := 1,
    (* Set CAN node number *)
    TargetID := 50,
    (* Select PDO (Standard for PDO1 control word, setpoint1, setpoint2, setpoint3) *)
    PDO := 1,
    (* Specify length of data (Standard for PDO1 is 8 *)
    LENGTH := 8,
    (* Transmit *)
    Dir := 1);

  IF not PDOConfig.Done THEN
    RETURN;
  END_IF;

  (* Transmit PDO - Set device to status Ready for Operation *)
  PDOSend(Execute := TRUE, Number := 1, Word1 := 1150, Word2 := 0, Word3 := 0, Word4 := 0);
  IF NOT PDOSend.Done THEN
    RETURN;
  END_IF;

  PDOSend(Execute := FALSE);
  bFirstTime := FALSE;
END_IF;

CASE State OF
  0:
    (* If digital input 1 is set *)
    IF _5_State_digital_input.0 THEN
      (* Transmit PDO - Set device to status Ready for Operation *)
      PDOSend(Execute := TRUE, Number := 1, Word1 := 1150, Word2 := 0, Word3 := 0,
        Word4 := 0);
      State := 10;
      RETURN;
    END_IF;

    (* If digital input 2 is set *)
    IF _5_State_digital_input.1 THEN
      (* Transmit PDO - Enable device and setpoint frequency to 50% of the maximum
        frequency *)
      PDOSend(Execute := TRUE, Number := 1, Word1 := 1151, Word2 := 16#2000, Word3 := 0,
        Word4 := 0);
      State := 10;
      RETURN;
    END_IF;

  10:
    PDOSend;
    IF PDOSend.Done THEN
      PDOSend(Execute := FALSE);
      State := 0;
    END_IF;
END_CASE;

```



```
END_IF;
END_CASE;
```

10.3.2 带飞剪的电子齿轮

对于电子齿轮（“角度同步操作”）和子功能飞剪，有两个功能块可以启用这些功能。此外，必须设置各种参数以正确执行在主机和从机变频器中的两个功能块。下表显示了这一示例（由SK 540E的示例进行说明）。

主机FI			从机FI		
参数	设置	描述	参数	设置	描述
P502[-01]	20	根据频率斜坡的设定值频率	P509	10 *	CANopen广播*
P502[-02]	15	实际位置包括高位字	P510[-01]	10	CANopen广播
P502[-03]	10	实际位置包括低位字（最小	P510[-02]	10	CANopen广播
P503	3	CANopen	P505	0	0,0 Hz
P505	0	0.0 Hz	P515[-02]	P515[-03] _{Master}	广播从机地址
P514	5	250 kBaud（最小 100 kBaud）	P546[-01]	4	频率添加
P515[-03]	P515[-02] 从机	广播主机地址	P546[-02]	24	设定值位置包括高位字
			P546[-03]	23	设定值位置包括低位字
			P600	1.2	位置控制开
			仅适用FB_Gearing		
			P553[-01]	21	位置设定值位置低位字
			P553[-02]	22	位置设定值位置高位字

* (P509) 不一定要设置为{10}“CANopen广播”。但是，在这种情况下，主机 (P502[-01]) 必须设置为{21}“不滑动的实际频率”。

信息

实际位置-传输格式

主机的实际位置必须以“增量” (Inc) 格式进行通信。

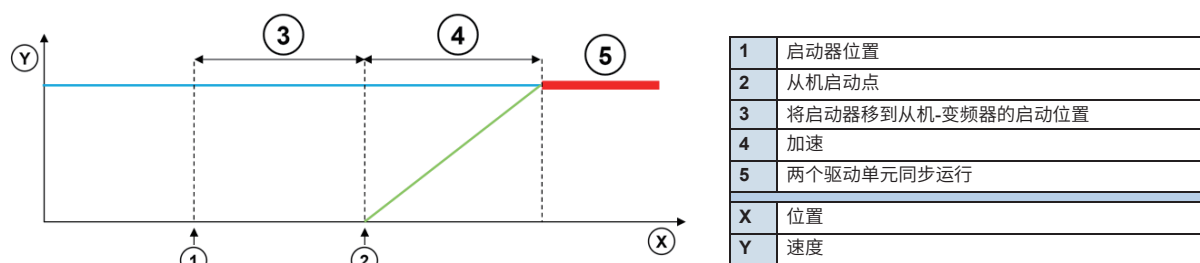
10.3.2.1 概述

功能模块	描述
FB_Gearing	FB用于简单减速机功能
FB_FlyingSaw	FB用于带飞剪的减速机功能

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
可用性	X	X	X	X	X	

飞剪功能是减速机功能的扩展。借助于该功能，可以将运行的驱动单元同步到精确的位置。与FB_Gearing相比，同步是相对的，即从轴同步移动到“飞剪”开始时应用的主机位置。同步过程如下图所示。



如果启动功能，从机变频器加速到主轴速度。加速由**ACCELERATION**指定。在低速时，斜坡更平坦，在高速时，从机变频器有个陡峭的斜坡。如果P553指定为设定值位置，则加速路径以转数（1000=1000rev.）表示。如果设定值位置INC用于P553，则加速路径以增量指定。

如果以保存在**ACCELERATION**中的从动驱动器的位置的设置启动器，则从机与主机驱动器的触发位置精确同步。

FB必须通过**ENABLE**输入打开。该功能可以通过数字输入（P420[-xx]=64，启动飞剪）或通过**EXECUTE**启动。然后变频器加速到主轴的速度。当实现与主轴同步时，**DONE**输出切换到1。

通过**STOP**输入或数字输入功能P420[-xx]=77，停止飞剪，减速机功能关闭，变频器制动到0Hz，并保持停止状态。通过**HOME**输入，将变频器移到绝对位置0。在**HOME**或**STOP**命令结束后，相关的分配输出激活。可以通过重新激活**EXECUTE**或数字输入来重新启动减速机功能。通过数字输入功能（P420[-xx]=63，停止同步），可以停止减速机功能，并将该功能移到绝对位置0。

如果MC_Stop功能中断该功能，则**ABORT**设置为1。如果出现错误，则**ERROR**设置为1，错误代码在**ERRORID**中设置。如果**ENABLE**切换到0，则这三个输出将复位。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	启用	BOOL	VALID	达到指定的设定值频率	BOOL
EXECUTE	开始同步	BOOL	DONEHOME	回复原位完成	
STOP	停止同步	BOOL	DONESTOP	执行停止命令	
HOME	移到位置0	BOOL	ABORT	中止命令	BOOL
ACCELERATION	加速路径 (1转 = 1.000)	DINT	ERROR	FB错误	BOOL
			ERRORID	错误代码	INT
ERRORID	描述				
0	无错误				
1000h	未启用FI				
1200h	未激活位置控制				

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
可用性	X	X	X	X	X	

变频器的位置和速度可以通过功能模块FB_Gearing与主机变频器的位置和速度同步。使用此功能的从机始终跟随主机变频器的移动。同步是绝对的，即从机的位置与主机的位置始终相同。

信息

如果从机在与主机不同的位置切换到减速机模式，则从机以最大频率移动到主机的位置。如果指定了传动比，则再次打开时也会产生新的位置。

执行同步的位置值以及速度必须通过广播通道进行通信。该功能通过输入**ENABLE**启用。为此，必须启用位置控制和输出级。输出级可以通过例如功能MC_Power进行启用。如果**ENABLE**设置为0，则变频器制动到0Hz，并保持停止状态。现在变频器再次在位置控制模式下。如果激活MC_Stop，则变频器从减速机模式退出，并且**ABORT**输出更改为1。如果FB出现错误，则**ERROR**更改为1，并且错误的原因在**ERRORID**中显示。通过将**ENABLE**设置为0，**ERROR**，**ERRORID**和**ABORT**可以复位。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	同步运行激活	BOOL	VALID	减速机功能激活	BOOL
RELATIVE	相对模式 (V2.1及以上)	BOOL	ABORT	中止命令	BOOL
			ERROR	FB错误	BOOL
			ERRORID	错误代码	INT
ERRORID	描述				
0	无错误				
1000h	未启用FI				
1200h	未激活位置控制				
1201h	PLC设定值位置高位未参数化				
1202h	PLC设定值位置低位未参数化				

可以通过参数P607[-05]或P608[-05]设置传动比或旋转方向的更改。更多详情请参见使用手册BU0510 (POSICON位置控制补充使用手册)。

10.3.3 运动控制

运动控制库基于PLCopen规范“运动控制功能块”。包含控制和移动变频器的功能块，并提供对其参数的访问。为使移动块起作用，必须对设备的参数进行若干设置。

功能块	所需设置
MC_MoveVelocity	P350 = PLC激活 P351 = 来自PLC的主设定值 P553 [-xx] = 设定值频率 P600 = 禁用位置控制 (定位模式)
MC_MoveAbsolute	P350 = PLC激活
MC_MoveRelative	P351 = 来自PLC的主设定值
MC_MoveAdditive	P600 = 启用位置控制 (定位模式)
MC_Home	在P553 [-xx] (PLC_Setpoints) 中，必须对设定值位置高位字进行参数化。 在P553 [-xx] (PLC_Setpoints) 中，必须对设定值位置低位字进行参数化。 在P553 [-xx] (PLC_Setpoints) 中，必须对设定值频率进行参数化
MC_Power	P350 = PLC激活
MC_Reset	P351 = 来自PLC的控制字
MC_Stop	

i 信息

PLC_Setpoints 1至5和PLC控制字可通过过程变量进行描述。但是，如果使用运动控制FB，则在变量表中不声明任何相应的过程变量，否则将覆盖运动控制FB的输出。

i 信息

检测信号边缘

为了使以下功能块在输入处检测到边缘，必须在输入处以不同状态执行两次功能调用。

功能块	描述
MC_ReadParameter	读取设备参数的访问
MC_WriteParameter	写入设备参数的访问
MC_MoveVelocity	在速度模式下的移动命令
MC_MoveAbsolute	说明绝对位置的移动命令
MC_MoveRelative	说明相对位置的移动命令
MC_MoveAdditive	有位置附加说明的移动命令
MC_Home	运行回复原位
MC_Power	打开或关闭电机电压
MC_ReadStatus	设备状态
MC_ReadActualPos	读取实际位置
MC_Reset	设备错误复位
MC_Stop	停止所有激活的移动命令

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
可用性	X	X	X	X	X	

该FB用于控制FI，并提供了以比MC_Power更详细的形式创建FI控制字的可能性。通过输入**ENABLE**、**DISABLEVOLTAGE**和**QUICKSTOP**控制FI。请参见下表。

输入模块			变频器行为
ENABLE	QUICKSTOP	DISABLEVOLTAGE	
高	低	低	变频器打开。
低	低	低	变频器制动到0Hz (P103)，然后将电机从电压电源断开。

X	X	高	变频器立即与电压电源断开，并且电机在不制动的情况下运行至停止状态。
X	高	低	变频器紧急停机（P426），然后断开电机的电压。

可以通过输入**PARASET**设置激活参数集。如果输出**STATUS=1**，则FI打开，电流供应至电机。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	启用	BOOL	STATUS	电机通电	BOOL
DISABLEVOLTAGE	关闭电压	BOOL	ERROR	FB错误	BOOL
QUICKSTOP	快速停机	BOOL	ERRORID	错误代码	INT
PARASET	激活参数集值范围： 0 – 3	字节			
ERRORID	描述				
0	无错误				
1001h	停止功能激活				
1300h	FI处于意外状态				

ST示例:

```
(* Device enabled with Dig3 *)
Control.Enable := _5_State_digital_input.2;
(* Parameter sets are specified via Dig1 and Dig2. *)
Control.ParaSet := INT_TO_BYTE(_5_State_digital_input and 2#11);
Control;
(* Is the device enabled? *)
if Control.Status then
    (* Should a different position be moved to? *)
    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;

(* Move to position if device is enabled. *)
Move(Execute := Control.Status);
```

10.3.3.2 MC_Control_MS

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

		SK 52xE			SK 190E	SK 175E-FDS
可用性						X

该FB用于控制启动器（MS）。

输入模块				变频器行为
ENABLE_RIG HT	ENABLE_LEF T	QUICKSTOP	DISABLEVOLTAGE	
高	低	低	低	MS顺时针旋转打开
低	高	低	低	MS逆时针旋转打开
高	高	低	低	MS关闭
低	低	低	低	MS制动至0Hz（P103），然后将电机从电压电源断开
X	X	X	高	MS立即与电压电源断开，并且电机在不制动的情况下运行至停止状态。
X	X	高	低	MS紧急停机（P426），然后断开电机的电压。

(X = 输入的电平无关)

如果输出**STATUS**=1，则MS打开，电流供应至电机。

如果**OPENBRAKE**设置为1，则制动器释放。

VAR_INPUT			VAR_OUTPUT		
输入	解释	型号	输出	解释	型号
ENABLE_RIGHT	右转启用	BOOL	STATUS	电机通电	BOOL
ENABLE_LEFT	左转启用	BOOL	ERROR	FB错误	BOOL
DISABLEVOLTAGE	关闭电压	BOOL	ERRORID	错误代码	INT
QUICKSTOP	快速停机	BOOL			
OPENBRAKE	释放制动器	BOOL			
ERRORID	解释				
0	无错误				
1001h	停止功能激活				
1300h	MS处于意外状态				

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
可用性	X	X	X	X	X	

如果EXECUTE从0更改为1（侧面），则使变频器启动参考运行。变频器以在VELOCITY中输入的设定值频率移动。如果带有位置参考信号（P420[-xx]=参考点）的输入变为激活状态，则旋转方向反转。通过位置参考信号的负侧面，采用POSITION值。然后变频器制动至0Hz，DONE信号更改为1。在整个HOME运行期间，BUSY输出激活。如果输入为“MODE”，设置为“True”，则变频器在回零点后在启动器中间运行。

如果要中止过程（例如由不同的MC功能模块），则设置COMMANDABORTED。如果出现错误，ERROR设置为1。这这种情况下，DONE为0。然后应用在ERRORID中相应的错误代码。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	启用	BOOL	DONE	达到指定的设定值位置	BOOL
POSITION	设定值位置	DINT	COMMAND-ABORTED	中止命令	BOOL
VELOCITY	设定值频率	INT	ERROR	FB错误	BOOL
MODE	回复原位模式（从V2.1起）	BOOL	ERRORID	错误代码	INT
			BUSY	回复原位运行激活	BOOL

ERRORID	描述
0	无错误
1000h	未启用FI
1200h	未激活位置控制
1201h	高位位置尚未输入PLC设定值 (P553)
1202h	低位位置尚未输入PLC设定值 (P553)
1D00h	不支持绝对值编码器

ST示例:

(* One digital input must be set as the reference point (23).
The setpoint frequency and the setpoint position must be set in the PLC setpoint (553.x). POSICON must be enabled (P600) *)

```
(* Enable device with DIG1 *)
Power.Enable := _5_State_digital_input.0;
(* Position after the reference run *)
Home.Position := 5000;
(* Speed during the reference run 50% of the max. frequency (P105) *)
Home.Velocity := 16#2000;
(* Perform reference run when the device is switched on *)
Home.Execute := Power.Status;
```

```
Home;
Power;
```

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
可用性	X	X	X	X	X	

如果EXECUTE从0更改为1（侧面），则将位置和速度设定值写入变频器。设定值频率VELOCITY根据MC_MoveVelocity中说明的缩放比例进行传输。

POSITION:

MODE = False:

设定值位置由传输到POSITION的值产生。

MODE = True:

传输到POSITION的值对应于参数P613中增加1的索引。在该参数索引中保存的位置对应于设定值位置。

示例:

模式 = True; 位置 = 12

FB移动到在实时参数集P613[-13]的位置。

如果变频器已到达设定值位置，则DONE设置为1。DONE由复位EXECUTE删除。如果在到达目标位置之前删除EXECUTE，则DONE设置为1对应一个周期。在移动到设定值位置期间，BUSY激活。如果要

中止过程（例如由另一个MC功能模块），则设置**COMMANDABORTED**。如果出现错误，则**ERROR**设置为1，相应错误代码在**ERRORID**中设置。在这种情况下，**DONE**为0。通过**EXECUTE**上负侧面，将所有输出复位为0。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	启用	BOOL	DONE	达到指定的设定值位置	BOOL
POSITION	设定值位置	DINT	BUSY	未达到设定值位置	BOOL
VELOCITY	设定值频率	INT	COMMAND-ABORTED	中止命令	BOOL
MODE	模式源为设定值位置	BOOL	ERROR	FB错误	BOOL
			ERRORID	错误代码	INT
ERRORID	描述				
0	无错误				
0x1000	未启用FI				
0x1200	未激活位置控制				
0x1201	高位位置尚未输入PLC设定值（P553）				
0x1202	低位位置尚未输入PLC设定值（P553）				

ST示例:

```
(* The device is enabled if DIG1 = TRUE *)
Power(Enable := _5_State_digital_input.0);
IF Power.Status THEN
  (* The device is enabled and moves to position 20000 with 50% max. frequency.
  For this action the motor requires an encoder and position control must be enabled. *)
  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
可用性	X	X	X	X	X	

除了输入**DISTANCE**之外，这在所有点上都与MC_MoveAbsolute相对应。设定值位置由实际设定值位置和传输的**DISTANCE**之和产生。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	启用	BOOL	DONE	达到指定的设定值位置	BOOL
DISTANCE	设定值位置	DINT	COMMAND-	中止命令	BOOL

VELOCITY	设定值频率	INT	ERROR	FB错误	BOOL
MODE	模式源为设定值位置	BOOL	ERRORID	错误代码	INT
			BUSY	未达到设定值位置	BOOL
ERRORID	描述				
0	无错误				
1000h	未启用FI				
1200h	未激活位置控制				
1201h	高位位置尚未输入PLC设定值 (P553)				
1202h	低位位置尚未输入PLC设定值 (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
可用性	X	X	X	X	X	

除了输入**DISTANCE**之外，这在所有点上都与MC_MoveAbsolute相对应。设定值位置由当前实际位置和传输的**DISTANCE**之和产生。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	启用	BOOL	DONE	达到指定的设定值位置	BOOL
DISTANCE	设定值位置	DINT	COMMAND-ABORTED	中止命令	BOOL
VELOCITY	设定值频率	INT	ERROR	FB错误	BOOL
MODE	模式源为设定值位置	BOOL	ERRORID	错误代码	INT
			BUSY	未达到设定值位置	BOOL
ERRORID	描述				
0	无错误				
1000h	未启用FI				
1200h	未激活位置控制				
1201h	高位位置尚未输入PLC设定值 (P553)				
1202h	低位位置尚未输入PLC设定值 (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
可用性	X	X	X	X	X	

如果**EXECUTE**从0更改为1（侧面），则设置变频器的设定值频率。如果变频器已到达设定值频率，则**INVELOCITY**设置为1。在FI加速到设定值频率时，**BUSY**输出激活。如果**EXECUTE**已设置为0，则**INVELOCITY**只能设置为1对应一个周期。如果要中止过程（例如由另一个MC功能模块），则设置**COMMANDABORTED**。

通过**EXECUTE**上负侧面，将所有输出复位为0。

VELOCITY根据以下公式按缩放比例输入：

$$\text{VELOCITY} = (\text{设定值频率 (Hz)} \times 0x4000) / P105$$

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	启用	BOOL	INVELOCITY	达到指定的设定值频率	BOOL
VELOCITY	设定值频率	INT	BUSY	尚未到达设定值频率	BOOL
			COMMAND-ABORTED	中止命令	BOOL
			ERROR	FB错误	BOOL
			ERRORID	错误代码	INT
ERRORID	错误代码				
0	无错误				
1000h	未启用FI				
1100h	FI未在速度模式（位置控制启用）				
1101h	设定值频率未参数化（P553）				

AWL示例：

```
CAL Power
CAL Move
```

```
LD TRUE
ST Power.Enable
```

```
(* Set 20 Hz (Max. 50 Hz) *)
LD DINT#20
MUL 16#4000
DIV 50
```

```
DINT_TO_INT
ST Move.Velocity
```

```
LD Power.Status
ST Move.Execute
```

ST示例:

```
(* Device ready for operation if DIG1 set *)
Power(Enable := _5_State_digital_input.0);
IF Power.Status THEN
  (* Device enabled with 50% of max. frequency if DIG2 set *)
  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
可用性	X	X	X	X	X	X

使用该功能，可以打开和关闭设备的输出级。如果**ENABLE**输入设置为1，则启用输出级。这样操作的条件是设备处于“开关禁止”或“准备打开”状态。如果FI处于“故障”或“故障反应激活”状态，则必须首先纠正和确认故障。只有这时才能通过该块进行启用。如果设备处于“未准备好打开”状态，则不可打开。在所有情况下，FB进入错误状态，必须将**ENABLE**设置为0，以确认故障。

如果**ENABLE**输入设置为0，则设备关闭。如果电机运行时发生这种情况，则首先通过在P103中设置的斜坡将其降至0Hz。

如果打开设备的输出级，则输出**STATUS**为1；否则为0。

如果**ENABLE**切换到0，则**ERROR**和**ERRORID**将复位。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	启用	BOOL	STATUS	电机通电	BOOL
			ERROR	FB错误	BOOL
			ERRORID	错误代码	INT
ERRORID	描述				
0	无错误				
1001h	停止功能激活				
1300h	设备未处于“准备打开”或“开关禁止”状态				

AWL示例:

CAL Power
CAL Move

LD TRUE
ST Power.Enable

(* (* Set 20 Hz (Max. 50 Hz) *) *)
LD DINT#20
MUL 16#4000
DIV 50

DINT_TO_INT
ST Move.Velocity

LD Power.Status
ST Move.Execute

ST示例:

```
(* Enable Power Block *)
Power(Enable := TRUE);
IF Power.Status THEN
  (* The device is ready for switch - on *)
  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
可用性	X	X	X	X	X	

如果**ENABLE**设置为1，则连续交付变频器的实际位置。只要在输出存在有效位置，则**VALID**会设置为有效。如果出现错误，**ERROR**设置为1，在这种情况下，**VALID**为0。

位置缩放比例：1电机旋转 = 1000

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	启用	BOOL	VALID	输出有效	BOOL
			ERROR	FB错误	BOOL
			POSITION	当前位置	DINT

ST示例:

```
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
可用性	X	X	X	X	X	X

只要**ENABLE**设置为1，则就从设备周期性地读取参数。所读取的参数保存在值中，并且如果**DONE**设置为1时，该参数有效。在读取过程中，**BUSY**输出设置为1。如果**ENABLE**保持设置为1，则周期性读取参数。当**ENABLE**激活时，可随时更改参数号和索引。但是，由于**DONE**信号始终保持为1，因此很难识别何时读取新值。在这种情况下，建议将**ENABLE**信号设置为0对应一个周期，因为**DONE**信号随后会复位。参数索引由文件中的索引减去1得出。例如通过参数索引2查询P700索引3（“开关禁止原因”）。如果错误**ERROR**设置为1。在这种情况下，**DONE**设置为0，**ERRORID**包含错误代码。如果**ENABLE**信号设置为0，则删除所有信号和**ERRORID**。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	启用	BOOL	DONE	值有效	BOOL
PARAMETERNUMBER	参数号	INT	ERROR	无法读取	BOOL
PARAMETERINDEX	参数索引	INT	BUSY	过程未完成	BOOL
			ERRORID	错误代码	INT
			VALUE	读取参数	DINT
ERRORID	描述				
0	参数号无效				
3	参数号不正确				
4	无数组				
201	最后接收到指令的指令元件无效				
202	内部响应标签无法描述				

ST示例:

```
(* Motion module 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
可用性	X	X	X	X	X	X

读取设备的状态。状态机的方向符合PLCopen规范“运动控制功能块”。只要**ENABLE**设置为1，则读取状态。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	启用	BOOL	VALID	输出有效	BOOL
			ERROR	FB错误	BOOL
			ERRORSTOP	设备出现错误	BOOL
			DISABLED	设备的输出级关闭	BOOL
			STOPPING	停止命令激活	BOOL
			DISCRETEMOTION	三个定位FB之一激活	BOOL
			CONTINUOUSMOTION	The MC_Velocity is active	BOOL
			HOMING	The MC_Home is active	BOOL
			STANDSTILL	设备没有激活的移动命令。处于转速为0 rpm的停止状态，输出级打开。	BOOL

ST示例:

```

ReadStatus(Enable := TRUE);
IF ReadStatus.Valid THEN
  fError := ReadStatus.ErrorStop;
  fDisable := ReadStatus.Disabled;
  fStopping := ReadStatus.Stopping;
  flnMotion := ReadStatus.DiscreteMotion;
  flnVelocity := ReadStatus.ContinuousMotion;
  flnHome := 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
可用性	X	X	X	X	X	X

从EXECUTE的上升侧面，复位设备中的错误（故障确认）。如果出现错误，则ERROR设置为1，并ERRORID中输入故障原因。通过EXECUTE上负侧面，所有错误复位。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	启动	BOOL	DONE	设备错误复位	BOOL
			ERROR	FB错误	BOOL

			ERRORID	错误代码	INT
			BUSY	复位过程仍激活	BOOL
ERRORID	描述				
0	无错误				
1001h	停止功能激活				
1700h	无法执行错误复位，因为错误的原因仍然存在。				

ST示例:

```

Reset(Execute := TRUE);
IF Reset.Done THEN
  (* The error has been reset *)
  Reset(Execute := FALSE);
ELSIF Reset.Error THEN
  (* Reset could not be executed, as the cause of the error is still present *)
  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
可用性	X	X	X	X	X	X

当侧面上升时（0到1），设备设置为**STANDINGSTILL**状态。取消所有激活的移动功能。设备制动到0Hz，并关闭输出级。只要激活停止命令（**EXECUTE=1**），则阻止所有其它的移动FB。通过**EXECUTE**上的上升侧面，**BUSY**输出变为激活，并且保持激活状态直至**EXECUTE**上出现一个下降侧面。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	启动	BOOL	DONE	已执行命令	BOOL
			BUSY	命令激活	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
可用性	X	X	X	X	X	X

如果**EXECUTE**从0更改为1（侧面），则将16/32位参数写入设备。如果**DONE**设置为1，则已写入参数。在读取过程中，**BUSY**输出设置为1。如果出现错误，**ERROR**设置为1，**ERRORID**包含错误代码。信号**DONE**，**ERROR**，**ERRORID**保持设置，直到**EXECUTE**更改回0。如果**EXECUTE**信号更改为0，则不取消写入过程。只有**DONE**信号保持设置1个PLC周期。如果输入

RAMONLY设置为1，则值仅存储在RAM中。当关闭设备时，更改的设置将丢失。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	启用	BOOL	DONE	值有效	BOOL
PARAMETERNUMBER	参数号	INT	BUSY	写入过程激活	BOOL
PARAMETERINDEX	参数索引	INT	ERROR	无法读取	BOOL
VALUE	要写入的值	INT	ERRORID	错误代码	INT
RAMONLY	将值仅存储在RAM中 (从V2.1起)	BOOL			
ERRORID	描述				
0	参数号无效				
1	无法更改参数值				
2	超过较低或较高值限				
3	参数号不正确				
4	无数组				
5	数据类型无效				
6	仅可复位 (仅可写入0)				
7	无法更改描述元素				
201	最后接收到指令的指令元件无效				
202	内部响应标签无法描述				

ST示例:

```

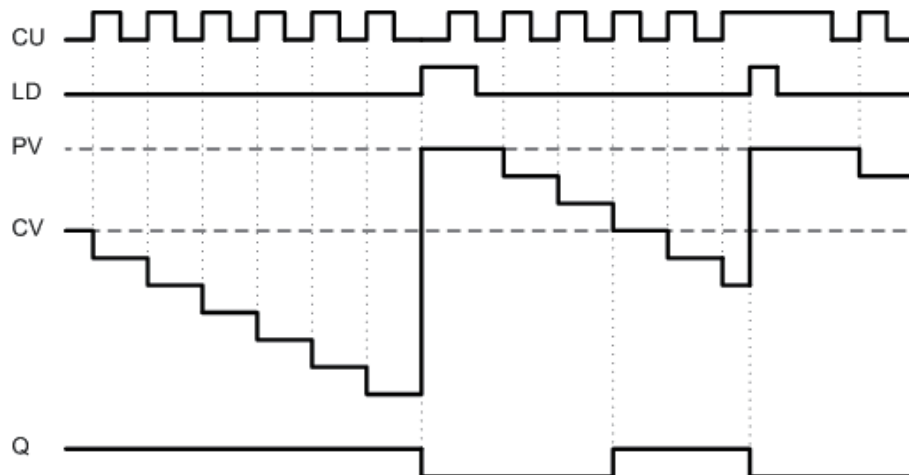
WriteParam16 (Execute := TRUE, ParameterNumber := 102, ParameterIndex := 0, Value := 300);
IF WriteParam16.Done THEN
    WriteParam16 (Execute := FALSE);
END_IF;
    
```

10.3.4 标准

10.3.4.1 CTD向下计数器

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

CD上侧面上升时，只要CV大于-32768，则功能块CV的计数器就减少一个。如果CV小于或等于0，则输出Q仍为TRUE。通过LD，计数器CV可以设置为在PV中保存的值。



VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
CD	计数器输入	BOOL	Q	TRUE, 如果CV <= 0	BOOL
LD	加载启动值	BOOL	CV	读取实际计数器	INT
PV	启动值	INT			

AWL示例:

```
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
```

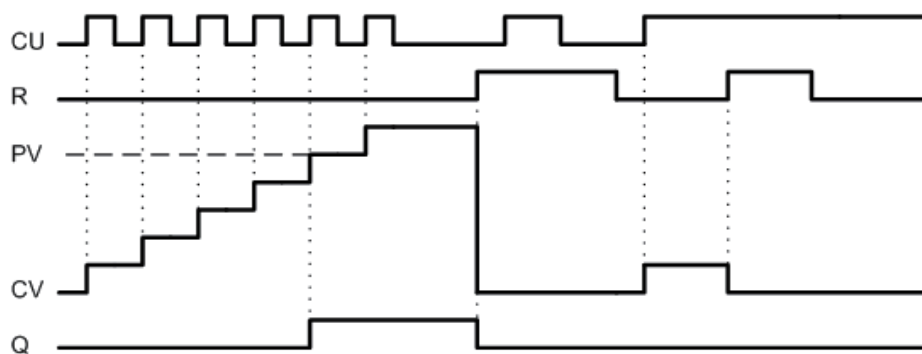
ST示例:

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

10.3.4.2 CTU向上计数器

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

CU上侧面上升时，则功能块CV的计数器就增加一个。CV可以计数多达值32767。只要CV大于或等于PV，则输出Q仍为TRUE。通过R，计数器CV可以复位到零。



VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
CU	计数器输入	BOOL	Q	TRUE, 如果CV >= PV	BOOL
R	复位: 读取计数器	BOOL	CV	读取实际计数器	INT
PV	最大计数数值	INT			

AWL示例:

```

LD VarBOOL1
ST CTUInst.CU
LD VarBOOL2
ST CTUInst.R
LD VarINT1
ST CTUInst.PV
CAL CTUInst
LD CTUInst.Q
ST VarBOOL3
LD CTUInst.CV
ST VarINT2
    
```

ST示例:

```

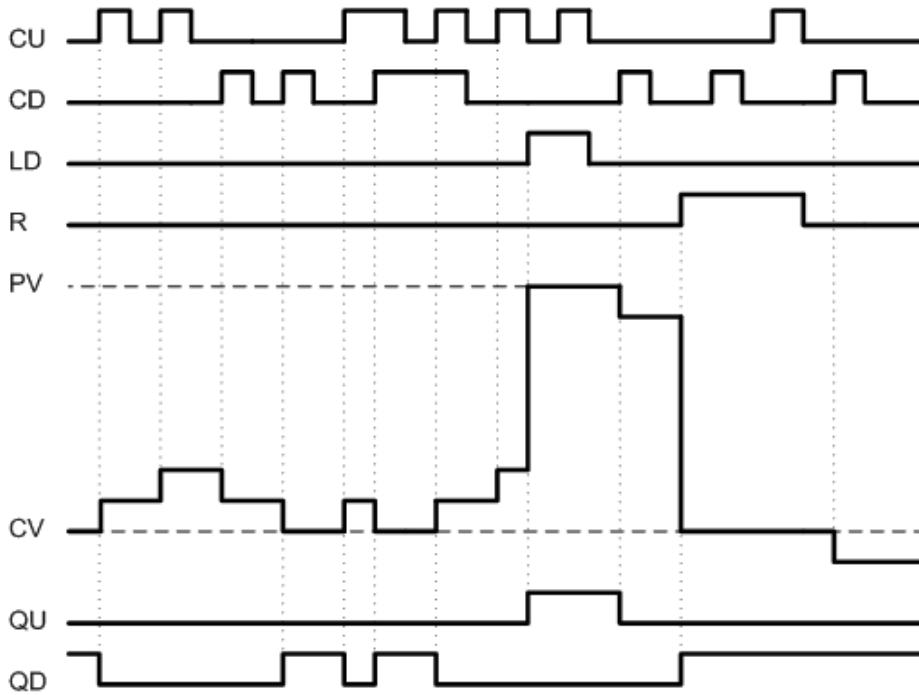
CTUInst (CU := VarBOOL1, R := VarBOOL2, PV := VarINT1);
VarBOOL3 := CTUInst.Q;
VarINT2 := CTUInst.CV;
    
```

10.3.4.3 CTUD向上和向下计数器

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

CU上侧面上升时, 只要CV小于32767, 则计数器CV就增加一个。CD上侧面上升时, 只要CV大于-32768, 则计数器CV就减少一个。通过R, 计数器CV可以设置为零。通过LD, 将在PV中保存的值复制到CV。

R优先于LD, CU和CV。PV可随时更改, QU始终与当前设置的值相关。



VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
CU	向上计数	BOOL	QU	TRUE, 如果CV >= PV	BOOL
CD	向下计数	BOOL	QD	TRUE, 如果CV <= 0	BOOL
R	复位: 读取计数器	BOOL	CV	读取实际计数器	INT
LD	加载启动值	BOOL			
PV	启动值	INT			

AWL示例:

```

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

ST示例:

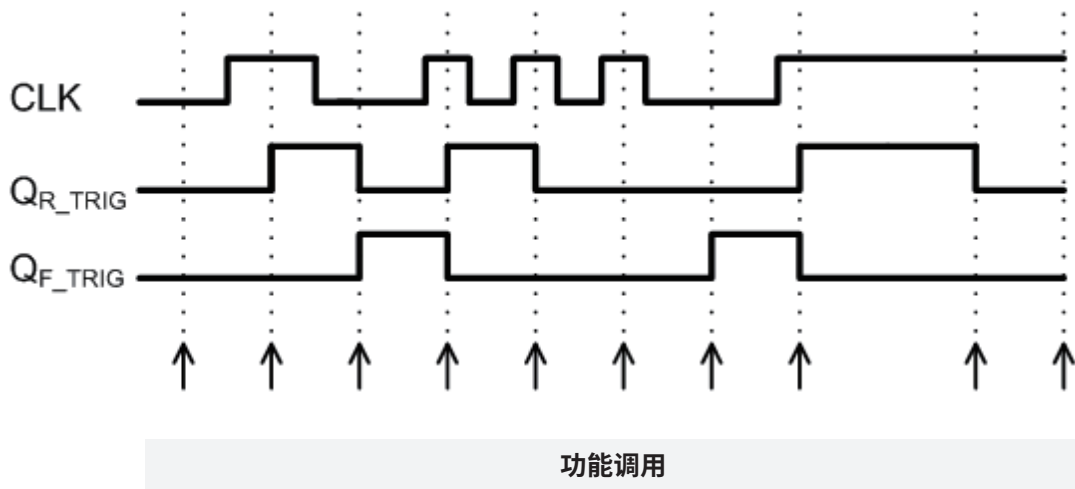
```
CTUDInst (CU:=VarBOOL1, R:=VarBOOL3, LD:=VarBOOL4, PV:=VarINT1);
VarBOOL5 := CTUDInst.Q;
VarBOOL5 := CTUDInst.QD;
VarINT2 := CTUDInst.CV;
```

10.3.4.4 R_TRIG und F_TRIG

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

两项功能都用于侧面检测。如果在CLK上检测到侧面，则Q设置为TRUE，直到下一次调用功能，之后将复位到FALSE。只有通过新侧面，Q才能再次变为TRUE对应一个周期。

- R_TRIG = 侧面上升
- F_TRIG = 侧面下降



VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
CLK	设置	BOOL	Q	输出	BOOL

AWL中的示例:

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

ST中的示例:

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

i 信息

只有当调用功能时功能输出才更改。因此，建议继续通过PLC周期调用侧面检测。

10.3.4.5 RS触发器

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

双稳态功能：通过**S**设置输出**Q1**，通过**R1**再次删除**Q1**。如果**R1**和**S**都为**TRUE**，则**R1**为主导。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
S	设置	BOOL	Q1	输出	BOOL
R1	重置	BOOL			

AWL中的示例：

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

ST中的示例：

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

10.3.4.6 SR触发器

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

双稳态功能：通过**S1**设置输出**Q1**，通过**R**再次删除**Q1**。如果**R**和**S1**都为**TRUE**，则**S1**为主导。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
S1	设置	BOOL	Q1	输出	BOOL
R	重置	BOOL			

AWL示例:

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

ST示例:

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

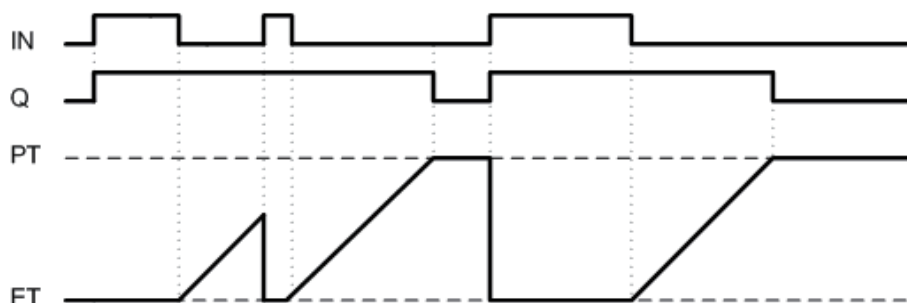
10.3.4.7 TOF关闭延时继电器

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

如果IN=TRUE，则Q设置为TRUE。如果IN更改为FALSE，则计时器向上计数。只要计时器在运行 (ET < PT)，则Q保持设置为TRUE。如果 (ET=PT)，则计时器停止，Q变为FALSE。在IN上有新上升的侧面时，计时器ET复位到零。

这里，文字可用于简单的输入，例如

- LD TIME#50s20ms = 50.020秒
- LD TIME#1d30m = 1天30分钟



VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
IN	计时器激活	BOOL	Q	TRUE & (ET<PT)	BOOL
PT	持续时间	DINT	ET	读取当前计时器	DINT

AWL示例:

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

ST示例:

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

信息 计时器ET

计时器ET独立于PLC周期运行。用IN启动计时器并设置输出Q只能通过功能调用“CAL”执行。功能调用在PLC周期内进行。但是，对于长度超过5 ms的PLC程序，这可能会导致出现抖动。

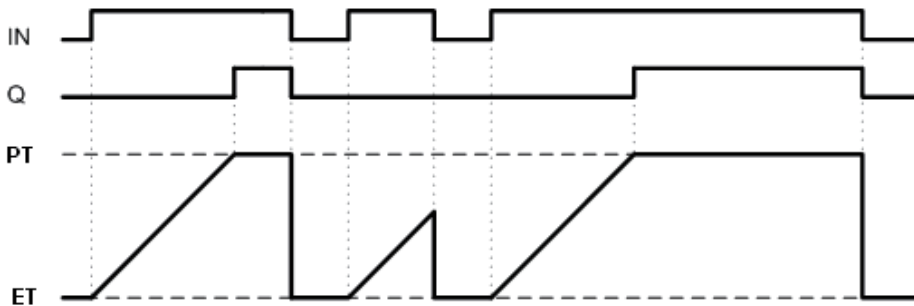
10.3.4.8 TON打开延迟

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

如果设置IN=TRUE，则计时器向上计数。如果ET=PT，则Q设置为TRUE，并且计时器停止。只要IN仍为TRUE，则Q保持为TRUE。在IN上有新上升的侧面时，计数器从零再次启动。计时器运行时，可以更改PT。PT中的时间段以ms为单位输入。这样使时间延迟在5 ms和24.8天之间。由于PLC的时基为5 ms，因此最小时间延迟也为5 ms。

这里，文字可用于简单的输入，例如

- LD TIME#50s20ms = 50.020秒
- LD TIME#1d30m = 1天30分钟



VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
IN	计时器激活	BOOL	Q	TRUE & (IN=TRUE & ET=PT)	BOOL
PT	持续时间	DINT	ET	读取当前计时器	DINT

AWL示例:

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

ST示例:

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

信息 计时器ET

计时器ET独立于PLC周期运行。用IN启动计时器并设置输出Q只能通过功能调用“CAL”执行。功能调用在PLC周期内进行。但是，对于长度超过5 ms的PLC程序，这可能会导致出现抖动。

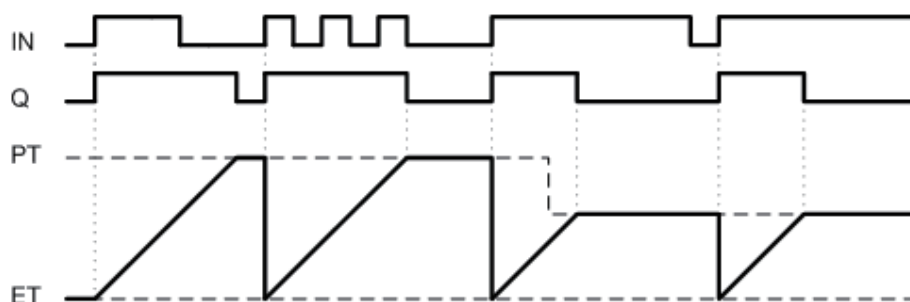
10.3.4.9 TP时间脉冲

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

IN上有正侧面时，计时器以值0启动。计时器运行到PT中输入的值，然后停止。这过程不得中断！在计数期间可以更改PT。只要计时器ET小于PT，则输出Q为TRUE。如果ET=PT，并且在IN上检测到上升的侧面，则计时器再次以零启动。

这里，文字可用于简单的输入，例如

- LD TIME#50s20ms = 50.020秒
- LD TIME#1d30m = 1天30分钟



VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
IN	计时器激活	BOOL	Q	TRUE β (ET<PT)	BOOL
PT	持续时间	DINT	ET	读取当前计时器	DINT

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	执行	BOOL	VALID	读取过程成功	BOOL
SIZE	存储器格式	BOOL	READY	已读取整个存储器	BOOL
MEMORY	选择存储区域	字节	ERROR	FB出现错误	BOOL
STARTINDEX	表示要写入的存储单元	INT	ERRORID	错误代码	INT
			ACTINDEX	实际存储器索引，将在下一个周期读取	INT
			VALUE	读取的值	DINT
ERRORID	描述				
0	无错误				
1A00h	超过SARTINDEX值范围				
1A01h	超过MEMORY值范围				

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
可用性	X	X	X	X		

通过该FB，单个值或大量值可以中间保存在FI中。这些值不是永久保存，即如果重新启动FI，这些值将丢失。

如果FB在ENABLE检测到正侧面，则采用输入处存在的所有参数。将VALUE中的值写入在SATARTINDEX和MEMORY中显示的存储地址。如果写入过程成功，则VALID输出更改为1。

如果现在调用几次FB，并且ENABLE输入保持为1，则每次调用FB时，读取并保存输入VALUE，并且存储器地址将增加1。下一个访问的实时存储器索引可以在输出ACTINDEX下读取。如果到达存储器的末尾，则输出FULL更改为1，并且保存过程停止。但是，如果输入OVERWRITE设置为1，则存储器索引复位为STARTINDEX，并且覆盖先前写入的值。

值可以以INT或DINT的格式保存。对于INT值，VALUE输入仅评估低位分量。通过SIZE输入进行分配；0代表INT值，1代表DINT值。

通过MEMORY输入进行存储器区域的分配：

MEMORY = 1 à P613[0-251] 对应于504 INT或252 DINT值

MEMORY = 0 à P900[0-247] 至 P906[0-111] 对应于3200 INT或1600 DINT值

其它块不可中断FB。通过ENABLE上的负侧面，所有输出设置为0，并且FB的功能结束。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	执行	BOOL	VALID	写入过程成功	BOOL
SIZE	存储器格式	BOOL	FULL	整个存储器已满	BOOL
OVERWRITE	可覆盖内存	BOOL	ERROR	FB出现错误	BOOL
MEMORY	选择存储区域	字节	ERRORID	错误代码	INT
STARTINDEX	表示要写入的存储单元	INT	ACTINDEX	实际存储器索引，下一个周期将对其进行保存	DINT
VALUE	要保存的值	DINT			
ERRORID	描述				
0	无错误				
1A00h	超过SARTINDEX值范围				
1A01h	超过MEMORY值范围				


信息

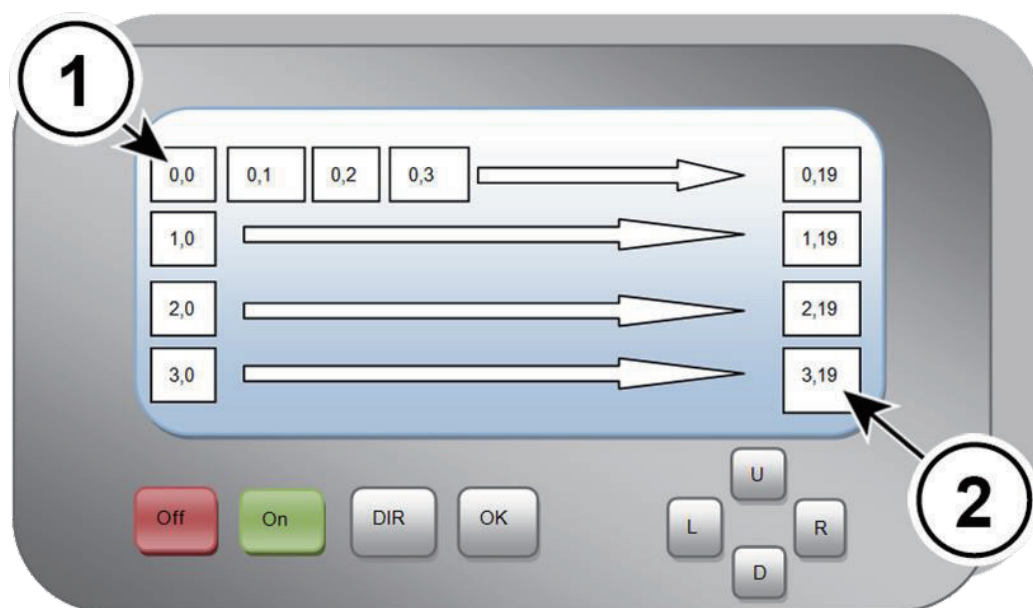
注意! 范围功能也可以使用设置MEMEORY=0中的存储区。使用范围功能会覆盖所保存的值!

10.3.6 参数盒可视化

在参数盒中，整个显示可用于信息的显示。为此，必须将参数盒切换到可视化模式。这可以通过参数盒（参数P1308）固件版本V4.3或更高实现，并按以下执行：

- 在“显示”菜单项中，将参数P1003设置为“PLC显示”
- 使用左或右箭头键切换到操作值显示
- 现在在参数盒中启用PLC显示，并保持永久启用。

在参数盒的可视化模式下，显示的内容可以通过以下所述的两个FB进行设置。但是，“允许参数盒功能模块”项之前必须在PLC配置对话框（按钮 ）中激活。使用过程值“Parameterbox_key_state”，可以查询参数盒的键盘状态。这样，可以执行输入进PLC程序。参数盒的显示结构和要读取的键如下图所示。



1	第一个字符	(0,0 → 行 = 0, 列 = 0)
2	最后一个字符	(3,19 → 行 = 3, 列 = 19)

10.3.6.1 可视化概述

功能模块	描述
FB_STRINGToPBox	将字符串复制到参数盒中
FB_DINTToPBox	将DINT值复制到参数盒中

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
可用性	X	X	X	X	X	X

该功能模块将DINT值转换为ASCII字符串，并将其复制到参数盒中。输出可以是十进制、二进制或十六进制格式；通过MODE进行选择。通过ROW和COLUMN，在参数盒显示中设置字符串的起始点。参数LENGTH在字符串中传输字符串的长度。在十进制MODE下，参数POINT将要显示的数字定位到小数点。在POINT中表明小数点右边有多少个字符。设置为0时，POINT功能禁用。如果数字包含的字符超过了长度允许的范围，并且未设置小数点，则溢出由字符“#”表示。如果数字中有小数点，则如果需要，可以省略小数点后的所有数字。在十六进制和二进制MODE下，如果设置长度太短，则显示最低值位。只要ENABLE设置为1，将立即采用输入的所有更改。如果VALID更改为1，则已正确传输字符串。如果错误ERROR设置为1。在这种情况下，VALID为0。则在ERRORID中，相关错误代码有效。通过ENABLE上的负侧面，复位VALID、ERROR和ERRORID。

示例：

设置	要显示的数字	参数盒显示
Length = 5	12345	12345
Point = 0		
Length = 5	-12345	#####
Point = 0		
Length = 10	123456789	123456,789
Point = 3		
Length = 8	123456789	123456,7
Point = 3		

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	传输值	BOOL	VALID	传输的值	BOOL
MODE	显示格式 0 = 十进制 1 = 二进制 2 = 十六进制 值范围 = 0至2	字节	ERROR	FB错误	BOOL
ROW	显示行 值范围 = 0至3	字节	ERRORID	错误代码	INT
COLUMN	显示列 值范围 = 0至19	字节			
POINT	小数点位置 值范围 = 0至10 0 = 禁用功能	字节			
LENGTH	输出长度值 范围 = 1至11	字节			
VALUE	要输出的数字	DINT			
ERRORID	描述				
0	无错误				
1500h	字符串覆盖参数盒数组的存储区域				
1501h	LINE输入超出值范围				
1502h	ROW输入超出值范围				
1504h	POINT输入超出值范围				
1505h	LENGTH输入超出值范围				
1506h	MODE输入超出值范围				

ST示例:

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

(* Query actual position *)
ActPos(Enable := TRUE);
if ActPos.Valid then
  (* Display position in the PBox displays (PBox P1003 = PLC display) *)
  DintToPBox.Value := ActPos.Position;
  DintToPBox.Column := 9;
  DintToPBox.LENGTH := 10;
  DintToPBox(Enable := True);
end_if;

(* Switch device on or off via DIG1 *)
Power(Enable := _5_State_digital_input.0);
if OldState <> Power.Status then
  OldState := Power.Status;
  (* Is device switched on? *)
  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_STRINGTOBOX

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

该功能模块将字符串（字符链）复制到参数盒的存储数组中。通过ROW和COLUMN，在参数盒显示中设置字符串的起始点。参数TEXT将所需的字符串传输到功能模块；字符串的名称可以从变量表中获取。只要ENABLE设置为1，将立即采用输入的所有更改。如果设置了CLEAR输入，则在写入所选的字符串之前，将用空字符覆盖整个显示内容。如果VALID更改为1，则已正确传输字符串。如果错误ERROR设置为1。在这种情况下，VALID为0。则在ERRORID中，相关错误代码有效。通过ENABLE上的负侧面，复位VALID、ERROR和ERRORID。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	传输字符串	BOOL	VALID	传输的字符串	BOOL
CLEAR	清除显示	BOOL	ERROR	FB错误	BOOL
ROW	显示行 值范围 = 0至3	字节	ERRORID	错误代码	INT
COLUMN	显示列 值范围 = 0至19	字节			
TEXT	要显示的文本	STRING			
ERRORID	描述				
0	无错误				
1500h	字符串覆盖参数盒数组的存储区域				
1501h	ROW输入超出值范围				
1502h	COLUMN输入超出值范围				
1503h	所选的字符串数字不存在				
1506h	“允许参数盒功能模块”选项未在PLC配置中激活。				

ST示例:

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

(* Query actual position *)
ActPos(Enable := TRUE);
if ActPos.Valid then
  (* Display position in the PBox displays (PBox P1003 = PLC display) *)
  DintToPBox.Value := ActPos.Position;
  DintToPBox.Column := 9;
  DintToPBox.LENGTH := 10;
  DintToPBox(Enable := True);
end_if;

(* Switch device on or off via DIG1 *)
Power(Enable := _5_State_digital_input.0);
if OldState <> Power.Status then
  OldState := Power.Status;
  (* Is device switched on? *)
  if Power.Sta    tus 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 (检测快速事件)

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

PLC的周期时间为5 ms。该周期可能太长而无法检测非常快的外部事件。通过FB采集，可以采集FI输入侧面的一些物理值。以1 ms为周期监测输入。保存的值可随后由PLC读取。

通过EXECUTE上的正侧面，将读取所有输入，并且启用采集功能。通过INPUT输入选择要监测的FI输入。通过EDGE，可以选择侧面的类型和模块的行为。

EDGE = 0 对于第一个正侧面，所选的值保存在OUTPUT1下，并且DONE1设置为1。下一个正侧面在OUTPUT2下保存，并且DONE2设置为1。FB则禁用。

EDGE = 1 EDGE=0时的行为，不同之处在于通过负侧面触发。

EDGE = 2 对于第一个正侧面，所选的值保存在OUTPUT1下，并且DONE1设置为1。下一个正侧面在OUTPUT2下保存，并且DONE2设置为1。FB则禁用。

EDGE = 3 EDGE=2时的行为，不同之处在于首先通过负侧面然后通过正侧面触发。

如果输入CONTINUOUS设置为1，则仅设置=1与EDGE相关。FB继续运行，并将始终将上一个触发事件保存在OUTPUT1下。从第一个事件开始，DONE1保持激活状态。不使用DONE2和OUTPUT2。

BUSY输出保持激活状态，直到发生两个采集事件（DONE1和DONE2）。

通过EXECUTE的负侧面可随时终止模块功能。所有输出都保留其值。首先通过EXECUTE上的正侧面，删除所有输出，然后启动模块功能。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	执行	BOOL	DONE1	OUTPUT1中的值有效	BOOL
CONTINUOUS	单一执行或连续操作	BOOL	DONE2	OUT中的值有效	BOOL
INPUT	SK54xE 要监测的输入 0 = 输入1 ---- 7 = 输入8 SK52xE, SK53xE, SK2xxE, SK2xxE FDS 0 = 输入1 ---- 3 = 输入4	字节	BUSY	FB仍等待采集事件	BOOL
EDGE	触发侧面	字节	ERROR	FB出现错误	BOOL
SOURCE	要保存的值 0 = 旋转位置 1 = 实际频率 2 = 扭矩	字节	ERRORID	错误代码	INT
			OUTPUT1	第一次采集事件的值	DINT
			OUTPUT2	第二次采集事件的值	DINT
ERRORID	描述				
0	无错误				
1900h	超过INPUT值范围				
1901h	超过EDGE值范围				
1902h	超过SOURCE值范围				
1903h	两个以上实例激活				

ST示例:

```

Power(ENABLE := TRUE);
IF Power.STATUS THEN
  Move(EXECUTE := TRUE, POSITION := Pos, VELOCITY := 16#2000);
  (* The FB waits for a High signal on DIG1. If this
    is detected, the FB saves the actual position. The value can
    be queried with the property "OUTPUT1". *)
  Capture(EXECUTE := TRUE, INPUT := 0);

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

```

i 信息

PLC程序中可能存在几个该FB的实例。但是，同一时间仅能激活两个实例！

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
可用性	V2.3 及以上	V3.1 及以上	V2.1 及以上	X	V1.1 及以上	

该功能块用于计数数字输入的脉冲。所有侧面（低-高和高-低）都计数在内。最小脉冲宽250µs。

通过ENABLE激活功能块。通过正侧面，采用输入PV、UD、DIN和MODE，并且删除所有输出。

UD定义计数方向

- 0 = 较大数字
- 1 = 较小数字

可以在PV中输入计数器值。这有不同效果，取决于MODE输入的设置。

MODE

- 0 = 溢出，计数器用作连续计数器。可以以正向或负向溢出。当功能启动时，则设置CV=PV。在该模式下，BUSY始终保持为1，并且Q始终保持为0。
- 1 = 无溢出
 - 向前计数：CV从0启动，BUSY=1，运行直到CV=>PV。然后，BUSY变为0，Q变为1。计数过程停止。
 - 向后计数：CV从PV启动，运行直到CV<=0。这期间BUSY=1，并且当到达计数结束时，切换到0。因此，Q变为1。
 - 计数器通过ENABLE输入的新侧面重新启动

DIN定义测量输入。输入的数量取决于特定的FI。

- 输入1 = 0
- 输入2 = 1

等

VAR_INPUT			VAR_OUTPUT		
输入	解释	型号	输出	解释	型号
ENABLE	启用	BOOL	Q	计数完成	BOOL
UD	计数方向 0 = 较大数字 1 = 较小数字	BOOL	BUSY	计数器运行	BOOL
PV	计数器值	INT	ERROR	FB出现错误	BOOL
MODE	模式	字节	ERRORID	错误代码	INT
DIN	测量输入	字节	CV	计数器值	INT
			CF	计数频率 (分辨率0.1)	INT
ERRORID	解释				
0	无错误				
0x1E00	另一个计数器已使用数字输入				
0x1E01	数字输入不存在				
0x1E02	超过数值范围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
可用性	X	X	X	X	X	

该功能模块产生映射控制。所定义的点可以传输到功能块，这样可以模拟功能。输出则根据所保存的映射运行。在各个基点之间进行线性插值。基点由X和Y值定义。X值始终为INT型，而Y值可以是INT型，也可以是DINT型，这取决于最大基点的大小。如果使用DINT，则需要更多内存。在变量窗口的“初始值”列中输入基点。如果在ENABLE输入中检测到TRUE，则根据输入值INVALUE计算相应的输出值OUTVALUE。VALID=TRUE表示输出值OUTVALUE有效。只要VALID为FALSE，则输出OUTVALUE的值为0。如果输入值INVALUE超过特性范围的上限或下限，则特性范围的第一个或最后一个输出值保持不变，直到INVALUE返回特性范围的区域内。如果超过或低于特性范围，则相应的输出MINLIMIT或MAXLIMIT设置为TRUE。如果特性范围的横坐标值（X值）未持续增加，或者表格未初始化，则ERROR变为TRUE。相应的错误由ERRORID输出，起始值为0。如果ENABLE=FALSE，则错误复位。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	执行	BOOL	VALID	输出值有效	BOOL
INVALUE	输入值 (X)	INT	ERROR	FB错误	BOOL
			ERRORID	错误代码	INT
			MAXLIMIT	达到的最大限值	BOOL
			MINLIMIT	达到的最小限值	BOOL
			OUTVALUE	输出值 (Y)	DINT
ERRORID	描述				
0	无错误				
1400h	特性范围的横坐标值 (X值) 未能始终增加				
1401h	映射未初始化				

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
可用性	X	X	X	X	X	X

P-I-DT1是可自由参数化的单独控制器。如果不需要单个分量或P、I或DT1分量，则将其参数写入为0。T1分量仅能与D分量一起使用。因此，PT1控制器无法参数化。由于内部存储器的限制，控制参数仅限于以下区域：

控制参数的允许值范围			
参数	值范围	缩放比例	结果值范围
P (Kp)	0 – 32767	1/100	0.00 – 32.767
I (Ki)	0 – 10240	1/100	0.00 – 10.240
D (Kd)	0 – 32767	1/1000	0.000 – 3.2767
T1 (ms)	0 – 32767	1/1000	0.000 – 3.2767
最大	-32768 – 32767		
最小	-32768 – 32767		

当ENABLE设置为TRUE时，控制器开始计算。控制参数仅能通过ENABLE的上升侧面进行采用。当ENABLE为TRUE时，控制参数的更改无效。如果ENABLE设置为FALSE，则输出保持为其最后一个值。

只要Q的输出值在最小和最大限值之内，并且ENABLE输入为TRUE，则设置输出位VALID。

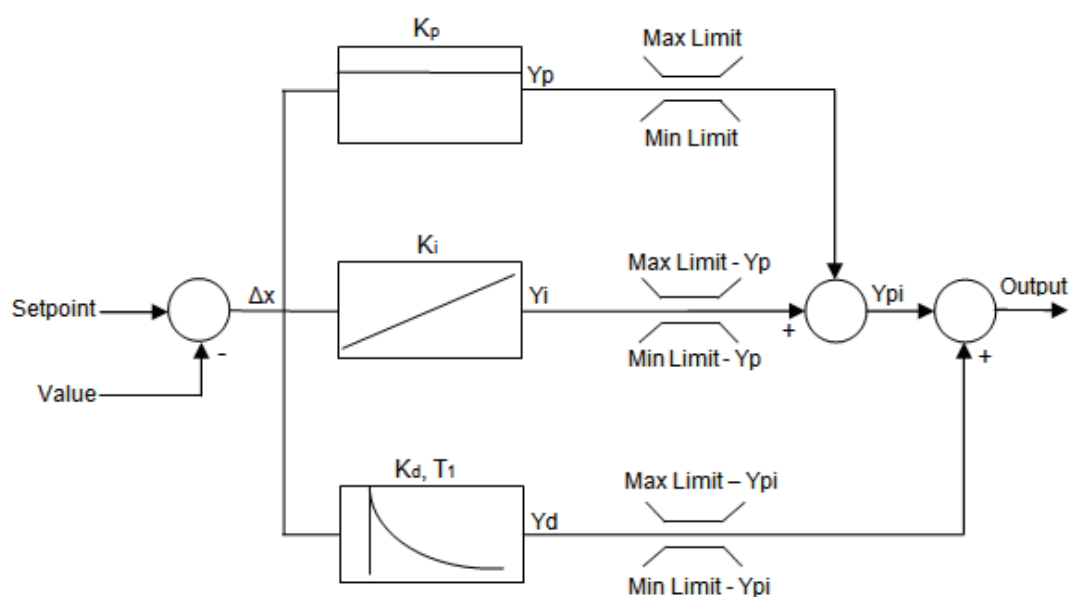
只要出现错误就设置ERROR。VALID位则为FALSE，并且可以从ERRORID中识别故障原因（请见下表）。

如果RESET位设置为TRUE，则积分器和微分器的内容设置为0。如果ENABLE输入为FALSE，则OUTPUT输出也设置为0。如果ENABLE输入设置为TRUE，则仅P分量对OUTPUT输出有影响。

如果输出值OUTPUT在最大或最小输出值范围之外，则设置相应位MAXLIMIT或MINLIMIT，并且VALID位设置为FALSE。

i 信息

如果无法在一个PLC周期内执行整个程序，则控制器通过旧的扫描值再次计算输出值。这确保了恒定的扫描速率。因此，必须在每个PLC周期并且仅在PLC程序结束时执行PIDT1控制器的CAL命令。



VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
ENABLE	执行	BOOL	VALID	输出值有效	BOOL
RESET	复位输出	BOOL	ERROR	FB错误	BOOL
P	P分量 (Kp)	INT	ERRORID	错误代码	INT
I	I分量 (Ki)	INT	MAXLIMIT	达到的最大限值	BOOL
D	D分量 (Kd)	INT	MINLIMIT	达到的最小限值	BOOL
T1	以ms为单位的T1分量	INT	OUTPUT	输出值	INT
MAX	最大输出值	INT			
MIN	最小输出值	INT			
SETPOINT	设定值	INT			
VALUE	实际值	INT			
ERRORID	描述				
0	无错误				
1600h	P分量未在值范围之内				
1601h	I分量未在值范围之内				
1602h	D分量未在值范围之内				
1603h	T1分量未在值范围之内				

10.3.11 FB_ResetPostion

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	V2.3 及以上	V3.1 及以上	V2.1 及以上	X	V1.2 及以上	

EXECUTE输入有侧面时，实际位置设置为在位置中输入的值。对于绝对值编码器，只能将实际位置复位为0。不使用位置中的值。

VAR_INPUT			VAR_OUTPUT		
输入	解释	型号	输出	解释	型号
EXECUTE	执行	BOOL			
Position	位置	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
可用性	V2.3 及以上	V3.1 及以上	V2.1 及以上	X	V1.2 及以上	

该模块用于确定在匀速移动时的平均扭矩。根据该值，可以确定物理值，例如正在移动的重量。

通过EXECUTE输入的正侧面启动FB。通过侧面，FB采用所有输入。FI以在SPEED中设置的速度移动。在STARTTIME设置的时间过后启动测量。测量持续时间在MEASURETIME下进行定义。测量时间过后，FI停止。如果输入REVERSE=1，则再次启动测量过程，但速度为负。否则测量完成，输出DONE更改为1，并且测量结果在VALUE中。

只要测量过程在运行，则BUSY为激活状态。测量结果VALUE的缩放比例为1=电机额定扭矩的0.01%。另一个移动FB的调用将停止测量功能，并且输出ABORT更改为1。所有FB输出通过EXECUTE的新正侧面复位。

VAR_INPUT			VAR_OUTPUT		
输入	描述	型号	输出	描述	型号
EXECUTE	执行	BOOL	DONE	测量结束	BOOL
REVERSE	更改旋转方向	BOOL	BUSY	测量运行	BOOL
STARTTIME	以ms为单位的测量开始时间	INT	ABORT	测量中止	BOOL
MEASURETIME	以ms为单位的测量时间	INT	ERROR	FB出现错误	BOOL
SPEED	以百分比为单位的测量速度 (标准化到最大频率， 16#4000对应于100%)	INT	ERRORID	错误代码	INT
			VALUE	测量结果	INT
ERRORID	描述				
0	无错误				
0x1000	FI未打开				
0x1101	设定值频率未参数化为设定值 (P553)				
0x1C00	超过STARTTIME值范围				
0x1C01	超过MEASURETIME值范围				
0x1C02	测量值相互之间的公差大于1/8				

ST示例:

```
(* Enable device *)
Power(Enable := TRUE);
(* Is the device enabled? *)
if Power.Status then
  (* Specify starting time 2000 ms *)
  Weigh.STARTTIME := 2000;
  (* Specify measuring time 1000 ms *)
  Weigh.MEASURETIME := 1000;
  (* Specify speed 25% of maximum speed *)
  Weigh.SPEED := 16#1000;
end_if;

Weigh(EXECUTE := Power.Status);
(* Was weighing completed? *)
if Weigh.done then
  Value := Weigh.Value;
end_if;
```

i 信息

PLC程序中仅允许有一个FB实例!

10.4 运算符

10.4.1 算术运算符

i 信息

以下一些运算符可能包含其它命令。这些必须放置在运算符后的括号中。必须注意的是，在开括号后必须有一个空格。结束括号必须放置在程序的单独行上。

```
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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型			X	X		

形成累加器的绝对值。

AWL示例:

```
LD - 10      (* Load the value - 10 *)
ABS          (* Accumulator = 10 *)
ST Value1    (* Saves the value 10 in Value1 *)
```

ST示例:

```
Value1 := ABS(- 10); (* The result is 10 *)
```

10.4.1.2 ADD和ADD (

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

添加变量和常量以及正确的前缀。第一个添加的值在AE/累加器中，第二个则用ADD命令加载，或在括号内。可以将几个变量或常量添加到ADD命令中。对于括号添加，将累加器添加到括号中表达式的结果中。最多可以有6个括号等级。要添加的值必须属于同一类型的变量。

AWL示例:

```
LD 10
ADD 204          (* Addition of two constants *)
ST Value
LD 170          (* Addition of a constant and 2 variables. *)
ADD Var1, Var2  (* 170dez + Var1 + Var2 *)
ST Value
LD Var1
ADD( Var2
SUB Var3        (* Var1 + ( Var2 - Var3 ) *)
)
ST Value
```

ST示例:

```
Result := 10 + 30; (* The result is 40 *)
Result := 10 + Var1 + Var2;
```

10.4.1.3 DIV和DIV (

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

将累加器除以操作数。对于零除，将最大的可能结果输入累加器中，例如，对于具有INT值的除法，如果除数为负，则值为0x7FFF或值0x8000。对于括号除法，将累加器除以括号中表达式的结果。最多可以有6个括号等级。要除的值必须属于同一类型的变量。

AWL示例:

```
LD 10
DIV 3          (* Division of two constants *)
ST iValue     (* The result is 9 *)
LD 170        (* Division of a constant and 2 variables. *)
DIV Var1, Var2 (* (170dez : Var1) : Var2 *)
ST Value
LD Var1
DIV( Var2     (* Divide Var1 by the content of the brackets *)
SUB Var3
)
ST Value     (* Var1 : (Var2 - Var3) *)
```

ST示例:

Result := 30 / 10; (* The result is 3 *)
 Result := 30 / Var1 / Var2;

10.4.1.4 极限

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

命令将累加器中的值限制为传输的最小和最大值。如果超过该限制，则在累加器中输入最大值，如果低于该限制，则输入最小值。如果值介于限制之间，则没有影响。

AWL示例:

LD 10 (* Loads the value 10 into the accumulator *)
 LIMIT 20, 30 (* The value is compared with the limits 20 and 30. *)
 (* The value in the accumulator is smaller, the Accumulator is overwritten with 20 *)
 ST iValue (* Saves the value 20 in Value1 *)

ST示例:

Result := Limit(10, 20, 30); The result is 20 *)

10.4.1.5 最大值

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

该值确定两个变量或常量的最大值。为此，将累加器的当前值与在MAX命令中传输的值进行比较。命令发出后，两个值中的较大值在累加器中。两个值必须都属于同一类型的变量。

AWL示例:

LD 100 (* Load 100 into the accumulator *)
 MAX 200 (* Compare with the value 200 *)
 ST iValue (* Save 200 in Value2 (because larger value) *)

ST中的示例:

Result := Max(100, 200); (* The result is 200 *)

10.4.1.6 最小值

	SK 54xE	SK 53xE	SK 2xxE	SK 2xxE-FDS	SK 180E	SK 155E-FDS
可用性	X	X	X	X	X	X

		SK 52xE				SK 190E	SK 175E-FDS
可用性	X	X	X	X	X	X	
	BOOL	BYTE	INT	DINT			
数据类型		X	X	X			

该命令确定两个变量或常量的最小值。为此，将累加器的当前值与在MIN命令中传输的值进行比较。命令发出后，两个值中的较小值在累加器中。两个值必须都属于同一类型的变量。

AWL示例:

```
LD 100 (* Load 100 into the accumulator *)
MIN 200 (* Compare with the value 200 *)
ST Value2 (* Save 100 in Value2 (because smaller value) *)
```

ST示例:

```
Result := Min(100, 200); (* Save 100 in Value2 (because smaller value) *)
```

10.4.1.7 MOD和MOD (

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

累加器除以一个或多个变量或常量，除法的剩余部分是累加器中的结果。对于括号模数，累加器除以括号中表达式的结果，并由此形成模数。最多可以有6个括号等级。

AWL示例:

```
LD 25 (* Load the dividend *)
MOD 20 (* Division 25/20 to Modulus = 5 *)
ST Var1 (* Save result 5 in Var1 *)

LD 25 (* Load the dividend *)
MOD( Var1 (* Result = 25/(Var1 + 10) to Modulus in the Accu *)
ADD 10
)
ST Var3 (* Save result 10 in Var3 *)
```

ST示例:

```
Result := 25 MOD 20; (* Save result 5 in Var1 *)
Result := 25 MOD (Var1 + 10); (* Result = 25/(Var1 + 10) per modulus into the Accumulator *)
```

10.4.1.8 MUL和MUL (

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

	BOOL	BYTE	INT	DINT
数据类型		X	X	X

累加器与一个或几个变量或常量的乘积。对于括号乘积，累加器乘以括号中表达式的结果。最多可以有6个括号等级。两个值必须都属于同一类型的变量。

AWL示例:

```
LD 25          (* Load the multiplier *)
MUL Var1, Var2 (* 25 * Var1 * Var2 *)
ST Var2       (* Save result *)

LD 25          (* Load the multiplier *)
MUL( Var1     (* Result = 25*(Var1 + Var2) *)
ADD Var2
)
ST Var3       (* Save result as variable Var3 *)
```

ST示例:

```
Result := 25 * Var1 * Var2;
Result := 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
可用性	X	X	X	X	X	X

	BOOL	BYTE	INT	DINT
数据类型		X	X	X

可以通过位于累加器命令前部的索引来选择各种常量或变量。第一个值可通过索引0进行访问。将所选的值加载到累加器中。值的数量仅受到程序存储器的限制。

AWL示例:

```
LD 1          (* Select the required element *)
MUX 10,20,30,40,Value1 (* MUX command with 4 constants and a variable *)
ST Value     (* Save result = 20 *)
```

ST示例:

```
Result := Mux(1, 10, 20, 30, 40, Value1) (* Save result = 20 *)
```

10.4.1.10 SUB和SUB (

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

	BOOL	BYTE	INT	DINT
数据类型		X	X	X

从一个或几个变量或常量中减去累加器。对于括号减法，从括号中表达式的结果中减去累加器。最多可以有6个括号等级。要减去的值必须属于同一类型的变量。

AWL示例:

```
LD 10
SUB Var1          (* Result = 10 - Var1 *)
ST Result

LD 20
SUB Var1, Var2, 30 (* Result = 20 - Var1 - Var2 - 30 *)
ST Result

LD 20
SUB( 6            (* Subtract 20 from the contents of the bracket *)
AND 2
)
ST Result          (* Result = 20 - (6 AND 2) *)
ST Result          (* Result = 18 *)
```

ST示例:

```
Result := 10 - Value1;
```

10.4.2 扩展数学运算符

i 信息

这里列出的运算符需要进行密集计算。这可能导致PLC程序相当长久的运行时间。

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
可用性	X	X	X	X		
	BOOL	BYTE	INT	DINT		
数据类型				X		

相关数学函数的计算。要计算的值必须以弧分为单位。缩放比例对应于1=1000。

换算：以弧度表示的角度 = (以度数表示的角度*PI/180) *1000

例如，90°的角度换算如下： 90° * 3.14 / 180) *1000 = 1571

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

AWL示例:

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

ST示例:

Result := COS(1234); (* Result = 330 *)
 Result := ACOS(330); (* Result = 1234 *)
 Result := SIN(1234); (* Result = 943 *)
 Result := ASIN(943); (* Result = 1231 *)
 Result := TAN(999); (* Result = 1553 *)
 Result := ATAN(1553); (* Result = 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
可用性	X	X	X	X		
	BOOL	BYTE	INT	DINT		
数据类型				X		

从累加器以欧拉数（2.718）为基础形成指数函数。小数点后最多3位数，例如，1.002必须输入为1002。

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

AWL示例:

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

ST示例:

```
Result := EXP(1000); (* Result = 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
可用性	X	X	X	X		
	BOOL	BYTE	INT	DINT		
数据类型				X		

以e（2.718）为底的对数。小数点后最多3位数，例如，1.000必须输入为1000。

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

AWL示例:

```
LD 1234
LN
ST Result
```

ST示例:

Result := LN(1234); (* Result = 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
可用性	X	X	X	X		
	BOOL	BYTE	INT	DINT		
数据类型				X		

从累加器形成以10为底的对数。小数点后最多3位数，例如，1.000必须输入为1000。

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

AWL示例:

LD 1234
LOG
ST Result (* Result = 91 *)

ST示例:

Result := LOG(1234); (* Result = 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
可用性	X	X	X	X		
	BOOL	BYTE	INT	DINT		
数据类型				X		

从累加器形成平方根。小数点后最多3位数，例如，1.000必须输入为1000。

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

AWL示例:

LD 1234
SQRT
ST Result (* Result = 1110 *)

ST示例:

Result := SQRT(1234); (* Result = 1110 *)

10.4.3 位运算符

10.4.3.1 AND和AND (

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型	X	X	X	X		

AE/累加器与一个或两个变量或常量的按位AND连接。与AE/累加器和先前在括号中形成的AE/累加器进行按位AND (...) 连接。最多可以有6个括号等级。所有值都必须属于同一类型的变量。

AWL中的示例:

```
LD 170
AND 204 (* AND link between 2 constants *)
(* Akku = 136 (see the example below the table) *)

LD 170 (* link between a constant and 2 variables.*)
AND Var1, Var2 (* Akku = 170dez AND Var1 AND Var2 *)

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

ST中的示例:

```
Result := 170 AND 204; (* Result = 136dec *)
```

Var2	Var1	结果
0	0	0
0	1	0
1	0	0
1	1	1

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

10.4.3.2 ANDN和ANDN (

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型	X	X	X	X		

AE/累加器与否定操作数的按位AND连接。AE/累加器与括号的否定操作数的按位AND (...) 连接。最多可以有6个括号等级。要连接的值必须属于同一类型的变量。

AWL示例:

```
LD 2#0000_1111
ANDN 2#0011_1010 (* ANDN link between 2 constants *)
(* Accu = 2#1111_0101 *)

LD 170
ANDN Var1, Var2 (* Link between a constant and 2 variables. *)
(* Accu = 170d ANDN Var1 ANDN Var2 *)

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

Var2	Var1	结果
0	0	1
0	1	1
1	0	1
1	1	0

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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型	X	X	X	X		

累加器的按位求反。

AWL示例:

```
LD BYTE#10 (* Load the value 10dec into the ACCU in Byte format *)
NOT (* The value is resolved on the Bit level (0000 1010), *)
(* Negated bit - wise (1111 0101) and then converted back *)
(* to a decimal value, result = 245dec *)
ST Var3 (* Save result as variable Var3 *)
```

ST示例:

```
Result := not BYTE#10; (* Result = 245dec *)
```

10.4.3.4 OR和OR (

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		

数据类型	X	X	X	X
------	---	---	---	---

AE/累加器与一个或两个变量或常量的按位OR连接。与AE/累加器和先前在括号中形成的AE/累加器进行按位OR (...) 连接。最多可以有6个括号等级。所有值都必须属于同一类型的变量。

AWL示例:

```
LD 170
OR 204      (* OR link between 2 constants *)

LD 170      (* Link between a constant and 2 variables. *)
OR Var1, Var2 (* Accu = 170d OR Var1OR Var2 *)

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

ST示例:

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

Var2	Var1	结果
0	0	0
0	1	1
1	0	1
1	1	1

10.4.3.5 ORN和ORN (

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型	X	X	X	X		

AE/累加器与否定操作数的按位OR连接。AE/累加器与括号的否定操作数的按位OR (...) 连接。最多可以有6个括号等级。要连接的值必须属于同一类型的变量。

AWL示例:

```
LD 2#0000_1111
ORN 2#0011_1010 (* ORN link between 2 constants *)
(* Accu = 2#1100_0000 *)

LD 170      (* Link between a constant and 2 variables. *)
ORN Var1, Var2 (* Accu = 170d ORN Var1 ORN Var2 *)

LD Var1
ORN ( Var2   (* AE/Accu = Var1 ORN ( Var2 OR Var3 ) *)
OR Var3
)
```

ST示例:

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

Var2	Var1	结果
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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

累加器向左按位旋转。累加器的内容向左移动n次，从而右侧再次插入左位。

AWL示例:

LD 175 (* Loads the value 1010_1111*)
 ROL 2 (* Accu content is rotated 2x to the left *)
 ST Value1 (* Saves the value 1011_1110 *)

ST示例:

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

10.4.3.7 ROR

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

累加器向右按位旋转。累加器的内容向右移动n次，从而左侧再次插入右位。

AWL示例:

LD 175 (* Loads the value 1010_1111 *)
 ROR 2 (* Accu content is rotated 2x to the right *)
 ST Value1 (* Saves the value 1110_1011 *)

ST示例:

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

10.4.3.8 S和R

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型	X					

如果前一个连接（AE）的结果为TRUE，则设置和复位布尔变量。

AWL示例:

LD TRUE (* Loads the AE with TRUE *)
 S Var1 (* VAR1 is set to TRUE *)
 R Var1 (* VAR1 is set to FALSE *)

ST示例:

Result := TRUE;
 Result := 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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

累加器的按位左移。累加器的内容向左移动n次，被推出的位丢失。

AWL示例:

LD 175 (* Loads the value 1010_1111 *)
 SHL 2 (* Accu content is shifted 2x to the left *)
 ST Value1 (* Saves the value 1011_1100 *)

ST示例:

Result := SHL(BYTE#175, 2); (* Result = 2#1011_1100 *)
 Result := SHL(INT#175, 2); (* Result = 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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

累加器的按位右移。累加器的内容向右移动n次，被推出的位丢失。

AWL示例:

```
LD 175 (* Loads the value 1010_1111 *)
SHR 2 (* Accu content is shifted 2 x to the right *)
ST Value1 (* Saves the value 0010_1011 *)
```

ST示例:

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

10.4.3.11 XOR和XOR (

	SK 54xE	SK 53xE SK 52xE	SK 2xxE	SK 2xxE-FDS	SK 180E SK 190E	SK 155E-FDS SK 175E-FDS
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型	X					

AE/累加器与一个或两个变量或常量的按位“异或”连接。第一个值位于AE/累加器中，第二个则用命令加载，或在括号内。要连接的值必须属于同一类型的变量。

AWL示例:

```
LD 2#0000_1111
XOR 2#0011_1010 (* XOR link between 2 constants *)
(* Accu = 2#0011_0101 *)

LD 170 (* Link between a constant and 2 variables. *)
XOR Var1, Var2 (* Accu = 170d XOR Var1 XOR Var2 *)

LD Var1
XOR ( Var2 (* AE/Accu = Var1 XOR ( Var2 OR Var3 *) *)
OR Var3
)
```

ST示例:

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

Var2	Var1	结果
0	0	0
0	1	1
1	0	1
1	1	0

10.4.3.12 XORN和XORN (

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

		SK 52xE				SK 190E	SK 175E-FDS
可用性	X	X	X	X	X	X	
	BOOL	BYTE	INT	DINT			
数据类型	X						

AE/累加器与否定操作数的按位异或连接。AE/累加器与括号的否定操作数的按位异或 (...) 连接。最多可以有6个括号等级。要连接的值必须属于同一类型的变量。

AWL示例:

```
LD 2#0000_1111
XORN 2#0011_1010 (* XORN link between 2 constants *)
(* Accu = 2#1100_1010 *)

LD 170
XORN Var1, Var2 (* Link between a constant and 2 variables. *)
(* Accu = 170d XORN Var1 XORN Var2 *)

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

ST示例:

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

Var2	Var1	结果
0	0	1
0	1	0
1	0	0
1	1	1

10.4.4 加载和存储运算符 (AWL)

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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型	X	X	X	X		

将一个常量或变量加载到AE或累加器中。

AWL示例:

```
LD 10 (* Loads 10 as BYTE *)
LD - 1000 (* Loads - 1000 as INTEGER *)
LD Value1 (* Loads variable Value 1 *)
```

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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型	X	X	X	X		

将否定的布尔变量加载到AE。

AWL示例:

LDN Value1 (* Value1 = TRUE à AE = FALSE *)
ST Value2 (* Speicher auf 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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型	X	X	X	X		

将AE/累加器的内容存储到变量。要保存的变量必须与先前加载和处理的数据类型相匹配。

AWL示例:

LD 100 (* Loads the value 1010_1111 *)
ST Value1 (* Accumulator content 100 is saved in 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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型	X	X	X	X		

将AE的内容存储到变量，并取反。要保存的变量必须与先前加载和处理的数据类型相匹配。

AWL示例:

LD Value1 (* Value1 = TRUE à AE = TRUE *)
STN Value2 (* Store to Value2 = FALSE *)

10.4.5 比较运算符

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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

将累加器的内容与一个变量或常量进行比较。如果值相等，则AE设置为TRUE。

AWL示例:

```
LD Value1 (* Value1 = 5 *)
EQ 10 (* AE = Is 5 equal to 10 ? *)
JMPC NextStep (* AE = FALSE - program does not jump *)
ADD 1
NextStep:
ST Value1
```

ST示例:

```
(* Ist 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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

将累加器的内容与一个变量或常量进行比较。如果累加器中的值大于或等于变量或常量，则AE设置为TRUE。

AWL示例:

```
LD Value1 (* Value1 = 5 *)
GE 10 (* Is 5 greater than or equal to 10? *)
JMPC NextStep (* AE = FALSE - program does not jump *)
ADD 1

NextStep:
ST Value1
```

ST示例:

```
(* Is 5 greater than or equal to 10? *)
if Value >= 10 then
    Value := Value - 1
end_if;
```

10.4.5.3 GT

	SK 54xE	SK 53xE	SK 2xxE	SK 2xxE-FDS	SK 180E	SK 155E-FDS
可用性	X	X	X	X	X	X

		SK 52xE				SK 190E	SK 175E-FDS
可用性	X	X	X	X	X	X	
	BOOL	BYTE	INT	DINT			
数据类型		X	X	X			

将累加器的内容与一个变量或常量进行比较。如果累加器中的值大于变量或常量，则AE设置为TRUE。

AWL示例:

```
LD Value1(* Value1 = 12 *)
GT 8 (* Is 12 greater than 8? *)
JMPC NextStep (* AE = TRUE - program jumps *)
ADD 1
NextStep:
ST Value1
```

ST示例:

```
(* Is 12 greater than 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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

将累加器的内容与一个变量或常量进行比较。如果累加器中的值小于或等于变量或常量，则AE设置为TRUE。

AWL示例:

```
LD Value1 (* Value1 = 5 *)
LE 10 (* Is 5 less than or equal to 10? *)
JMPC NextStep:
ST Value1
```

ST示例:

```
(* Is 5 less than or equal to 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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		

数据类型		X	X	X
------	--	---	---	---

将累加器的内容与一个变量或常量进行比较。如果累加器中的值小于变量或常量，则AE设置为TRUE。

AWL示例:

```
LD Value1 (* Value1 = 12 *)
LT 8 (* Is 12 less than 8? *)
JMPC NextStep (* AE = FALSE - program does not jump *)
ADD 1
NextStep:
ST Value1
```

ST示例:

```
(* Is 12 less than 8? *)
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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X	X	X		

将累加器的内容与一个变量或常量进行比较。如果累加器中的值不等于变量或常量，则AE设置为TRUE。

AWL示例:

```
LD Value1 (* Value1 = 5 *)
NE 10 (*Is 5 not equal to 10?*)
JMPC NextStep (* AE = TRUE - program jumps *)
ADD 1
NextStep:
ST Value1
```

ST示例:

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

10.5 处理值

所有的模拟和数字输入和输出或总线设定值和实际值可以由PLC读取和处理，或可以由PLC设置（如果它们是输出值）。可以通过下列过程值来访问单个值。对于所有输出值，必须对输出（例如数字输出或PLC设定值）进行编程，以使PLC为事件的来源。在每个周期开始时，由设备从PLC读取所有过程数据，并仅在程序结束时将其写入设备。下表列出了所有可以由PLC直接访问的值。所有其它的过程值必须通过功能块MC_ReadParameter或MC_WriteParameter进行访问。

10.5.1 输入和输出

这里列出了所有描述设备I/O界面的过程值。

名称	功能	标准化	型号	访问	设备
_0_Set_digital_output	设置数字输出	位0: Mfr1 位1: Mfr2 位2: Dout1 位3: Dout2 位4: dig. Fct. Aout 位5: Dout3 (Din7) 位6: Statusword Bit 8 位7: Statusword Bit 9 位8: BusIO Bit0 位9: BusIO Bit1 位10: BusIO Bit2 位11: BusIO Bit3 位12: BusIO Bit4 位13: BusIO Bit5 位14: BusIO Bit6 位15: BusIO Bit7	UINT	R/W	SK 54xE
_0_Set_digital_output	设置数字输出	位0: Mfr1 位1: Mfr2 位2: DOUT1 位3: DOUT2 位4: Dig. Analog Out 位5: free 位6: Bus PZD Bit 10 位7: Bus PZD Bit 13 位8: BusIO Bit0 位9: BusIO Bit1 位10: BusIO Bit2 位11: BusIO Bit3 位12: BusIO Bit4 位13: BusIO Bit5 位14: BusIO Bit6 位15: BusIO Bit7	UINT	R/W	SK 52xE SK 53xE
_0_Set_digital_output	设置数字输出	位0: DOUT1 位1: BusIO Bit0 位2: BusIO Bit1 位3: BusIO Bit2 位4: BusIO Bit3 位5: BusIO Bit4 位6: BusIO Bit5 位7: BusIO Bit6 位8: BusIO Bit7 位9: Bus PZD Bit 10 位10: Bus PZD Bit 13 位11: DOUT2	UINT	R/W	SK 2xxE
_0_Set_digital_output	设置数字输出	位0: DOUT1 位1: DOUT2	UINT	R/W	SK 180E SK 190E

名称	功能	标准化	型号	访问	设备
		位2: BusIO Bit0 位3: BusIO Bit1 位4: BusIO Bit2 位5: BusIO Bit3 位6: BusIO Bit4 位7: BusIO Bit5 位8: BusIO Bit6 位9: BusIO Bit7 位10: Bus PZD Bit 10 位11: Bus PZD Bit 13			SK 2xxE-FDS
_0_Set_digital_output	设置数字输出	位0: DOUT1 位1: DOUT2 位2: DOUT_BRAKE 位3: DOUT_BUS1 位4: DOUT_BUS2	UINT	R/W	SK 155E-FDS SK 175E-FDS
_1_Set_analog_output	设置模拟输出1。 IOE	10.0V = 100	BYTE	R/W	SK 54xE SK 53xE SK 52xE
_2_Set_external_analog_out1	设置模拟输出1。 IOE	10.0V = 100	BYTE	R/W	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_3_Set_external_analog_out2	设置模拟输出2。 IOE	10.0V = 100	BYTE	R/W	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_4_State_digital_output	数字输出状态	位0: Mfr1 位1: Mfr2 位2: Dout1 位3: Dout2 位4: dig. Fkt. Aout 位5: Dout3 (Din7) 位6: 总线状态字位8 位7: 状态字位9 位8: BusIO Bit0 位9: BusIO Bit1 位10: BusIO Bit2 位11: BusIO Bit3 位12: BusIO Bit4 位13: BusIO Bit5 位14: BusIO Bit6 位15: BusIO Bit7	INT	R	SK 54xE
_4_State_digital_output	数字输出状态	P711	BYTE	R	SK 52xE SK 53xE SK 2xxE

名称	功能	标准化	型号	访问	设备
					SK 2xxE-FDS SK 180E SK 190E
_4_State_digital_output	数字输出状态	位0: DOUT1 位1: DOUT2 位2: DOUT_BRAKE 位3: DOUT_BUS1 位4: DOUT_BUS2	BYTE	R	SK 155E-FDS SK 175E-FDS
_5_State_Digital_input	数字输入状态	位0: DIN1 位1: DIN2 位2: DIN3 位3: DIN4 位4: DIN5 位5: DIN6 位6: DIN7 位7: 数字功能AIN1 位8: 数字功能AIN2	INT	R	SK 54xE
_5_State_Digital_input	数字输入状态	位0: DIN1 位1: DIN2 位2: DIN3 位3: DIN4 位4: DIN5 位5: DIN6 位6: DIN7	INT	R	SK 52xE SK 53xE
_5_State_Digital_input	数字输入状态	位0: DIN1 位1: DIN2 位2: DIN3 位3: DIN4/AIN1 位4: AIN2 位5: PTC 位6: free 位7: free 位8: DIN1 IOE 1 位9: DIN2 IOE 1 位10: DIN3 IOE 1 位11: DIN4 IOE 1 位12: DIN1 IOE 2 位13: DIN2 IOE 2 位14: DIN3 IOE 2 位15: DIN4 IOE 2	INT	R	SK 2xxE SK 180E SK 190E
_5_State_Digital_input	数字输入状态	位0: DIN1 位1: DIN2 位2: DIN3 位3: TF (PTC) 位4: DIN-BUS1 (ASiI1) 位5: DIN-BUS2 (ASiI2) 位6: DIN-BUS3 (ASiI3)	INT	R	SK 155E-FDS SK 175E-FDS

名称	功能	标准化	型号	访问	设备
		位7: DIN-BUS4 (ASi4) 位8: STO 位9: BDD11 (ASIO3) 位10: BDD12 (ASIO4)			
_5_State_Digital_input	数字输入状态	位0: DIN1 位1: DIN2 位2: DIN3 位3: DIN4 位4: DIN5 位5: DIN6/AIN1 位6: DIN7/AIN2 位7: PTC 位8: DIN1 IOE 1 位9: DIN2 IOE 1 位10: DIN3 IOE 1 位11: DIN4 IOE 1 位12: DIN1 IOE 2 位13: DIN2 IOE 2 位14: DIN3 IOE 2 位15: DIN4 IOE 2	INT	R	SK 2xxE-FDS
_6_Delay_digital_inputs	根据P475的数字输入状态	位0: DIN1 位1: DIN2 位2: DIN3 位3: DIN4 位4: DIN5 位5: DIN6 位6: DIN7 位7: 数字功能AIN1 位8: 数字功能AIN2	INT	R	SK 54xE
_6_Delay_digital_inputs	根据P475的数字输入状态	位0: DIN1 位1: DIN2 位2: DIN3 位3: DIN4 位4: DIN5 位5: DIN6 位6: DIN7	INT	R	SK 52xE SK 53xE
_6_Delay_digital_inputs	根据P475的数字输入状态	位0: DIN1 位1: IN2 位2: DIN3 位3: AIN1 位4: AIN2 位5: PTC 位6: free 位7: free 位8: DIN1 IOE 1 位9: DIN2 IOE 1 位10: DIN3 IOE 1 位11: DIN4 IOE 1	INT	R	SK 2xxE SK 180E SK 190E

名称	功能	标准化	型号	访问	设备
		位12: DIN1 IOE 2 位13: DIN2 IOE 2 位14: DIN3 IOE 2 位15: DIN4 IOE 2			
_6_Delay_digital_inputs	根据P475的数字输入状态	位0: DIN1 位1: DIN2 位2: DIN3 位3: DIN4 位4: DIN5 位5: DIN6/AIN1 位6: DIN7/AIN2 位7: PTC 位8: DIN1 IOE 1 位9: DIN2 IOE 1 位10: DIN3 IOE 1 位11: DIN4 IOE 1 位12: DIN1 IOE 2 位13: DIN2 IOE 2 位14: DIN3 IOE 2 位15: DIN4 IOE 2	INT	R	SK 2xxE-FDS
_7_Analog_input1	模拟输入1的值 (AIN1)	10.00V = 1000	INT	R	全部
_8_Analog_input2	模拟输入2的值 (AIN2)	10.00V = 1000	INT	R	全部
_9_Analog_input3	模拟功能DIN2的值	10.00V = 1000	INT	R	SK 54xE SK 155E-FDS SK 175E-FDS
_10_Analog_input4	模拟功能DIN3的值	10.00V = 1000	INT	R	SK 54xE SK 155E-FDS SK 175E-FDS
_11_External_analog_input1	模拟输入的值1 (1.IOE)	10.00V = 1000	INT	R	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_12_External_analog_input2	模拟输入2的值 (1.IOE)	10.00V = 1000	INT	R	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_13_External_analog_input3	模拟输入1的值 (2.IOE)	10.00V = 1000	INT	R	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_14_External_analog_input4	模拟输入2的值 (2.IOE)	10.00V = 1000	INT	R	SK 54xE SK 2xxE

名称	功能	标准化	型号	访问	设备
					SK 2xxE-FDS SK 180E SK 190E
_15_State_analog_output	模拟输出的状态	10.0V = 100	BYTE	R	SK 54xE
_16_State_ext_analog_out1	模拟输出的状态 (1.IOE)	10.00V = 1000	INT	R	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_17_State_ext_analog_out2	模拟输出的状态 (2.IOE)	10.00V = 1000	INT	R	SK 54xE SK 2xxE SK 180E SK 190E
_18_Dip_switch_state	DIP开关状态	位0: DIP1 位1: DIP2 位2: DIP3 位3: DIP4 位4: DIP_I1 位5: DIP_I2 位6: DIP_I3 位7: DIP_I4	INT	R	SK 155E-FDS SK 175E-FDS

10.5.2 PLC设定值和实际值

这里列出的过程值形成PLC和设备之间的界面。PLC设定值的功能在（P553）中指定。

信息

过程值PLC_control_word覆盖功能块MC_Power。PLC设定值覆盖功能块MC_Move...和MC_Home。

名称	功能	标准化	型号	访问	设备
_20_PLC_control_word	PLC控制字	对应于USS配置文件	INT	R/W	全部
_21_PLC_set_val1	PLC设定值1	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_22_PLC_set_val2	PLC设定值2	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE

名称	功能	标准化	型号	访问	设备
					SK 2xxE-FDS SK 180E SK 190E
_23_PLC_set_val3	PLC设定值3	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_24_PLC_set_val4	PLC设定值4	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS
_25_PLC_set_val5	PLC设定值5	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS
_26_PLC_additional_control_word1	PLC附加控制字1	对应于USS配置文件	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_27_PLC_additional_control_word2	PLC附加控制字2	对应于USS配置文件	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_28_PLC_status_word	PLC状态字	对应于USS配置文件	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_29_PLC_act_val1	PLC实际值1	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E

名称	功能	标准化	型号	访问	设备
_30_PLC_act_val2	PLC实际值2	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_31_PLC_act_val3	PLC实际值3	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_32_PLC_act_val4	PLC实际值4	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS
_33_PLC_act_val5	PLC实际值5	100% = 4000h	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS
_34_PLC_Busmaster_Control_word	通过PLC的主机功能控制字（总线主机功能）	对应于USS配置文件	INT	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_35_PLC_32Bit_set_val1	32位PLC设定值 - P553[1] = 32位值的低位部分 - P553[2] = 32位值的高位部分	—	LONG	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_36_PLC_32Bit_act_val1	32位PLC实际值 - PLC实际值1 = 32位值的低位部分 - PLC实际值2 = 32位值的高位部分	—	LONG	R/W	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_37_PLC_status_bits	虚拟PLC状态输出	位0: PLC-DOUT1 位1: PLC-DOUT2	INT	R/W	SK 155E-FDS SK 175E-FDS
_38_PLC_control_bits	虚拟PLC控制	Bit 0: PLC-DIN1	INT	R/W	SK 155E-FDS

名称	功能	标准化	型号	访问	设备
	outputs	位1: PLC-DIN2 位2: PLC-DIN3 位3: PLC-DIN4 位4: PLC-DIN5 位5: PLC-DIN6 位6: PLC-DIN7 位7: PLC-DIN8			SK 175E-FDS

10.5.3 总线设定值和实际值

这些过程值显示了所有通过各种总线系统传输到设备的设定值和实际值。

名称	功能	标准化	型号	访问	设备
_40_Inverter_status	FI状态字	对应于USS配置文件	INT	R	全部
_41_Inverter_act_val1	FI实际值1	100% = 4000h	INT	R	全部
_42_Inverter_act_val2	FI实际值2	100% = 4000h	INT	R	全部
_43_Inverter_act_val3	FI实际值3	100% = 4000h	INT	R	全部
_44_Inverter_act_val4	FI实际值4	100% = 4000h	INT	R	SK 54xE
_45_Inverter_act_val5	FI实际值5	100% = 4000h	INT	R	SK 54xE
_46_Inverter_lead_val1	广播主机功能: 主机值1	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_47_Inverter_lead_val2	广播主机功能: 主机值2	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_48_Inverter_lead_val3	广播主机功能: 主机值3	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_49_Inverter_lead_val4	广播主机功能: 主机值4	100% = 4000h	INT	R	SK 54xE
_50_Inverter_lead_val5	广播主机	100% = 4000h	INT	R	SK 54xE

名称	功能	标准化	型号	访问	设备
	功能：主机值5				
_51_Inverter_control_word	产生的总线控制字	对应于USS配置文件	INT	R	SK 54xE SK 53xE SK 52xE SK 2xE SK 2xE-FDS SK 180E SK 190E
_52_Inverter_set_val1	产生的主总线设定值1	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xE SK 2xE-FDS SK 180E SK 190E
_53_Inverter_set_val2	产生的主总线设定值2	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xE SK 2xE-FDS SK 180E SK 190E
_54_Inverter_set_val3	产生的主总线设定值3	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xE SK 2xE-FDS SK 180E SK 190E
_55_Inverter_set_val4	产生的主总线设定值4	100% = 4000h	INT	R	SK 54xE
_56_Inverter_set_val5	产生的主总线设定值5	100% = 4000h	INT	R	SK 54xE
_57_Broadcast_set_val1	广播从机：辅助设定值1	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xE SK 2xE-FDS SK 180E SK 190E
_58_Broadcast_set_val2	广播从机：辅助设定值2	100% = 4000h	INT	R	SK 54xE SK 53xE SK 52xE SK 2xE SK 180E SK 190E
_59_Broadcast_set_val3	广播从机：辅助设定值3	100% = 4000h	INT	R	SK 54xE SK 53xE

名称	功能	标准化	型号	访问	设备
					SK 52xE SK 2xxE SK 180E SK 190E
_60_Broadcast_set_val4	广播从机：辅助设定值4	100% = 4000h	INT	R	SK 54xE
_61_Broadcast_set_val5	广播从机：辅助设定值5	100% = 4000h	INT	R	SK 54xE
_62_Inverter_32Bit_set_val1	产生的32位主设定值1总线	- P546[1]中的低位部分 - P546[2]中的高位部分	LONG	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_63_Inverter_32Bit_act_val1	FI32位实际值1	- P543[1]中的低位部分 - P543[2]中的高位部分	LONG	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_64_Inverter_32Bit_lead_val1	32位超前值1	- P502[1]中的低位部分 - P502[2]中的高位部分	LONG	R	SK 54xE SK 2xxE SK 180E SK 190E
_65_Broadcast_32Bit_set_val1	32位广播从机辅助设定值1	- P543[1]中的低位部分 - P543[2]中的高位部分	LONG	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_66_BusIO_input_bits	传入总线I/O数据	- 位0–7 = 在位0–7中的总线I/O - 位8 = 标志1 - 位9 = 标志2 - 位10 = 总线控制字的位8 - 位11 = 总线控制字的位9	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E

10.5.4 控制盒和参数盒

控制盒可以通过下列过程值进行访问。这可以执行简单的HMI应用程序。

名称	功能	标准化	型号	访问	设备
_70_Set_controlbox_show_val	显示控制盒的值	显示值 = 位29 – 位0	DINT	R/W	全部

		小数点位置 = 位31 - 位30			
_71_Controlbox_key_state	控制盒键盘状态	位0: ON 位1: OFF 位2: DIR 位3: UP 位4: DOWN 位5: Enter	BYTE	R	全部
_72_Parameterbox_key_state	参数盒键盘状态	位0: ON 位1: OFF 位2: DIR 位3: UP 位4: DOWN 位5: Enter 位6: Right 位7: Left	BYTE	R	全部

10.5.5 信息参数

这里列出的是设备最重要的实际值。

名称	功能	标准化	型号	访问	设备
_80_Current_fault	实际故障数	错误10.0 = 100	BYTE	R	全部
_81_Current_warning	实际警告	警告10.0 = 100	BYTE	R	全部
_82_Current_reason_FI_blocked	电源阻断状态的实际原因	问题10.0 = 100	BYTE	R	全部
_83_Input_voltage	实际电源电压	100 V = 100	INT	R	全部
_84_Current_frequenz	实际频率	10Hz = 100	INT	R	全部
_85_Current_set_point_frequency1	从设定值源的实际设定值频率	10Hz = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_86_Current_set_point_frequency2	实际变频器设定值频率	10Hz = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_87_Current_set_point_frequency3	斜坡后的实际设定值频率	10Hz = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E

名称	功能	标准化	型号	访问	设备
					SK 190E
_88_Current_Speed	计算的实时速度	100rpm = 100	INT	R	全部
_89_Actual_current	实际输出电流	10.0A = 100	INT	R	全部
_90_Actual_torque_current	实际扭矩电流	10.0A = 100	INT	R	全部
_91_Current_voltage	实际电压	100V = 100	INT	R	全部
_92_Dc_link_voltage	实际连接电路电压	100V = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_93_Actual_field_current	实际励磁电流	10.0A = 100	INT	R	全部
_94_Voltage_d	实际电压分量d-轴	100V = 100	INT	R	全部
_95_Voltage_q	实际电压分量q-轴	100V = 100	INT	R	全部
_96_Current_cos_phi	实际功率因数	0.80 = 80	BYTE	R	全部
_97_Torque	实际扭矩	100% = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_98_Field	实际励磁	100% = 100	BYTE	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_99_Apparent_power	实际视在功率	1.00KW = 100	INT	R	全部
_100_Mechanical_power	实际机械功率	1.00KW = 100	INT	R	全部
_101_Speed_encoder	实际测量速度	100rpm = 100	INT	R	SK 54xE SK 53xE SK 52xE
_102_Usage_rate_motor	实际电机使用率（瞬时值）	100% = 100	INT	R	全部
_103_Usage_rate_motor	实际电机使用率	100% = 100	INT	R	SK 54xE

名称	功能	标准化	型号	访问	设备
r_l2t	l2t				SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_104_Usage_rate_brake_resistor	实际制动电阻使用率	100% = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_105_Head_sink_temp	实际散热装置温度	100°C = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_106_Inside_temp	实际内部温度	100°C = 100	INT	R	SK 54xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_107_Motor_temp	实际电机温度	100°C = 100	INT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 2xxE-FDS SK 180E SK 190E
_108_Actual_net_frequency	实际净频率	10Hz = 100	INT	R	SK 155E-FDS SK 175E-FDS
_109_Mains_phase_sequence	电源相序	0=CW, 1=CCW	BYTE	R	SK 155E-FDS SK 175E-FDS
_141_Pos_Sensor_Inc	增量式编码器位置	0.001转	DINT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_142_Pos_Sensor_Abs	绝对值编码器位置	0.001转	DINT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_143_Pos_Sensor_Uni	通用编码器位置	0.001转	DINT	R	SK 54xE

名称	功能	标准化	型号	访问	设备
_144_Pos_Sensor_HTL	HTL编码器位置	0.001转	DINT	R	SK 54xE
_145_Actual_pos	实际位置	0.001转	DINT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_146_Actual_ref_pos	实际设定值位置	0.001转	DINT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E
_147_Actual_pos_diff	设定值和实际值之间的位置不同	0.001转	DINT	R	SK 54xE SK 53xE SK 52xE SK 2xxE SK 180E SK 190E

10.5.6 PLC错误

设备错误E23.0至E23.2可以通过用户错误标志从PLC程序进行设置。

名称	功能	标准化	型号	访问	设备
_110_ErrorFlags	在设备中产生用户错误	位0: E 23.0 位1: E 23.1 位2: E 23.2 位3: E 23.3 位4: E 23.4 位5: E 23.5 位6: E 23.6 位7: E 23.7	BYTE	R/W	全部
_111_ErrorFlags_ext	在设备中产生用户错误	位0: E 24.0 位1: E 24.1 位2: E 24.2 位3: E 24.3 位4: E 24.4 位5: E 24.5 位6: E 24.6 位7: E 24.7	BYTE	R/W	全部

10.5.7 PLC参数

PLC参数P355, P356和P360可以通过本组过程数据直接访问。

名称	功能	标准化	型号	访问	设备
_115_PLC_P355_1	PLC INT参数 P355 [-01]	-	INT	R	全部
_116_PLC_P355_2	PLC INT参数 P355 [-02]	-	INT	R	全部
_117_PLC_P355_3	PLC INT参数 P355 [-03]	-	INT	R	全部
_118_PLC_P355_4	PLC INT参数 P355 [-04]	-	INT	R	全部
_119_PLC_P355_5	PLC INT参数 P355 [-05]	-	INT	R	全部
_120_PLC_P355_6	PLC INT参数 P355 [-06]	-	INT	R	全部
_121_PLC_P355_7	PLC INT参数 P355 [-07]	-	INT	R	全部
_122_PLC_P355_8	PLC INT参数 P355 [-08]	-	INT	R	全部
_123_PLC_P355_9	PLC INT参数 P355 [-09]	-	INT	R	全部
_124_PLC_P355_10	PLC INT参数 P355 [-10]	-	INT	R	全部
_125_PLC_P356_1	PLC LONG参数 P356 [-01]	-	DINT	R	全部
_126_PLC_P356_2	PLC LONG参数 P356 [-02]	-	DINT	R	全部
_127_PLC_P356_3	PLC LONG参数 P356 [-03]	-	DINT	R	全部
_128_PLC_P356_4	PLC LONG参数 P356 [-04]	-	DINT	R	全部
_129_PLC_P356_5	PLC LONG参数 P356 [-05]	-	DINT	R	全部
_130_PLC_P360_1	PLC显示参数 P360[-01]	-	DINT	R/W	全部
_131_PLC_P360_2	PLC显示参数 P360[-02]	-	DINT	R/W	全部
_132_PLC_P360_3	PLC显示参数 P360[-03]	-	DINT	R/W	全部
_133_PLC_P360_4	PLC显示参数 P360[-04]	-	DINT	R/W	全部
_134_PLC_P360_5	PLC显示参数 P360[-05]	-	DINT	R/W	全部
_135_PLC_Scope_Int_1	PLC范围显示值1	-	INT	R/W	全部
_136_PLC_Scope_Int_	PLC范围显示值	-	INT	R/W	全部

名称	功能	标准化	型号	访问	设备
2	2				
_137_PLC_Scope_Int_3	PLC范围显示值3	-	INT	R/W	全部
_138_PLC_Scope_Int_4	PLC范围显示值4	-	INT	R/W	全部
_139_PLC_Scope_Bool_1	PLC范围显示值5	-	INT	R/W	全部
_140_PLC_Scope_Bool_2	PLC范围显示值6	-	INT	R/W	全部

10.6 语言

10.6.1 指令列表 (AWL/IL)

10.6.1.1 一般信息

数据类型

PLC支持下列数据类型。

名称	所需存储器空间	值范围
BOOL	1位	0至1
BYTE	1字节	0至255
INT	2字节	-32768至32767
DINT	4字节	-2147483648至2147483647
LABEL_ADDRES	2字节	跳转标记

文字

为了更加清晰，可以以各种显示格式为所有数据类型输入常量。下表概述了所有可能的版本。

文字	示例	以十进制显示的数字
Bool	FALSE	0
	TRUE	1
	BOOL#0	0
	BOOL#1	1
双数 (基2)	2#01011111	95
	2#0011_0011	51
	BYTE#2#00001111	15
	BYTE#2#0001_1111	31
八进制 (基8)	8#0571	377
	8#05_71	377
	BYTE#8#10	8
	BYTE#8#111	73
	BYTE#8#1_11	73
十六进制 (基16)	16#FFFF	-1
	16#0001_FFFF	131071
	INT#16#1000	4096
	DINT#16#0010_2030	1056816
整数 (基10)	10	10
	-10	-10
	10_000	10000
	INT#12	12
	DINT#-100000	-100000
时间	TIME#10s50ms	10.050秒
	T#5s500ms	5.5秒
	TIME#5.2s	5.2秒
	TIME#5D10H15M	5天+10小时+15分钟
	T#1D2H30M20S	1天+2小时+30分钟+20秒

注释

建议在程序的各部分提供注释，以便在以后可以理解PLC程序。在应用程序中，这些注释通过以“（*”开头和以“*”结尾的字符序列进行标记，如以下示例所示。

示例：

```
(* Comment about a program block *)  
LD 100 (* Comment after a command *)  
ADD 20
```

跳转标记

借助运算符JMP、JMPC或JMPCN，可以绕开程序的整个部分。跳转标记指定为目标地址。除了附加符和“β”，可以包含所有字母、数字0到9和下划线；不允许使用其它字符。跳转标记以冒号结束。这可能是独立的。跳转标记后的同一行内可能还会有其它命令。

可能出现的版本如下：

示例：

```
Jump mark:  
LD 20  
  
This_is_a_jumpmark:  
ADD 10  
  
MainLoop: LD 1000
```

另一个版本是将跳转标记转换为变量。该变量必须在变量表中定义为类型LABLE_ADDRESS，然后才能加载到变量“跳转标记”中。这样，可以非常简单地创建状态机，如下所示。

示例：

```
LD FirstTime  
JMPC AfterFirstTime  
(* The label address must be initialized at the beginning *)  
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  
  
Ende:
```

功能调用

编辑器支持一种形式的功能调用。在以下版本中，功能CTD通过实例I_CTD进行调用。结果保存在变量中。下面使用的功能的含义将在本使用手册的后面部分进行详细说明。

示例


```

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
    
```

按位访问变量

可以使用简化形式访问变量或过程变量中的位。

命令	描述
LD Var1.0	将Var1的位0加载到AE中
ST Var1.7	将AE存储到Var1位7上
EQ Var1.4	将AE与Var1位4进行比较

10.6.2 结构化文本 (ST)

结构化文本由一系列指令组成，这些指令以普通语言（“IF..THEN..ELSE）或以环路（WHILE..DO）的形式执行。

示例：

```

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

10.6.2.1 常用

数据类型 (ST)

PLC支持下列数据类型。

名称	所需存储器空间	值范围
BOOL	1位	0至1
BYTE	1字节	0至255
INT	2字节	-32768至32767
DINT	4字节	-2147483648至2147483647

赋值运算符

在赋值的左边有一个操作数（变量，地址），右边表达式的值用赋值运算符“=”赋值给操作数。

示例：

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

在执行该行后，Var1的值是Var2的十倍。

调用ST中的功能块

通过写入功能块实例的名称，然后分配括号中参数的值，在ST中调用功能块。在以下实例中，通过分配其参数IN和PT调用计时器。然后将结果变量Q分配给变量A。

在IL中使用功能块名称、下一个句点和变量名称来访问结果变量。

示例：

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

表达式求值

通过根据某些连接规则处理运算符来进行表达式的求值。首先处理具有最强连接的运算符，然后处理具有下一个最强连接的运算符，等等，直到处理完所有运算符。具有相同强度连接的运算符从左至右进行处理。

下表按连接强度的顺序显示ST运算符：

运算	符号	连接强度
括号	(表达式)	最强
函数调用	函数名称 (参数列表)	
否定补充组成	NOT	
乘 除 模数 AND	* / MOD AND	
加 减 OR XOR	+ - OR XOR	
比较 相等 不等	<,>,<=,>= = <>	弱

10.6.2.2 过程

Return

例如，根据条件，RETURN指令可用于跳转到程序的末尾。

IF

使用IF指令，可以测试条件，并根据该条件执行指令。

句法：

```
IF <Boolean_Expression1> THEN
  <IF_Instruction>
ELSIF <Boolean_Expression2> THEN
  <ELSIF_Instruction1>
ELSIF <Boolean_Expression n> THEN
  <ELSIF_Instruction n - 1>
ELSE
  <ELSE_Instruction>}
END_IF;
```

大括号{}中的部分为可选。如果<Boolean_Expression1>为TRUE，则仅执行<IF_Instructions>，不执行其它任何指令。否则，从<Boolean_Expression2>开始，按序列进行布尔表达式的求值，直到其中一个表达式为TRUE。然后，仅对该布尔表达式之后或在下一个ELSE或ELSIF之前的表达式进行求值。如果没有一个布尔表达式为TRUE，则仅对<ELSE_Instructions>进行求值。

示例：

```
IF temp < 17 THEN
  Bool1 := TRUE;
ELSE
  Bool2 := FALSE;
END_IF;
```

CASE

使用CASE指令，几个具有相同条件变量的条件指令可组合成一个指令。

句法：

```
CASE <Var1> OF
  <Value1>: <Instruction 1>
  <Value2>: <Instruction 2>
  <Value3, Value4, Value5>: <Instruction 3>
  <Value6 .. Value10> : <Instruction 4>
  ...
  <Value n>: <Instruction n>
ELSE <ELSE-Instruction>
END_CASE;
```

根据下列模式处理CASE指令：

- 如果<Var1>中的变量具有值<Value i>, 则执行指令<指令i>。
- 如果<Var 1>不具有任何额定的值, 则执行<ELSE指令>。
- 如果要对变量的几个值执行相同的指令, 则可以按序列分别写入这些值, 并用逗号作为公共指令的条件。
- 如果要对变量的一系列值执行相同的指令, 则可以写入起始值和结束值, 并用冒号作为公共指令的条件进行分隔。

示例:

```

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;
  
```

FOR循环

可以使用FOR循环对重复过程进行编程。

句法:

```

FOR <INT_Var> := <INIT_VALUE> TO <END_VALUE> {BY <STEP>} DO
  <Instruction>
END_FOR;
  
```

大括号{}中的部分为可选。只要计数器<INT-Var>不大于<END_VALUE>, 就执行<Instructions>。在执行<Instructions>之前进行检查, 以便在<INIT_VALUE>大于<END_VALUE>时从不执行<Instructions>。每当执行<Instructions>时, <INIT-Var>将增加<步长>。步长可以具有任何整数值。如果缺失, 则将其设置为1。<INT_Var>较大时, 必须终止循环。

示例:

```

FOR Zaehler :=1 TO 5 BY 1 DO
  Var1 := Var1 * 2;
END_FOR
  
```

REPEAT循环

REPEAT循环与WHILE循环的不同之处在于, 仅在执行环路之后才测试终止条件。因此, 无论终止条件如何, 循环必须至少运行一次。

句法:

```

REPEAT
  <Instruction>
UNTIL
  <Boolean Expression>
END_REPEAT;
  
```

执行<Instructions>, 直到<Boolean Expression>为TRUE。如果<Boolean Expression>在第一次求值时为TRUE, <Instructions>只执行了一次。如果<Boolean Expression>从未为TRUE, 则<Instructions>将被无休止地执行, 这将创建运行时间错误。

信息

程序员必须通过更改循环部分的条件, 例如向上或向下计数的计数器, 来确保不会产生无限循环。

示例:

```
REPEAT
  Var1 := Var1 * 2;
  Count := Count - 1;
UNTIL
  Count = 0
END_REPEAT
```

WHILE循环

WHILE循环可以以与FOR循环相同的方式使用, 不同之处在于终止条件可以是任何布尔表达式。这意味着表明了一个条件, 如果条件为TRUE, 则将导致循环的执行。

句法:

```
WHILE <Boolean Expression> DO
  <Instructions>
END_WHILE;
```

只要<Boolean_Expression>为TRUE, 就重复执行<Instructions>。如果<Boolean_Expression>在第一次求值时为FALSE, 则将从不执行<Instructions>。如果<Boolean_Expression>从未为FALSE, 则将无限重复<Instructions>。

信息

程序员必须通过更改循环部分的条件, 例如向上或向下计数的计数器, 来确保不会产生无限循环。

示例:

```
WHILE Count <> 0 DO
  Var1 := Var1*2;
  Count := Count - 1;
END_WHILE
```

Exit

如果在FOR、WHILE或REPEAT循环中发生EXIT指令, 则无论终止条件如何, 将终止最内层的环路。

10.7 跳转

10.7.1 JMP

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

		SK 52xE			SK 190E	SK 175E-FDS
可用性	X	X	X	X	X	X

无条件跳转到跳转标记。

AWL示例:

```
JMP NextStep (* Unconditional jump to 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
可用性	X	X	X	X	X	X

如果AE=TRUE，有条件跳转至跳转点，命令JMPC跳转至额定跳转点。

AWL示例:

```
LD 10
JMPC NextStep (* AE = TRUE - program jumps *)
ADD 1
```

```
NextStep:
ST Value1
```

10.7.3 JMPCN

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

有条件跳转至跳转点，如果AE注册=FALSE，则JMPCN跳转。否则程序继续进行下一个指令。

AWL示例:

```
LD 10
JMPCN NextStep (* AE = TRUE - program does not jump *)
ADD 1
```

```
NextStep:
ST Value1
```

10.8 型号转换

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

可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型	X					

将AE的数据类型从BOOL转换到BYTE。如果AE为FALSE，则将累加器转换为0。如果AE为TRUE，则将累加器转换为1。

AWL示例:

```
LD TRUE
BOOL_TO_BYTE (* AE = 1 *)
```

ST示例:

```
Result := BOOL_TO_BYTE(TRUE); (* Result = 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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X				

将数据类型从BYTE转换到BOOL。只要BYTE不等于零，则转换结果始终为TRUE。

AWL示例:

```
LD 10
BYTE_TO_BOOL (* AE = TRUE *)
```

ST示例:

```
Result := BYTE_TO_BOOL(10); (* Result = 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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型		X				

将数据类型从BYTE转换到INT。将BYTE复制到INT的低位分量中，INT的高位分量设置为0。

AWL示例:

```
LD 10
BYTE_TO_INT (* Akku = 10 *)
```

ST示例:

Result := BYTE_TO_INT(10); (* Result = 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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型				X		

将数据类型从DINT转换到INT。不传输DINT值的高位分量。

AWL示例:

```
LD 200000
DINT_TO_INT      (* Akku = 3392 *)

LD DINT# - 5000
DINT_TO_INT      (* Akku = - 5000 *)

LD DINT# - 50010
DINT_TO_INT      (* Akku = 15526 *)
```

ST示例:

Result := DINT_TO_INT(200000); (* Result = 3392 *)
 Result := DINT_TO_INT(- 5000); (* Result = - 5000 *)
 Result := DINT_TO_INT(- 50010); (* Result = 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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型			X			

将数据类型从INT转换到BYTE。这里，不传输INT值的高位分量。前缀丢失，因为BYTE类型不具有前缀。

AWL示例:

```
LD 16#5008
INT_TO_BYTE (* Akku = 8 *)
```

ST示例:

Result := INT_TO_BYTE(16#5008); (* Result = 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
可用性	X	X	X	X	X	X
	BOOL	BYTE	INT	DINT		
数据类型			X			

将数据类型从INT转换到DINT。将INT复制到DINT的低位分量中，DINT的高位分量设置为0。

AWL示例:

```
LD 10
INT_TO_DINT (* Akku = 10 *)
```

ST示例:

```
Result := INT_TO_DINT(10); (* Result = 10 *)
```

10.9 PLC故障信息

故障信息导致设备关闭，以防止设备故障。通过PLC故障信息，PLC停止执行，并且进入“PLC错误”状态。通过其它故障信息，PLC继续运行。确认错误后，PLC自动重新启动。

PLC通过PLC用户故障23.X继续运行!

SimpleBox		故障 参数盒中的文本	原因 处理方法
组	P700[-01] / P701详情		
E022	22.0	无PLC程序	已启动PLC，但设备中无PLC程序 - 将PLC程序加载到FI
	22.1	PLC程序故障	通过PLC程序进行的校验和检查产生了错误。 - 重新启动设备（打开电源），然后重试 - 或者，重新加载PLC程序
	22.2	跳转地址不正确	程序错误，行为与错误22.1相同
	22.3	堆栈溢出	程序运行期间，打开了6个以上的括号等级 - 检查程序的运行时间错误
	22.4	超过最大PLC周期	超过了PLC程序的额定最大周期时间 - 更改周期时间或检查程序

SimpleBox		故障 参数盒中的文本	原因 处理方法
组	P700[-01] / P701详情		
	22.5	未知命令代码	无法执行程序中的命令代码，因为代码未知。 - 程序错误，行为与错误22.1相同 - PLC版本和NORD CON版本不匹配
	22.6	PLC写入访问	PLC程序运行时，程序内容更改
	22.9	PLC错误	无法精确确定该故障的原因 - 行为与错误22.1相同
E023	23.0	PLC用户故障1	该错误可由PLC程序触发，以便在外部显示PLC程序执行中的问题。通过写入过程变量“错误标志”触发。
	23.1	PLC用户故障2	
	23.2	PLC用户故障3	

11 项目模式

11.1 一般信息

项目模式是普通模式的扩展。默认情况下，处于停用状态，必须在设置中激活。该模式允许用户管理项目。可以加载和保存项目。项目包括变频器及其数据（参数和PLC程序）、与外部参数文件或PLC程序的连接，以及应用程序的布局。通过NORD CON的新启动，始终加载已保存的最后一个项目。如果找不到项目，则创建新项目。为以下应用开发了项目模式：

- 11.2 “HMI”
- 11.3 “保存和恢复”

类别	名称	描述
文件	新项目	该操作创建一个空项目。
	打开项目...	该操作打开文件选择对话框，并且用户必须选择一个项目文件 (*.ncpx)。
	保存项目	该操作打开文件选择对话框，并且用户为项目文件 (*.ncpx) 指定名称。之后，项目保存在该名称下。
	保存全部	操作
项目	发送所有数据	该操作将所有参数和PLC程序发送到设备。
	读取所有数据	该操作从设备加载所有参数，并将参数保存到项目文件。此外，设备中的PLC程序与项目中的PLC程序进行比较。如果两者不同，则在日志中显示警告。
	去除参数	该操作从项目删除所选设备的参数。
	去除PLC程序	该操作从项目删除所选设备的PLC程序。
	添加PLC程序	该操作为突出显示的设备添加保存的PLC程序。
	导出参数	该操作将所选设备的参数导出到一个文件。
	导出PLC程序	该操作将所选设备的PLC程序导出到一个文件。
PLC	保存	该操作将PLC保存在项目文件中。
	另存为...	该操作打开文件选择对话框，并且用户必须选择一个文件名称。此后，PLC程序保存在一个单独文件中。
参数设置	保存	该操作将参数保存在项目文件中。
	另存为...	该操作打开文件选择对话框，并且用户必须选择一个文件

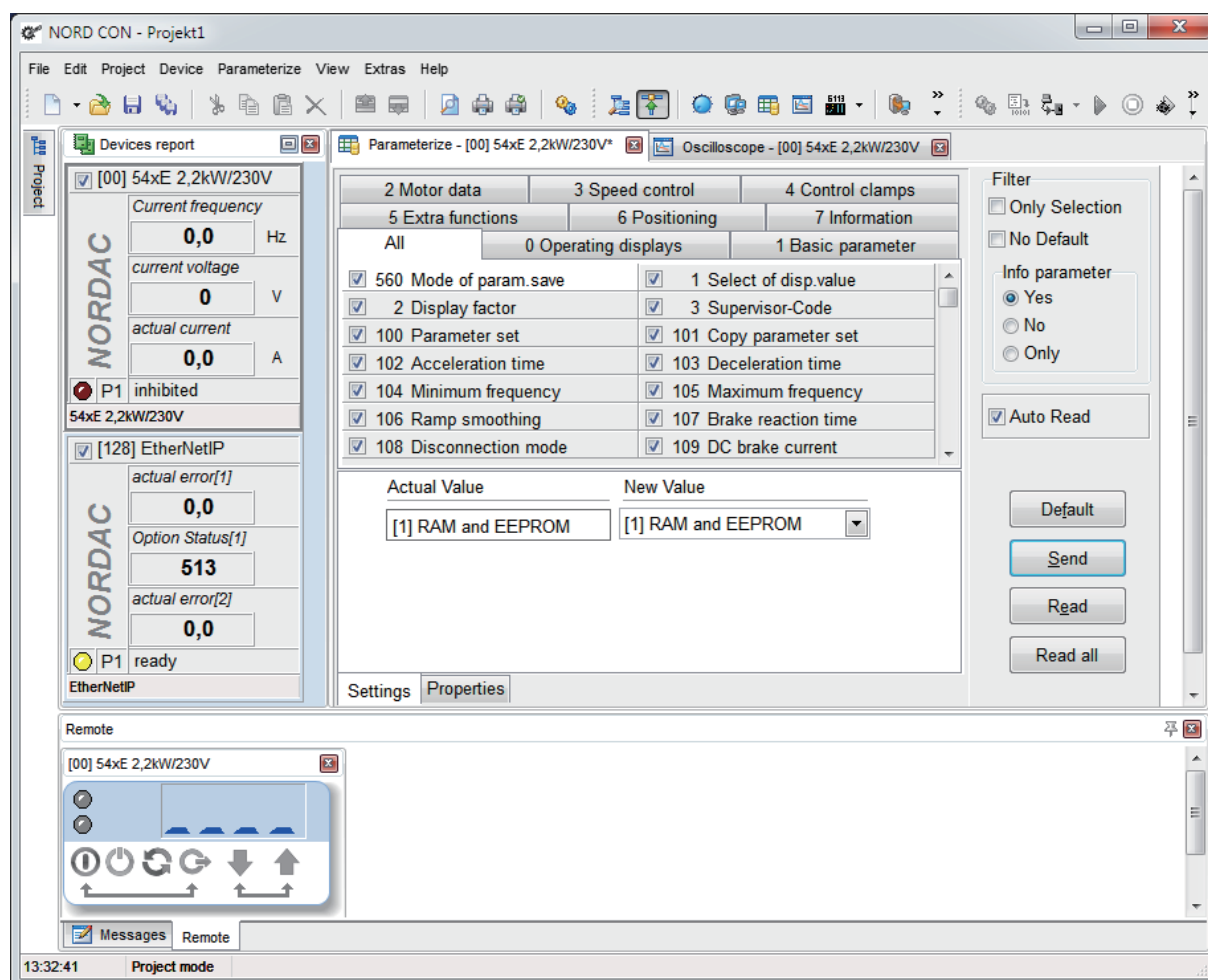
类别	名称	描述
		名称。此后，参数保存在一个单独文件中。

11.2 HMI

该项目模式非常适合良好的可视化效果。用户将PC连接到系统并启动设备搜索（总线扫描Ctrl. F5）。设备搜索完成后，用户可以将设备所需的显示元件放置在工作区域，例如参数窗口，示波器或控制窗口。此后，必须保存该项目。打开项目后，将恢复设备和布局。这使用户可以始终使用相同的用户界面。

信息

加载项目时，不会检查项目中包含的设备是否实际连接。如果其它设备连接到总线，则可能导致通信错误。如果使用系统总线，请注意始终使用同一设备进行通信连接。



11.3 保存和恢复

该项目模式可用于保存和恢复参数及PLC程序。在设备搜索（总线扫描）之后，可以进一步限制所使用的设备列表。通过在设备概述中取消激活设备，可以将设备从保存和恢复中排除。

该过程可能需要几分钟，因为取决于特定的系统，设备列表可能包含几个设备。操作进度显示在一个单独的窗口中。在此过程中不能使用NORD CON。

信息

加载项目时，不会检查项目中包含的设备是否实际连接。如果其它设备连接到总线，则可能导致通信错误。如果使用系统总线，请注意始终使用同一设备进行通信连接。

保存

设备搜索（总线扫描）后，“读取所有数据”操作将读取找到的所有设备的参数。必须首先将参数保存在NORD CON中，然后手动保存在项目文件（保存全部）中。用户可以使用三个选项来执行“读取所有数据”操作。这些选项可以在设置对话框中激活或取消激活。

选项	描述
取消时删除所有数据记录	如果激活该选项，当取消“读取所有数据”功能时，将删除项目中包含的所有设备的数据记录。否则，并非所有参数都会被读取，并且项目文件中的数据不完整。
删除不完整的数据记录	如果启用该选项，则在“读取所有数据”功能期间，若发生错误，将删除设备的数据记录。
删除设备中不能用于通信的数据记录	如果启用该选项，则在执行“读取所有数据”功能期间，若设备无通信，将删除设备的数据记录。

在当前NORD CON版本中无法读取PLC程序。因此，在“读取所有数据”操作期间，设备的程序与项目文件进行比较。如果两者不同，则在NORD CON中给出警告。如果没有为设备保存PLC程序，则跳过该操作。

如果设备参数保存在项目文件中，则在项目结构中以特殊的设备符号表示。同样适用于PLC程序。但是，设备符号的存在并不提供关于数据的当前状态和完整性的任何信息。读取后，可以通过参数编辑器编辑参数。用户在项目树中选择设备，并且打开参数编辑器（参数化F7）。可以在编辑器中读取或删除参数。“保存”操作将所选设备的参数保存在项目中，并且还将项目保存在硬盘驱动器上。如果要将参数保存在单独的文件中，则必须执行“另存为”操作。

信息

如果在“读取所有数据”期间发生错误，则会在日志中记录这些错误，并继续进行备份。日志中记录的所有参数不会保存在项目文件中。我们建议纠正故障，并且为设备执行新的备份。

恢复

可以通过主菜单打开项目后执行该功能。为此，将保存在项目文件中的参数发送到设备。作为标准，始终将所有参数发送到设备。但是，在大多数情况下，不建议这样操作，只会占用时间。要减少参数数量，用户必须启用“仅传输启用参数”选项，并且在参数编辑器中启用所需的参数。

第二步，加载、翻译保存在项目中的PLC程序，并将其发送到设备。与正常模式一样，设备的PLC程序通过PLC编辑器进行编辑。打开编辑器时，PLC程序将自动从项目文件加载。编辑后，程序可以通过“保存”操作再次保存在项目文件中。如果要将PLC程序保存在该模式一个单独的文件中，则必须执行“另存为”操作。

 信息

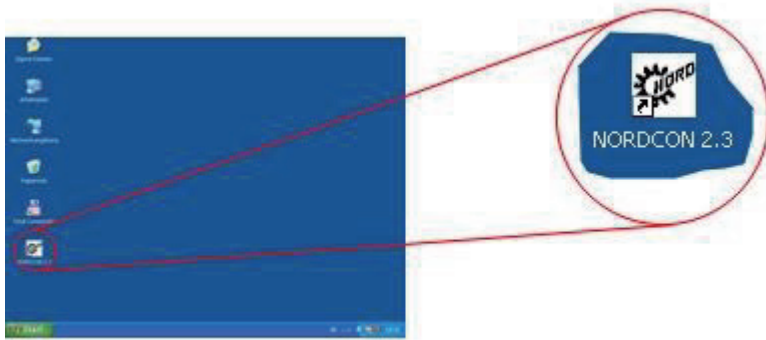
如果在该过程中出现错误，则会在日志中记录并继续该过程。日志中记录的所有参数不会保存在设备中。这同样适用于PLC程序。我们建议纠正故障并重新启动操作。

12 固件

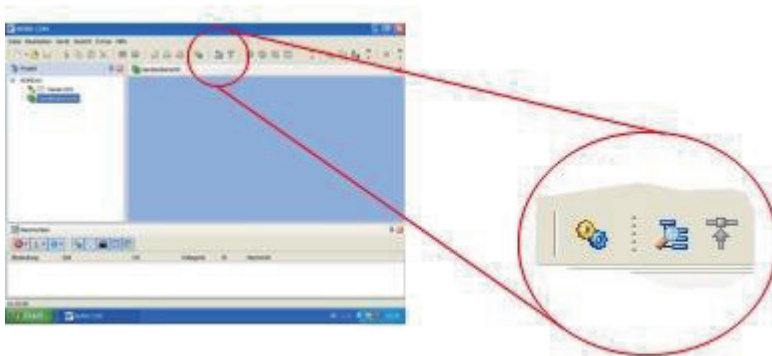
12.1 如何更新固件

必须执行以下输入步骤：

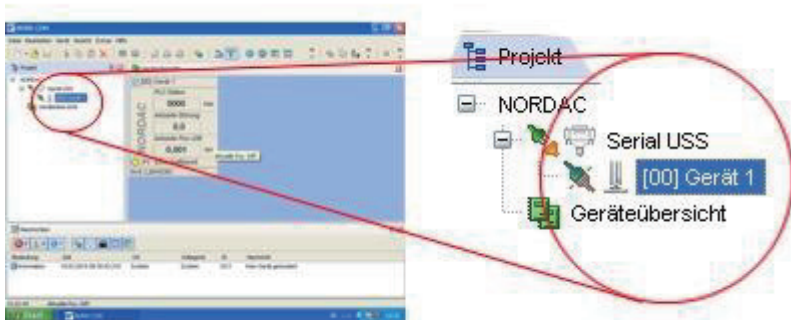
1.启动NORD CON



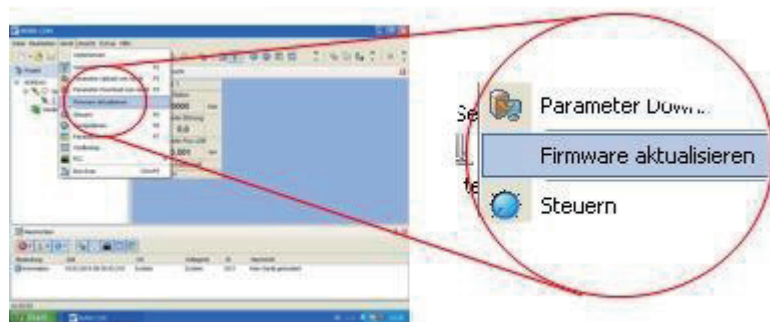
2.进行设备搜索



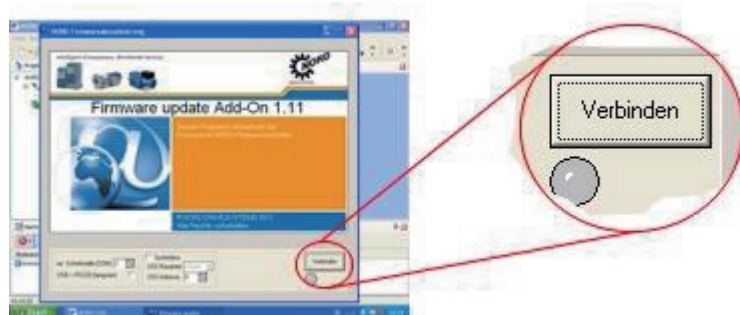
3.突出显示项目树中的所需设备



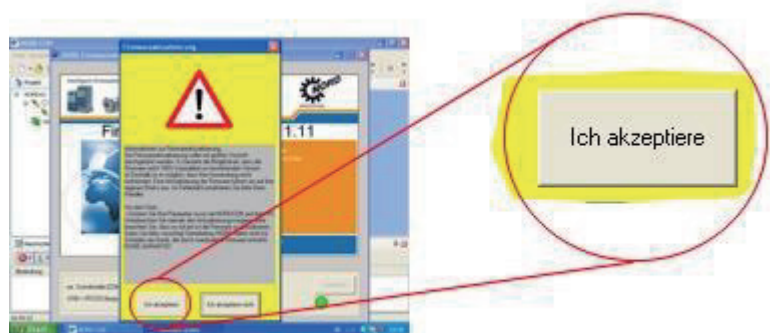
4.通过“设备->更新固件”菜单项启动固件更新程序



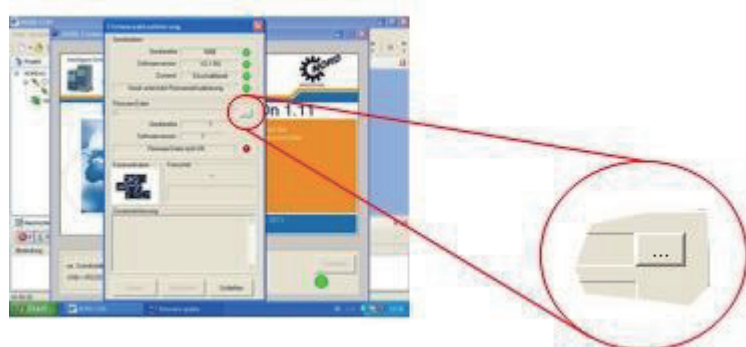
5.单击连接



6.仔细阅读预警信息，并通过“我同意”按钮进行确认



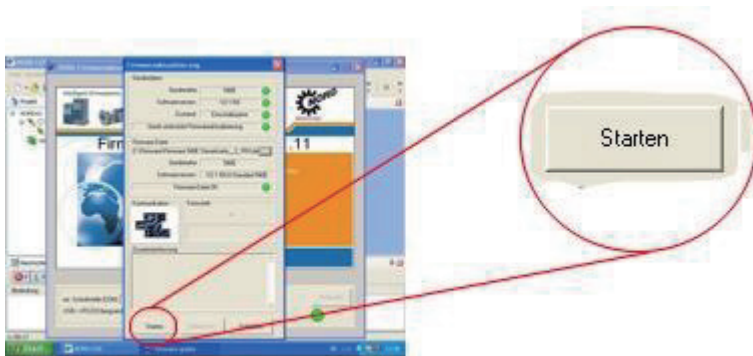
7.借助“...”按钮选择固件文件



8.选择固件文件，并通过“打开”进行确认



9.通过“启动”按钮启动固件传输



i 信息

只有当设备的地址为0并且波特率设置为38400位/秒时，才能更新固件。

i 信息

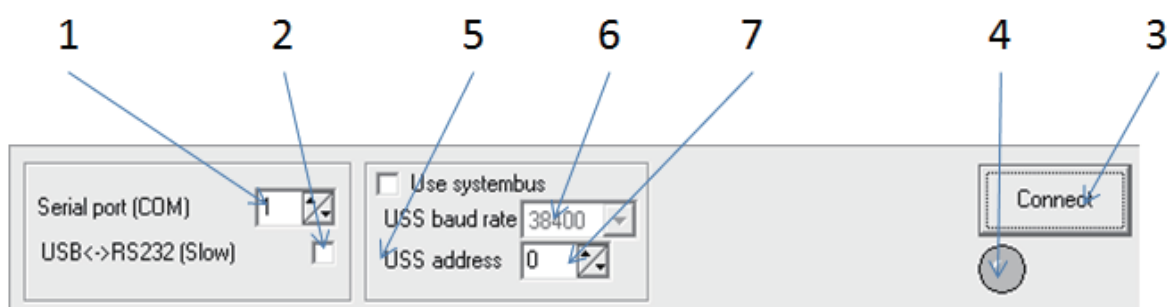
如果固件传输中断或未正确执行，请重新启动设备。如果随后在总线扫描中找不到设备，也可以手动启动固件更新程序（FirmwareUpd.exe）。程序位于主目录NORD CON。

12.2 固件更新程序

1. 设置

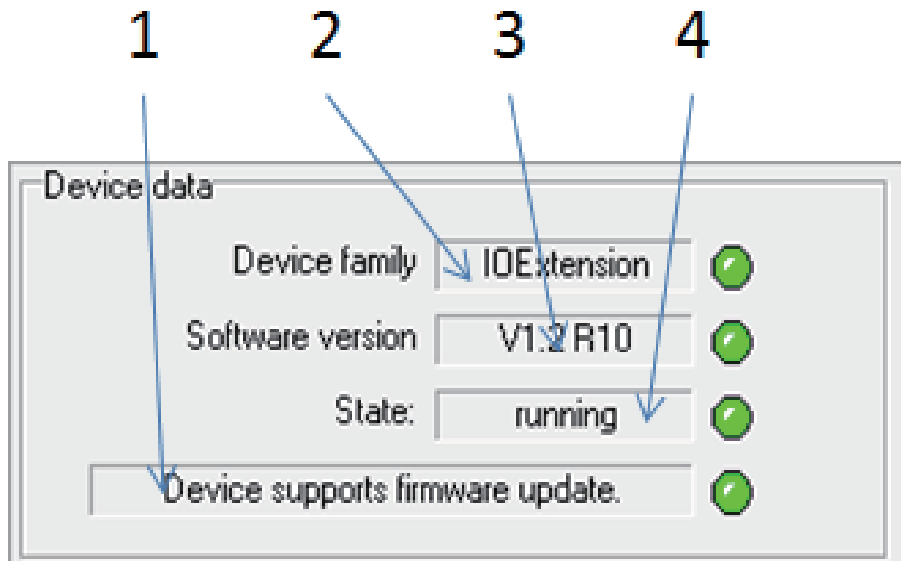
编号	描述
1	在选择框中，用户指定与变频器连接的PC的COM端口。如果通过NORD CON调用程序，则无需设置该参数。
2	对于一些USB到RS232转换器，该设置可以启用更稳定的通信。只有在连接有问题时才能选择该设置。
3	“连接”按钮创建到所连接设备的连接。如果找到设备，则LED（4）

编号	描述
	呈绿色，并且固件下载窗口打开。
4	LED指示连接状态。 灰色 - 尚未建立连接。 绿色 - 程序连接到设备。 红色 - 未找到设备。
5	该选项通过系统总线激活固件更新。
6	该选择框指定PC与所连接设备之间的传输速度。
7	该选择框指定设备的映射USS地址。



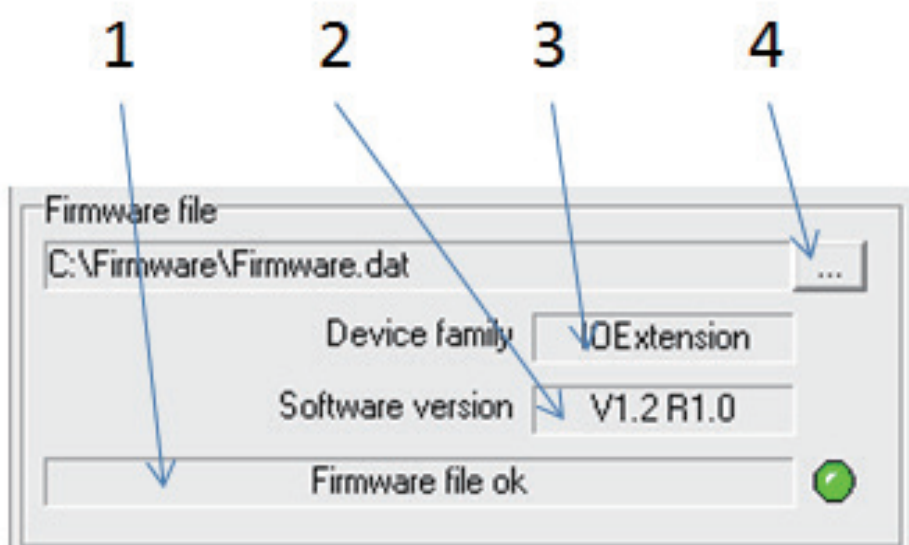
2. 变频器数据

编号	描述
1	该字段指示所连接的变频器是否支持固件更新。如果不支持，则字段旁边的LED亮起红色。
2	该字段显示所连接设备的变频器系列。如果无法检测到设备，则字段旁边的LED为红色，无法进行固件更新。
3	该字段显示所连接变频器的版本号。
4	该字段显示所连接变频器的状态。如果启用变频器，则字段旁边的LED为红色，无法进行固件更新。



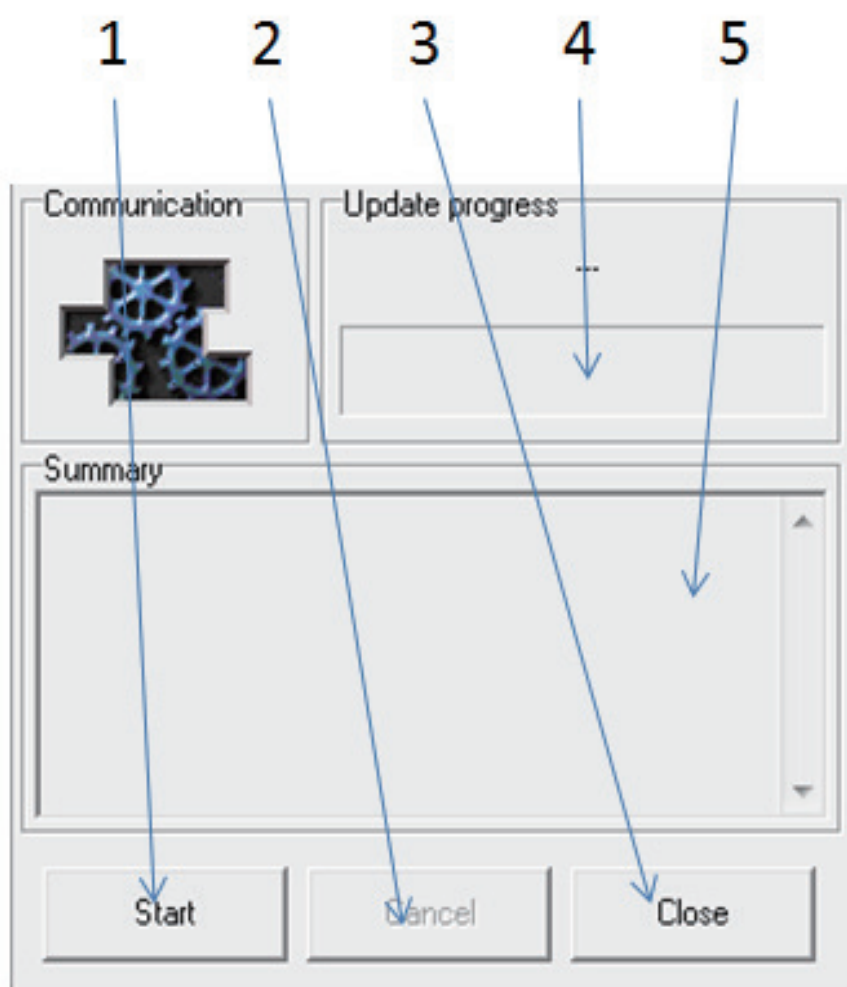
3. 选择固件更新文件

编号	描述
1	该字段显示当前加载固件的状态。如果无法加载固件文件或固件与所连接设备不匹配，则字段旁边的LED为红色。无法进行固件更新。
2	该字段显示当前加载固件的版本信息。
3	该字段显示当前加载固件支持的变频器系列。
4	单击“...”按钮打开文件选择对话框。用户可以在窗口中选择固件文件。通过单击“打开”采用该文件，并且将其保存在程序配置文件中。



4. 更新固件

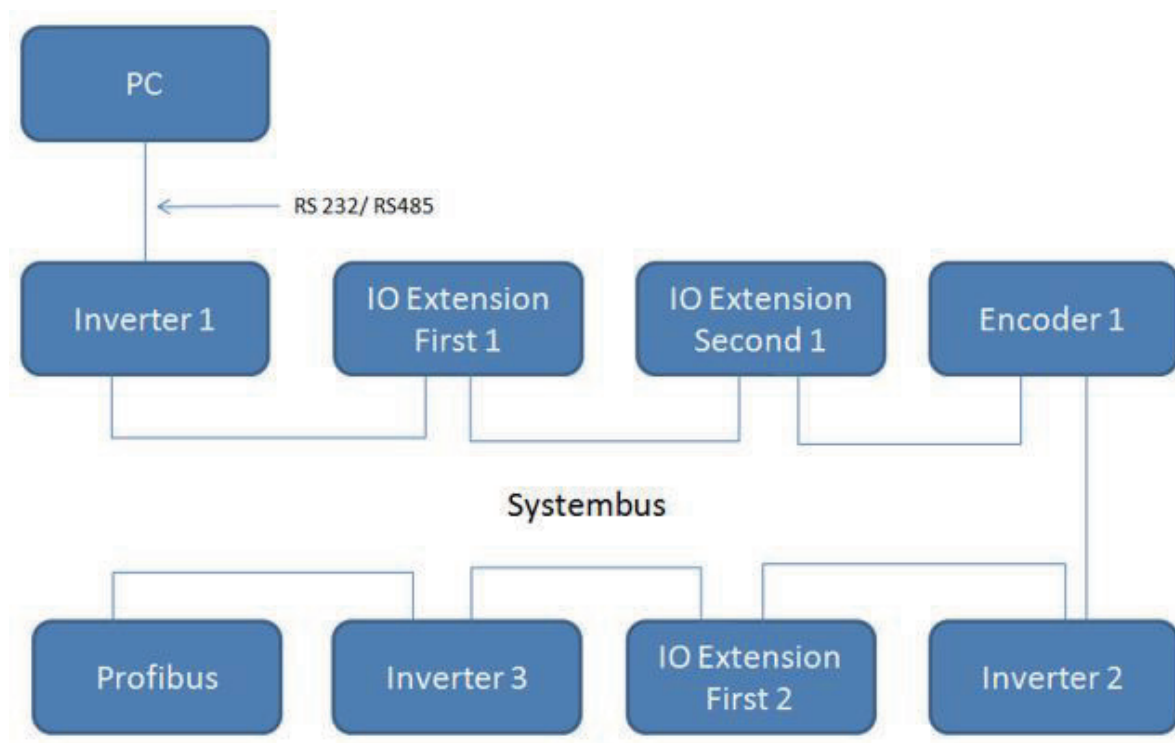
编号	描述
1	必须按下“启动”按钮以启动固件更新。如果未激活该按钮，则无法将所选固件加载到设备。
2	已启动的更新可通过按下“取消”按钮进行取消。仅在初始阶段才能取消。
3	更新期间不能关闭下载窗口。下载之前或之后，用户可以通过按下“关闭”按钮取消更新。
4	进度显示表明了更新进度和当前状态。
5	更新后，在“总结”字段显示结果。



12.3 通过系统总线进行固件更新

系统总线是诺德公司在CAN总线的基础上开发的一种总线。该总线适用于所有带有内部CAN接口的SK2xxE和SK5xxE变频器，以及各种附加模块。多达四个变频器，每个具有2个附加模块和一个CANopen编码器，以及可以同时连接总线模块，从而在最大配置中将17个设备连接到系统总线。用于

系统总线的协议对应于CANopen。各个变频器的CAN地址在系统总线中有固定的分配，并且无法更改。



所有连接到系统总线的诺德模块可以通过使用NORDCON的RS232/RS485接口的参与者进行可视化和参数化。为此，通信查询可以通过NORD CON或连接到固件更新程序的设备进行通道传输。以下映射过程用于通道传输查询。

USS地址	模块
0	必须在连接到NORD CON的模块中设置地址。
1	变频器1 (CAN ID: 32)
2	变频器2 (CAN ID: 34)
3	变频器3 (CAN ID: 36)
4	变频器4 (CAN ID: 38)
10	变频器1的附加模块1 (I/O扩展)
11	变频器2的附加模块1 (I/O扩展)
12	变频器3的附加模块1 (I/O扩展)
13	变频器4的附加模块1 (I/O扩展)
19	固件更新中止的变频器
20	变频器1的附加模块2 (I/O扩展)
21	变频器2的附加模块2 (I/O扩展)

USS地址	模块
22	变频器3的附加模块2 (I/O扩展)
23	变频器4的附加模块2 (I/O扩展)
30	总线模块

以下设备支持固件更新的通道传输：

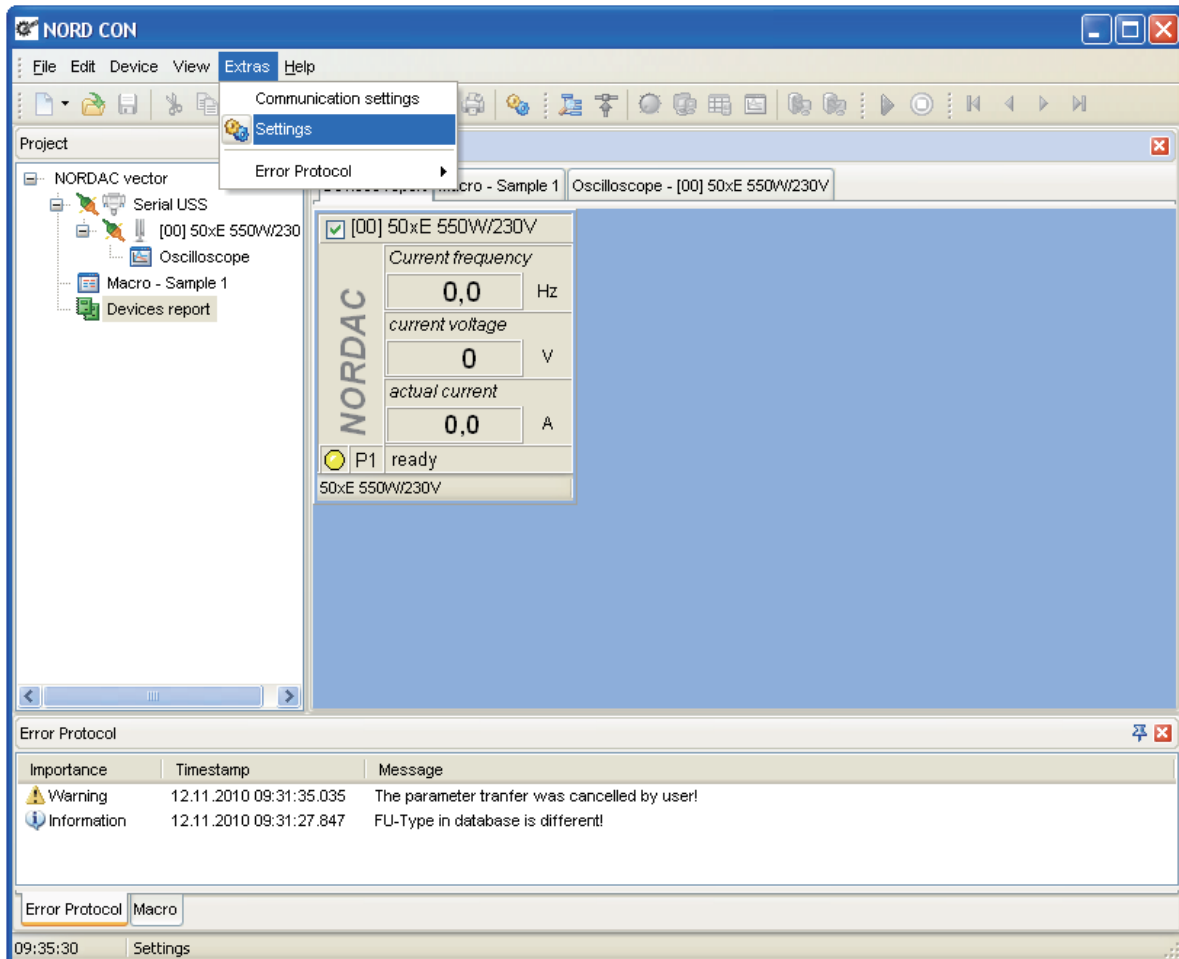
设备	版本
SK 1xxE	全部
SK 2xxE	V1.3及以上
SK 5xxE	V2.0及以上
SK 540E	V2.0及以上
SK TU4-DEV	V1.4及以上
SK TU4-CAO	V2.2及以上
SK TU4-PBR	V1.2及以上
SK TU4-POL	全部
SK TU4-PNT	全部
SK TU4-IOE	V1.2及以上
SK TU4-EIP	全部

以下设备支持通过系统总线进行固件更新：

设备	版本
SK 1xxE	全部
SK 540E	V2.0及以上
SK TU4-DEV, SK CU4-DEV	V1.4及以上
SK TU4-CAO, SK CU4-CAO	V2.2及以上
SK TU4-PBR, SK CU4-PBR	V1.2及以上
SK TU4-POL, SK CU4-POL	全部
SK TU4-PNT, SK CU4-PNT	全部
SK TU4-IOE, SK CU4-IOE	V1.2及以上
SK TU4-EIP, SK CU4-EIP	全部

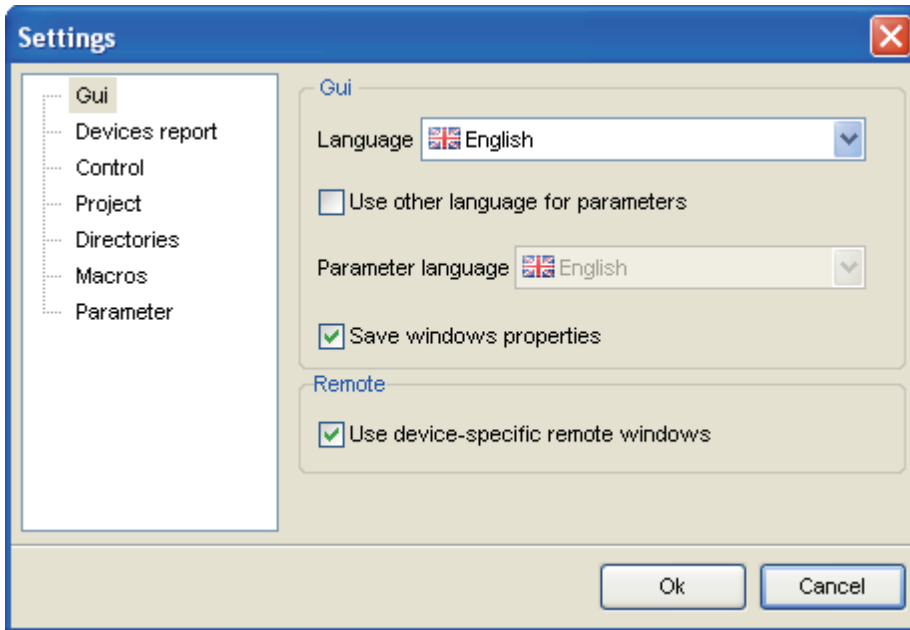
13 设置

用户可以使用“附加->设置”菜单项来更改当前程序设置。分为以下几类：



13.1 用户界面

该类别中，用户可以更改用户界面的设置。可以使用以下选项：



语言

用户可以通过该选项更改界面的语言。

使用参数设置的其它语言

通过选择该选项，用户可以在“参数语言”选择框的“参数化”对话框中为参数名称选择不同的语言。

参数语言

通过选择该选项，用户可以在“参数化”对话框中为参数名称选择不同的语言。可以通过“使用参数设置的其它语言”选项来激活该选择。

保存窗口设置

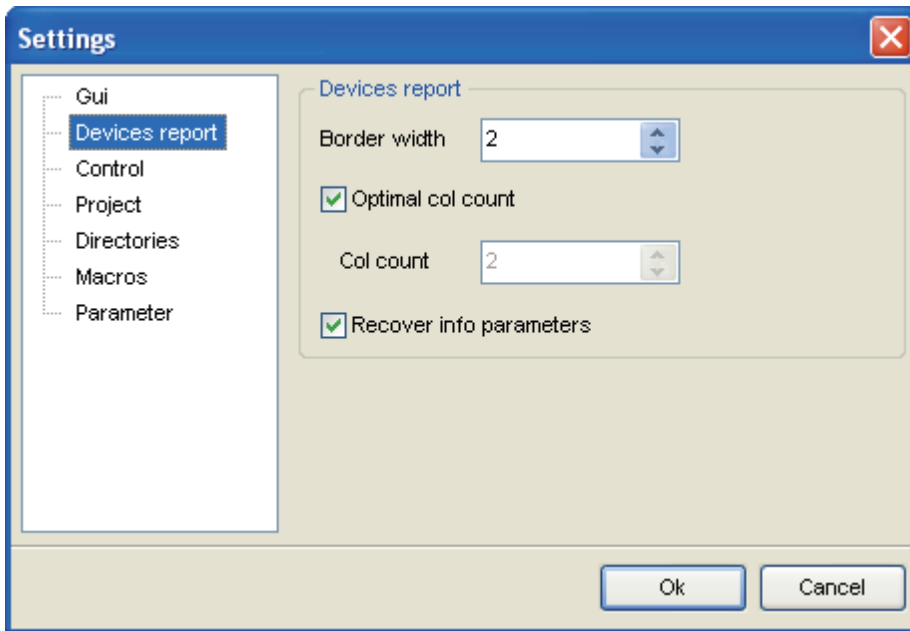
通过激活该选项，存储并在再次打开后重新激活窗口设置，例如位置和大小。

使用设备专用远程窗口

如果激活该选项，则为每种型号的设备生成特殊的远程窗口。否则使用标准窗口。

13.2 设备报告

该类别中，用户可以更改“设备概述”窗口中的设置。



边框宽度


通过参数，用户可以更改设备显示的边框宽度。值可以设置在0到10像素之间。如果注册的值较大或较小，则自动使用最大值或最小值。

最佳列数

如果选择该选项，则应用程序将计算最佳列数。

列数

通过该参数，用户可以指定固定列数。值可以设置在1到10之间。如果注册的值较大或较小，则自动使用最大值或最小值。

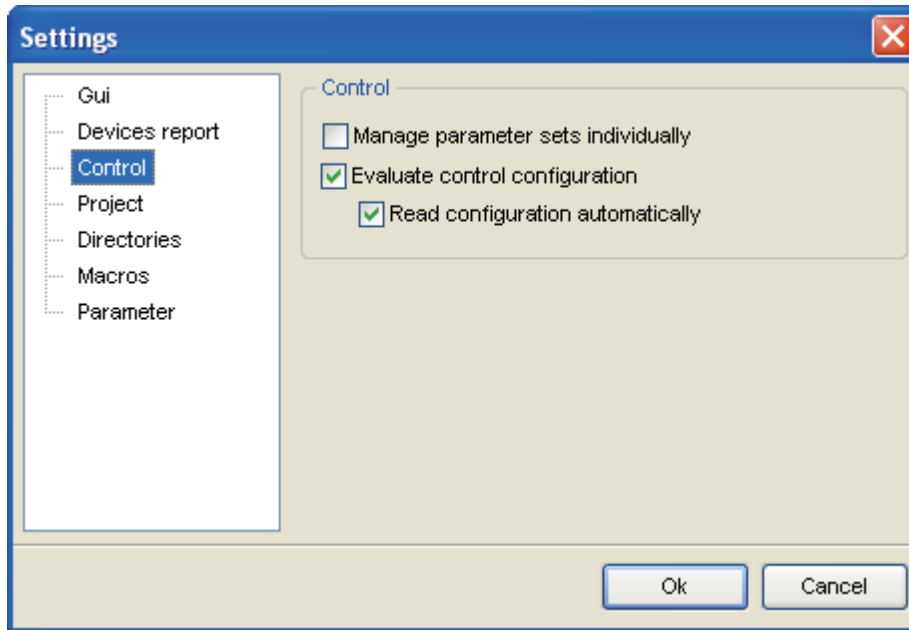
<p>注意</p> 	<p>仅在未选择“最佳列数”选项时才能更改该参数。</p>
-----------------------------------------------------------------------------------------------	-------------------------------

恢复信息参数

如果选择该选项，则调整后的设备信息参数通过总线扫描或重新启动应用程序进行存储和恢复。

13.3 控制

在类别5“控制”中，用户可以更改“控制”窗口的设置。



单独管理参数集


通过激活该选项，可以在“控制”窗口中分别管理设置值和实际值。

评估控制配置

该选项激活或停用控制配置。通过激活该功能，检查配置后将释放或阻断一些功能。此外，参数化的设置值功能或实际值功能的名称以清楚的文本形式显示在窗口中。

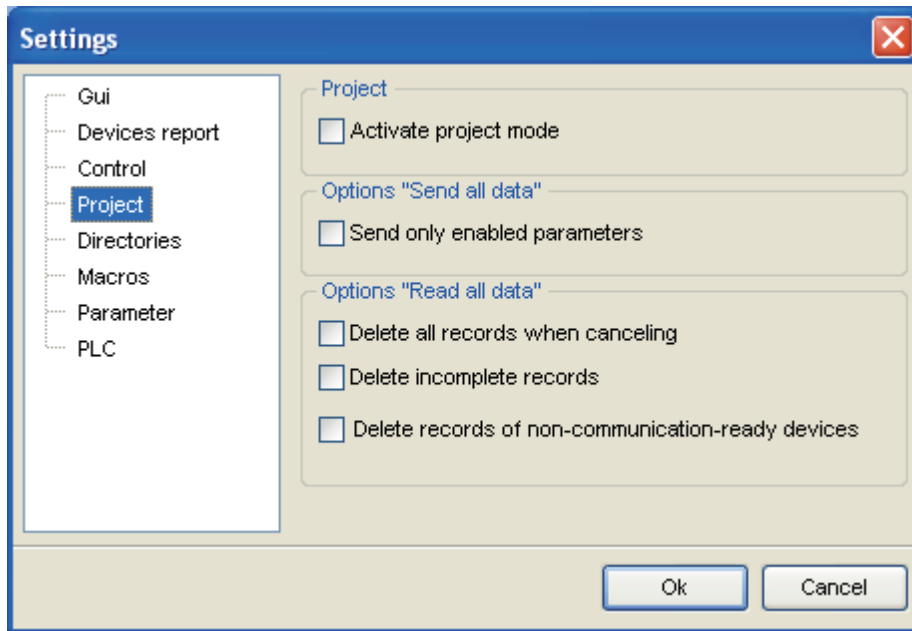
自动读取配置

该选项激活或停用配置的自动检查。通过激活该功能，集中窗口后将再次检查控制配置。

<p>说明</p> 	<p>所有设备都无法使用“评估控制配置”功能！</p>
-----------------------------------------------------------------------------------------------	-----------------------------

13.4 项目

在该标题中，用户可以指定项目文件的路径。设置，例如所使用的界面、总线扫描设置、设备名称等等，保存在该文件中。可通过选择一个现有文件重新加载旧设置。



激活项目模式

可以通过该选项启用或禁用项目模式。在项目模式下，用户可以自由地对总线上设备的类型和数量进行参数化。设备参数和应用程序设置保存在项目文件中。

仅传输启用参数

如果启用该选项，则通过“发送所有数据”功能，仅将已被用户启用的参数发送到设备。作为标准，始终启用所有参数。可以在参数编辑器中更改参数的启用情况。

取消时删除所有数据记录

如果激活该选项，当取消“读取所有数据”功能时，将删除项目中包含的所有设备的数据记录。

删除不完整的数据记录

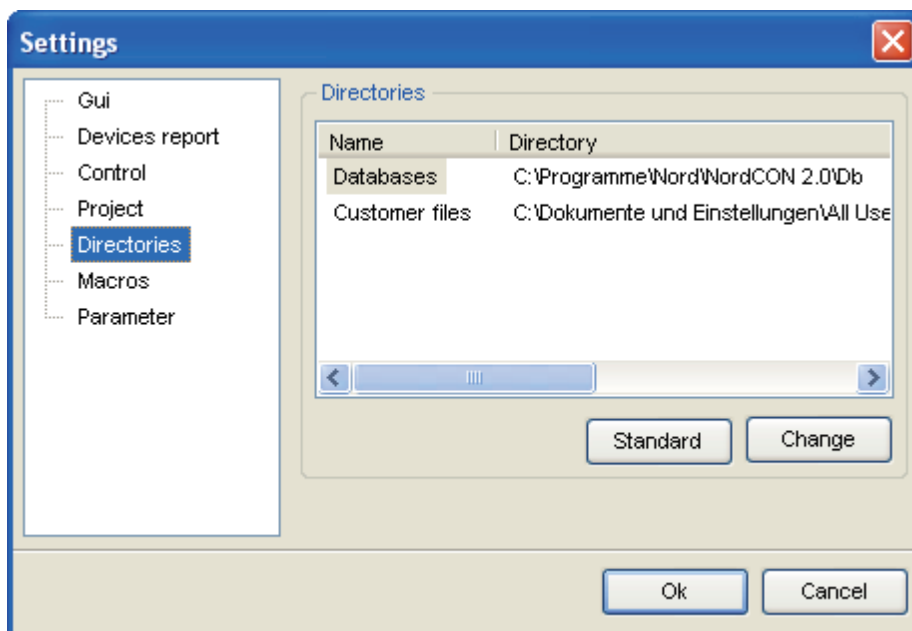
如果启用该选项，则在“读取所有数据”功能期间，若发生错误，将删除设备的数据记录。

删除设备中不能用于通信的数据记录

如果启用该选项，则在执行“读取所有数据”功能期间，若设备无通信，将删除设备的数据记录。

13.5 目录

在该标题中，可以设置参数数据库、配置文件、宏指令文件和内部数据库所在的目录。要更改其中一个路径，必须在列表中突出显示所需的目录。通过单击“更改”按钮可以选择新路径。可以借助“标准”按钮输入每个类别的标准目录。



客户文件

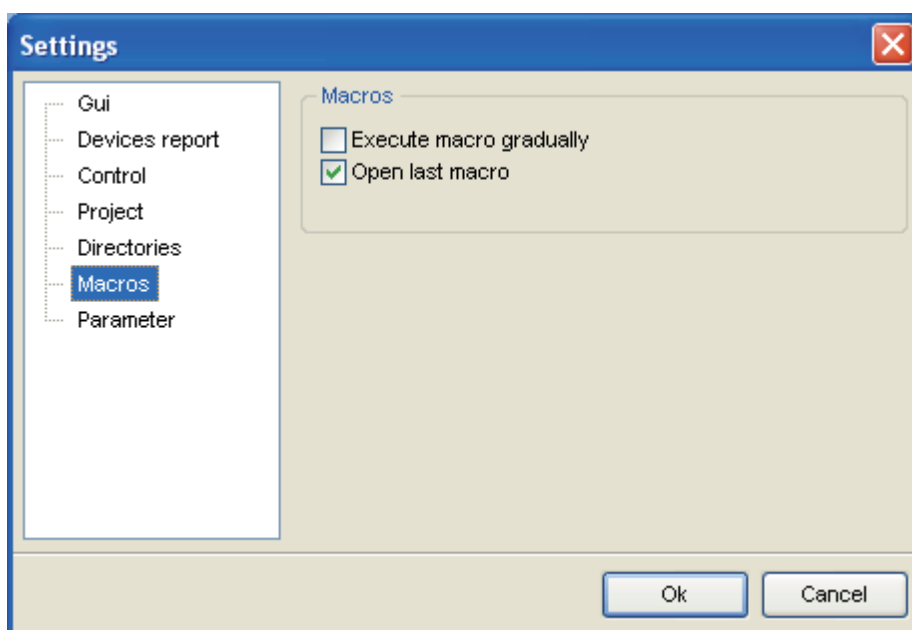
所有客户专用文件，例如宏指令或参数文件，都保存在该目录中。

内部数据库

这些数据库用于程序的内部执行。特定变频器系列的参数结构保存在这些数据库中。

13.6 宏指令编辑器

在该类别中，可以选择8“宏指令编辑器”的设置。



逐步执行宏指令

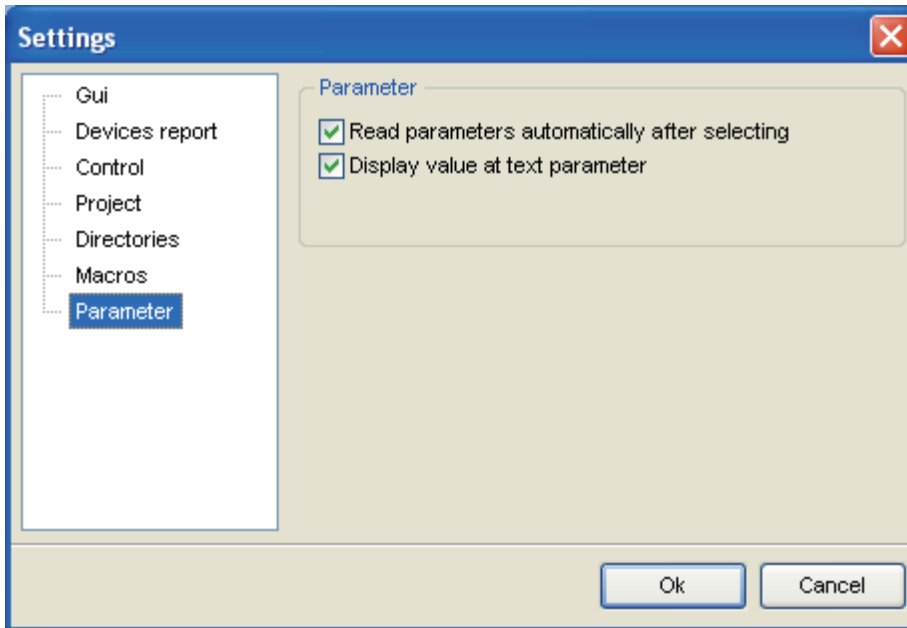
该选项激活或停用逐步执行宏指令。通过激活该选项，必须分别激活每一个宏指令步骤（周期/启动）。

打开最后一个宏指令

该选项激活或停用加载最后一个已打开的宏指令的功能。

13.7 参数

在该类别中，可以选择4“参数化”的设置。



选择后自动读取参数

该选项激活或停用选择后参数的自动读取。

显示带有文本参数的值

该选项激活或停用带有文本参数的数值显示。

13.8 PLC

编译前删除旧日志条目

如果启用该选项，则在编译之前将删除旧的日志条目。

跳转到实时断点（调试模式）

如果启用该选项，则实时断点所在的行将移到可视范围。

14 消息

14.1 错误和信息

发生故障时，将显示在程序中注册的错误编号以及简明的错误信息。

错误信息的解释如下所示：

编号	描述
100	参数编号不可接受
101	无法更改参数值
102	超过参数限值
103	子索引错误
104	无数组
105	无法更改描述
106	描述数据不存在
107	超时接收数据
108	超时发送数据
109	接收数据错误
110	不同指令和回应
200	无法打开串行端口！
201	无法关闭串行端口！
202	首先关闭旧串行端口！
203	未打开串行端口！
204	无法设置通信模块的设置。检查是否支持当前的波特率。
205	无法设置缓冲区！
206	无法超时设置！
207	无法通信！
208	内部对象错误！
210	写入文件错误！
211	无法创建电报！
212	未找到高分辨率计时器！
213	未找到设备！
214	仅适用16位设定值！

编号	描述
215	变频器正在运行。关闭？
216	仅在设备地址为零时才能进行固件更新！
217	无法启动固件更新工具！要修复此问题，请再次安装NORD CON。
218	请添加通信模块！
219	要将文件导入到在线视图吗？
220	无法在这里添加设备！
221	总线处发现不止1台设备！更新可能会导致问题。要继续吗？
222	如果同时使用宏指令和控制窗口，则控制数据不一致！请关闭所有控制窗口或宏指令编辑器。
223	因为参数编辑器打开，所以无法启动传输！请关闭编辑器，并重新启动功能。
224	无法找到在线帮助！要修复此问题，请再次安装NORD CON。
225	因为仍至少有一个设备窗口打开，所以无法断开设备。
226	因为仍至少有一个设备窗口打开，所以无法断开设备。
227	无法读取文件！
228	文件格式未知！
229	用户更改了文件。
230	因为设备未连接，所以无法执行操作！
231	设置已更改。要存储变更吗？
232	计算机不支持中文字符，因此出现表示错误！
233	无法将值转换为INT16！
234	当前版本的设备不支持通过系统总线进行固件更新！
235	当前版本的技术盒不支持通过系统总线进行固件更新！
236	地址0处的设备不支持通过系统总线进行固件更新！
237	PLC未注册！请联系支持（+49（0）180 500 61 84）。
238	注册代码不正确！请联系支持（+49（0）180 500 61 84）。
239	只有使用38400波特的波特率才能进行固件下载！
240	未安装打印机，所以无法打印报告！
241	无法在系统中找到文件！
242	当前版本的技术盒TU3不支持固件更新！
243	无法添加更多设备！
244	项目已更改！要保存项目吗？
245	无法联系设备！

编号	描述
246	未找到设备的PLC程序!
247	未找到设备的参数
248	用户已取消传输!
249	传输期间出现至少一个错误!
250	传输期间出现至少一个警告!
251	输入的IP地址无效!
252	无法添加更多设备!
253	文件已损坏或被操纵!
254	在“USS over TCP”模式下无法进行固件更新!
255	变更需要总线扫描! 要接受变更吗?
256	无法找到项目文件!
257	请首先插入总线模块。
258	无法将所有设置传输到所选的总线模块! 要继续吗?
259	在写入操作期间发生错误!
260	设备的PLC程序不正确!
261	现在要保存项目吗?
262	IP地址已使用!
263	无法找到目录!
300	内部数据库的路径不正确!
301	内部数据库的路径不正确。NORD CON将中止。
302	打开数据库时出错!
303	数据库中的FU型不兼容!
304	数据库中的FU型不相同!
305	保存实际数据库吗?
306	无法打开数据库!
307	不允许的文件夹!
308	无法存储数据库!
309	立即全部读取参数吗?
310	请更新NORD CON! 无法保证无故障的参数传输。
311	打印机安装不正确!
312	对不起, 无法同时运行两个参数窗口。查看打开的窗口吗?
313	必须关闭参数窗口以退出NORD CON!
314	必须关闭参数窗口以进行总线扫描!

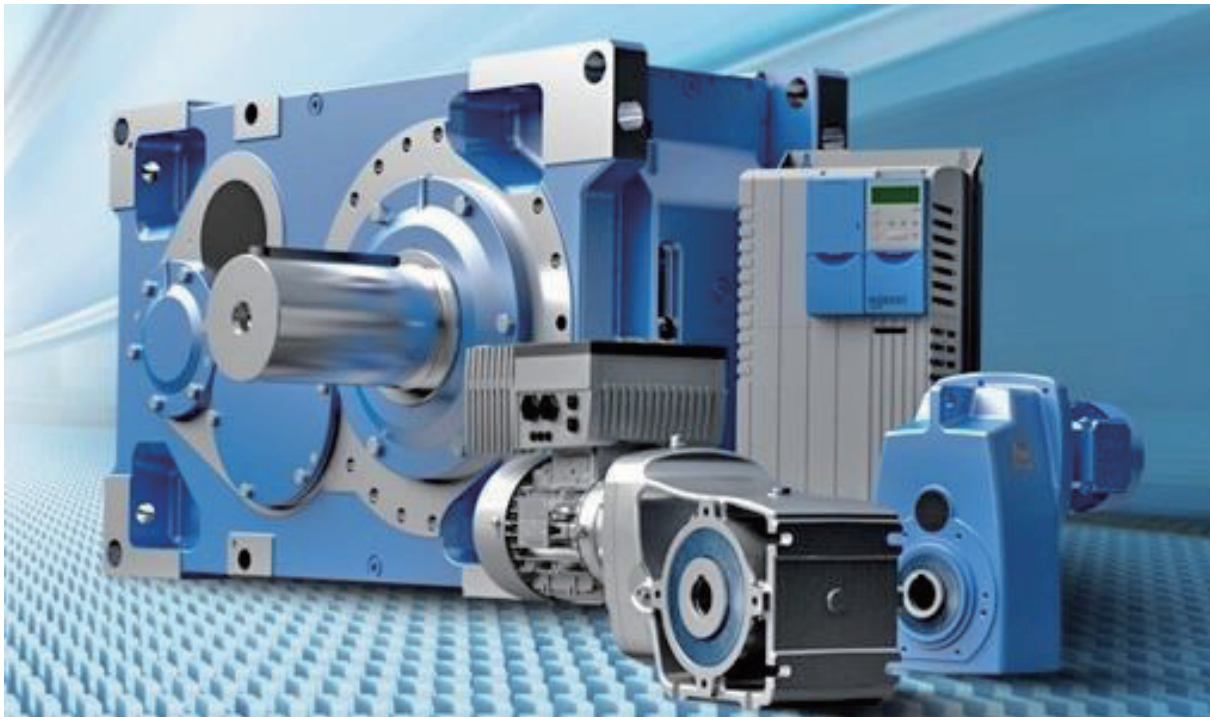
编号	描述
315	参数比较仅能保存为PDF格式。
316	参数仍未永久保存在设备中。一定要关闭?
400	文件版本未知, 所以无法加载文件!
401	文件格式未知, 所以无法加载文件!
402	用户更改了文件!
403	打开文件错误!
405	无宏指令文件!
406	空宏指令列表!
407	已执行宏指令列表!
408	无法找到跳转标记!
409	调度器已启动, 所以无法执行功能。
410	要将变更保存在宏指令中吗?
411	用户更改了文件! 要打开文件吗?
500	仅加载设置吗?
501	设备型号不同吗? 要打开文件吗?
502	文件格式版本未知, 所以无法打开文件!
503	因为文件格式未知, 所以无法打开文件!
504	用户更改了文件! 要打开文件吗?
600	设备控制减少或不可用, 原因如下: 控制字 (P509) 不适用于USS配置!
601	设备控制减少或不可用, 原因如下: 设定值1 (P510.0) 的来源未配置为USS!
602	设备控制减少或不可用, 原因如下: 设定值2 (P510.1) 的来源未配置为USS!
603	设备控制减少或不可用, 原因如下: 控制字 (P509) 和设定值1 (P510.0) 的来源未配置为USS!
604	设备控制减少或不可用, 原因如下: 控制字 (P509) 和设定值2 (P510.1) 的来源不适用于USS配置!
605	设备控制减少或不可用, 原因如下: 设定值1 (P510.0) 和设定值2 (P510.1) 的来源未配置为USS
606	设备控制减少或不可用, 原因如下: 控制字 (P509) 和设定值1 (P510.0) 和设定值2 (P510.1) 的来源未配置为USS!
607	电报超时 (P513) 未激活!
700	因为设备的连接损坏, 所以无法执行操作!
701	因为以下原因远程操作受限! 参数为只读。
800	参数已成功传输

编号	描述
801	参数传输期间出错!
802	用户已取消参数传输!
803	参数传输期间出错! 要保存吗?
804	用户已取消参数传输! 要保存吗?
805	要查看报告吗?
806	用户已取消生成报告!
807	设备连接现已重建! 要继续吗?
900	最多可将5个变量注册到监视列表!
901	文件在翻译之前必须保存。要创建新文件吗?
902	因为文件格式未知, 所以无法打开文件!
903	无法读取文件!
904	用户更改了文件! 要打开文件吗?
905	尚不能执行该功能!
906	启动程序之前必须保存PLC程序!
907	PLC程序已更改! 要保存吗?
908	设置已更改! 要保存吗?
909	不支持PLC格式1.0。

15 诺德驱动系统

通往成功之路的诺德

诺德成立于1965年，如今净销售额已达约4.6亿欧元。我们成功跻身齿轮电机制造商的精英行列，这归功于我们倾听客户并与客户密切合作的战略。在客户的帮助下，我们创建了最佳驱动解决方案，并且公司也一直稳健发展。



全球知识和本地支持

诺德齿轮遍布全球60多个国家。拥有3100多名员工，可确保最短的交货时间和快速的客户服务，无论您身在何处，都可以收到您的驱动器并问题的回复。

让一切移动

凭借我们强大的驱动解决方案，我们甚至能让这个世界上的“巨人们”动起来：港口设施中的大型起重机、可伸缩屋顶的运动场、机场中的行李输送带和滑雪缆车。不管是什么应用，诺德一定会让它动起来

向前驱动



我们的产品体现了紧凑型机械和智能电子之间的创新结合。我们销售和生产机械和电子驱动组件的完整产品线，包括优质齿轮减速器、电机、变频器、伺服控制器和分布式驱动技术。

一起进步

我们的高质量和服务标准源于我们对客户的关注。我们根据客户的意见，开发出完全符合客户要求的创新驱动解决方案。诺德和客户一起建立了成功的长期业务关系。

15.1 诺德驱动系统企业历史

自1965年诺德成立以来，集团各公司一直秉持了满足客户需求的共同战略。	
	
1977	建造现代齿轮生产工厂
1979	在全球范围内建立子公司，并扩大装配中心
1980	采用诺德UNICASE™ “防漏” 箱体设计
1983	建造诺德第一家电机制造厂
1985	建造诺德第一家变频器制造厂
1992	建成铸钢件加工厂。
	
1997	建造意大利电机制造厂
1998	法国新装配厂
2000	英国和奥地利新装配厂
2001	中国新装配厂
2002	中国又一家新装配厂
2003	建造俄罗斯装配厂
2004	建造意大利新电机厂
2005	诺德齿轮40年 在德国巴格特海德开设高架存储系统

	建造中国新装配中心。
2006	在德国奥里希开设新电子产品生产厂
2007	在印度和捷克建造新装配厂
2008	<p>扩建Getriebebau NORD, Bargteheide</p> <ul style="list-style-type: none"> - 建造停车场
	
2009	<p>扩建Getriebebau NORD, Bargteheide</p> <ul style="list-style-type: none"> - 建造下一个高架仓库 - 建造工业减速机装配中心
2011	诺德驱动系统庆祝加德布施工厂第四期建造工程的开工仪式，以及加拿大布兰普顿诺德齿轮有限公司成立25周年。在中国，诺德庆祝在北京东南约100公里的天津开设第二家工厂，而在第五大陆，诺德澳大利亚子公司在达尔文开业。
2012	目前，诺德驱动系统在全球拥有35家子公司。60多个国家的销售和服务合作伙伴不断扩充着诺德的销售和服务网络。公司拥有一支积极进取的员工团队和一整套技术卓越、质量上乘的驱动技术产品，是迎接未来挑战的理想装备。
2013	新建苏州生产厂
2014	巴格特海德总部的服务区和喷漆厂的现代化改造
	
2015	<p>公司成立50周年</p> <p>建造新办公楼</p>

15.2 变频器

15.2.1 SK 135E



性能：0.25 7.5 kW

SK 135E – 分布式电机启动器

包括材料处理领域在内的许多应用都需要电子启动和停止驱动单元。为此，诺德已经开发出了创新的新型电机启动器SK 135E。多功能性，不仅可以启动电机功能，还可以实现缓启动或反转模式。广泛的监测功能，例如防止过热。由于具有I2t触发特性，因此无需配置电机保护开关。电机启动器SK 135E（带有电机安装）的集成电源滤波器能满足最高的EMC要求。

特点和特性

- 缓启动功能
- 反转功能
- 电机或墙式安装
- IP55, (可选IP66)
- 功率范围：3~200 240V从0.25 kW至4.0 kW 3~380, 500V从0.25 kW至7.5 kW
- 机电制动器的控制和连接
- 集成电源滤波器（EMC C1/C2类）
- 2路数字输入
- 2路数字输出

- 温度传感器输入 (TF+/TF-)
- 通过RJ12插口的RS232接口
- 可选ATEX Zone 22 3D (准备中)

请在此处查找电机启动器SK 135E的更多详细信息

15.2.2 SK 180E



性能: 0.25 2.2 kW

SK 180E - 经济型分布式变频器

SK 180E是低功率范围内所有应用的最佳选择，其主要任务是速度控制。诺德专有技术在使用前要经过试验和测试，这样经验证的无传感器的电流矢量控制可始终确保最佳的电压/频率比。SK 180E在EMC分类方面具有明显优势。因此，带有集成电源滤波器的电机安装变频器甚至可用于住宅环境（C1类）。

15.2.3 SK 200E



新型 - SK 200E具有0.33-30hp分布式控制多功能性

经过多年的电机安装交流矢量驱动器的经验，随着诺德SK 200E的发布，现在推出了一系列新型设备，这些设备可以实现各种分布式驱动技术解决方案。这些稳健、可靠和经济的系统适用于工厂应用，例如运输，并针对价格敏感的细分市场进行了专门优化。与SK 500E面板安装系列类似，提供了面向应用的性能级别和相同的高质量功能。分布式组件的预期功能，例如稳健的设计、接插件的集成、通信和I/O信号的快速更换和分布式模块，确保了现场级分布式驱动单元的可靠集成。

供货范围 - SK 200E:

- 1~ 115 V 0.25 – 0.75 kW
1~ 230 V 0.25 – 1.1 kW
1~ 230 V 0.25 – 4.0 kW
3~ 480 V 0.55 – 7.5 kW
- 墙式安装版本
- 分布式模块（同时具有网关功能）

标准防护等级IP55。可选:

- 型号1至3: IP66（带有“C” =涂层的组件）
- 型号4: 带有“C” =涂层的组件，防护等级IP55
- 型号1至3: ATEX Zone 22, 3D或恶劣的环境条件

SK 205E基本单元

- 通过无传感器的电流矢量控制（ISD）的高质量控制过程
- 外部24DCV控制卡电源
- 4个控制输入，可参数化为各种数字功能
- 外部可见LED状态指示灯（控制输入信号状态）外部可调设定值电位计
- 插入式存储器存储模块（EEPROM）
- 电机参数自动识别
- 四个参数集，可在线切换
- 增量式编码器评估（HTL）
- 通过可选制动电阻操作可再生四象限发电机4Q
- PID控制器和过程控制器
- RS232和RS485（RJ12连接器）诊断接口

电机电位计功能

SK 215E

- 基本设备 – 如SK 205E（请见上文）
- 安全功能“安全停机”符合EN 954-1（EN13849）的要求，最高类别4，停止类别0和1。

SK 225E

- 基本设备 – 如SK 205E (请见上文)
- 板载ASi接口

SK 235E

- 基本设备 – 如SK 205E (请见上文)
- 安全功能“安全停机”符合EN 954-1 (EN13849) 的要求, 最高类别4, 停止类别0和1。
- 板载ASi接口

15.2.4 SK 500E



性能:	0.25 2.2 kW 1/3 AC 200 ... 240 V 3.0 18.5 kW 3 AC 200 ... 240 V 0.55 90 kW 3 AC 380 ... 480 V 0.25 160kW
输出频率:	0 ... 400 Hz
使用手册	SK 5xxE SK 54xE

关键字索引

A		通过系统总线进行固件更新	190
关于NORD CON	9	第一步	9
B		H	
备份参数	41	帮助	18
总线错误	32	HMI	181, 182
总线扫描	20	如何更新固件	185
C		如何使用NORD CON	9
取消	21	I	
关闭	13	导入	13
通信	193	信息	200
比较报告	40	插入指令	67
连接	15, 20, 78	界面和视图	12
控制	9, 15, 44, 45, 78, 193, 195	介绍	9
复制	14	L	
复制指令	67	语言	51, 193
创建新指令	68	加载	62
剪切	14	日志	17
剪切指令	67	日志窗口	66
D		M	
删除	14	宏指令	9, 63, 198
详细控制	45, 46	宏指令编辑器	193, 198
设备	15, 78	宏指令生成器	9, 63, 198
设备	20	主菜单	12, 13, 15, 78
设备报告	193, 194	主机 (USS Anfrage)	70
目录	193, 197	测量	61
停靠	26	测量	59
向下	14	菜单	12, 13, 14, 17, 18
E		N	
编辑	14	名称	32
错误	200	新建	13
导出	62	NORD CON	9
导出	13	诺德公司历史	206
附加	18	O	
F		打开	13
文件	13	Oszilloskop	58, 59, 61, 62
筛选器	39	Oszilloskop显示	58
固件更新程序	187	Oszilloskop Drucken	62

P	
参数	39, 40, 193, 199
参数比较	40
参数下载	42
下载参数到设备	15, 78
参数语言	51, 193
离线参数	39
参数设置	15, 78
从设备上传参数	15, 41, 78
粘贴	14
PLC	74, 199
ABS	133
ACOS	138
ADD	134
ADD(.....	134
AND	141
AND(.....	141
ANDN	141
ANDN(.....	141
算术运算符	133
ASIN	138
分配运算符	171
ATAN	138
位运算符	141
按位访问变量	171
BOOL_TO_BYTE	176
总线设定值和实际值	160
BYTE_TO_BOOL	177
BYTE_TO_INT	177
调用ST中的功能块	172
CANopen通信	78
CASE	173
注释	169
比较运算符	148
配置	83
控制盒	77
控制盒和参数盒	162
COS	138
CTD	108
CTU	109
CTUD	110
通过累加器进行数据处理	76
数据类型	168
ST中的数据类型	171
调试	82
DINT_TO_INT	178
DIV	134
DIV(.....	134
编辑器	78
带飞剪的电子齿轮	91
带飞剪的电子齿轮	77
EQ	148
故障信息	179
错误	166
表达式求值	172
退出	175
EXP	139
扩展数学运算符	138
F_TRIG	112
FB_FunctionCurve	128
FB_PIDT1	129
FB_ResetPostion	131
FB_Capture	125
FB_DinCounter	127
FB_DINTToPBOX	120
FB_FlyingSaw	92
FB_Gearing	93
FB_NMT	84
FB_PDConfig	85
FB_PDOResceive	87
FB_PDOSend	89
FB_ReadTrace	117
FB_STRINGToPBOX	123
FB_Weigh	132
FB_WriteTrace	118
FOR循环	174
功能块	83
功能调用	170
GE	149
GT	149
待检点	82
IF	173

过程映像	75	MIN	135
信息参数	163	MOD	136
输入窗口	80	MOD(.....)	136
输入和输出	151	运动控制库	77
指令列表 (AWL/IL)	168	MUL.....	136
INT_TO_BYTE.....	178	MUL(.....)	136
INT_TO_DINT.....	178	MUX.....	137
JMP	175	NE.....	151
JMPC	176	NOT.....	142
JMPCN.....	176	检测点	82
跳转标记	170	运算符	133
跳转.....	175	OR	142
语言.....	168	OR(.....)	142
LD.....	147	ORN.....	143
LDN	147	ORN(.....)	143
LE	150	概述可视化	120
LIMIT	135	参数盒	77
文字.....	168	参数	166
LN.....	139	过程控制器	77
加载和存储运算符	147	处理值	151
加载, 保存和打印	78	程序任务	76
LOG.....	140	R	145
LT	150	R_TRIG	112
MAX	135	REPEAT循环.....	174
MC_MoveAbsolute.....	99	返回	173
MC_WriteParameter_16	107	ROL	144
MC_WriteParameter_32	107	ROR.....	144
MC_Control.....	95	RS触发器	113
MC_Control_MS	96	S	145
MC_Home.....	98	功能范围	77
MC_MoveAdditive.....	100	处理设定值	76
MC_MoveRelative.....	101	设定值和实际值	157
MC_MoveVelocity	101	SHL.....	145
MC_Power	103	SHR	145
MC_ReadActualPos.....	104	SIN.....	138
MC_ReadParameter	104	单步	82
MC_ReadStatus.....	105	说明	74
MC_Reset	106	SQRT	140
MC_Stop	107	SR触发器	113
存储器	75	ST	148
消息窗口	81	标准功能块	108

STN	148	恢复参数	42
结构化文本 (ST)	171	S	
SUB	137	保存	13
SUB(.....)	137	保存和恢复	181, 182
TAN	138	另存为	13
TOF	114	全选	14
TON.....	115	设置	18, 51, 193, 196
TP.....	116	模拟硬件	32
将PLC程序传输到设备	81	标准	19
类型转换	176	启动	21
变量和FB声明	79	启动序列	69
可视化	77	停止	21
参数盒可视化	119	T	
监视和断点显示窗口	80	电报错误	32
WHILE循环	175	工具栏	17, 19, 20, 21
XOR	146	工具栏	11, 12, 19
XOR(.....)	146	U	
XORN.....	146	取消	14
XORN(.....)	146	取消停靠	26
端口	32	向上	14
打印.....	62	USS Anfrage	70, 72
打印.....	13	USS Antwort	71, 72
项目	11, 12, 17, 18, 21, 193, 196	V	
项目文件	196	视图	12, 17, 18, 25
属性窗口	63	查看	40
R		W	
远程	12, 18, 25	窗口变量	63
远程控制	15, 78		
重命名	15, 78		

诺德传动集团

集团总部及研发中心

位于德国汉堡附近的巴格特海德市

创新的驱动解决方案

服务于众多行业分支领域

机械产品

平行轴、斜齿轮、伞齿轮和蜗轮蜗杆减速机

电气产品

IE2/IE3/IE4电机

电子产品

集中式和分布式变频器、电机软启动器和现场分布式系统

7座技术先进的生产基地

供应驱动零部件

遍及5大洲36个国家的子公司和经销商

提供本地库存、组装装配、生产、技术支持和客户服务

全球雇员总数超过3,900名

为您提供定制化驱动解决方案

www.nord.com/locator

诺德（中国）传动设备有限公司

地址：苏州工业园区长阳街510号

邮编：215026

电话：+86-512-8518 0277

传真：+86-512-8518 0278

info@nord.com.cn, www.nord.com

诺德传动集团成员

