
Wiki 4: visual programming languages

Docs

Release 0.0.1

Franz Höpfinger

15.04.2024

Contents:

1	Wiki 0: Meisterschulen-am-Ostbahnhof-Munchen	3
2	Wiki 1: Install-ISOBUS-Environment-docs	5
3	Wiki 2: ISOBUS-VT-Objects	7
4	Wiki 3: ISOBUS-other	9
5	Wiki 4: visual-programming-languages	11
6	Wiki 5: Werkzeug	13
6.1	Gedicht im Stil Göthes	13
6.2	E	14
6.2.1	Abgrenzung	14
6.2.1.1	Abgrenzung visuelle/textuelle Programmiersprachen	14
6.2.1.2	Untergliederung textuelle Programmiersprachen	14
6.2.1.3	Untergliederung visuelle Programmiersprachen	14
6.2.1.4	Weitere Untergliederung visuelle Programmiersprachen	14
6.2.1.4.1	Abgrenzung visuelle Programmiersprachen vs. textuelle Programmiersprachen	14
6.2.2	Visuelle Programmiersprachen	15
6.2.2.1	Literatur:	15
6.2.2.1.1	Allgemeine Visuelle Programmiersprachen	15
6.2.2.1.2	Visuelle Programmiersprachen in der Informatik	15
6.2.2.1.3	Visuelle Programmiersprachen in der Steuerungstechnik	15
6.2.2.1.4	Automatisierungspyramide	16
6.2.2.1.4.1	Visuelle Programmiersprachen in der IEC 61131	16
6.2.2.1.4.2	Visuelle Programmiersprachen in der IEC 61499	16
6.2.2.1.4.3	IEC 61499 Function Block	17
6.2.2.1.4.4	IEC 61499 EEC (Execution Control Chart)	17
6.2.2.1.4.5	Vergleich IEC 61499 zu IEC 61131-3	17
6.2.3	norms	17
6.2.3.1	Normen: IEC 61499 und co.	17
6.2.3.1.1	Beuth Verlag: DIN	17
6.2.3.1.1.1	DIN EN 61499-1	17
6.2.3.1.1.2	DIN EN 61499-2	17
6.2.3.1.1.3	DIN EN 61499-4	17

6.2.3.1.2	IEC	18
6.2.3.1.2.1	IEC 61499-1	18
6.2.3.1.2.2	IEC 61499-2	18
6.2.3.1.2.3	IEC 61499-4	18
6.2.3.1.3	EVS	18
6.2.3.1.3.1	EVS-EN 61499-1	18
6.2.3.1.3.2	EVS-EN 61499-2	18
6.2.3.1.3.3	EVS-EN 61499-4	18
6.2.3.2	Weitere Normen	19
6.2.3.2.1	IEC 61131-3	19
6.2.3.2.2	ISOBUS	19
6.2.3.2.2.1	ISO 11783-1	19
6.2.3.2.2.2	ISO 11783-2	19
6.2.3.2.2.3	ISO 11783-3	19
6.2.3.2.2.4	ISO 11783-4	19
6.2.3.2.2.5	ISO 11783-5	19
6.2.3.2.2.6	ISO 11783-6	19
6.2.3.2.2.7	ISO 11783-7	19
6.2.3.2.2.8	ISO 11783-8	19
6.2.3.2.2.9	ISO 11783-9	19
6.2.3.2.2.10	ISO 11783-10	19
6.2.3.2.2.11	ISO 11783-11	19
6.2.3.2.2.12	ISO 11783-12	19
6.2.3.2.2.13	ISO 11783-13	19
6.2.3.2.2.14	ISO 11783-14	19
6.2.3.2.3	Symbole	19
6.2.3.2.3.1	ISO 7000	20
6.2.3.2.3.2	ISO 7010	20
6.2.4	Allgemeines	20
6.2.4.1	Zahlen	20
6.2.4.2	Datentypen	20
6.2.4.2.1	Datentypen in der IEC 61131-3	20
6.2.4.2.2	Datentypen in der IEC 61499	20
6.2.5	FunctionBlock	23
6.2.6	Literatur zu IEC 61499	24
6.2.6.1	Deutsch	24
6.2.6.1.1	Bücher Deutsch	24
6.2.6.1.1.1	IEC 61499 Function Blocks für den Entwurf von Eingebetteten und Verteilten Systemen	24
6.2.6.1.1.2	Prozess-Steuerungen: Projektierung und Inbetriebnahme mit dem Softwaretool SPaS	24
6.2.6.1.2	Videos Deutsch	24
6.2.6.1.2.1	Trend im Engineering - IEC 61131 versus IEC 61499	24
6.2.6.1.2.2	4diac Playlist	25
6.2.6.1.2.3	nxtControl GmbH	25
6.2.6.1.2.4	Alois Zoitl deutsch	25
6.2.6.1.2.5	4Diac andere Deutsch	25
6.2.6.1.2.6	4Diac	25
6.2.6.1.3	Sonstiges Deutsch	25
6.2.6.1.3.1	universalautomation.org	26
6.2.6.2	Englisch	26
6.2.6.2.1	Bücher Englisch	26
6.2.6.2.2	Videos Englisch	26
6.2.6.2.3	Sonstiges Englisch	27

6.2.7	Hilfe	27
6.2.8	Hardware	27
6.2.8.1	SPS	28
6.2.8.2	PC	28
6.2.8.2.1	dual WLAN	28
6.2.8.2.1.1	ChatGPT	28
6.2.8.2.1.2	unter Windows 10 mit 2 WLAN Verbindungen arbeiten	28
6.2.8.2.1.3	Vergleichstest verschiedener WLAN Adapter:	29
6.2.8.2.1.4	Tabelle	30
6.2.8.2.1.5	Beschreibung der Verschiedenen Adapter:	30
6.2.8.2.1.6	Hama N150 Nano-WLAN-USB-Stick, 2,4 GHz	30
6.2.8.2.1.7	Edimax EW-7811Un V2 Wi-Fi 4 nano Adapter, Version 2	31
6.2.8.2.1.8	Edimax EW-7811UN Super Mini WLAN USB-Stick (150MBit), Version 1	32
6.2.8.2.1.9	Cudy WU650S AC 650Mbit/s drahtloser USB WLAN Stick	32
6.2.8.2.1.10	ALL-WA1200AC	33
6.2.8.2.1.11	ALL-WA0300AC	35
6.2.8.2.1.12	ALL-WA0150N (MTK7601)	36
6.2.8.2.1.13	ALL-WA0100N (RTL8188EU)	37
6.2.9	Eclipse 4diac™	38
6.2.9.1	Über Eclipse 4diac™	38
6.2.9.1.1	Gedicht über 4diac im Stile Göthes	38
6.2.9.1.2	xx	38
6.2.9.1.3	Geschichte:	38
6.2.9.1.4	Open Source	39
6.2.9.2	Installation Eclipse 4diac™	39
6.2.9.3	Eclipse 4diac™ Update	49
6.2.9.3.1	Update	50
6.2.9.3.2	Update URLs for 4Diac	51
6.2.9.3.2.1	Release	51
6.2.9.3.2.2	Release Candidate	51
6.2.9.3.2.3	Milestone	51
6.2.9.3.2.4	Nightly Build	52
6.2.9.4	Eclipse 4diac™ IDE Icon	52
6.2.10	Runtime	52
6.2.10.1	ISOBUS	52
6.2.10.1.1	SoftKey Reduction	53
6.2.11	Das erste Projekt	56
6.2.11.1	Video	56
6.2.11.2	4Diac neues Projekt anlegen	56
6.2.11.3	Steuerung Bohrmaschinenschalter erstellen	60
6.2.12	Interface	71
6.2.12.1	NetworkInterface	72
6.2.12.2	ProcessInterface	72
6.2.12.2.1	Mapping	72
6.2.12.2.2	Eingänge	73
6.2.12.2.3	Ausgänge	73
6.2.12.2.4	Eingänge und Buttons	73
6.2.13	Übungsaufgaben	75
6.2.13.1	Die Aufgaben bauen aufeinander auf, so dass es sich empfiehlt diese in der angege- benen Reihenfolge zu bearbeiten.	75
6.2.13.1.1	Eingänge IX / QX (alles tastend)	75
6.2.13.1.1.1	Aufgabe Einfache Verbindung I	75
6.2.13.1.1.2	Aufgabe Einfache Verbindung II	75

6.2.13.1.2	Logik	75
6.2.13.1.2.1	Aufgabe	75
6.2.13.1.3	Flip Flops I	75
6.2.13.1.3.1	Aufgabe Klick	75
6.2.13.1.4	Rendevouz und Flip-Flop	75
6.2.13.1.4.1	Aufgabe Rendevouz und Flip-Flop	75
6.2.13.1.4.2	Aufgabe Reihenfolge	75
6.2.13.1.5	Flip Flops II	76
6.2.13.1.5.1	Aufgabe Ein verrückter Designer	76
6.2.13.1.5.2	Aufgabe Schreibtischlampe	76
6.2.13.1.6	Flip Flops und Equals	76
6.2.13.1.6.1	Aufgabe Flip Flops und Equals	76
6.2.13.1.7	Flip Flops und Logikfunktionen	76
6.2.13.1.7.1	Aufgabe Flip Flops und Logikfunktionen	76
6.2.13.1.8	Flip Flops und E Funktionen	76
6.2.13.1.8.1	Aufgabe Flip Flops und E Funktionen	76
6.2.13.1.8.2	Aufgabe Fahrtrichtungsanzeiger	76
6.2.13.1.8.3	Aufgabe Warnblinker	77
6.2.13.1.9	Flip Flops und led_strip	77
6.2.13.1.9.1	Aufgabe Flip Flops und led_strip	77
6.2.14	constants	77
6.2.15	TypeLibrary	77
6.2.15.1	devices	78
6.2.15.1.1	BeagleBoneBlack	78
6.2.15.1.2	FBRT_WINDOW	78
6.2.15.1.3	FORTE_PC	78
6.2.15.1.4	PFC200	78
6.2.15.1.5	RaspberryPI	78
6.2.15.2	esp32	78
6.2.15.2.1	nvs	78
6.2.15.2.1.1	NVS_UDINT	78
6.2.15.2.1.2	NVS_UINT	79
6.2.15.3	esp-addons	79
6.2.15.3.1	hsv2rgb	79
6.2.15.3.2	strip_set_pixel	80
6.2.15.4	events	80
6.2.15.4.1	ARTimeOut	80
6.2.15.4.2	ATimeOut	80
6.2.15.4.3	E_CTD	80
6.2.15.4.4	E_CTU	81
6.2.15.4.5	E_CTUD	82
6.2.15.4.6	E_CTUD_UDINT	83
6.2.15.4.7	E_CYCLE Taktgenerator (Blinker)	83
6.2.15.4.8	E_DELAY	84
6.2.15.4.9	E_DEMUX	84
6.2.15.4.10	E_D_FF	85
6.2.15.4.11	E_F_TRIG	85
6.2.15.4.12	E_MERGE	86
6.2.15.4.13	E_N_TABLE	86
6.2.15.4.14	E_PERMIT	87
6.2.15.4.15	E_RDELAY	87
6.2.15.4.16	E_REND	87
6.2.15.4.17	E_RESTART	88
6.2.15.4.18	E_RS	88

6.2.15.4.19	E_RTimeOut	89
6.2.15.4.20	E_R_TRIG	89
6.2.15.4.21	E_SELECT	90
6.2.15.4.22	E_SPLIT	90
6.2.15.4.23	E_SR = (Holzspalter Schalter)	91
6.2.15.4.24	E_SWITCH	92
6.2.15.4.25	E_TABLE	92
6.2.15.4.26	E_TABLE_CTRL	93
6.2.15.4.27	E_TimeOut	93
6.2.15.4.28	E_TRAIN	94
6.2.15.4.29	E_T_FF	95
6.2.15.4.29.1	E_T_FF Anwendungen	95
6.2.15.4.29.2	E_T_FF Anwendungen Taster	95
6.2.15.4.29.3	E_T_FF Anwendungen Blinker	97
6.2.15.5	iec61131-3	98
6.2.15.5.1	arithmetic	98
6.2.15.5.1.1	ADD_2	98
6.2.15.5.1.2	ADD_3	99
6.2.15.5.1.3	ADD_4	99
6.2.15.5.1.4	ADD_5	99
6.2.15.5.1.5	ADD_6	99
6.2.15.5.1.6	F_DIV	100
6.2.15.5.1.7	F_MOD	100
6.2.15.5.1.8	F_MUL	100
6.2.15.5.1.9	F_SUB	100
6.2.15.5.2	booleanOperators	100
6.2.15.5.2.1	AND_2	100
6.2.15.5.2.2	AND_3	103
6.2.15.5.2.3	AND_4	103
6.2.15.5.2.4	AND_5	103
6.2.15.5.2.5	AND_6	103
6.2.15.5.2.6	F_NOT	103
6.2.15.5.2.7	OR_2	103
6.2.15.5.2.8	OR_3	104
6.2.15.5.2.9	OR_4	104
6.2.15.5.2.10	OR_5	105
6.2.15.5.2.11	OR_6	105
6.2.15.5.2.12	XOR_2	106
6.2.15.5.2.13	XOR_3	106
6.2.15.5.2.14	XOR_4	106
6.2.15.5.2.15	XOR_5	106
6.2.15.5.2.16	XOR_6	107
6.2.15.5.3	comparison	107
6.2.15.5.3.1	F_EQ	107
6.2.15.5.3.2	F_GE	107
6.2.15.5.3.3	F_GT	107
6.2.15.5.3.4	F_LE	108
6.2.15.5.3.5	F_LT	108
6.2.15.5.3.6	F_NE	108
6.2.15.5.4	conversion	108
6.2.15.5.4.1	TIME	108
6.2.15.5.4.2	F_TIME_IN_MS_TO_UDINT	108
6.2.15.5.4.3	F_TIME_IN_NS_TO_UDINT	109
6.2.15.5.4.4	F_TIME_IN_S_TO_UDINT	109

6.2.15.5.4.5	F_TIME_IN_US_TO_UDINT	109
6.2.15.5.5	numerical	109
6.2.15.5.5.1	F_ABS	109
6.2.15.5.5.2	F_ACOS	109
6.2.15.5.5.3	F_ASIN	110
6.2.15.5.5.4	F_ATAN	110
6.2.15.5.5.5	F_COS	110
6.2.15.5.5.6	F_EXP	110
6.2.15.5.5.7	F_LN	110
6.2.15.5.5.8	F_LOG	111
6.2.15.5.5.9	F_SIN	111
6.2.15.5.5.10	F_SQRT	111
6.2.15.5.5.11	F_TAN	111
6.2.15.5.6	selection	111
6.2.15.5.6.1	F_LIMIT	111
6.2.15.5.6.2	F_MAX	112
6.2.15.5.6.3	F_MIN	112
6.2.15.5.6.4	F_MUX_2	113
6.2.15.5.6.5	F_SEL	113
6.2.15.6	io	113
6.2.15.6.1	DataPanel	113
6.2.15.6.1.1	DataPanel_xDB0808_DIO_SAE_J1939	113
6.2.15.6.1.2	DataPanel_xDB_08_PVE_SAE_J1939	114
6.2.15.6.1.3	DataPanel_xDB_16_ADI_SAE_J1939	114
6.2.15.6.1.4	DataPanel_xDB_16_DO_SAE_J1939	115
6.2.15.6.1.5	DataPanel_xDB_16_PWM_SAE_J1939	115
6.2.15.6.2	io esp32 Taster/Ausgänge Hutschienenmopped	116
6.2.15.6.2.1	Hutschienenmopped_3_DI	116
6.2.15.6.2.2	Hutschienenmopped_3_DI_DOUBLE_CLICK	116
6.2.15.6.2.3	Hutschienenmopped_3_DI_DOWN	116
6.2.15.6.2.4	Hutschienenmopped_3_DI_LONG_PRESS_HOLD	116
6.2.15.6.2.5	Hutschienenmopped_3_DI_LONG_PRESS_START	117
6.2.15.6.2.6	Hutschienenmopped_3_DI_REPEAT	117
6.2.15.6.2.7	Hutschienenmopped_3_DI_SINGLE_CLICK	117
6.2.15.6.2.8	Hutschienenmopped_3_DI_UP	117
6.2.15.6.2.9	Hutschienenmopped_4_DO	118
6.2.15.6.2.10	Hutschienenmopped_6_DIA	118
6.2.15.6.2.11	Hutschienenmopped_6_DIA_DOUBLE_CLICK	118
6.2.15.6.2.12	Hutschienenmopped_6_DIA_DOWN	119
6.2.15.6.2.13	Hutschienenmopped_6_DIA_LONG_PRESS_HOLD	119
6.2.15.6.2.14	Hutschienenmopped_6_DIA_LONG_PRESS_START	119
6.2.15.6.2.15	Hutschienenmopped_6_DIA_REPEAT	120
6.2.15.6.2.16	Hutschienenmopped_6_DIA_SINGLE_CLICK	120
6.2.15.6.2.17	Hutschienenmopped_6_DIA_UP	120
6.2.15.6.3	io isobus	121
6.2.15.6.3.1	UT_AuxiliaryInputMapping	121
6.2.15.6.3.2	UT_ButtonMapping	121
6.2.15.6.3.3	UT_ButtonMapping_DOWN	122
6.2.15.6.3.4	UT_ButtonMapping_HOLD	122
6.2.15.6.3.5	UT_ButtonMapping_UP	122
6.2.15.6.3.6	UT_SoftkeyMapping Softkey Taster am Isobus Display	123
6.2.15.6.3.7	UT_SoftkeyMapping_DOWN	123
6.2.15.6.3.8	UT_SoftkeyMapping_HOLD	124
6.2.15.6.3.9	UT_SoftkeyMapping_UP	124

6.2.15.6.4	IE - Input Event (0 Bit)	124
6.2.15.6.5	IX - Input BOOL (1 Bit)	125
6.2.15.6.6	IB - Input BYTE (8 Bit)	125
6.2.15.6.7	IW - Input WORD (16 Bit)	126
6.2.15.6.8	ID - Input DWORD (32 Bit)	126
6.2.15.6.9	IL - Input LWORD (64 bit)	126
6.2.15.6.10	QX - Output BOOL (1 Bit)	126
6.2.15.6.11	QB - Output Byte (8 Bit)	127
6.2.15.6.12	QW - Output WORD (16 bit)	127
6.2.15.6.13	QD - Output DWORD (32 bit)	127
6.2.15.6.14	QL - Output LWORD (64 Bit)	128
6.2.15.7	isobus	128
6.2.15.7.1	c	128
6.2.15.7.1.1	IsoCommonDef	128
6.2.15.7.1.2	IsoVtcApi	129
6.2.15.7.2	colours	150
6.2.15.7.2.1	COLOR_BLACK	150
6.2.15.7.2.2	COLOR_BLUE	151
6.2.15.7.2.3	COLOR_CYAN	151
6.2.15.7.2.4	COLOR_GREEN	151
6.2.15.7.2.5	COLOR_GREY	151
6.2.15.7.2.6	COLOR_LIME	151
6.2.15.7.2.7	COLOR_MAGENTA	152
6.2.15.7.2.8	COLOR_MAROON	152
6.2.15.7.2.9	COLOR_NAVY	152
6.2.15.7.2.10	COLOR_OLIVE	152
6.2.15.7.2.11	COLOR_PURPLE	152
6.2.15.7.2.12	COLOR_RED	153
6.2.15.7.2.13	COLOR_SILVER	153
6.2.15.7.2.14	COLOR_TEAL	153
6.2.15.7.2.15	COLOR_WHITE	153
6.2.15.7.2.16	COLOR_YELLOW	153
6.2.15.7.3	CbVtStatus	154
6.2.15.7.4	Q_ActiveMask	154
6.2.15.7.5	Q_Attribute	154
6.2.15.7.6	Q_BackgroundColour (Farbprogrammierung IsoBus Display)	155
6.2.15.7.7	Q_ChangeObjectLabel	155
6.2.15.7.8	Q_ChangePolygonPoint	156
6.2.15.7.9	Q_ChangePolygonScale	156
6.2.15.7.10	Q_ChildLocation	156
6.2.15.7.11	Q_ChildPosition	156
6.2.15.7.12	Q_CtrlAudioSignal	156
6.2.15.7.13	Q_EndPoint	157
6.2.15.7.14	Q_ESC	157
6.2.15.7.15	Q_ExecuteExtendedMacro	157
6.2.15.7.16	Q_ExecuteMacro	157
6.2.15.7.17	Q_FillAttributes	157
6.2.15.7.18	Q_FontAttributes	158
6.2.15.7.19	Q_GetAttribute	158
6.2.15.7.20	Q_GraphicsContext	158
6.2.15.7.21	Q_LineAttributes	158
6.2.15.7.22	Q_ListItem	158
6.2.15.7.23	Q_LockUnlockMask	159
6.2.15.7.24	Q_NumericValue	159

6.2.15.7.25	Q_ObjEnableDisable	162
6.2.15.7.26	Q_ObjHideShow	162
6.2.15.7.27	Q_ObjSelectInput	162
6.2.15.7.28	Q_Priority	163
6.2.15.7.29	Q_SelectActiveWorkingSet	163
6.2.15.7.30	Q_SelectColourMap	163
6.2.15.7.31	Q_SetAudioVolume	163
6.2.15.7.32	Q_Size	163
6.2.15.7.33	Q_SoftKeyMask	164
6.2.15.7.34	Q_StringValue	164
6.2.15.8	net	165
6.2.15.8.1	CLIENT_1	165
6.2.15.8.2	CLIENT_2_1	165
6.2.15.8.3	PUBLISH_0	165
6.2.15.8.4	PUBLISH_1	166
6.2.15.8.5	PUBLISH_10	166
6.2.15.8.6	PUBLISH_2	167
6.2.15.8.7	PUBLISH_3	167
6.2.15.8.8	PUBLISH_4	168
6.2.15.8.9	PUBLISH_5	168
6.2.15.8.10	PUBLISH_6	169
6.2.15.8.11	PUBLISH_7	169
6.2.15.8.12	PUBLISH_8	170
6.2.15.8.13	PUBLISH_9	171
6.2.15.8.14	SERVER_1	171
6.2.15.8.15	SERVER_1_2	172
6.2.15.8.16	SUBSCRIBE_0	172
6.2.15.8.17	SUBSCRIBE_1	172
6.2.15.8.18	SUBSCRIBE_10	173
6.2.15.8.19	SUBSCRIBE_2	173
6.2.15.8.20	SUBSCRIBE_3	174
6.2.15.8.21	SUBSCRIBE_4	174
6.2.15.8.22	SUBSCRIBE_5	175
6.2.15.8.23	SUBSCRIBE_6	175
6.2.15.8.24	SUBSCRIBE_7	176
6.2.15.8.25	SUBSCRIBE_8	176
6.2.15.8.26	SUBSCRIBE_9	177
6.2.15.9	reconfiguration	177
6.2.15.9.1	EC_KILL_ELEM	177
6.2.15.9.2	EC_SET_EVT	178
6.2.15.9.3	EC_START_ELEM	178
6.2.15.9.4	EC_STOP_ELEM	178
6.2.15.9.5	ST_CREATE_CONN	178
6.2.15.9.6	ST_CREATE_FB	178
6.2.15.9.7	ST_DEL_CONN	178
6.2.15.9.8	ST_DEL_FB	179
6.2.15.9.9	ST_REC_CONN	179
6.2.15.9.10	ST_SET_PARM	179
6.2.15.10	resources	179
6.2.15.10.1	EMB_RES	179
6.2.15.11	segments	179
6.2.15.11.1	Ethernet	179
6.2.15.12	utils	179
6.2.15.12.1	E_STOPWATCH	179

6.2.15.12.2	FB_RANDOM	180
6.2.15.12.3	OUT_ANY_CONSOLE	180
7	Links	181
7.1	logiBUS®	181
7.2	ISomaX®	181
7.3	Eclipse 4diac™	181
8	Indices and tables	183
	Stichwortverzeichnis	185



Need help or have a question? Join the chat at [Gitter](#) or open a new Discussion

<https://docs.ms-muc-docs.de/projects/visual-programming-languages-docs/de/latest/>

<https://github.com/Meisterschulen-am-Ostbahnhof-Muenchen/visual-programming-languages-docs/discussions>

Wiki 0: Meisterschulen-am-Ostbahnhof-Munchen

Wiki 0: [Wiki](#) das „über“ Wiki

ein Suche im über Wiki findet Ergebnisse aus allen 5 Wikis

KAPITEL 2

Wiki 1: Install-ISOBUS-Environment-docs

Wiki 1: [Wiki Install-ISOBUS-Environment](#)

KAPITEL 3

Wiki 2: ISOBUS-VT-Objects

Wiki 2: [Wiki ISOBUS-VT-Objects](#)

KAPITEL 4

Wiki 3: ISOBUS-other

Wiki 3: [Wiki ISOBUS-other](#)

Wiki 4: visual-programming-languages

Wiki 4: [Wiki visual-programming-languages](#)

Wiki 5: Werkzeug

Wiki 5: [Wiki Werkzeug](#)

Welcome to visual-programming-languages documentation!

Einleitung

Willkommen bei Visual Programming Languages Docs!

Haben Sie sich jemals gefragt, wie man Programme visuell erstellt? Mit unserer Website sind Sie genau an der richtigen Stelle. Hier finden Sie alle Informationen und Ressourcen, die Sie benötigen, um in die Welt der visuellen Programmierung einzutauchen.

Wir bieten umfassende Dokumentationen und Tutorials für verschiedene visuelle Programmiersprachen, einschließlich Blockly, Scratch und vielen mehr. Unsere Inhalte sind für Anfänger und Fortgeschrittene geeignet und werden von erfahrenen Entwicklern und Dozenten erstellt.

Auf unserer Website finden Sie auch Links zu hilfreichen Ressourcen und Tools, die Ihnen bei der Erstellung von visuellen Programmen helfen werden.

Also, worauf warten Sie noch? Beginnen Sie jetzt mit der visuellen Programmierung und werden Sie Teil der wachsenden Gemeinschaft von Entwicklern, die sich für diese spannende Technologie interessieren. Besuchen Sie uns jetzt und starten Sie Ihre Reise in die Welt der visuellen Programmierung.

IEC 61499

6.1 Gedicht im Stil Göthes

Die IEC 61499, ein System moderner Prägung. Wie ein Uhrwerk, stets in Line, gestaltet es die Automatisierung.

In Schichten aufgebaut, wie ein Turm aus Bauklotzsteinen. Jeder Baustein gut durchdacht, um Funktionen zu vereinen.

Die Programmierung wird leicht, als wäre es ein Kinderspiel. Einheitlichkeit im ganzen Bereich, doch ist jedes System individuell.

Wie ein Gedanke, der durch den Kopf geht, so fließt das Programm durch die Schichten. Dynamisch, flexibel, stets bereit, die Systeme zu steuern und zu richten.

Die IEC 61499, ein Meisterwerk moderner Technik. Wie ein Dichter, der mit Worten spricht, so steuert es die Anlagen geschickt.

6.2 E

Die IEC 61499 ist eine internationale Norm für die Echtzeit-Verarbeitung von Steuerungssystemen. Sie definiert eine Architektur und eine Methode zur Modellierung von Steuerungssystemen und bietet damit eine gemeinsame Sprache für die Kommunikation zwischen verschiedenen Systemen. Die IEC 61499 bietet eine flexible Plattform für die Entwicklung von Steuerungssystemen und ist in vielen Industrien weit verbreitet.

Eclipse 4 DIAC

Eclipse 4 DIAC (Distributed Industrial Automation and Control) ist eine Entwicklungsumgebung für die Erstellung von Anwendungen, die auf der IEC 61499 basieren. Es bietet viele nützliche Werkzeuge und Funktionen, um die Entwicklung von Steuerungssystemen zu vereinfachen und zu beschleunigen. Eclipse 4 DIAC ist eine kostenlose Open-Source-Anwendung und eine beliebte Wahl für viele Entwickler von Industriesteuerungssystemen.

Bemerkung: Fragen/Anregungen: hier lang:

[Discussion Forum](#)

Bemerkung: This project is under active development.

Contents

Welcome to the visual-programming-languages wiki!

Fragen/Anregungen: hier lang:

[Discussion Forum](#)

6.2.1 Abgrenzung

6.2.1.1 Abgrenzung visuelle/textuelle Programmiersprachen

6.2.1.2 Untergliederung textuelle Programmiersprachen

6.2.1.3 Untergliederung visuelle Programmiersprachen

6.2.1.4 Weitere Untergliederung visuelle Programmiersprachen

6.2.1.4.1 Abgrenzung visuelle Programmiersprachen vs. textuelle Programmiersprachen

Literatur:

https://de.wikipedia.org/wiki/Visuelle_Programmiersprache

https://en.wikipedia.org/wiki/Visual_programming_language

<https://www.youtube.com/watch?v=MxJcdqOX9V0>

→ dieses Video bringt es recht prägnant auf den Punkt.

<https://www.c-plusplus.net/forum/topic/350943/visuelle-oder-textuelle-programmiersprachen>

<https://wiki.print2forms.de/doku.php?id=print2forms:skripte:textuellvsvisuell>

6.2.2 Visuelle Programmiersprachen

6.2.2.1 Literatur:

https://de.wikipedia.org/wiki/Visuelle_Programmiersprache

https://en.wikipedia.org/wiki/Visual_programming_language

6.2.2.1.1 Allgemeine Visuelle Programmiersprachen

Beispiele für Allgemeine Visuelle Programmiersprachen sind zum Beispiel Scratch von Google.

<https://sites.google.com/ochieseeducation.ca/teachertech/coding/scratch>

[https://de.m.wikipedia.org/wiki/Scratch_\(Programmiersprache\)](https://de.m.wikipedia.org/wiki/Scratch_(Programmiersprache)) Scratch (Programmiersprache) erziehungsorientierte visuelle Programmiersprache für Kinder und Jugendliche

Eine Umsetzung davon findet sich bei M5Stack:

<https://flow.m5stack.com/>

https://m5stack.github.io/UIFlow_doc/en/

<https://shop.m5stack.com/pages/uiflow>

<https://www.youtube.com/watch?v=Og6fYoyiTik>

UIFlow von M5Stack nutzt Blockly von Google.

Blockly/Scratch usw. lehnt sich in der Struktur stark an Textuelle Programmiersprachen an. oft ist ein Umschalten zwischen einer Textuellen und Visuellen Ansicht möglich.

In erster Linie dienen diese Sprachen dem leichteren Erlernen von Textuellen Programmiersprachen auf Visuellem Weg.

6.2.2.1.2 Visuelle Programmiersprachen in der Informatik

https://de.wikipedia.org/wiki/Unified_Modeling_Language

Siehe Hierzu auch <https://resourcecenter.ies.ieee.org/education/webinars/IESWEB0230.html> ab 17:00

6.2.2.1.3 Visuelle Programmiersprachen in der Steuerungstechnik

neben diesen „erziehungsorientierten“ Visuellen Programmiersprachen gibt es weitere die sich mehr an das Ingenieurswesen wenden, zum Beispiel LabVIEW.

<https://www.ni.com/de-de/shop/labview.html>

Im Bereich der Programmierung von Steuerungen gibt es ebenfalls schon viele Jahre diverse visuelle Programmiersprachen zum Beispiel Ladder logic: <https://de.wikipedia.org/wiki/Kontaktplan>

6.2.2.1.4 Automatisierungspyramide

<https://de.wikipedia.org/wiki/Automatisierungspyramide>

6.2.2.1.4.1 Visuelle Programmiersprachen in der IEC 61131

- https://en.wikipedia.org/wiki/IEC_61131
 - https://en.wikipedia.org/wiki/IEC_61131-3
 - * IL https://en.wikipedia.org/wiki/Instruction_list
 - * LD https://en.wikipedia.org/wiki/Ladder_logic
 - * FBD https://en.wikipedia.org/wiki/Function_block_diagram
 - * SFC https://en.wikipedia.org/wiki/Sequential_function_chart
 - * ST https://en.wikipedia.org/wiki/Structured_text
- https://de.wikipedia.org/wiki/EN_61131
 - EN 61131-3
 - * AWL <https://de.wikipedia.org/wiki/Anweisungsliste>
 - ist eine textuelle Programmiersprache die an Assembler erinnert.
 - * KOP <https://de.wikipedia.org/wiki/Kontaktplan>
 - ist eine visuelle Programmiersprache die an einen Schaltplan erinnert. So wird beispielsweise die Möller/Eaton Easy 719 programmiert.
 - * FBS <https://de.wikipedia.org/wiki/Funktionsbausteinsprache>
 - ist eine visuelle Programmiersprache die mit Blöcken arbeitet. So wird beispielweise die Siemens Logo programmiert. Bei Siemens heißt diese Sprache FUP Funktionsplan
 - * AS <https://de.wikipedia.org/wiki/Ablaufsprache>
 - ist eine visuelle Programmiersprache die auch als GRAFCET bekannt ist.
 - * ST https://de.wikipedia.org/wiki/Strukturierter_Text
 - ist eine Textuelle Programmiersprache die an Pascal erinnert.

6.2.2.1.4.2 Visuelle Programmiersprachen in der IEC 61499

https://en.wikipedia.org/wiki/IEC_61499

https://de.wikipedia.org/wiki/EN_61499

6.2.2.1.4.3 IEC 61499 Function Block

Die visuelle Programmiersprache die in der IEC 61499 definiert ist ähnelt dem FBS aus der IEC 61131-3.

dadurch lassen sich alle Funktionsbausteine die aus der IEC 61131-3 bekannt sind, und auch solche von Drittbibliotheken z.B. OSCAT auch in der IEC 61499 implementieren. Umgekehrt ist das nicht in jedem Fall möglich.

6.2.2.1.4.4 IEC 61499 EEC (Execution Control Chart)

EEC ist eine State Machine innerhalb IEC 61499

6.2.2.1.4.5 Vergleich IEC 61499 zu IEC 61131-3

Z.B. Video Siehe Hierzu auch <https://resourcecenter.ies.ieee.org/education/webinars/IESWEB0230.html> ab 43:13

6.2.3 norms

6.2.3.1 Normen: IEC 61499 und co.

Allgemeine Infos:

https://de.wikipedia.org/wiki/EN_61499

https://en.wikipedia.org/wiki/IEC_61499

6.2.3.1.1 Beuth Verlag: DIN

6.2.3.1.1.1 DIN EN 61499-1

<https://www.beuth.de/de/norm/din-en-61499-1/209152944>

6.2.3.1.1.2 DIN EN 61499-2

<https://www.beuth.de/de/norm/din-en-61499-2/209156091>

6.2.3.1.1.3 DIN EN 61499-4

<https://www.beuth.de/de/norm/din-en-61499-4/191866520>

6.2.3.1.2 IEC

6.2.3.1.2.1 IEC 61499-1

<https://webstore.iec.ch/publication/5506>

6.2.3.1.2.2 IEC 61499-2

<https://webstore.iec.ch/publication/5507>

6.2.3.1.2.3 IEC 61499-4

<https://webstore.iec.ch/publication/5508>

6.2.3.1.3 EVS

<https://www.evs.ee/en>

die EVS ist Quasi die DIN von Estland.

dort sind die meisten Normen viel viel billiger als bei BEUTH.

Empfehlung: die 3 User Licence kaufen!

6.2.3.1.3.1 EVS-EN 61499-1

<https://www.evs.ee/en/evs-en-61499-1-2013>

6.2.3.1.3.2 EVS-EN 61499-2

<https://www.evs.ee/en/evs-en-61499-2-2013>

6.2.3.1.3.3 EVS-EN 61499-4

<https://www.evs.ee/en/evs-en-61499-4-2013>

6.2.3.2 Weitere Normen

6.2.3.2.1 IEC 61131-3

6.2.3.2.2 ISOBUS

6.2.3.2.2.1 ISO 11783-1

6.2.3.2.2.2 ISO 11783-2

6.2.3.2.2.3 ISO 11783-3

6.2.3.2.2.4 ISO 11783-4

6.2.3.2.2.5 ISO 11783-5

6.2.3.2.2.6 ISO 11783-6

6.2.3.2.2.7 ISO 11783-7

6.2.3.2.2.8 ISO 11783-8

6.2.3.2.2.9 ISO 11783-9

6.2.3.2.2.10 ISO 11783-10

6.2.3.2.2.11 ISO 11783-11

6.2.3.2.2.12 ISO 11783-12

6.2.3.2.2.13 ISO 11783-13

6.2.3.2.2.14 ISO 11783-14

6.2.3.2.3 Symbole

Abruf unter: <https://www.iso.org/obp>

6.2.3.2.3.1 ISO 7000

ISO 7000 Graphical symbols for use on equipment

ISO 7000 / IEC 60417 Graphical symbols for use on equipment

6.2.3.2.3.2 ISO 7010

ISO 7010 — Graphical symbols — Safety colours and safety signs — Registered safety signs

6.2.4 Allgemeines

6.2.4.1 Zahlen

Natürliche Zahl – Wikipedia

Ganze Zahl – Wikipedia

Rationale Zahl – Wikipedia

Reelle Zahl – Wikipedia

6.2.4.2 Datentypen

Integer (Datentyp) – Wikipedia

Gleitkommazahl – Wikipedia

6.2.4.2.1 Datentypen in der IEC 61131-3

Data Types

https://help.codesys.com/webapp/_cds_datatype_integer;product=codesys;version=3.5.17.0

Datentypen

6.2.4.2.2 Datentypen in der IEC 61499

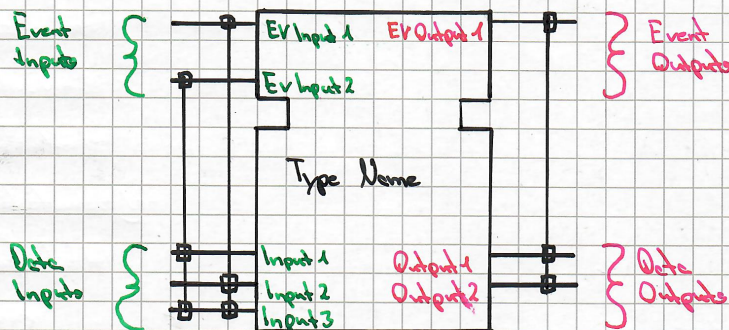
Siehe auch: IEC 61499, Table E.1

Typ	BYTE Länge	Bit Länge	Wertebereich	Wertebereich
			min	max
BOOL	0	1 Bit	0 = FALSE	1= TRUE
SINT	1	8 Bit	128	127
INT	2	16 Bit	32.768	32.767
DINT	4	32 Bit	2.147.483.648	2.147.483.647
LINT	8	64 Bit	9.223.372.036.854.775.808	9.223.372.036.854.775.807
USINT	1	8 Bit	0	255
UINT	2	16 Bit	0	65.535
UDINT	4	32 Bit	0	4.294.967.295
ULINT	8	64 Bit	0	18.446.744.073.709.551.615
REAL	4	32 Bit		
LREAL	8	64 Bit		
DATE	8	64 Bit		
TIME	8	64 Bit		
TIME_OF_DAY	12	96 Bit	1970-01-01-00:00:00.000	
DATE_AND_TIME	20	160 Bit	1970-01-01-00:00:00.000	
BYTE	1	8 Bit	0	255
WORD	2	16 Bit	0	65.535
DWORD	4	32 Bit	0	4.294.967.295
LWORD	8	64 Bit	0	18.446.744.073.709.551.615

6.2.5 FunctionBlock

Beschreibung Function Blocks

Der obere Teil wird als Kopf bezeichnet.



Der untere Teil wird als Körper bezeichnet.

- Event Eingänge und Ausgänge sind im Kopf.
- Daten Eingänge und Ausgänge sind im Körper.

Events: Events stehen für die Synchronisation und die Interaktion zwischen den Mechanismen der Ausführungskontrolle der Function Blocks, die im Netzwerk verbunden sind.

Ein Block der Daten sendet, muss ein Event an den Empfängerblock übertragen, um sicherzustellen, dass dieser die Daten liest.

Daten sind typisiert in der Programmiersprache.

Bei den Function Blocks werden Zuordnungen zwischen Events und relevanten Daten ^{dargestellt} (vertikale Linien). Die Linien verbinden das Event mit seine zugeordneten Daten-Ein- und Ausgänge.

Heißt die Werte der Variablen die mit einem bestimmten Event assoziiert sind, werden beim Eintreten des Events aktualisiert.

6.2.6 Literatur zu IEC 61499

6.2.6.1 Deutsch

6.2.6.1.1 Bücher Deutsch

6.2.6.1.1.1 IEC 61499 Function Blocks für den Entwurf von Eingebetteten und Verteilten Systemen

Valeriy Vyatkin: IEC 61499 Function Blocks für den Entwurf: Dritte Ausgabe

<https://www.isa.org/products/iec-61499-function-blocks-fur-den-entwurf-von-eing>

<https://www.amazon.de/Function-Entwurf-Eingebetteten-Verteilten-Systemen/dp/0876640447>

<https://www.yumpu.com/de/document/view/51443348/books-preview-dr-valeriy-vyatkin>

<https://www.fb61499.com/IEC61499German.html>

Buch "IEC 61499 Function Blocks für den Entwurf von Eingebetteten und Verteilten Systemen"

Autor: Valeriy Vyatkin, Übersetzung: Ewald Matull

ISBN: 978-0-876640-44-9, ISA-Verlag

Musterseiten Preview

Direkt bestellen: www.isa.org/IEC61499German

6.2.6.1.1.2 Prozess-Steuerungen: Projektierung und Inbetriebnahme mit dem Softwaretool SPaS

<https://www.amazon.de/Prozess-Steuerungen-Projektierung-Inbetriebnahme-Softwaretool-SPaS/dp/3540710833>

6.2.6.1.2 Videos Deutsch

6.2.6.1.2.1 Trend im Engineering - IEC 61131 versus IEC 61499

Das nachfolgende Video

14.12.2011

Live-Interview mit Dieter Hess (Geschäftsführer, 3S-Smart Software Solutions) beim publish-industry VIP-Talk auf der ROTEN COUCH, SPS/IPC/DRIVES 2011

jedoch ist von Dieter Hess, der absolut kein Freund der IEC 61499 ist. Schauen Sie selbst:

<https://www.youtube.com/watch?v=Go75w3-HJhw>

6.2.6.1.2.2 4diac Playlist

<https://www.youtube.com/channel/UCt0MINo6Y7dXCZSDACXFi1g>

<https://www.youtube.com/@4diac177/about>

6.2.6.1.2.3 nxtControl GmbH

https://www.youtube.com/playlist?list=PLD-CHxmB4P7fdBSAS_woF0Gd5XUqk9o4g

<https://www.youtube.com/@nxtControl/about>

6.2.6.1.2.4 Alois Zoitl deutsch

<https://www.youtube.com/playlist?list=PLD-CHxmB4P7cT89D0MPK9bppQurDQ8wuj>

6.2.6.1.2.5 4Diac andere Deutsch

https://www.youtube.com/playlist?list=PLD-CHxmB4P7eQ4_rTjP3MxaVv9xyiTRAm

6.2.6.1.2.6 4Diac

(diese Videos sind von MS-MUC selbst erstellt)

<https://www.youtube.com/playlist?list=PLD-CHxmB4P7cGEqEQIC-rgRDKBVHfbcYW>

6.2.6.1.3 Sonstiges Deutsch

<https://docplayer.org/71630572-Verteilte-steuerungen-nach-iec-61499.html>

<https://slideplayer.org/slide/879446/>

https://issuu.com/hk-gt/docs/tech_201203_df55f692a05682

<http://vyatkin.org/uploads/talk.pdf>

https://www.profinet.felser.ch/was_ist_in_iec_61499_dann_anders_als_in_iec_61131.html

<http://vyatkin.org/iec61499day/>

<http://vyatkin.org/iec61499day/pdfs/Vyatkin%20Intro%20IEC%2061499.pdf>

<https://link.springer.com/content/pdf/10.1007/s00502-021-00918-w.pdf> (PDF Download!)

https://de.frwiki.wiki/wiki/CEI_61499

<https://www.jku.at/lit-cyber-physical-systems-lab/ueber-uns/team/univ-prof-di-dr-alois-zoitl/> (unten auf der Seite)

<https://wirautomatisierer.industrie.de/steuerungstechnik-ipc-sps/iec-61499-als-offener-standard/>

<https://wirautomatisierer.industrie.de/steuerungstechnik-ipc-sps/schneider-electric-mit-dem-automation-expert-auf-neuen-pfaden/>

https://de.wikipedia.org/wiki/EN_61499

<https://www.sps-magazin.de/allgemein/automatisieren-mit-der-iec-61499/>

<https://prozesstechnik.industrie.de/chemie/automation-chemie/iec-61499-der-traum-vom-offenen-standard-lebt/>

<https://www.nxtcontrol.com/verteilung/>

<https://www.computer-automation.de/steuerungsebene/steuern-regeln/ein-neues-buendnis-auf-basis-der-iec-61499.191725.html>

6.2.6.1.3.1 universalautomation.org

<https://universalautomation.org/>

<https://www.computer-automation.de/steuerungsebene/steuern-regeln/das-single-line-engineering.130428.html>

6.2.6.2 Englisch

6.2.6.2.1 Bücher Englisch

Valeriy Vyatkin: IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design

<https://www.amazon.de/Function-Embedded-Distributed-Control-Systems/dp/1936007932>

<http://www.vyatk.in.org/book/Pages%20from%20IEC%2061499%20Vyatkin%20Second%20Edition.pdf>

Distributed Control Applications: Guidelines, Design Patterns, and Application Examples with the Iec 61499 (Industrial Information Technology)

<https://www.amazon.de/Distributed-Control-Applications-Application-Information/dp/1138892955>

Modelling Control Systems Using IEC 61499 (Iet Control Engineering, Band 95)

<https://www.amazon.de/Modelling-Control-Systems-Using-Engineering/dp/1849197601>

6.2.6.2.2 Videos Englisch

<https://youtu.be/quCrTcl8qSc>

<https://www.youtube.com/playlist?list=PLD-CHxmB4P7fPvyNI7OhlzT0CQF7CI6uT>

<https://www.youtube.com/playlist?list=PLD-CHxmB4P7dD6R8jOnamCuvV4NwSgrQT>

<https://www.youtube.com/playlist?list=PLD-CHxmB4P7c1DEYmsE2eJaq-jF82GRRi>

<https://resourcecenter.ies.ieee.org/education/webinars/IESWEB0230.html>

<https://www.youtube.com/watch?v=quCrTcl8qSc>

<https://www.youtube.com/watch?v=6ppKrOIzWE8>

<https://www.youtube.com/watch?v=K9iItQBC-ac>

<https://www.youtube.com/watch?v=rptqwLv9s-M>

<https://www.youtube.com/watch?v=sUu5Dq8OiZ4>

<https://www.youtube.com/watch?v=0oqRrviF7GQ>

<https://www.youtube.com/watch?v=R58kH0JMd7o>

<https://www.youtube.com/watch?v=VqQVdPE6lm0> (Awite Bioenergie GmbH)

https://www.youtube.com/watch?v=U_ZMVoDH8oQ

6.2.6.2.3 Sonstiges Englisch

https://publik.tuwien.ac.at/files/PubDat_251131.pptx

IEC 61499 Applications on LinkedIn

IEC 61499 FAQ on LinkedIn

https://acris.aalto.fi/ws/portalfiles/portal/87471087/Jhunjhunwala_Monitoring_design_pattern_for_distributed_automation_systems.pdf

<https://www.vyatkin.org/>

<https://iec61499.com/>

https://www.eclipse.org/4diac/en_help.php?helppage=html/before4DIAC/iec61499.html

https://www.eclipse.org/4diac/events/docs/2019/EclipseInsight_SelfConfigurableAutomationSystems.pdf

https://www.eclipse.org/4diac/events/docs/2016/UW_10_Melik.pdf

https://en.wikipedia.org/wiki/IEC_61499

- Wikipedia article on IEC 61499
- Introduction to IEC 61499 in the Eclipse 4diac documentation
- Modelling Control Systems Using IEC 61499 by Robert W. Lewis and Alois Zoitl
- Real-Time Execution for IEC 61499 by Alois Zoitl
- Distributed Control Applications by Alois Zoitl and Thomas Strasser

<https://www.eclipse.org/4diac/index.php>

https://www.eclipse.org/4diac/en_help.php

https://www.eclipse.org/4diac/en_help.php?helppage=html/4diacIDE/overview.html

https://www.eclipse.org/4diac/en_help.php?helppage=html/4diacIDE/use4diacLocally.html

6.2.7 Hilfe

https://bugs.eclipse.org/bugs/buglist.cgi?list_id=21301704&product=4DIAC

https://www.eclipse.org/forums/index.php?t=thread&frm_id=308

<https://mattermost.eclipse.org/eclipse/channels/4diac>

<https://accounts.eclipse.org/mailling-list/4diac-dev>

6.2.8 Hardware

Ein PC (Personal Computer) ist ein allgemein verwendbarer Computer, der für verschiedene Anwendungen verwendet werden kann, wie zum Beispiel Büroarbeit, Surfen im Internet, Spielen und Multimedia. Eine SPS (Speicherprogrammierbare Steuerung) hingegen ist ein spezialisiertes Gerät, das hauptsächlich in industriellen Anwendungen verwendet wird, um Prozesse automatisch zu steuern und zu überwachen. Ein SPS hat in der Regel keine Möglichkeit für Benutzerinteraktion und ist auf die Steuerung von Maschinen und Anlagen ausgelegt. Im Gegensatz dazu ist ein PC flexibel und kann für viele verschiedene Aufgaben verwendet werden.

6.2.8.1 SPS

Eine SPS (Speicherprogrammierbare Steuerung) ist ein elektronisches Gerät, das in industriellen Anwendungen verwendet wird, um Prozesse automatisch zu steuern und zu überwachen. Es besteht aus einem Mikroprozessor oder einem Mikrocontroller, der das Steuerprogramm enthält, sowie aus Eingabe- und Ausgabeschnittstellen, die es ermöglichen, Signale von Sensoren und Aktoren zu empfangen und zu senden. Eine SPS kann verwendet werden, um Maschinen und Anlagen zu steuern, indem sie bestimmte Aktionen ausführt, wenn bestimmte Bedingungen erfüllt sind. Sie kann auch verwendet werden, um Prozesse zu überwachen, indem sie Daten sammelt und speichert, die später für die Fehlerdiagnose und die Optimierung des Prozesses verwendet werden können.

6.2.8.2 PC

6.2.8.2.1 dual WLAN

6.2.8.2.1.1 ChatGPT

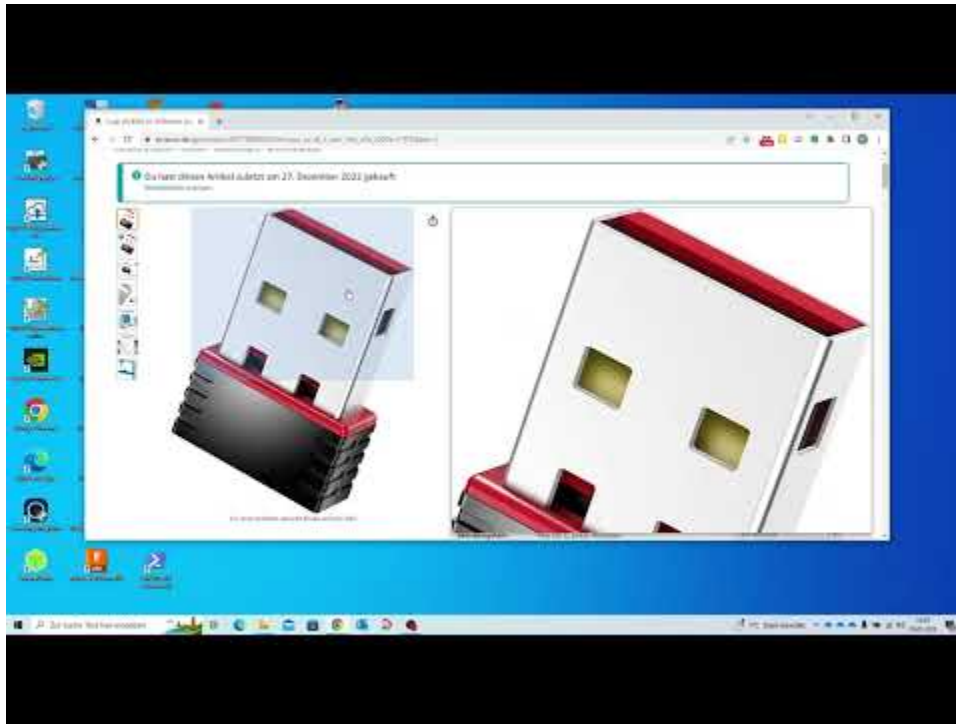
alle Texte auf dieser Seite sind erstellt mit <https://chat.openai.com/chat>

Erstaunlich

6.2.8.2.1.2 unter Windows 10 mit 2 WLAN Verbindungen arbeiten

In diesem Video wird gezeigt, wie man unter Windows 10 mit zwei WLAN-Verbindungen arbeiten kann. Egal ob Sie auf der Arbeit oder zu Hause sind, es kann vorkommen, dass Sie sich mit einem öffentlichen WLAN und einem privaten WLAN verbinden müssen. Oder vielleicht möchten Sie Ihre Internetverbindung zwischen zwei verschiedenen Netzwerken wechseln, ohne die Verbindung zu unterbrechen. Mit diesem Video lernen Sie, wie Sie mehrere WLAN-Verbindungen unter Windows 10 einrichten und verwalten können, um Ihre Arbeit effizienter und produktiver zu gestalten.

Video: [unter Windows 10 mit 2 WLAN Verbindungen arbeiten](#)



6.2.8.2.1.3 Vergleichstest verschiedener WLAN Adapter:

In diesem Artikel werden mehrere WLAN USB-Adapter unter Windows 10 und Ubuntu 22.04 LTS getestet. Der Fokus liegt dabei auf der Funktionalität der Adapter sowie auf der Frage, ob sie ohne die Installation eines Treibers sofort erkannt werden. Dies ist von großem Vorteil, da es Zeit und Mühe spart, wenn man keinen extra Treiber suchen, herunterladen und installieren muss. Der Artikel gibt einen Überblick über die Ergebnisse des Tests und zeigt, welche Adapter ohne Probleme funktionieren und welche eventuell zusätzliche Schritte erfordern. Insgesamt wird der Artikel eine wertvolle Ressource für jeden sein, der einen WLAN USB-Adapter unter Windows 10 oder Ubuntu 22.04 LTS verwenden möchte.

In diesem Artikel werden insgesamt sieben verschiedene WLAN USB-Adapter getestet. Diese umfassen:

1. Hama N150 Nano-WLAN-USB-Stick, 2,4 GHz
2. Edimax EW-7811Un V2 Wi-Fi 4 nano Adapter, Version 2
3. Edimax EW-7811UN Super Mini WLAN USB-Stick (150MBit), Version 1
4. Cudy WU650S AC 650Mbit/s drahtloser USB WLAN Stick
5. ALL-WA1200AC
6. ALL-WA0300AC
7. ALL-WA0150N (MTK7601)
8. ALL-WA0100N (RTL8188EU)

TODO Edimax EW-7611ULB

Diese Adapter werden sowohl unter Windows 10 als auch unter Ubuntu 22.04 LTS getestet, um ihre Funktionalität und ihre Fähigkeit zur sofortigen Erkennung ohne Treiberinstallation zu bewerten. Der Artikel gibt einen Überblick über die Ergebnisse des Tests und zeigt, welche Adapter ohne Probleme funktionieren und welche eventuell zusätzliche

Schritte erfordern. Insgesamt wird der Artikel eine wertvolle Ressource für jeden sein, der einen WLAN USB-Adapter unter Windows 10 oder Ubuntu 22.04 LTS verwenden möchte.

6.2.8.2.1.4 Tabelle

- läuft unter Windows 10 ohne Treiberinstallation
- läuft unter Ubuntu 22.04 LTS ohne Treiberinstallation

Typ	Link	Win- dows 10	Ubuntu 22.04 LTS	CCI.OS- 2.0.3	CCI.OS- 3.3.0
Hama N150 Nano-WLAN-USB-Stick, 2,4 GHz	link	nicht ge- testet	nicht getes- tet	nicht getestet	nicht getestet
Edimax EW-7811Un V2 Wi-Fi 4 nano Adapter, Version 2	link	geht	geht	nicht funktio- nal	geht
Edimax EW-7811UN Super Mini WLAN USB-Stick (150MBit), Version 1	link	geht	geht	geht	geht
Cudy WU650S AC 650Mbit/s drahtloser USB WLAN Stick	link	geht	nicht funk- tional	nicht funktio- nal	nicht funktio- nal
ALL-WA1200AC	link – in- net24	geht	geht	nicht funktio- nal	nicht funktio- nal
ALL-WA0300AC	link – in- net24	geht	nicht funk- tional	nicht funktio- nal	nicht funktio- nal
ALL-WA0150N (MTK7601)	link – in- net24	geht	geht	nicht funktio- nal	nicht funktio- nal
ALL-WA0100N (RTL8188EU)	link – in- net24	geht	geht	geht	geht

6.2.8.2.1.5 Beschreibung der Verschiedenen Adapter:

6.2.8.2.1.6 Hama N150 Nano-WLAN-USB-Stick, 2,4 GHz

Der Hama N150 Nano-WLAN-USB-Stick ist ein WLAN-Adapter, der auf dem 802.11n-Standard basiert und eine maximale Datenübertragungsrate von 150Mbps unterstützt. Er nutzt einen 2,4GHz-Funk und unterstützt nur das 2.4GHz Frequenzband. Es unterstützt auch die neuesten Sicherheitsstandards wie WPA, WPA2 und WEP, um Ihre Verbindung zu schützen. Der Adapter hat einen sehr kleinen und leichten Formfaktor, was es einfach zu transportieren und zu verwenden macht. Es ist kompatibel mit Windows, MacOS und Linux und kann einfach an einen USB-Anschluss angeschlossen werden. Der Hama N150 Nano-WLAN-USB-Stick ist eine gute Wahl für Personen die einen kleinen, einfachen und preiswerten WLAN-Adapter suchen.

6.2.8.2.1.7 Edimax EW-7811Un V2 Wi-Fi 4 nano Adapter, Version 2



Der Edimax EW-7811Un V2 Wi-Fi 4 nano Adapter ist ein WLAN-Adapter, der auf dem 802.11n-Standard basiert und eine maximale Datenübertragungsrate von 150Mbps unterstützt. Er nutzt einen 2,4GHz-Funk und unterstützt nur das 2.4GHz Frequenzband. Es unterstützt auch die neuesten Sicherheitsstandards wie WPA, WPA2 und WEP, um Ihre

Verbindung zu schützen. Der Adapter hat einen sehr kleinen und leichten Formfaktor, was es einfach zu transportieren und zu verwenden macht. Es ist kompatibel mit Windows, MacOS und Linux und kann einfach an einen USB-Anschluss angeschlossen werden. Es ist die Version 2 des Edimax EW-7811Un nano Adapter, und kann sich durch seine kompakte Größe und gute Leistung auszeichnen.

6.2.8.2.1.8 Edimax EW-7811UN Super Mini WLAN USB-Stick (150MBit), Version 1

Der Edimax EW-7811UN Super Mini WLAN USB-Stick ist ein kleiner und leichter WLAN-Adapter, der an den USB-Anschluss eines Computers angeschlossen werden kann. Mit einer Datenübertragungsrate von 150MBit/s ist er in der Lage, eine schnelle und stabile Verbindung zu einem WLAN-Netzwerk herzustellen. Dieser USB-Stick ist in der Version 1 erhältlich und unterstützt die WLAN-Standards 802.11b/g/n. Er ist einfach zu installieren und zu verwenden und eignet sich perfekt für den Einsatz zu Hause oder im Büro.

6.2.8.2.1.9 Cudy WU650S AC 650Mbit/s drahtloser USB WLAN Stick



Der Cudy WU650S AC 650Mbit/s drahtloser USB WLAN Stick ist ein WLAN-Adapter, der an den USB-Anschluss eines Computers angeschlossen werden kann. Mit einer Datenübertragungsrate von 650Mbit/s bietet er eine schnelle

und stabile Verbindung zu WLAN-Netzwerken. Er unterstützt die WLAN-Standards 802.11ac und ist damit in der Lage, hohe Datenraten und eine lange Reichweite zu bieten. Der Stick ist klein und leicht und einfach zu installieren und zu verwenden. Er eignet sich perfekt für den Einsatz zu Hause, im Büro oder unterwegs. Zusätzlich bietet er eine starke Signalstärke und eine gute Abdeckung, was ihn zu einer guten Wahl für die Nutzung von Online-Spielen, Video-Streaming oder anderen Anwendungen mit hohem Datenverbrauch macht.

Der Cudy WU650S AC 650Mbit/s drahtloser USB WLAN Stick ist ein WLAN-Adapter, der unter Linux (Ubuntu 22.04 LTS) nicht ohne Treiberinstallation funktioniert. Das bedeutet, dass man zusätzliche Software oder Treiber installieren muss, damit der Stick auf dem System erkannt und verwendet werden kann. Dies kann entweder durch die Verwendung von proprietären Treibern oder durch die Verwendung von Open-Source-Treibern erfolgen. Es ist wichtig, dass man die richtigen Treiber für das betreffende System und die verwendete Hardware auswählt, um Probleme zu vermeiden und eine optimale Leistung zu erzielen. Es ist auch ratsam, die Anweisungen des Herstellers oder die Dokumentation des Produkts zu lesen, bevor man die Treiber installiert, um sicherzustellen, dass man die richtigen Schritte ausführt.

6.2.8.2.1.10 ALL-WA1200AC





Der ALL-WA1200AC ist ein WLAN USB-Adapter, der auf dem neuesten Wireless-AC-Standard basiert. Er unterstützt die neuesten WLAN-Technologien, wie z.B. MU-MIMO und Beamforming, um eine höhere Geschwindigkeit und eine größere Reichweite zu erreichen. Mit einer Datenübertragungsrate von bis zu 1200Mbps, ist der ALL-WA1200AC in der Lage, hochauflösende Videos und Daten in Echtzeit zu übertragen. Es unterstützt auch Dual-Band-Funktionen, was bedeutet das es sowohl 2,4Ghz als auch 5Ghz Frequenzen unterstützt. Dies ermöglicht es dem Adapter, Interferenzen zu vermeiden und eine höhere Geschwindigkeit und Zuverlässigkeit zu erreichen. Es hat einen kleinen und leichten Formfaktor, was es einfach zu transportieren und zu verwenden macht.

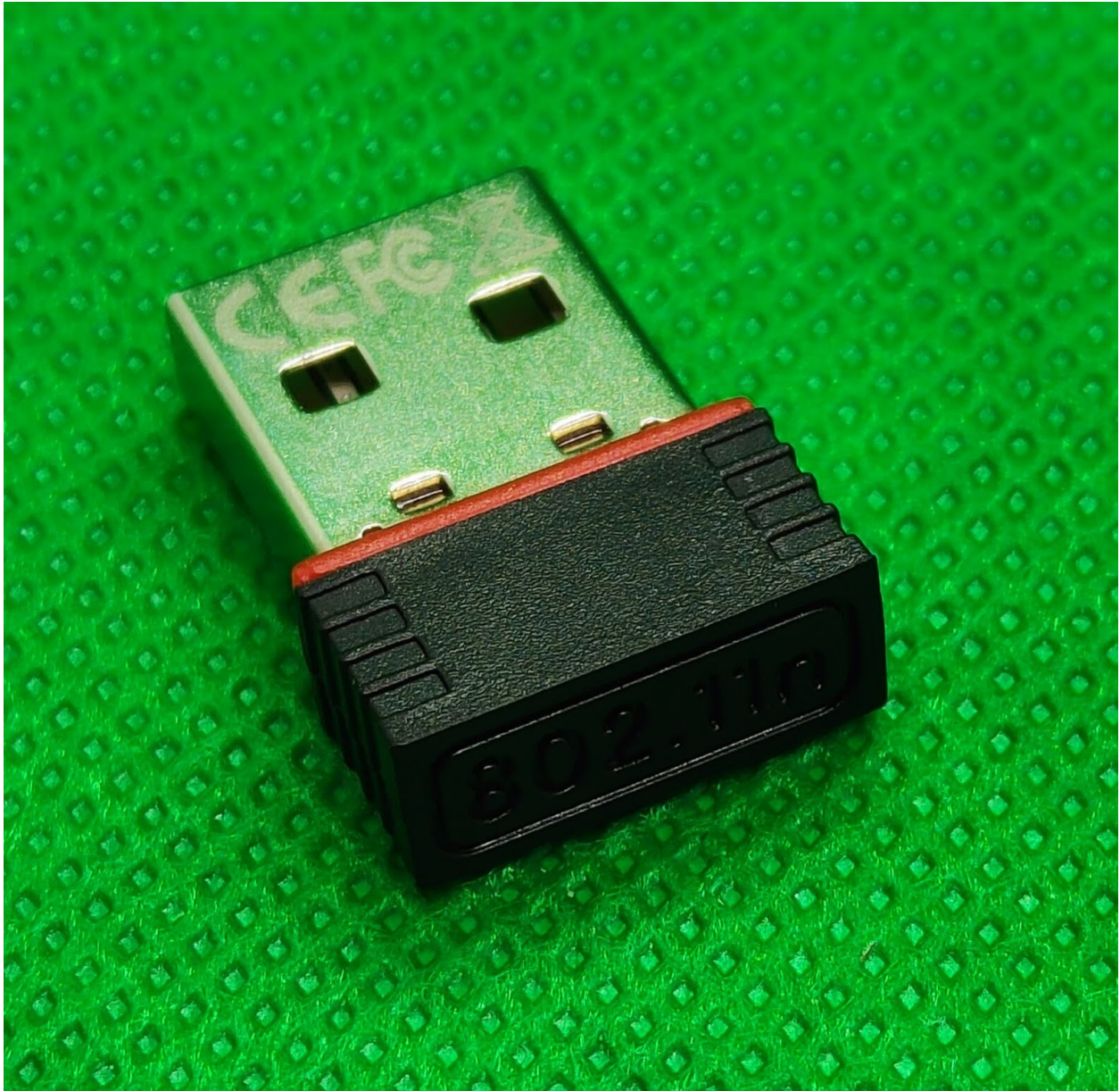
6.2.8.2.1.11 ALL-WA0300AC



Der ALL-WA0300AC ist ein WLAN USB-Adapter, der auf dem Wireless-AC-Standard basiert und eine maximale Datenübertragungsrate von 300Mbps unterstützt. Es unterstützt sowohl 2.4GHz als auch 5GHz Frequenzbänder und hat eine erweiterte Funktionen wie MU-MIMO und Beamforming, die es ermöglichen eine höhere Geschwindigkeit und eine größere Reichweite zu erreichen. Dieser Adapter unterstützt auch die neuesten Sicherheitsstandards wie WPA3, WPA2, WPA und WEP, um Ihre Verbindung zu schützen. Es hat einen kleinen und leichten Formfaktor, was es einfach zu transportieren und zu verwenden macht. Es ist kompatibel mit Windows, MacOS und Linux und kann einfach an einen USB-Anschluss angeschlossen werden.

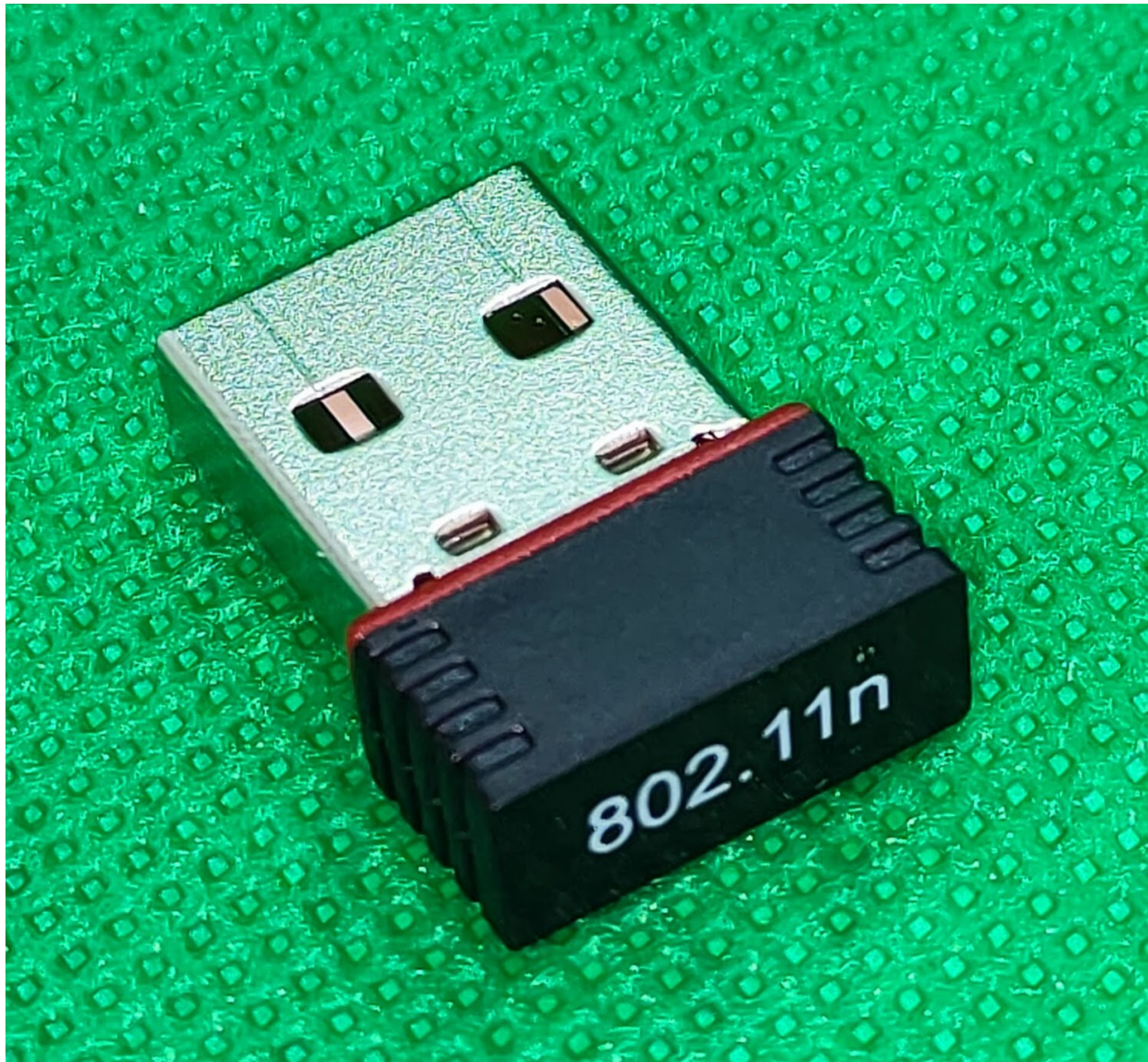
Der ALL-WA0300AC ist ein drahtloser USB-WLAN-Stick, der unter Linux (Ubuntu 22.04 LTS) nicht ohne Treiberinstallation funktioniert. Das bedeutet, dass man zusätzliche Software oder Treiber installieren muss, damit der Stick auf dem System erkannt und verwendet werden kann. Dies kann entweder durch die Verwendung von proprietären Treibern oder durch die Verwendung von Open-Source-Treibern erfolgen. Es ist wichtig, dass man die richtigen Treiber für das betreffende System und die verwendete Hardware auswählt, um Probleme zu vermeiden und eine optimale Leistung zu erzielen. Es ist auch ratsam, die Anweisungen des Herstellers oder die Dokumentation des Produkts zu lesen, bevor man die Treiber installiert, um sicherzustellen, dass man die richtigen Schritte ausführt. Der ALL-WA0300AC unterstützt Dual-Band-WLAN (2,4 und 5GHz) und bietet eine hohe Datenübertragungsrate von bis zu 300Mbps. Es ist kompatibel mit den meisten WLAN-Routern und bietet eine starke Signalstärke und eine gute Abdeckung.

6.2.8.2.1.12 ALL-WA0150N (MTK7601)



Der ALL-WA0150N (MTK7601) ist ein WLAN USB-Adapter, der auf dem 802.11n-Standard basiert und eine maximale Datenübertragungsrate von 150Mbps unterstützt. Er nutzt den Mediatek MTK7601 Chip und unterstützt nur das 2.4GHz Frequenzband. Es unterstützt auch die neuesten Sicherheitsstandards wie WPA, WPA2 und WEP, um Ihre Verbindung zu schützen. Der Adapter hat einen kleinen und leichten Formfaktor, was es einfach zu transportieren und zu verwenden macht. Es ist kompatibel mit Windows, MacOS und Linux und kann einfach an einen USB-Anschluss angeschlossen werden. Allerdings ist es nicht so leistungsfähig wie die neuesten WLAN-Adapter und hat auch eine geringere Reichweite und Geschwindigkeit.

6.2.8.2.1.13 ALL-WA0100N (RTL8188EU)



Der ALL-WA0100N (RTL8188EU) ist ein WLAN USB-Adapter, der auf dem 802.11n-Standard basiert und eine maximale Datenübertragungsrate von 150Mbps unterstützt. Es nutzt den Realtek RTL8188EU Chip und unterstützt sowohl 2.4GHz als auch 5GHz Frequenzbänder. Es unterstützt auch die neuesten Sicherheitsstandards wie WPA, WPA2 und WEP, um Ihre Verbindung zu schützen. Der Adapter hat einen kleinen und leichten Formfaktor, was es einfach zu transportieren und zu verwenden macht. Es ist kompatibel mit Windows, MacOS und Linux und kann einfach an einen USB-Anschluss angeschlossen werden. Allerdings ist es nicht so leistungsfähig wie die neuesten WLAN-Adapter und hat auch eine geringere Reichweite und Geschwindigkeit.

6.2.9 Eclipse 4diac™

6.2.9.1 Über Eclipse 4diac™

6.2.9.1.1 Gedicht über 4diac im Stile Göthes

Im 4diac fließen Ströme von Code, es lebt und pulsiert in seiner eig'nen Ode. Es tanzt im Rhythmus der Bits und Bytes, verschmilzt Elektronik und Automatisierung in eine Einheit.

In seinen Tiefen, wie ein Ozean aus Daten, fließt das Leben von Prozessen und Maschinenraten. Es webt ein Netz von Sensor zu Aktor, schafft Verbindungen von der Fernsteuerung bis zur Leitwarte.

Das System 4diac ist der Taktgeber, lenkt die Schritte der Robotik und steuert die Hebel. Es ist das Herz, das schlägt im Takt, und alle Maschinen in harmonischem Gleichklang bringt.

O Göthe, wärest du nur noch hier, du sähest das 4diac in all seiner Pracht und Zier. Doch seine Schönheit ist für uns alle sichtbar, wir staunen und bewundern dieses technische Wunderbar.

6.2.9.1.2 xx

Literatur.md#sonstiges-englisch

<https://www.eclipse.org/4diac/>

<https://www.acin.tuwien.ac.at/en/>

<https://www.fortiss.org/>

https://www.eclipse.org/4diac/events/docs/2019/EclipseInsight_SelfConfigurableAutomationSystems.pdf

fortiss GmbH Landesforschungsinstitut des Freistaats Bayern für softwareintensive Systeme

Guerickestr. 25 · 80805 München · Deutschland

<https://www.fortiss.org/ergebnisse/software/4diac>

<https://www.fortiss.org/aktuelles/details/professur-fuer-fortiss-wissenschaftler>

<https://www.jku.at/lit-cyber-physical-systems-lab/ueber-uns/team/univ-prof-di-dr-alois-zoitl/>

6.2.9.1.3 Geschichte:

am 7.7.2007 wurde die erste 4diac Version vom ACIN TU Wien vorgestellt.

davor waren schon ca. 2 Jahre Forschung und Entwicklung

dazwischen wurde 4diac am fortiss, und heute an der JKU entwickelt.

am 7.7.2022 feierte 4diac 15 Jähriges.

6.2.9.1.4 Open Source

https://de.wikipedia.org/wiki/Open_Source

4diac ist dabei komplett Open Source. Das heißt es ist sowohl kostenlos verfügbar, wie auch dass jedermann das Programm selbst weiterentwickeln und anpassen kann.

Details regelt dieser Vertrag:

<https://www.eclipse.org/legal/epl-2.0/>

https://de.wikipedia.org/wiki/Eclipse_Public_License

6.2.9.2 Installation Eclipse 4diac™

Wie man 4 diac installiert

<https://cloud.ms-muc.de/>

/volume1/Gruppenlaufwerke/Schueler/Landmaschinenmechanik/Unterricht/SL/Visuelle-Programmierung/4diac-ide.zip

4diac-ide.zip

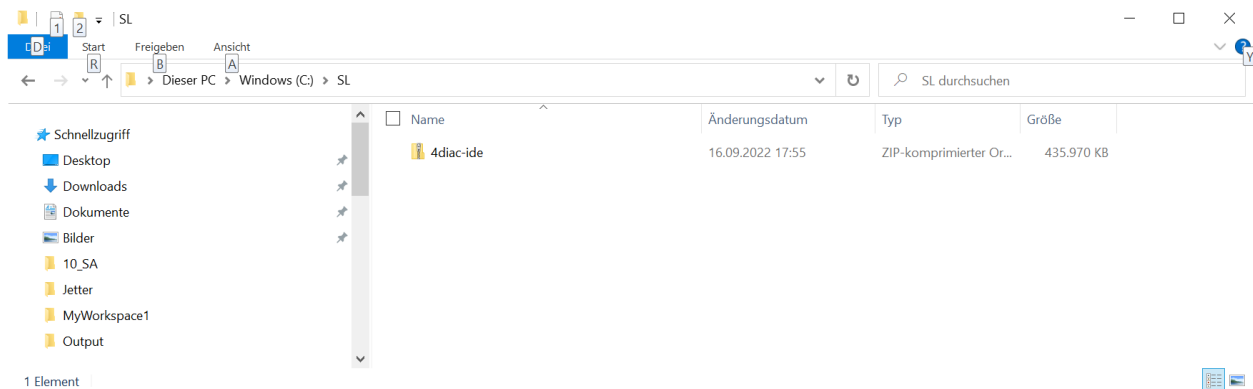
oder: <https://drive.google.com/file/d/14FY9VwpcvyYthvsfPYKd3vDPIA09ZFST/view?usp=sharing>

Idealerweise:


neuer Ordner:

C:\SL\

entpacken:





←  ZIP-komprimierte Ordner extrahieren

Wählen Sie ein Ziel aus und klicken Sie auf "Extrahieren".

Dateien werden in diesen Ordner extrahiert:


C:\S\|

Durchsuchen...

☐ Dateien nach Extrahierung anzeigen

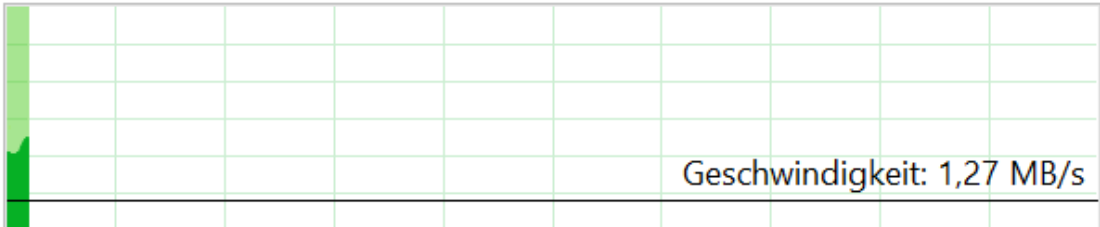
Extrahieren

Abbrechen

 2% abgeschlossen


4.641 Elemente werden von 4diac-ide nach SL kopiert


2% abgeschlossen II ×



Geschwindigkeit: 1,27 MB/s

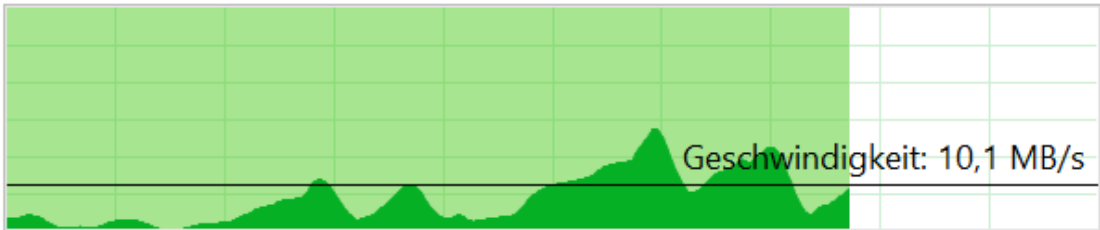
Name: overview16
Restdauer: Ungefähr 2 Minuten
Verbleibende Elemente: 4.432 (480 MB)

 Weniger Details

 77% abgeschlossen


4.641 Elemente werden von 4diac-ide nach SL kopiert

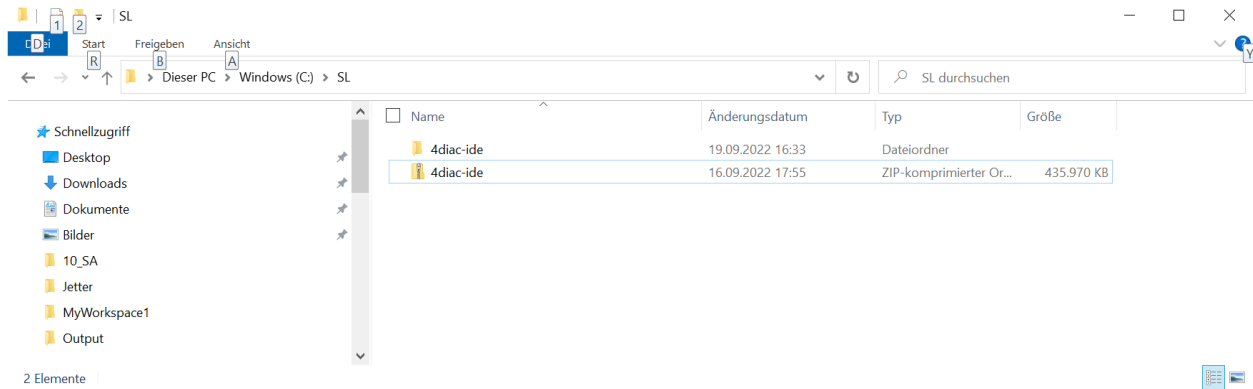
77% abgeschlossen II ×



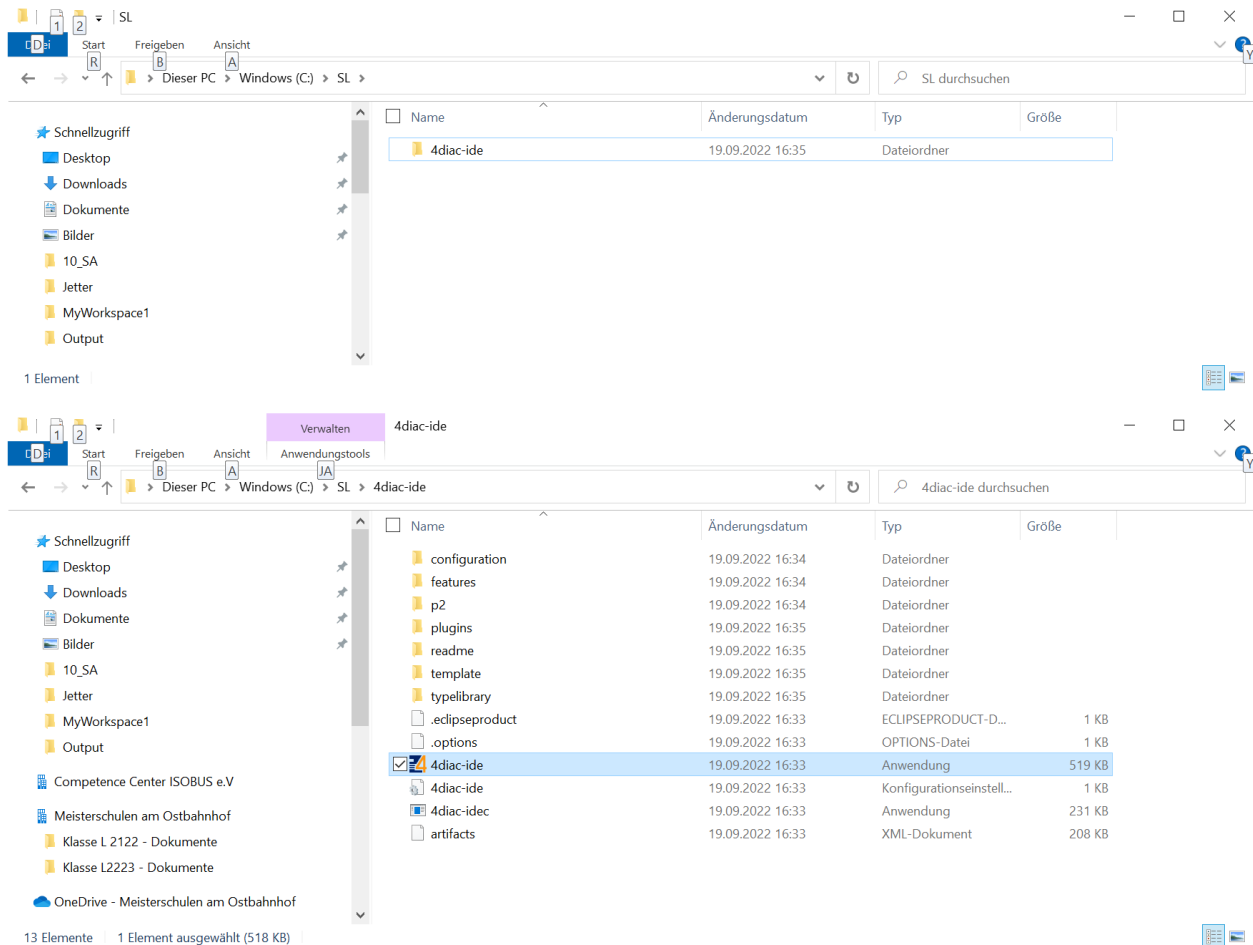
Geschwindigkeit: 10,1 MB/s

Name: org.junit.platform.suite.api_1.7.1.v20210222-1948.jar
Restdauer: Ungefähr 25 Sekunden
Verbleibende Elemente: 1.713 (91,7 MB)

 Weniger Details



jetzt kann man die ZIP Datei löschen wenn man möchte:

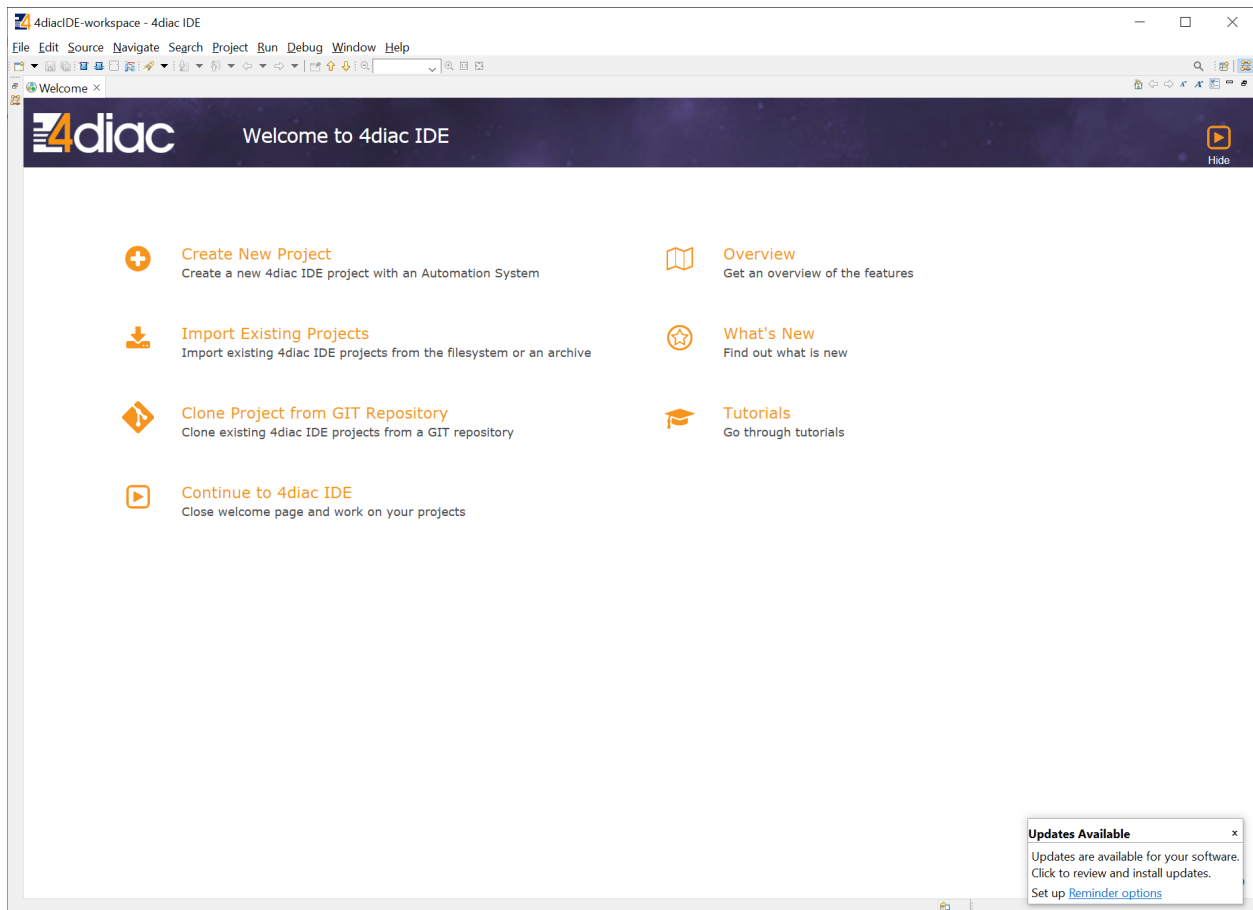
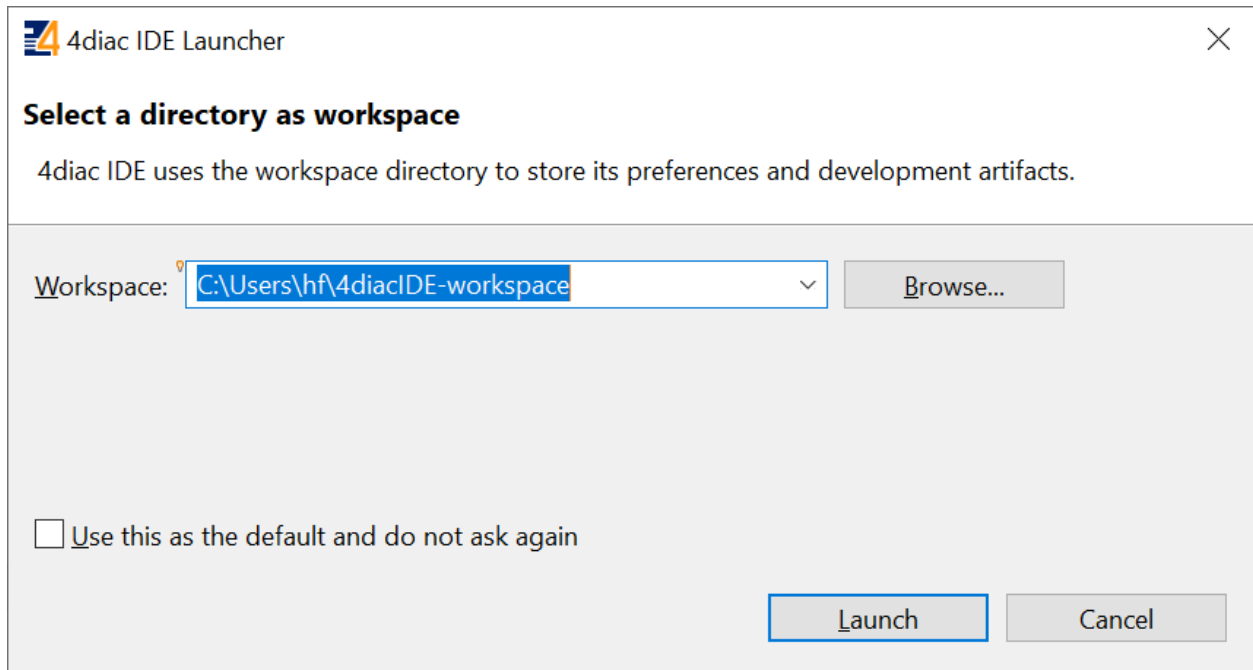


4diac-ide doppelklicken

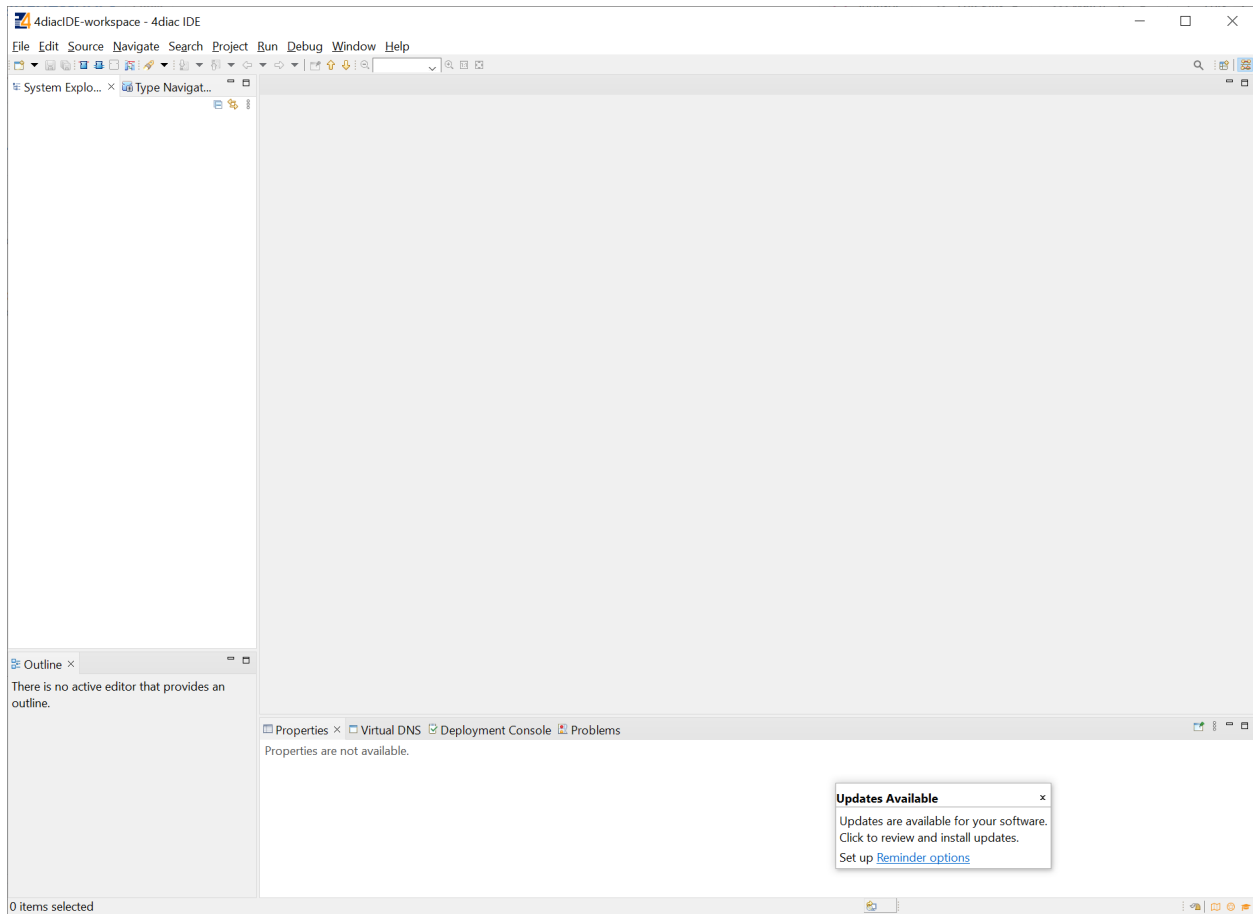




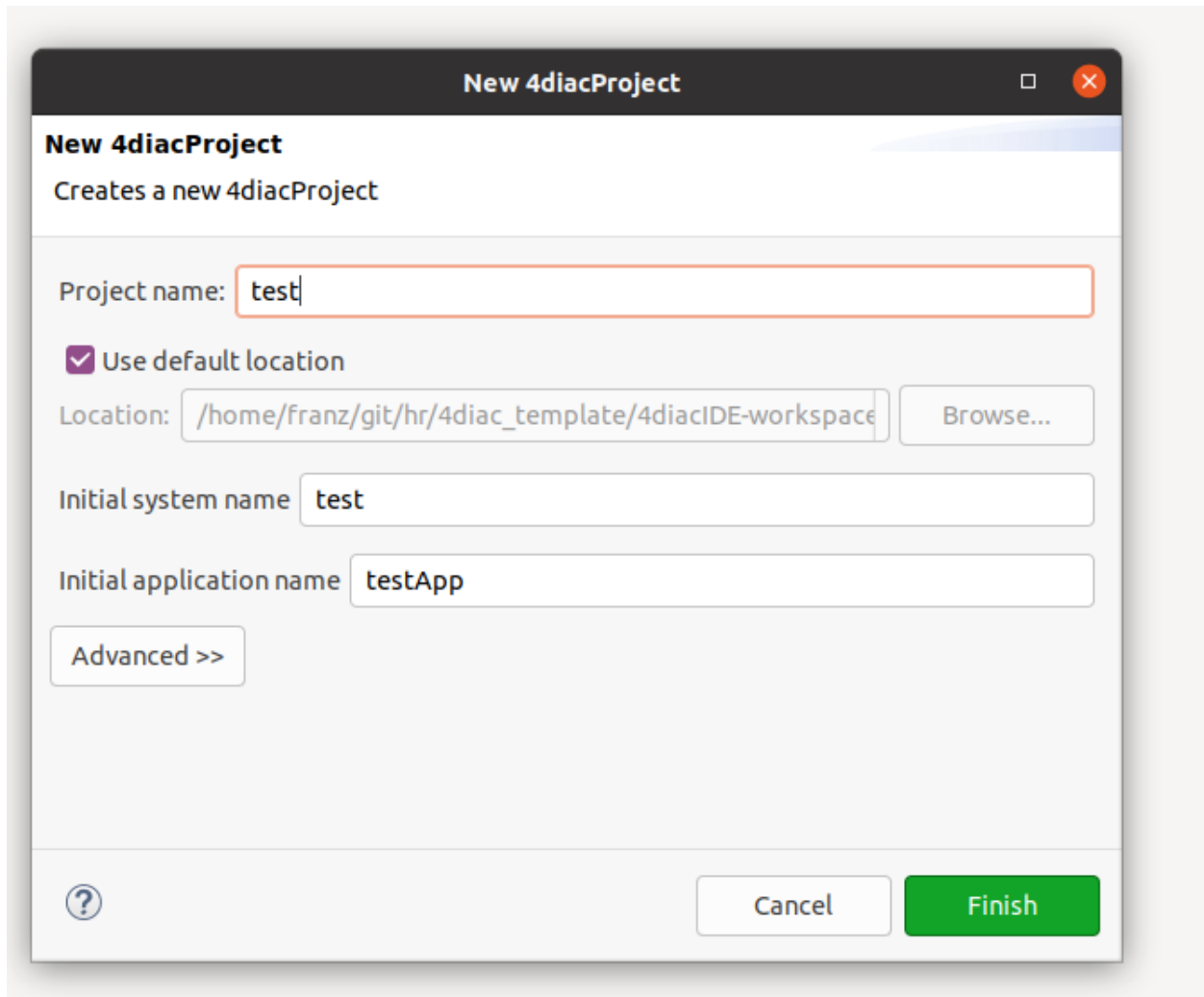
Trotzdem ausführen.

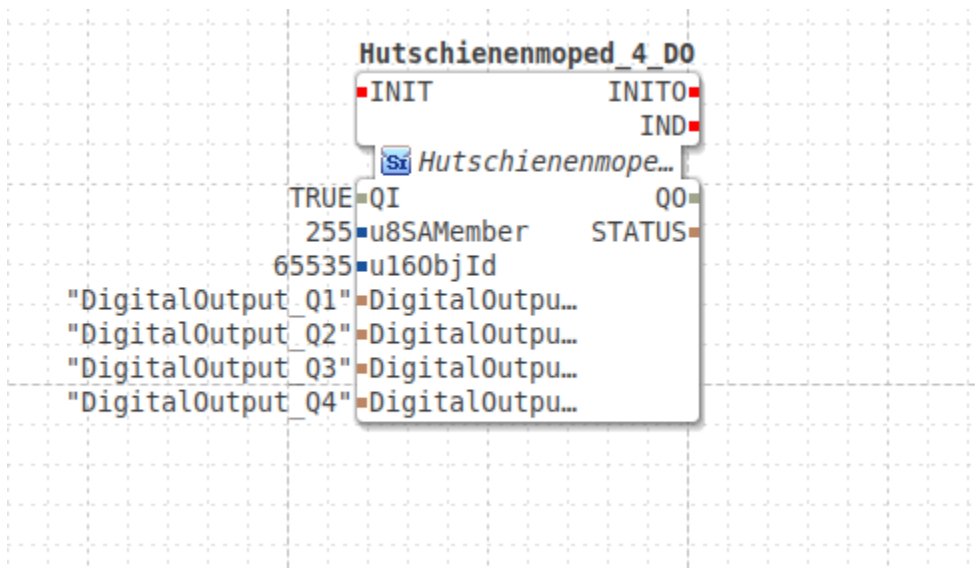
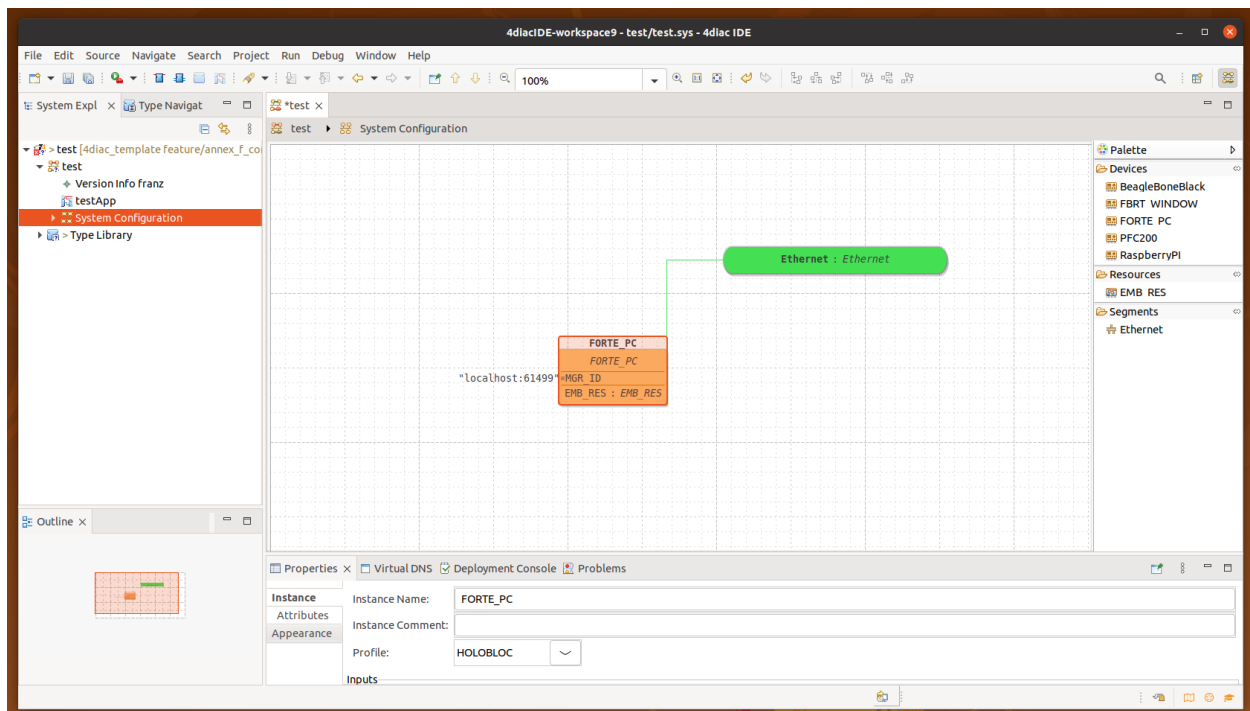


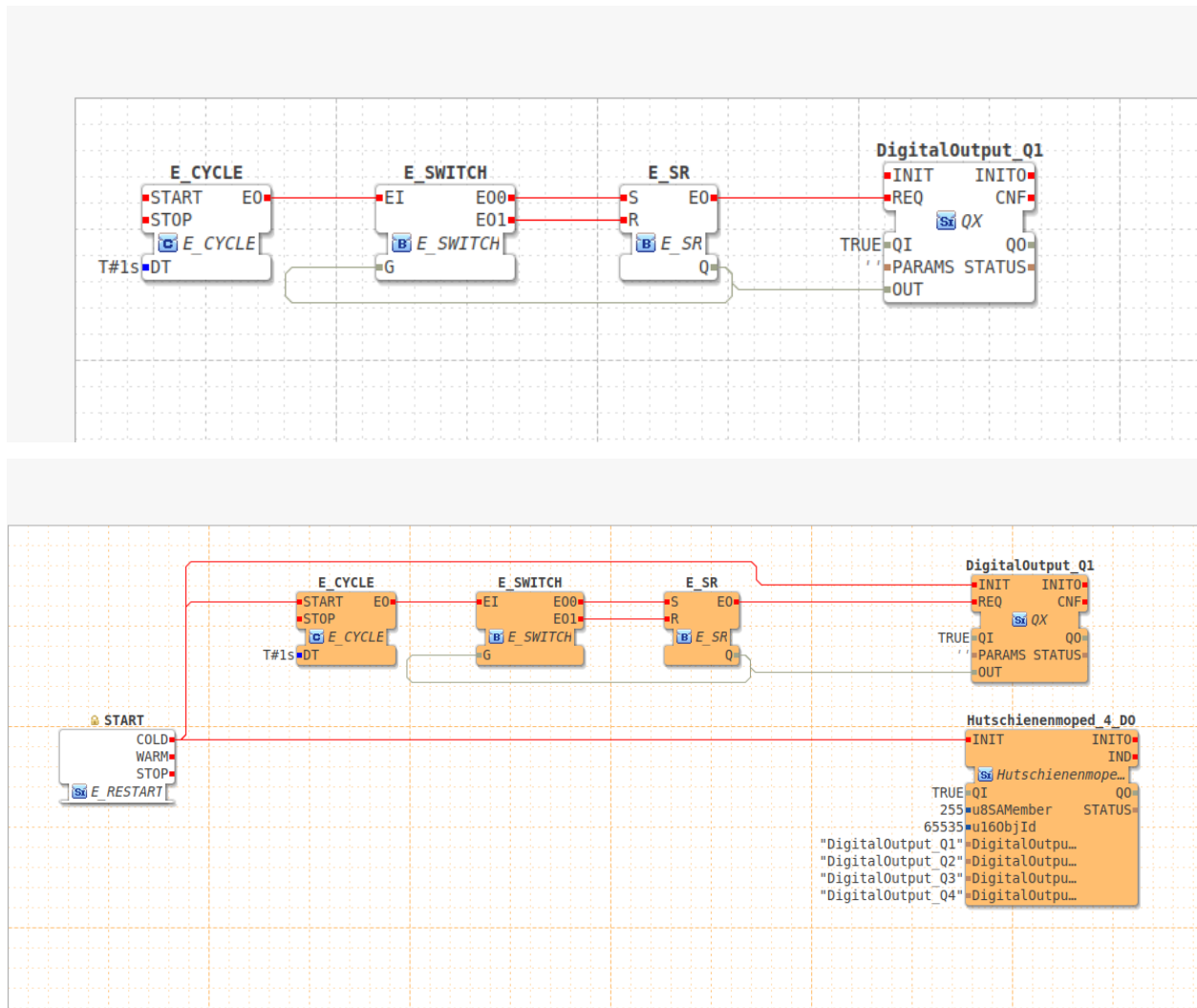
Hide



Als nächstes legen wir ein Projekt an:







Deploy,

Watch

Monitor

6.2.9.3 Eclipse 4diac™ Update

NEU:

seit einiger Zeit finden sich auf allen Update-Seiten unten auch Links mit denen ein komplettes „sauberes“ Paket heruntergeladen werden kann.

Die Links werden teils jede Nacht neu generiert, und sind auch manchmal nicht vorhanden ... das ist normal.

Direktlinks

Betriebssystem	rc	milestone	nightly
*****	*****	*****	*****
win32.win32.x86_64.zip	link	link	link
linux.gtk.x86_64.tar.gz	link	link	link
macosx.cocoa.aarch64.dmg	link	link	link
macosx.cocoa.aarch64.tar.gz	link	link	link
macosx.cocoa.x86_64.dmg	link	link	link
macosx.cocoa.x86_64.tar.gz	link	link	link

Links mit Mirrors

Betriebssystem	rc	milestone	nightly
*****	*****	*****	*****
win32.win32.x86_64.zip	link	link	link
linux.gtk.x86_64.tar.gz	link	link	link
macosx.cocoa.aarch64.dmg	link	link	link
macosx.cocoa.aarch64.tar.gz	link	link	link
macosx.cocoa.x86_64.dmg	link	link	link
macosx.cocoa.x86_64.tar.gz	link	link	link

6.2.9.3.1 Update

In Eclipse 4diac können Sie Updates auf verschiedene Weise durchführen. Hier sind die Schritte, um ein Update durchzuführen:

1. Starten Sie Eclipse 4diac und wählen Sie „Help“ im Menü.
2. Klicken Sie auf „Check for Updates“. Das Update-Management-Fenster wird geöffnet.
3. Klicken Sie auf „Select All“ oder wählen Sie die Updates aus, die Sie installieren möchten.
4. Klicken Sie auf „Next“. Es werden die Lizenzvereinbarungen angezeigt, die Sie akzeptieren müssen, bevor Sie fortfahren können.
5. Klicken Sie auf „Finish“ um den Download und die Installation der Updates zu beginnen.
6. Warten Sie bis der Download und die Installation abgeschlossen sind. Sie werden ggf. aufgefordert, Eclipse 4diac neu zu starten, um die Updates zu aktivieren.

Alternativ, Sie können auch Updates direkt aus der URL herunterladen und installieren.

1. Starten Sie Eclipse 4diac und wählen Sie „Help“ im Menü.
2. Klicken Sie auf „Install New Software...“. Das Installationsfenster wird geöffnet.
3. Geben Sie die Update-URL in das Feld „Work with“ ein und klicken Sie auf „Add“.
4. Wählen Sie die gewünschten Erweiterungen aus und klicken Sie auf „Next“.
5. Folgen Sie den Anweisungen auf dem Bildschirm, um die Installation abzuschließen.

Es ist wichtig zu beachten, dass es manchmal sein kann, dass die Updates nicht kompatibel mit der installierten 4diac Version sind, in diesem Fall müssen Sie ggf. die 4diac Version updaten, bevor Sie das Update durchführen können.

6.2.9.3.2 Update URLs for 4Diac

Update URLs (Uniform Resource Locators) sind Webadressen, die auf einen bestimmten Bereich einer Website verweisen, in dem aktualisierte Versionen einer Software oder Erweiterungen zur Verfügung stehen. In Bezug auf 4diac werden Update URLs verwendet, um die neuesten Versionen der 4diac-Software und Erweiterungen zu verfügbar zu machen. Benutzer von 4diac können diese URLs verwenden, um ihre Installationen von 4diac zu aktualisieren, indem sie die neuesten Updates herunterladen und installieren. Die offizielle Update-URL für 4diac ist <https://download.eclipse.org/4diac/updates/latest/>.

6.2.9.3.2.1 Release

Eine Release Update URL (Uniform Resource Locator) ist eine Webadresse, die auf die neuesten Versionen einer Software oder Erweiterungen verweist, die als offizielle Veröffentlichungen (Releases) betrachtet werden. Im Falle von 4diac, stellt die Release Update URL die neueste stabile Version der 4diac-Software und Erweiterungen zur Verfügung, die von der 4diac-Entwicklergemeinschaft getestet und freigegeben wurden. Benutzer von 4diac können diese URL verwenden, um ihre Installationen von 4diac auf die neueste offizielle Version zu aktualisieren, indem sie die neuesten Updates herunterladen und installieren.

Eclipse 4diac Latest

<https://download.eclipse.org/4diac/updates/latest/>

6.2.9.3.2.2 Release Candidate

Ein Release Candidate (RC) ist eine Vorabversion einer Software, die als bereit für die Veröffentlichung betrachtet wird, aber noch von der Community getestet und überprüft werden muss, bevor sie offiziell veröffentlicht wird. Ein Release-Kandidat-Update-URL für 4diac ist eine Webadresse, die auf einen Bereich einer Website verweist, auf dem die neueste RC-Version von 4diac zum Download zur Verfügung steht.

Benutzer von 4diac können diese URLs verwenden, um die neueste RC-Version von 4diac herunterzuladen und zu testen, bevor sie offiziell veröffentlicht wird. Dies hilft dem Entwicklerteam von 4diac, potenzielle Fehler oder Probleme zu identifizieren und zu beheben, bevor die Software allgemein verfügbar gemacht wird. Ein Beispiel einer Release-Kandidat-Update-URL für 4diac wäre: <https://download.eclipse.org/4diac/updates/rc/>

Es ist jedoch zu beachten, dass Release-Kandidat-Updates möglicherweise nicht so stabil sind wie die offiziellen Versionen und daher für Produktionsumgebungen nicht empfohlen werden.

Eclipse 4diac Release Candidate

<https://download.eclipse.org/4diac/updates/rc/>

6.2.9.3.2.3 Milestone

Eclipse 4diac ist eine open-source Plattform für die Entwicklung von Industrie 4.0 Anwendungen, die auf der Eclipse-Plattform aufbaut. Ein Milestone Build ist eine Zwischenversion, die während des Entwicklungsprozesses erstellt wird, um einen bestimmten Meilenstein oder einen bestimmten Entwicklungsfortschritt zu markieren.

Ein Eclipse 4diac Milestone Build ist somit eine Zwischenversion der Eclipse 4diac-Plattform, die verschiedene neue Funktionen, Verbesserungen und Fehlerkorrekturen gegenüber der vorherigen stabilen Version enthält, aber auch noch nicht vollständig getestet wurde und daher möglicherweise noch einige Fehler enthält. Diese Versionen werden normalerweise an Entwickler und interessierte Anwender verteilt, um Feedback zu sammeln und bei der weiteren Entwicklung der Plattform zu unterstützen.

Eclipse 4diac Milestone

<https://download.eclipse.org/4diac/updates/milestone/>

6.2.9.3.2.4 Nightly Build

Nightly Builds sind tägliche automatisch erstellte Versionen einer Software, die die neuesten Änderungen und Entwicklungen enthalten. Diese Builds werden häufig von Entwicklern genutzt, um die neuesten Funktionen zu testen und Fehler zu finden. Nightly Build Update URLs für 4diac sind Webadressen, die auf die neuesten täglich erstellten Versionen der 4diac-Software verweisen. Diese URLs werden verwendet, um Benutzern die Möglichkeit zu geben, die neuesten Entwicklungen von 4diac zu testen und Feedback zu geben. Nightly Builds sind jedoch nicht als stabile Versionen gedacht und sollten daher nicht in produktiven Umgebungen verwendet werden. Die Nightly Build Update-URL für 4diac ist nicht immer offen zugänglich.

Ich empfehle folgende Vorgehensweise für das Nightly:

1. https://www.eclipse.org/4diac/en_dow.php
2. Update auf RC
3. jetzt erst das Nightly eintragen.

Eclipse 4diac Nightly

<https://download.eclipse.org/4diac/updates/nightly/>

6.2.9.4 Eclipse 4diac™ IDE Icon

finden Sie hier eine Legende: https://eclipse.dev/4diac/en_news.php?s=09#en_news_20210621

Hier die Icons nochmal wiedergegeben, damit Sie diese auch in der Volltextsuche finden.

6.2.10 Runtime

Ein Runtime System ist eine Art von Betriebssystem, das speziell für die Steuerung und Überwachung von eingebetteten Systemen entwickelt wurde. Es wird verwendet, um die Ausführung von Programmen auf einer Steuerungs- oder Prozessoreinheit zu steuern und zu überwachen.

4diac FORTE ist ein offenes Runtime System, das speziell für die Steuerung und Überwachung von industriellen Automatisierungssystemen entwickelt wurde. Es unterstützt die IEC 61499-Standard, ein internationaler Standard für die Entwicklung von Steuerungs- und Regelungssystemen. Mit 4diac FORTE können Entwickler eigene Funktionsbausteine erstellen und diese in ihre Automatisierungsanwendungen integrieren. Es ermöglicht zudem die Verwaltung und Überwachung von Geräten in einem Netzwerk, die Kommunikation zwischen Geräten und die Steuerung der Ausführung von Programmen. 4diac FORTE ist somit ein leistungsfähiges und flexibles Tool für die Entwicklung von Automatisierungsanwendungen in der industriellen Automatisierungstechnik.

6.2.10.1 ISOBUS

ISOBUS ist ein internationaler Standard (ISO 11783) für die Kommunikation zwischen landwirtschaftlichen Maschinen und landwirtschaftlichen Steuergeräten. Der Standard definiert eine gemeinsame Schnittstelle, die es ermöglicht, verschiedene Arten von landwirtschaftlichen Maschinen, wie z.B. Traktoren, Mähdrescher und Pflanzenschutzspritzen, mit verschiedenen Steuergeräten von unterschiedlichen Herstellern zu verbinden. Dies ermöglicht eine einfachere Bedienung und eine höhere Effizienz bei der Steuerung und Überwachung der Maschinen.

6.2.10.1.1 SoftKey Reduction

Die ISO 11783-6 legt fest, dass landwirtschaftliche Steuergeräte in der Lage sein sollten, sich an unterschiedliche Displays anzupassen. Eine wichtige Anforderung dabei ist die Unterstützung von Softkey-Reduktion.

Softkeys sind Tasten auf einem Display, die dem Benutzer schnellen Zugriff auf häufig verwendete Funktionen ermöglichen. Die meisten Displays haben in der Regel 12 Softkeys, aber es gibt auch Displays mit weniger Softkeys, wie z.B. 6, 8 oder 10. Um sich an diese unterschiedlichen Displays anzupassen, muss das Steuergerät in der Lage sein, die Anzahl der angezeigten Softkeys zu reduzieren, um sicherzustellen, dass alle Funktionen auf dem Display angezeigt werden können.

Die Notwendigkeit der Softkey-Reduktion in der ISO 11783-6 liegt darin, dass es unterschiedliche Displays mit unterschiedlichen Anforderungen in Bezug auf die Anzahl der Softkeys gibt. Durch die Unterstützung von Softkey-Reduktion kann sichergestellt werden, dass das Steuergerät auf jedem Display ordnungsgemäß funktioniert und der Benutzer alle Funktionen bequem erreichen kann.

In einer beispielhaften Softkeymask, mit 12 Softkeys. Eine Strategie kann wie folgt sein: gegeben seien 12 Softkeys (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11) wobei die Anordnung derer auf den meisten Displays so ist, dass in der rechten Spalte 0 bis 5, und in der linken Spalte 6 bis 11 sind. Es liegen sich also gegenüber: 0 und 6, 1 und 7, 2 und 8, 3 und 9, 4 und 10 sowie 5 und 11. Um nun diese Paare beizubehalten löscht man bei der Reduktion von 12 auf 10 Keys das Paar 5/11, bei der weiteren Reduktion von 10 auf 8 Keys das Paar 4/10, bei der weiteren Reduktion von 8 auf 6 Keys das Paar 3/9.

Ein Beispiel für eine Softkey-Maske mit 12 Softkeys könnte wie folgt aussehen:

- Softkey 0: Funktion A
- Softkey 1: Funktion B
- Softkey 2: Funktion C
- Softkey 3: Funktion D
- Softkey 4: Funktion E
- Softkey 5: Funktion F
- Softkey 6: Funktion G
- Softkey 7: Funktion H
- Softkey 8: Funktion I
- Softkey 9: Funktion J
- Softkey 10: Funktion K
- Softkey 11: Funktion L

Eine mögliche Strategie für die Softkey-Reduktion könnte sein, die Softkeys paarweise zu behalten und dann jeweils das letzte Paar zu entfernen, wenn die Anzahl der Softkeys reduziert wird.

	6	0
	7	1
	8	2
	9	3
	10	4
	11	5

Bei einer Reduktion von 12 auf 10 Softkeys würde man das Paar 5/11 entfernen, wobei die Softkeys 0 bis 4 und 6 bis 10 verbleiben.

	6	0
	7	1
	8	2
	9	3
	10	4

Bei einer weiteren Reduktion von 10 auf 8 Softkeys würde man das Paar 4/10 entfernen, wobei die Softkeys 0 bis 3 und

6 bis 9 verbleiben.

6	0
7	1
8	2
9	3

Bei einer weiteren Reduktion von 8 auf 6 Softkeys würde man das Paar 3/9 entfernen, wobei die Softkeys 0 bis 2 und 6 bis 8 verbleiben.

6	0
7	1
8	2

Diese Strategie ermöglicht es, die wichtigsten Funktionen beizubehalten, während die Anzahl der Softkeys reduziert wird, und sicherzustellen, dass die Funktionen auf dem Display in einer logischen und einfach zu verwendenden Anordnung angezeigt werden.

die gezeigten Beispiele befinden sich hier:

[4diac-SoftkeyReduction-sample](#)

6 key

8 key

10 key

12 key

6.2.11 Das erste Projekt

6.2.11.1 Video

Neuster Stand von 4diac, logiBUS®, Hutschienenmoped.

6.2.11.2 4Diac neues Projekt anlegen

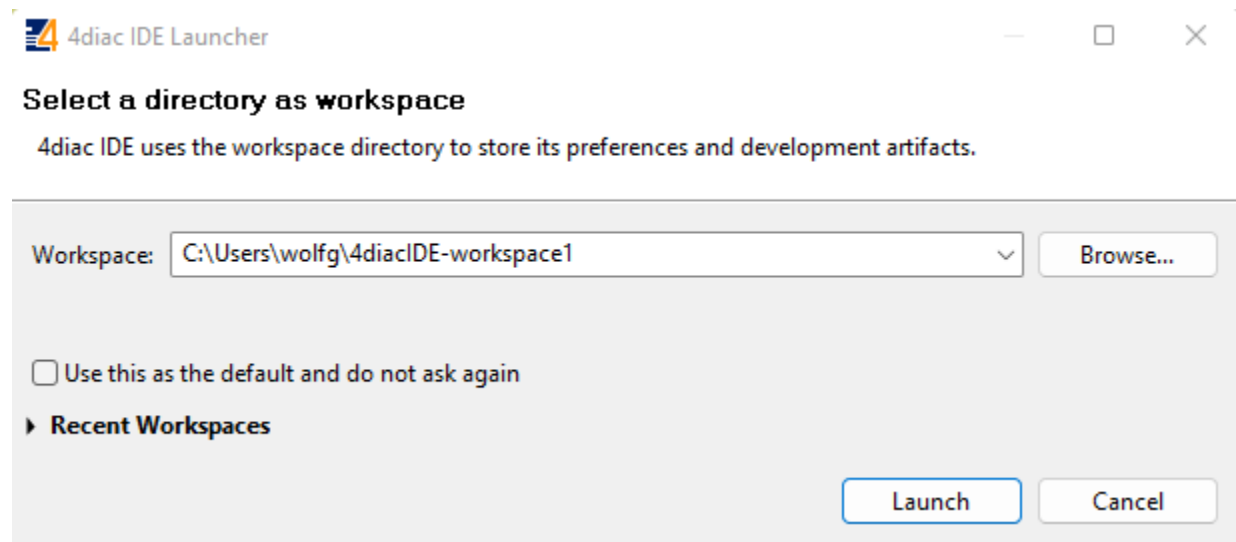
Grundverständnis von 4Diac

4Diac besteht aus mehreren Teilen (3)

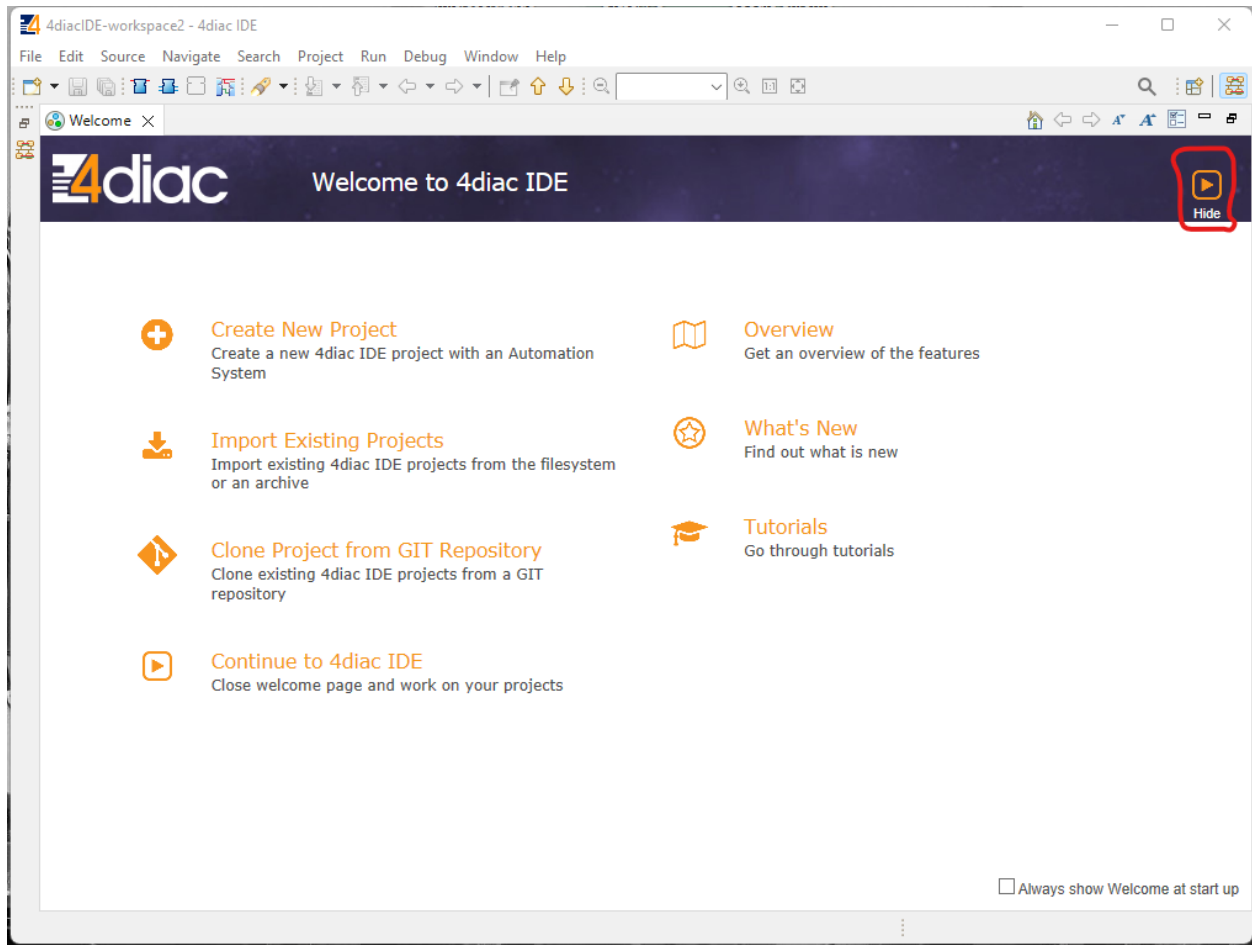
1. IDE = Programmierung am PC
2. FORTE = Programm auf Zielgerät (Hutschienenmoped)
3. Type Library = Bausteinbibliothek (Teil in FORTE und Teil in IDE) enthalten.

Neues Projekt starten bzw. anlegen

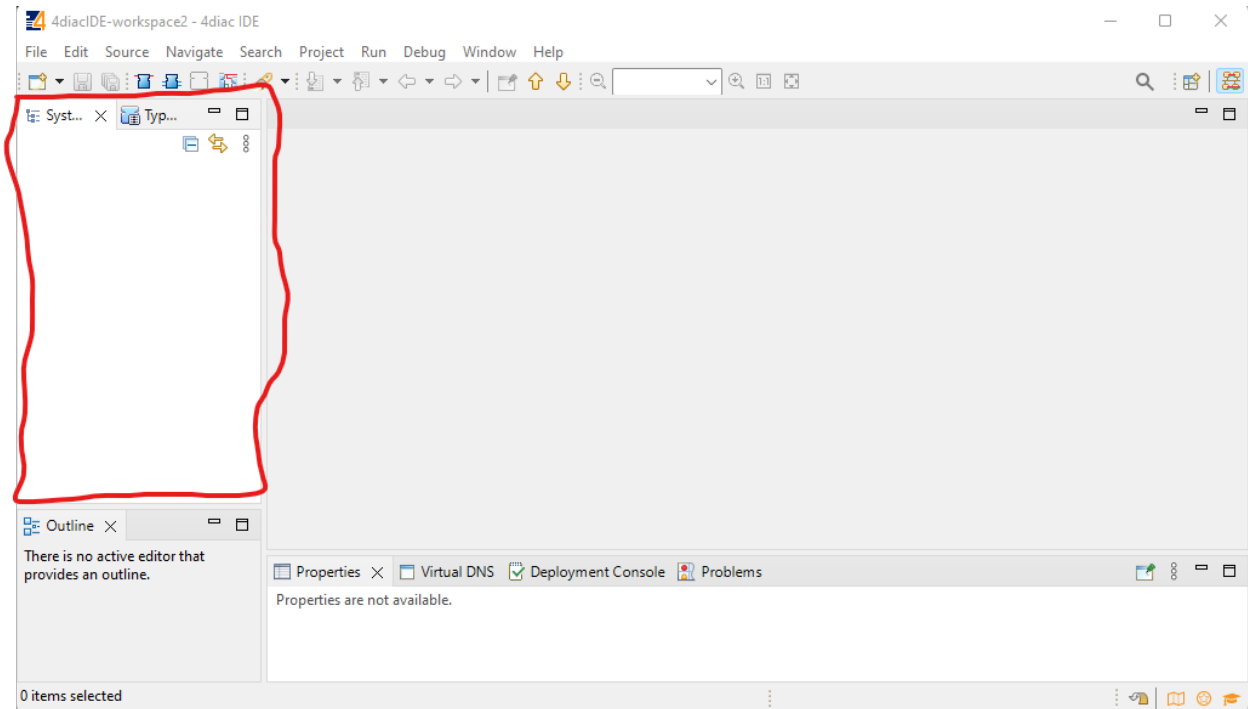
1. 4diac-ide (Desktopverknüpfung) starten. -> Fenster mit Workspace öffnet.
2. Workspace auswählen oder neu erstellen. (Jeder Workspace-namen kann nur 1x verwendet werden, notfalls Benennung mit Zahl 1... ergänzen)



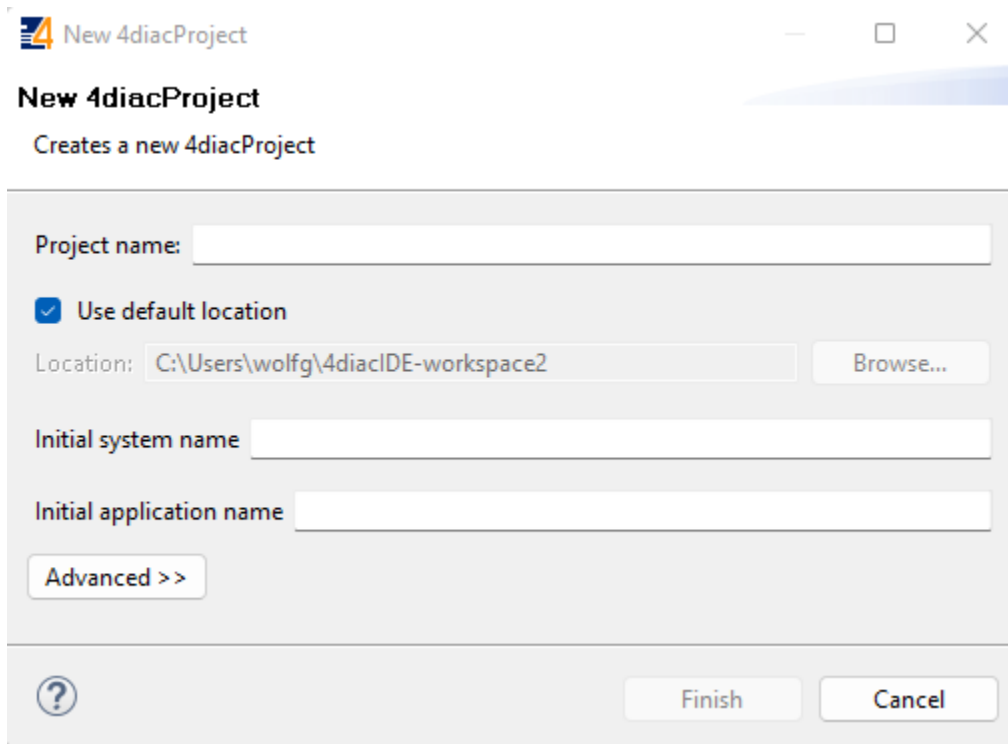
3. Workspace mit Launch bestätigen. -> Workspace wird geöffnet wenn ein schon vorhandener ausgewählt wird. Wird ein neuer erstellt öffnet die 4diac Startseite.



4. Zur Erstellung eines neuen Projekts ganz oben rechts den im Bild markierten Button Hide klicken. -> Ein neuer Workspace öffnet.

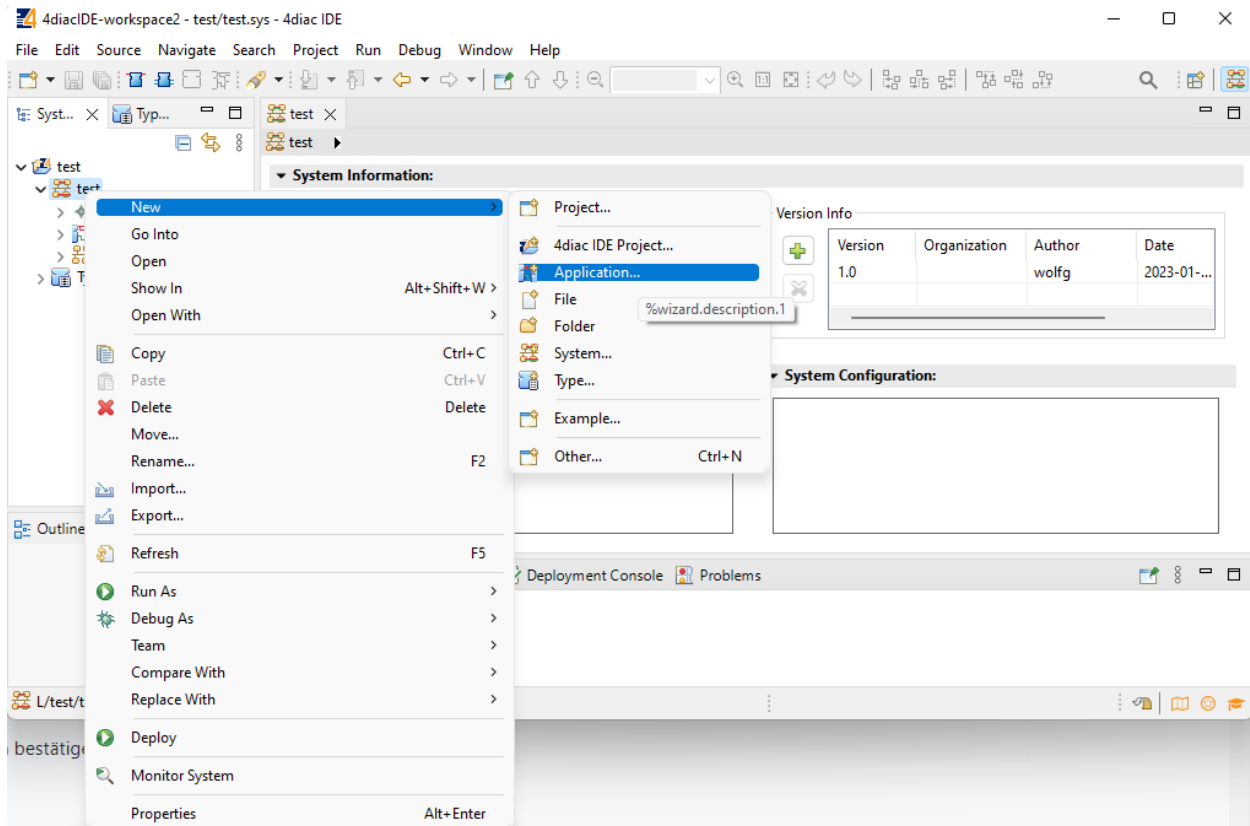


5. Zum erstellen eines neuen Projekts in das markierte Feld im Bild einen Rechtsklick machen. -> Fenster öffnet, im Fenster New wählen -> dann 4diac IDE Project... wählen.



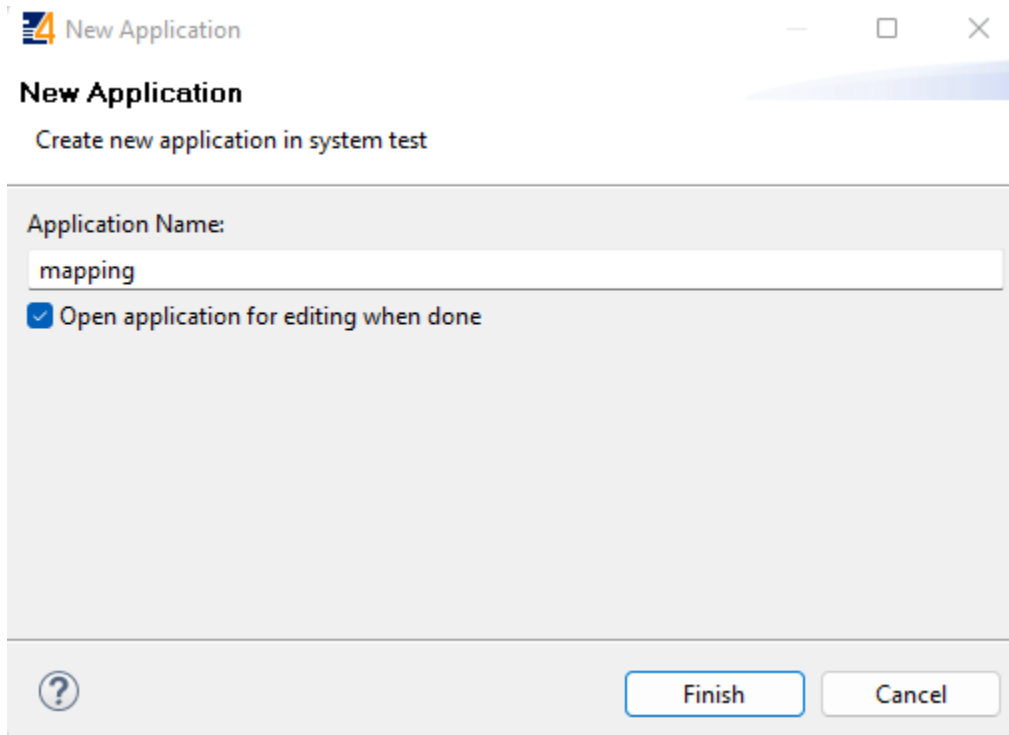
6. Project name eintragen. (Das Programm muss „test“ heißen, wenn die Steuerung immer wieder von selbst „starten“ soll wenn das Hutschienenmopfed verbunden wird.)

7. Nach Eintrag des Namens mit Finish bestätigen. -> Neues Projekt erscheint in Auswahlbaum links „test“



8. In Auswahlbaum links Projekt „test“ öffnen, -> „test“ wird aufgeklappt. Auf unteren „test“ Rechtsklick -> Fenster öffnet. New wählen -> weiteres Fenster öffnet. Application... wählen.

-> Fenster mit New Application öffnet.



9. Application Name mapping vergeben und mit Finish bestätigen. -> Application wurde angelegt.

Es kann mit dem mapping begonnen werden. Weiteres Vorgehen in File Neue Steuerung erstellen.

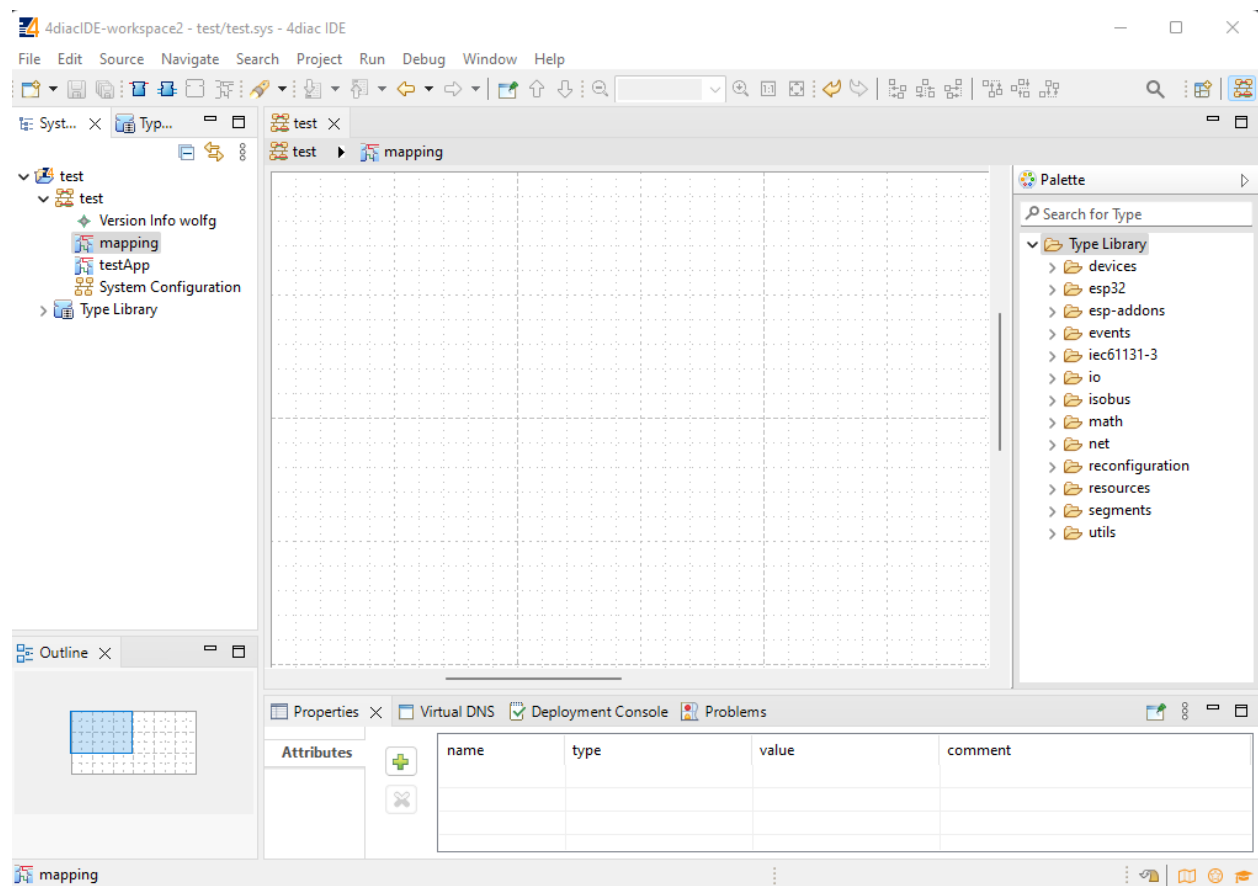
6.2.11.3 Steuerung Bohrmaschinenschalter erstellen

Neue Steuerung erstellen Bohrmaschinenschalter. (Ein- Ausschalter an Ständerbohrmaschine)

(Mit einem Knopf wird ein Ausgang an geschalten, mit einem anderen Knopf wird ein Ausgang aus geschalten)

Type Library auf der rechten Seite ist Schreibgeschützt, deshalb sollte diese anstatt der linken verwendet werden.

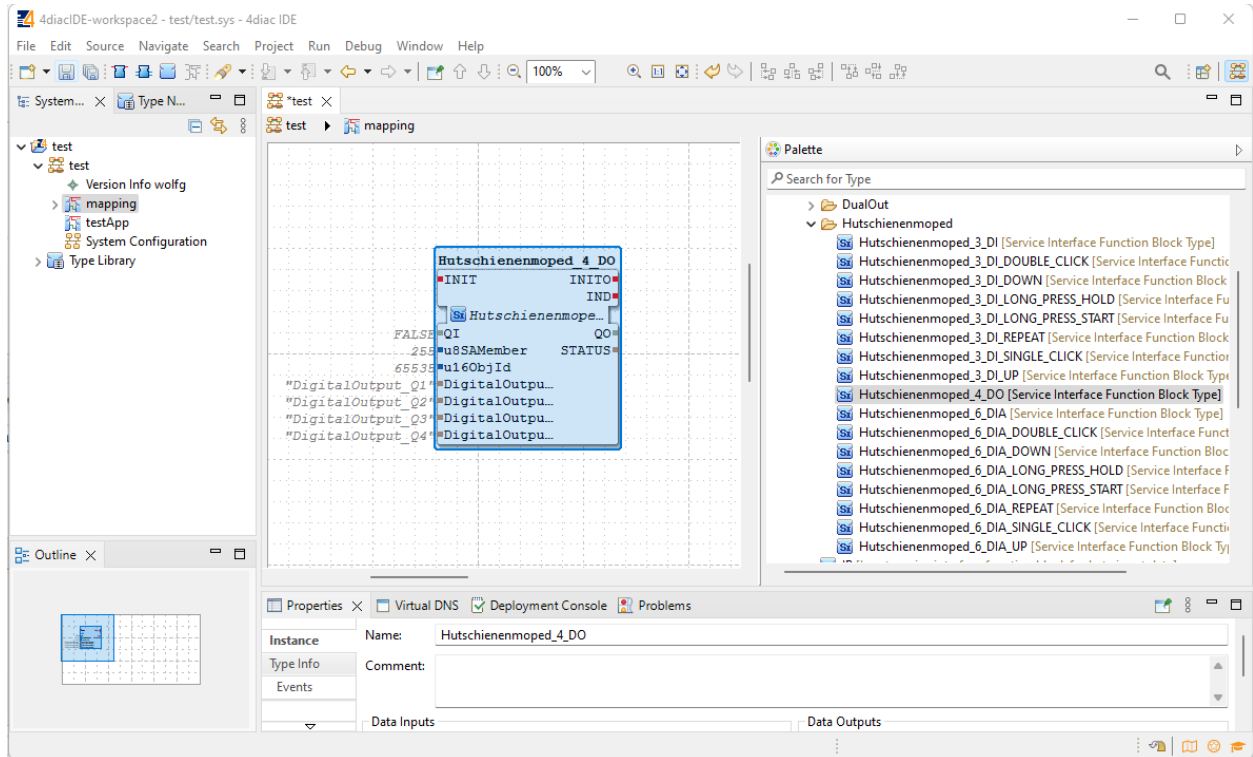
1. Mapping in Baum links mit Doppelklick wählen.



2. Type Library öffnen. Ordner io öffnen. Ordner Hutschienenmoped öffnen. Baustein „Hutschienenmoped_4_DO“ mit Maus in Arbeitsfenster ziehen.

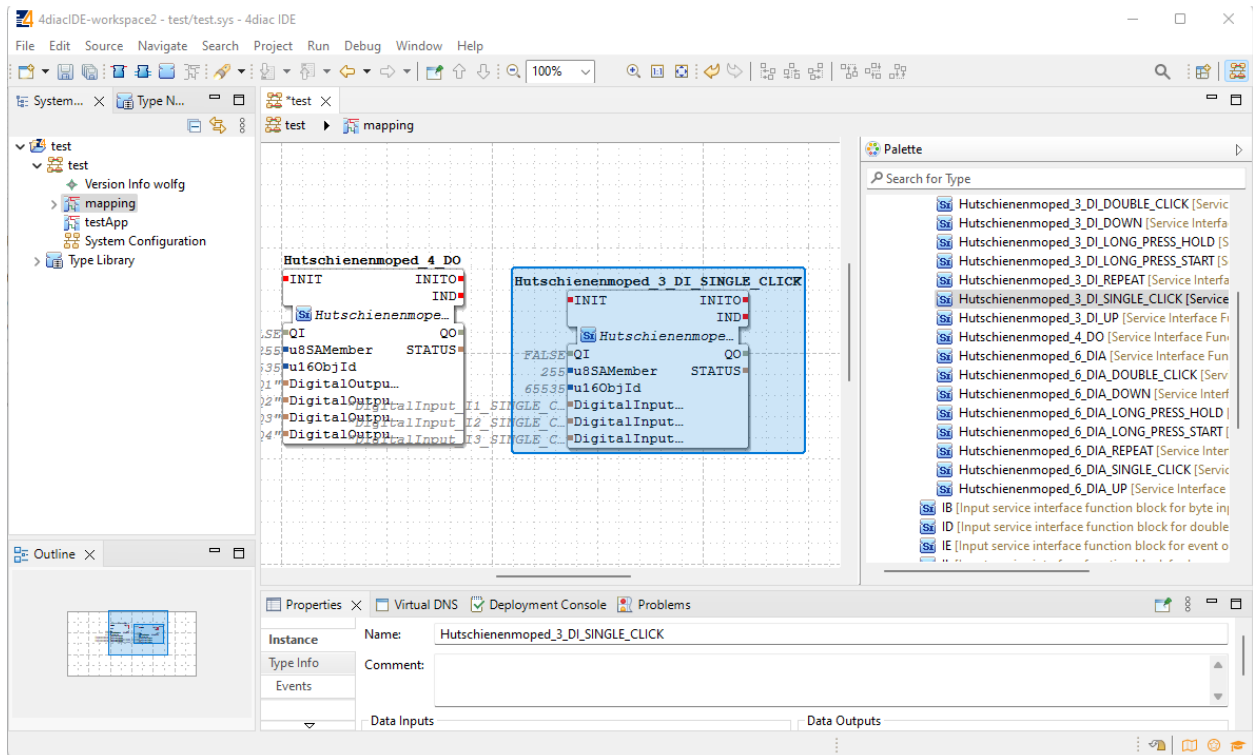
{Hutschienenmoped mit 4 Ausgängen DigitalOutput Q1 - Q4 (Lampen)}

-> Baustein wird eingefügt.



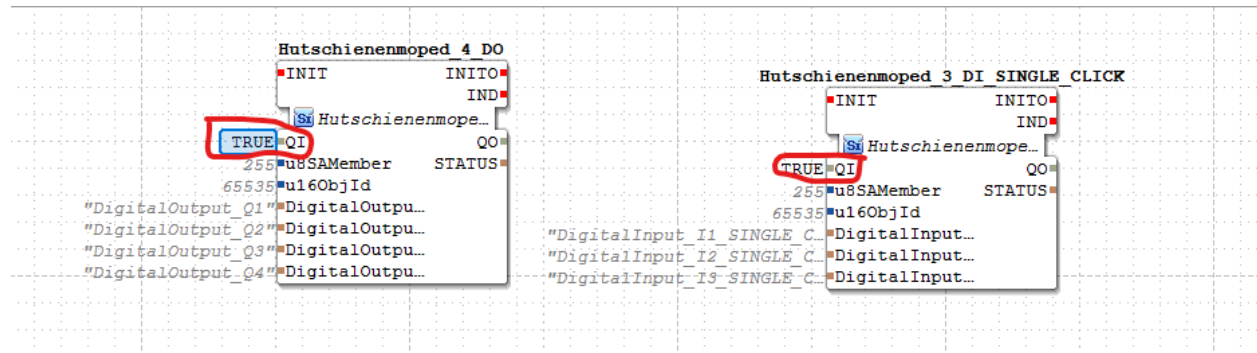
3. Type Library öffnen. Ordner io öffnen. Ordner Hutschienenmoped öffnen. Baustein „Hutschienenmoped_3_DI_SINGLE_CLICK“ mit Maus in Arbeitsfenster ziehen (zusätzlich)

(Schalter mit 3 Eingängen DigitalInput I1 - I3 {SingleClick Schalter blau, rot und Atom})



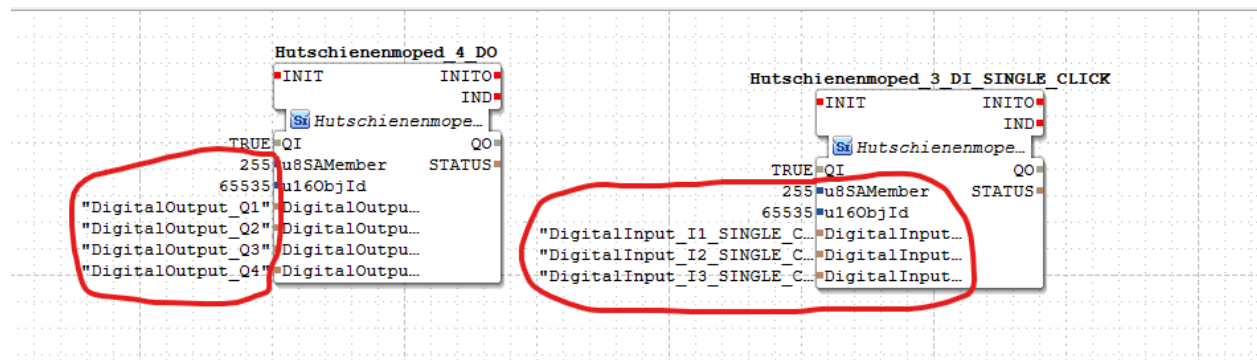
4. Damit die Bausteine nach dem Start funktionieren muss jeweils QI auf TRUE gesetzt werden.

Doppelklick auf FALSE und TRUE eingeben.



5. Damit die Ein- und Ausgänge funktionieren, müssen die grau hinterlegten Bezeichnungen angeklickt (Doppelklick) und mit Enter bestätigt werden.

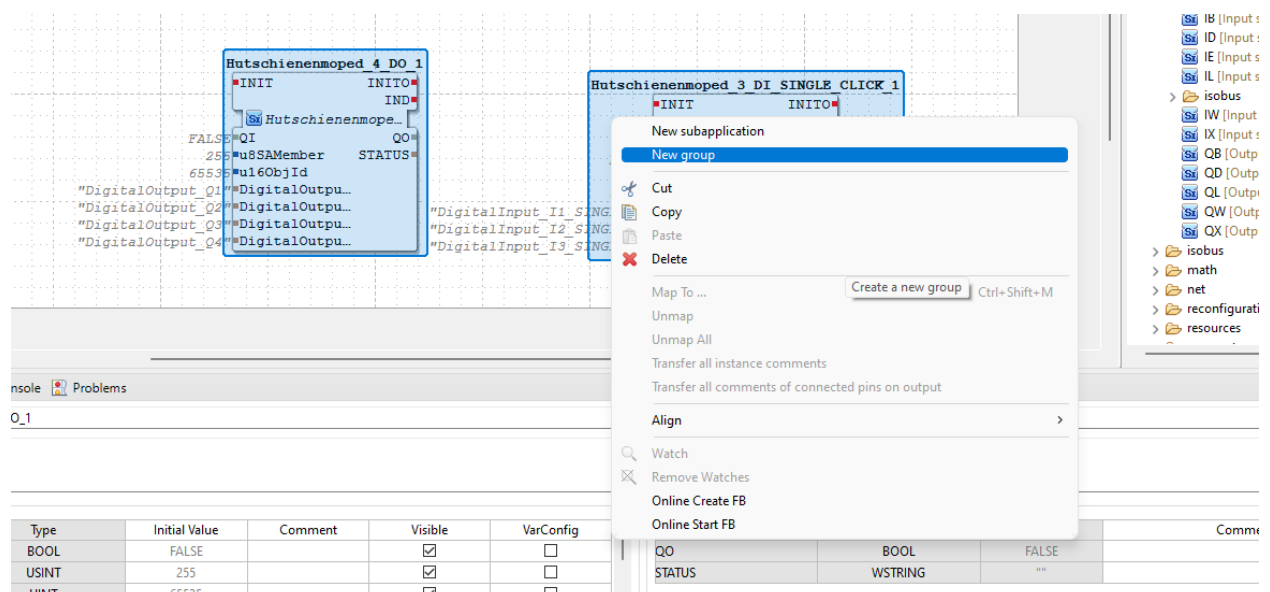
-> Die Schriftfarbe ändert sich dann zu schwarz.



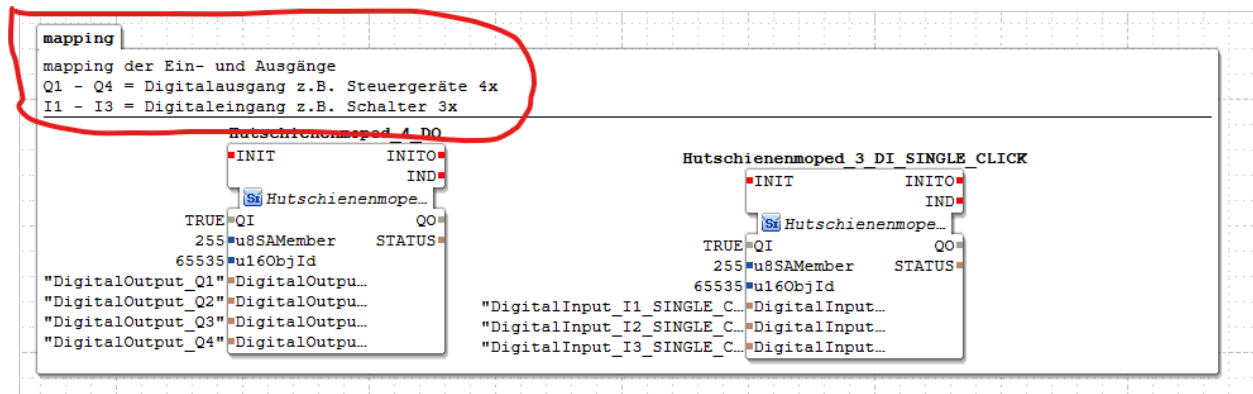
6. Die Bausteine können zu einer Gruppe zusammengefügt werden.

Dazu mit gedrückter linker Maustaste ein Fenster über die Bausteine aufziehen und loslassen. Beide Bausteine werden blau. Rechten Mausklick auf einen der gewählten Bausteine.

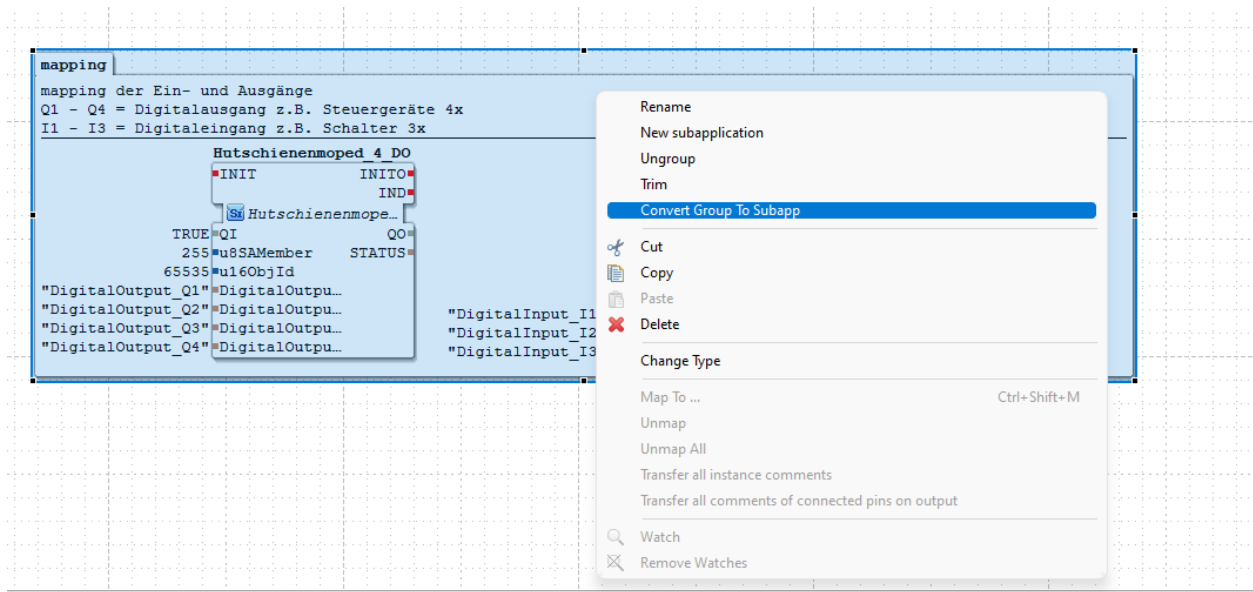
-> Fenster öffnet. New Group wählen. -> Bausteine werden zu einer Gruppe zusammengefügt.



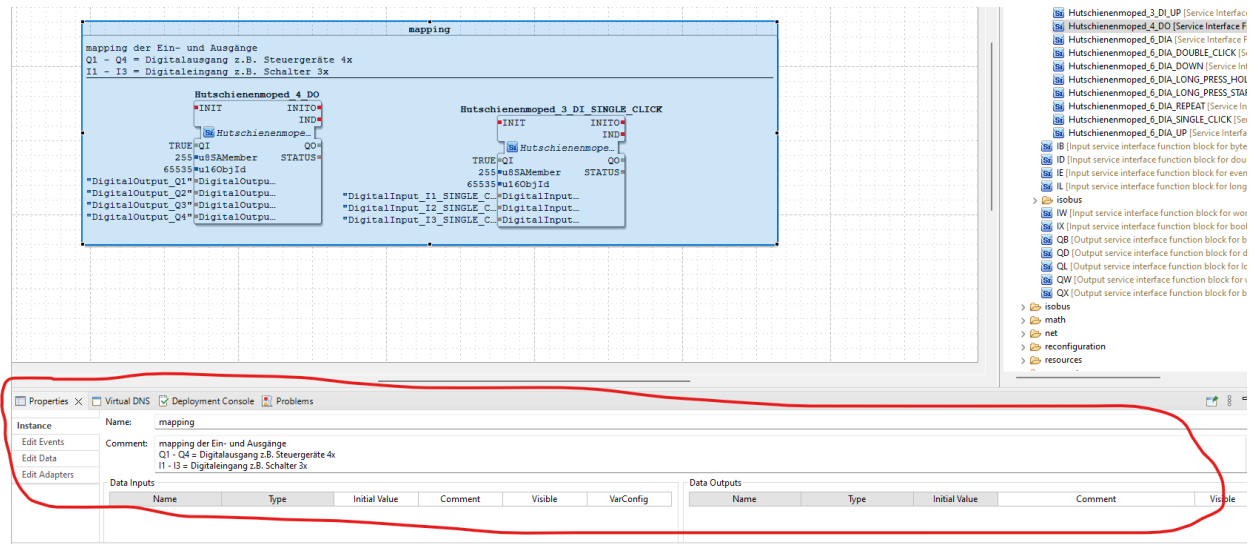
7. Gruppennamen vergeben (mapping) und Gruppenbeschreibung bei Comment eingeben (mapping der Ein- und Ausgänge....)



8. Gruppe anwählen -> wird blau. Rechtsklick -> Fenster öffnet. Convert Group to Subapp wählen.



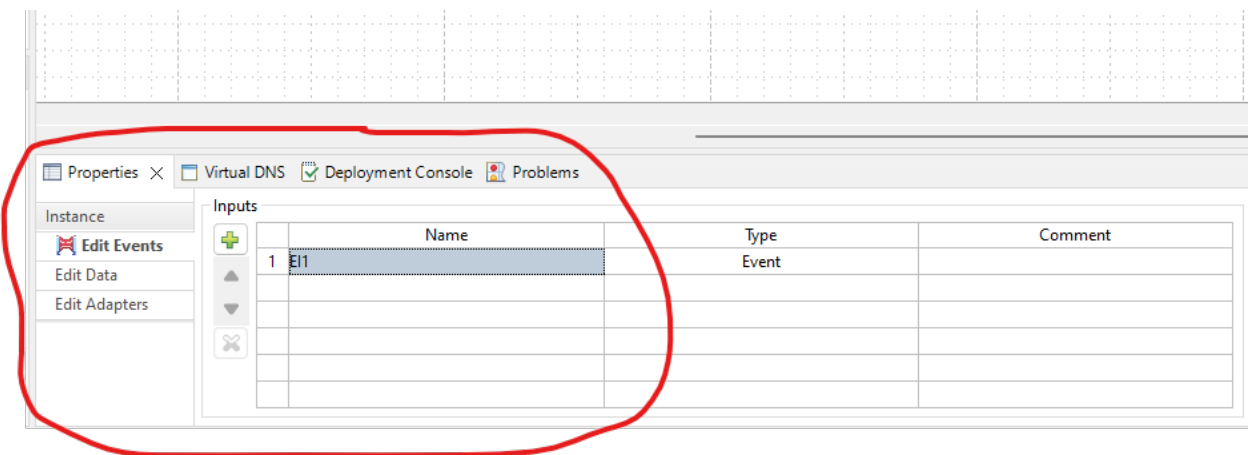
-> Untere Leiste bei Properties ändert sich.



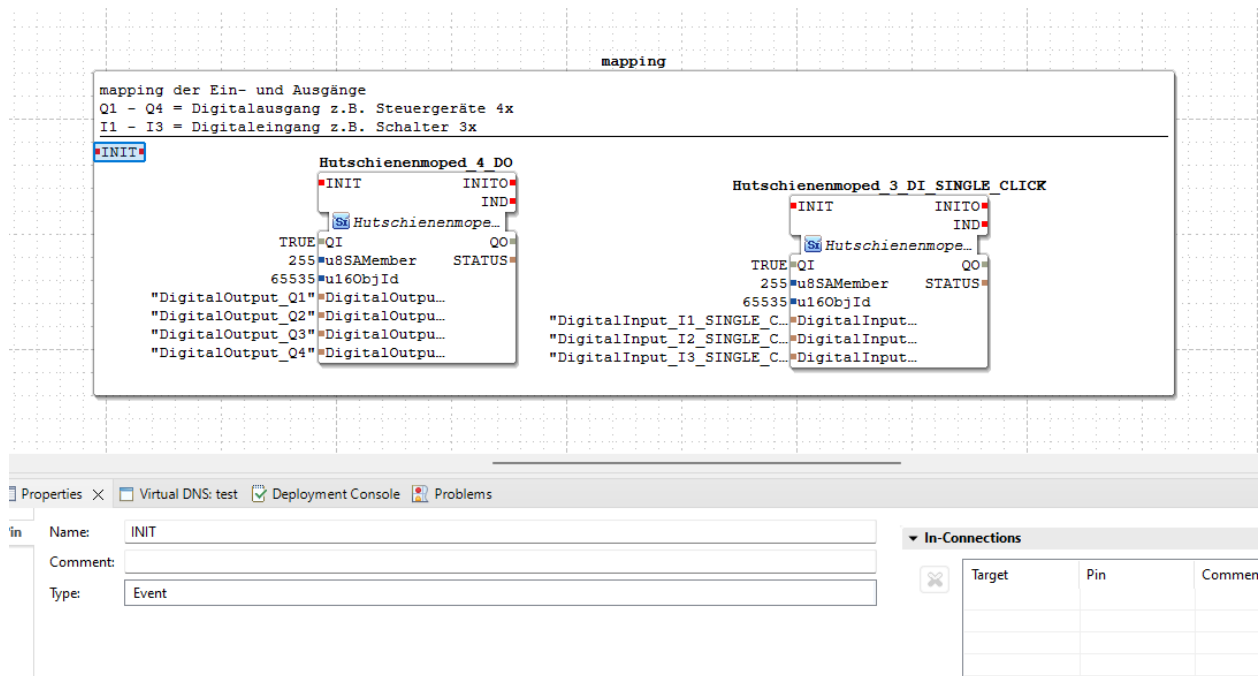
9. Properties anwählen. Edit Events anwählen. Rechts daneben in Leiste öffnen sich Inputs und Outputs mit einem grünen Plus daneben.

+ auf der Seite Inputs drücken 1x.

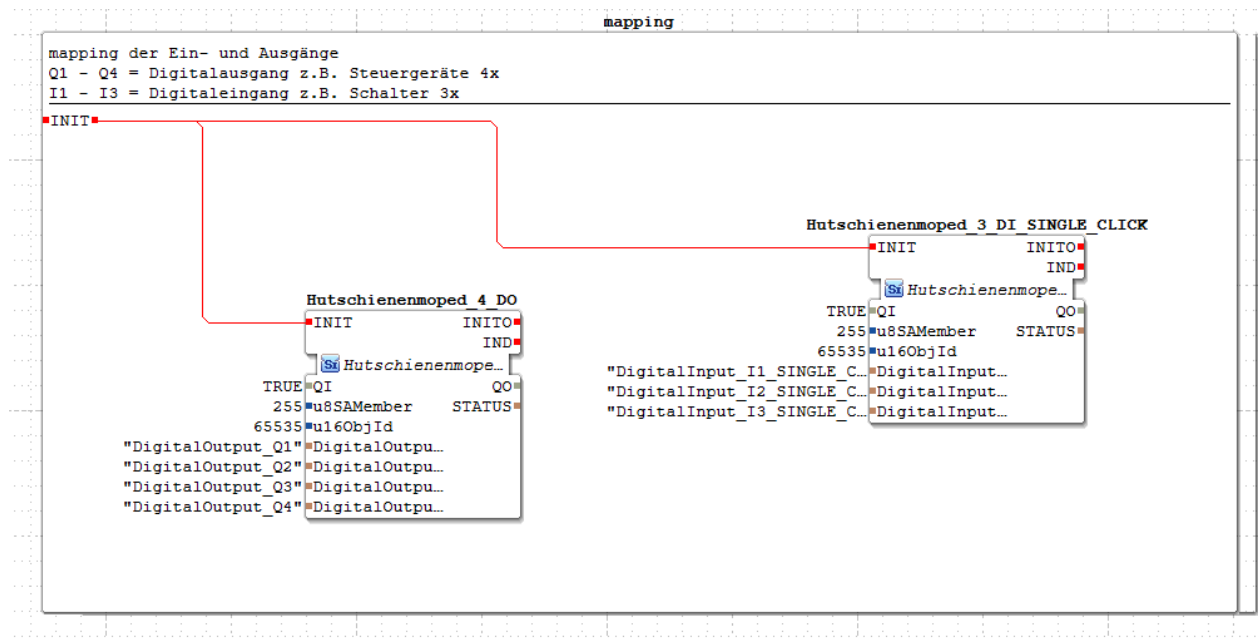
-> Ein neuer Input EI1 wird in der Subapp hinzugefügt. Sichtbar in Tabelle und oben links in der Gruppe.



10. Doppelklick auf EI1 und Name INIT eingeben. -> Statt EI1 wird INIT bei Name und oben in Gruppe angezeigt.



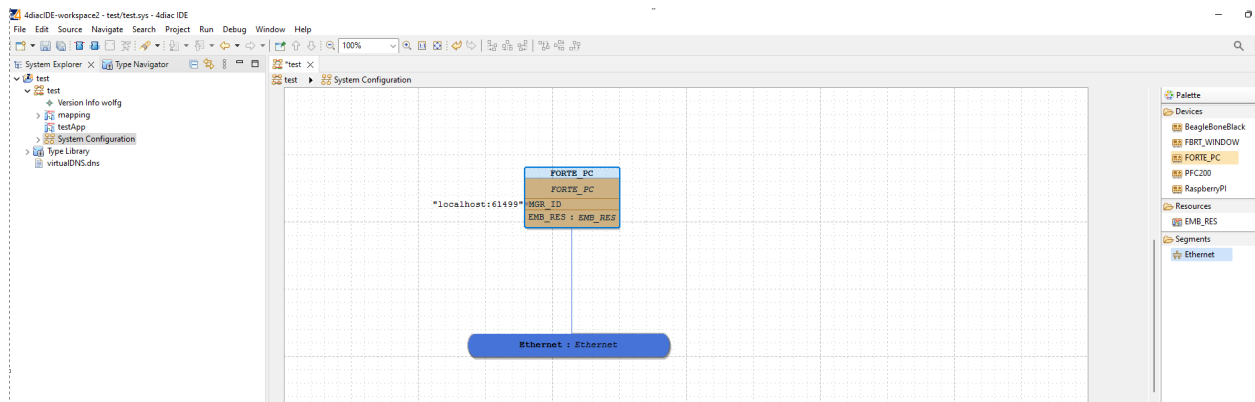
11. Alle INIT Events miteinander verbinden.



12. System Configuration anlegen.

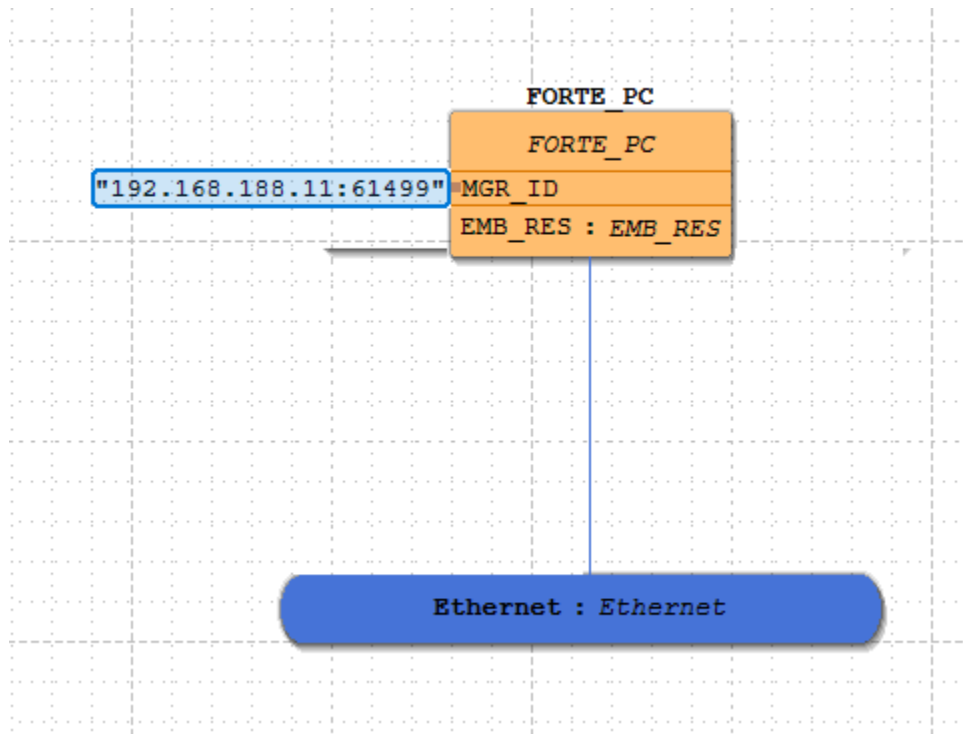
In Baum links test öffnen. System Configuration mit Doppelklick anwählen. Aus Baum rechts FORTE_PC und Ethernet in Arbeitsbildschirm ziehen.

FORTE_PC und Ethernet miteinander verbinden.

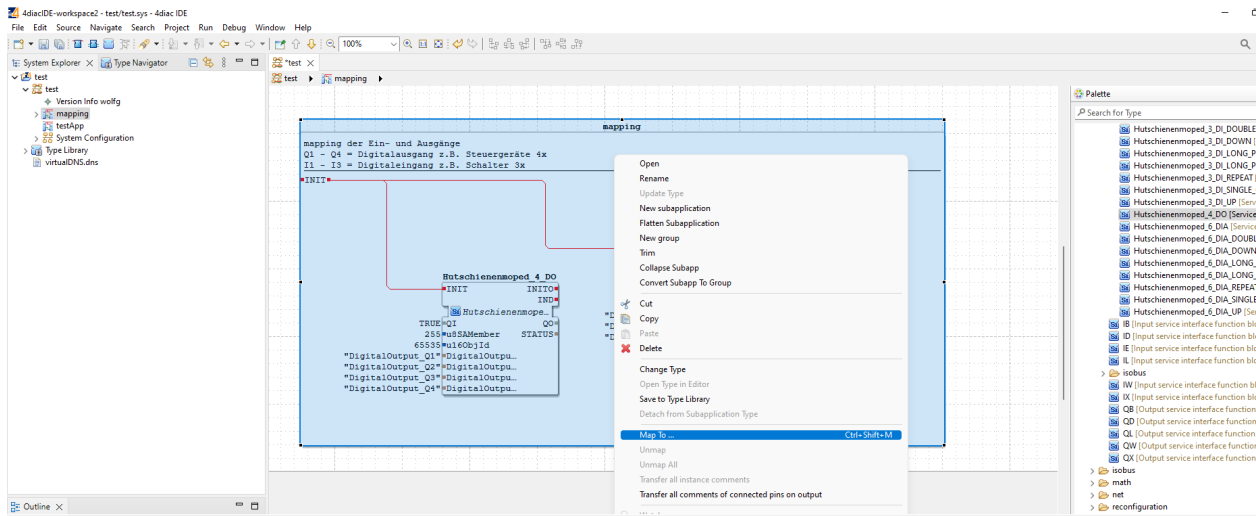


13. „localhost:61499“ mit Doppelklick anwählen und umbenennen in „IP auf Moped:61499“ aus Beispielbild mit L11 („192.168.188.11:61499“) mit Enter bestätigen.

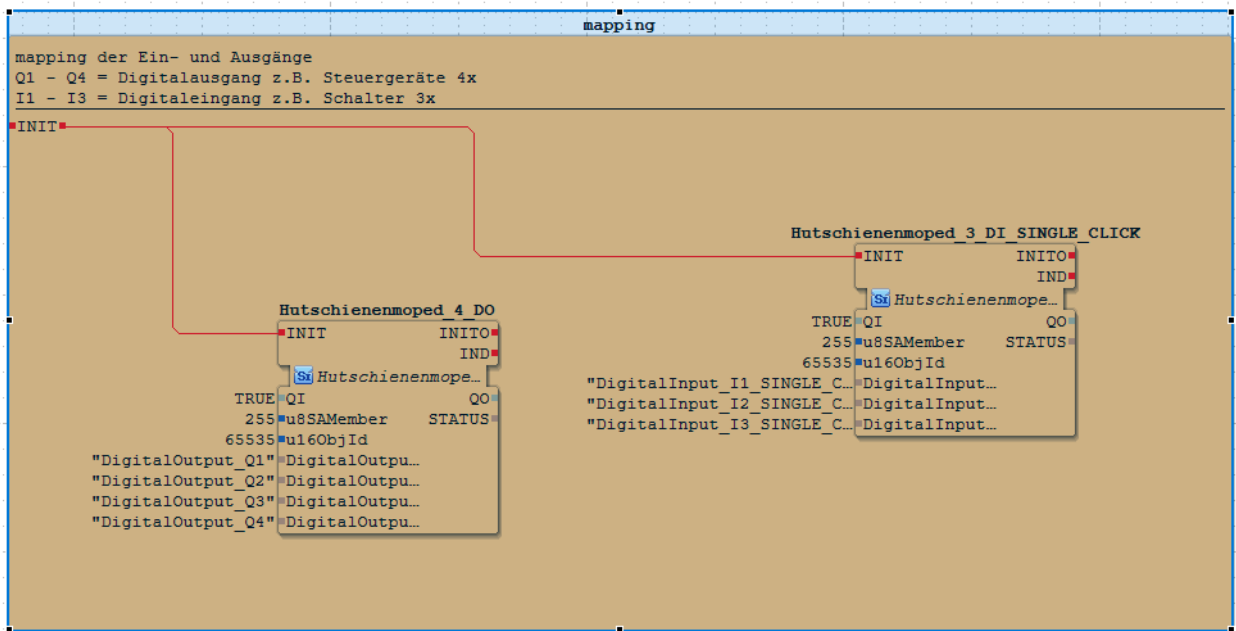
Gänsefüßchen müssen bleiben!



14. Wieder links im Baum mapping wählen. Gruppe anwählen -> Gruppe wird blau und Fenster öffnet. Map To... wählen.

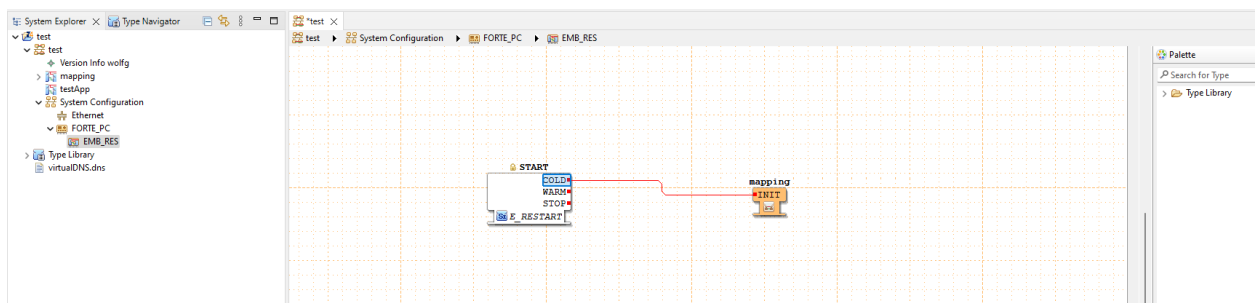


15. Gruppe wurde gemappt. -> Gruppe wird Orange.

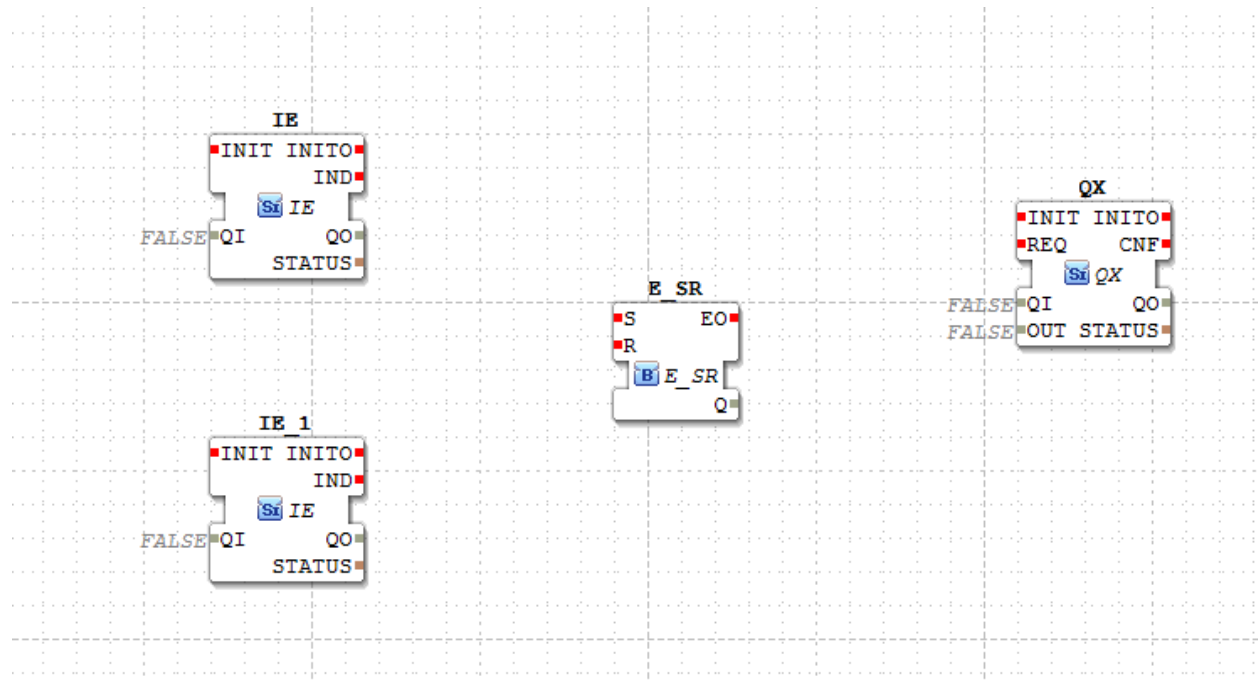


16. In Baum links System Configuration wählen. FORTE_PC wählen. EMB_RES wählen. -> Baustein START und mapping sind auf Arbeitsbildschirm.

Event INIT mit COLD aus START verbinden. (Nach verbinden kein mappen nötig.)

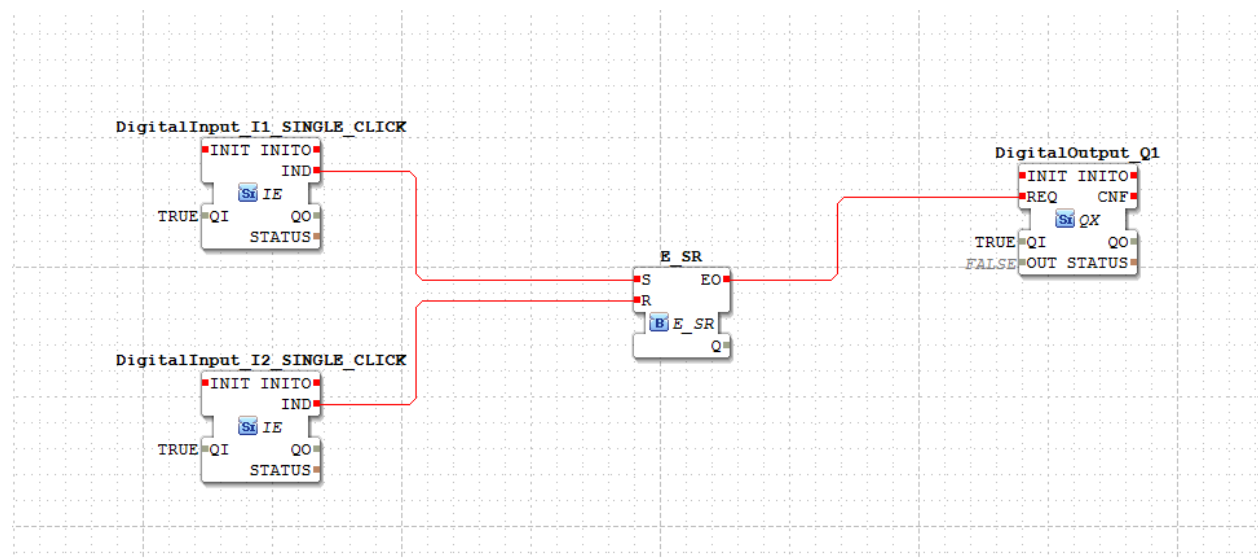


17. Nötige Bausteine für Schaltung aus Type Library mit gedrückter Maustaste in Arbeitsfenster ziehen (Suchfunktion kann genutzt werden, Bausteine können im Arbeitsfenster kopiert werden.

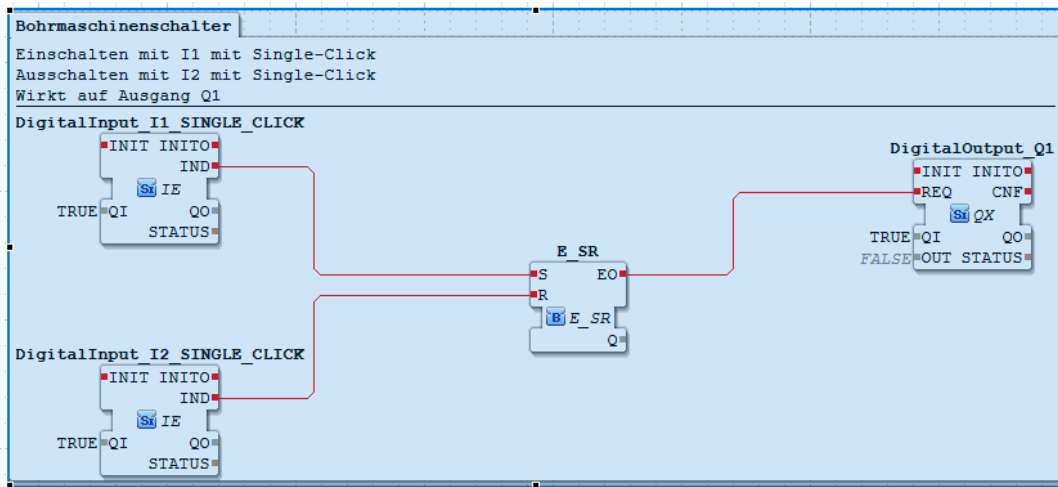


18. Bausteine IE ; IE_1 und QX benennen. Doppelklick auf Namen auf Baustein. Namen aus mapping aus den Bausteinen kopieren und hier einfügen. Keine Gänsefüßchen, Namen dürfen jeweils nur 1x verwendet werden, da sonst ein Fehler erscheint.

QI der Bausteine jeweils auf TRUE setzen. Steuerung miteinander verbinden.

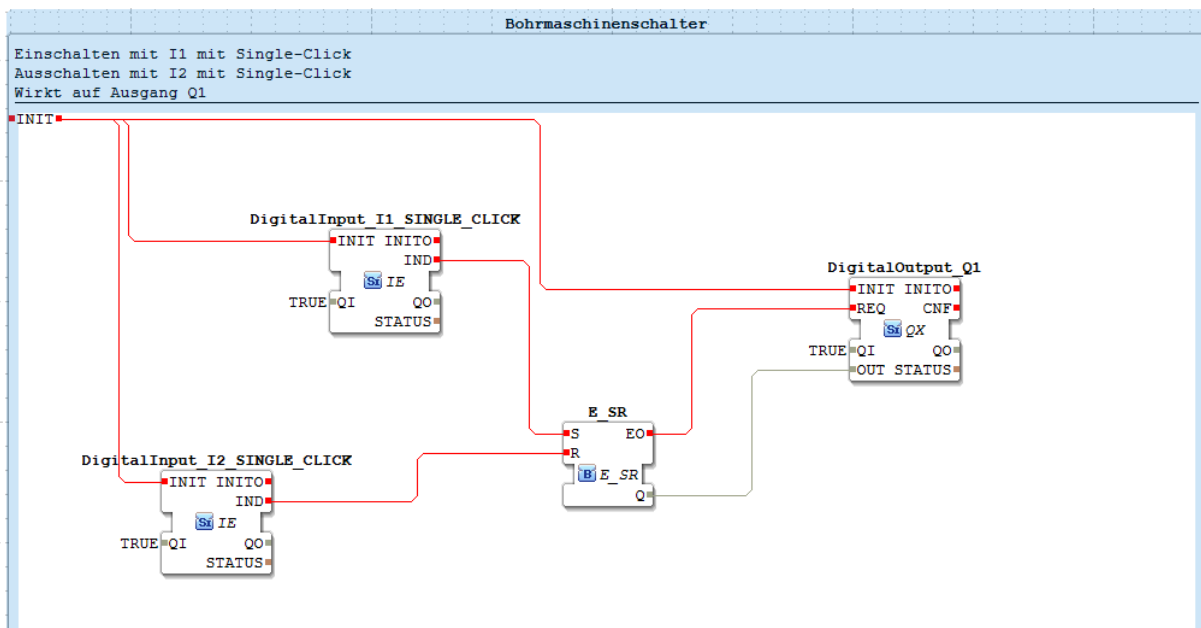


19. Wie bei Punkt Nr. 6 eine neue Gruppe erstellen. Gruppe benennen und Comment (Beschreibung einfügen)

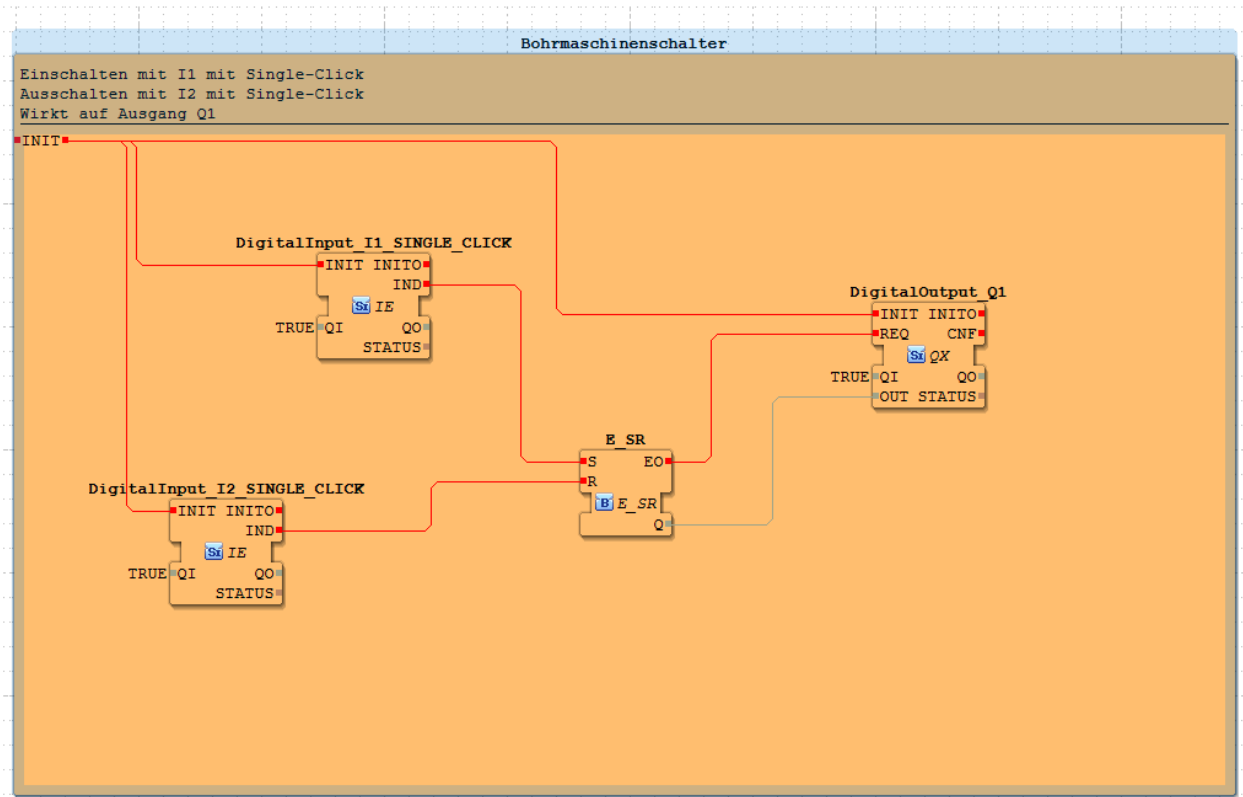


20. Wie bei Punkt 8. Convert Group to Subapp. Wie bei Punkt 10. Event hinzufügen und in INIT unbenennen.

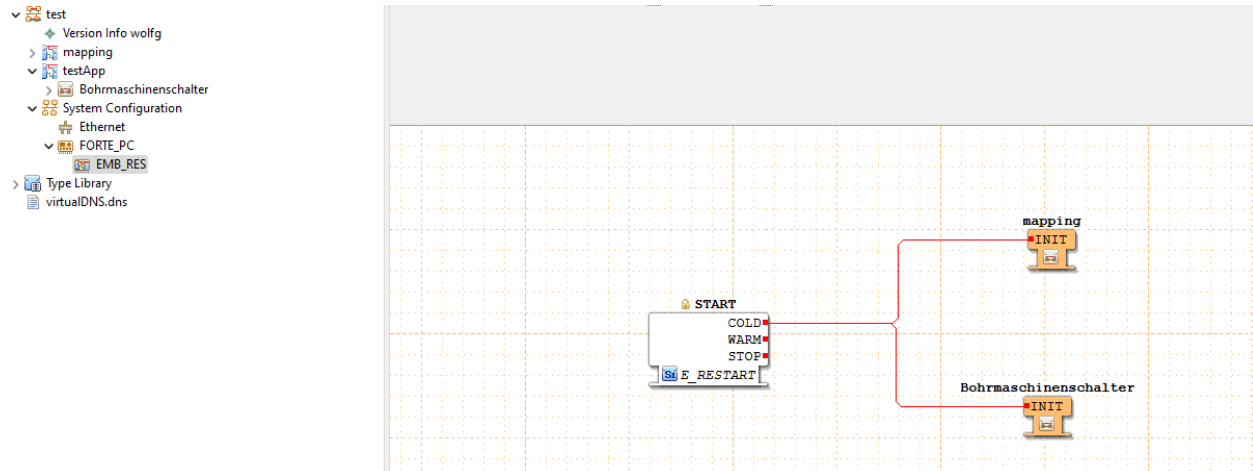
INIT aller Bausteine verbinden + zusätzlich Datenverbindung Q von E_SR und OUT von DigitalOutput_Q1 verbinden.



21. Gruppe Bohrmaschinenschalter anwählen, Rechtsklick und mit map to mappen. -> Gruppe wird Orange.

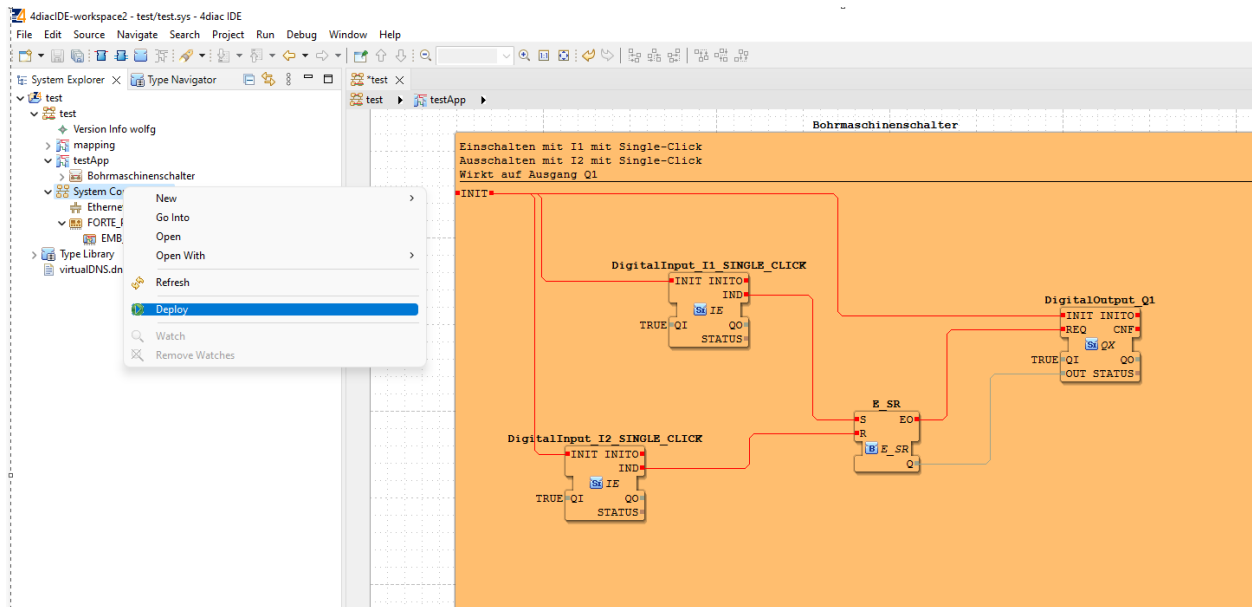


22. Wie bei Punkt 16. unter EMB_RES nun auch den zweiten INIT mit COLD verbinden.

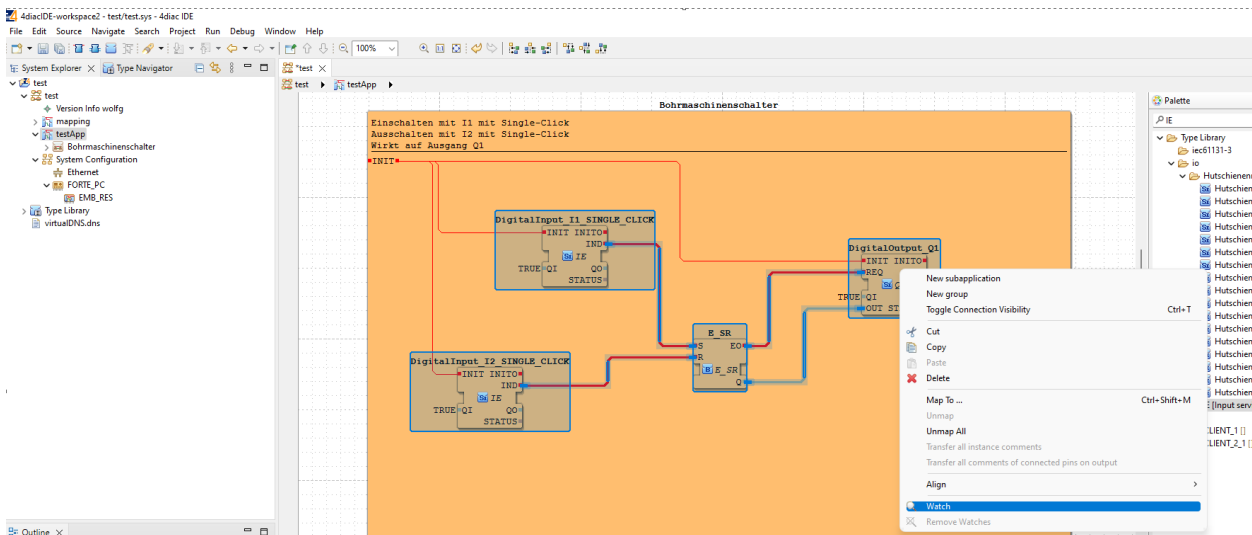


23. Hutschienenmoped mit PC verbinden. Richtiges WLAN -> Eingabeaufforderung -> Eingabe von: ping 192.168.188.11 -t -> Verbindung wird aufgebaut: auf Antwort warten.

24. Links im Baum Rechtsklick auf System Configuration und Deploy wählen. Meldungen bestätigen -> replace all.



25. Steuerung sollte funktionieren. Fenster über Bausteine aufziehen, Rechtsklick dann Watch wählen und die Steuerung kann getestet und kontrolliert werden.



Schalter 1 rot oder blau -> Schaltet Ausgang Q1 und LED ein. (1x drücken)

Schalter 2 blau oder rot -> Schaltet Ausgang Q1 und LED wieder aus. (1x drücken)

6.2.12 Interface

Das Network Interface und das Process Interface sind wichtige Konzepte im Umfeld von Eclipse 4diac. Sie sind wesentliche Bestandteile der IEC 61499 Architektur und ermöglichen die Kommunikation und den Austausch von Daten zwischen verschiedenen Steuerungssystemen.

Das Network Interface ist für die Kommunikation zwischen Steuerungssystemen über ein Netzwerk zuständig. Es stellt sicher, dass die Daten zwischen den verschiedenen Systemen sicher und zuverlässig übertragen werden. Das Network Interface unterstützt verschiedene Protokolle, wie zum Beispiel EtherCAT, PROFINET und Modbus, um sicherzustellen, dass es in verschiedenen Industrieumgebungen eingesetzt werden kann.

Das Process Interface ist für die Kommunikation zwischen verschiedenen Prozessen innerhalb eines Steuerungssystems zuständig. Es stellt sicher, dass die Prozesse miteinander kommunizieren und die benötigten Daten austauschen können. Das Process Interface ist auch für die Synchronisation von Prozessen verantwortlich und sorgt dafür, dass sie sich gegenseitig nicht behindern.

Eclipse 4diac ist eine Entwicklungsumgebung, die es ermöglicht, Anwendungen auf der Basis der IEC 61499 zu erstellen. Es bietet viele Werkzeuge und Funktionen, um die Entwicklung von Steuerungssystemen zu vereinfachen und zu beschleunigen. Mit Eclipse 4diac können Entwickler ihre eigenen Anwendungen erstellen, die das Network Interface und das Process Interface nutzen, um sich mit anderen Steuerungssystemen zu verbinden und zu kommunizieren.

6.2.12.1 NetworkInterface

6.2.12.2 ProcessInterface

6.2.12.2.1 Mapping

In 4diac existieren grundsätzlich 2 Konzepte zur Anbindung von IOs,

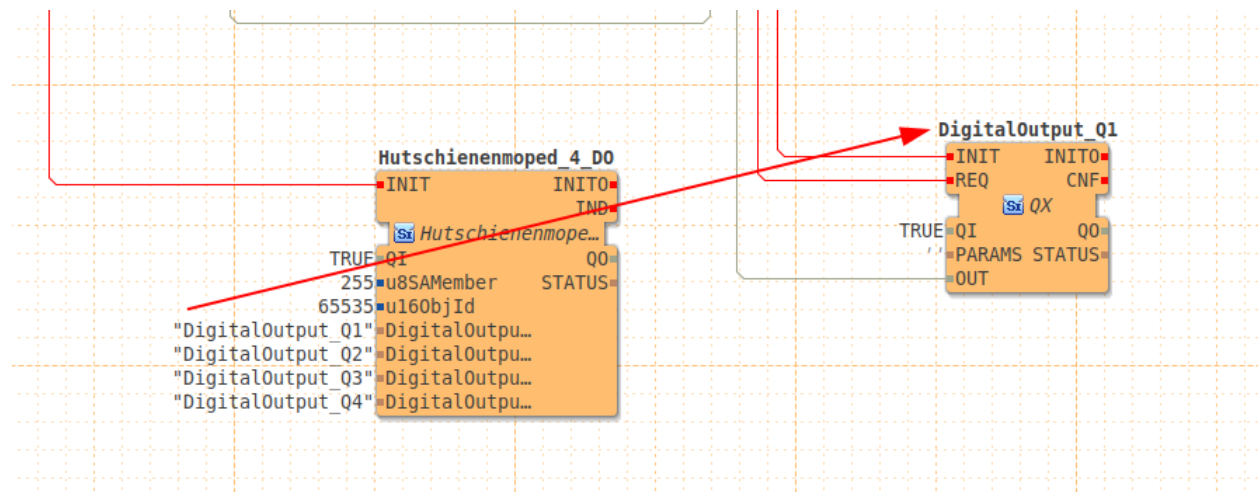
- old IO
- new IO

Wir verwenden ausschließlich new IO.

```
(-DFORTE_IO=ON)
```

In diesem Fall muss bei „Params“ nichts angegeben werden.

Der Name des Bausteins (Typ IX oder QX) muss dem entsprechenden Mapping entsprechen (ohne die Anführungszeichen)



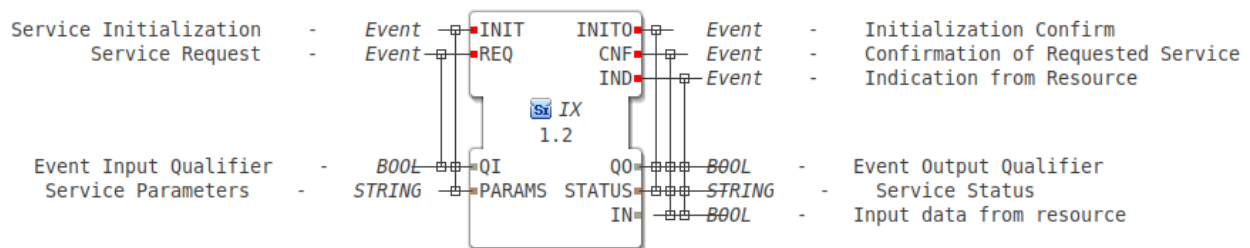
6.2.12.2.2 Eingänge

6.2.12.2.3 Ausgänge

6.2.12.2.4 Eingänge und Buttons

Der Baustein „IX“ ist für Eingänge vom Typ BOOL vorgesehen.
also Eingänge mit 1 Bit Datenbreite,

oder anders ausgedrückt: ein Schaltsignal das nur 2 Zustände kennt: 0 (Taster losgelassen) oder 1 (Taster gedrückt)



Eingänge:

INIT Service Initialization

- muss beim Programmstart 1x aufgerufen werden. Wird in der Regel mit dem Ausgang „COLD“ des Bausteins „E_RESTART“ verbunden. Siehe auch: https://www.eclipse.org/4diac/en_help.php?helppage=html/4diacIDE/use4diacLocally.html und https://www.eclipse.org/4diac/en_help.php?helppage=html/parameters/parameters.html

REQ Service Request

- wird nicht benötigt. Dieser Eingang kommt von alten Zeiten, als man die Eingänge noch pollen musste.

QI Event Input Qualifier

- auf true setzen, damit dieser Baustein funktioniert.

PARAMS Service Parameters

- leer lassen. (das ist mit dem FORTE-IO das wir benutzen nicht mehr verwendet)

Ausgänge:

INITO Initialization Confirm

- wenn INIT aufgerufen wurde, und ausgeführt wurde wird das hier bekanntgegeben.

CNF Confirmation of Requested Service

- wird nicht benötigt. Dieser Eingang kommt von alten Zeiten, als man die Eingänge noch pollen musste.

IND Indication from Resource

- Dieser Ausgang meldet eine Änderung des Status. Bei einem einfachen Ausgang also sowohl die Änderung 0->1 wie auch 1->0

QO Event Output Qualifier

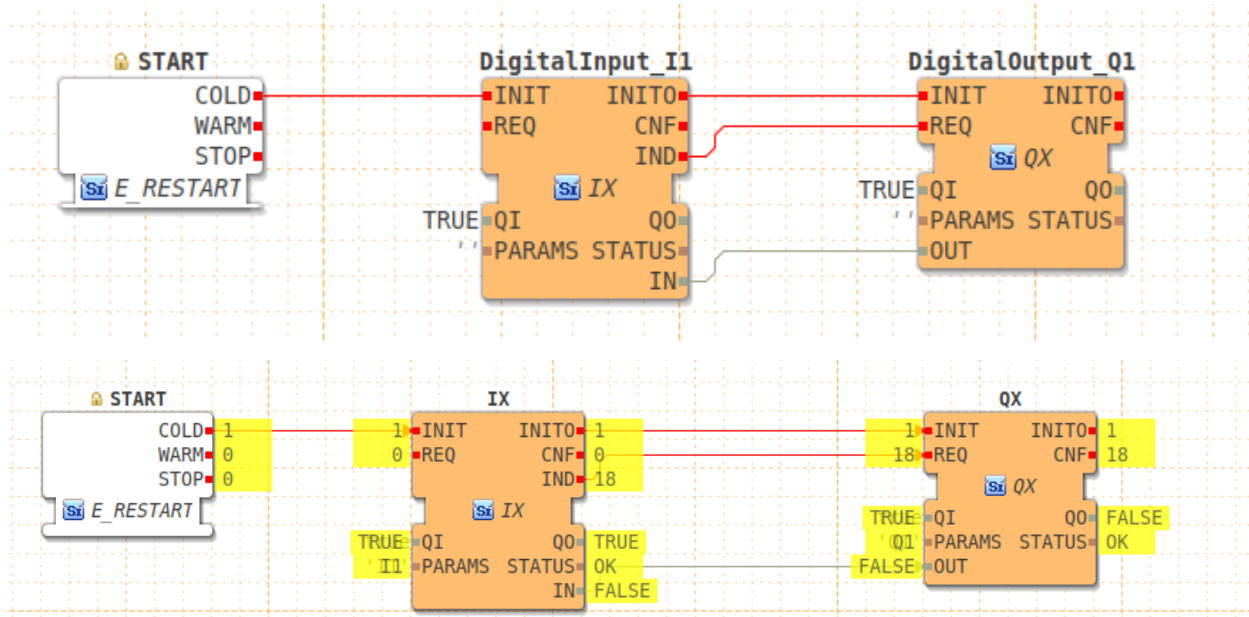
- wird „true“ wenn QI true ist und INIT aufgerufen wurde.

STATUS Service Status

- Hier werden Fehlermeldungen ausgegeben. Wenn kein Fehler vorliegt dann wird OK ausgegeben.

IN Input data from resource

- Das ist der Zustand des Eingangs, also 0 oder 1



- I1 => GPIO_NUM_26
- I2 => GPIO_NUM_32
- I3 => GPIO_NUM_39

damit ist folgende Applikation machbar:

ein M5 Stack Atom Lite (<https://shop.m5stack.com/products/atom-lite-esp32-development-kit>) plus ein Dual Button (<https://shop.m5stack.com/products/mini-dual-button-unit>) ergeben folgende Zuordnung:

- I1 => der Button „B“ (gelbe Drahtfarbe) roter Knopf
- I2 => der Button „A“ (weiße Drahtfarbe) blauer Knopf
- I3 => der Button auf dem M5 Atom Lite

AIx

.....

damit sind die Werte AI1, AI2, AI3, AI4, AI5, AI6 gemeint, diese bilden folgende Eingänge ab:

Nur der Kanal GPIO_NUM_32 also der weiße draht ist in der Lage Analoge Buttons zu verarbeiten.

Der Schaltplan findet sich hier: https://github.com/Meisterschulen-am-Ostbahnhof-Muenchen/ISOBUS_Hardware/tree/feature/ButtonBoards/ButtonBoards/6ButtonTHT

wenn die Eingänge AIx verwendet werden darf der Eingang I2 natürlich nicht gleichzeitig verwendet werden. Die Eingänge I1 und I3 sind davon nicht betroffen.

es können also 6 Buttons mit einem Draht eingelesen werden. Drücken von 2 Tasten gleichzeitig funktionieren dann natürlich nicht.

6.2.13 Übungsaufgaben

6.2.13.1 Die Aufgaben bauen aufeinander auf, so dass es sich empfiehlt diese in der angegebenen Reihenfolge zu bearbeiten.

6.2.13.1.1 Eingänge IX / QX (alles tastend)

6.2.13.1.1.1 Aufgabe Einfache Verbindung I

Verbinden Sie Eingang I1 mit Ausgang Q1, so dass beim drücken von I1 der Aktuator an Q1 betätigt wird.

6.2.13.1.1.2 Aufgabe Einfache Verbindung II

Verbinden Sie I1 mit Ausgang Q1 und Q3.

6.2.13.1.2 Logik

6.2.13.1.2.1 Aufgabe

eine Raketenstarttrampe (Q1) soll nur starten wenn die Tasten I1, I2 und I3 gleichzeitig gedrückt werden.

6.2.13.1.3 Flip Flops I

6.2.13.1.3.1 Aufgabe Klick

Bauen Sie ein Programm, das mit einem Klick das Licht an Q1 einschaltet und mit einem weiteren Klick wieder aus.

6.2.13.1.4 Rendezvous und Flip-Flop

6.2.13.1.4.1 Aufgabe Rendezvous und Flip-Flop

eine andere Raketenstarttrampe soll nur starten wenn der Taster I3 gehalten wird, und dann nacheinander I1 und I2 angeklickt werden. Wird I3 losgelassen, soll nach erneutem drücken von I3 wieder ein Klick auf I1 und I2 notwendig sein.

6.2.13.1.4.2 Aufgabe Reihenfolge

Testen Sie das Programm: wenn Sie zuerst I1 und I2 drücken und dann I3 drücken und halten, darf die Rakete nicht starten. Korrigieren Sie das Programm so dass das funktioniert.

6.2.13.1.5 Flip Flops II

6.2.13.1.5.1 Aufgabe Ein verrückter Designer

Ein verrückter Designer will in seiner Wohnung möglichst wenige und kleine Taster haben. Erstellen Sie ein Programm bei dem das Zimmerlicht (Q1) an geht mit einem Einfachklick, und mit einem Doppelklick wieder aus.

6.2.13.1.5.2 Aufgabe Schreibtischlampe

Der Designer hat sich eine Schreibtischlampe gekauft, will aber keinen separaten Schalter dafür haben. Erweitern Sie das Programm so, dass die Schreibtischlampe (Q4) mit einem langen Tastendruck an geht. Beim Doppelklick sollen Zimmerlicht und Schreibtischlampe aus gehen.

6.2.13.1.6 Flip Flops und Equals

6.2.13.1.6.1 Aufgabe Flip Flops und Equals

Der Designer ruft an, nun will er doch die Schreibtischlampe separat ausschalten können. Der Doppelklick soll wie bisher beide Lampen ausschalten, der Dreifachklick soll nur die Schreibtischlampe ausschalten.

6.2.13.1.7 Flip Flops und Logikfunktionen

6.2.13.1.7.1 Aufgabe Flip Flops und Logikfunktionen

Ein Lastenaufzug hat 2 Funktionen: Hoch (Q1) und Runter (Q2). 3 Taster steuern den Aufzug: Hoch (I1), Runter(I2) und Stop (I3). Die Ausgänge Q1 und Q2 dürfen niemals gleichzeitig betätigt werden, sonst brennt der Motor durch. Bevor man die Richtung wechseln kann muss immer erst Stop gedrückt werden.

6.2.13.1.8 Flip Flops und E Funktionen

6.2.13.1.8.1 Aufgabe Flip Flops und E Funktionen

Ahmen Sie das Beispiel „Step # - Use 4diac Locally (Blinking Tutorial)“ nach

6.2.13.1.8.2 Aufgabe Fahrtrichtungsanzeiger

Fahrtrichtungsanzeiger: die Eingänge I1 und I2 sollen die Fahrtrichtungsanzeiger für links und rechts sein. (nehmen Sie an die Taster wären rastend wie beim Traktor)

6.2.13.1.8.3 Aufgabe Warnblinker

Fahrtrichtungsanzeiger: I3 soll zusätzlich der Warnblinker sein.

6.2.13.1.9 Flip Flops und led_strip

6.2.13.1.9.1 Aufgabe Flip Flops und led_strip

Bauen Sie ein Programm bei dem die LED rot leuchtet wenn Sie die rote Taste drücken, und blau wenn Sie die blaue Taste drücken.

6.2.14 constants

Konstanten in 4diac

6.2.15 TypeLibrary

Willkommen auf unserer Seite zur Typelibrary! Auf dieser Seite finden Sie eine umfassende Sammlung von Typendefinitionen für verschiedene visuelle Programmiersprachen. Die Typelibrary ist ein wichtiges Werkzeug für Entwickler, da sie die Verwendung von verschiedenen Datentypen in ihren Programmen ermöglicht.

Die Typelibrary bietet eine Reihe von Funktionen, die Ihnen dabei helfen, Ihre Programmierarbeit zu vereinfachen und zu beschleunigen. Sie können beispielsweise die verfügbaren Typendefinitionen durchsuchen, um den für Ihre Bedürfnisse geeigneten Datentyp zu finden. Sie können auch Ihre eigenen Typendefinitionen erstellen und in die Typelibrary aufnehmen, um sie in zukünftigen Projekten wiederverwenden zu können.

In diesem Abschnitt werden wir Ihnen die Grundlagen der Typelibrary erklären und Ihnen zeigen, wie Sie sie effektiv in Ihren Projekten verwenden können. Wir hoffen, dass Sie viel Spaß beim Erkunden der Typelibrary haben und sich von ihren Möglichkeiten inspirieren lassen.

schauen Sie auch mal in den FB Navigator unter www.holobloc.com/doc/navigator

6.2.15.1 devices

6.2.15.1.1 BeagleBoneBlack

6.2.15.1.2 FBRT_WINDOW

6.2.15.1.3 FORTE_PC

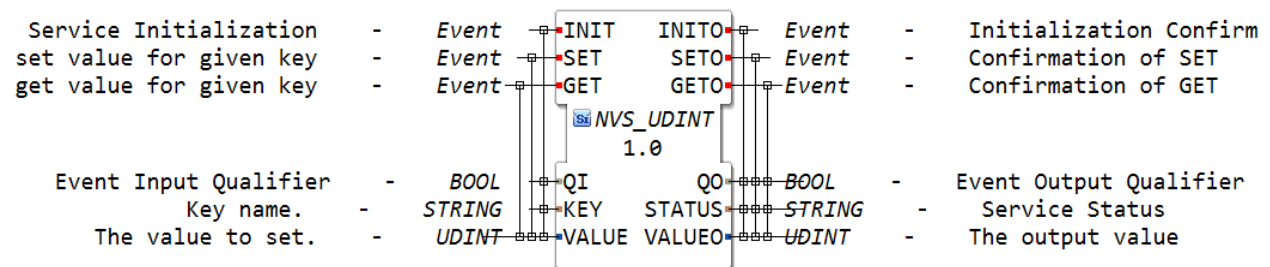
6.2.15.1.4 PFC200

6.2.15.1.5 RaspberryPI

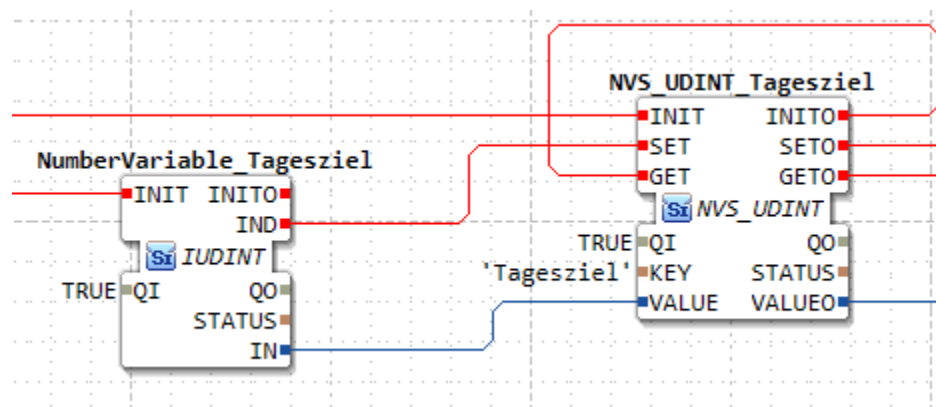
6.2.15.2 esp32

6.2.15.2.1 nvs

6.2.15.2.1.1 NVS_UDINT

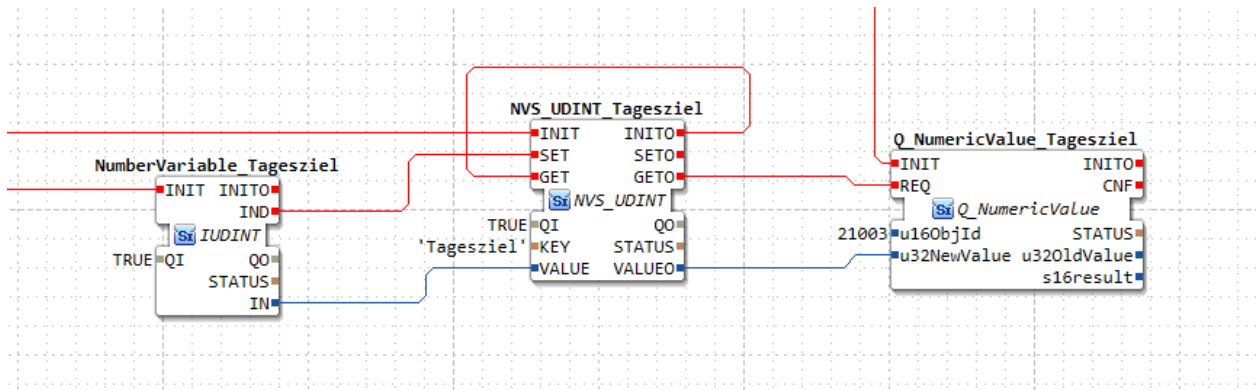


Typische Verwendung:



durch die Schaltung von INITO -> GET wird unmittelbar nach dem Init der Wert mit dem „Key“ geladen, und steht an Ausgang VALUEO zur Verfügung.

soll die Variable dann noch am Display angezeigt werden erweitert man das ganze so:



dadurch ist sowohl die Eingabe als auch die Ausgabe; sowohl das Speichern wie auch das Laden der Variable abgedeckt.

ACHTUNG !!

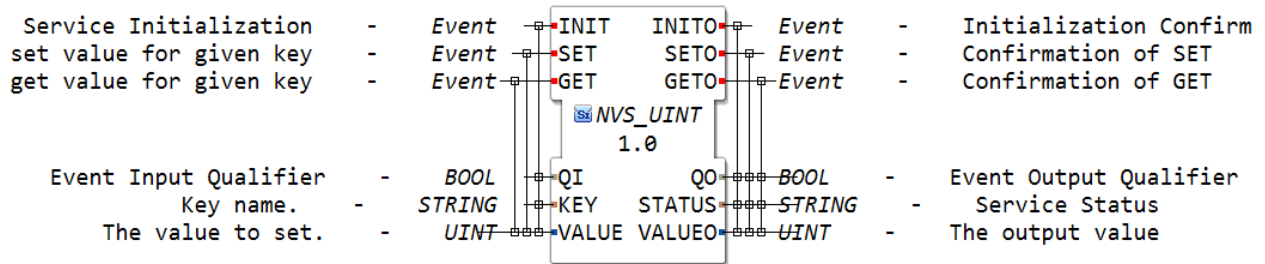
https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/storage/nvs_flash.html

die Länge des Keys muss den Anforderungen der o.g. Library entsprechen.

z.B.

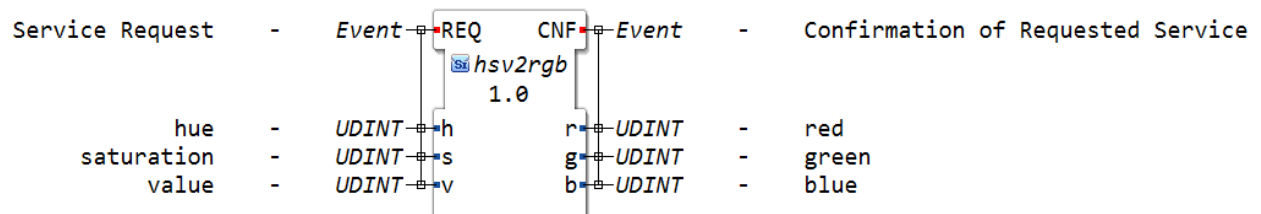
the maximum key length is currently 15 characters

6.2.15.2.1.2 NVS_UINT

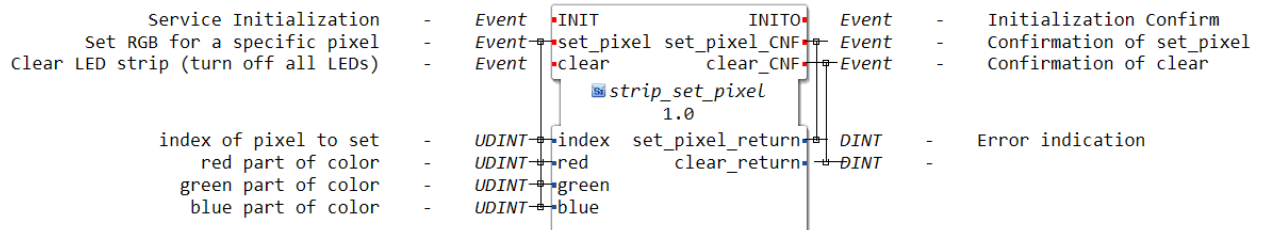


6.2.15.3 esp-addons

6.2.15.3.1 hsv2rgb

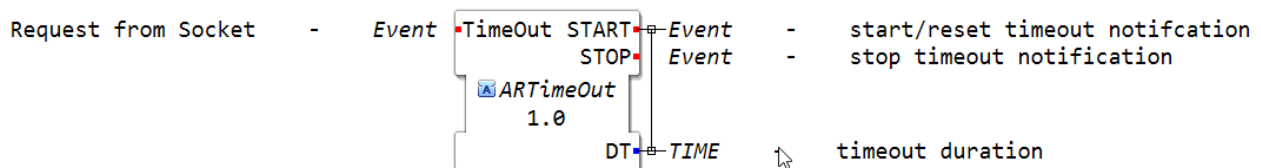


6.2.15.3.2 strip_set_pixel

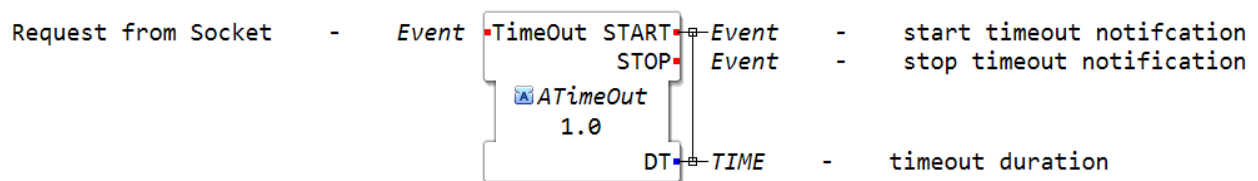


6.2.15.4 events

6.2.15.4.1 ARTimeOut

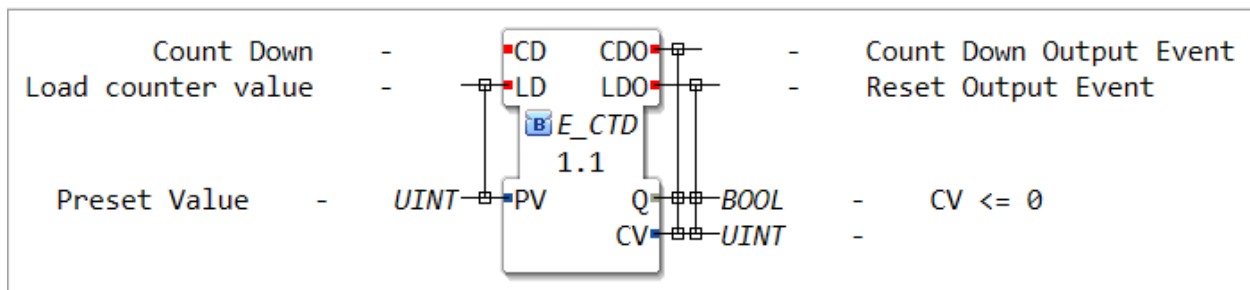


6.2.15.4.2 ATimeOut



6.2.15.4.3 E_CTD

Event-Driven Down Counter



- Input
 - CD Count Down
 - LD Load counter value
- Output

- CDO Count Down Output Event
- LDO Reset Output Event

Input:

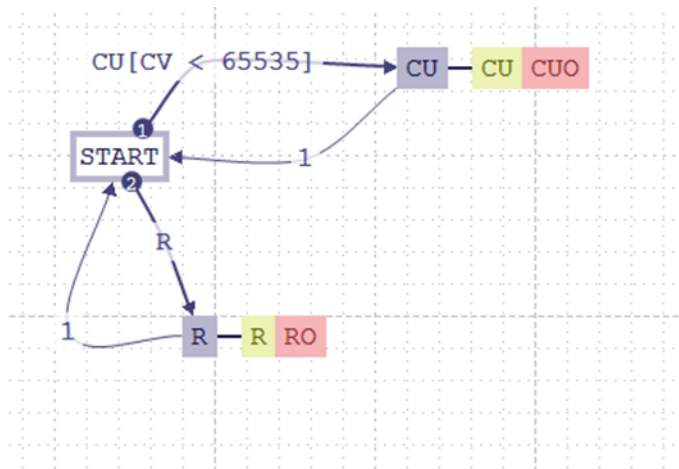
CD: Count Down Ereignis zum Abwärtszählen

LD: Zählwert wird geladen

Output:

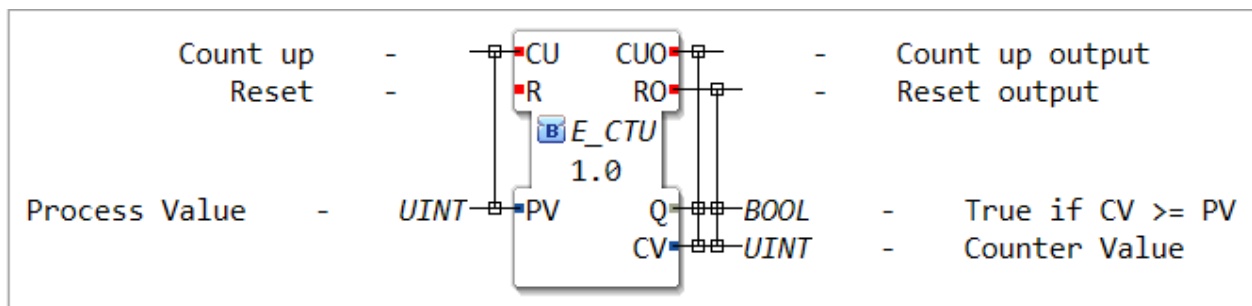
CDO: Countdown Ausgangsereignis

LDO: Ausgangsereignis zurücksetzen



6.2.15.4.4 E_CTU

Event driven up counter



- Input
 - CU Count up
 - R Reset
 - PV Process Value
- Output
 - CUO Count up output
 - RO Reset output
 - Q True if CV >= PV

- CV Counter Value

Input:

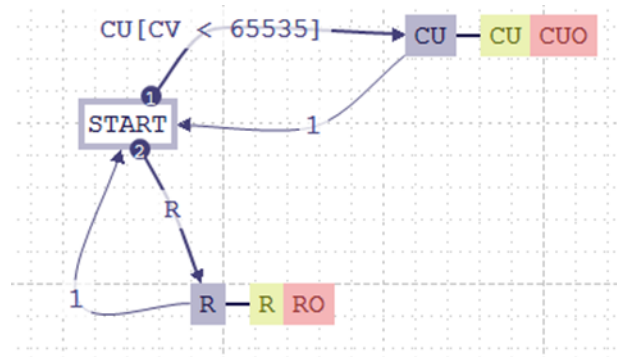
CU: Count up Ereignis zum Aufwärtszählen

R: Zurücksetzen

Output:

CUO: Countup Ausgangsereignis

R: Ausgang zurücksetzen



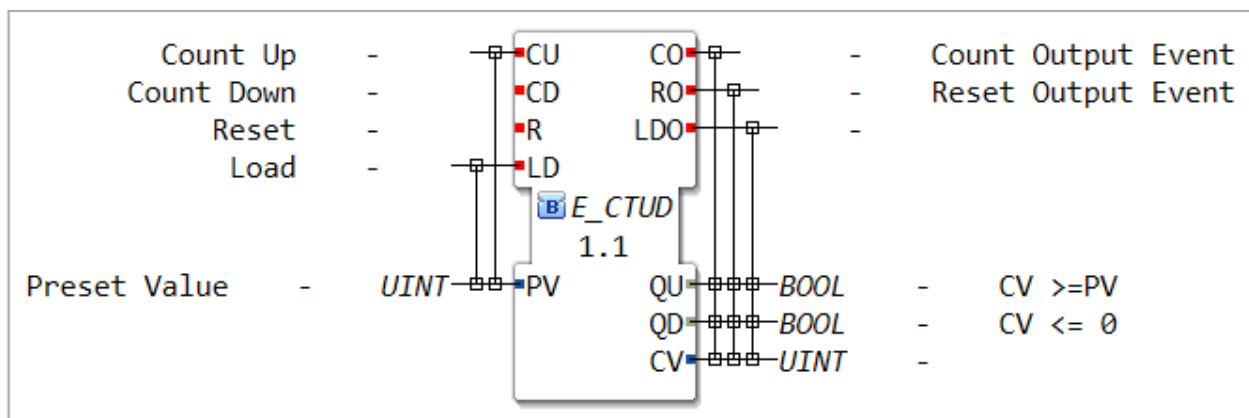
6.2.15.4.5 E_CTUD

Event-Driven Up-Down Counter

Der Funktionsblock E_CTUD (Counter Up/Down) ist ein wichtiger Bestandteil der IEC 61499 Norm und wird häufig in Steuerungssystemen verwendet. Er dient dazu, Zählerwerte zu erhöhen oder zu verringern und kann in vielen verschiedenen Anwendungen eingesetzt werden.

Der E_CTUD Funktionsblock besteht aus mehreren Eingängen und Ausgängen, die für die Steuerung und Überwachung des Zählerwerts verwendet werden. Der wichtigste Eingang ist der „CU“ Eingang, der verwendet wird, um den Zählerwert zu erhöhen oder zu verringern. Der „RESET“ Eingang dient zum Zurücksetzen des Zählerwerts auf null, während der „LOAD“ Eingang verwendet wird, um den Zählerwert auf einen festgelegten Wert zu laden.

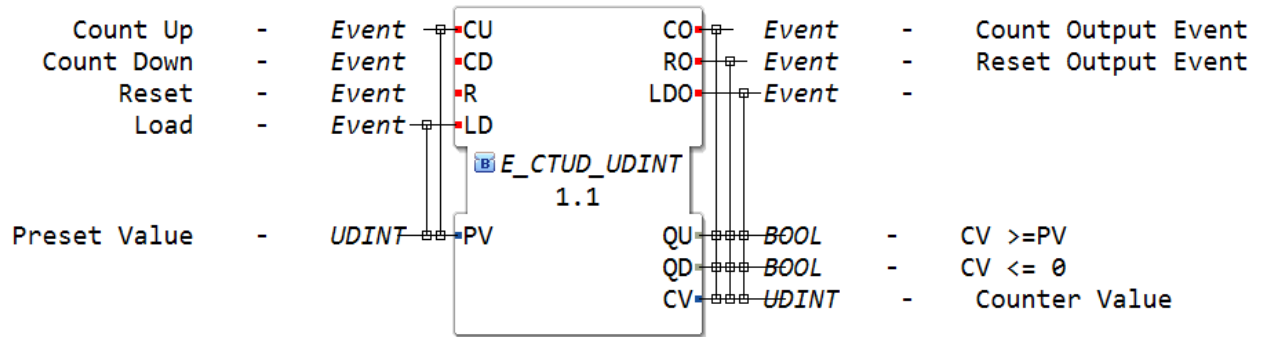
Der E_CTUD Funktionsblock hat auch mehrere Ausgänge, die für die Überwachung des Zählerwerts und die Steuerung von Prozessen verwendet werden.



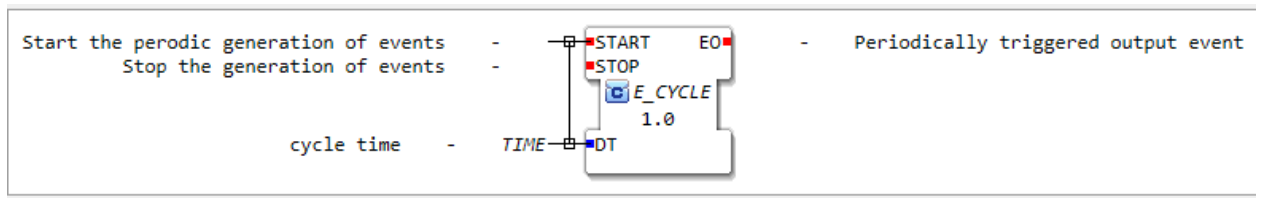
- Input
 - CU Count Up

- CD Count Down
- R Reset
- LD Load
- PV Preset Value
- Output
 - CO Count Output Event
 - RO Reset Output Event
 - QU CV >= PV
 - QD CV <= 0

6.2.15.4.6 E_CTUD_UDINT



6.2.15.4.7 E_CYCLE Taktgenerator (Blinker)



Zeitschaltrelai zb. Blinker

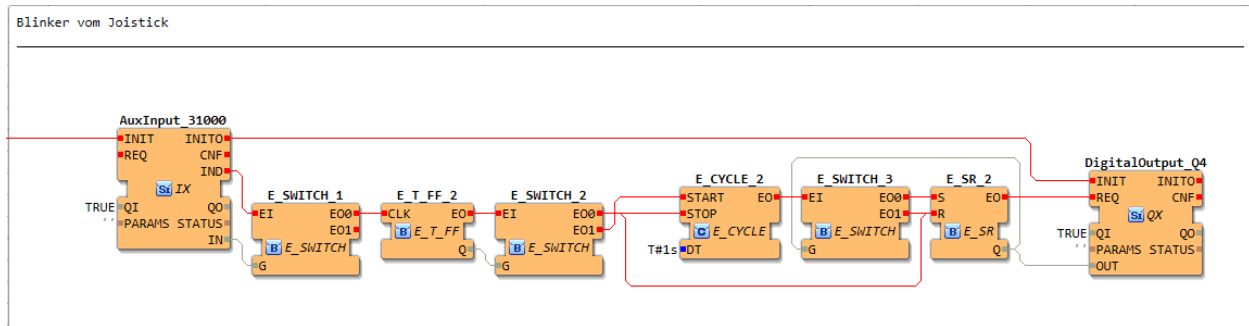
Start= Event vom Schalter zum Starten

Stop= Event vom Schalter zum Ausschalten

Cycle Tyme= Einstellen der Länge des Events (Ausgang wird so lange bestromt)

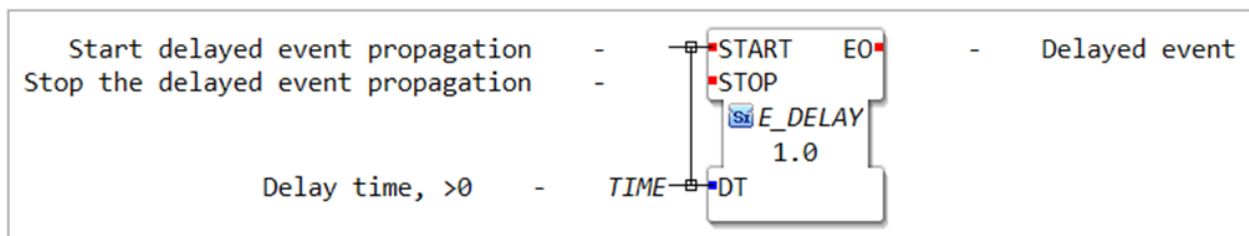
EQ= Ausgang des Events

Nur beim Joisick !!!



6.2.15.4.8 E_DELAY

Verzögerte Ausbreitung eines Ereignisses - abbrechbar



EVENT_INPUT

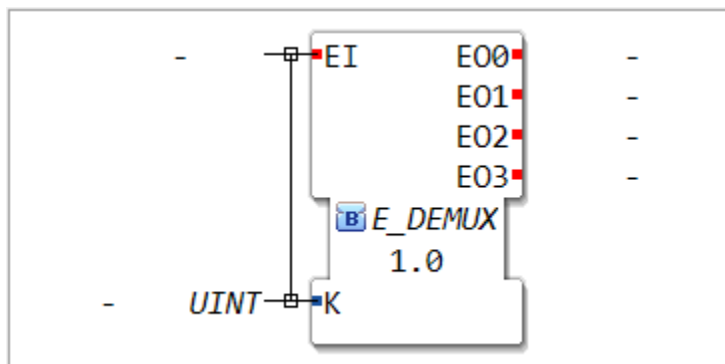
EVENT_OUTPUT

VAR_INPUT

Nach dem Auftreten eines Ereignisses am **Eingang START** und dem **Zeitintervall Zeit DT** wird ein **Ereignis an EO** erzeugt. Die Ereignisverzögerung wird durch ein Auftreten eines Ereignisses am **Eingang STOP** widerrufen. Wenn mehrere Ereignisse am **Eingang START** vor dem Auftreten eines **Ereignisses an EO** auftreten, tritt nach dem ersten Ereignisauftritt am **Eingang START** und dem anschließenden **Verstreichen von DT** nur ein einzelnes **Ereignis an EO** auf. Es wird keine Zeitverzögerung initiiert werden, wenn ein Ereignis am **START-Eingang** mit dem **Wert von DT** auftritt, **der nicht größer als T # 0s**.

6.2.15.4.9 E_DEMUX

Event demultiplexor



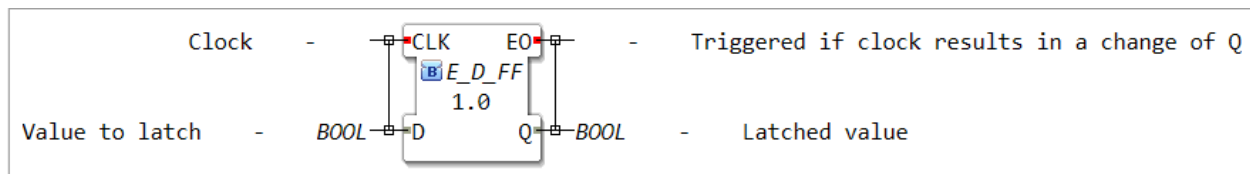
6.2.15.4.10 E_D_FF

Data latch (d) flip flop

Ein Datenlatch ist eine Schaltung in der Elektronik, die es ermöglicht, ein Datensignal zu speichern. Es besteht aus zwei Hauptkomponenten: einem Flip-Flop und zwei Eingängen, die als „D“ und „CLK“ bezeichnet werden. Der Eingang „D“ steht für das Datensignal, das gespeichert werden soll, während der Eingang „CLK“ ein Taktsignal darstellt, das angibt, wann das Datensignal gespeichert werden soll.

Wenn das Flip-Flop aktiviert wird, speichert es den Wert des Eingangs „D“ und hält ihn fest, bis es erneut aktiviert wird. Dies geschieht, indem das Flip-Flop auf das Taktsignal „CLK“ reagiert und den Wert des Eingangs „D“ speichert, wenn das Taktsignal einen vorbestimmten Zustand erreicht.

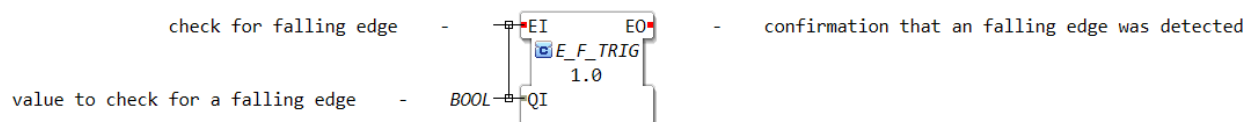
Das Datenlatch wird häufig in Schaltungen verwendet, die Daten zwischen verschiedenen Komponenten übertragen, um sicherzustellen, dass die Daten zum richtigen Zeitpunkt verfügbar sind. Es kann auch verwendet werden, um den Zustand eines Systems zu speichern, z.B. um die Eingabe eines Benutzers zu speichern, während andere Operationen ausgeführt werden.



- Input
 - CLK Clock
 - D Value to latch
- Output
 - EO Triggered if clock results in a change of Q
 - Q Latched value

6.2.15.4.11 E_F_TRIG

Boolean falling edge detection



- Input
 - EI check for falling edge
 - QI value to check for a falling edge
- Output
 - EO confirmation that an falling edge was detected

6.2.15.4.12 E_MERGE

(Zusammenführung (OR) mehrerer Events.

Eventeingänge:

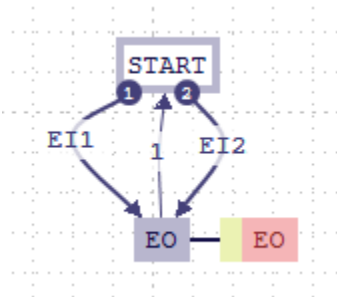
- EI1: Erster Ereigniseingang
- EI2: Zweiter Ereigniseingang

Eventausgang:

- EO: Ausgangsereignis

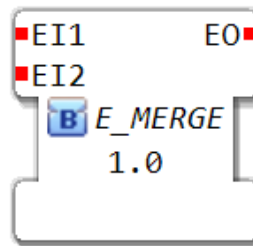
Das Auftreten eines Events an einem der Eingänge EI1 oder EI2 verursacht das Auftreten eines Events an EO.

Beispiel: Zwei Knöpfe betätigen die gleiche Funktion, egal welcher Knopf betätigt wird löst ein Event an EO aus.



First input event
Second input event

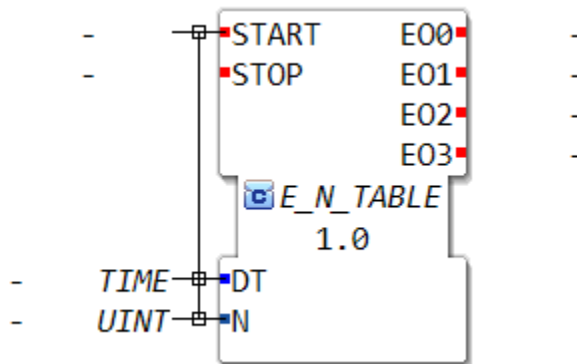
-
-



- Output Event

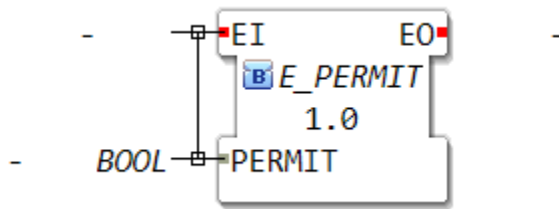
6.2.15.4.13 E_N_TABLE

Generation of a finite train of sperate events



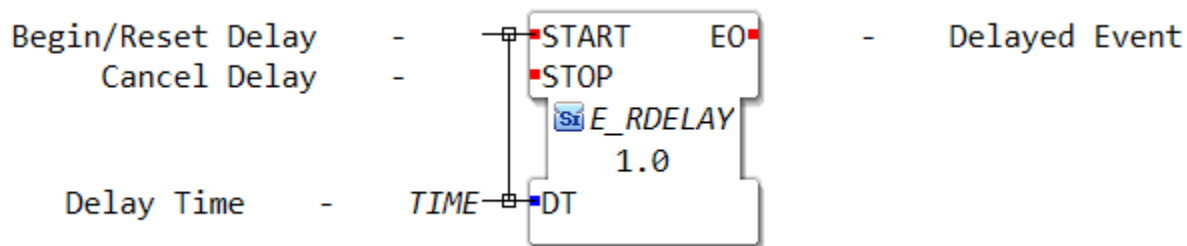
6.2.15.4.14 E_PERMIT

permissive propagation of an event



6.2.15.4.15 E_RDELAY

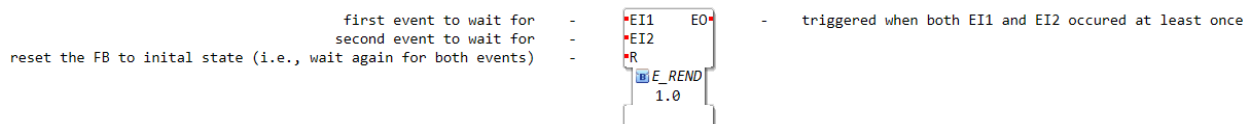
Reloadable delayed propagation of an event-Cancellable



- Input
 - START Begin/Reset Delay
 - STOP Cancel Delay
 - DT Delay Time
- Output
 - EO Delayed Event

6.2.15.4.16 E_REND

Rendezvous (AND) of two events



- Input
 - EI1 first event to wait for
 - EI2 second event to wait for
 - R reset the FB to initial state (i.e., wait again for both events)
- Output
 - EO triggered when both EI1 and EI2 occurred at least Eingabe/Input:

Eingang/Input

- EI1 erstes Ereignis, auf das gewartet werden soll
- EI2 zweites Ereignis, auf das gewartet werden soll
- R (wie ein reset-Knopf)
setzt den FB auf den Ausgangszustand zurück (d.h. wartet erneut auf beide Ereignisse)

Ausgabe/Output:

- EO wird nur ausgelöst,
wenn sowohl EI1 als auch EI2 mindestens einmal aufgetreten sind!

6.2.15.4.17 E_RESTART

Service Interface Function Block Type

Tipp: Bei Betätigung der Resettaste wird beim Hutschienenmopped** *immer* **auf Cold geschaltet. :::

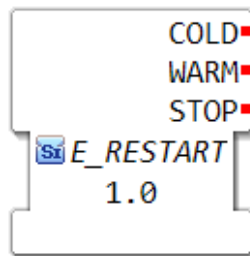
Bemerkung: Dieses ist sehr oft mit INIT verbunden. :::

Beim Kaltstart (COLD) werden alle Daten die auf den Programmspeicher hinterlegt sind zurückgesetzt wie z.B. Zeiten, Zähler

Beim Warmstart (WARM) findet keine Initialisierung des Programmablaufs statt.

Hierfür ist notwendig, dass der Zustand nach dem Einschalten mit dem Zustand vor dem Ausschalten gleich ist.

BEI STOP wird die komplette Anwendung gestoppt.

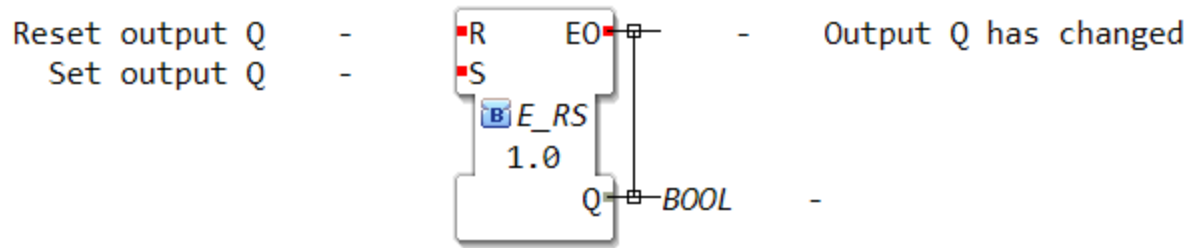


- Information on cold restart
- Information on warm restart
- information that the resource is to be stopped

6.2.15.4.18 E_RS

Tipp: In der IEC 61131-3 wird noch noch zwischen SR(vorrangiges Setzen) und RS(vorrangiges Rücksetzen) unterschieden, in der IEC 61499-1 nicht mehr. Das bedeutet RS und SR haben die gleiche Funktion nur die Anschlüsse Input S und R sind vertauscht

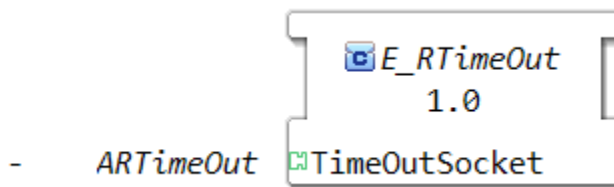
Event-driven bistable



- Input
 - R Reset output Q
 - S Set output Q
- Output
 - EO Output Q has changed
 - Q

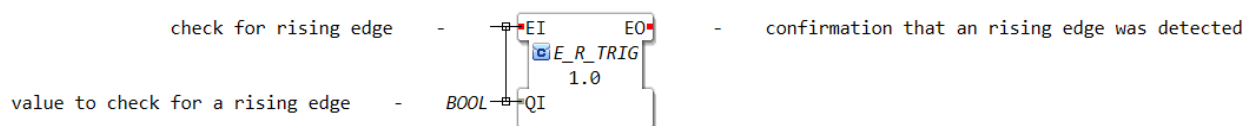
6.2.15.4.19 E_RTimeOut

Simple implementation of the timeout services



6.2.15.4.20 E_R_TRIG

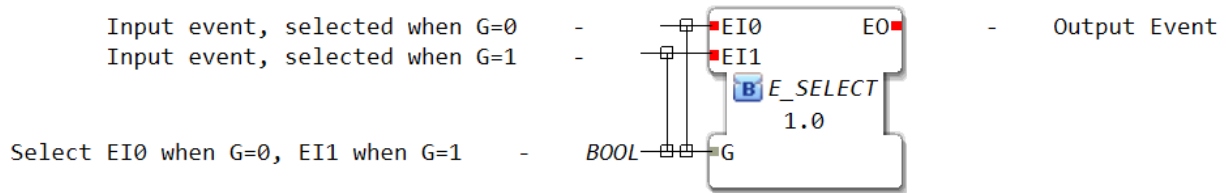
Boolean rising edge detection



- Input
 - EI check for rising edge
 - QI value to check for a rising edge
- Output
 - EO confirmation that an rising edge was detected

6.2.15.4.21 E_SELECT

Auswahl aus zwei Ereignissen



EVENT_INPUT

EVENT_OUTPUT EO Ausgangsereignis

VAR_INPUT G: *BOOL* Wählt EI0, wenn G=0 / EI1, wenn G=1

Siehe auch: https://www.holobloc.com/doc/fb/rt/events/E_SELECT.htm

6.2.15.4.22 E_SPLIT

Aufsplitten eines Events.

Eventeingang:

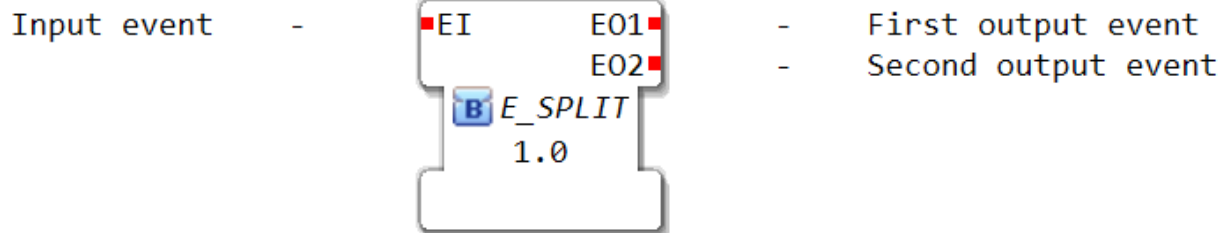
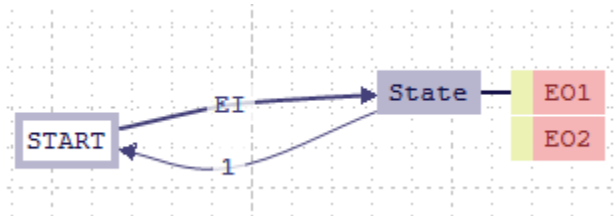
- EI: Ereigniseingang

Eventausgänge:

- EO1 Erstes Ausgangsereignis
- EO2 Zweites Ausgangsereignis

Das Auftreten eines Events EI verursacht das Auftreten der Events EO1 und EO2.

Beispiel: Ein Knopf wird betätigt und es werden je auf EO1 und EO2 ein Ereignis ausgelöst.



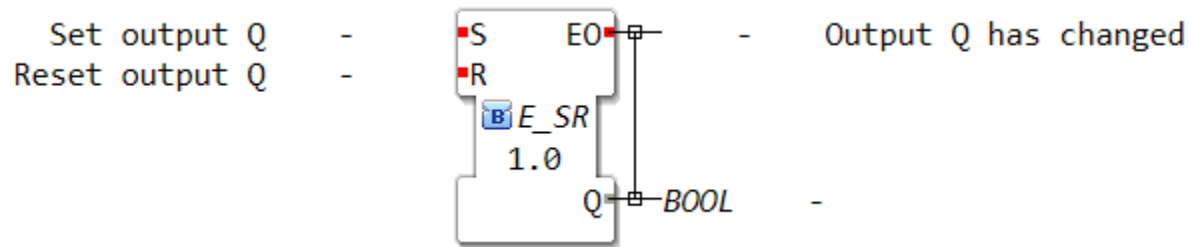
6.2.15.4.23 E_SR = (Holzspalter Schalter)

Tipp: In der IEC 61131-3 wird noch zwischen SR(vorrangiges Setzen) und RS(vorrangiges Rücksetzen) unterschieden, in der IEC 61499-1 nicht mehr. Das bedeutet RS und SR haben die gleiche Funktion nur die Anschlüsse Input S und R sind vertauscht

Event-driven bistable

Der Funktionsblock E_SR (Set-Reset) ist ein wichtiger Bestandteil der IEC 61499 Norm und wird häufig in Steuerungssystemen verwendet. Er dient dazu, eine bestimmte Aktion auszulösen, wenn eine bestimmte Bedingung erfüllt ist.

Der Funktionsblock E_SR besteht aus zwei Eingängen und einem Ausgang. Der erste Eingang (S) dient dazu, die Aktion auszulösen, während der zweite Eingang (R) dazu dient, die Aktion zurückzusetzen. Der Ausgang des Funktionsblocks gibt an, ob die Aktion ausgelöst wurde oder nicht.



- Input
 - S Set output Q
 - R Reset output Q
- Output
 - EO Output Q has changed



<https://agro-center.de/schalter-fuer-holzspaltermotor-etzxxx961020.html>

- Ein Schalter = Einschalten
- Aus Schalter = Ausschalten

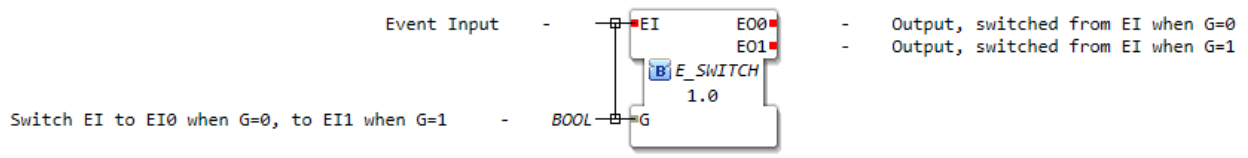
Siehe auch: [https://www.holobloc.com/doc/fb/rt/events/E_SR.htm](https://www.holobloc.com/doc/fb/rt/events/E_SR.htm)

6.2.15.4.24 E_SWITCH

Der Funktionsbaustein E_SWITCH ist ein Standardfunktionsbaustein der IEC 61499-1 und wird verwendet, um die Steuerung von Schaltvorgängen in automatisierten Systemen zu realisieren. Er ermöglicht es, bestimmte Funktionen oder Aktionen basierend auf bestimmten Bedingungen auszuführen.

Der Funktionsbaustein E_SWITCH hat einen Eingang G, der auch als „Kriterium“ bezeichnet wird. Jedes Kriterium repräsentiert eine bestimmte Bedingung, die erfüllt sein muss, damit eine bestimmte Aktion ausgeführt wird. Die Aktionen werden als „Zweige“ bezeichnet und können verschiedene Funktionen oder Aktionen darstellen.

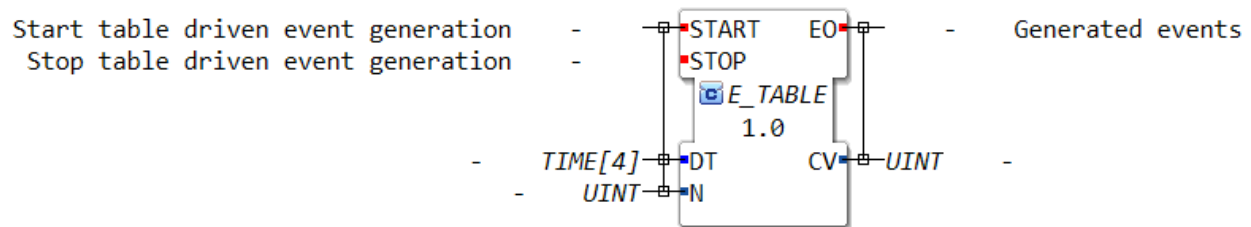
Der Funktionsbaustein E_SWITCH wird wie folgt verwendet: Wenn ein Kriterium erfüllt ist, wird die zugehörige Aktion ausgeführt. Wenn keines der Kriterien erfüllt ist, wird die Aktion des „sonst“-Zweigs ausgeführt. Es gibt auch die Möglichkeit, mehrere Kriterien gleichzeitig zu verwenden und die Aktionen der verschiedenen Zweige zu kombinieren.



Siehe auch: https://www.holobloc.com/doc/fb/rt/events/E_SWITCH.htm

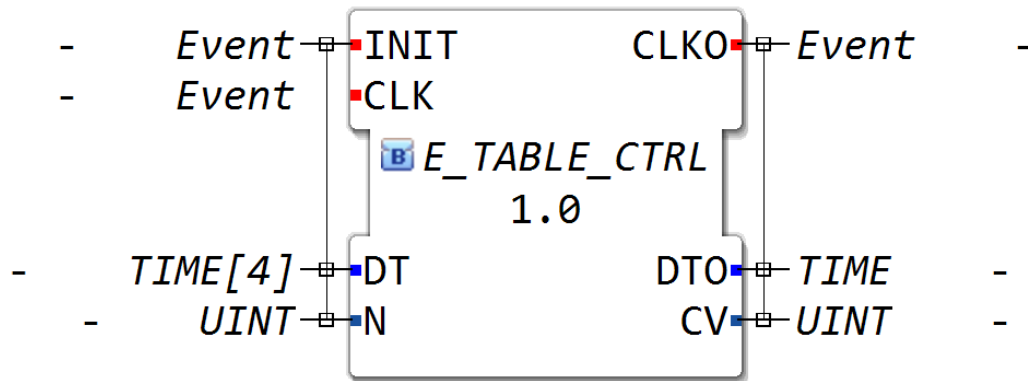
6.2.15.4.25 E_TABLE

Composite Function Block Type

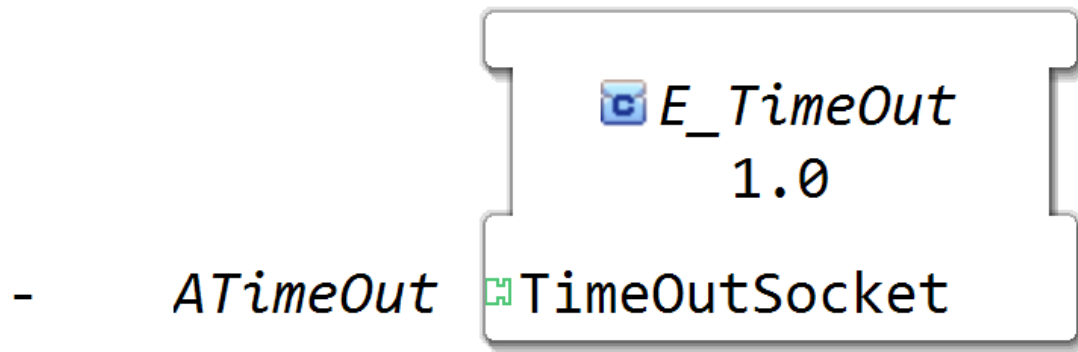


- Input
 - START Start table driven event generation
 - STOP Stop table driven event generation
 - DT
 - N
- Output
 - EO Generated events
 - CV

6.2.15.4.26 E_TABLE_CTRL



6.2.15.4.27 E_TimeOut



Adapter fassen Schnittstellenelemente zusammen. Sie reduzieren die Anzahl der Schnittstellenelemente (Ereignis, Dateneingänge und Ausgänge) und damit die Anzahl der Verbindungen, die zwischen interagierenden Funktionsblöcken erstellt werden müssen. In 4diac IDE werden Adapter in der Typbibliothek aufgelistet und durch dieses Symbol dargestellt: Adapter.



socket: wird als Eingang eines Funktionsblocks hinzugefügt und innerhalb der Schnittstelle des Funktionsblocks dargestellt.

plug: wird als Ausgang eines Funktionsblocks hinzugefügt und innerhalb der Schnittstelle des Funktionsblocks dargestellt.

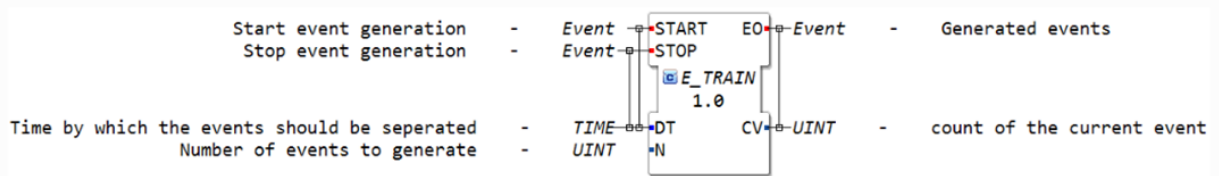
Time: In Sekunden die der Baustein das Event sperrt

Der **E_TimeOut** ist ein zusammengesetzter Funktionsblock. Innerhalb des Netzwerks eines zusammengesetzten Funktionsblocks wird jeder Adapter, der zu seiner Schnittstelle hinzugefügt wird, durch einen Adapterblock repräsentiert,

der wie ein Funktionsblock aussieht. Die Schnittstellenelemente dieses Adapterblocks sind wie ein Funktionsblock verbunden.

6.2.15.4.28 E_TRAIN

15.4.28. E_TRAIN

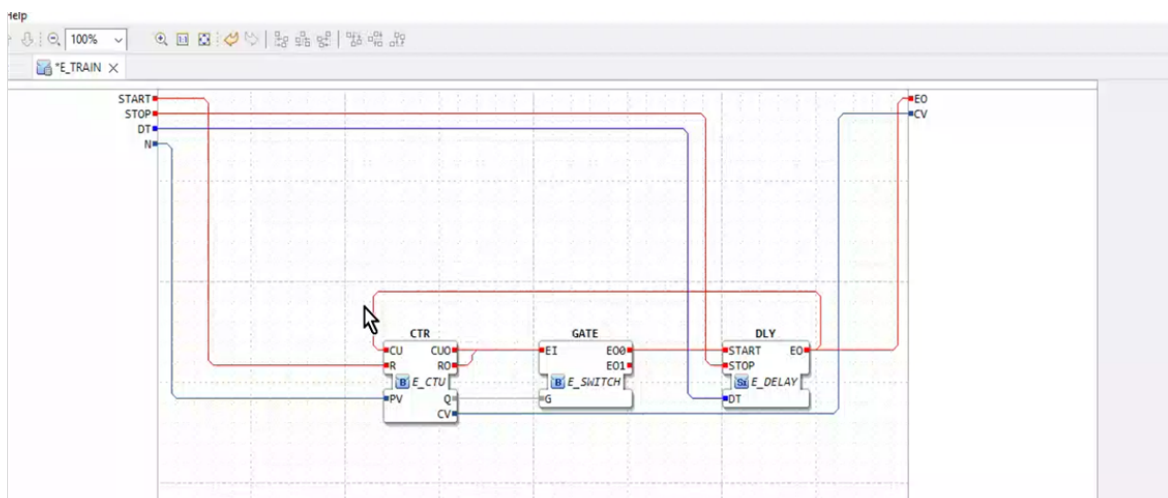


Zeigt eine Ereignisfolge

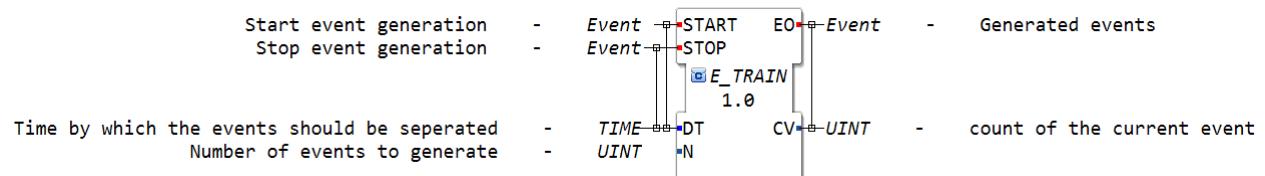
CV N Ereignisse zur Periode DT, beginnend bei DT nach START

DT Time Periode zwischen Ereignissen

N UINT Anzahl der zu erzeugenden Ereignisse



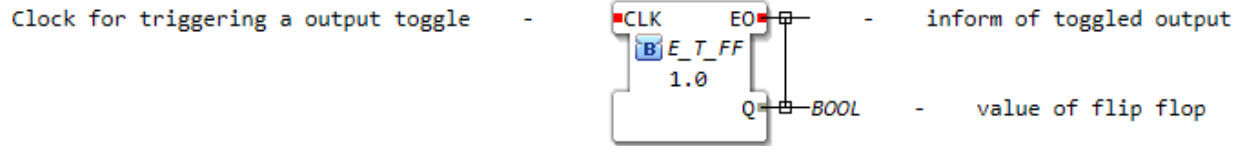
Über Event Start wird das Ereignis gestartet und über Stopp angehalten. Durch CV werden die aktuellen Ergebnisse gezählt und auf EO das generierte Ergebnis Time stellt die Zeit da in welchen abständen das Event auftreten soll Unit wie viel Event bereit durchgeführt wurden



6.2.15.4.29 E_T_FF

Toggle flip flop

IEC 61499-1 Annex A



- Input
 - CLK Clock for triggering a output toggle
- Output
 - EO inform of toggled output
 - Q value of flip flop

das Toggle flip flop stellt einen „Stromstoßschalter“ dar.

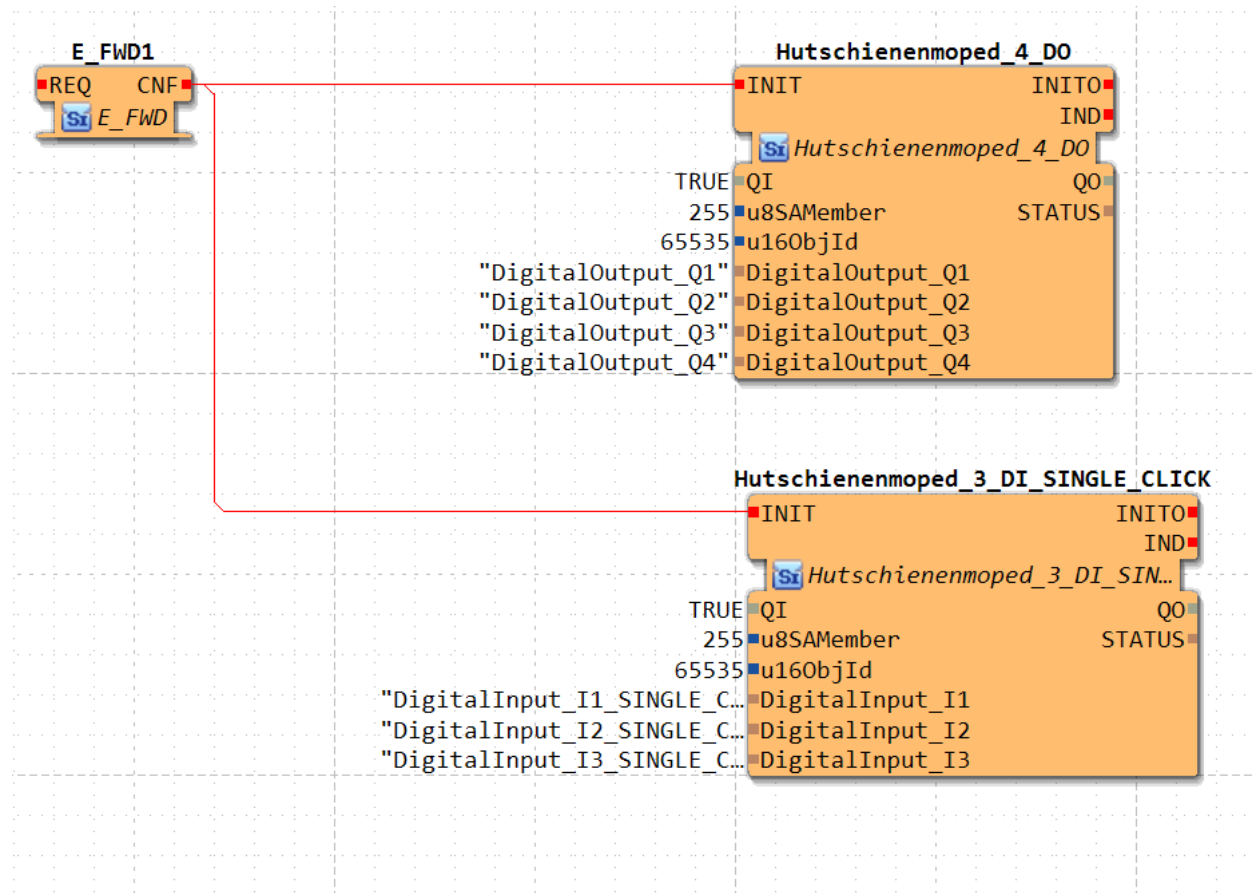
<https://de.wikipedia.org/wiki/Stromsto%C3%9Fschalter>

6.2.15.4.29.1 E_T_FF Anwendungen

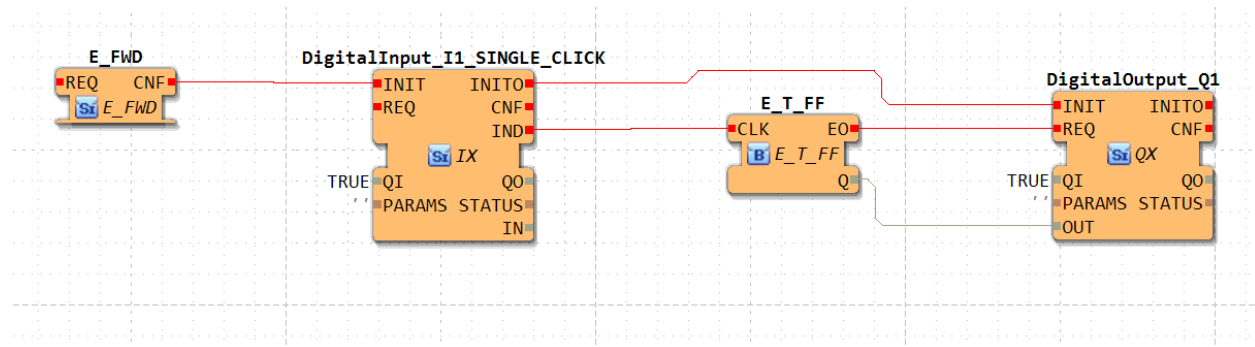
6.2.15.4.29.2 E_T_FF Anwendungen Taster

mit einem E_T_FF lässt sich so ein Taster realisieren, der eine Lampe ein und ausschaltet:

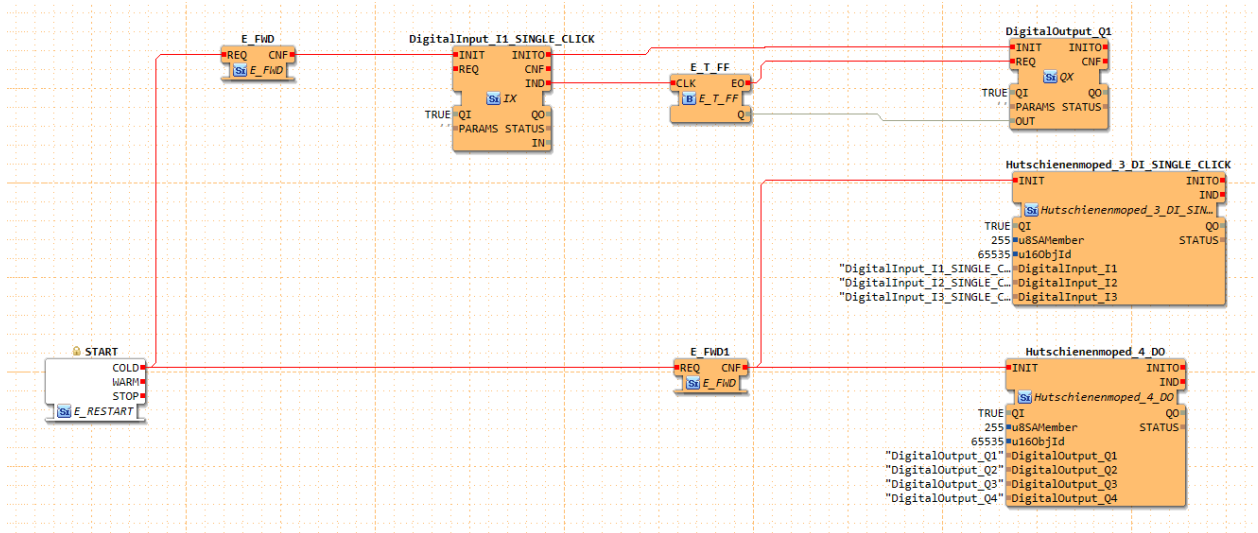
mapping:



App:



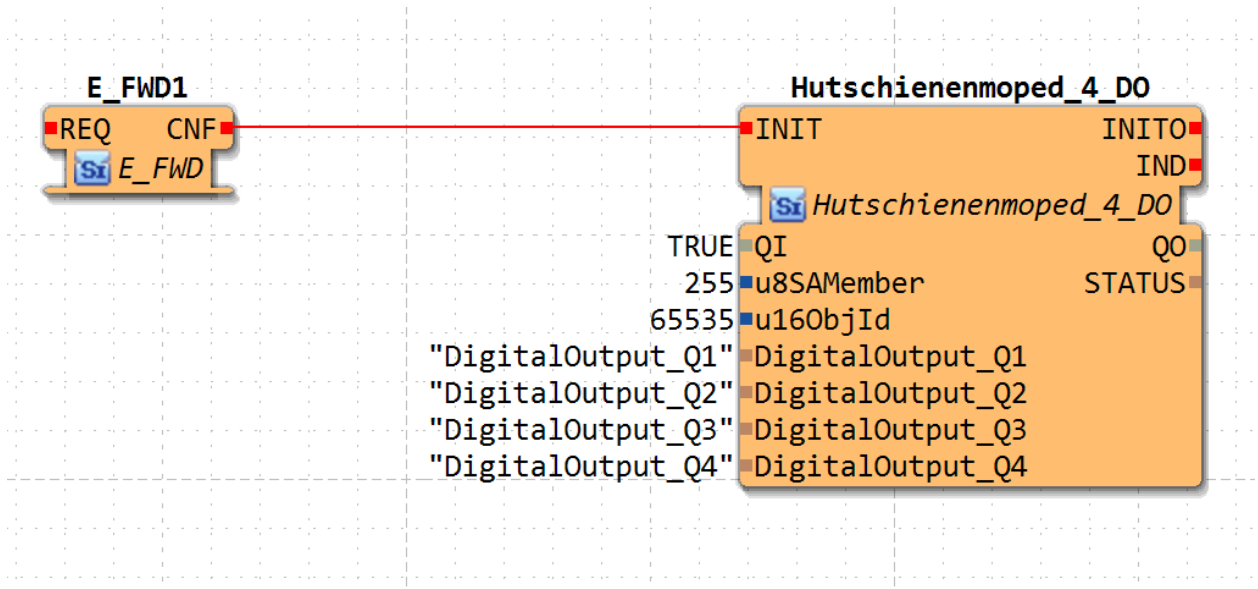
EMB_RES:



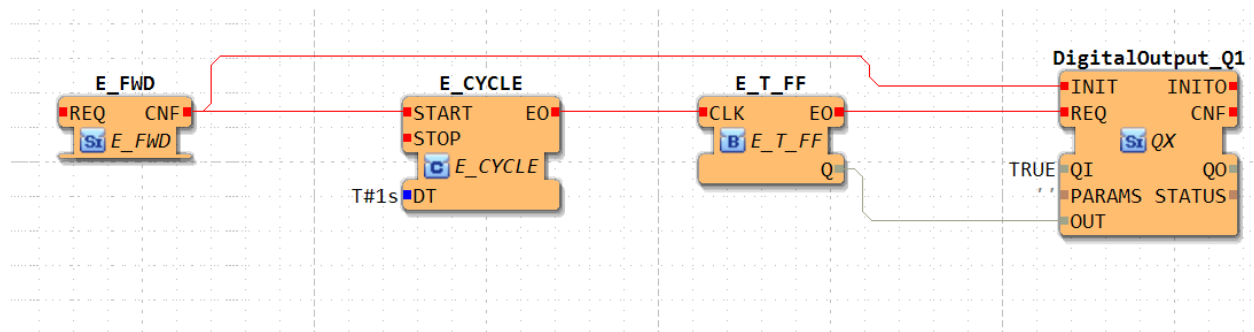
6.2.15.4.29.3 E_T_FF Anwendungen Blinker

mit einem E_T_FF lässt sich so ein Blinker realisieren:

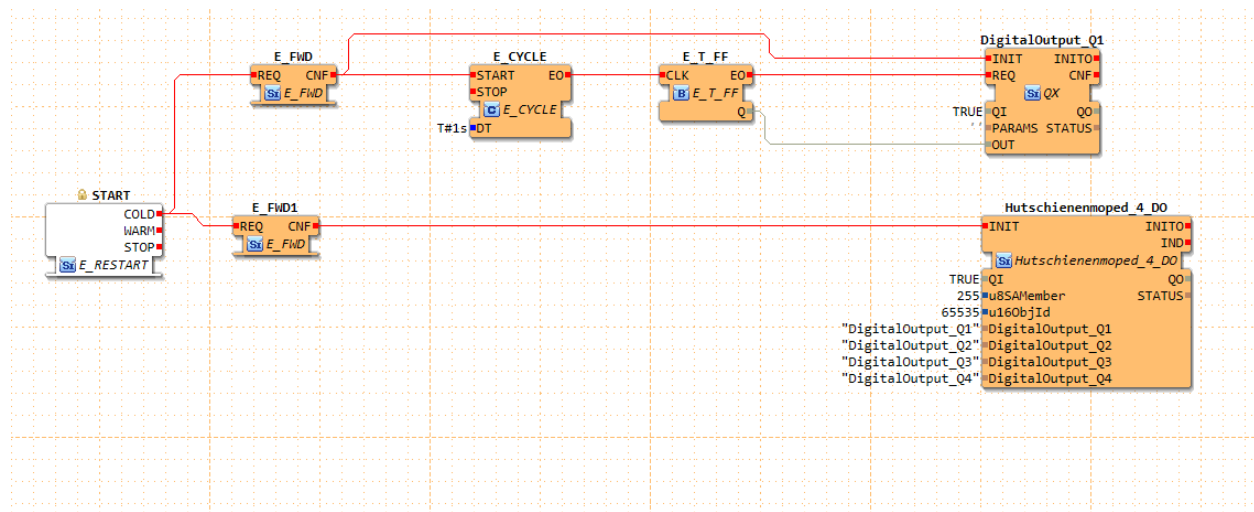
mapping:



App:



EMB_RES:

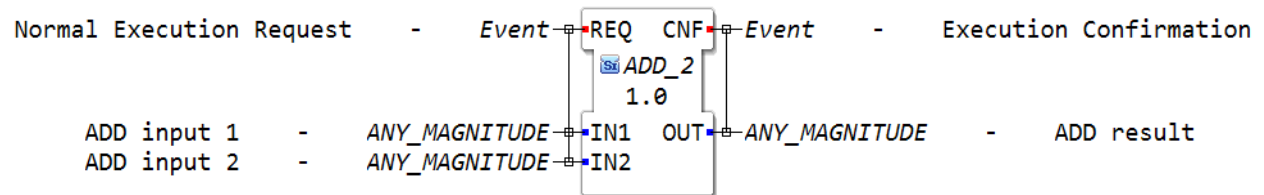


Siehe auch: https://www.holobloc.com/doc/fb/rt/events/E_T_FF.htm

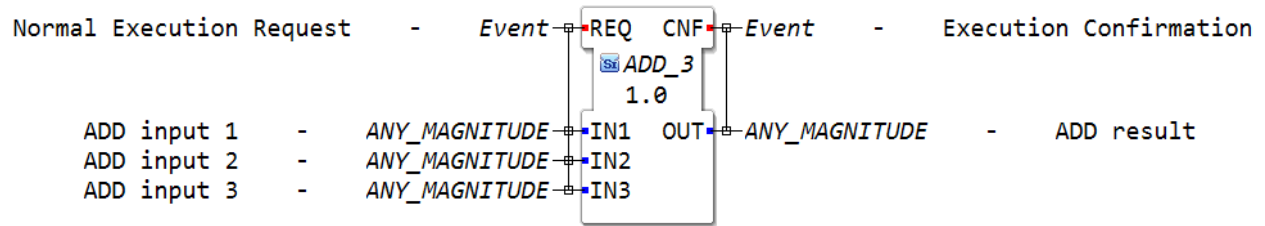
6.2.15.5 iec61131-3

6.2.15.5.1 arithmetic

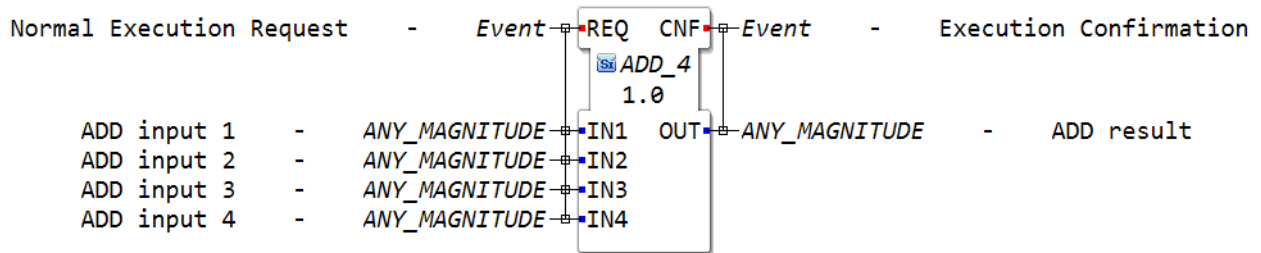
6.2.15.5.1.1 ADD_2



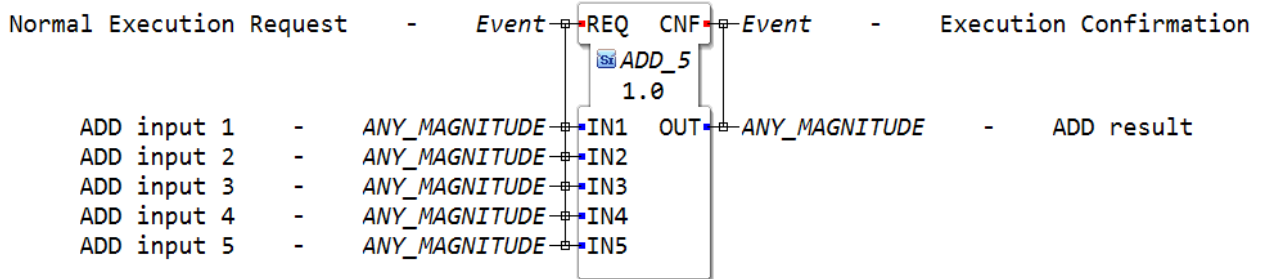
6.2.15.5.1.2 ADD_3



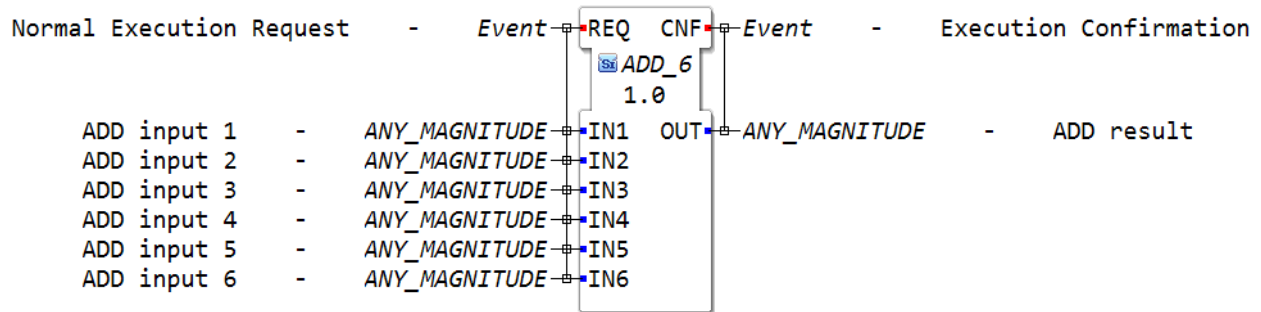
6.2.15.5.1.3 ADD_4



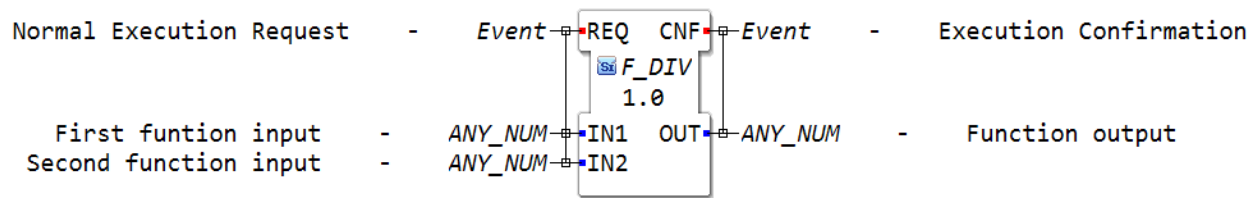
6.2.15.5.1.4 ADD_5



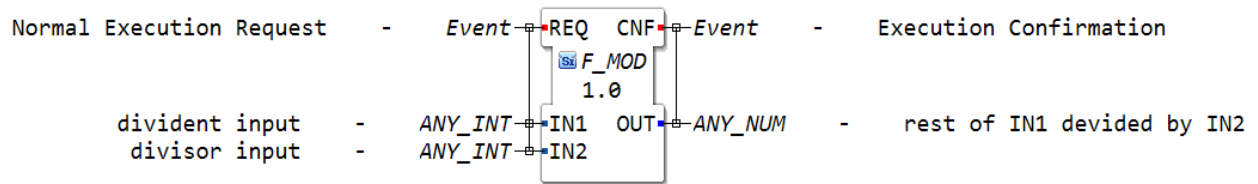
6.2.15.5.1.5 ADD_6



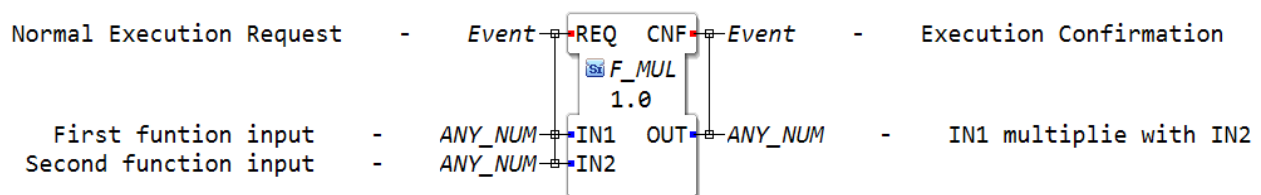
6.2.15.5.1.6 F_DIV



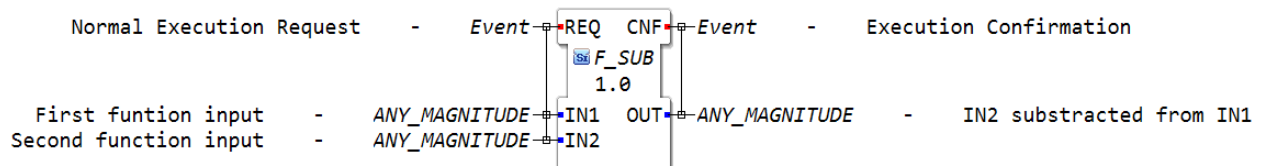
6.2.15.5.1.7 F_MOD



6.2.15.5.1.8 F_MUL

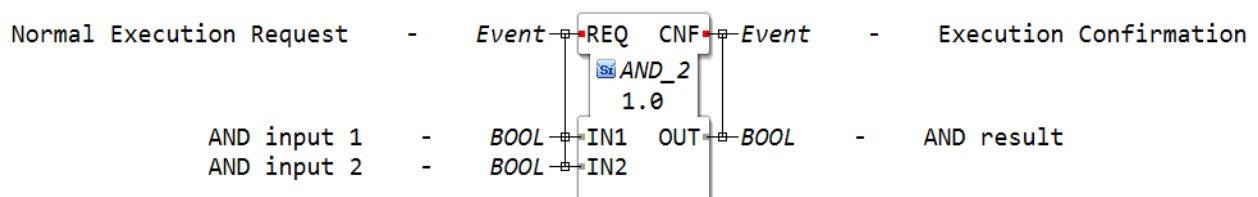


6.2.15.5.1.9 F_SUB

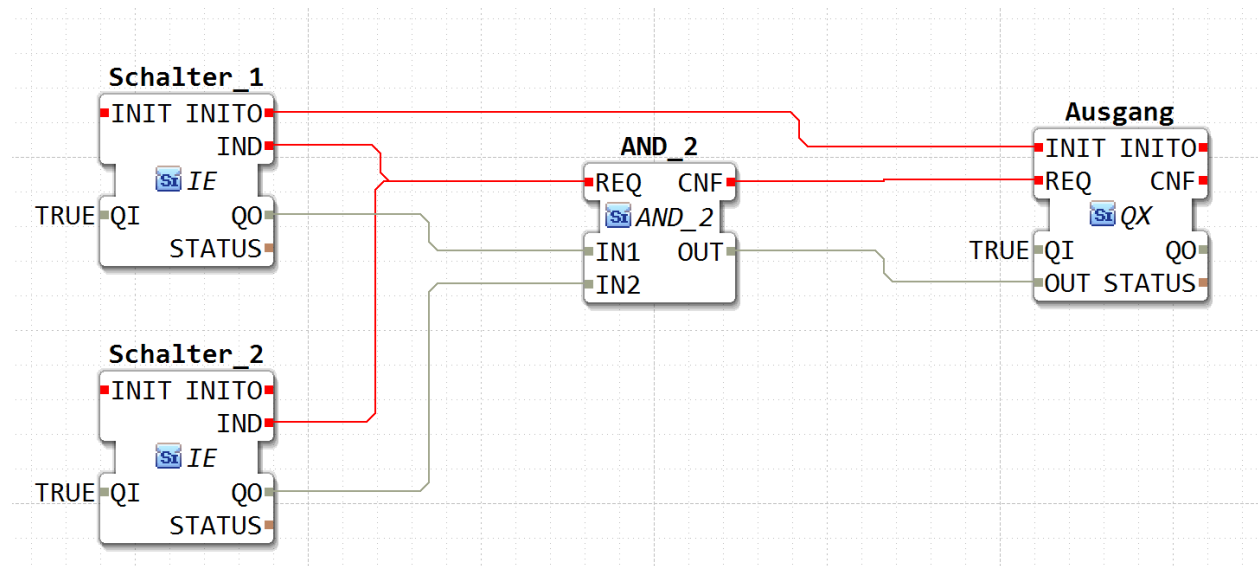


6.2.15.5.2 booleanOperators

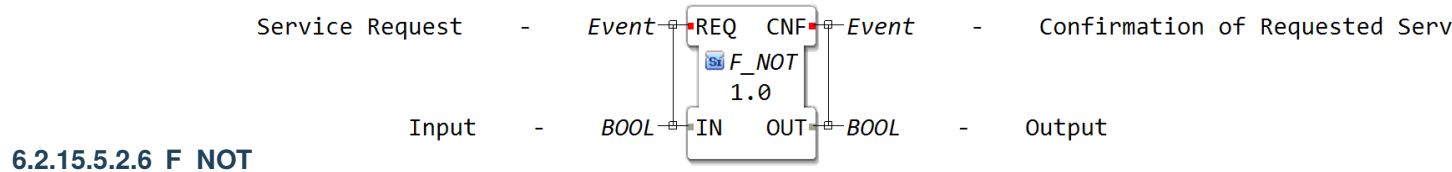
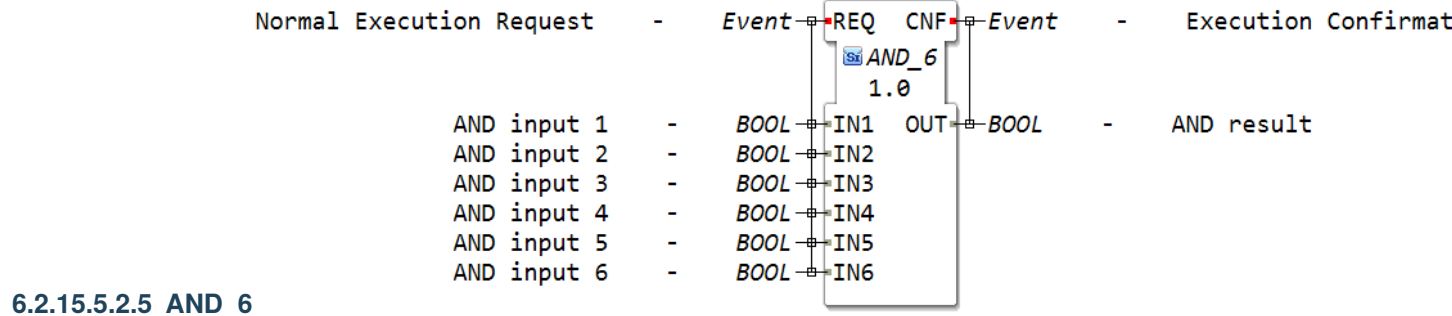
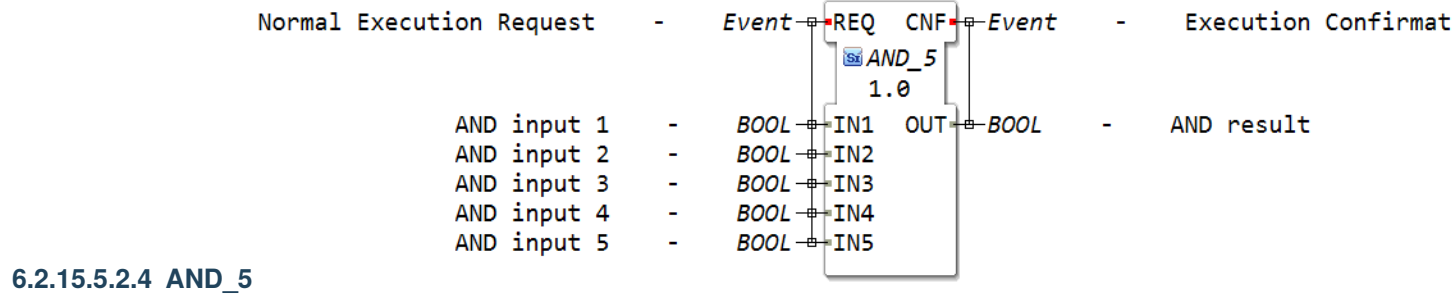
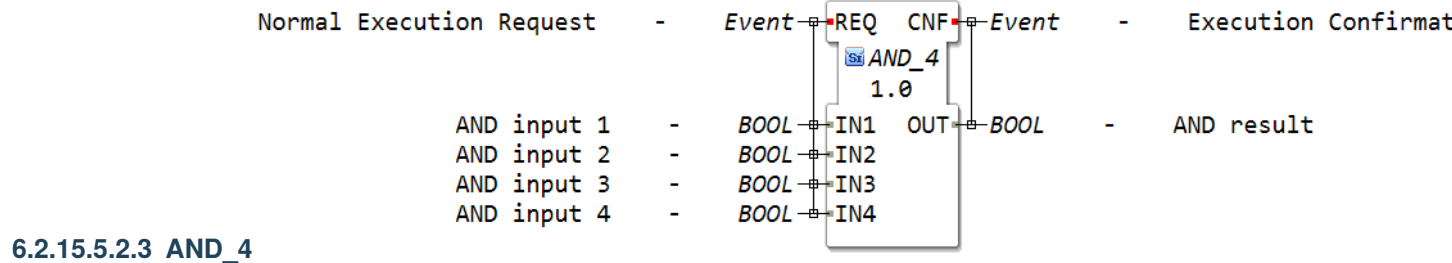
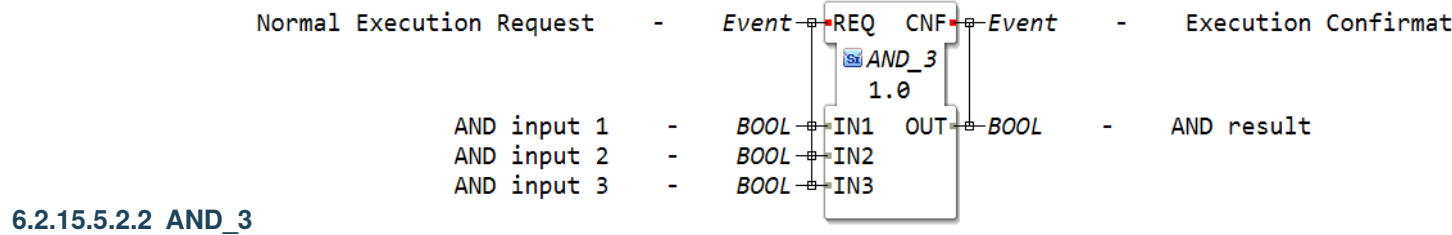
6.2.15.5.2.1 AND_2



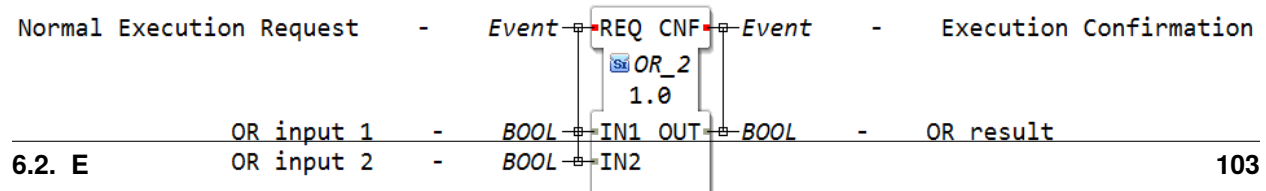
OUT wird nur geschaltet wenn an IN1 und IN2 ein Bool (Signal anliegt) Z.b. von einem Schalter



bei allen anderen „AND“ Bausteinen selbe prinzip nur dass immer alle IN eingänge immer True sein müssen



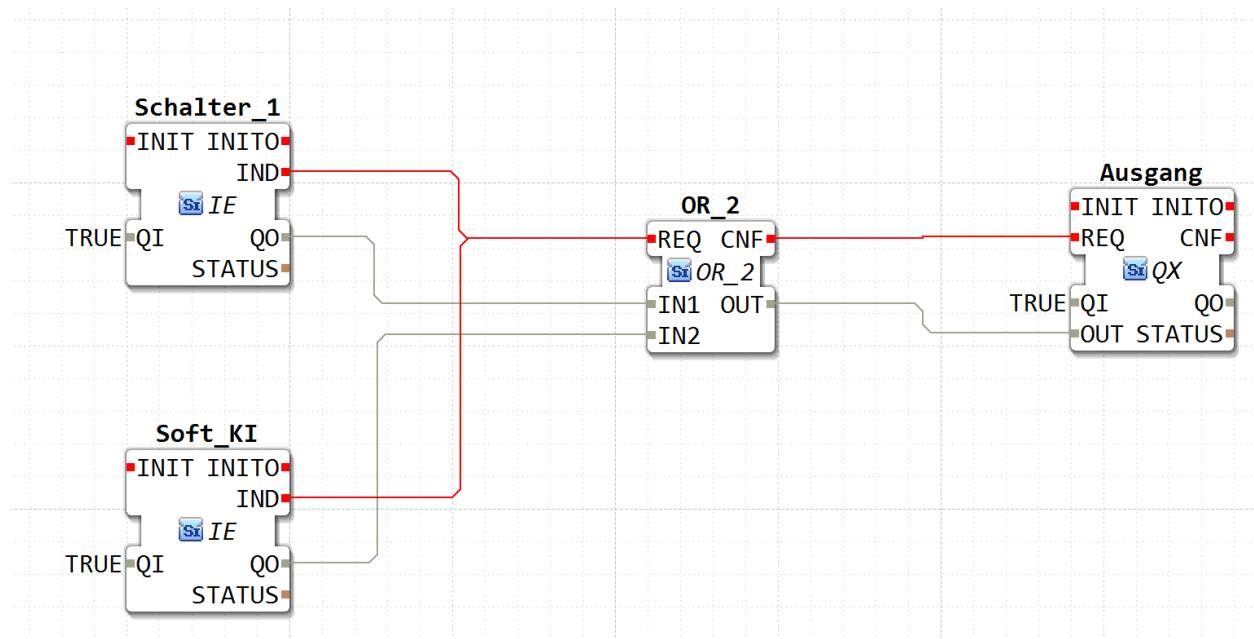
6.2.15.5.2.7 OR_2



Der OR Baustein hat die Aufgabe wenn man mit zwei Schalter einen Ausgang Schalten möchte.

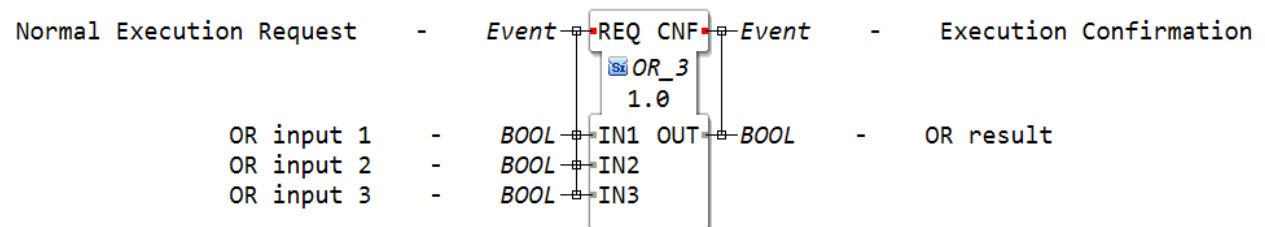
Ein häufiger Anwendungsfall ist ein Soft KI und ein AUX Button

TRUE dominiert, also sobald an einem IN ein TRUE anliegt wird der OUT auf TRUE geschalten

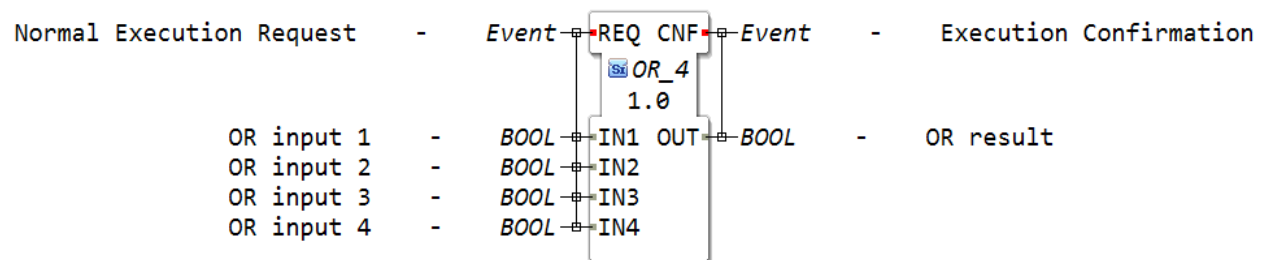


Bei allen weiteren OR Bausteinen ist die gleiche Funktion nur mit mehr Eingängen

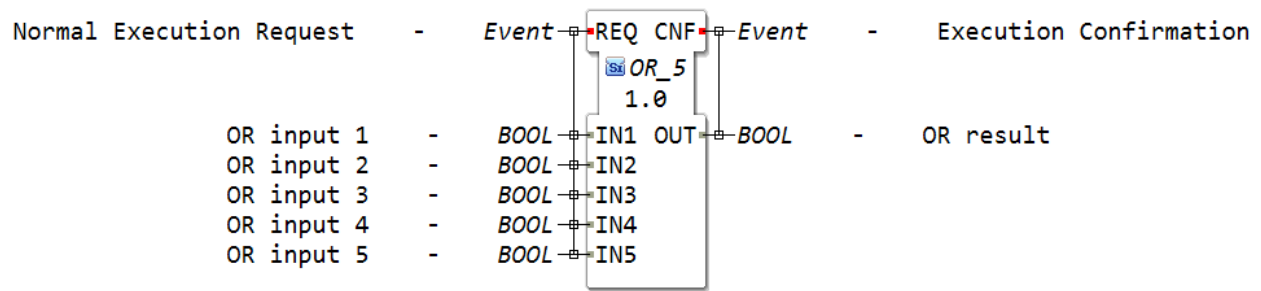
6.2.15.5.2.8 OR_3



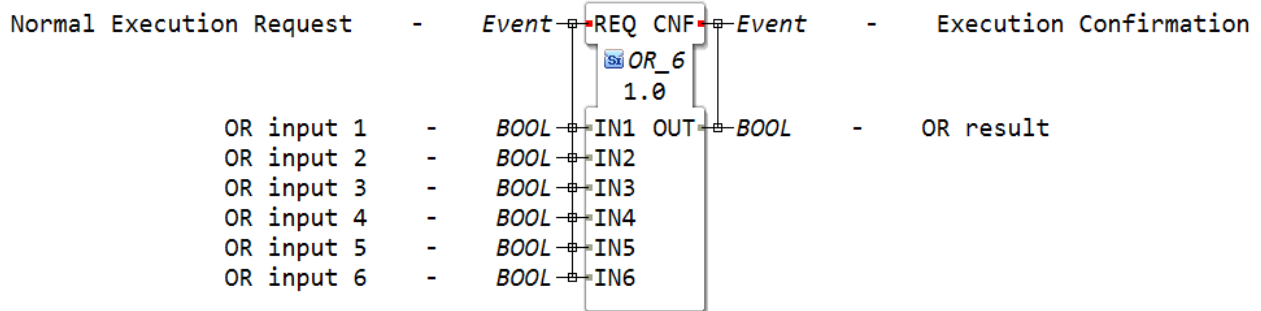
6.2.15.5.2.9 OR_4



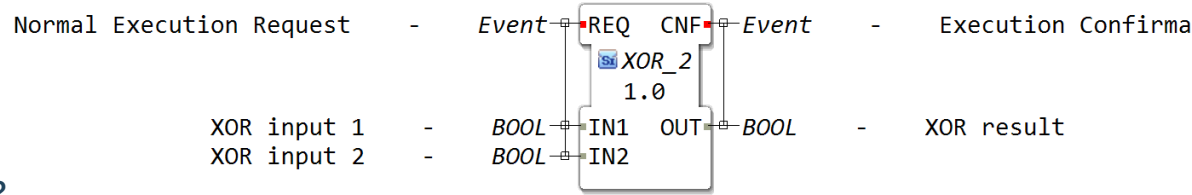
6.2.15.5.2.10 OR_5



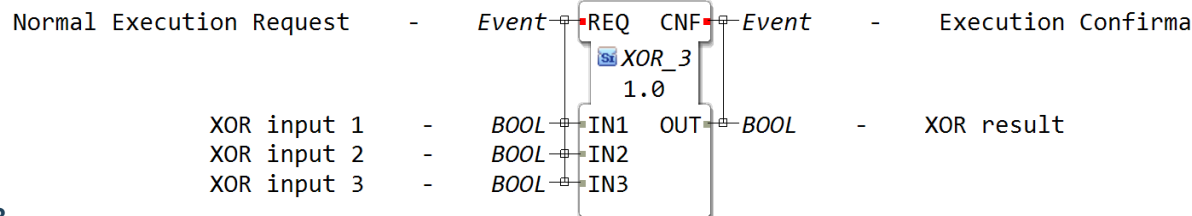
6.2.15.5.2.11 OR_6



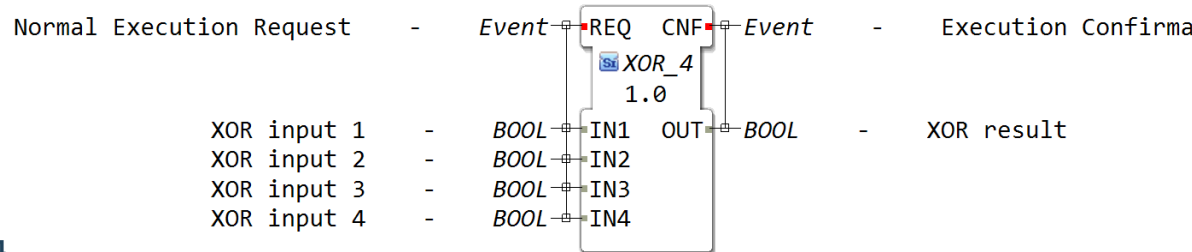
6.2.15.5.2.12 XOR_2



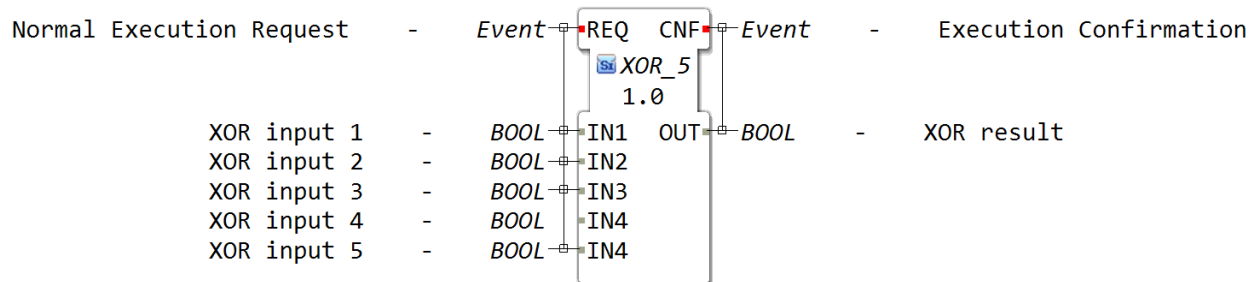
6.2.15.5.2.13 XOR_3

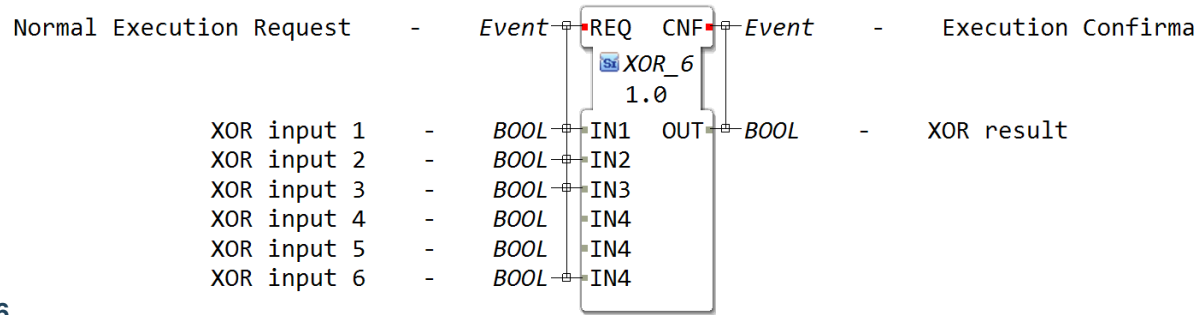


6.2.15.5.2.14 XOR_4



6.2.15.5.2.15 XOR_5

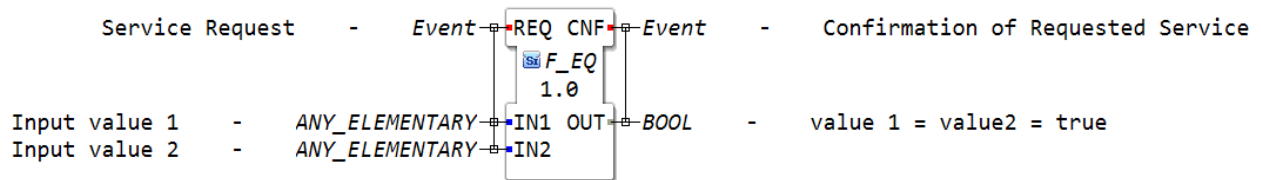




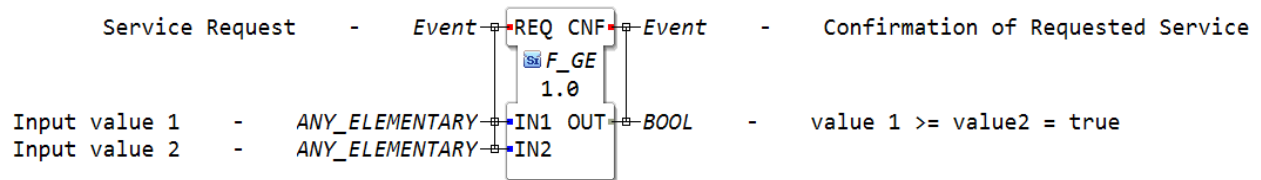
6.2.15.5.2.16 XOR_6

6.2.15.5.3 comparison

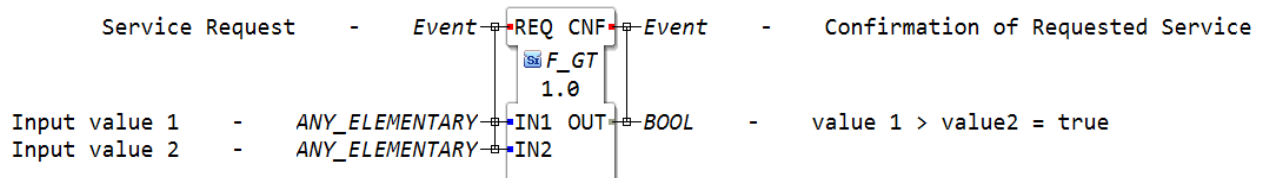
6.2.15.5.3.1 F_EQ



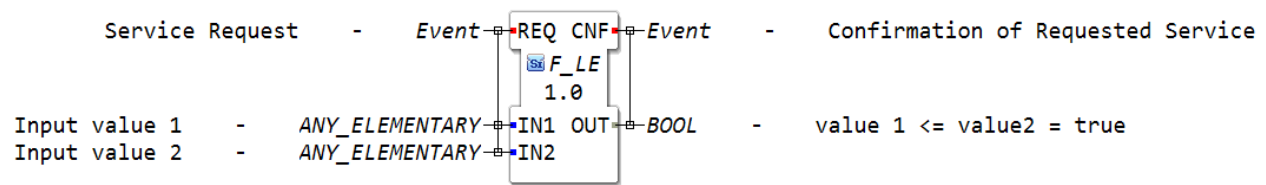
6.2.15.5.3.2 F_GE



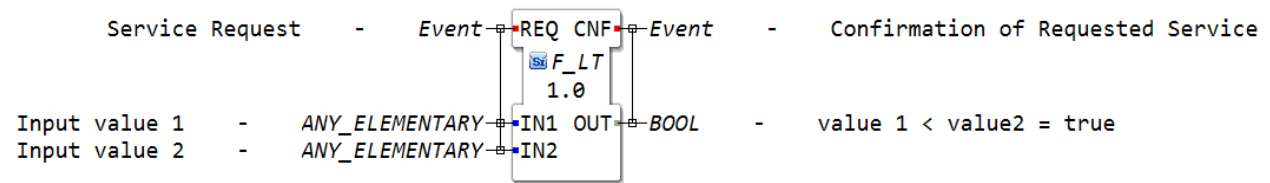
6.2.15.5.3.3 F_GT



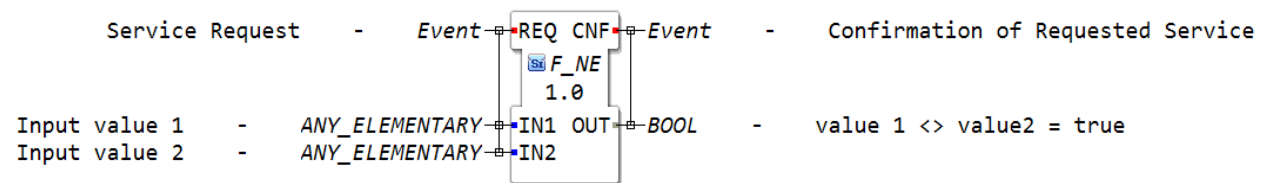
6.2.15.5.3.4 F_LE



6.2.15.5.3.5 F_LT



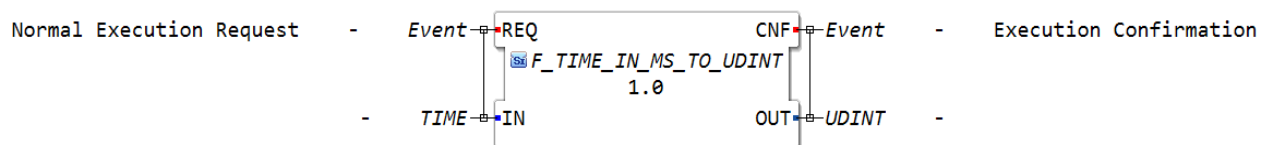
6.2.15.5.3.6 F_NE



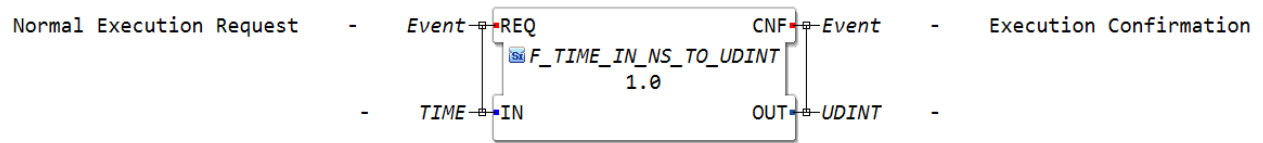
6.2.15.5.4 conversion

6.2.15.5.4.1 TIME

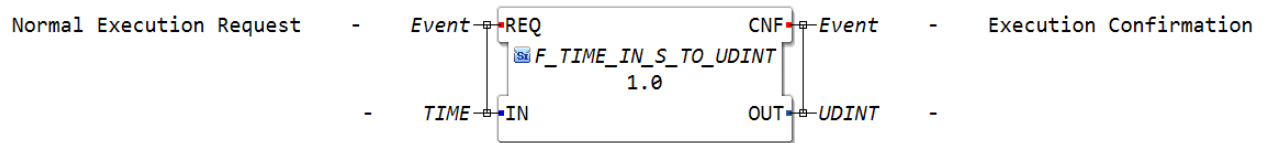
6.2.15.5.4.2 F_TIME_IN_MS_TO_UDINT



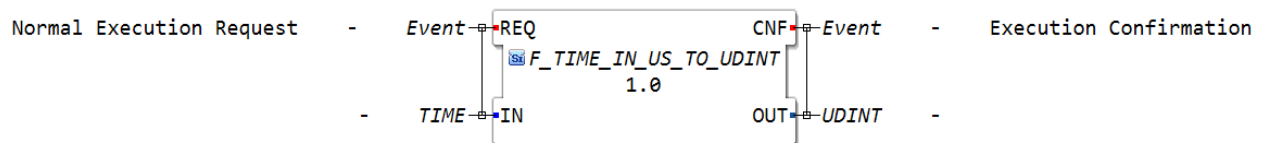
6.2.15.5.4.3 F_TIME_IN_NS_TO_UDINT



6.2.15.5.4.4 F_TIME_IN_S_TO_UDINT

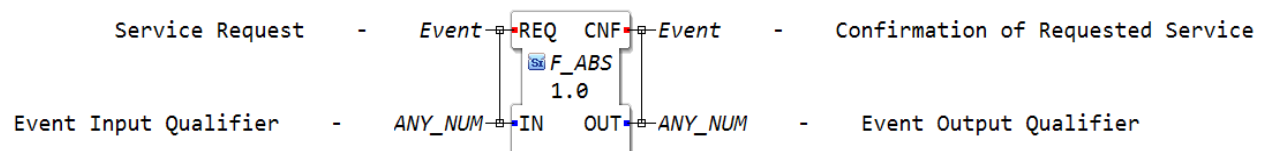


6.2.15.5.4.5 F_TIME_IN_US_TO_UDINT

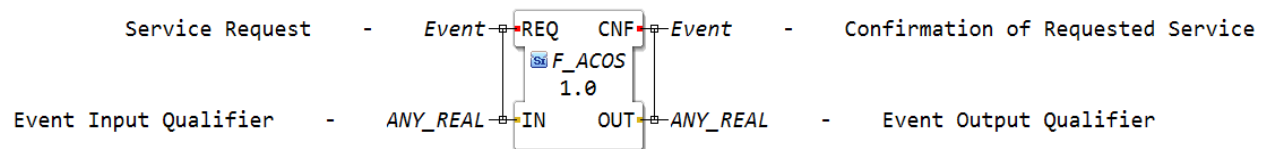


6.2.15.5.5 numerical

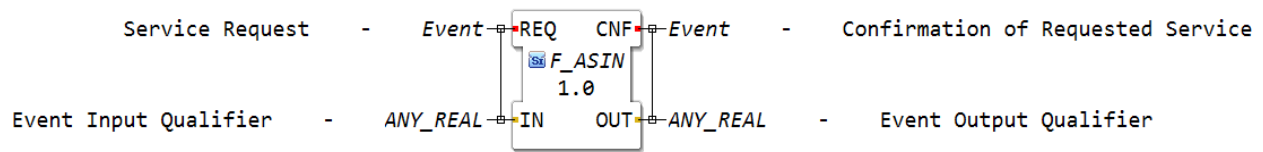
6.2.15.5.5.1 F_ABS



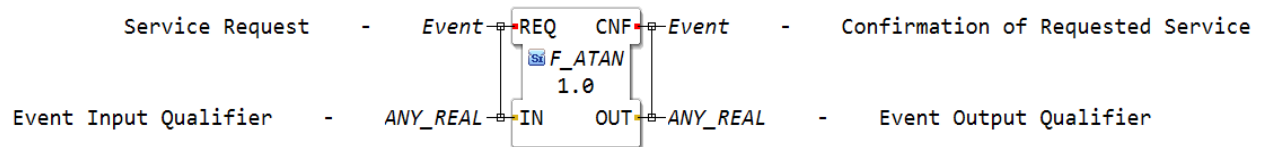
6.2.15.5.5.2 F_ACOS



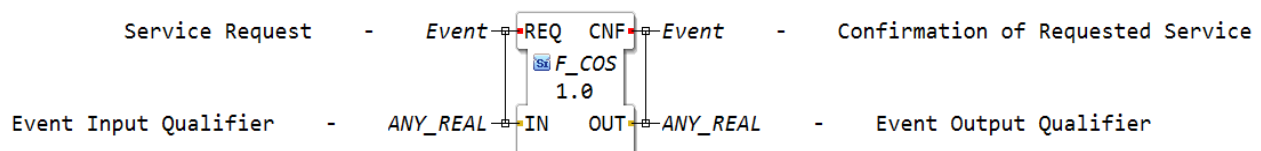
6.2.15.5.3 F_ASIN



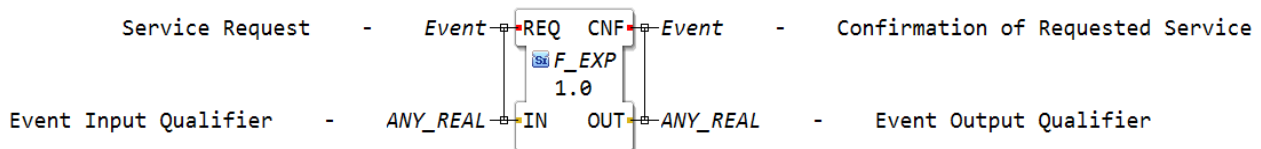
6.2.15.5.4 F_ATAN



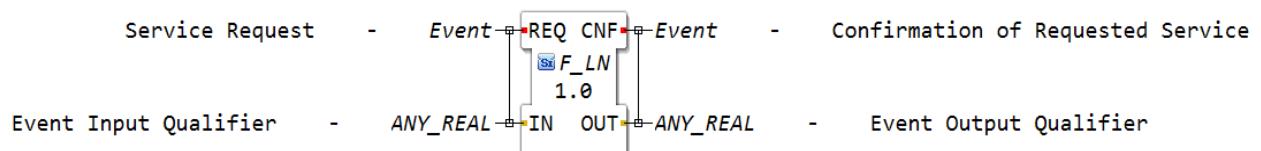
6.2.15.5.5 F_COS



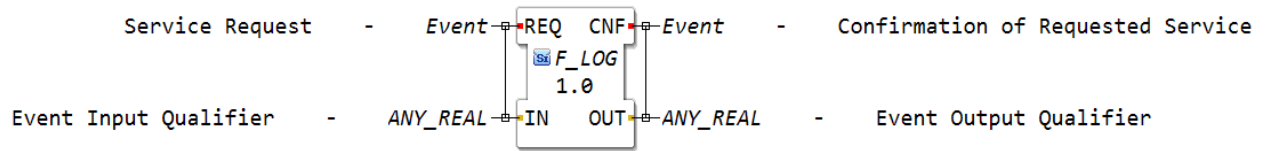
6.2.15.5.6 F_EXP



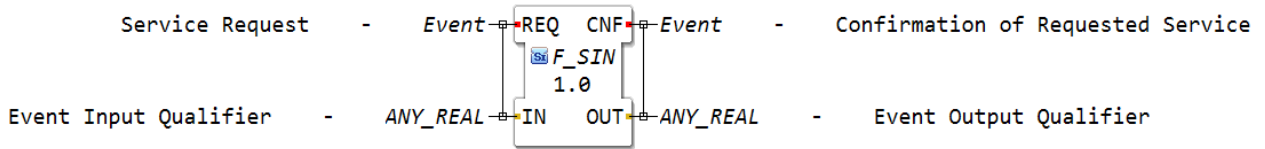
6.2.15.5.7 F_LN



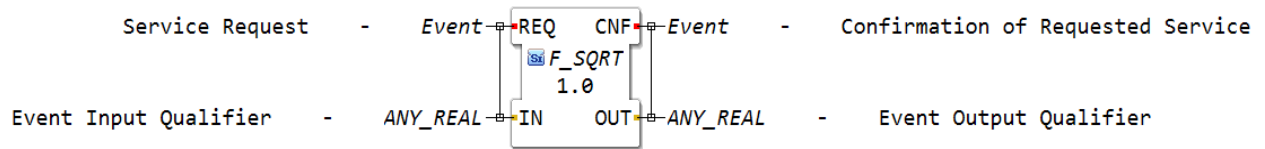
6.2.15.5.8 F_LOG



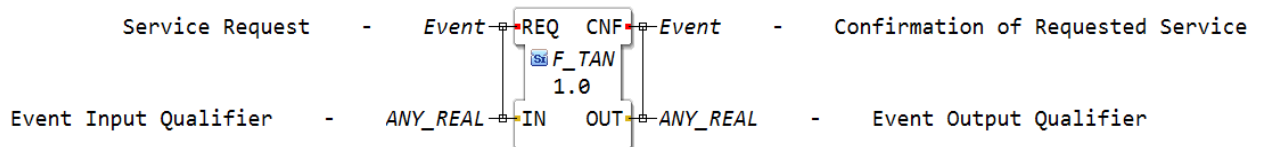
6.2.15.5.9 F_SIN



6.2.15.5.10 F_SQRT



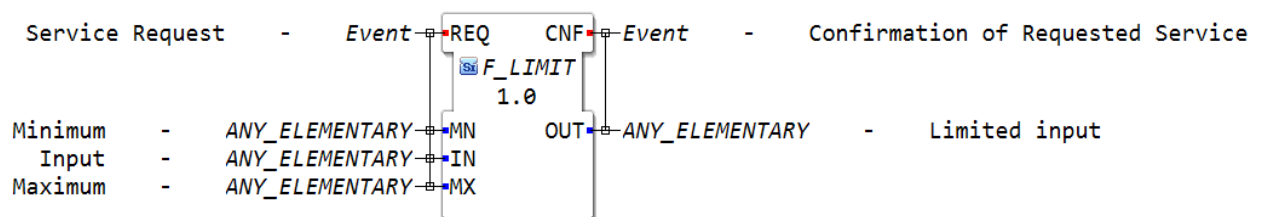
6.2.15.5.11 F_TAN



6.2.15.5.6 selection

6.2.15.5.6.1 F_LIMIT

Aus IEC 61131-3

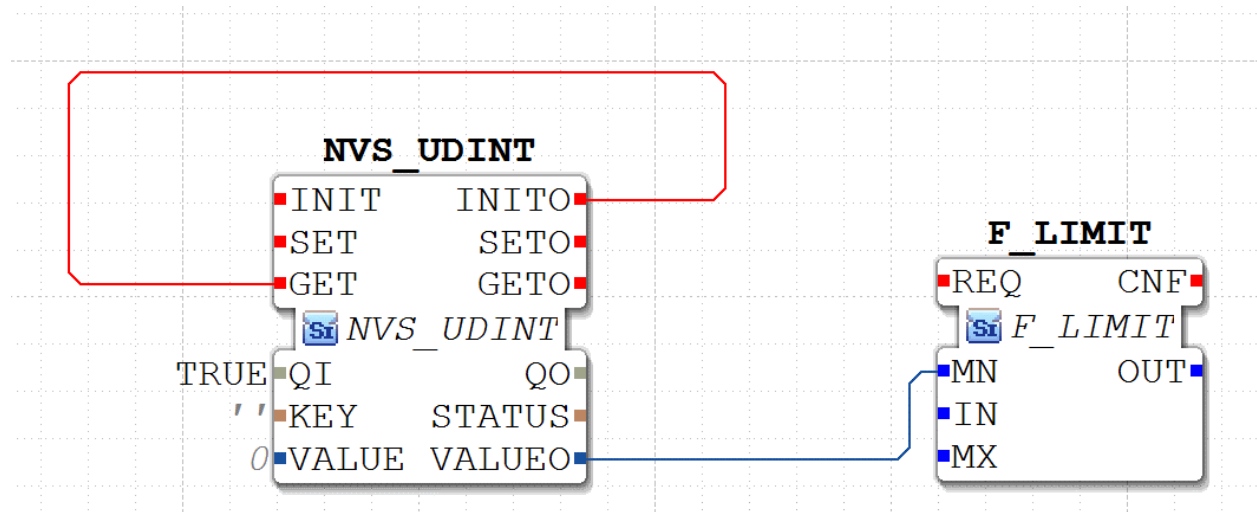


Minimum bedeutet, unterschreitet IN diesen Wert sendet der Baustein das Signal für Wert unterschritten.

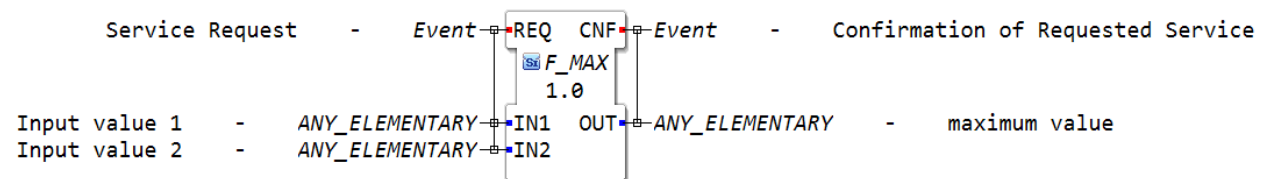
Wert kann fix eingespeichert werden (durch Eingabe DINT#Wert) Bsp. Motortemperatur unter/überschritten. Oder Kraftstoff auf Reserve.

Vorgabewert kann aber auch individuell vom Anwender eingegeben werden. Über Terminal (Bsp. Zielvorgabe Slurry Tanker)

Bsp. Zapfwellendrehzahl festlegen mit Grenzwerten, Ausbringmenge Höchstwert Minimalwert festlegen.



6.2.15.5.6.2 F_MAX



Der Baustein F_MAX dient der maximal Funktion. Er liefert von zwei Werten den größeren Wert aus.

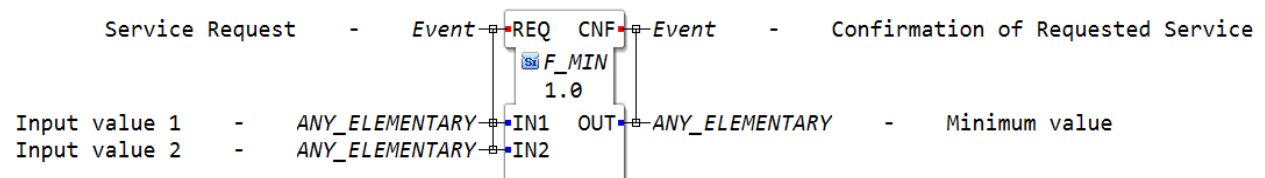
Diese Werte werden bei IN1 und IN2 eingespielt und bei Out dann der größere der beiden Werte ausgegeben.

z.B. IN1= 10

IN2= 15

Out=15

6.2.15.5.6.3 F_MIN



Der Baustein F_MIN dient der minimal Funktion. Er liefert von zwei Werten den kleineren Wert aus.

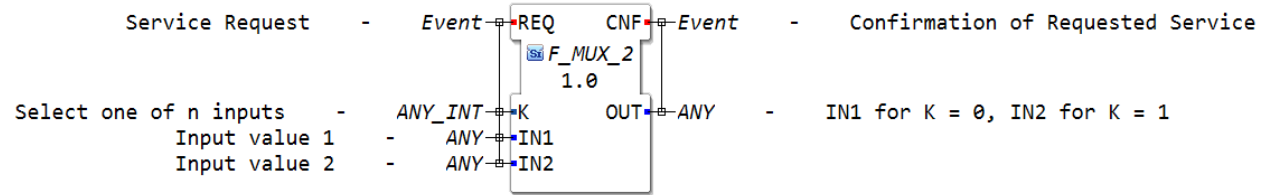
Diese Werte werden bei IN1 und IN2 eingespielt und bei Out wird dann der kleinere der beiden Werte ausgegeben.

z.B. IN1= 10

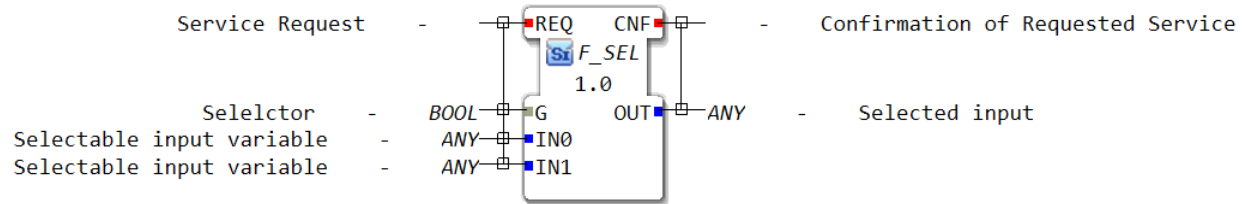
IN2= 15

Out=10

6.2.15.5.6.4 F_MUX_2



6.2.15.5.6.5 F_SEL



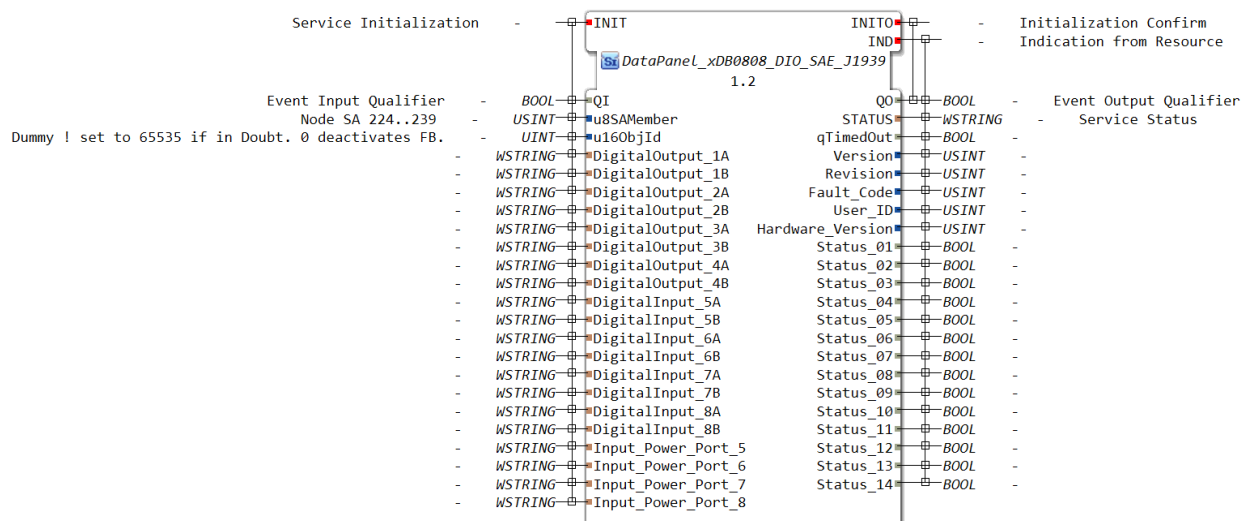
https://help.codesys.com/webapp/_cds_operator_sel;product=codesys;version=3.5.17.0

https://content.helpme-codesys.com/de/CODESYS%20Development%20System/_cds_operator_sel.html

6.2.15.6 io

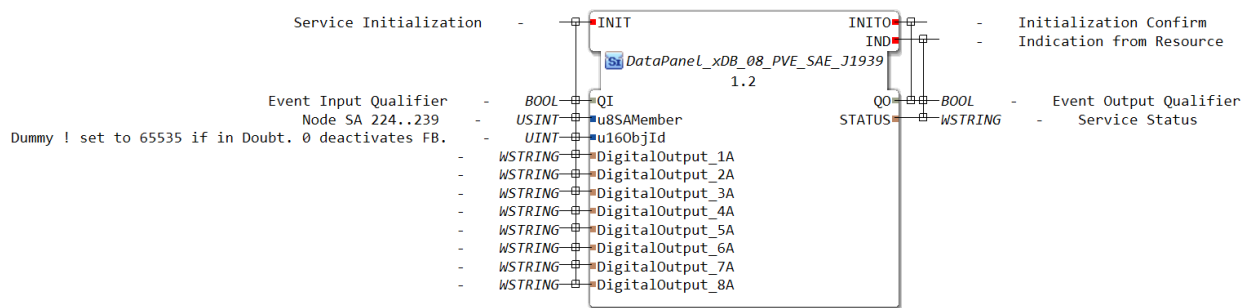
6.2.15.6.1 DataPanel

6.2.15.6.1.1 DataPanel_xDB0808_DIO_SAE_J1939



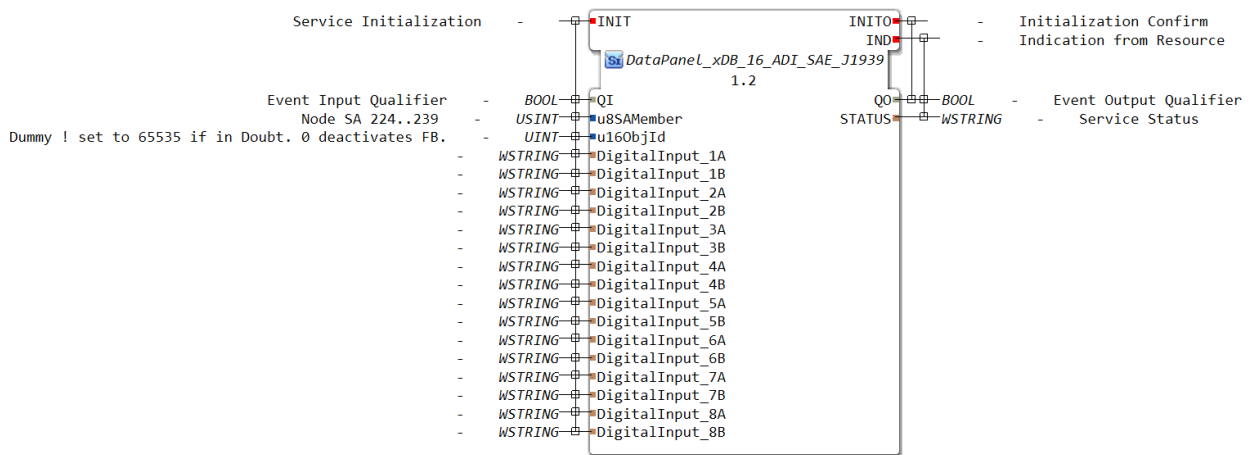
this maps to *IX* / *QX*

6.2.15.6.1.2 DataPanel_xDB_08_PVE_SAE_J1939



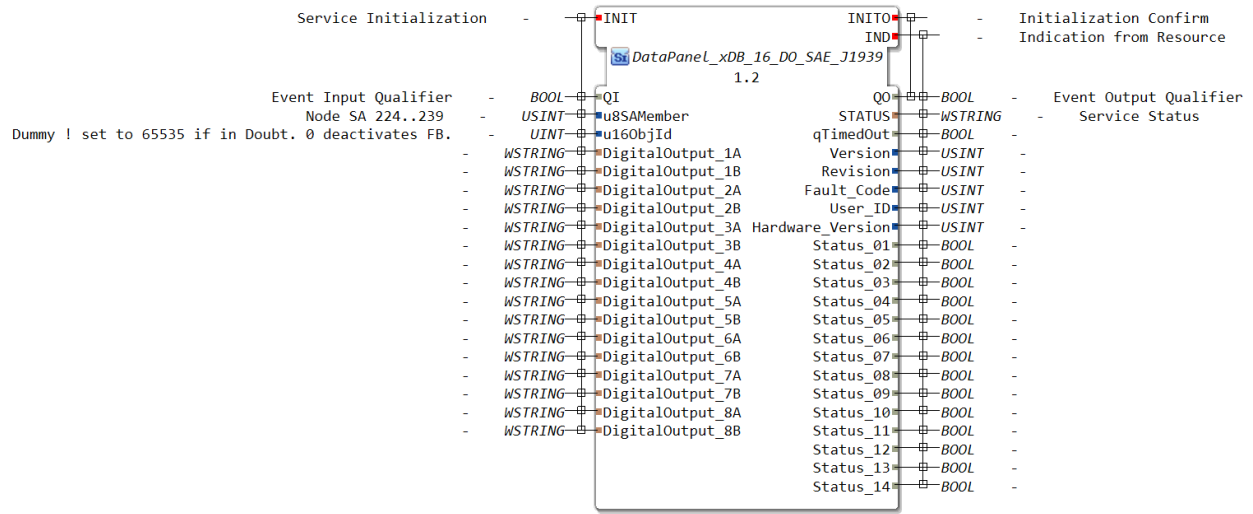
this maps to *QX*

6.2.15.6.1.3 DataPanel_xDB_16_ADI_SAE_J1939



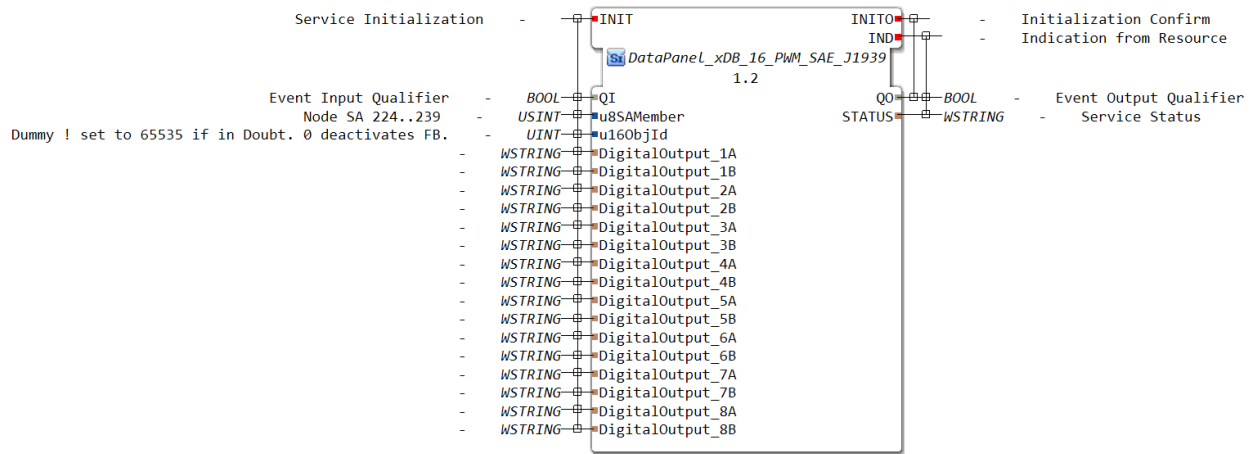
this maps to *IX*

6.2.15.6.1.4 DataPanel_xDB_16_DO_SAE_J1939



this maps to *QX*

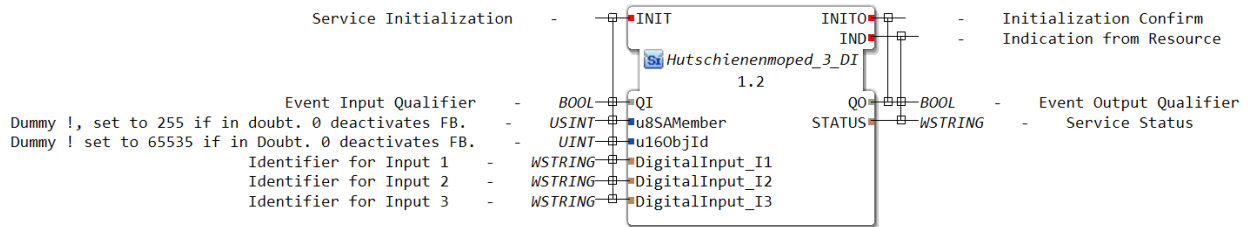
6.2.15.6.1.5 DataPanel_xDB_16_PWM_SAE_J1939



this maps to *QX*

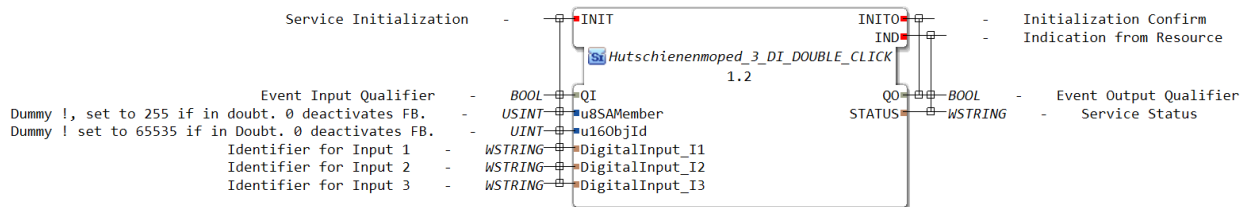
6.2.15.6.2 io esp32 Taster/Ausgänge Hutschienenmopped

6.2.15.6.2.1 Hutschienenmoped_3_DI



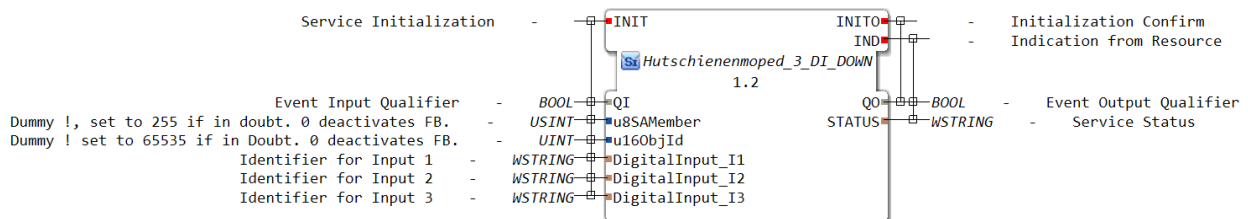
this maps to *IX*

6.2.15.6.2.2 Hutschienenmoped_3_DI_DOUBLE_CLICK



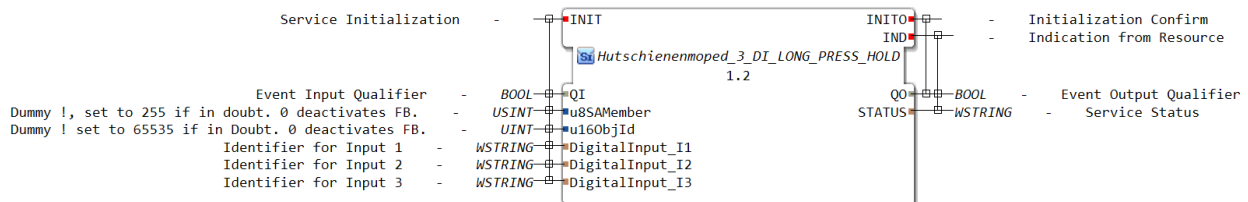
this maps to *IE*

6.2.15.6.2.3 Hutschienenmoped_3_DI_DOWN



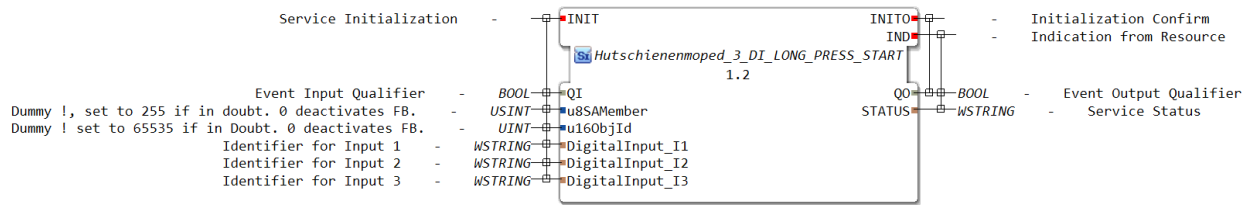
this maps to *IE*

6.2.15.6.2.4 Hutschienenmoped_3_DI_LONG_PRESS_HOLD



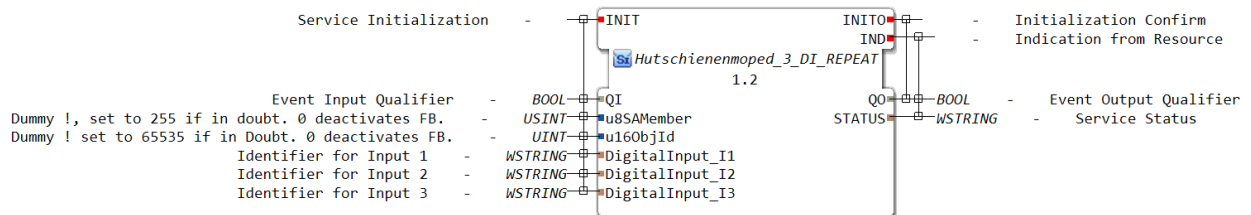
this maps to *IE*

6.2.15.6.2.5 Hutschienenmoped_3_DI_LONG_PRESS_START



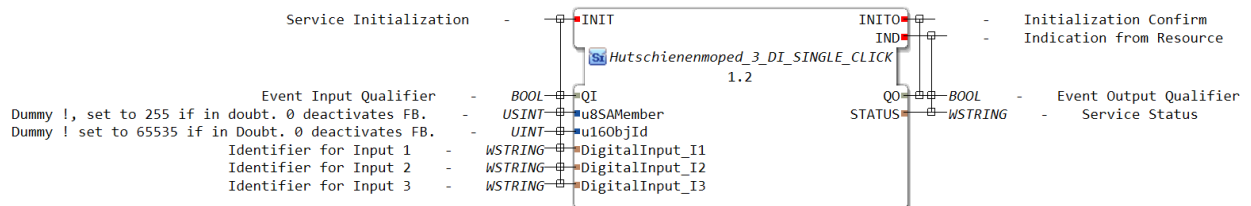
this maps to *IE*

6.2.15.6.2.6 Hutschienenmoped_3_DI_REPEAT



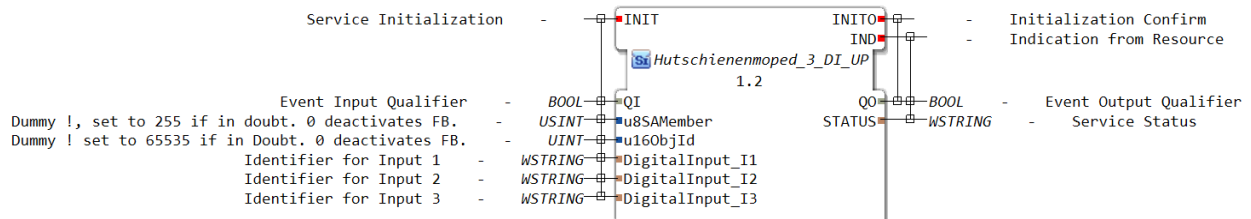
this maps to *IB*

6.2.15.6.2.7 Hutschienenmoped_3_DI_SINGLE_CLICK



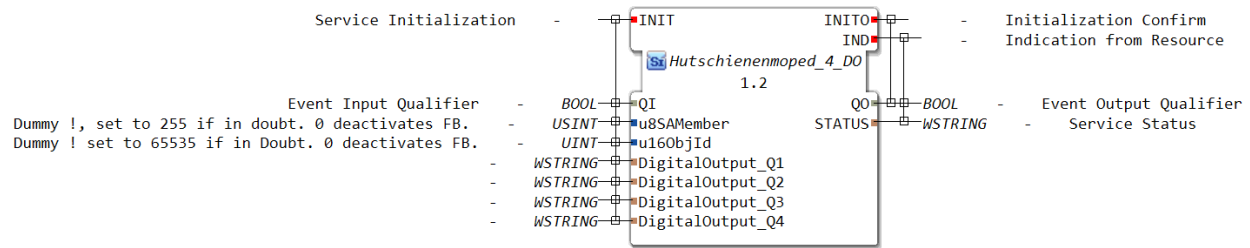
this maps to *IE*

6.2.15.6.2.8 Hutschienenmoped_3_DI_UP



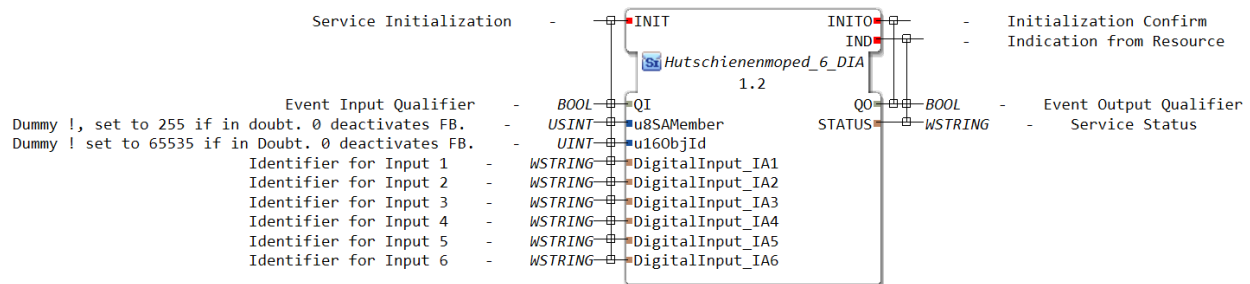
this maps to *IE*

6.2.15.6.2.9 Hutschienenmoped_4_DO



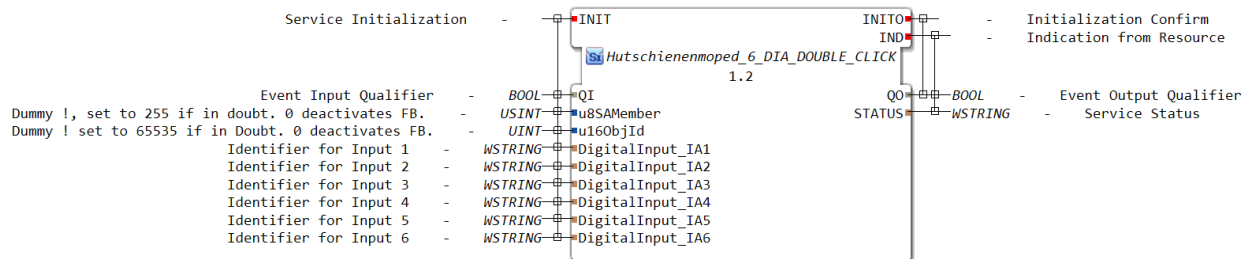
this maps to *QX*

6.2.15.6.2.10 Hutschienenmoped_6_DIA



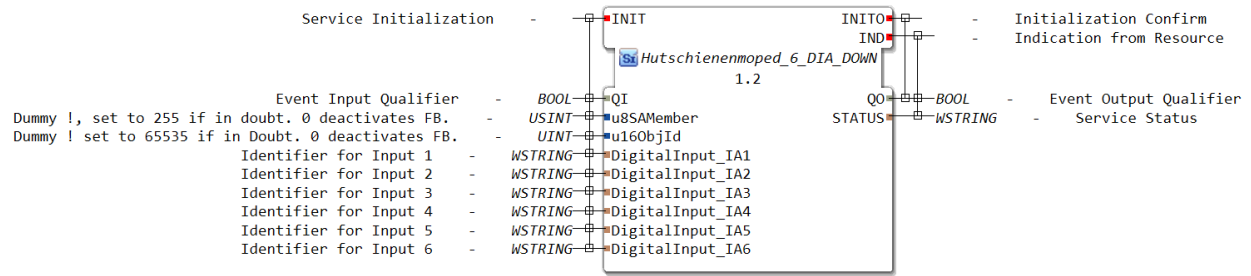
this maps to *IX*

6.2.15.6.2.11 Hutschienenmoped_6_DIA_DOUBLE_CLICK



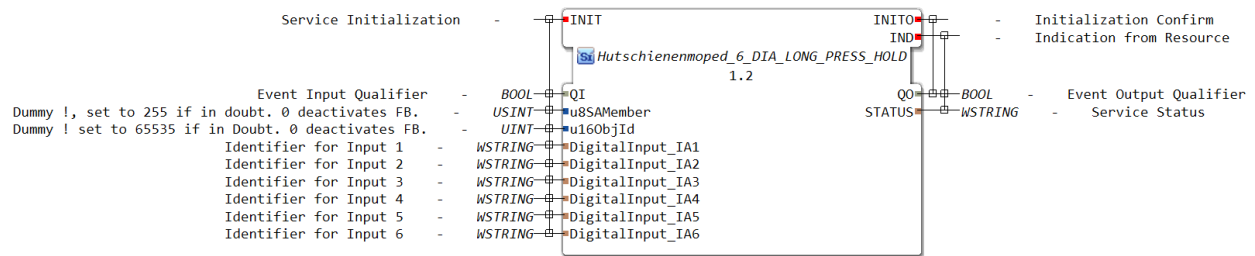
this maps to *IE*

6.2.15.6.2.12 Hutschienenmoped_6_DIA_DOWN



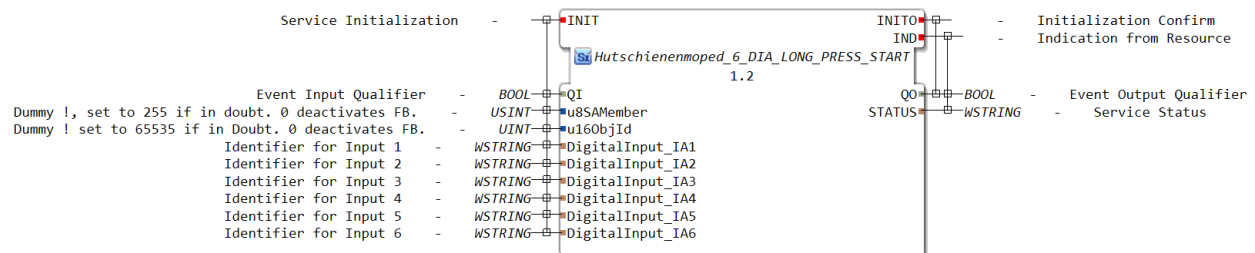
this maps to *IE*

6.2.15.6.2.13 Hutschienenmoped_6_DIA_LONG_PRESS_HOLD



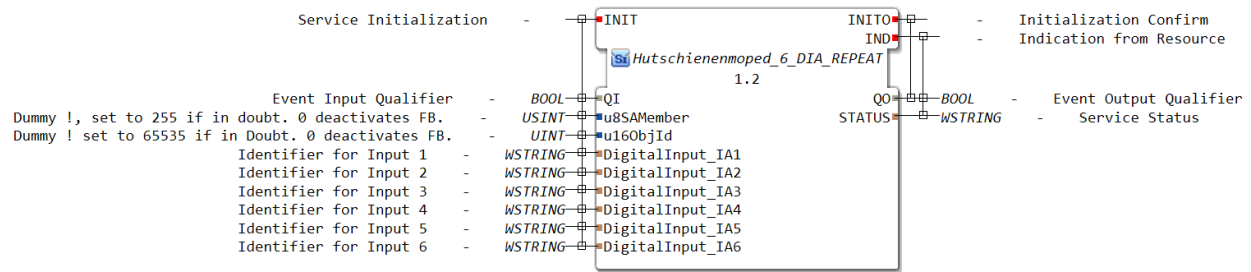
this maps to *IE*

6.2.15.6.2.14 Hutschienenmoped_6_DIA_LONG_PRESS_START



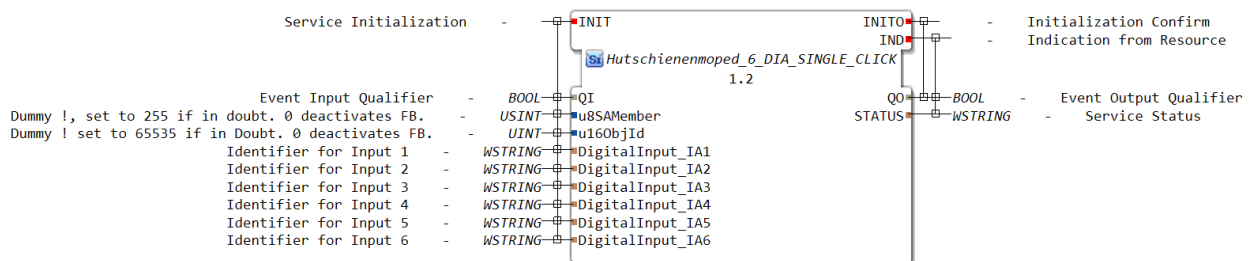
this maps to *IE*

6.2.15.6.2.15 Hutschienenmoped_6_DIA_REPEAT



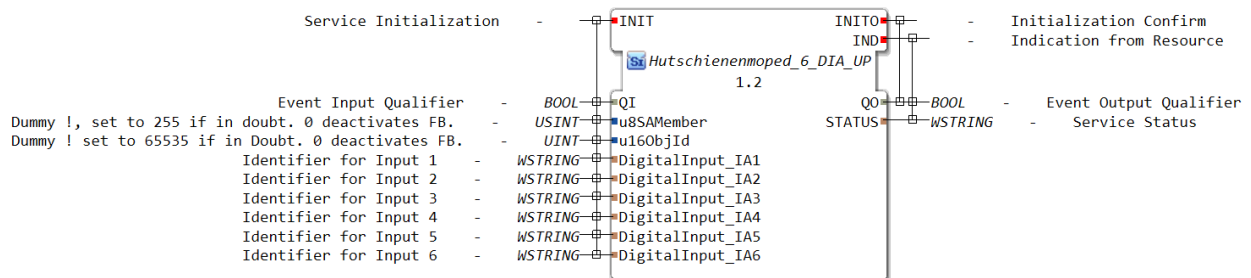
this maps to *IB*

6.2.15.6.2.16 Hutschienenmoped_6_DIA_SINGLE_CLICK



this maps to *IE*

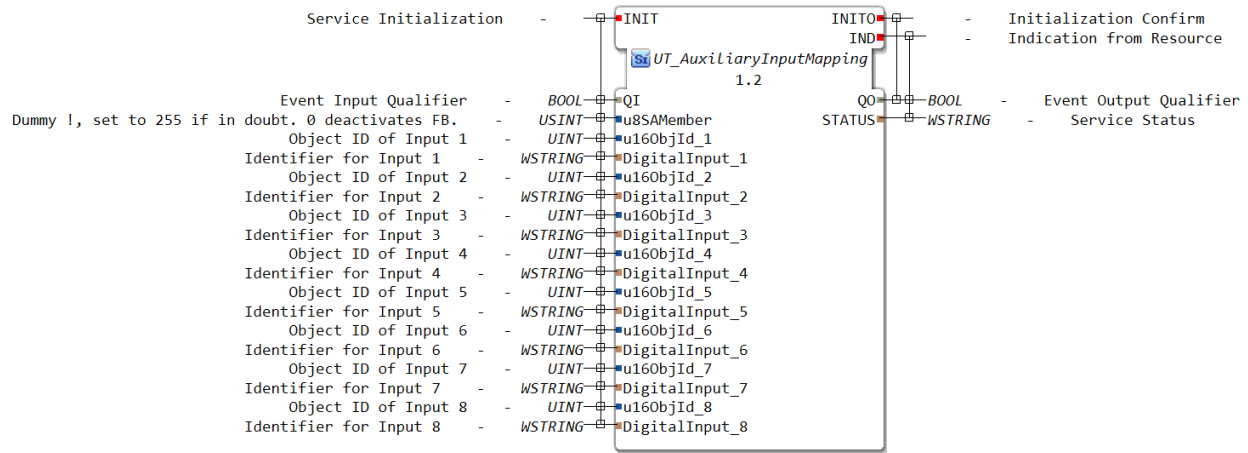
6.2.15.6.2.17 Hutschienenmoped_6_DIA_UP



this maps to *IE*

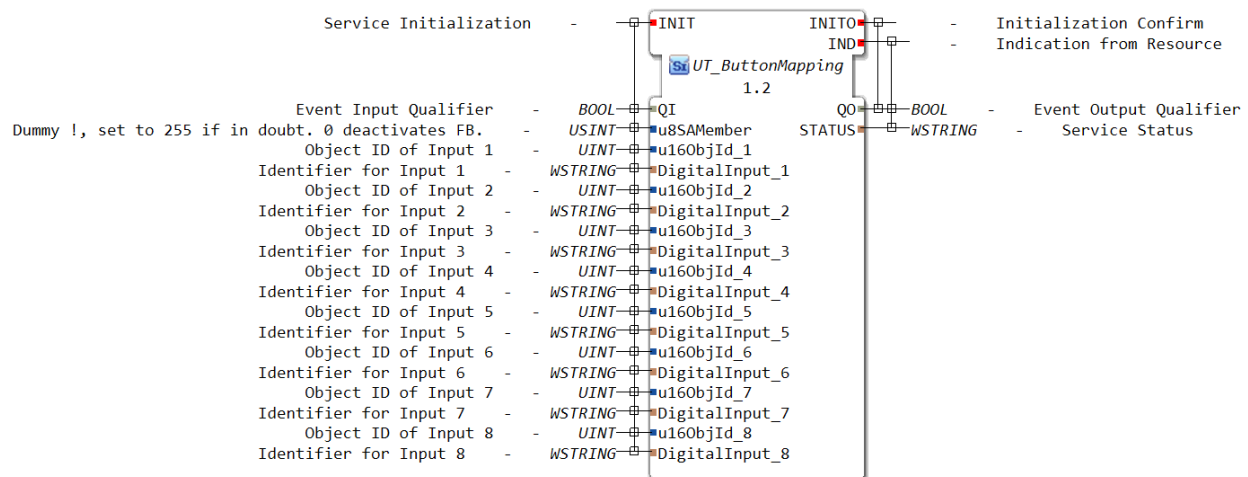
6.2.15.6.3 io isobus

6.2.15.6.3.1 UT_AuxiliaryInputMapping



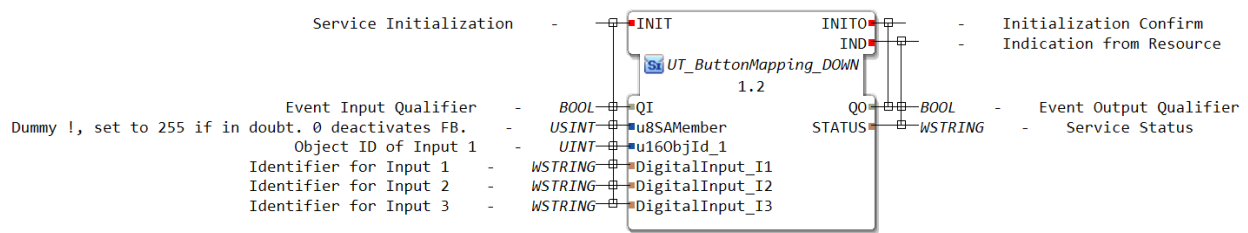
this maps to *IX*

6.2.15.6.3.2 UT_ButtonMapping



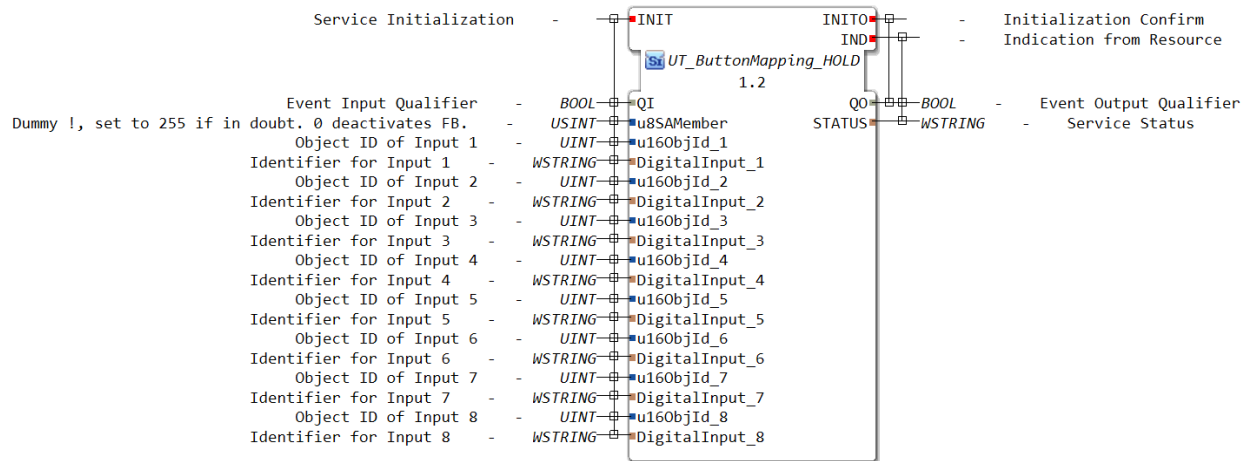
this maps to *IX*

6.2.15.6.3.3 UT_ButtonMapping_DOWN



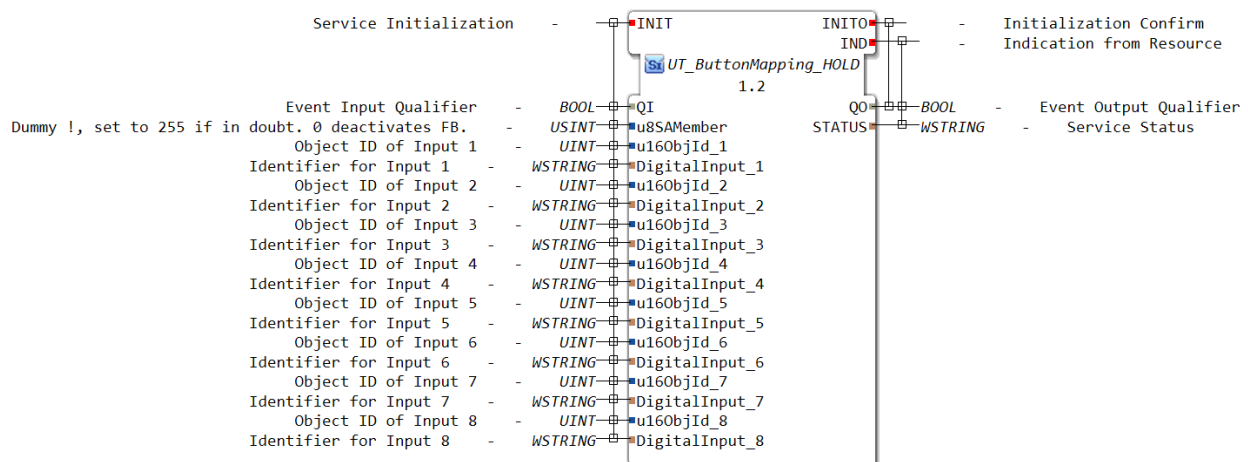
this maps to *IE*

6.2.15.6.3.4 UT_ButtonMapping_HOLD



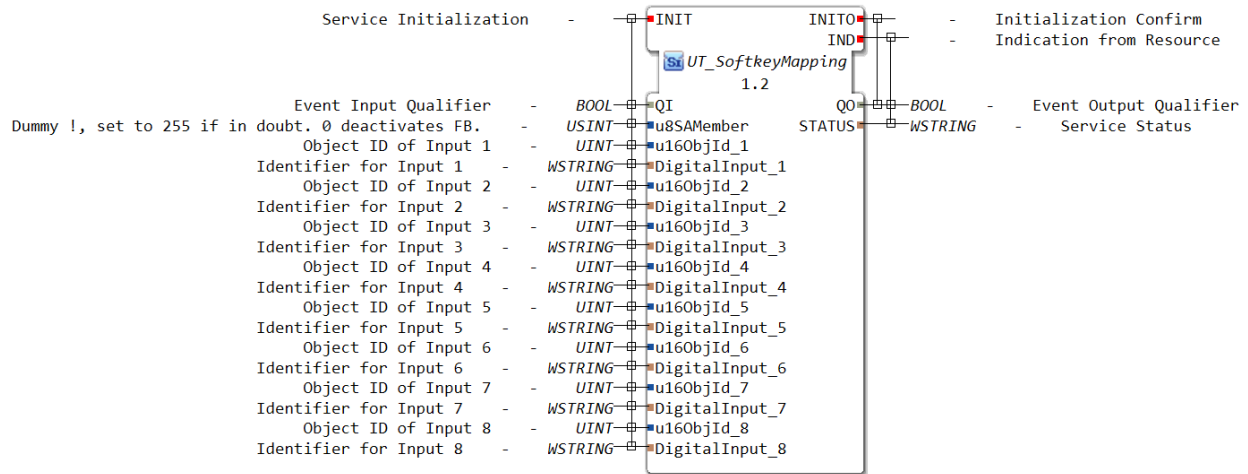
this maps to *IE*

6.2.15.6.3.5 UT_ButtonMapping_UP



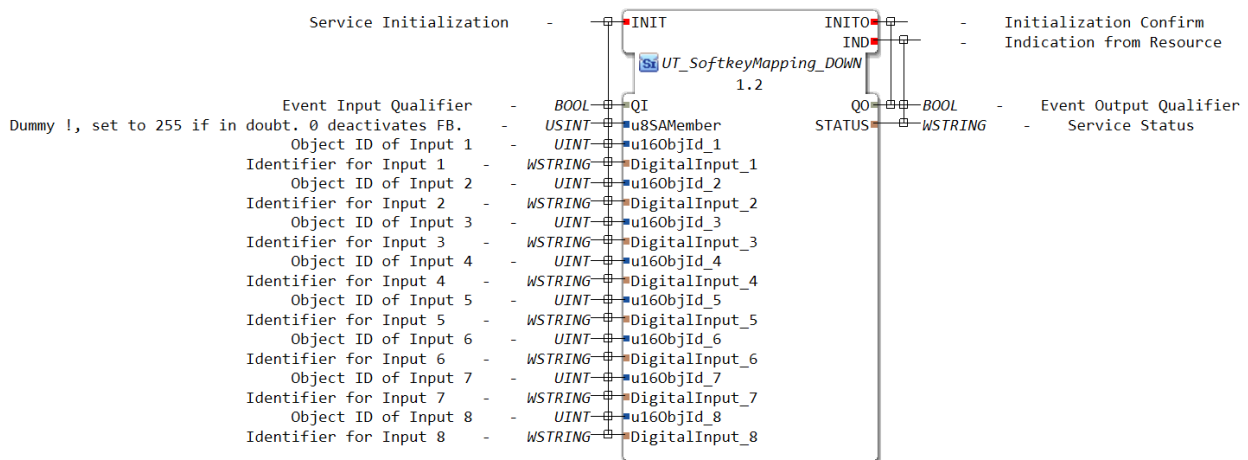
this maps to *IE*

6.2.15.6.3.6 UT_SoftkeyMapping Softkey Taster am Isobus Display



this maps to *IX*

6.2.15.6.3.7 UT_SoftkeyMapping_DOWN



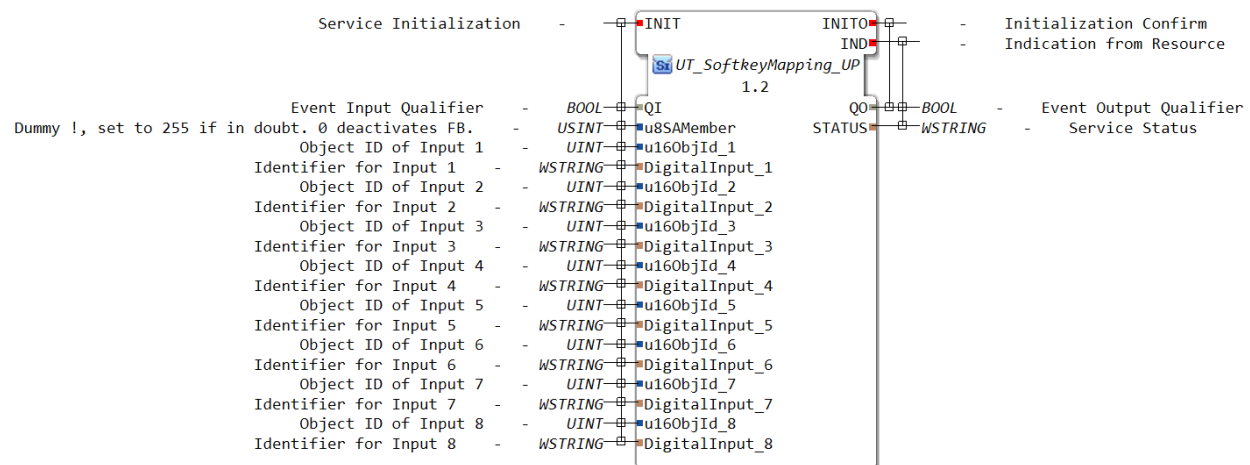
this maps to *IE*

6.2.15.6.3.8 UT_SoftkeyMapping_HOLD



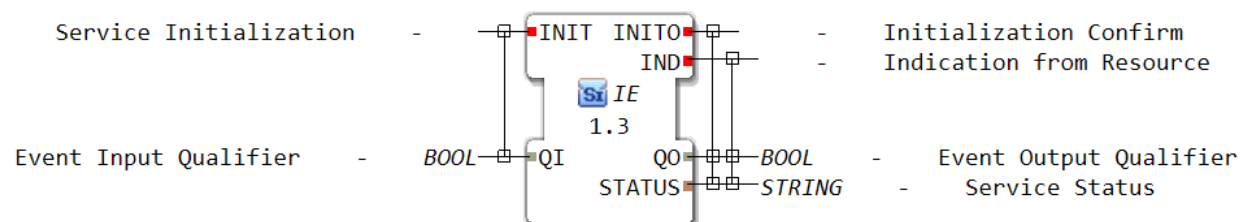
this maps to *IE*

6.2.15.6.3.9 UT_SoftkeyMapping_UP



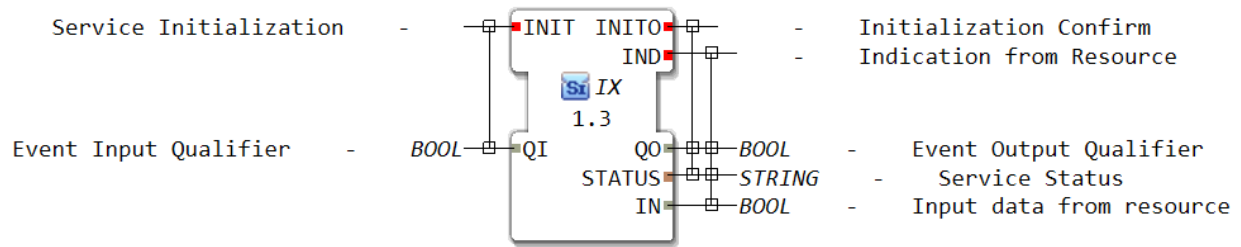
this maps to *IE*

6.2.15.6.4 IE - Input Event (0 Bit)



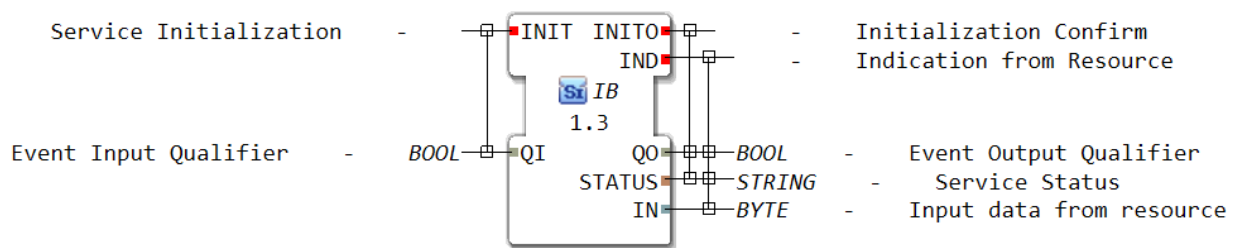
6.2.15.6.5 IX - Input BOOL (1 Bit)

Input service interface function block for boolean input data

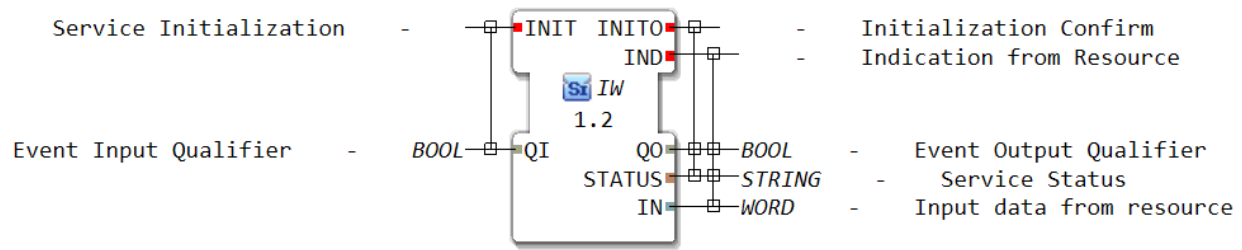


- Input Event
 - INIT Service Initialization
- Input Data
 - QI Event Input Qualifier
- Output Event
 - INITO Initialization Confirm
 - IND Indication from Resource
- Output Data
 - QO Event Output Qualifier
 - STATUS Service Status
 - IN Input data from resource

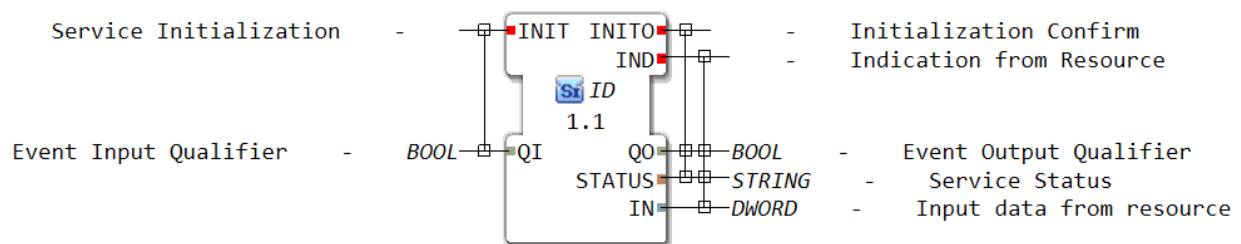
6.2.15.6.6 IB - Input BYTE (8 Bit)



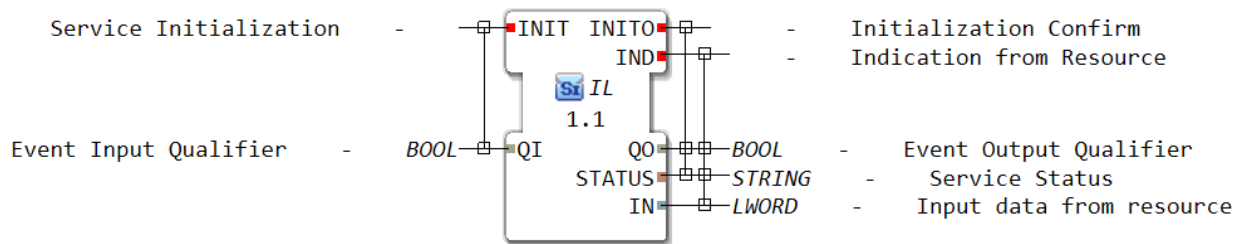
6.2.15.6.7 IW - Input WORD (16 Bit)



6.2.15.6.8 ID - Input DWORD (32 Bit)

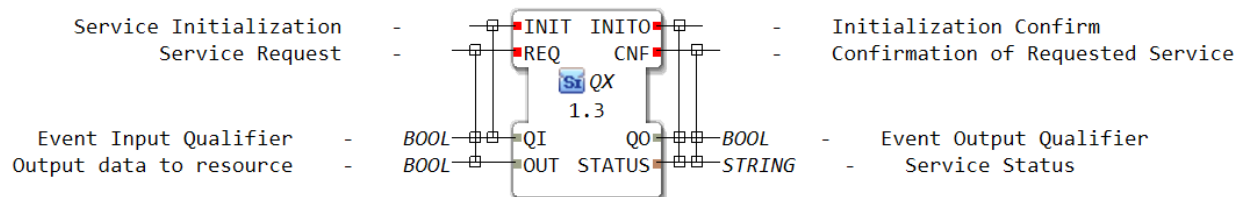


6.2.15.6.9 IL - Input LWORD (64 bit)



6.2.15.6.10 QX - Output BOOL (1 Bit)

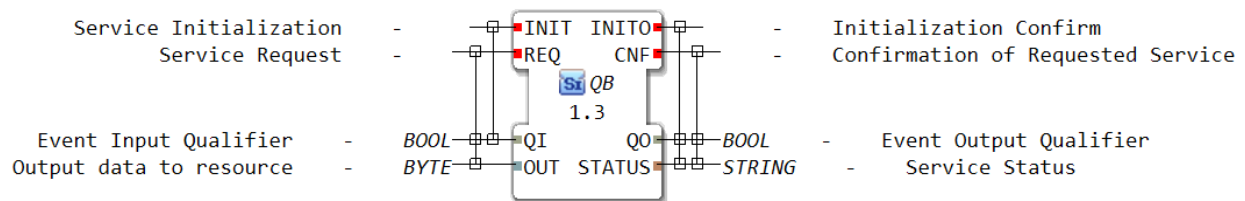
Service Interface Function Block Type



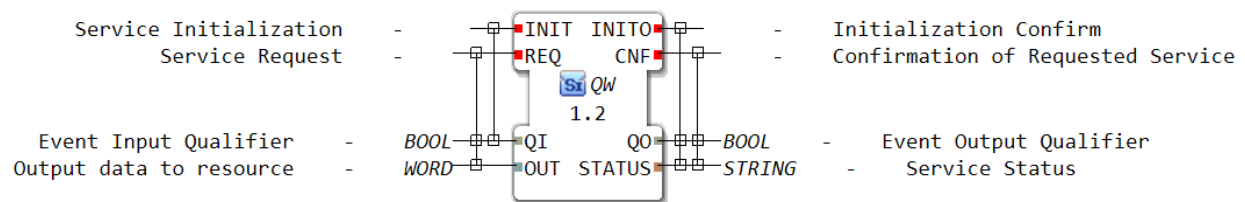
- Input Event
 - INIT Service Initialization
 - REQ Service Request

- Input Data
 - QI Event Input Qualifier
 - OUT Output data to resource
- Output Event
 - INIT0 Initialization Confirm
 - CNF Confirmation of Requested Service
- Output Data
 - QO Event Output Qualifier
 - STATUS Service Status

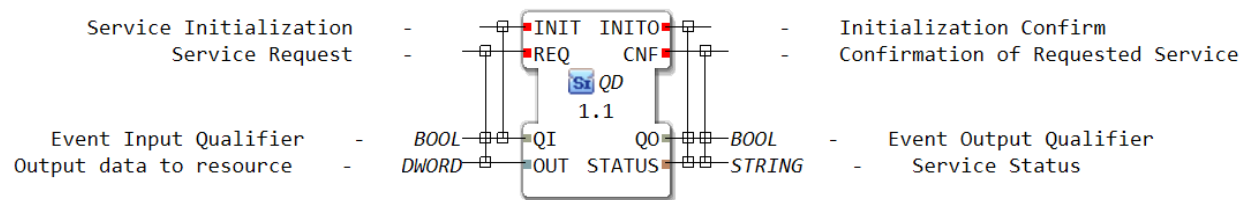
6.2.15.6.11 QB - Output Byte (8 Bit)



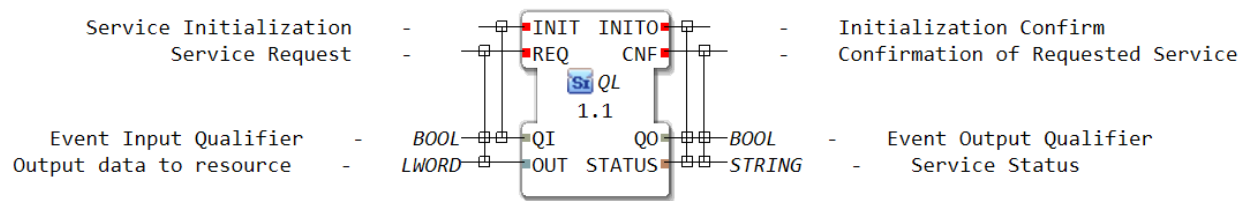
6.2.15.6.12 QW - Output WORD (16 bit)



6.2.15.6.13 QD - Output DWORD (32 bit)



6.2.15.6.14 QL - Output LWORD (64 Bit)



6.2.15.7 isobus

6.2.15.7.1 c

6.2.15.7.1.1 IsoCommonDef

E_BUSY

(-4) ErrorCause: Repeated calling of a function

E_CANMSG_MISSED

(-22) ErrorCause: CAN message was not received

E_CHECKSUM

(-11) ErrorCause: Checksum error

E_COM

(-9) ErrorCause: Communication error with other application hardware

E_CONFIG

(-10) ErrorCause: Configuration error (function call not possible)

E_DATA_LINK

(-20) ErrorCause: Transport protocol used ...

E_DEFAULT

(-3) ErrorCause: Called function works with default values

E_ERROR_INDI

(-16) ErrorCause: ISO error indicator

E_NACK

(-25) ErrorCause: NACK or PDNACK received of bus

E_NOACT

(-8) ErrorCause: Called functionality has not been executed

E_NOT_AVAILABLE

(-17) ErrorCause: Parameter, value ... not available

E_NO_ERR

(0) ErrorCause: No error

E_NO_INSTANCE

(-21) ErrorCause: Instance not found

E_OUT_OF_MEMORY

(-24) ErrorCause: VT, TC has run out of memory

E_OVERFLOW

(-6) ErrorCause: Buffer overflow

E_PGN_UNKNOWN

(-18) ErrorCause: unknown PGN

E_RANGE

(-5) ErrorCause: One ore more input parameter wrong

E_RD_WR

(-7) ErrorCause: Write- or read-error

E_RESPTOCMDERROR

(-26) ErrorCause: Response to Command with ErrorCode != 0

E_UNKNOWN_ERR

(-1) ErrorCause: Unknown error

E_USER_ERROR

(-19) ErrorCause: IsoUser unknown, not announced ...

E_VALUE_WRONG

(-23) ErrorCause: Value or parameter has the wrong value

E_WARN

(-2) ErrorCause: System warnings

6.2.15.7.1.2 IsoVtcApi

void **IsoVtcCloseInstance**(iso_u8 u8Instance)

brief Removes the VT client instance

Triggers the shutdown of the VT client instance. After the connection to the VT is closed the instance will be removed.

par Hint: If the instance has active pool on the VT, the #delete_object_pool command is send in the next cycle to the VT. After the VT response the instance is removed. If it is necessary to kill the instance immediately, the following command can be used: n IsoVtcDataSet(u8Instance, VT_LOGOUT, 1u);

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

iso_s16 **IsoVtcCmd_ActiveMask**(iso_u8 u8Instance, iso_u16 u16WorkSetId, iso_u16 u16NewMaskId)

brief Command change active mask(Part 6 - F.34)

This command is used to change the active mask of a working set to either a data or an alarm mask object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16WorkSetId, iso_u16}

WorkingSet Object ID

param[in] wp{u16NewMaskId, iso_u16}

Object ID of new active (data/alarm)-mask

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_Attribute**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u8 u8IdAttribute, iso_u32 u32ValueAttribute)

brief Command change attribute(Part 6 - F.38)

This command is used to change any attribute with an assigned AID. This message cannot be used to change strings.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wp{u8IdAttribute, iso_u8}

Attribute ID (AID) - see description of wished object

param[in] wp{u32ValueAttribute, iso_u32}

New value for attribute

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_BackgroundColour**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u8 u8Colour)

brief Command change background colour (Part 6 - F.20)

This command is used to change the background colour of an object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wp{u8Colour, iso_u16}

New Background colour see Part 6 - A.3 VT standard colour palette

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_ChangeObjectLabel**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u16 u16ObjIdStringVar, iso_u8 u8FontType, iso_u16 u16ObjIdGrafic)

brief Command Change Object Label(Part 6 - F.50)

This command is used by an ECU to change a label of an object. See also B.21 Object Label Reference List object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID of object to associate label with

param[in] wp{u16ObjIdStringVar, iso_u16}

Object ID of a String Variable object that contains the label string (32 characters maximum) or FFFF if no text is supplied

param[in] wp{u8FontType, iso_u8}

Font type (ignored if String Variable object reference is NULL or the string contains a WideString)

param[in] wp{u16ObjIdGrafic, iso_u16}

Object ID of an object to be used as a graphic representation of the object label or FFFF16 if no designator supplied.

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_ChangePolygonPoint**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u8 u8PointIndex, iso_u16 u16NewXValue, iso_u16 u16NewYValue)

brief Command Change Polygon Point(Part 6 - F.52)

This command is used by a Working Set to modify a point in an Output Polygon object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID of the Output Polygon object to change

param[in] wp{u8PointIndex, iso_u8}

Point index of the point to replace.

param[in] wp{u16NewXValue, iso_u16}

New X value of a point relative to the top left corner of the polygon

param[in] wp{u16NewYValue, iso_u16}

New Y value of a point relative to the top left corner of the polygon

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_ChangePolygonScale**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u16 u16NewWidth, iso_u16 u16NewHeight)

brief Command Change Polygon Scale(Part 6 - F.54)

This command is used by a Working Set to change the scale of a complete Output Polygon object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID of the Output Polygon object to scale

param[in] wp{u16NewWidth, iso_u16}

New width attribute

param[in] wp{u16NewHeight, iso_u16}

New height attribute

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_ChildLocation**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u16 u16ObjIdParent, iso_u8 u8Xchange, iso_u8 u8Ychange)

brief Command change child location (Part 6 - F.14)

The Change Child Location command is used to change the relative position of an object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wp{u16ObjIdParent, iso_u16}

Parent object ID

param[in] wp{u8Xchange, iso_u8}

Relative change in x position (interpreted as signed!)

param[in] wp{u8Ychange, iso_u8}

Relative change in y position (interpreted as signed!)

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_ChildPosition**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u16 u16ObjIdParent, iso_s16 s16Xposition, iso_s16 s16Yposition)

brief Command change child position (Part 6 - F.16)

The Change Child Location command is used to change the absolute position of an object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wp{u16ObjIdParent, iso_u16}

Parent object ID

param[in] wp{s16Xposition, iso_s16}

New X position relative to the top left corner of parent object

param[in] wp{s16Yposition, iso_s16}

New Y position relative to the top left corner of parent object.

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_CtrlAudioSignal**(iso_u8 u8Instance, iso_u8 u8NumOfRepit, iso_u16 u16Frequency, iso_u16 u16OnTimeMs, iso_u16 u16OffTimeMs)

brief Command Control audio device (Part 6 - F.10).

This command can be used to control the audio on the VT.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u8NumOfRepit, iso_u8}

Number of repetitions Range: 1 - 255

param[in] wp{u16Frequency, iso_u16}

Frequency

param[in] wp{u16OnTimeMs, iso_u16}

On time duration in ms

param[in] wp{u16OffTimeMs, iso_u16}

Off time duration in ms

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_DeleteVersion**(iso_u8 u8Instance, const iso_u8 au8VersionString[])

brief Command for deleting object pool of VT ROM

This function releases a command to the VT. The pool selected with the version string is deleted of Flash.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wpp{au8VersionString, const #iso_u8[] }

Pointer to version String which shall be deleted n Length of Bytearray: 7 Characters

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 IsoVtcCmd_ESC(iso_u8 u8Instance)

brief Esc Command

This command is used to abort operator input (Part 6 - F.8).

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 IsoVtcCmd_EndPoint(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u16 u16Width, iso_u16 u16Height, iso_u8 u8LineDirection)

brief Command change end point (Part 6 - F.26)

This command is used to change the end point of a line object by changing the width, height and/or line direction attributes.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wp{u16Width, iso_u16}

X value of end point

param[in] wp{u16Height, iso_u16}

Y value of end point

param[in] wp{u8LineDirection, iso_u8}

Line direction see Part 6 - B.27

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 IsoVtcCmd_ExecuteExtendedMacro(iso_u8 u8Instance, iso_u16 u16ObjIdMacro)

brief Command Execute Extended Macro(Part 6 - F.62)

This command is used to execute a Macro with 16-bit Object ID. This message is available in VT version 5 and later.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjIdMacro, iso_u16}

Object ID of Macro object

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_ExecuteMacro**(iso_u8 u8Instance, iso_u16 u16ObjId)

brief Command Execute Macro(Part 6 - F.48)

This message is available in VT Version 4 and later. This command is used to execute a Macro.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID of Macro object (max. value 255)

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_FillAttributes** **IsoVtcCmd_FillAttributes**(iso_u8 u8Instance ,
iso_u16 u16ObjId , iso_u8 u8FillType , iso_u8 u8FillColour , iso_u16 u16FillPatternId)

brief Command change fill attributes (Part 6 - F.32)

This command is used to change the fill attributes in a fill attributes object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wp{u8FillType, iso_u8}

New fill type (see Part 6 - B.14.4)

param[in] wp{u8FillColour, iso_u8}

New fill colour (see Part 6 - B.14.4)

param[in] wp{u16FillPatternId, iso_u16}

New fill pattern (see Part 6 - B.14.4)

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 IsoVtcCmd_FontAttributes IsoVtcCmd_FontAttributes(iso_u8 u8Instance , iso_u16 u16ObjId , iso_u8 u8FontColour , iso_u8 u8FontSize , iso_u8 u8FontType , iso_u8 u8FontStyle)

brief Command change font attributes (Part 6 - F.28)

This command is used to change the font attributes in a font attributes object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wp{u8FontColour, iso_u8}

New font colour (see Part 6 - A.3)

param[in] wp{u8FontSize, iso_u8}

New font size (see Part 6 - B.14)

param[in] wp{u8FontType, iso_u8}

New font type (see Part 6 - B.14)

param[in] wp{u8FontStyle, iso_u8}

New font style (see Part 6 - B.14)

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 IsoVtcCmd_GetAttribute(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u8 u8AID)

brief Command GetAttribute(Part 6 - F.58)

This message is available in VT Version 4 and later. This command is used by a Working Set to query the VT for the current state of objects within the VT.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wp{u8AID, iso_u8}

Attribute ID of the object

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 IsoVtcCmd_GetSupportedObjects(iso_u8 u8Instance)

brief This technical message requests for supported objects (D.14)

This command is used by the WS to get the list of all object types supported by the VT. This message applies to version 4 and later VTs.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 IsoVtcCmd_GetSupportedWidechar(iso_u8 u8Instance, iso_u8 u8CodePlane, iso_u16 u16RangeFirstWidechar, iso_u16 u16RangeLastWidechar)

brief This technical message requests the supported widechars for a certain code plane (D.10)

The Get Supported Widechars message is used by the Working Set to determine the WideChars supported by the VT. This message only applies to version 4 and later VTs.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u8CodePlane, iso_u8}

Codeplane - Range: 0 ... 16

param[in] wp{u16RangeFirstWidechar, iso_u16}

First widechar of inquired code plane

param[in] wp{u16RangeLastWidechar, iso_u16}

Last widechar of inquired code plane

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 IsoVtcCmd_GetWindowMaskData(iso_u8 u8Instance)

brief This technical message requests background colours (D.12)

The Working Set sends the Get Window Mask Data message to request the background colour of UserLayout Data Mask and the background colour of the Key Cells on a User-Layout Soft Key Mask. This message applies to version 4 and later VTs.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 IsoVtcCmd_GraphicsContext(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u8 u8SubCommandID, const iso_u8 u8ParaList[], iso_u16 u16LenBytes)

brief Command Graphics Context (Part 6 - F.56)

This command is used to manipulate a graphics Context object (only on version 4 or later VTs).

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID of a Graphics Context object

param[in] wp{u8SubCommandID, iso_u8}

Sub-Command ID

param[in] wpp{u8ParaList[], const #iso_u8[]}

Parameters based on sub-command ID byte

param[in] wp{u16LenBytes, iso_u16}

Length in bytes of parameter list

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

note par - Function uses string buffer (Parameterlist has to be smaller than ISO_VTC_CMD_STR_MAX_LENGTH)

iso_s16 **IsoVtcCmd_IdentifyVT**(iso_u8 u8Instance)

brief Command Identify VT (Part 6 - D.18)

The Identify VT message may be sent by either Working Sets or VTs. This message is available in VT Version 4 and later.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_LineAttributes** **IsoVtcCmd_LineAttributes**(iso_u8 u8Instance ,
iso_u16 u16ObjId , iso_u8 u8LineColour , iso_u8 u8LineWidth , iso_u16 u16LineArt)

brief Command change line attributes (Part 6 - F.30)

This command is used to change the line attributes in a line attributes object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wp{u8LineColour, iso_u8}

New line colour (see Part 6 - A.3)

param[in] wp{u8LineWidth, iso_u8}

New line width (see Part 6 - B.14.3)

param[in] wp{u16LineArt, iso_u16}

New line art (see Part 6 - B.14.3)

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_ListItem**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u8 u8ListIndex, iso_u16 u16NewObjId)

brief Command change list item (Part 6 - F.42)

This command is used to change a list item in a list input object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID of a list input object

param[in] wp{u8ListIndex, iso_u8}

List index (-> see list)

param[in] wp{u16NewObjId, iso_u16}

New object ID or 0xFFFF to set empty

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_LockUnlockMask**(iso_u8 u8Instance, iso_u8 u8LockCmd, iso_u16 u16MaskId, iso_u16 u16LockTimeoutMs)

brief Command Lock/UnlockMask(Part 6 - F.46)

This message is available in VT Version 4 and later. This command is used by a Working Set to disallow or allow screen refreshes at the VT for the visible Data Mask or User-Layout Data Mask owned by the requesting Working Set.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u8LockCmd, iso_u8}

0 = Unlock, 1 = Lock data mask or user-layout data mask

param[in] wp{u16MaskId, iso_u16}

Object ID of data mask or user layout data mask to lock or unlock

param[in] wp{u16LockTimeoutMs, iso_u16}

Lock timeout in ms or zero for no timeout

retval iso_s16

- #E_NO_ERR - OK

- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_NumericValue**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u32 u32NewValue)

brief Command change numeric value (Part 6 - F.22)

This command is used to change the value of an object. It applies only to objects that have a numeric value attribute.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wp{u32NewValue, iso_u32}

New value of ID

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_NumericValueRef**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u32 u32NewValue)

brief Command change numeric value with command buffer reference (Part 6 - F.22)

This command is used to change the value of an object. It applies only to objects that have a numeric value attribute. If the objectID is already in the command FIFO then the value of this FIFO entry is updated.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wp{u32NewValue, iso_u32}

New value of ID

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_ObjEnableDisable**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_bool qAbility)

brief Command enable/disable (input) object - (Part 6 - F.4)

This command is used to enable or disable an input field object and pertains to the accessibility of an input object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID of input object

param[in] wp{qAbility, iso_bool}

0 == disable, 1 == enable

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_ObjHideShow**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_bool qVisible)

brief Command Hide/Show Object (Container) (Part 6 - F.2)

The Hide/Show Object command is used to hide or show a container object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16 }

Object ID of Container

param[in] wp{qVisible, iso_bool}

0 = Hide, 1 = Show

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_ObjSelectInput**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u8 u8Option)

brief Command Select input object (Part 6 - F.6)

This command is used to force the selection of an input field object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID of input object

param[in] wp{u8Option, iso_u8}

normal: 0xFF, for Version 4 also 0 possible

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_Priority**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u8 u8Priority)

brief Command change priority(Part 6 - F.40)

This command is used to change the priority of an alarm mask.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID of alarm mask

param[in] wp{u8Priority, iso_u8}

New priority

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_ScreenCapture**(iso_u8 u8Instance, iso_u8 u8ItemRequested, iso_u8 u8Path)

brief Command Screen capture (Part 6 - D.16)

This command, available in VT version 6 and later, requests a lossless screen (image) capture from the VT This message is available in VT Version 6 and later.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u8ItemRequested, iso_u8}

Item requested (0 = Screen image)

param[in] wp{u8Path, iso_u8}

Path (1 = VT accessible storage/removable media)

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_SelectActiveWorkingSet**(iso_u8 u8Instance, const iso_u8 pau8Name[])

brief Command Select Active Working Set(Part 6 - F.64)

This command is used by the currently active Working Set to request that the VT select a new active Working Set. This message is available in VT version 6 and later.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wpp{pau8Name, const #iso_u8[]}

Iso NAME of the new active working set

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_SelectColourMap**(iso_u8 u8Instance, iso_u16 u16ObjIdColourMap)

brief Command Select Colour Map (VT version 4 and later) (Part 6 - F.60)

The Select Colour Map command is used to select the active Colour Map. This command can take a long time to execute. The command applies to any presentation from the originating Working Set, which includes objects that may be shown on other Working Set screens (e.g. Auxiliary Control objects as may be presented on VT proprietary and other Working Set masks using the Auxiliary Control Designator Type 2 Object Pointer).

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjIdColourMap, iso_u16}

Object ID of Colour map object or FFFF to restore the default colour table (see A.3)

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_SetAudioVolume**(iso_u8 u8Instance, iso_u8 u8Volume)

brief Command SetAudioVolume

This command can be used to control the audio on the VT (volume) (Part 6 - F.12)

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u8Volume, iso_u8}

Volume 0 - 100 (%)

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - Buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NOT_AVAILABLE - VT has got no adjustable volume output
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_Size**(iso_u8 u8Instance, iso_u16 u16ObjId, iso_u16 u16Width, iso_u16 u16Height)

brief Command change size (Part 6 - F.18)

The Change Size command is used to change the size of an object.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wp{u16Width, iso_u16}

New width

param[in] wp{u16Height, iso_u16}

New height

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_SoftKeyMask**(iso_u8 u8Instance, iso_u8 u8MaskType, iso_u16 u16DataMaskId, iso_u16 u16SoftKeyMaskId)

brief Command change soft key mask(Part 6 - F.36)

This command is used to change the soft key mask associated with a data or alarm mask object.

param[in] wp{u8Instance, iso_u8}
Identifier of the client instance

param[in] wp{u8MaskType, iso_u8}
1 == Data, 2 == Alarm

param[in] wp{u16DataMaskId, iso_u16}
Object ID of active Mask

param[in] wp{u16SoftKeyMaskId, iso_u16}
Object ID of (new) soft key mask

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_StoreVersion**(iso_u8 u8Instance, const iso_u8 au8VersionString[])

brief Store object pool with version label from VT RAM to VT ROM

This function stores the actual pool in VT RAM into the Flash (ROM) of the VT.

param[in] wp{u8Instance, iso_u8}
Identifier of the client instance

param[in] wpp{au8VersionString, const #iso_u8[] }
Pointer to version String which shall be deleted n Length of Bytearray: 7 Characters

retval iso_s16

- #E_NO_ERR - OK
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_String**(iso_u8 u8Instance, iso_u16 u16ObjId, const iso_u8 pau8String[])

brief Command Change String Value (Part 6 - F.24)

This command sends a string to the VT. The length of the string including ,0‘ may not be longer than the define ISO_VTC_CMD_STR_MAX_LENGTH. Take care that ,0‘ is trailed.

param[in] wp{u8Instance, iso_u8}
Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wpp{pau8String, const #iso_u8[] }

Pointer to string

retval iso_s16

- #E_NO_ERR - transport started
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcCmd_StringRef**(iso_u8 u8Instance, iso_u16 u16ObjId, const iso_u8 pau8String[])

brief Command Change String Value with command buffer reference (Part 6 - F.24)

This command sends a string to the VT. If the ObjectID is already in the command FIFO then the value of this FIFO entry is updated. The length of the string including ,0' may not be longer than the define ISO_VTC_CMD_STR_MAX_LENGTH. Take care that ,0' is trailed.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{u16ObjId, iso_u16}

Object ID

param[in] wpp{pau8String, const #iso_u8[] }

Pointer to string

retval iso_s16

- #E_NO_ERR - transport started
- #E_OVERFLOW - buffer overflow
- #E_NOACT - Command not possible in current state
- #E_NO_INSTANCE - No VT client available

iso_s16 **IsoVtcContinue**(iso_u8 u8Instance)

brief Continue VT Login state machine

There are some cases where VT Login stops for application interaction. This function continues the VT state machine.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

retval iso_s16

- E_NO_ERR
- E_ERROR_INDI - actual state don't supported of this function
- E_NO_INSTANCE

par

- **Supported states:**
 - waiting_before_store

iso_u8 **IsoVtcCreateInstance**(iso_s16 s16CfHandleMaster, const ISO_USER_PARAM_T userParam, ISOVT_CbStatus_t pfVTStatusCb, ISOVT_CbMsg_t pfVtMsgRecCb, ISOVT_CbConnectionEv_t pfVtConnEvCb, ISOVT_CbAuxPrefAssignment_t pfAuxPrefAssignCb, const ISO_CF_NAME_T *pau8NamePreferredVT, iso_u8 u8BootTimePreferredVT)

brief Creates a VT client instance for the given working set master

param[in] wp{s16CfHandleMaster, iso_s16}

CF handle of a working set master

param[in] wpp{userParam, const #ISO_USER_PARAM_T}

User parameter value for the callback function calls. n Use #ISO_USER_PARAM_DEFAULT if not used/needed.

param[in] wp{pfVTStatusCb, ISOVT_CbStatus_t}

Pointer to callback function which is called in case of opening a new data, alarm or window mask, or of

- Parameter rqInfo is set to ISO_TRUE, if the working set is active.
- Parameter rwPage contains the object ID of the open mask, or NULL object ID if rqInfo is ISO_FALSE.

param[in] wp{pfVtMsgRecCb, ISOVT_CbMsg_t}

Pointer to callback function which is called in case of activation messages and command responses to the user application. See #ISOVT_MSG_STA_T for more information.

param[in] wp{pfVtConnEvCb, ISOVT_CbConnectionEv_t}

Pointer to the connection event callback function or NULL pointer if not used.

param[in] wp{pfAuxPrefAssignCb, ISOVT_CbAuxPrefAssignment_t}

Pointer to the preferred assignment callback function (Can set to the NULL pointer if the object pool does not contain auxiliary type 2 functions).

param[in] wpp{pau8NamePreferredVT, const #ISO_CF_NAME_T*}

Pointer to NAME of preferred VT

param[in] wp{u8BootTimePreferredVT, iso_u8}

Boot time of preferred VT in seconds n (Range 0 - 254; if 0xFF for not available ISO_WAITVT_SEC will be used)

retval iso_u8

- VT client instance identification number
- #ISO_INSTANCE_INVALID if there is no instance free

par Hints:

- Preferred VT: During Login the driver waits a boot time for this VT. This can be useful in order to choice the last plugged VT. The actual VT must be stored in NVM and during login initialised with this function.
- If the VT client has has auxiliary type 2 functions and connects not to the primary VT, a second Aux instance is automatically created later to connect to the primary VT. This instance has also an other instance identification number.
- To create another instance with a different VT for a CF, IsoVtcCreateInstance() can be called again with a different VT name.

iso_s16 **IsoVtcDataSet**(iso_u8 u8Instance, ISOVT_STATUS_e eVTInfo, iso_u16 wDatNew)

brief Setting information of a VT client

This function enables the change of some Working Set settings or the control of the state machine. For settings it should be called after `IsoVtcCreateInstance()`. Furthermore it can be completely resetted a VT client with `VT_LOGOUT`.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{eVTInfo, ISOVT_STATUS_e}

VT_BOOTTIME - Boot time of preferred VT in seconds n VT_STATEOFANNOUNCING - Forced setting of Login state machine (test purposes)

see ISOVT_ANNOUNCE_e n

WS_VERSION_NR - Version number of working set (-> Used object in pool)n VT_LOGOUT - Shut down a VT client instance (set wDatNew to 1u to kill instance; 0u for normal logout) n

For a normal logout you can also use `IsoVtcCloseInstance()`

param[in] wp{wDatNew, iso_u16}

New Value

retval iso_s16

- E_NO_ERR
- E_NO_INSTANCE

iso_u16 IsoVtcGetCMDFifoFree IsoVtcGetCMDFifoFree(iso_u8 u8Instance)

brief Calculate free elements of Command FIFO

This function determines the number of free elements of the command FIFO. It has been implemented for LIB Implementation (initiated of Grimme)

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

retval iso_u16

Number of free elements of command FIFO n Range: 0 - ISO_VTC_CMD_BUFFER_MAX

iso_u8 IsoVtcGetInstanceID(iso_s16 s16CfHandleMaster, ISOVT_INSTANCE_TYPE_Te eInstanceType)

brief Helper function to get the VT client instance identifier to a client CF handle

This function determines the number of free elements of the command FIFO. It has been implemented for LIB Implementation (initiated of Grimme)

param[in] wp{s16CfHandleMaster, iso_s16}

CF handle of a working set master

param[in] wp{eInstanceType, ISOVT_INSTANCE_TYPE_Te}

Instance type which should be returned

retval iso_u8

Identifier of the client instance or #ISO_INSTANCE_INVALID if not found

iso_u16 IsoVtcGetStatusInfo(iso_u8 u8Instance, ISOVT_STATUS_e eVTInfo)

brief Reading information of a VT client instance

This function can be used to read information about a VT client and the connected VT. This information are useful for controlling the pool on VT e. g. with `VT_PHYSICALSOFTKEYS` it is possible depending on number of physical softkeys of the VT to choose between different soft key masks.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{eVTInfo, ISOVT_STATUS_e}

Enum of wished information of VT n Most of them are of the technical data messages

retval iso_u16 Requested value depending of ISOVT_STATUS_e

- #VT_SOURCE_ADDRESS: actual SA of VT or ISO_NULL_ADDRESS
- #CF_SOURCE_ADDRESS: actual SA of WS or ISO_NULL_ADDRESS
- #ID_VISIBLE_DATA_MASK: actual data mask object in VT status message
- #ID_VISIBLE_SOFTKEY_MASK: actual soft key mask object in VT status message
- #VT_BUSY_CODE: see VT status message
- #AUXUNITS_TYPE1_ONBUS: ISO_TRUE -> Auxiliary unit type 1 is on bus
- #VT_ALIVE: VT alive check (3s), 1 = alive
- #VT_DOWNLOAD_FINISHED: announce state > announce_finished? 1
- #VT_POOL_ACTIVE_onVT: 1: Own pool selected on VT
- #VT_STATEOFANNOUNCING: see enum ISOVT_ANNOUNCE_e
- #WS_VERSION_NR: Supported version of own pool see ISO_VTVERSION
- #VT_NUMBOFVERSIONSTRINGS: Number of version strings stored in VT
- #VT_VERSIONNR: Supported version of VT see Part 6 - D.3
- #VT_NAVSOFTKEYS: Navigation softkeys (of Version 4 else 0xFF)
- **#VT_SOFTKEYXDOT ... VT_DATAMASKRESOLUTION: n**
Technical data messages of actual VT see Part 6 Annex D
- #VT_BGCOLOURULDM, VT_BGCOLOURKEYCELL: colours see A.3

note par

- VT_BOOTTIME: If the VT supports Boottime (>=IL4 and Value != 0xFF) the value (in sec) can be read, and then stored in NVM and set with IsoVtcDataSet() next startup.

iso_s16 IsoVtcMultipleNextVT(iso_u8 u8Instance, const ISO_CF_NAME_T *rpabVtName)

brief Calling this function releases the procedure to move to another VT

The application have to select an available VT and releases with this function the VT change.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wpp{rpabVtName, const #ISO_CF_NAME_T*}

Pointer to NAME of „Next VT“ see ISO11783 - 5

retval iso_s16

- E_NO_ERR - OK
- E_NOACT - VT is used by an other mask instance of this client
- E_ERROR_INDI - Changing only possible in case of full log-on
- E_USER_ERROR - User „rpabVtName“ not found or no VT

par

- Using see example in VT application

iso_bool **IsoVtcObjTypeIsSupported**(iso_u8 u8Instance, OBJTYP_e eObjType)

brief Check if object type is supported of VT

Precondition for this function: VT must be version 4 or higher. Then VT sends a sorted list during login with all supported objects. It can be used for controlling the command messages

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{eObjType, OBJTYP_e}

Object type (Byte 3 of every object) n Range: 0 - 36, 40 see ISO11783 - 6 - A.1.1 or OBJTYP_e

retval iso_bool ISO_TRUE if VT supports this object

void **IsoVtcObjTypeParsableSet**(iso_u8 u8Instance, OBJTYP_e eObjType)

brief Manual setting of a supported (parsable) object type

This function is used to add object types of higher versions to a list which is used for parsing decisions

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{eObjType, OBJTYP_e}

Object type (Byte 3 of every object)

iso_s16 **IsoVtcReadWorkingSetLanguageData**(iso_u8 u8Instance, iso_u8 abLanguageCmd[])

brief This function reads the language of the working set

The language command of the VT which displays the data masks is returned. If there is no VT the language command of the TECU is returned.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[out] wp{abLanguageCmd, iso_u8[]}

Pointer to language command data array of this working set n Size of the array must be (at least 8 bytes)

retval iso_s16

- E_NO_ERR - OK
- E_ERROR_INDI - WS not found

par - Attention: Function works only correct after login has passed the LC reception

(e. g. waiting_after_GetVersions) else it provides the TECU LC

iso_s16 **IsoVtcRestartInstance**(iso_u8 u8Instance, iso_bool qReConnect)

brief Restarts a VT client instance

Triggers the shutdown of the VT client instance. After the current pool is removed from the VT, the instance starts again.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{qReConnect, iso_bool}

- #ISO_TRUE: The connection is „closed“ and restarted at the beginning of the state machine
- #ISO_FALSE After the pool is removed the instance restarts with the pool upload procedure

retval iso_s16

- #E_NO_ERR - restart triggered
- #E_NOACT - function called again
- #E_NO_INSTANCE - Instance not found

iso_s16 **IsoVtcSetNewPreferredVT**(iso_u8 u8Instance, const ISO_CF_NAME_T *rpabVtName)

brief Function to set the preferred VT to a new CF name.

The function can only called before the client has selected a VT (safe state). n In this case the given name is set as new preferred VT and the VT boot timeout is reseted.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wpp{rpabVtName, const #ISO_CF_NAME_T*}

Pointer to the NAME of the VT or dummy name (8 * 0xFFu)

retval iso_s16

- E_NO_ERR - On success
- E_RANGE - rpabVtName is the NULL pointer or WS instance is not set
- E_ERROR_INDI - Changing only possible while waiting for the currently set preferred VT
- E_USER_ERROR - CF name „rpabVtName“ is not a valid VT name (wrong name function)

par

- This function should be used to select the wished VT during the #IsoEvConnSelectPreferredVT event callback.

iso_u8 **IsoVtcVersionStringGet**(iso_u8 u8Instance, iso_u8
au8VersionStrings[][ISO_VTC_LEN_VERSIONSTR])

brief Function reads the Version strings received from VT with (Extended) Get Versions

The version strings received of VT can be read. If both Working Set and VT are Version 5 or higher the version strings are (exactly)32 bytes else 7 bytes.

param[in] wp{u8Instance, iso_u8}

Identifier of the client instance

param[in] wp{au8VersionStrings, iso_u8[][ISO_VTC_LEN_VERSIONSTR]}

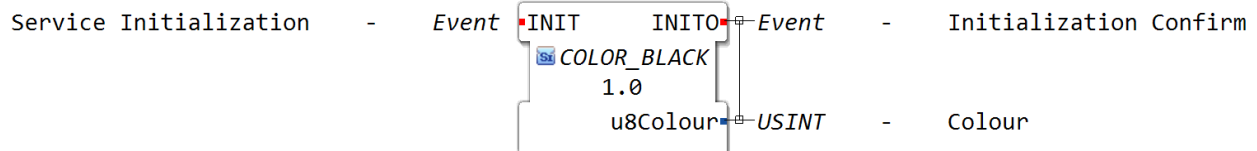
Pointer to 2d-array - driver writes the version strings in this array - first element is number of version strings (see ISO_VTC_MAX_POOLVERSIONS) - second element is string length #ISO_VTC_LEN_VERSIONSTR

retval iso_s16

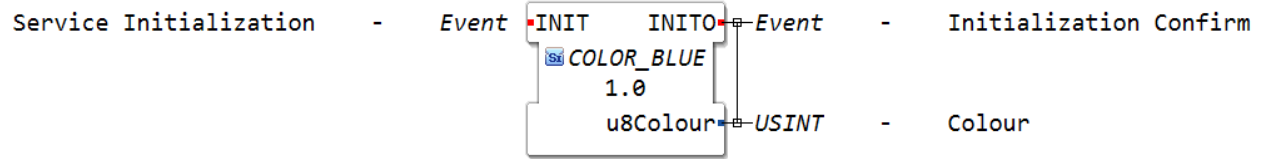
Number of version strings received with (Extended) Get Version

6.2.15.7.2 colours

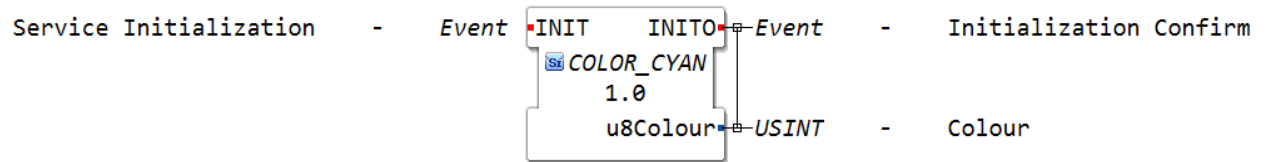
6.2.15.7.2.1 COLOR_BLACK



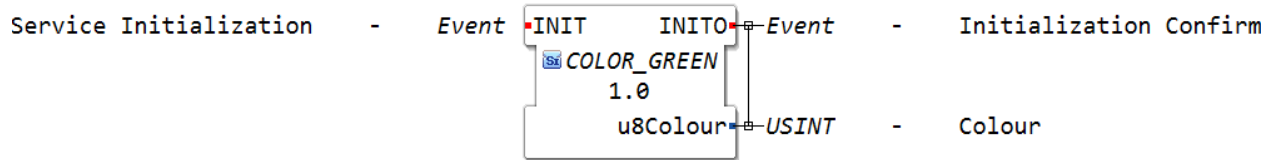
6.2.15.7.2.2 COLOR_BLUE



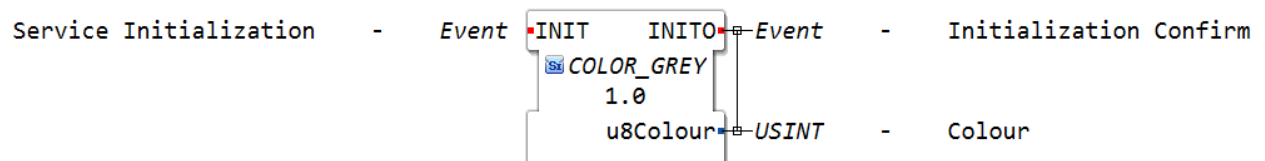
6.2.15.7.2.3 COLOR_CYAN



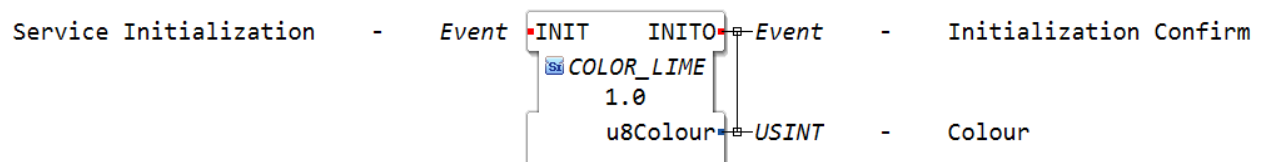
6.2.15.7.2.4 COLOR_GREEN



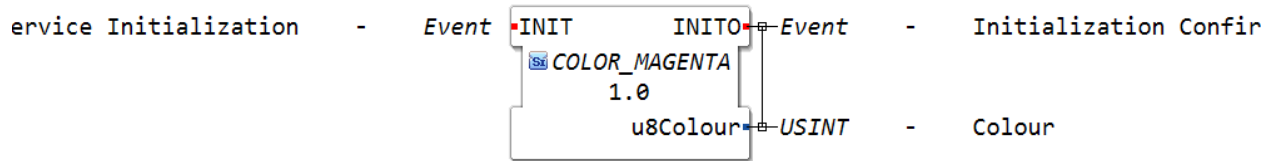
6.2.15.7.2.5 COLOR_GREY



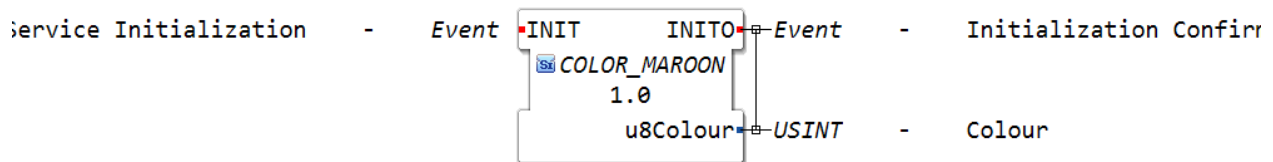
6.2.15.7.2.6 COLOR_LIME



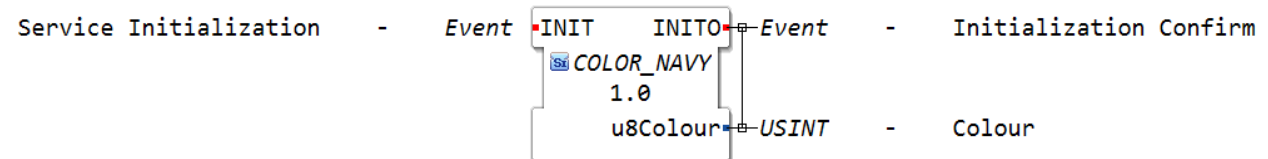
6.2.15.7.2.7 COLOR_MAGENTA



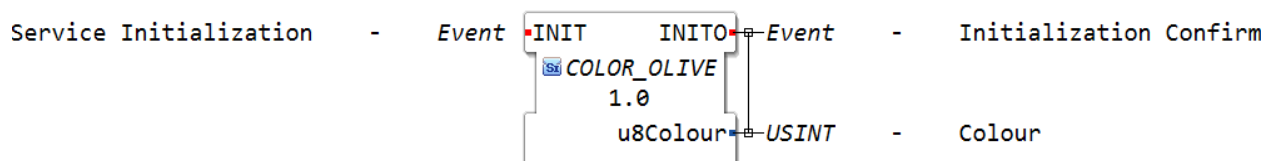
6.2.15.7.2.8 COLOR_MAROON



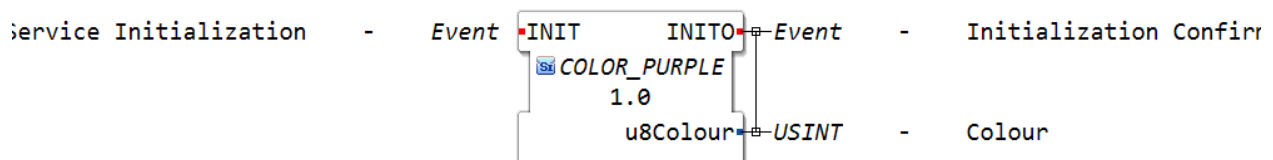
6.2.15.7.2.9 COLOR_NAVY



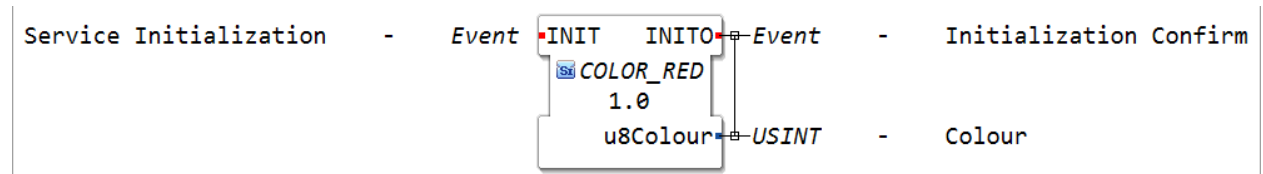
6.2.15.7.2.10 COLOR_OLIVE



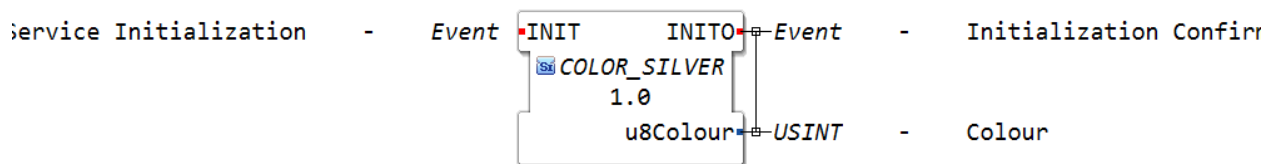
6.2.15.7.2.11 COLOR_PURPLE



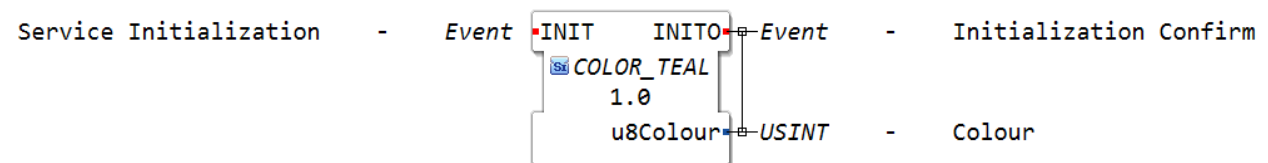
6.2.15.7.2.12 COLOR_RED



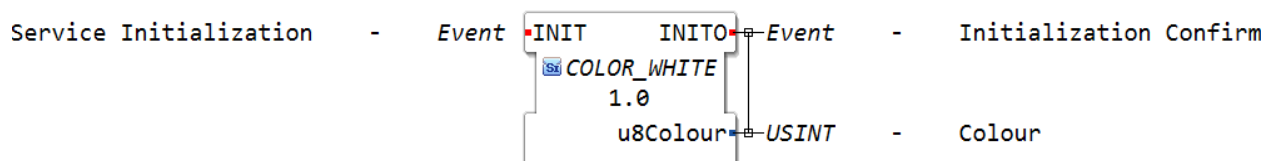
6.2.15.7.2.13 COLOR_SILVER



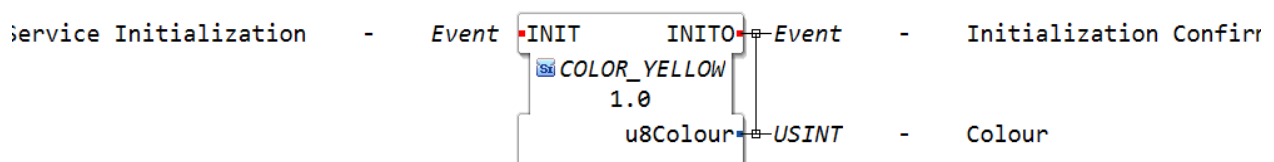
6.2.15.7.2.14 COLOR_TEAL



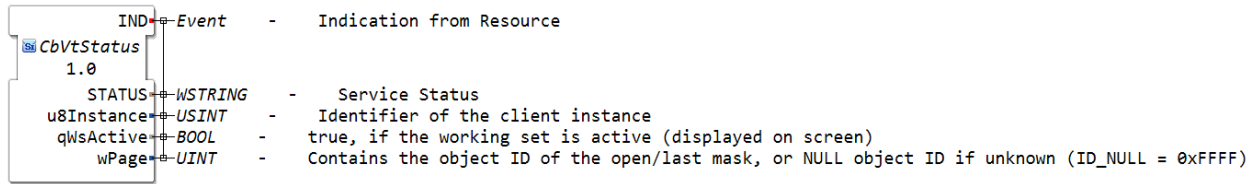
6.2.15.7.2.15 COLOR_WHITE



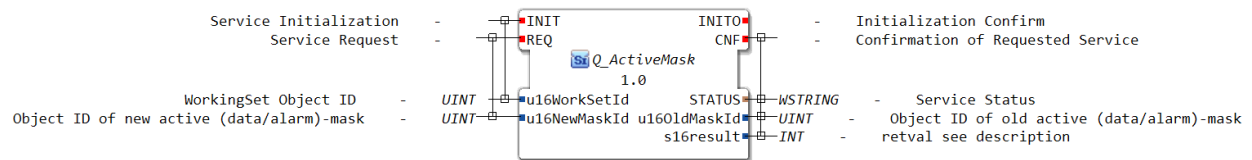
6.2.15.7.2.16 COLOR_YELLOW



6.2.15.7.3 CbVtStatus



6.2.15.7.4 Q_ActiveMask



ISO 11783-6

Command change active mask(Part 6 - F.34)

This command is used to change the active mask of a working set to either a data or an alarm mask object.

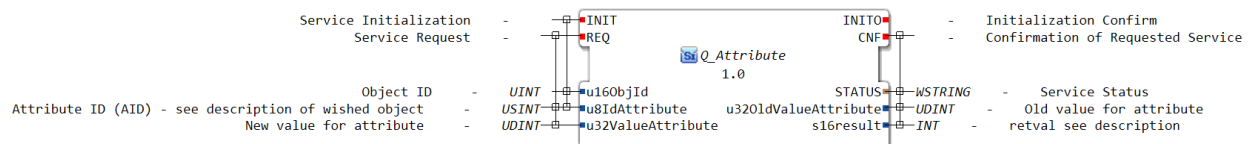
Return values

iso_s16

- *E_NO_ERR* (0) - OK
- *E_OVERFLOW* (-6) - buffer overflow
- *E_NOACT* (-8) - Command not possible in current state
- *E_NO_INSTANCE* (-21) - No VT client available

Related to: *IsoVtcCmd_ActiveMask*

6.2.15.7.5 Q_Attribute



ISO 11783-6

Command change attribute(Part 6 - F.38)

This command is used to change any attribute with an assigned AID. This message cannot be used to change strings.

Return values

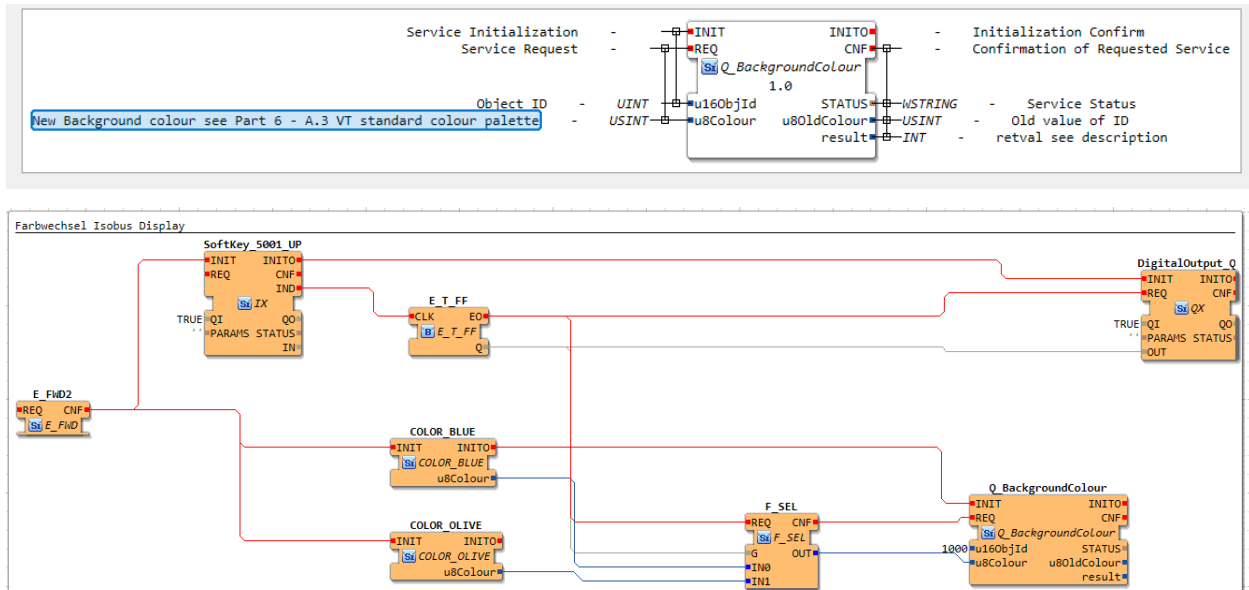
iso_s16

- *E_NO_ERR* (0) - OK
- *E_OVERFLOW* (-6) - buffer overflow

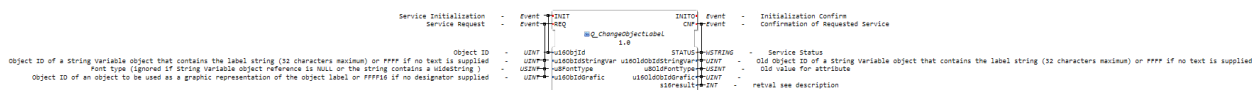
- *E_NOACT* (-8) - Command not possible in current state
- *E_NO_INSTANCE* (-21) - No VT client available

Related to: *IsoVtcCmd_Attribute*

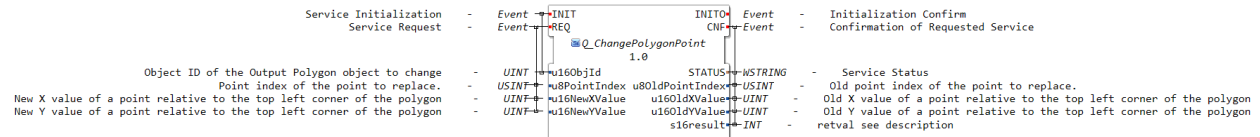
6.2.15.7.6 Q_BackgroundColour (Farbprogrammierung IsoBus Display)



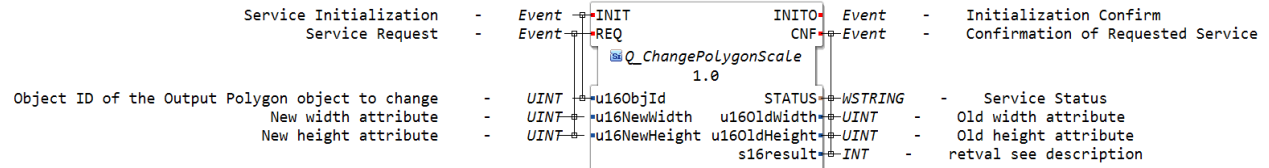
6.2.15.7.7 Q_ChangeObjectLabel



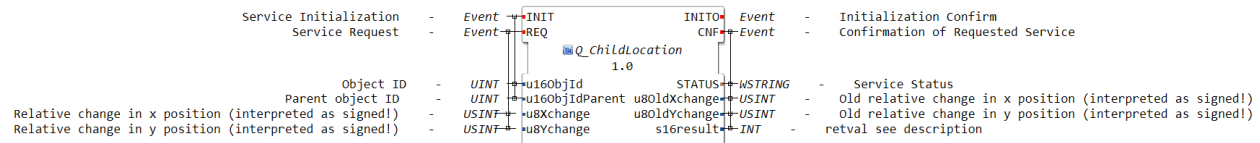
6.2.15.7.8 Q_ChangePolygonPoint



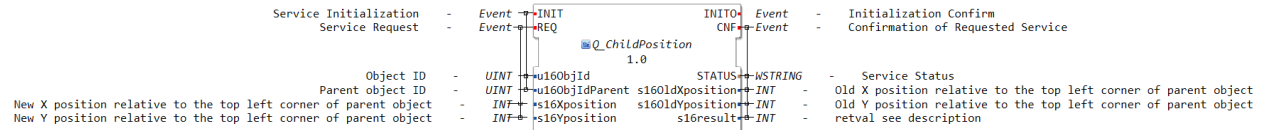
6.2.15.7.9 Q_ChangePolygonScale



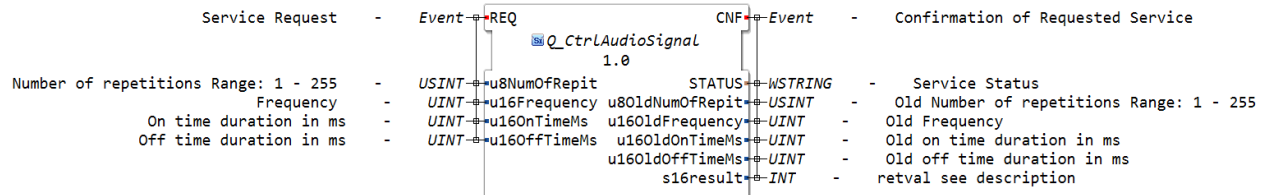
6.2.15.7.10 Q_ChildLocation



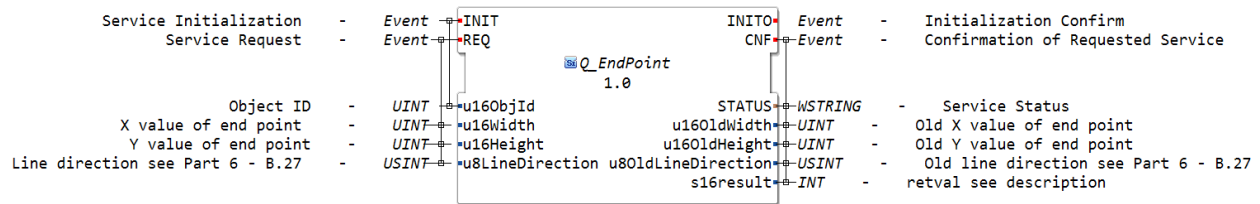
6.2.15.7.11 Q_ChildPosition



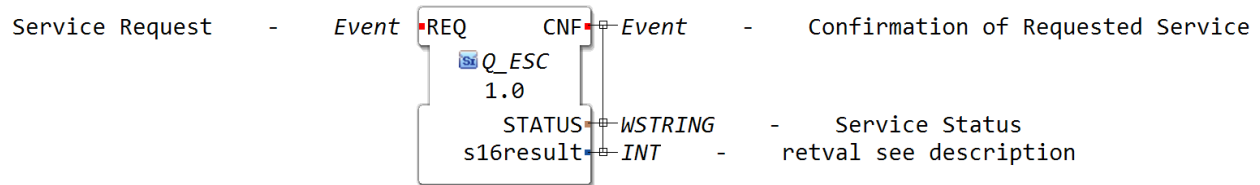
6.2.15.7.12 Q_CtrlAudioSignal



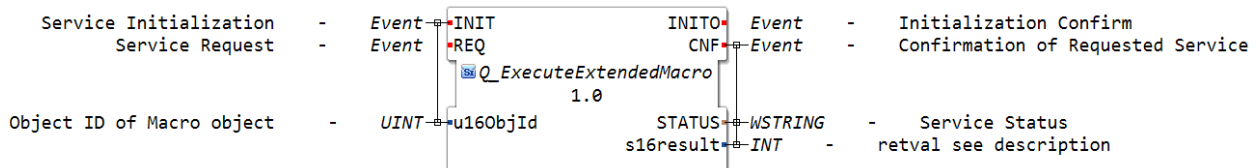
6.2.15.7.13 Q_EndPoint



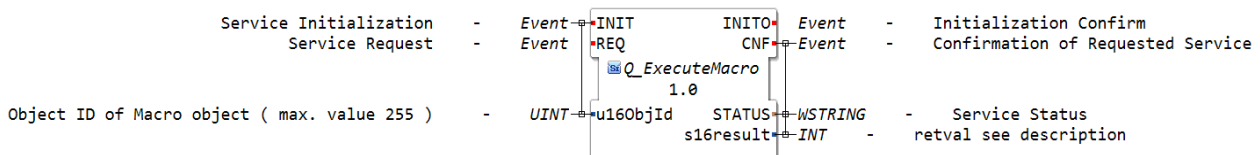
6.2.15.7.14 Q_ESC



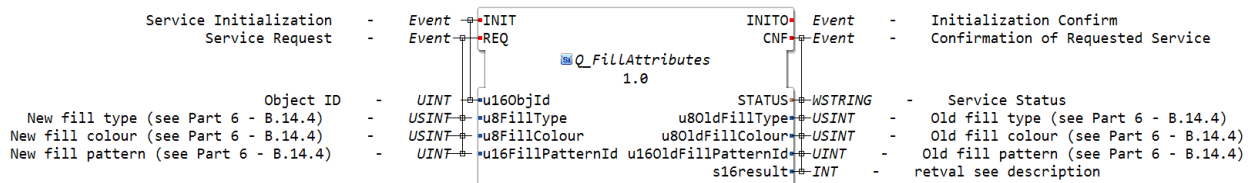
6.2.15.7.15 Q_ExecuteExtendedMacro



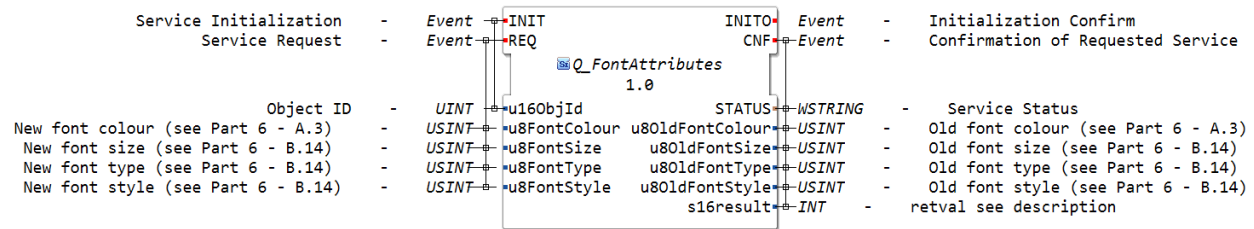
6.2.15.7.16 Q_ExecuteMacro



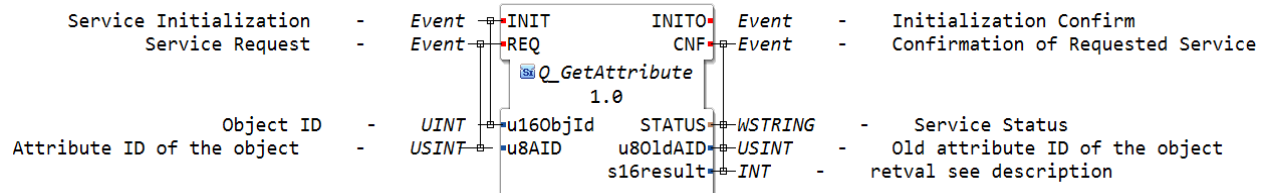
6.2.15.7.17 Q_FillAttributes



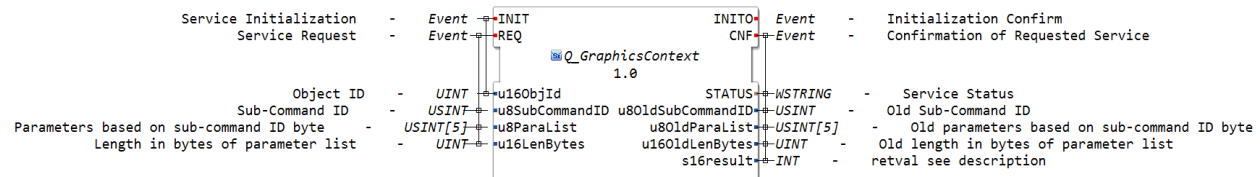
6.2.15.7.18 Q_FontAttributes



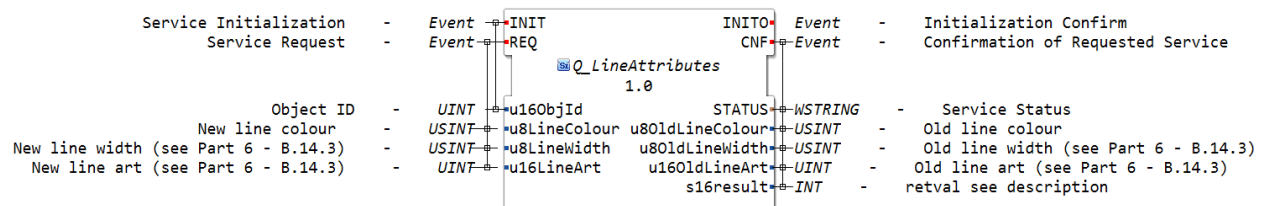
6.2.15.7.19 Q_GetAttribute



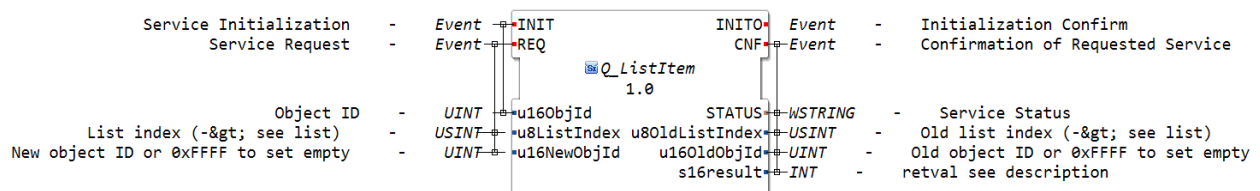
6.2.15.7.20 Q_GraphicsContext



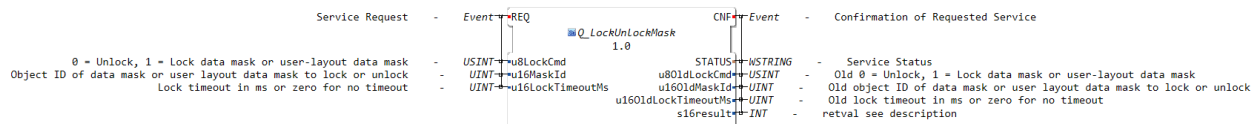
6.2.15.7.21 Q_LineAttributes



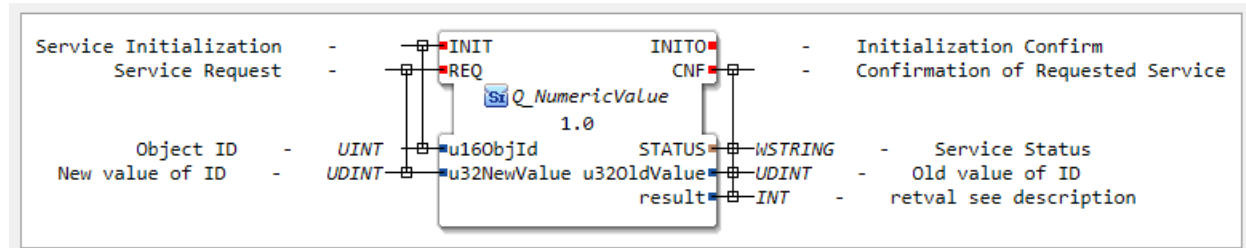
6.2.15.7.22 Q_ListItem



6.2.15.7.23 Q_LockUnlockMask



6.2.15.7.24 Q_NumericValue



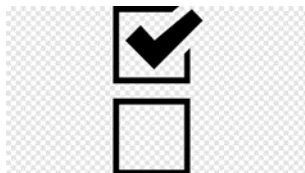
Normativen Referenz:

Dieser Funktionsbaustein bezieht sich auf den Abschnitt ISO 11783-6 F.22 Change Numeric Value command

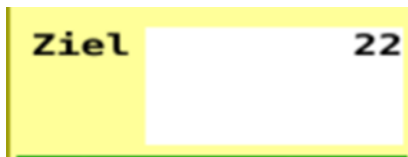
ACHTUNG: dieser Befehl kann so oft aufgerufen werden wie vom Anwendungsprogrammierer gewünscht, jedoch kann ein zu häufiges Aufrufen den ISOBUS stark überlasten, weshalb dieser Baustein mit Bedacht eingesetzt werden sollte.

Aufgabe dieses Bausteins ist es eine der folgenden Objekte die auf einem VT dargestellt werden zu aktualisieren:

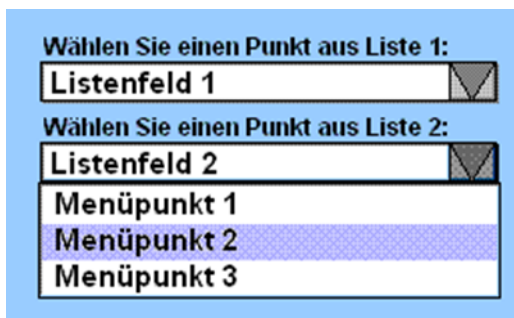
- Boolean input field: 1 byte for TRUE/FALSE



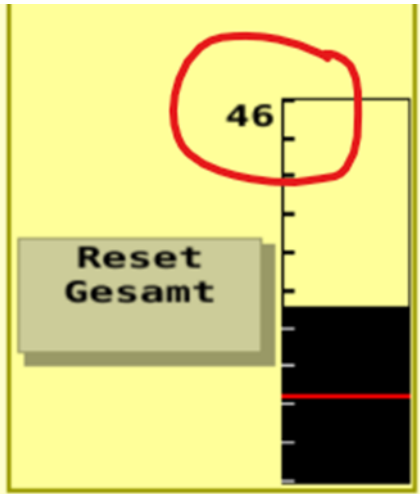
- Number input field: 4 bytes for integer input



- List input field: 1 byte for list index



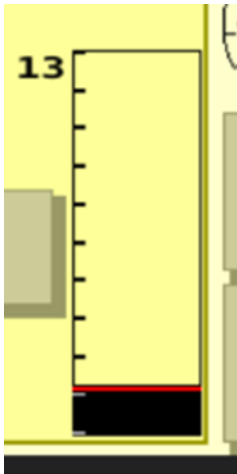
- List output field: 1 byte for list index
 - (wie list Input field, jedoch in VT3 nicht vorhanden)
- Number output field: 4 bytes for integer output



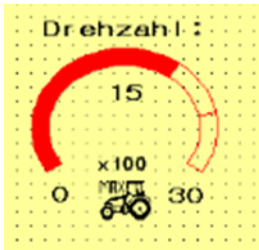
- Meter: 2 bytes for integer value



- Linear bar graph: 2 bytes for integer value



- Arched bar graph: 2 bytes for integer value



- Number variable: 4 bytes for integer value
 - Die Number Variable kann nicht direkt auf dem Display angezeigt werden, kann aber mit allen bisher genannten Objekten verknüpft werden; Vorteil: auch mehrfache Verknüpfungen Möglich.
- Object pointer: 2 bytes for Object ID
 - Sinnvoll um andere Objekte an Stelle des Pointers ein- und auszublenden.

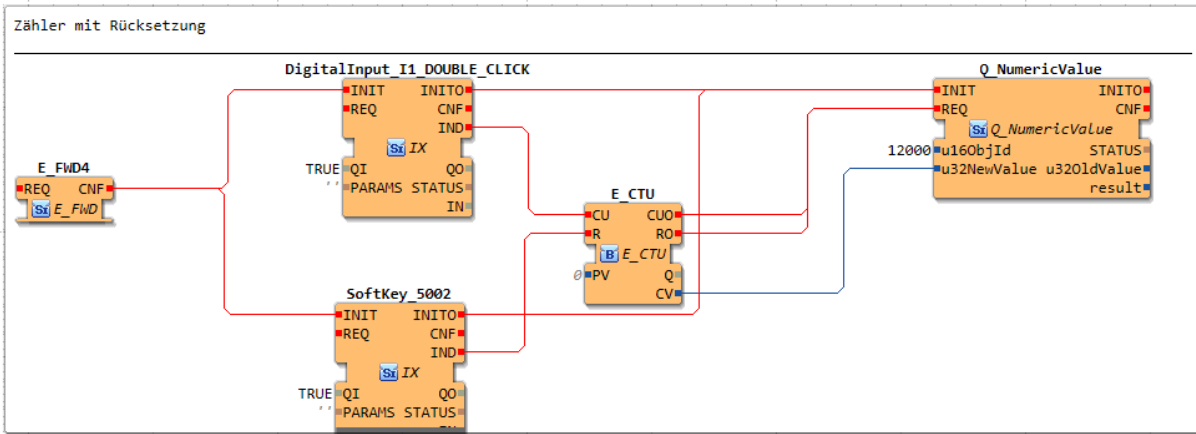


Querverweise:

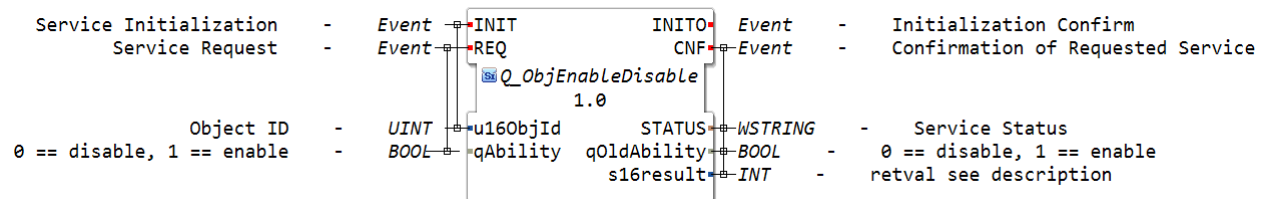
<https://docs.ms-muc-docs.de/projects/isobus-vt-objects-docs/de/latest/ID-07—Input-boolean—ISO-11783-6—B.8.2.html>

BEISPIEL

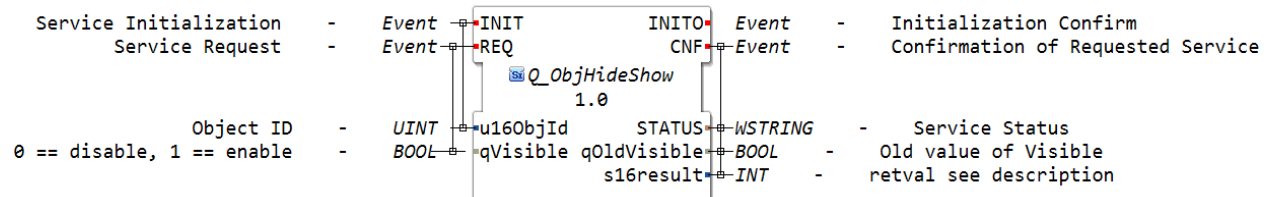
Zähler mit Rücksetzung



6.2.15.7.25 Q_ObjEnableDisable

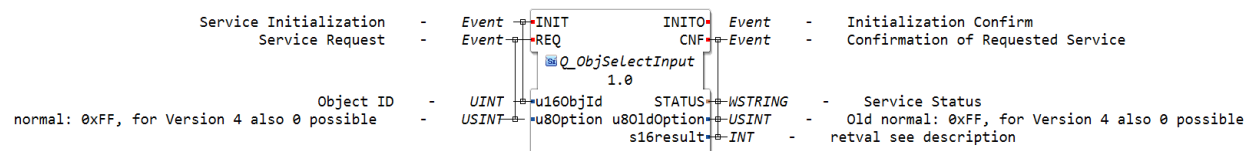


6.2.15.7.26 Q_ObjHideShow

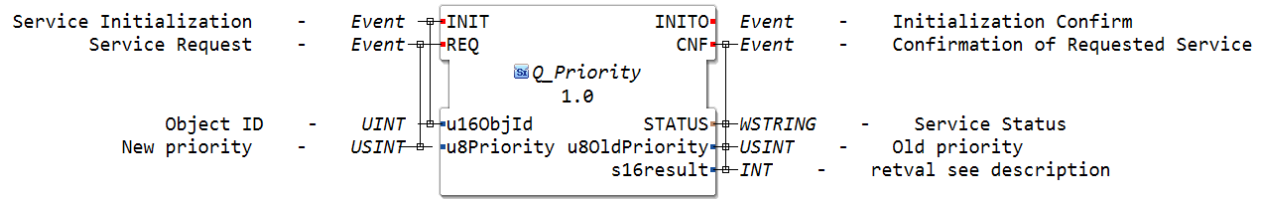


Command Hide/Show Object (Container) (Part 6 - F.2)

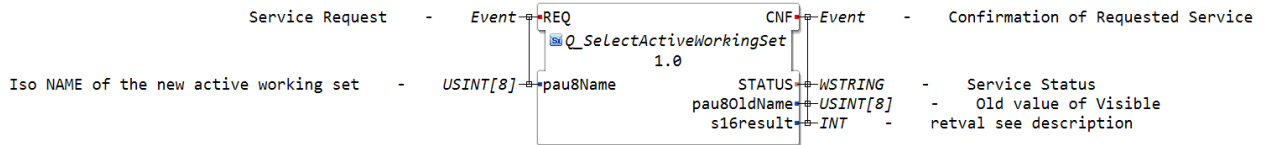
6.2.15.7.27 Q_ObjSelectInput



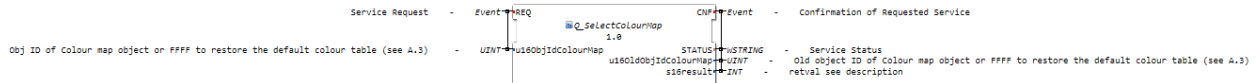
6.2.15.7.28 Q_Priority



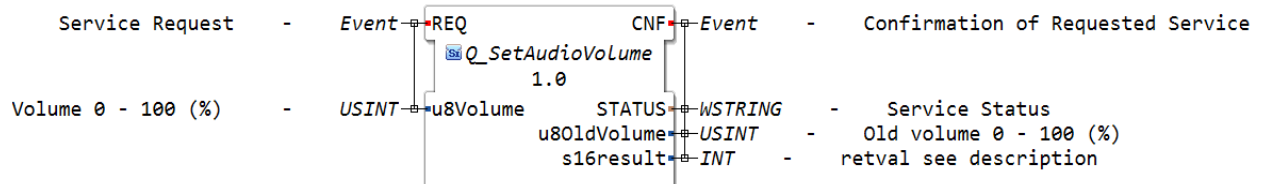
6.2.15.7.29 Q_SelectActiveWorkingSet



6.2.15.7.30 Q_SelectColourMap



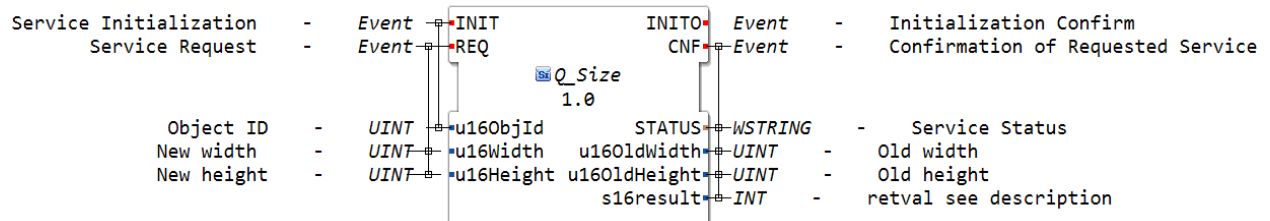
6.2.15.7.31 Q_SetAudioVolume



Command SetAudioVolume.

ISO 11783-6

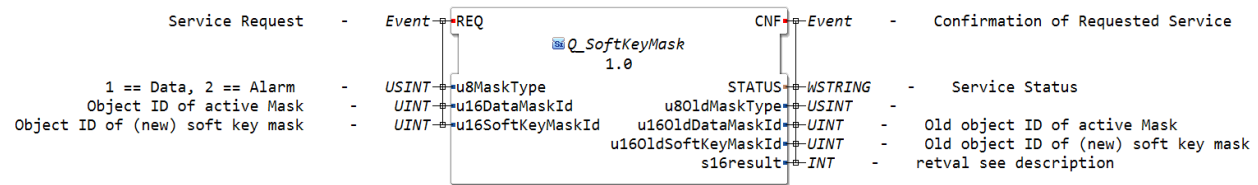
6.2.15.7.32 Q_Size



Command change size (Part 6 - F.18)

ISO 11783-6

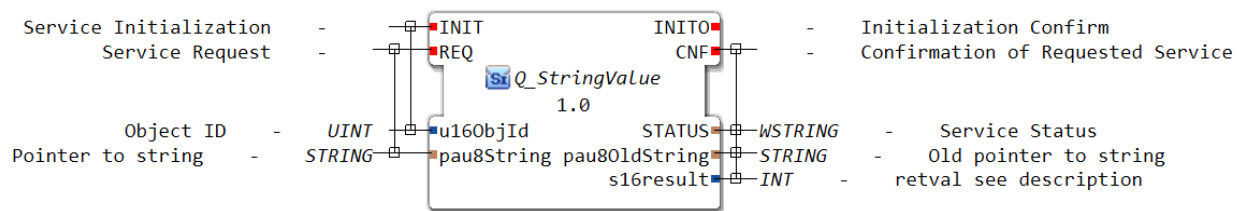
6.2.15.7.33 Q_SoftKeyMask



ISO 11783-6

Command change soft key mask(Part 6 - F.36)

6.2.15.7.34 Q_StringValue



Command Change String Value (Part 6 - F.24)

ISO 11783-6

This command sends a string to the VT. The length of the string including , ' may not be longer than the define ISO_VTC_CMD_STR_MAX_LENGTH. Take care that , ' is trailed.

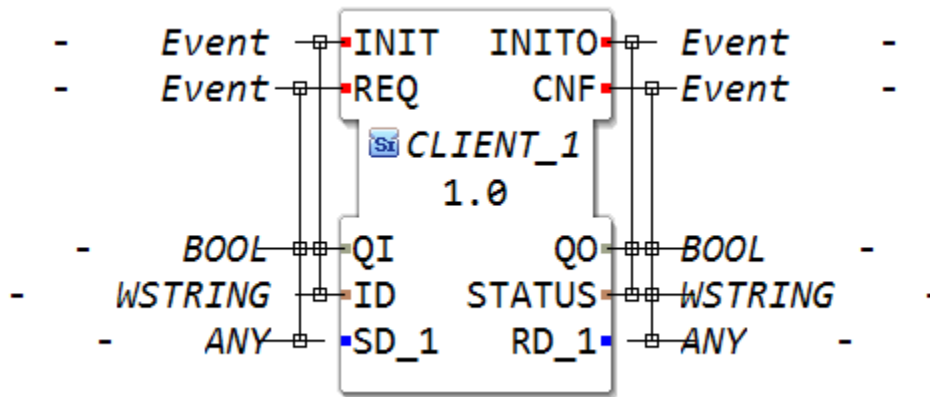
Return values

iso_s16

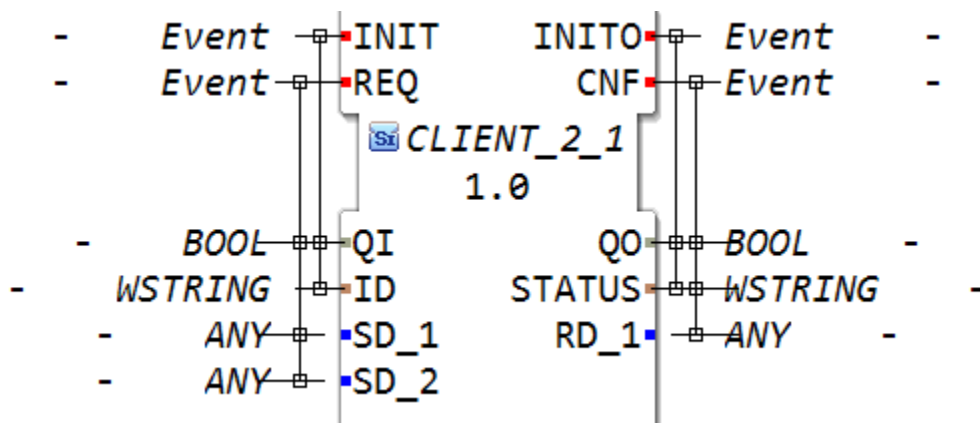
- E_NO_ERR - transport started
- E_OVERFLOW - buffer overflow
- E_NOACT - Command not possible in current state
- E_NO_INSTANCE - No VT client available

6.2.15.8 net

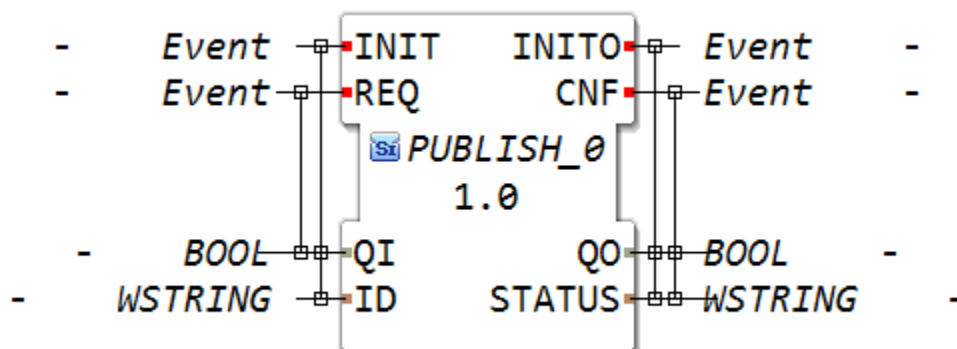
6.2.15.8.1 CLIENT_1



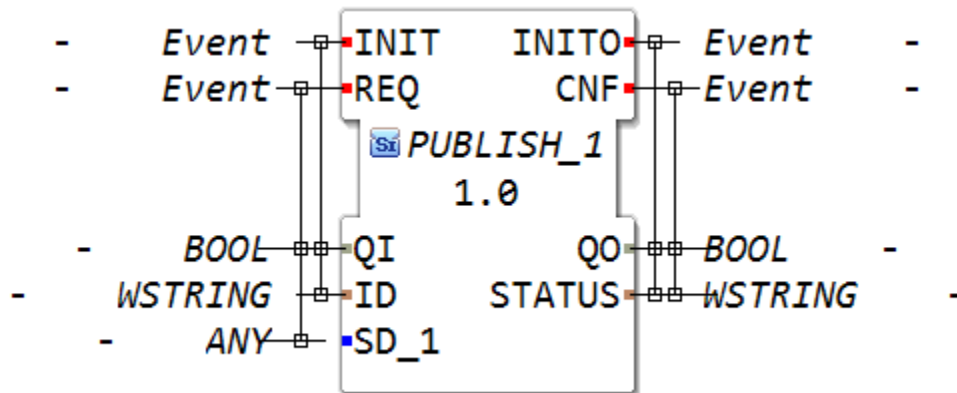
6.2.15.8.2 CLIENT_2_1



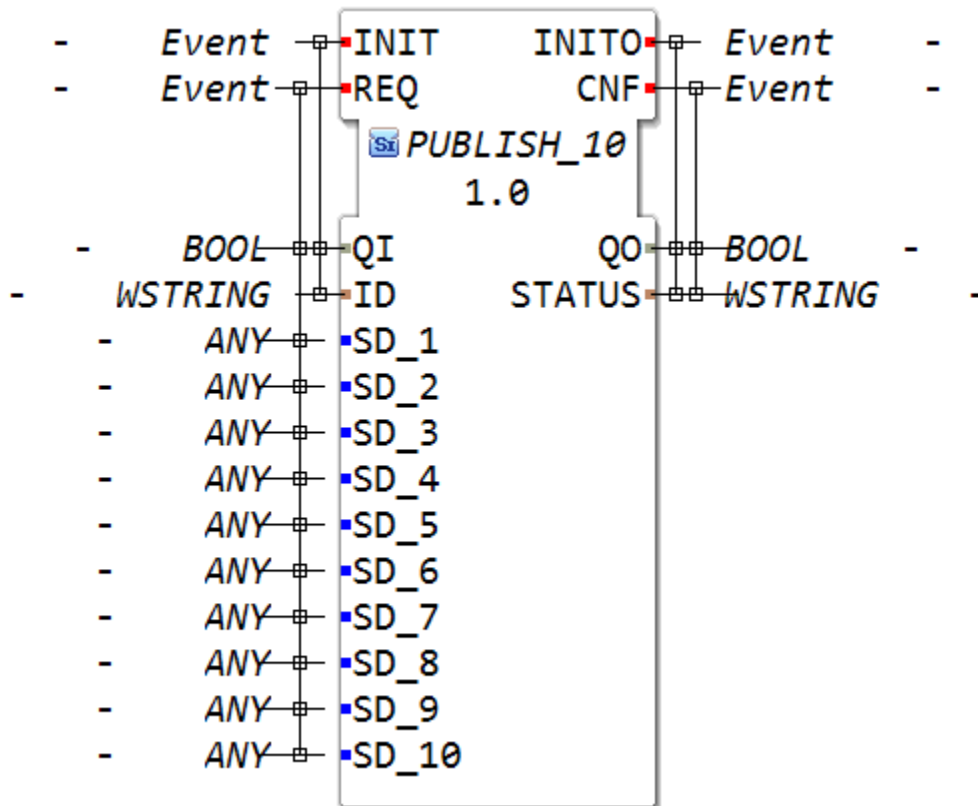
6.2.15.8.3 PUBLISH_0



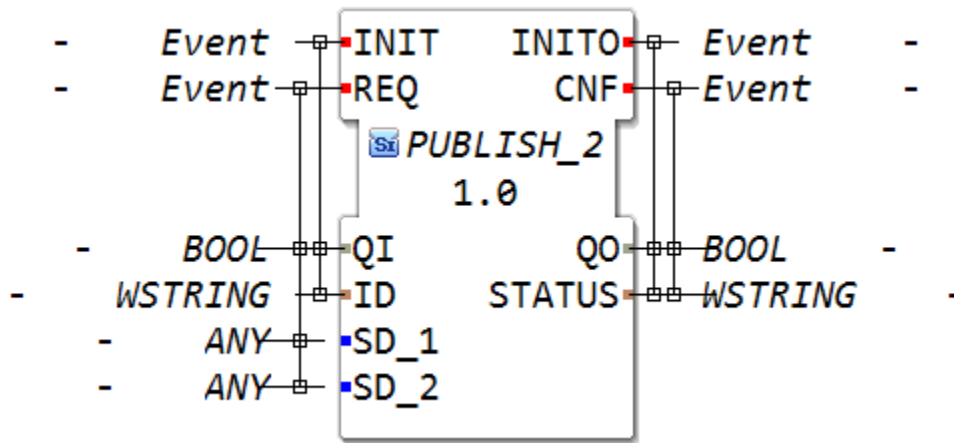
6.2.15.8.4 PUBLISH_1



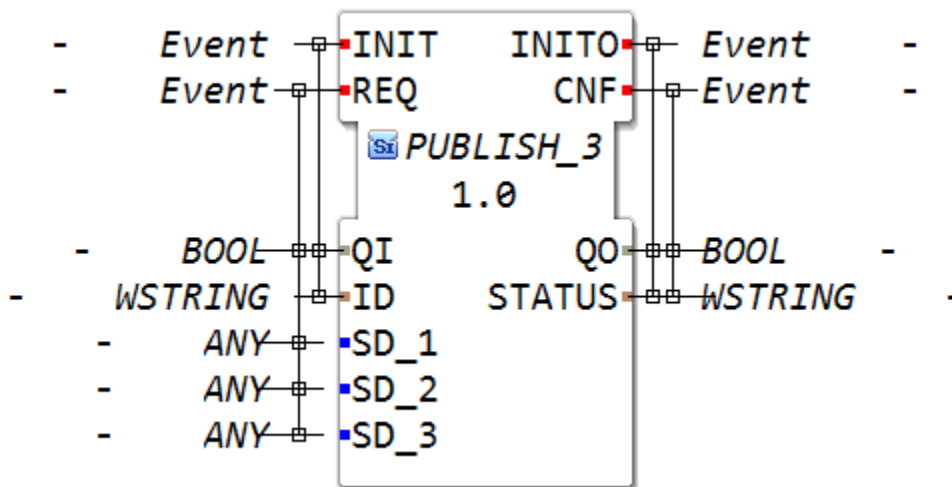
6.2.15.8.5 PUBLISH_10



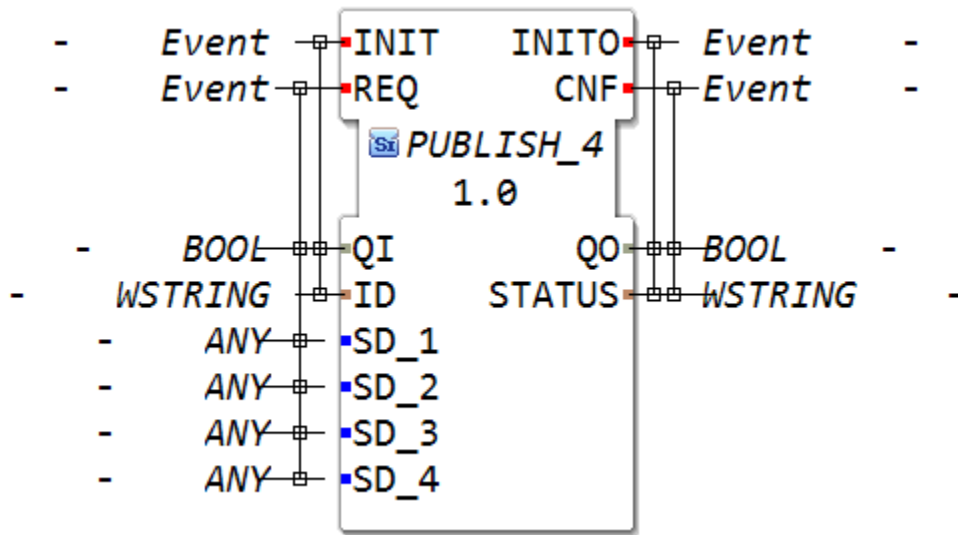
6.2.15.8.6 PUBLISH_2



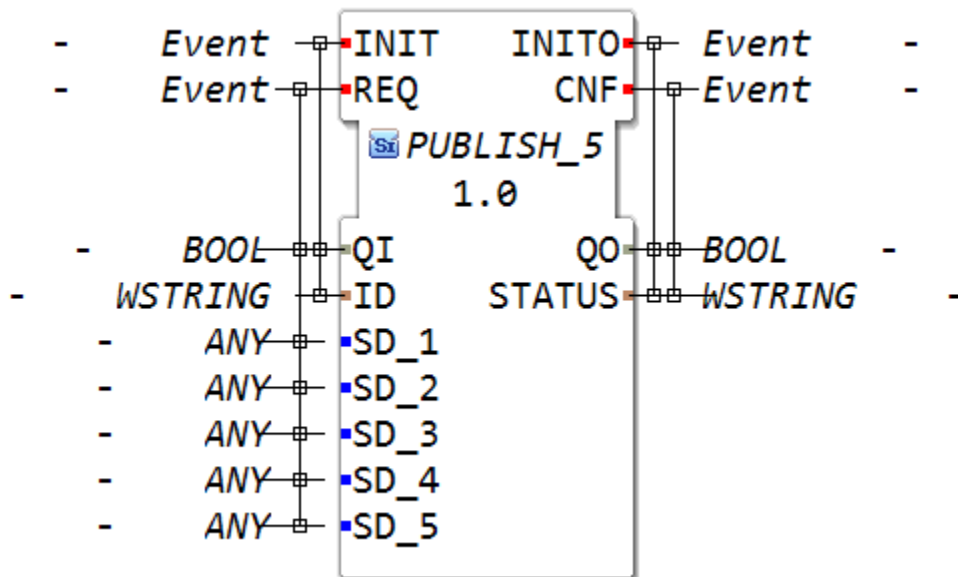
6.2.15.8.7 PUBLISH_3



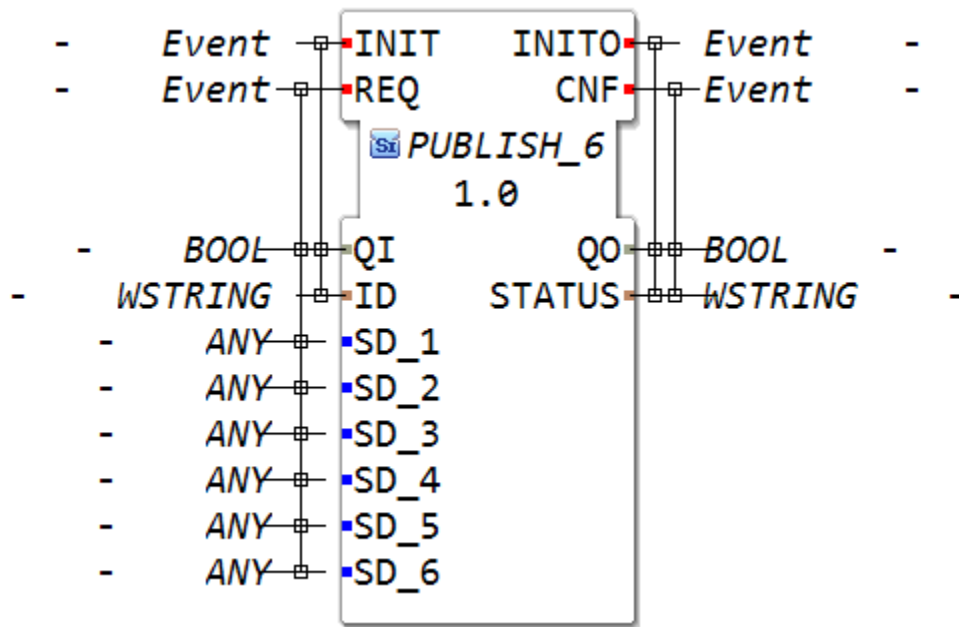
6.2.15.8.8 PUBLISH_4



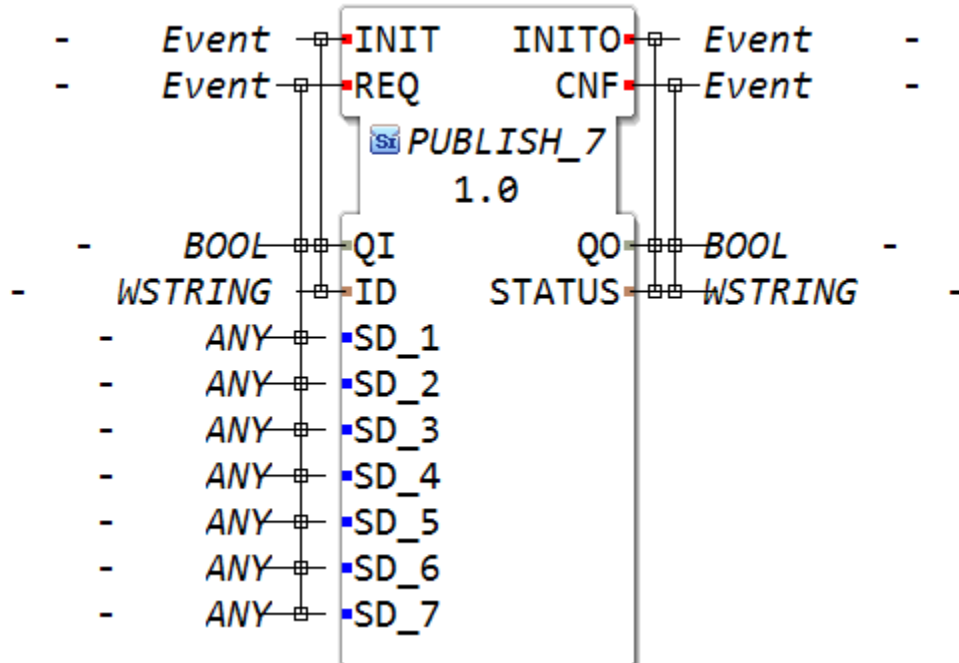
6.2.15.8.9 PUBLISH_5



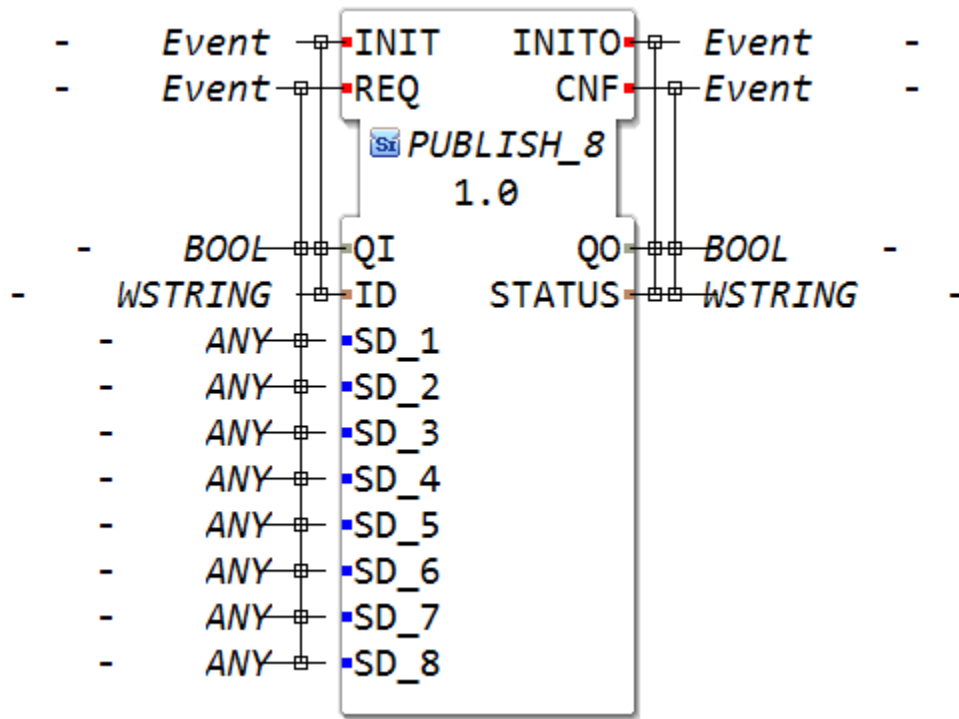
6.2.15.8.10 PUBLISH_6



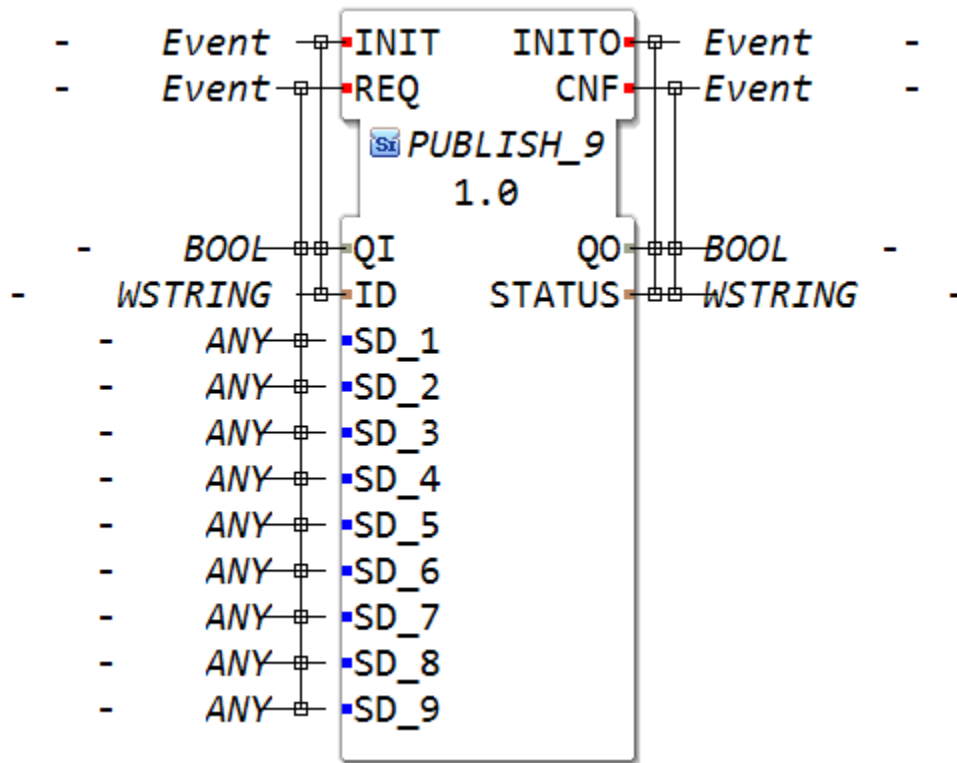
6.2.15.8.11 PUBLISH_7



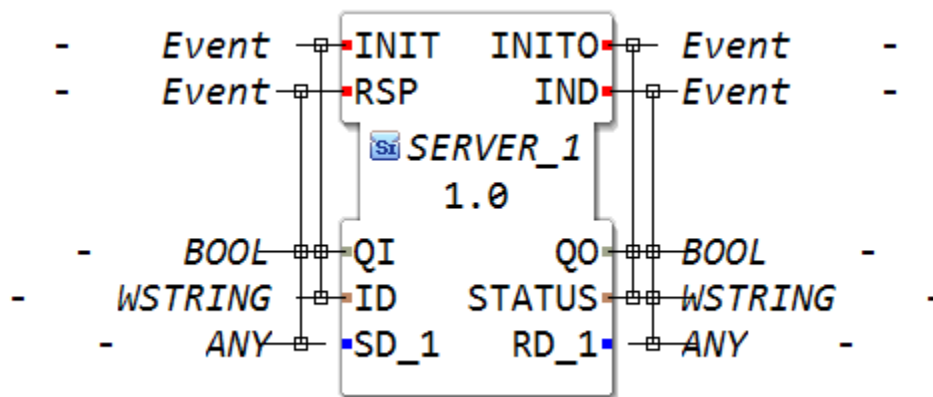
6.2.15.8.12 PUBLISH_8



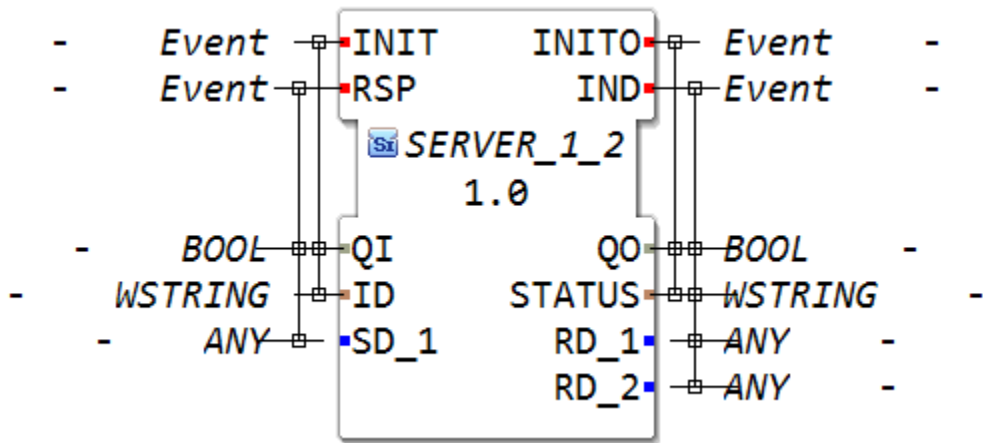
6.2.15.8.13 PUBLISH_9



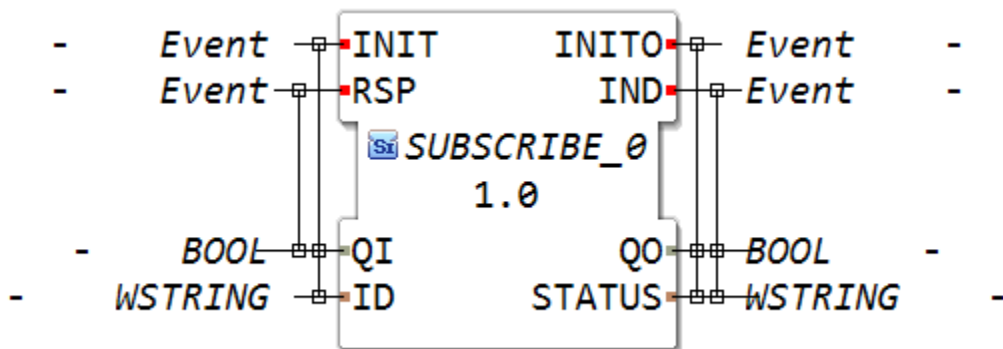
6.2.15.8.14 SERVER_1



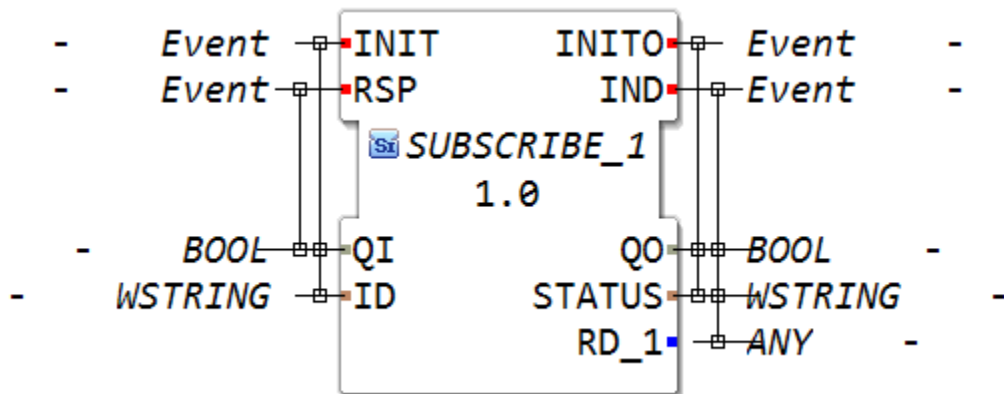
6.2.15.8.15 SERVER_1_2



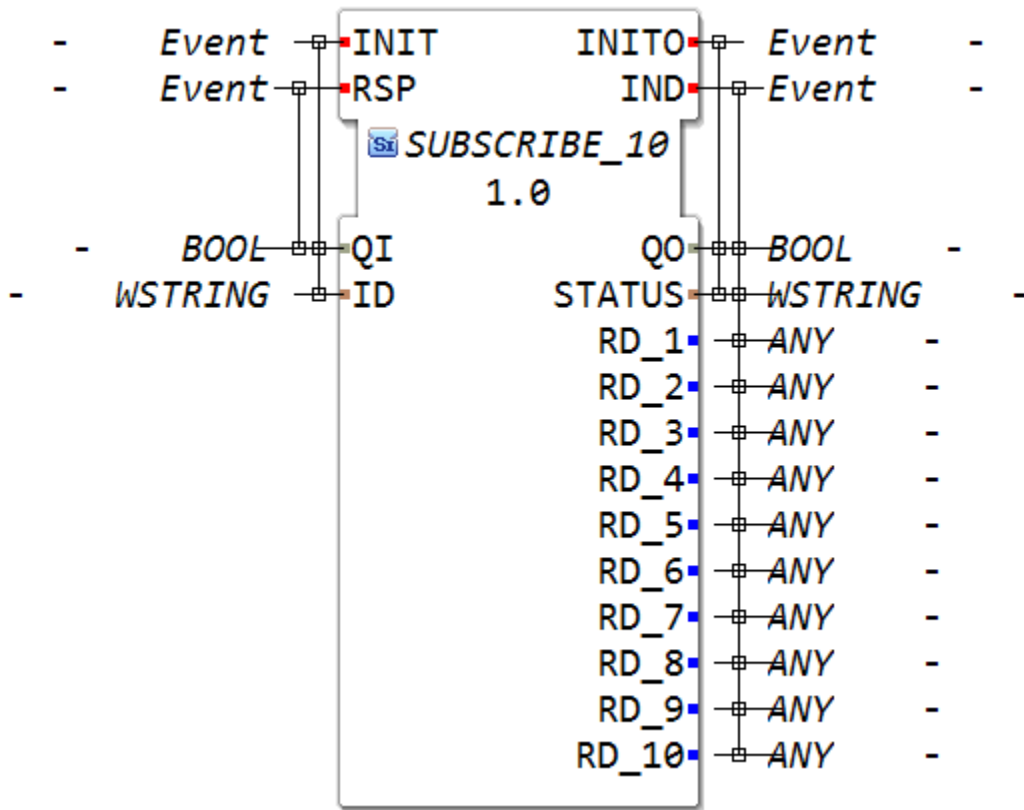
6.2.15.8.16 SUBSCRIBE_0



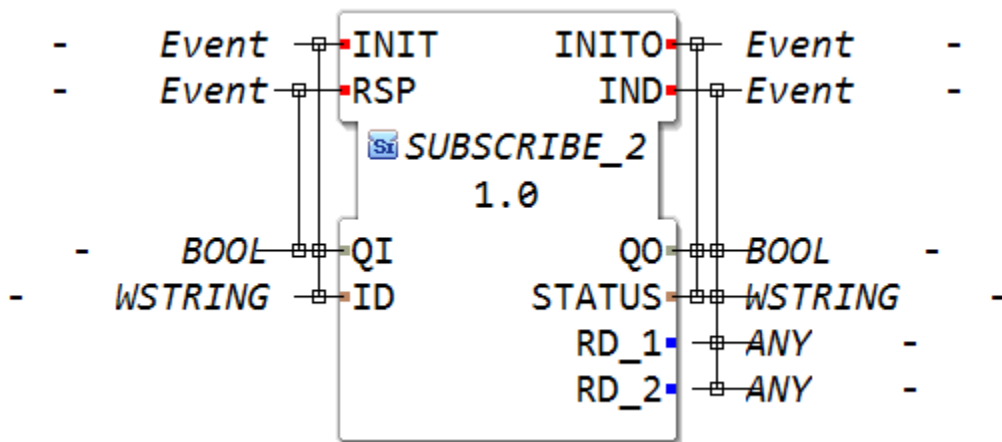
6.2.15.8.17 SUBSCRIBE_1



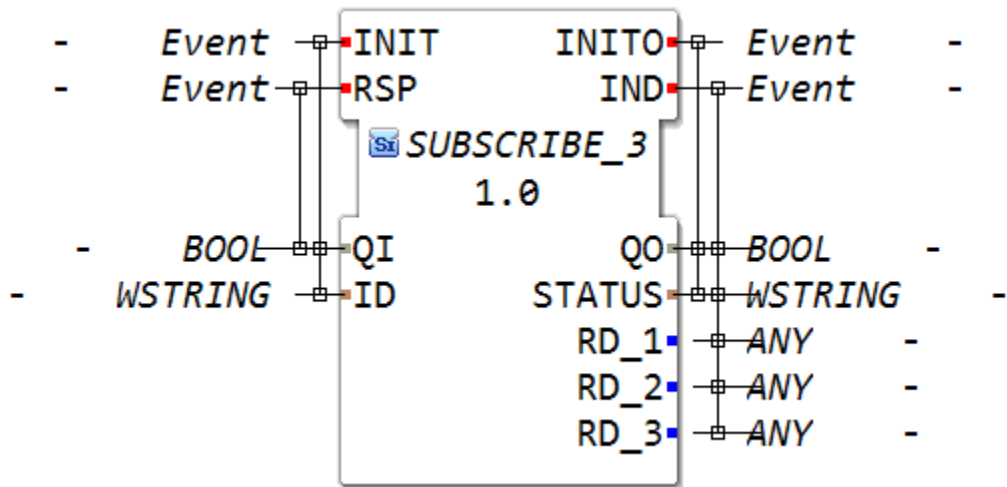
6.2.15.8.18 SUBSCRIBE_10



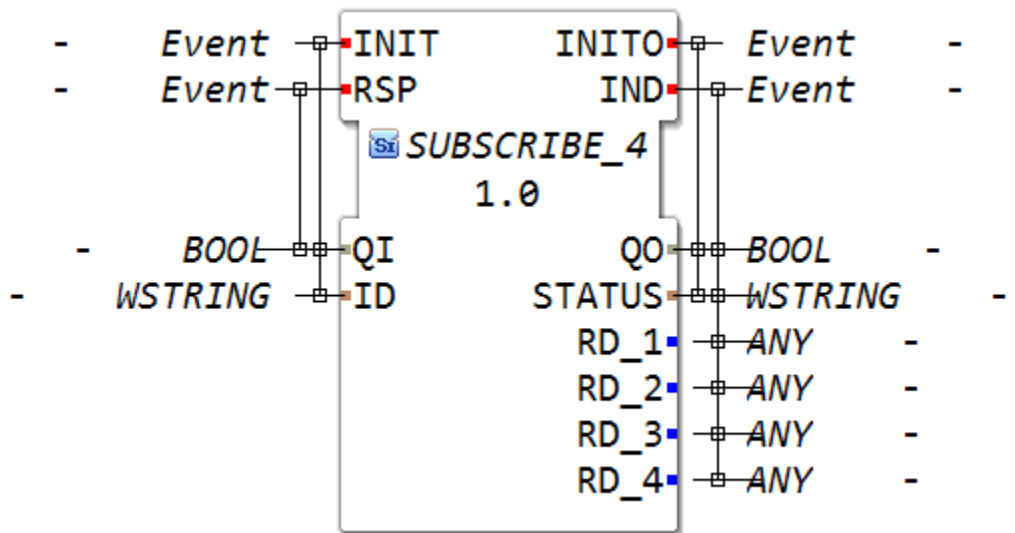
6.2.15.8.19 SUBSCRIBE_2



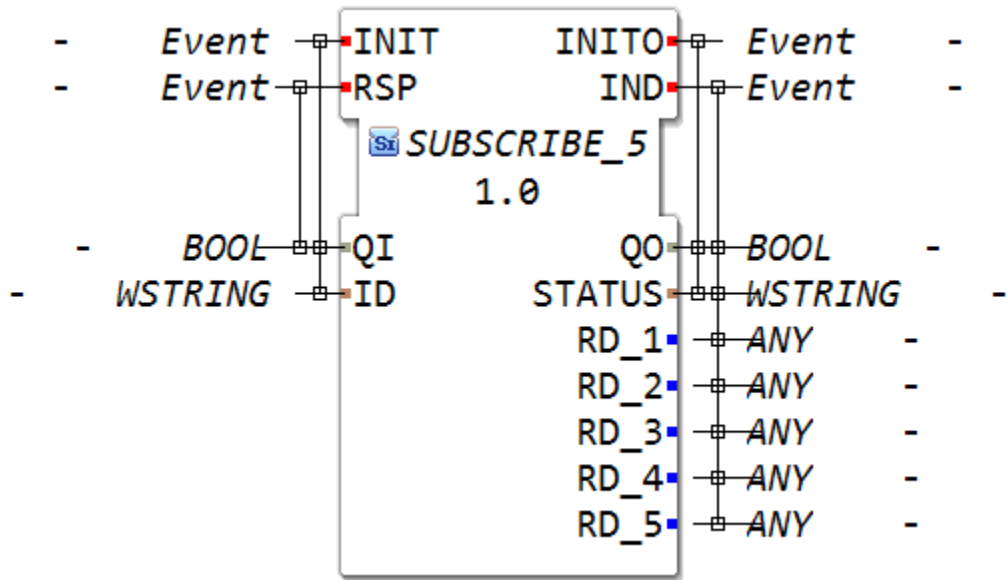
6.2.15.8.20 SUBSCRIBE_3



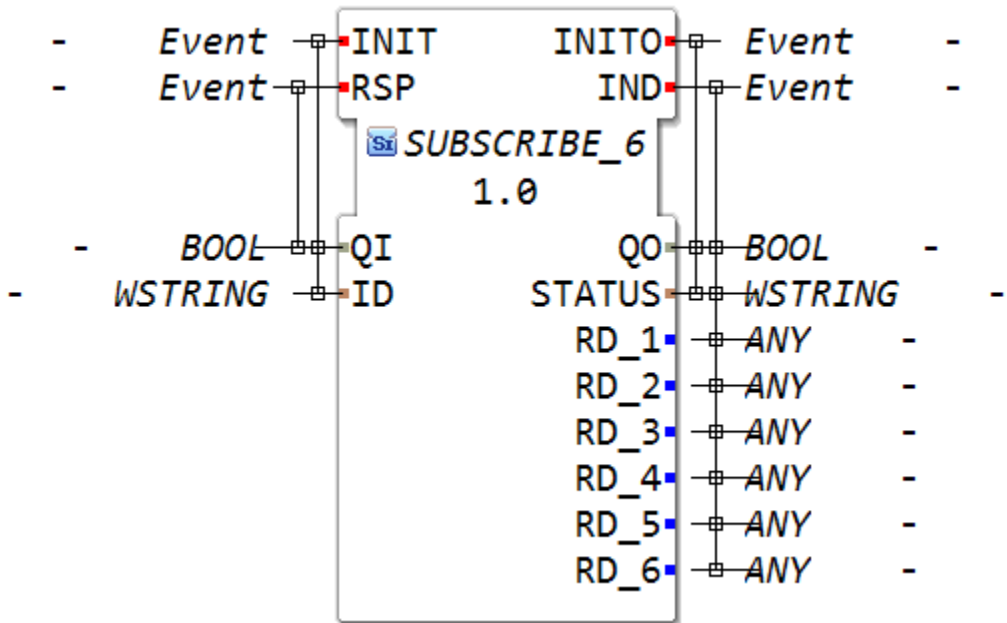
6.2.15.8.21 SUBSCRIBE_4



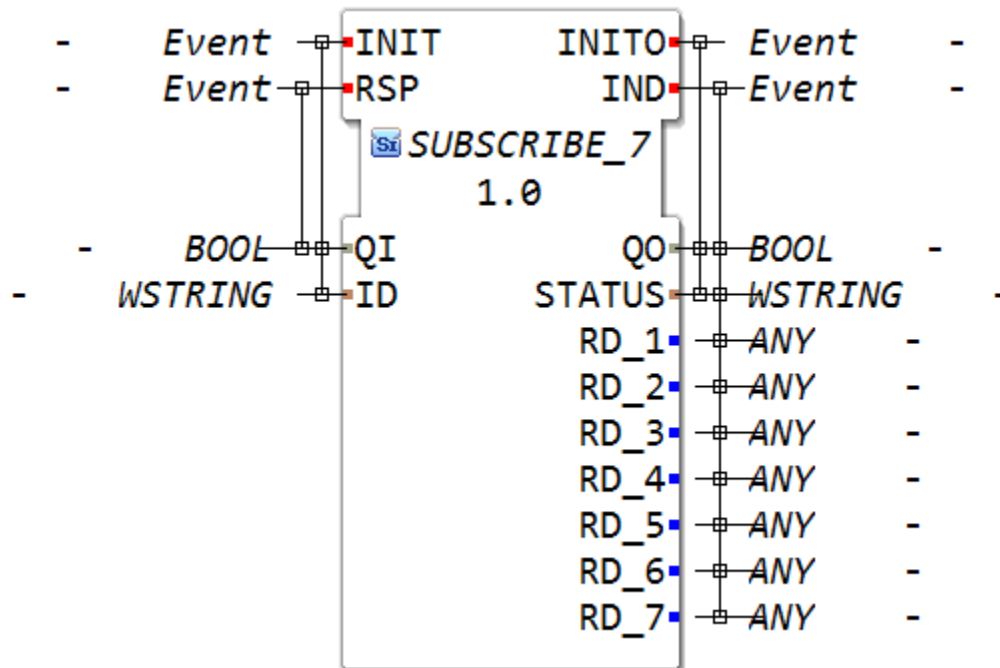
6.2.15.8.22 SUBSCRIBE_5



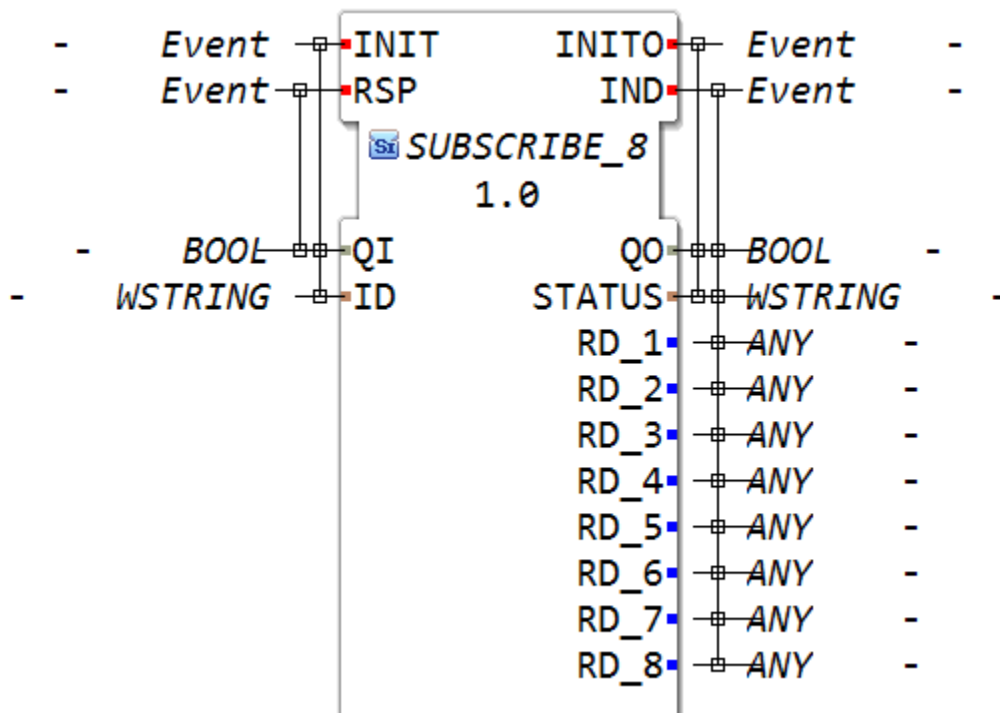
6.2.15.8.23 SUBSCRIBE_6



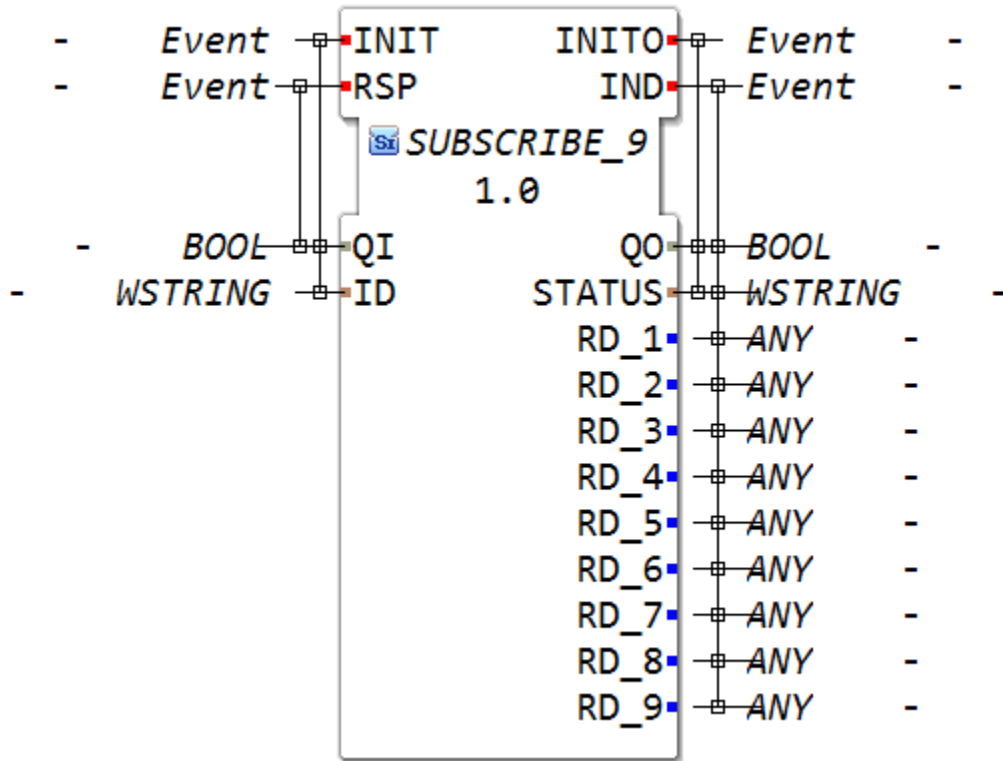
6.2.15.8.24 SUBSCRIBE_7



6.2.15.8.25 SUBSCRIBE_8

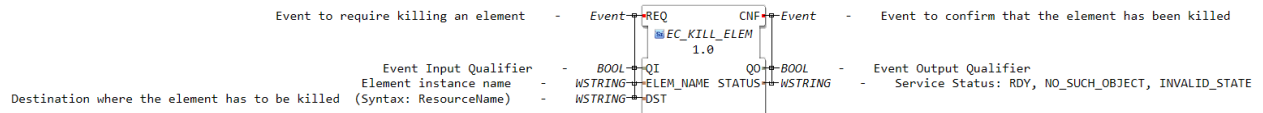


6.2.15.8.26 SUBSCRIBE_9

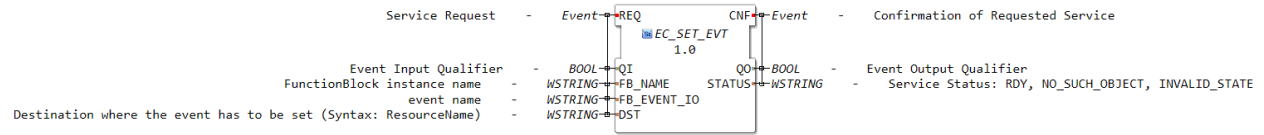


6.2.15.9 reconfiguration

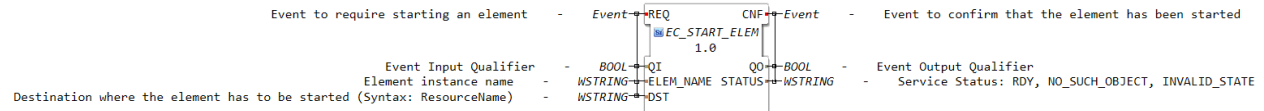
6.2.15.9.1 EC_KILL_ELEM



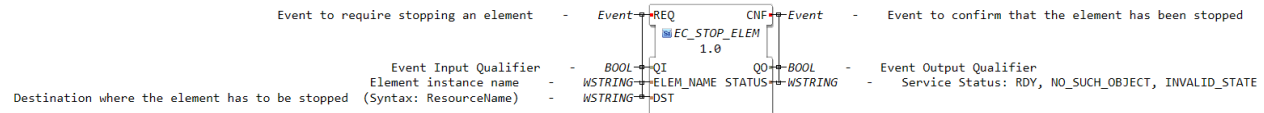
6.2.15.9.2 EC_SET_EVT



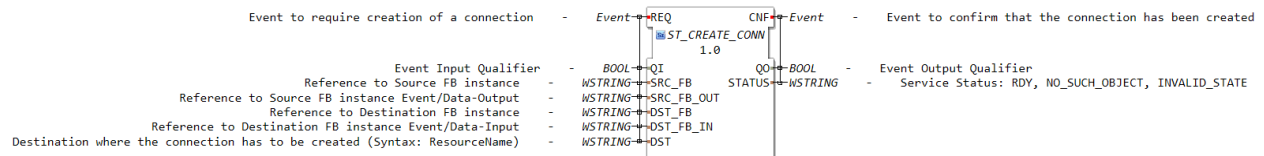
6.2.15.9.3 EC_START_ELEM



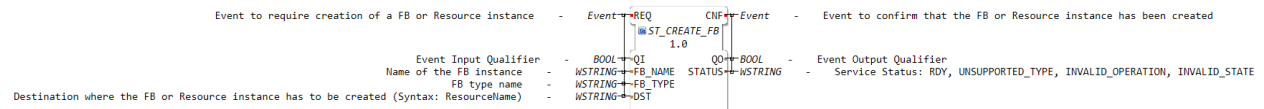
6.2.15.9.4 EC_STOP_ELEM



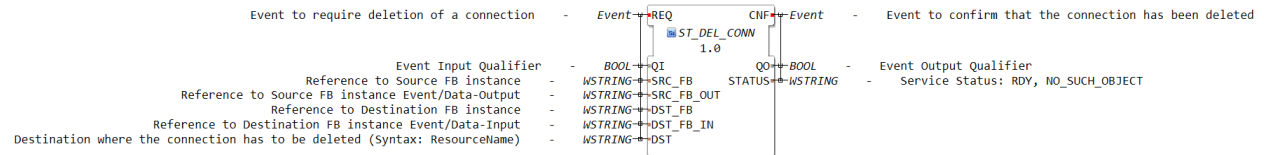
6.2.15.9.5 ST_CREATE_CONN



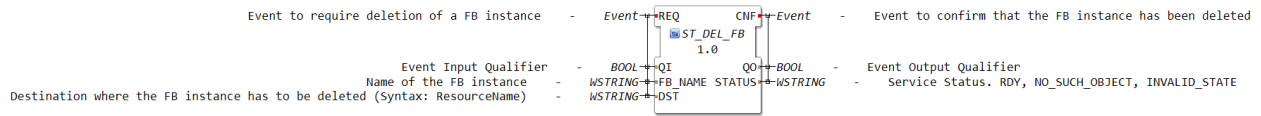
6.2.15.9.6 ST_CREATE_FB



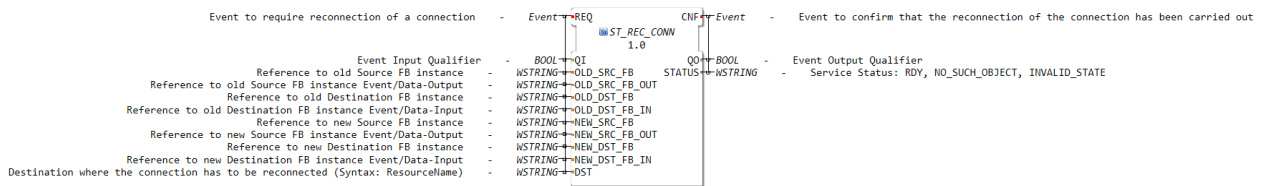
6.2.15.9.7 ST_DEL_CONN



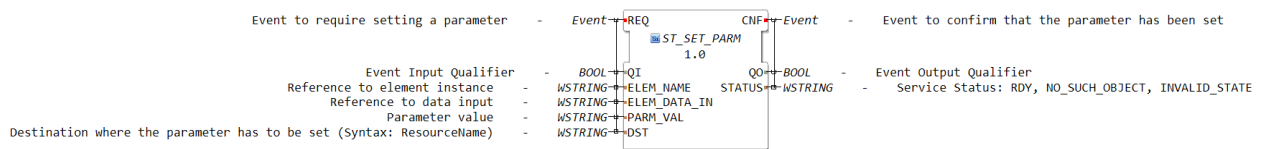
6.2.15.9.8 ST_DEL_FB



6.2.15.9.9 ST_REC_CONN



6.2.15.9.10 ST_SET_PARM



6.2.15.10 resources

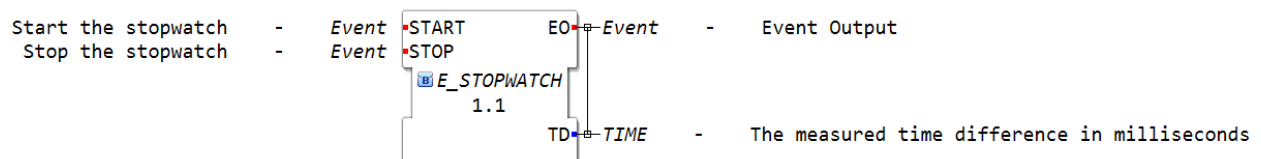
6.2.15.10.1 EMB_RES

6.2.15.11 segments

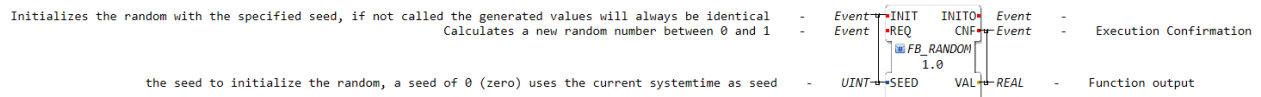
6.2.15.11.1 Ethernet

6.2.15.12 utils

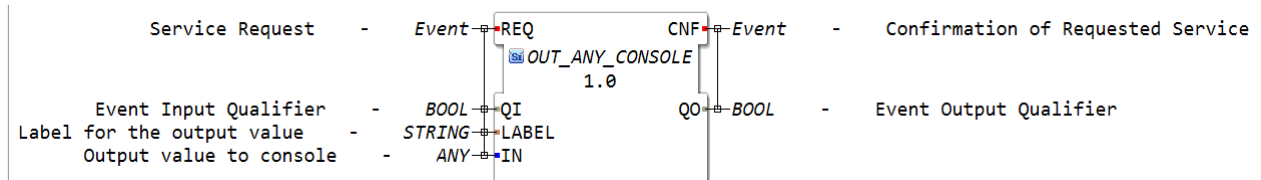
6.2.15.12.1 E_STOPWATCH



6.2.15.12.2 FB_RANDOM



6.2.15.12.3 OUT_ANY_CONSOLE



7.1 logiBUS®

<https://www.logibus.tech/>

7.2 ISOmaX®

<https://www.isomax.tech/>

7.3 Eclipse 4diac™

<https://www.eclipse.org/4diac/index.php>

https://eclipse.dev/4diac/en_news.php#en_news_20230809

Indices and tables

- [genindex](#)
- [search](#)

E

E_BUSY (*C macro*), 128
E_CANMSG_MISSED (*C macro*), 128
E_CHECKSUM (*C macro*), 128
E_COM (*C macro*), 128
E_CONFIG (*C macro*), 128
E_DATA_LINK (*C macro*), 128
E_DEFAULT (*C macro*), 128
E_ERROR_INDI (*C macro*), 128
E_NACK (*C macro*), 128
E_NO_ERR (*C macro*), 128
E_NO_INSTANCE (*C macro*), 128
E_NOACT (*C macro*), 128
E_NOT_AVAILABLE (*C macro*), 128
E_OUT_OF_MEMORY (*C macro*), 128
E_OVERFLOW (*C macro*), 129
E_PGN_UNKNOWN (*C macro*), 129
E_RANGE (*C macro*), 129
E_RD_WR (*C macro*), 129
E_RESPTOCMDERROR (*C macro*), 129
E_UNKNOWN_ERR (*C macro*), 129
E_USER_ERROR (*C macro*), 129
E_VALUE_WRONG (*C macro*), 129
E_WARN (*C macro*), 129

I

isobus

 Q_ActiveMask, 154
 Q_Attribute, 154
IsoVtcCloseInstance (*C function*), 129
IsoVtcCmd_ActiveMask (*C function*), 129
IsoVtcCmd_Attribute (*C function*), 130
IsoVtcCmd_BackgroundColour (*C function*), 130
IsoVtcCmd_ChangeObjectLabel (*C function*), 130
IsoVtcCmd_ChangePolygonPoint (*C function*), 131
IsoVtcCmd_ChangePolygonScale (*C function*), 131
IsoVtcCmd_ChildLocation (*C function*), 132
IsoVtcCmd_ChildPosition (*C function*), 132
IsoVtcCmd_CtrlAudioSignal (*C function*), 133

IsoVtcCmd_DeleteVersion (*C function*), 133
IsoVtcCmd_EndPoint (*C function*), 134
IsoVtcCmd_ESC (*C function*), 134
IsoVtcCmd_ExecuteExtendedMacro (*C function*), 134
IsoVtcCmd_ExecuteMacro (*C function*), 135
IsoVtcCmd_GetAttribute (*C function*), 136
IsoVtcCmd_GetSupportedObjects (*C function*), 136
IsoVtcCmd_GetSupportedWidechar (*C function*), 137
IsoVtcCmd_GetWindowMaskData (*C function*), 137
IsoVtcCmd_GraphicsContext (*C function*), 137
IsoVtcCmd_IdentifyVT (*C function*), 138
IsoVtcCmd_ListItem (*C function*), 139
IsoVtcCmd_LockUnlockMask (*C function*), 139
IsoVtcCmd_NumericValue (*C function*), 140
IsoVtcCmd_NumericValueRef (*C function*), 140
IsoVtcCmd_ObjEnableDisable (*C function*), 140
IsoVtcCmd_ObjHideShow (*C function*), 141
IsoVtcCmd_ObjSelectInput (*C function*), 141
IsoVtcCmd_Priority (*C function*), 141
IsoVtcCmd_ScreenCapture (*C function*), 142
IsoVtcCmd_SelectActiveWorkingSet (*C function*),
 142
IsoVtcCmd_SelectColourMap (*C function*), 142
IsoVtcCmd_SetAudioVolume (*C function*), 143
IsoVtcCmd_Size (*C function*), 143
IsoVtcCmd_SoftKeyMask (*C function*), 144
IsoVtcCmd_StoreVersion (*C function*), 144
IsoVtcCmd_String (*C function*), 144
IsoVtcCmd_StringRef (*C function*), 145
IsoVtcContinue (*C function*), 145
IsoVtcCreateInstance (*C function*), 145
IsoVtcDataSet (*C function*), 146
IsoVtcGetInstanceID (*C function*), 147
IsoVtcGetStatusInfo (*C function*), 147
IsoVtcMultipleNextVT (*C function*), 148
IsoVtcObjTypeIsSupported (*C function*), 148
IsoVtcObjTypeParsableSet (*C function*), 149
IsoVtcReadWorkingSetLanguageData (*C function*),
 149
IsoVtcRestartInstance (*C function*), 149

IsoVtcSetNewPreferredVT (*C function*), [150](#)
IsoVtcVersionStringGet (*C function*), [150](#)

Q

Q_ActiveMask
 [isobus](#), [154](#)
Q_Attribute
 [isobus](#), [154](#)