ECOM DEVICES

Program Support for Siemens PLC S7-1200/S7-1500

Date: **25.04.2023**

Author: **Filip Kinovič (filip.kinovic@ecomsro.cz)**

Content

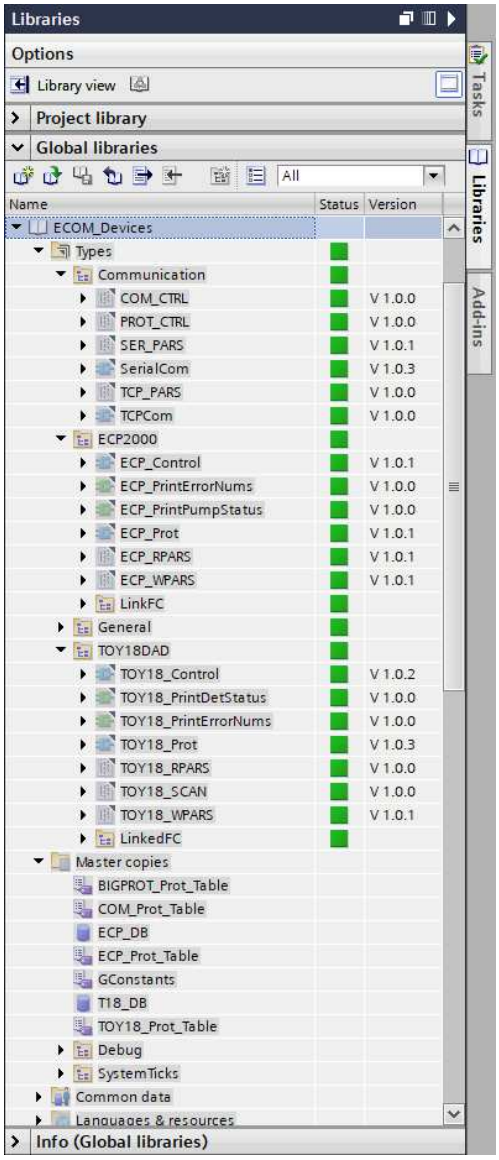
Library ECOM_Devices 4
Project example 1 (S71200_toy18_test_tcp1.ap17) 5
Project tree 5
Main [OB1] 6
T18_DB [DB101] 8
TOY18_Prot_Table 9
COM_Prot_Table 10
Project functionality 11
Project example 2 (S71200_toy18_test_ser1.ap17) 14
Project example 3 (S71200_ecp_test_tcp1.ap17) 15
Project tree 15
Main [OB1] 15
ECP_DB [DB102] 17
BIGPROT_Prot_Table 18
ECP_Prot_Table 19
Project functionality 20
Overview of main function blocks 21
TCPCom 21
SerialCom 22
TOY18_Prot 23
TOY18_Control 24
ECP_Prot 25
ECP_Control 26
Overview of types 27
TCP_PARS 27
SER_PARS 27
COM_CTRL 27
PROT_CTRL 27
TOY18_SCAN 27
TOY18_WPARS 28
TOY18_RPARS 28
ECP_WPARS 29
ECP_RPARS 30
List of TOY18 constants (TOY18_Prot_Table) 32

List of COM constants (COM_Prot_Table) 35
List of BIGProtocol constants (BIGPROT_Prot_Table) 35
List of ECP2000 constants (ECP_Prot_Table) 36
List of functions in ECOM_DEVICES library 37

Library ECOM_Devices

Library ECOM_Devices was designed in TIA Portal V17 and it includes basic parts for support of ECOM devices using TCP or serial communication. Functions and functions block are created from blocks available for both PLC models S7-1200 and S7-1500. Library is opened so it is possible to edit blocks according to needs. The most of functions and function blocks are written in SCL language.

Important parts:

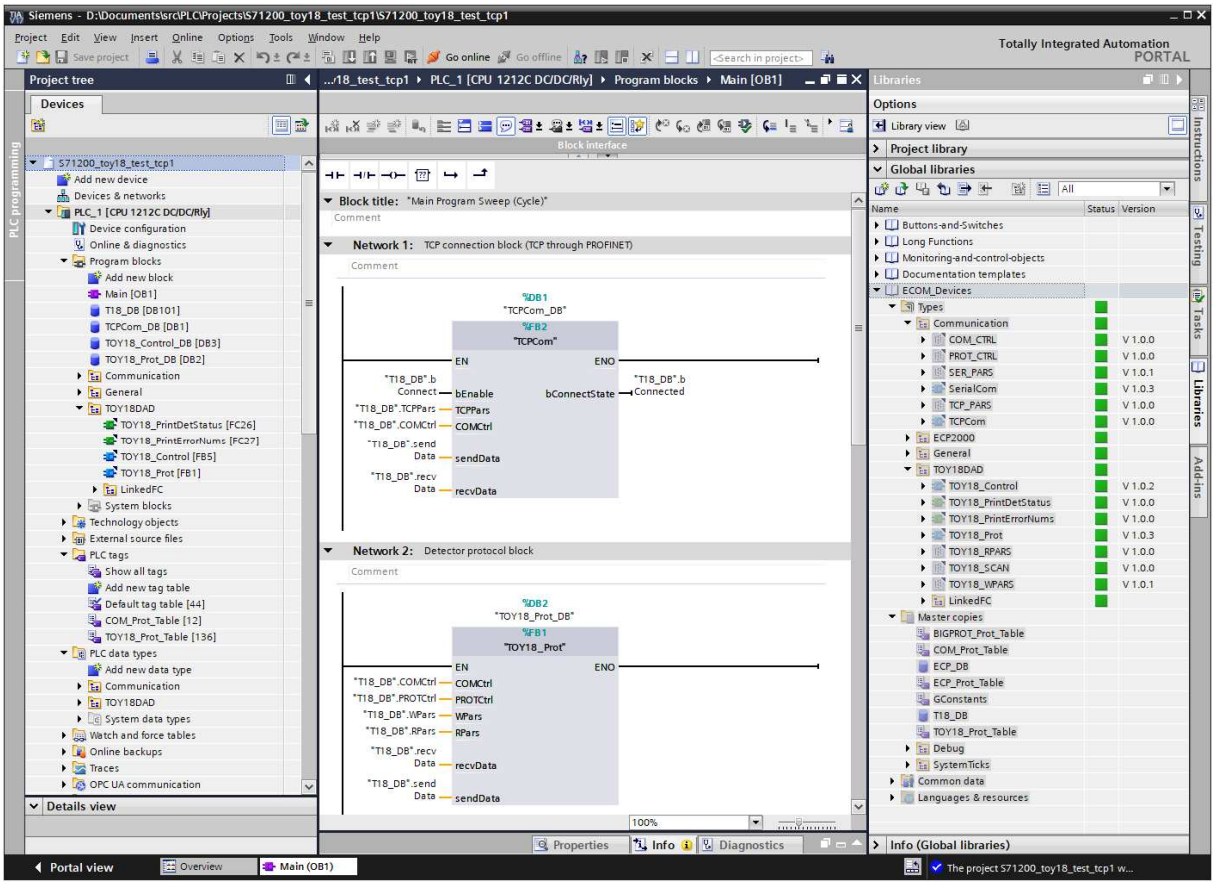


- Communication blocks TCPCom and SerialCom with its data type definitions.
- Detector TOY18DAD main blocks TOY18_Prot and TOY18_Control with its data type definitions (TOY18_SCAN, TOY18_WPARS, TOY18_RPARS) and support functions (LinkedFC).
- Pump ECP2000 main blocks ECP_Prot and ECP_Control with its data type definitions (ECP_WPARS, ECP_RPARS) and support functions.
- Additional useful functions (General).
- General data types TCP_PARS, SER_PARS, COM_CTRL and PROT_CTRL.
- Constant tag tables COM_Prot_table, BIGPROT_Prot_Table, TOY18_Prot_Table and ECP_Prot_Table with definitions of constants for parameters, commands, states and errors.
- Example data blocks T18_DB and ECP_DB.
- Additional useful blocks Debug and SystemTicks for testing purpose.

Project example 1 (S71200_toy18_test_tcp1.ap17)

Example project 1 demonstrates using of above library. Create new project and configure PLC. Add function blocks “TCPCom”, “TOY18_Prot” and “TOY18_Control”, tag tables “COM_Prot_Table”, “TOY18_Prot_Table” and data block “T18_DB” into project from ECOM_DEVICES library. Insert main function block into networks and define its parameters to match T18_DB. See pictures below.

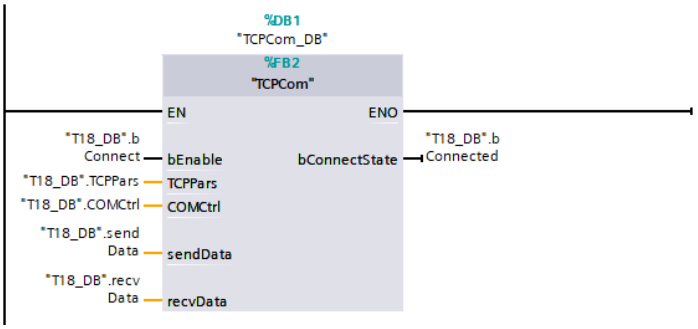
Project tree



Main [OB1]

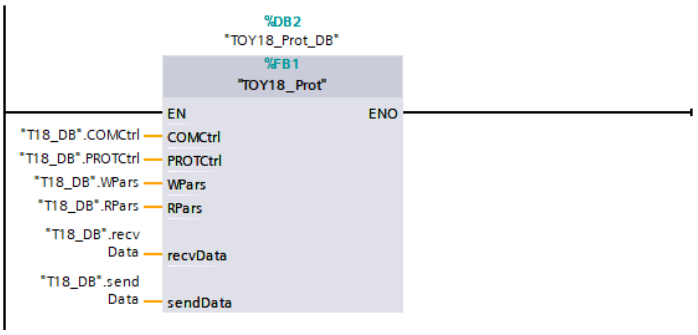
Network 1: TCP connection block (TCP through PROFINET)

Comment



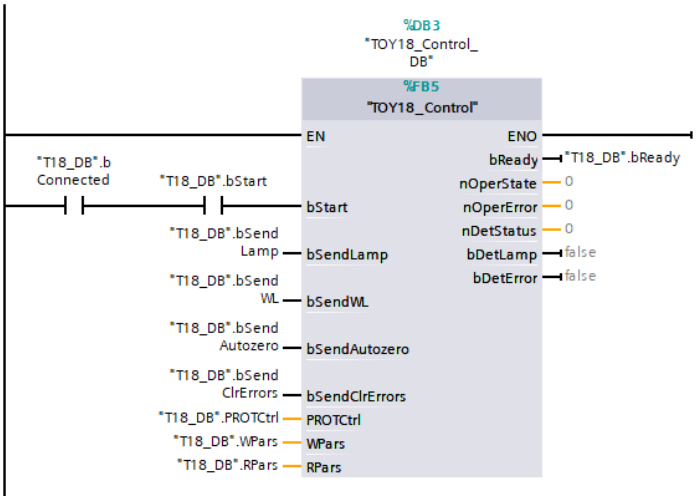
Network 2: Detector protocol block

Comment

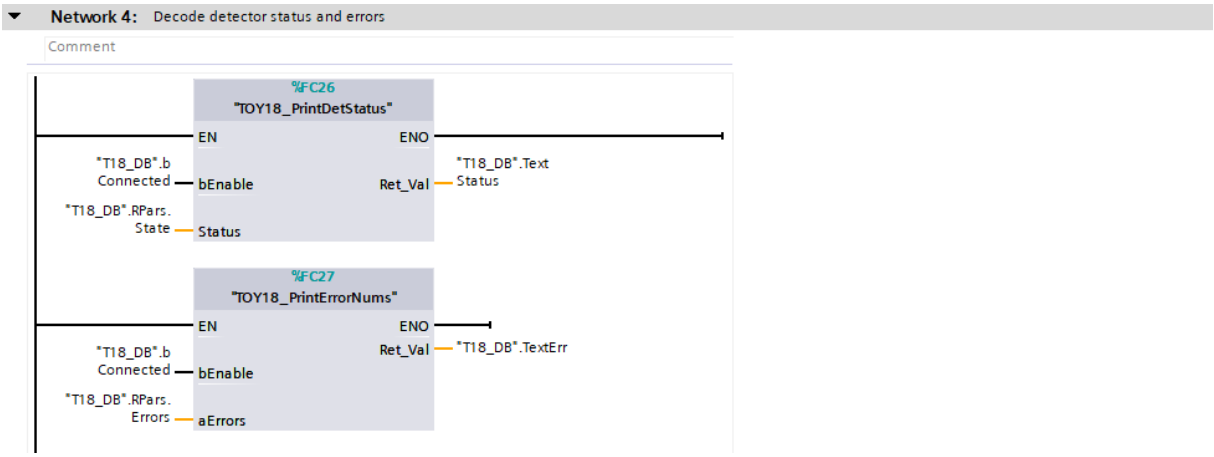


Network 3: Detector control block

Comment



Network 4 demonstrates additional functions for decoding detector status and errors to strings.



T18_DB [DB101]

Example of interconnection data block T18_DB. Set default parameters to set adequate demands.

T18_DB

	Name	Data type	Start value	Monitor value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static								
2	TCPPars	"TCP_PARS"				☑	☑	☑	
3	ID	CONN_OUC	1			☑	☑	☑	
4	RemoteAddress	IP_V4				☑	☑	☑	
5	RemotePort	UInt	10001			☑	☑	☑	
6	SerPars	"SER_PARS"				☑	☑	☑	☑
7	Port	UInt	"Local~CM_1241_(RS232)_1"			☑	☑	☑	
8	BaudIdx	UInt	"BAUDIDX_57600"			☑	☑	☑	
9	ParityIdx	UInt	"PARITYIDX_NONE"			☑	☑	☑	
10	DatabitsIdx	UInt	"DATABITSIDX_8"			☑	☑	☑	
11	StopbitsIdx	UInt	"STOPBITSIDX_1"			☑	☑	☑	
12	bConnect	Bool	false			☑	☑	☑	☑
13	bConnected	Bool	false			☑	☑	☑	
14	COMCtrl	"COM_CTRL"				☑	☑	☑	
15	PROTCtrl	"PROT_CTRL"				☑	☑	☑	☑
16	sendData	Array[0..127] of Byte				☑	☑	☑	
17	recvData	Array[0..8191] of B...				☑	☑	☑	
18	WPars	"TOY18_WPARS"				☑	☑	☑	
19	Lamp	Bool	true			☑	☑	☑	
20	Wavelength	Array[0..3] of UInt				☑	☑	☑	
21	Rate	USInt	"SAMPLE_RATE_10HZ"			☑	☑	☑	
22	TimeConstant	USInt	"TIMECONST_1S"			☑	☑	☑	
23	BandWidth	USInt	8			☑	☑	☑	
24	Beep	USInt	2			☑	☑	☑	
25	MathDFunction	USInt	"MATHDFCE_OFF"			☑	☑	☑	
26	NegRange	USInt	"NEGRNG_100MAU"			☑	☑	☑	
27	SoundMask	USInt	16#0E			☑	☑	☑	
28	LeakMode	USInt	"LEAKMODE_OFF"			☑	☑	☑	
29	KeyLock	USInt	"KEYLOCK_LOCK"			☑	☑	☑	
30	DisableRMT	Bool	false			☑	☑	☑	
31	IOState	USInt	0			☑	☑	☑	
32	Section	USInt	0			☑	☑	☑	
33	Subs	Struct				☑	☑	☑	
34	MinScanWL	UInt	200			☑	☑	☑	
35	MaxScanWL	UInt	400			☑	☑	☑	
36	RPars	"TOY18_RPARS"				☑	☑	☑	☑
37	bStart	Bool	true			☑	☑	☑	☑
38	bSendLamp	Bool	false			☑	☑	☑	
39	bSendWL	Bool	false			☑	☑	☑	
40	bSendAutozero	Bool	false			☑	☑	☑	
41	bSendClrErrors	Bool	false			☑	☑	☑	
42	bReady	Bool	false			☑	☑	☑	
43	TextStatus	String	''			☑	☑	☑	
44	TextErr	String	''			☑	☑	☑	

TOY18_Prot_Table

Tag table includes global user constants, which are used inside function blocks and support functions (e.g. TOY18_CreateCommandStr, TOY18_DecodeResponse). It needs to be part of a project!

Tags | User constants

TOY18_Prot_Table

	Name	Data type	Value	Comment
1	DETTYPE_UNKNOWN	USInt	0	
2	DETTYPE_TOY18	USInt	1	
3	DETTYPE_TOY20	USInt	2	
4	DETTYPE_BABY	USInt	3	
5	DETTYPE_TOY	USInt	4	
6	LAMPSTATE_OFF	USInt	0	
7	LAMPSTATE_ON	USInt	1	
8	LAMPSTATE_ERR	USInt	2	
9	LAMPSTATE_ERRI	USInt	3	
10	LAMPSTATE_ERRV	USInt	4	
11	SAMPLE_RATE_01HZ	USInt	1	
12	SAMPLE_RATE_02HZ	USInt	2	
13	SAMPLE_RATE_05HZ	USInt	5	
14	SAMPLE_RATE_10HZ	USInt	10	
15	SAMPLE_RATE_20HZ	USInt	20	
16	TIMECONST_500MS	USInt	0	
17	TIMECONST_750MS	USInt	1	
18	TIMECONST_1S	USInt	2	
19	TIMECONST_2S	USInt	3	
20	TIMECONST_4S	USInt	4	
21	TIMECONST_8S	USInt	5	
22	TIMECONST_16S	USInt	6	
23	TIMECONST_200MS	USInt	7	
24	TIMECONST_100MS	USInt	8	
25	MATHDFCE_OFF	USInt	0	D
26	MATHDFCE_SUM_AB	USInt	1	D=A+B
27	MATHDFCE_SUB_AB	USInt	2	D=A-B
28	MATHDFCE_AVER_AB	USInt	3	D=(A+B)/2
29	MATHDFCE_SUB_AB2	USInt	4	D(A-B)/2
30	MATHDFCE_MAX_AB	USInt	5	D=MAX(A,B)
31	MATHDFCE_DIV_AB	USInt	6	D=A/B
32	MATHDFCE_NEG_A	USInt	7	D=-A
33	MATHDFCE_ABS_A	USInt	8	D=|A|
34	MATHDFCE_LIN_A	USInt	9	D=A*M+O (from v1.33)
35	SOUNDMASK_OFF	USInt	16#00	
36	SOUNDMASK_KEYBOARD	USInt	16#01	
37	SOUNDMASK_ERRORS	USInt	16#02	

See List of TOY18 constants (TOY18_Prot_Table) on page 32 for details.

COM_Prot_Table

Tag table includes global constants for serial communication (baudrates, databits, parity, stopbits). It needs to be part of a project when SER_PARS variable is included (T18_DB)!

Tags | User constants

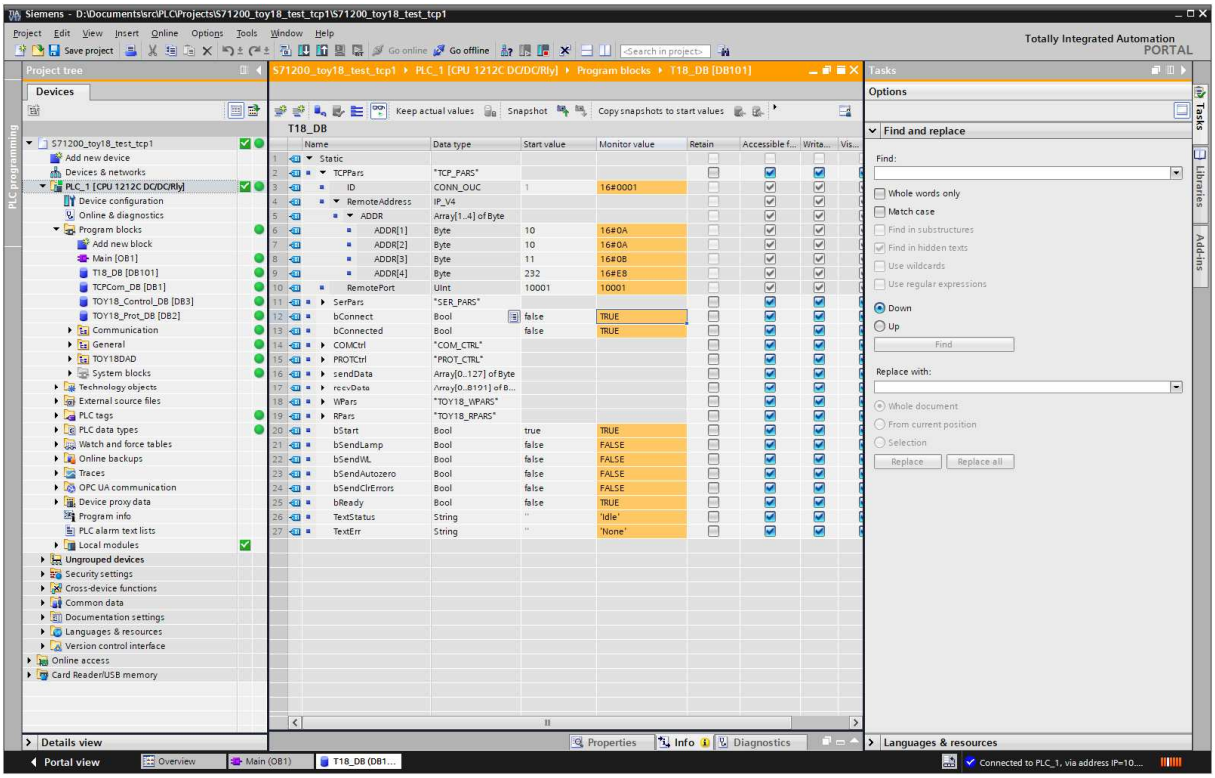
COM_Prot_Table

	Name	Data type	Value	Comment
1	BAUDIDX_9600	UInt	6	Index for baudrate 9600bd
2	BAUDIDX_19200	UInt	7	Index for baudrate 19200bd
3	BAUDIDX_38400	UInt	8	Index for baudrate 38400bd
4	BAUDIDX_57600	UInt	9	Index for baudrate 57600bd
5	BAUDIDX_115200	UInt	11	Index for baudrate 115200bd
6	PARITYIDX_NONE	UInt	1	Index for no parity
7	PARITYIDX_EVEN	UInt	2	Index for even parity
8	PARITYIDX_ODD	UInt	3	Index for odd parity
9	DATABITSIDX_8	UInt	1	Index for 8 databits
10	DATABITSIDX_7	UInt	2	Index for 7 databits
11	STOPBITSIDX_1	UInt	1	Index for 1 stopbit
12	STOPBITSIDX_2	UInt	2	Index for 2 stopbits
13	<Add new>			

See List of COM constants (COM_Prot_Table) on page 35 for details.

Project functionality

Project can be tested when it was compiled and downloaded to PLC successfully. Parameter T18_DB.TCPPars.RemoteAddress needs to be set correctly to connect real detector. Connection is started by toggling parameter T18_DB.bConnect from FALSE to TRUE. Successful connection is signaled by TRUE parameter T18_DB.bConnected. Picture below shows situation when data block T18_DB is watched.



Because parameter T18_DB.bStart was already TRUE, communication with detector controlled by TOY18_Control was started. Status subscription should be running. Actual detector status is “Idle”. The output T18_DB.bReady should be TRUE.

Lamp can be started by setting parameter T18_DB.WPars.Lamp to TRUE and changing parameter T18_DB.bSendLamp from FALSE to TRUE. Measurement state would be reached after lamp is on and status and absorbance subscriptions should be running.

20	bStart	Bool	true	TRUE
21	bSendLamp	Bool	false	TRUE
22	bSendWL	Bool	false	FALSE
23	bSendAutozero	Bool	false	FALSE
24	bSendClrErrors	Bool	false	FALSE
25	bReady	Bool	false	TRUE
26	TextStatus	String	''	'Measure'
27	TextErr	String	''	'None'

Absorbance is read using asynchronous reception of absorbance subscription into variable T18_RPars.Absorbance. Control software can read absorbance from this variable in the same or

lower rate than absorbance is measured (CMD_WRITE_RATE). But absorbance data can be buffered in communication and block TOT18_Prot can process more samples in one clock. If recording of exact samples needs to be guaranteed, it is recommended to record absorbance sample directly in function TOY18_DecodeResponse (CMD_SUBS_ABS).

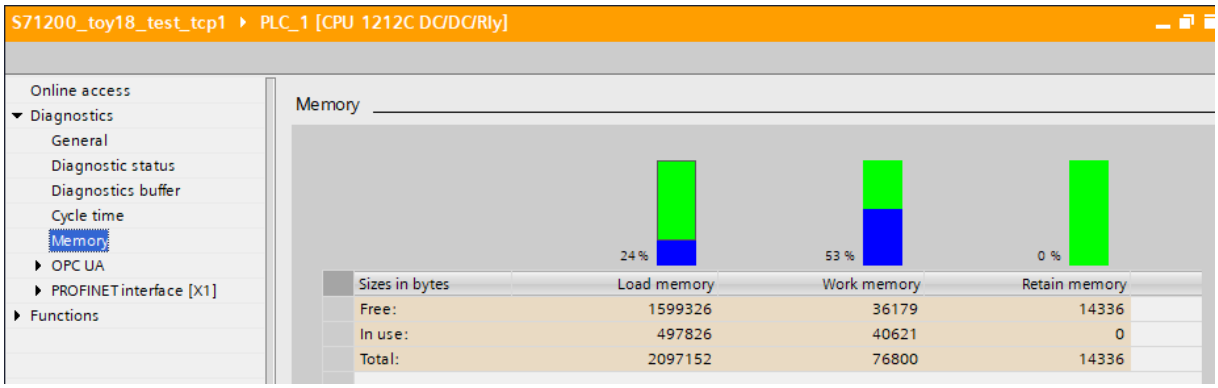
51	KeyLock	USInt	0	0
52	DisableRMT	Bool	false	FALSE
53	Absorbance	Array[0..3] of Real		
54	Absorbance[0]	Real	0.0	-4.84
55	Absorbance[1]	Real	0.0	-4.84
56	Absorbance[2]	Real	0.0	-4.84
57	Absorbance[3]	Real	0.0	-4.84
58	Abs_err	Array[0..3] of USInt		

Setting new wavelengths parameters can be done the same way (setting T18_DB.WPars.Wavelength and toggling T18_DB.bSendWL from FALSE to TRUE). Toggling T18_DB.bSendAutozero from FALSE to TRUE should make autozero and by parameter T18_DB.bSendClrErrors can be cleared errors and error sound.

It is possible to process detector commands manually when block TOY18_Control is disabled by T18_DB.bStart = FALSE. Then command needs to be set to T18_DB.ProtCtrl.nCmd (e.g. SAr = 16#534172) and toggling T18_DB.ProtCtrl.bCmd from FALSE to TRUE will do the job. Check T18_DB.RPars.Scan.

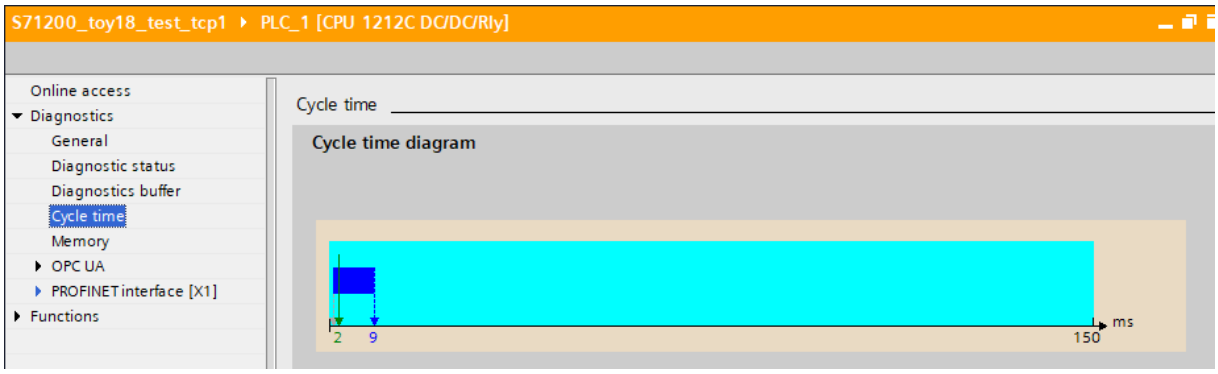
3	SerPars	"SER_PARS"		
4	bConnect	Bool	false	TRUE
5	bConnected	Bool	false	TRUE
6	COMCtrl	"COM_CTRL"		
7	PROTCtrl	"PROT_CTRL"		
8	bCmd	Bool	false	TRUE
9	nCmd	DWord	16#0	16#0053_4172
10	nTimeout	UDInt	1000	1000
11	bCmdDone	Bool	false	FALSE
12	bCmdBusy	Bool	false	FALSE
13	nCmdResult	UInt	0	0
14	bAsync	Bool	false	FALSE
15	nAsyncCmd	DWord	16#0	16#0041_5873
16	sendData	Array[0..127] of Byte		
17	recvData	Array[0..8191] of B...		
18	WPars	"TOY18_WPARS"		
19	RPars	"TOY18_RPARS"		
20	bStart	Bool	true	FALSE
21	bSendLamp	Bool	false	FALSE
22	bSendWL	Bool	false	FALSE
23	bSendAutozero	Bool	false	FALSE
24	bSendClrErrors	Bool	false	FALSE
25	bReady	Bool	false	FALSE
26	TextStatus	String	''	'Measure'

This project consumes aprox. 500 kB of Code memory (25 %) and aprox. 40 kB (53 %) of RAM memory for PLC S7-1200 1212C.



Note: RAM can be reduced by e.g. reducing of recvData buffer when commands CMD_READ_REPORT (upto 7kB), CMD_READ_SABS (upto 4kB) and CMD_READ_SINT (upto 3kB) are not used or by reducing of scan array in TOY18_SCAN when scanning commands (CMD_READ_SABS, CMD_READ_SINT) are not used.

Average PLC cycle time is 2-3 ms. The longest cycle time during general measurement (reading status and absorbance subscriptions) could take up to 10 ms.

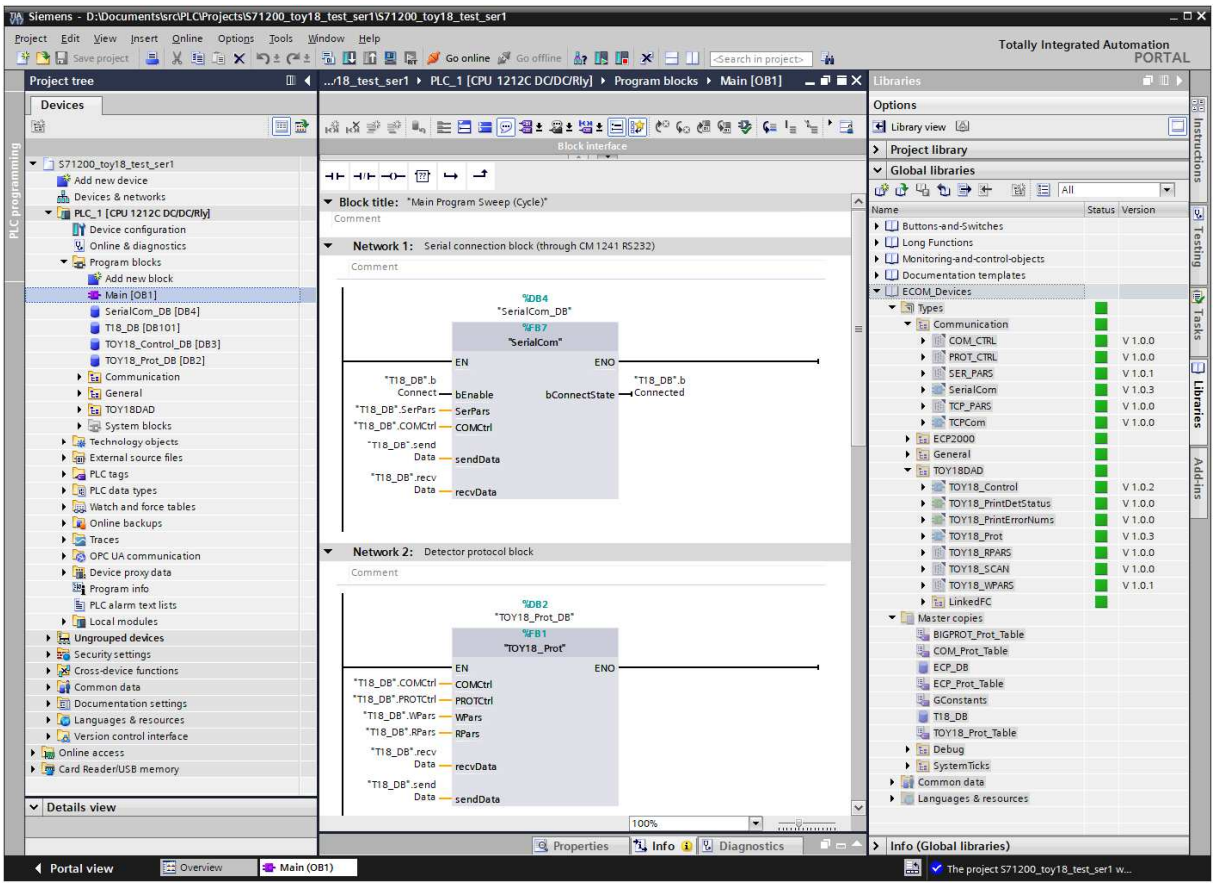


Note: Some commands (responses) can cause much longer cycle times. For example, response on command CMD_READ_SABS can prolong max. cycle time to aprox. 30 ms and response on command CMD_READ_REPORT up to aprox. 40 ms.

Project example 2 (S71200_toy18_test_ser1.ap17)

This project is almost the same as previous S71200_toy18_test_tcp1.ap17 with one difference. The block TCPCom was replaced by block SerialCom in Network 1 to realize serial communication with the TOY18DAD detector through CM 1241 module. This block needs T18_DB.SerPars parameters for selection and configuration of serial port. Such data type SER_PARS is defined in ECOM_Devices library.

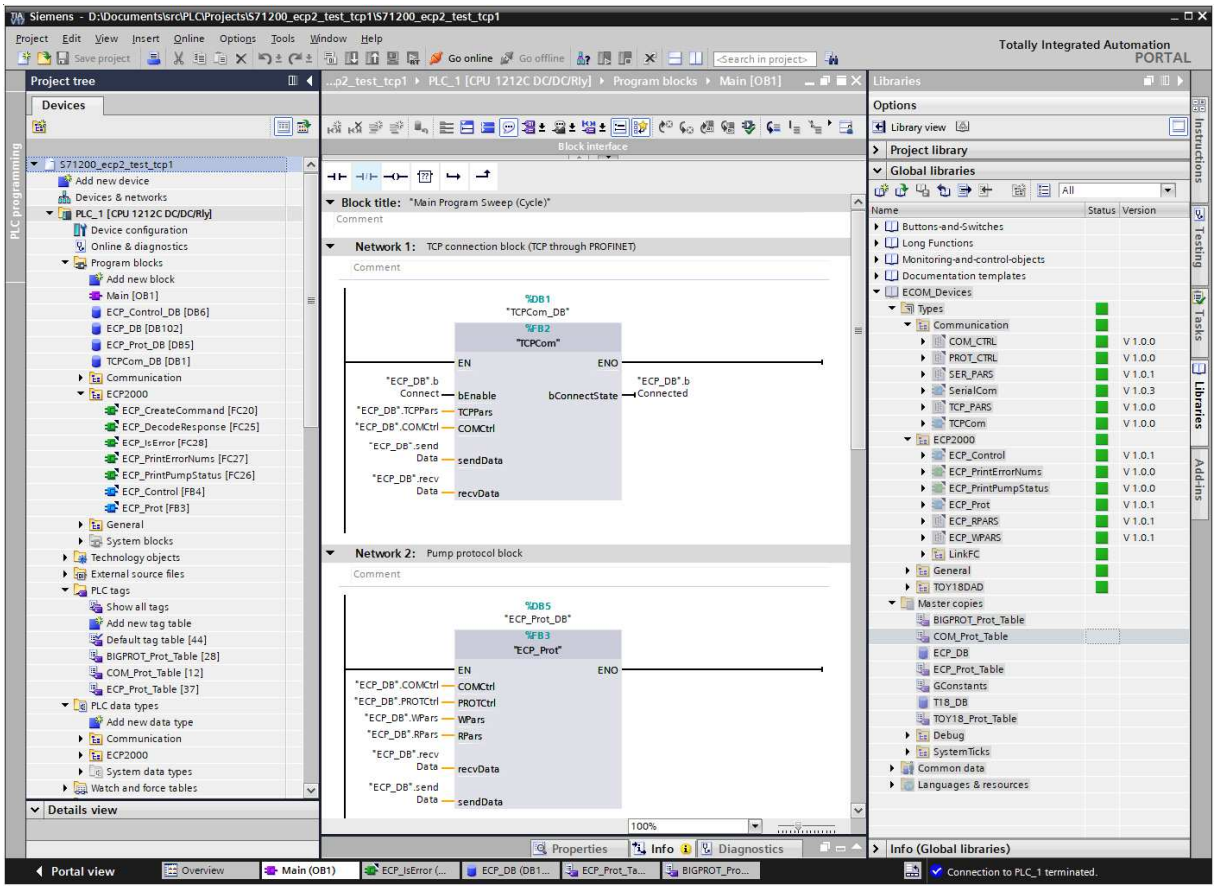
Warning: Serial communication through module CM 1241 doesn't work so well as TCP communication in Siemens PLC. Actually, I realized various bugs, strange behavior or slow performance when using Ptp Communication functions (Port_Config, Send_Config, Receive_Config, Receive_Reset, Send_P2P and Receive_P2P). Example project does work, but some operations can fail (changing serial settings (baudrate) fails first time; response on command CMD_READ_REPORT would be uncomplete – data loss).
It is recommended to use TCP communication (ERVIN7 module) if possible.



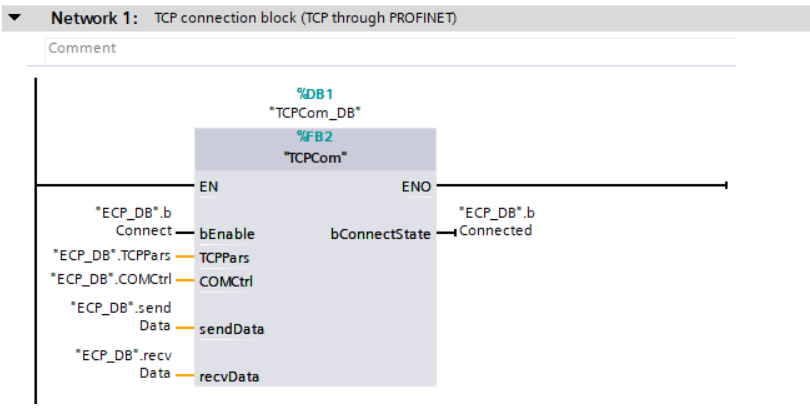
Project example 3 (S71200_ecp_test_tcp1.ap17)

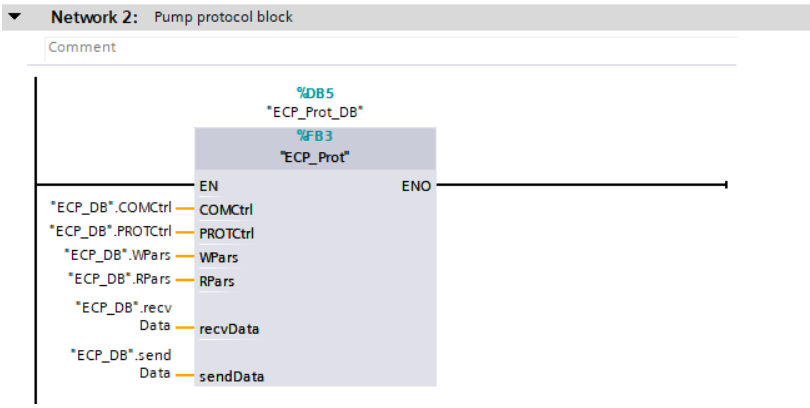
This project demonstrates communication with pump ECP2000 and concept is the same as previous project S71200_toy18_test_tcp1.ap17. Create new project and configure PLC. Add function blocks “TCPCom”, “ECP_Prot” and “ECP_Control”, tag tables “COM_Prot_Table”, “ECP_Prot_Table”, “BIGPROT_Prot_Table” and data block “ECP_DB” into project from ECOM_DEVICES library. Insert main function block into networks and define its parameters to match ECP_DB.

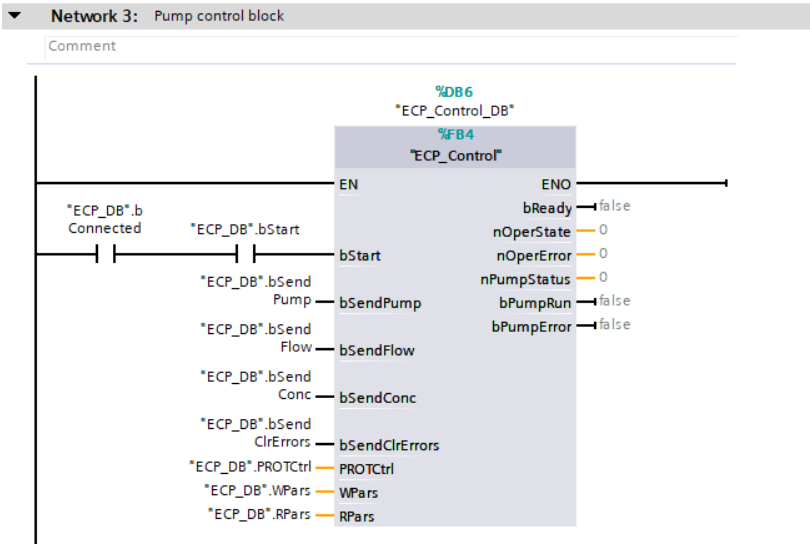
Project tree



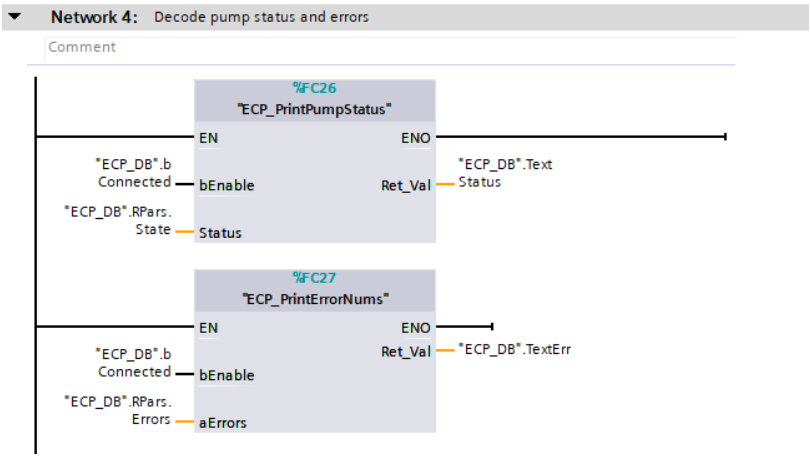
Main [OB1]







Network 4 demonstrates additional functions for decoding pump status and errors to strings.



ECP_DB [DB102]

Example of interconnection data block ECP_DB. Set default parameters to set adequate demands.

ECP_DB

	Name	Data type	Start value	Monito...	Retain	Acces...	Wri...	Visibl...	Setp...	Comment
1	Static									
2	TCPPars	"TCP_PARS"								
3	ID	CONN_OUC	1							Connection ID
4	RemoteAddress	IP_V4								Remote IP address
5	RemotePort	UInt	10001							Remote port
6	SerPars	"SER_PARS"								
7	bConnect	Bool	false							
8	bConnected	Bool	false							
9	COMCtrl	"COM_CTRL"								
10	PROTCtrl	"PROT_CTRL"								
11	sendData	Array[0..127] of Byte								
12	recvData	Array[0..8191] of Byte								
13	WPars	"ECP_WPARS"								
14	DoCmd	USInt	0							pump, stop, purge mode, ...
15	SetFlow	Real	0.0							ml/min
16	Concentration	Array[0..3] of Real								
17	SetPressure	Real	1.0							MPa
18	LowPressLimit	Real	0.0							MPa
19	HighPressLimit	Real	7.0							MPa
20	MaxPulsation	Real	1.0							MPa
21	PumpMode	USInt	0							0=flow
22	PressureUnit	USInt	0							0=MPa
23	CompMode	USInt	0							off
24	SolventUse	USInt	0							disabled
25	SolventLimit	Real	0.0							
26	AutoGrValveClosing	USInt	1							
27	Valves	USInt	0							
28	UsedSolvents	USInt	16#F							A, B, C, D
29	GradCycles	USInt	0							auto
30	ValveCount	USInt	4							
31	ValveMap	Array[0..3] of USInt								
32	RPars	"ECP_RPARS"								
33	bStart	Bool	true							
34	bSendPump	Bool	false							
35	bSendFlow	Bool	false							
36	bSendConc	Bool	false							
37	bSendClrErrors	Bool	false							
38	bReady	Bool	false							
39	TextStatus	String	''							
40	TextErr	String	''							

BIGPROT_Prot_Table

Tag table includes global BP constants, which are used inside function blocks and support functions. It needs to be part of a project!

BIGPROT_Prot_Table

	Name	Data type	Value	Comment
1	BP_CMD_GET_INFO	Word	16#0000	
2	BP_CMD_GET_USPEC	Word	16#0006	
3	BP_CMD_GET_BRDCOUNT	Word	16#0010	
4	BP_CMD_GET_BRDINFO	Word	16#0012	
5	BP_CMD_GET_ETHCFG	Word	16#0020	
6	BP_CMD_SET_ETHCFG	Word	16#0021	
7	BP_CMD_GET_ETHINFO	Word	16#0022	
8	BP_CMP_GET_LOGCOUNT	Word	16#0032	
9	BP_CMD_GET_LOGDATA	Word	16#0034	
10	BP_CMD_GET_CMPCOUNT	Word	16#0036	
11	BP_CMD_GET_CMPDATA	Word	16#0038	
12	BP_CMD_GET_REPORTCOUNT	Word	16#0040	
13	BP_CMD_GET_REPORTDATA	Word	16#0042	
14	BP_CMD_MAKE_BEEP	Word	16#0051	
15	BP_CMD_STOP_SOUND	Word	16#0053	
16	BP_CMDERR_NONE	UInt	0	no error
17	BP_CMDERR_UNKNOWN	UInt	1	unknown command, invalid pattern
18	BP_CMDERR_BUSY	UInt	2	unit busy
19	BP_CMDERR_SIZE	UInt	3	invalid length or format (reception)
20	BP_CMDERR_VALUE	UInt	4	invalid value range
21	BP_CMDERR_MODE	UInt	5	invalid mode
22	BP_CMDERR_PENDING	UInt	6	pending
23	BP_CMDERR_FAILED	UInt	7	operation failed
24	BP_CMDERR_BLOCKED	UInt	8	operation blocked
25	BP_CMDERR_UNCOMPLETE	UInt	10	uncomplete command (reception)
26	BP_CMDERR_CRC	UInt	11	invalid crc (reception)
27	BP_CMDERR_IGNORE	UInt	12	command should be ignored (creation)
28	BP_CMDERR_TIMEOUT	UInt	13	no response in time (reception)
29	<Add new>			

ECP_Prot_Table

Tag table includes global ECP2000 specific constants, which are used inside function blocks and support functions. It needs to be part of a project!

ECP_Prot_Table

	Name	Data type	Value	Comment
1	ECP_CMD_GET_STATUS	Word	16#0120	
2	ECP_CMD_DO_COMMAND	Word	16#0123	
3	ECP_CMD_GET_DESC	Word	16#0124	
4	ECP_CMD_GET_GRADCFG	Word	16#0162	
5	ECP_CMD_SET_GRADCFG	Word	16#0163	
6	ECP_CMD_GET_CFGA	Word	16#0166	Unsupported
7	ECP_CMD_SET_CFGA	Word	16#0167	Unsupported
8	ECP_CMD_GET_PUMPCFG2	Word	16#0168	
9	ECP_CMD_SET_PUMPCFG2	Word	16#0169	
10	ECP_CMD_GET_IFCCFG	Word	16#0170	Unsupported
11	ECP_CMD_SET_IFCCFG	Word	16#0171	Unsupported
12	ECP_CMD_GET_VALVEMAP	Word	16#0172	
13	ECP_CMD_SET_VALVEMAP	Word	16#0173	
14	ECP_CMD_GET_DFLOW	Word	16#0180	Read desired flow
15	ECP_CMD_SET_DFLOW	Word	16#0181	Write desired flow
16	ECP_CMD_GET_DPRESS	Word	16#0182	Read desired pressure
17	ECP_CMD_SET_DPRESS	Word	16#0183	Write desired pressure
18	ECP_CMD_GET_PRESSLIMITS	Word	16#0184	
19	ECP_CMD_SET_PRESSLIMITS	Word	16#0185	
20	ECP_CMD_GET_VALVE	Word	16#0186	
21	ECP_CMD_SET_VALVE	Word	16#0187	
22	ECP_CMD_GET_CONC	Word	16#0188	
23	ECP_CMD_SET_CONC	Word	16#0189	
24	ECP_CMD_GET_PRES	Word	16#01A0	
25	ECP_STATUS_INIT	USInt	0	
26	ECP_STATUS_PREP	USInt	1	
27	ECP_STATUS_STOPPED	USInt	2	
28	ECP_STATUS_RUN	USInt	3	
29	ECP_DOCMD_NONE	USInt	0	
30	ECP_DOCMD_PUMP	USInt	1	
31	ECP_DOCMD_STOP	USInt	2	
32	ECP_DOCMD_PUMPSAFE	USInt	8	
33	ECP_DOCMD_PURGEMODEON	USInt	101	
34	ECP_DOCMD_PURGEMODEOFF	USInt	102	
35	ECP_DOCMD_CLOSEVALVES	USInt	111	
36	ECP_DOCMD_QUITPROG	USInt	112	
37	ECP_DOCMD_OPENVALIDVALVE	USInt	113	
38	<Add new>			

Project functionality

Parameter ECP_DB.TCPPars.RemoteAddress needs to be set correctly to match real pump. Connection is started by toggling parameter ECP_DB.bConnect from FALSE to TRUE. Successful connection is signaled by TRUE parameter ECP_DB.bConnected.

It is recommended to set all parameters available in ECP_DB.WPars to demanded default values. Communication can be started by settings parameter ECP_DB.bStart from FALSE to TRUE (default state after connection). If communication is successful the most of ECP_DB.WPars parameters wound be written and the most of parameters in ECP_DB.RPars would be read and status variables would continue to be updated. If communication isn't running, check last protocol command ECP_DB.PROTCtrl.nCmd and its result ECP_DB.PROTCtrl.nCmdResult.

Simple pump control can be realized through ECP_Control block inputs. Pump would run by changing parameter ECP_DB.bSendPump from FALSE to TRUE and it would stop by changing the same parameter from TRUE to FALSE (this behavior influences parameter ECP_DB.WPars.DoCmd automatically). Setting flow can be done by modification parameter ECP_DB.WPars.SetFlow and changing ECP_DB.bSendFlow from FALSE to TRUE. Similar way can be used for concentration on gradient pump (set ECP_DB.WPars.Concentration array to be 100% in sum and change ECP_DB.bSendConc from FALSE to TRUE). Change of parameter ECP_DB.bSendClrErrors from FALSE to TRUE will send stop error sound command. Pump state, errors, warnings, pressure and other parameters are read continually in aprox. period 200 ms. Pressure variable is located at ECP_DB.RPars.ActualPressure.

11	RPars	ECP_RPARS		
12	bStart	Bool	true	TRUE
13	bSendPump	Bool	false	TRUE
14	bSendFlow	Bool	false	FALSE
15	bSendConc	Bool	false	FALSE
16	bSendClrErrors	Bool	false	FALSE
17	bReady	Bool	false	FALSE
18	TextStatus	String	''	'Running'
19	TextErr	String	''	'None'

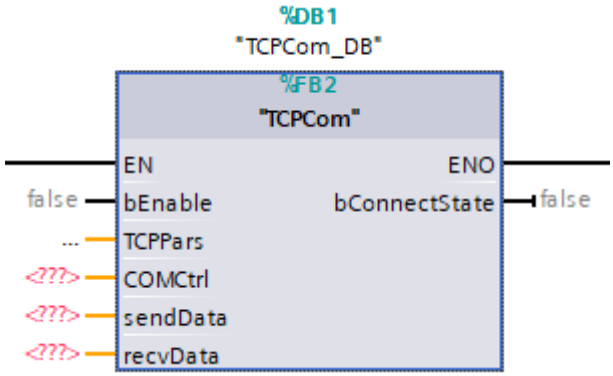
10	WPars	"ECP_WPARS"		
11	DoCmd	USInt	0	8
12	SetFlow	Real	0.0	1.2

30	LowPressureLimit	Real	0.0	0.0
31	HighPressureLimit	Real	0.0	60.0
32	ActualFlow	Real	0.0	1.2
33	ActualPressure	Real	0.0	0.0

Function block ECP_Control can be modified for customer's needs. It operates sending of commands through ECP_Prot block. All used blocks and tables are described later in more detail.

Overview of main function blocks

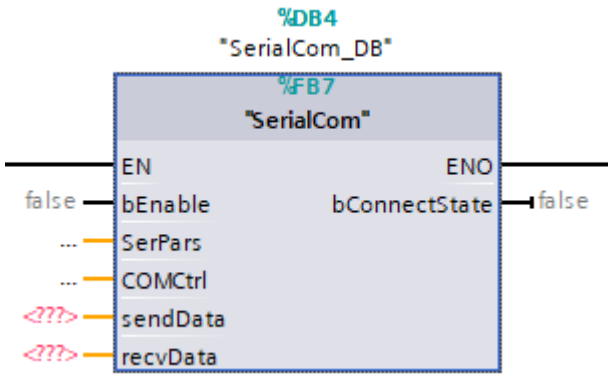
TCPCom



TCPCom block implements connection, disconnection, send and read operations using TCP communication. Connection/disconnection is controlled through bEnable input. Connection is realized according to TCPPars parameters and block tries to connect or hold connection even connection failed at once. Structure COMCtrl manages block send/read operations.

Parameter	*Direction*	*Data type*	*Default*	*Description*
bEnable	IN	Bool	False	Stores TCP parameters and starts connection upon a positive edge. Block tries to connect periodically when connection fails (partner not available). Starts disconnection upon a negative edge.
TCPPars	IN	TCP_PARS	---	Connection TCP parameters as ID, remote IP address and remote port.
COMCtrl	INOUT	COM_CTRL	---	Structure for control of sending (bSend, nWLen, bSent) and reading data (nRSize, nRIdx, bNdr, nRLen) from other control block. Block state and error (wState, wResult) can be read.
sendData	INOUT	Variant	---	Expects array of bytes for data to be sent. Size should be enough for the longest command.
recvData	INOUT	Variant	---	Expect array of bytes for read data. Size (nRSize) should be enough for the longest response.
bConnectState	OUT	Bool	False	Signaling connection state.

SerialCom

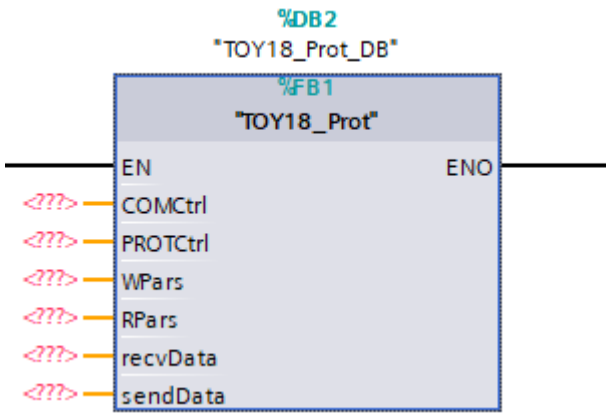


SerialCom block implements connection, disconnection, send and read operations using serial RS232 communication. Connection (configuration) is controlled through bEnable input. Structure COMCtrl manages block send/read operations.

Data block SerialCom_DB has to be not optimized!

Parameter	***Direction***	***Data type***	***Default***	***Description***
bEnable	IN	Bool	False	Stores serial parameters and starts connection (configuration) upon a positive edge. Starts disconnection (nothing) upon a negative edge.
SerPars	IN	SER_PARS	---	Connection serial parameters as port, baudrate selection, parity selection, databits selection and stopbits selection.
COMCtrl	INOUT	COM_CTRL	---	Structure for control of sending (bSend, nWLen, bSent) and reading data (nRSize, nRIdx, bNdr, nRLen) from other control block. Block state and error (wState, wResult) can be read.
sendData	INOUT	Variant	---	Expects array of bytes for data to be sent. Size should be enough for the longest command.
recvData	INOUT	Variant	---	Expect array of bytes for read data. Size (nRSize) should be enough for the longest response.
bConnectState	OUT	Bool	False	Signaling connection state.

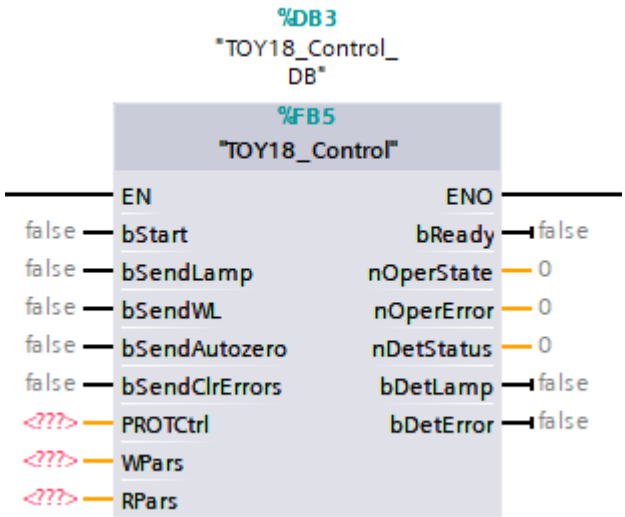
TOY18_Prot



TOY18_Prot block implements TOY18 detector protocol. Block was designed to operate together with communication block TCPCom or SerialCom.
Commands are created into sendData buffer according to WPars parameters. Responses are read into recvData buffer and decoded to RPars variables. Block is controlled through structure PROTCtrl.

Parameter	*Direction*	*Data type*	*Default*	*Description*
COMCtrl	INOUT	COM_CTRL	---	Structure for controlling communication block TCPCom or SerialCom.
PROTCtrl	INOUT		---	Structure for controlling of this block. Parameter nCmd defines code for demanded command (see constants TOY18_Prot_Table), which is created and sent upon a positive edge of bCmd. Output bCmdBusy characterized command processing and outputs bCmdDone and nCmdResult show command result (ok, error). Parameters bAsync and nAsyncCmd can signal asynchronous data reception.
WPars	INOUT	TOY18_WPARS	---	Structure includes demanded setup parameters for detector TOY18 like wavelengths, rate, time constant, etc. These parameters are used during write commands mainly (e.g. CMD_WRITE_WL).
RPars	INOUT	TOY18_RPARS	---	Structure includes actual setup parameters and measured data for detector TOY18 like wavelengths, rate, time constant, absorbances, intensities, lamp time, etc. These parameters are used during both write and read commands often (e.g. CMD_WRITE_WL, CMD_READ_WL).
sendData	INOUT	Array[*] of Byte	---	Expects array of bytes for data to be sent. Size should be enough for the longest command.
recvData	INOUT	Array[*] of Byte	---	Expect array of bytes for read data. Size should be enough for the longest response.

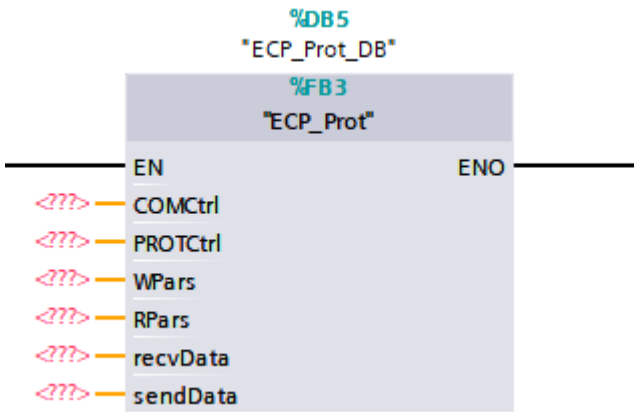
TOY18_Control



TOY18_Control block implements basic control of detector TOY18. Block was designed to operate together with protocol block TOY18_Prot.
Main functions are: read unit identification, setup main parameters, starts and holds data subscriptions (status, absorbance) according to lamp state, set lamp, set wavelengths, set autozero and clear errors.
Both block and detector states can be monitored through block outputs.

Parameter	***Direction***	***Data type***	***Default***	***Description***
bStart	IN	Bool	False	Starts unit setup (read identification, model, wl. range, write wavelengths, rate, tc, bandwidth, …) upon a positive edge. Then it holds subscriptions to read status and measured data. Stops all subscriptions upon a negative edge.
bSendLamp	IN	Bool	False	Starts lamp write command. See parameter WPars.Lamp.
bSendWL	IN	Bool	False	Starts wavelength write command. See array parameter WPars.Wavelength.
bSendAutozero	IN	Bool	False	Starts autozero write command.
bSendClrErrors	IN	Bool	False	Starts clear errors write command.
PROTCtrl	INOUT	COM_CTRL	---	Structure for controlling protocol block TOY18_Prot.
WPars	INOUT	TOY18_WPARS	---	Structure for write parameters.
RPars	INOUT	TOY18_RPARS	---	Structure for read parameters.
bReady	OUT	Bool	False	Signaling block ready state for sending commands (bSendLamp, …).
nOperState	OUT	UInt	0	Signaling block internal operation state (0 – idle, 1 – init, 2 – read id, 3 – set parameters, 4 – ready, 5 – error, 6 – deinit).
nOperError	OUT	UInt	0	Signaling block internal error when nOperState=5 (ERR_CMD_x).
nDetStatus	OUT	Uint	0	Detector status. See TOY18_Prot_Table (DETSTATE_x). The same as RPars.State.
bDetLamp	OUT	Bool	False	Signaling detector active lamp state. See RPars.D2LampState.
bDetError	OUT	Bool	False	Signaling any detector error. See RPars.Errors for details.

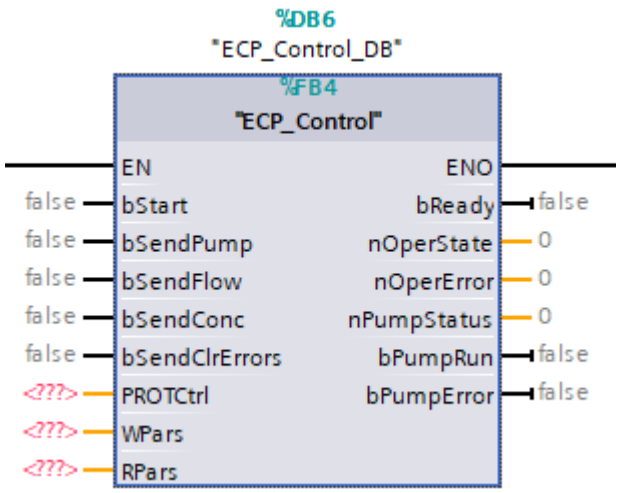
ECP_Prot



ECP_Prot block implements ECP2000 pump protocol. Block was designed to operate together with communication block TCPCom or SerialCom. Commands are created into sendData buffer according to WPars parameters. Responses are read into recvData buffer and decoded to RPars variables. Block is controlled through structure PROTCtrl.

Parameter	*Direction*	*Data type*	*Default*	*Description*
COMCtrl	INOUT	COM_CTRL	---	Structure for controlling communication block TCPCom or SerialCom.
PROTCtrl	INOUT		---	Structure for controlling of this block. Parameter nCmd defines code for demanded command (see constants BIGPROT_Prot_Table, ECP_Prot_Table), which is created and sent upon a positive edge of bCmd. Output bCmdBusy characterized command processing and outputs bCmdDone and nCmdResult show command result (ok, error). Parameters bAsync and nAsyncCmd can signal asynchronous data reception.
WPars	INOUT	ECP_WPARS	---	Structure includes demanded setup parameters for pump ECP2000 like flowrate, concentration and other configuration. These parameters are used during write commands mainly (e.g. ECP_CMD_SET_DFLOW).
RPars	INOUT	ECP_RPARS	---	Structure includes actual setup parameters and measured data for pump ECP2000 like set flow, actual flow, actual pressure, status, errors, etc. These parameters are used during read commands mainly (e.g. ECP_CMD_GET_DFLOW).
sendData	INOUT	Array[*] of Byte	---	Expects array of bytes for data to be sent. Size should be enough for the longest command.
recvData	INOUT	Array[*] of Byte	---	Expect array of bytes for read data. Size should be enough for the longest response.

ECP_Control



ECP_Control block implements basic control of pump ECP2000. Block was designed to operate together with protocol block ECP_Prot.
Main functions are: read unit identification, setup main parameters, read status and pressure periodically, set pump, set flow, set concentration and clear errors (stop sound).
Both block and pump states can be monitored through block outputs.

Parameter	***Direction***	***Data type***	***Default***	***Description***
bStart	IN	Bool	False	Starts unit setup (read identification, model, description, write flow, concentration, pres. limits, …) upon a positive edge. Then it holds periodic reading of status and pressure data. Stops flow and all communication on a negative edge.
bSendPump	IN	Bool	False	Starts pump on positive edge and stops it on negative edge. Operation modify parameter WPars.DoCmd automatically.
bSendFlow	IN	Bool	False	Starts flow write command. See parameter WPars.SetFlow.
bSendConc	IN	Bool	False	Starts concentration write command. See array parameter WPars.Concentration.
bSendClrErrors	IN	Bool	False	Starts stop error sound write command.
PROTCtrl	INOUT	COM_CTRL	---	Structure for controlling protocol block ECP_Prot.
WPars	INOUT	ECP_WPARS	---	Structure for write parameters.
RPars	INOUT	ECP_RPARS	---	Structure for read parameters.
bReady	OUT	Bool	False	Signaling block ready state for sending commands (bSendFlow, …).
nOperState	OUT	UInt	0	Signaling block internal operation states (see TOY18_Control).
nOperError	OUT	UInt	0	Signaling block internal error when nOperState=5 (BP_CMDERR_x).
nPumpStatus	OUT	Uint	0	Pump status. See ECP_Prot_Table (ECP_STATUS_x). The same as RPars.State.
bPumpRun	OUT	Bool	False	Signaling pump active pumping state.
bPumpError	OUT	Bool	False	Signaling any pump error. See RPars.Errors for details.

Overview of types

TCP_PARS

Name	*Data type*	*Default*	*Description*
ID	CONN_OUC	1	
RemoteAddress	IP_V4	192.168.91.24	Array[1..4] of Byte
RemotePort	UInt	10001	

SER_PARS

Name	*Data type*	*Default*	*Description*
Port	UInt	0	
BaudIdx	UInt	11	11=115200 bd
ParityIdx	UInt	1	1=None
DatabitsIdx	UInt	1	1=8
StopbitsIdx	UInt	1	1=1

COM_CTRL

Name	*Data type*	*Default*	*Description*
bSend	Bool	False	
nWLen	UInt	0	
nRSize	UInt	100	
nRIdx	UInt	0	
bSent	Bool	False	
bNdr	Bool	False	
nRLen	UInt	0	
wState	Word	16#0	
wResult	Word	16#0	

PROT_CTRL

Name	*Data type*	*Default*	*Description*
bCmd	Bool	False	
nCmd	UInt	0	
nTimeout	UInt	100	
bCmdDone	Bool	False	
bCmdBusy	Bool	False	
nCmdResult	UInt	0	
bAsync	UInt	0	
nAsyncCmd	UInt	0	

TOY18_SCAN

Name	*Data type*	*Default*	*Description*
nType	USInt	0	
nCnt	UInt	0	
rFrom	Real	0.0	

rStep	Real	0.0	
rData	Array[0..699] of Real	---	

TOY18_WPARS

Name	*Data type*	*Default*	*Description*
Lamp	Bool	False	
Wavelength	Array[0..3] of UInt	[254, 254, 254, 254]	
Rate	USInt	10	
TimeConstant	USInt	2	
BandWidth	USInt	8	
Beep	USInt	1	
MathDFunction	USInt	0	
NegRange	USInt	0	
SoundMask	USInt	16#1E	
LeakMode	USInt	0	
KeyLock	USInt	0	
DisableRMT	Bool	False	
IOState	USInt	0	
Section	USInt	0	
Subs + st + abs + its + qsa	Struct USInt USInt USInt USInt	 1 0 0 0	
MinScanWL	UInt	200	
MaxScanWL	UInt	400	

TOY18_RPARS

Name	*Data type*	*Default*	*Description*
Type	UInt	0	
Channels	UInt	4	
VarFix	Bool	False	
VarScan	Bool	False	
MinWL	UInt	0	
MaxWL	UInt	0	
Name	String	‘’	
OEMName	String	‘’	
Model	String	‘’	
SN	String	‘’	
HW	String	‘’	
FW	UInt	0	
State	USInt	0	
PrevState	USInt	0	
IOState	USInt	0	
Errors	Array[0..7] of Byte	---	

Errors_old	Array[0..3] of Byte	---	
D2LampState	USInt	0	
WLampState	USInt	0	
D2LampHours	Real	0.0	
WLampHours	Real	0.0	
DevHours	Real	0.0	
Wavelength	Array[0..3] of UInt	[254, 254, 254, 254]	
Rate	USInt	10	
TimeConstant	USInt	2	
BandWidth	USInt	8	
Beep	USInt	1	
MathDFunction	USInt	0	
NegRange	USInt	0	
SoundMask	USInt	16#1E	
LeakMode	USInt	0	
KeyLock	USInt	0	
DisableRMT	Bool	False	
Absorbance	Array[0..3] of Real	---	
Abs_err	Array[0..3] of USInt	---	
Intensity	Array[0..3] of Real	---	
Int_err	Array[0..3] of USInt	---	
Section	USInt	0	
Subs + st + abs + its + qsa	Struct USInt USInt USInt USInt	 1 0 0 0	
MinScanWL	UInt	200	
MaxScanWL	UInt	400	
Scan	TOY18_SCAN	---	
IsError	Bool	False	
Valid_st	Bool	False	
Cnt_st	UDInt	0	
Cnt_abs	UDInt	0	

ECP_WPARS

Name	***Data type***	***Default***	***Description***
DoCmd	USInt	0	
SetFlow	Real	0.0	
Concentration	Array[0..3] of Real	[100.0, 0.0, 0.0, 0.0]	
SetPressure	Real	1.0	
LowPressLimit	Real	0.0	

HighPressLimit	Real	7.0	
MaxPulsation	Real	1.0	
PumpMode	USInt	0	
PressureUnit	USInt	0	
CompMode	USInt	0	
SolventUse	USInt	0	
SolventLimit	Real	0.0	
AutoGrValveClosing	USInt	1	
Valves	USInt	0	
UsedSolvents	USInt	16#F	
GradCycles	USInt	0	
ValveCount	USInt	4	
ValveMap	Array[0..3] of USInt	[0, 1, 2, 3]	

ECP_RPARS

Name	***Data type***	***Default***	***Description***
Type	UInt	0	
Name	String	‘’	
OEMName	String	‘’	
Model	String	‘’	
SN	String	‘’	
FW	UInt	0	
Desc	Struct	see below!	
.MaxFlow	Real	0.0	
.MinPurgeFlow	Real	0.0	
.MaxPurgeFlow	Real	0.0	
.MaxPressure	Real	0.0	
.MinPurgePressure	Real	0.0	
.MaxPurgePressure	Real	0.0	
.MinCompPressLimit	Real	0.0	
.MaxCompPressLimit	Real	0.0	
State	USInt	0	
PurgeMode	Bool	0	
DIn	USInt	0	
DOut	USInt	0	
AIn	Real	0.0	
Program	USInt	0	
Errors	Array[0..7] of Byte	---	
Warnings	Array[0..7] of Byte	---	
SetFlow	Real	0.0	
Concentration	Array[0..3] of Real	[0.0, 0.0, 0.0, 0.0]	
SetPressure	Real	0.0	
LowPressureLimit	Real	0.0	
HighPressureLimit	Real	0.0	
ActualFlow	Real	0.0	

ActualPressure	Real	0.0	
MaxPulsation	Real	0.0	
PumpMode	USInt	0	
PressureUnit	USInt	0	
CompMode	USInt	0	
SolventUse	USInt	0	
SolventLimit	Real	0.0	
AutoGrValveClosing	USInt	0	
Valves	USInt	0	
IsGradient	Bool	false	
UsedSolvents	USInt	0	
GradCycles	USInt	0	
ValveCount	USInt	0	
ValveMap	Array[0..3] of USInt	[0, 0, 0, 0]	
IsError	Bool	false	
IsWarning	Bool	false	
Valid_st	Bool	false	

List of TOY18 constants (TOY18_Prot_Table)

Name	*Data Type*	*Value*	*Comment*
DETTYPE_UNKNOWN	USInt	0	
DETTYPE_TOY18	USInt	1	
DETTYPE_TOY20	USInt	2	
DETTYPE_BABY	USInt	3	
DETTYPE_TOY	USInt	4	
LAMPSTATE_OFF	USInt	0	
LAMPSTATE_ON	USInt	1	
LAMPSTATE_ERR	USInt	2	
LAMPSTATE_ERRI	USInt	3	
LAMPSTATE_ERRV	USInt	4	
SAMPLE_RATE_01HZ	USInt	1	
SAMPLE_RATE_02HZ	USInt	2	
SAMPLE_RATE_05HZ	USInt	5	
SAMPLE_RATE_10HZ	USInt	10	
SAMPLE_RATE_20HZ	USInt	20	
TIMECONST_500MS	USInt	0	
TIMECONST_750MS	USInt	1	
TIMECONST_1S	USInt	2	
TIMECONST_2S	USInt	3	
TIMECONST_4S	USInt	4	
TIMECONST_8S	USInt	5	
TIMECONST_16S	USInt	6	
TIMECONST_200MS	USInt	7	
TIMECONST_100MS	USInt	8	
MATHDFCE_OFF	USInt	0	D
MATHDFCE_SUM_AB	USInt	1	D=A+B
MATHDFCE_SUB_AB	USInt	2	D=A-B
MATHDFCE_AVER_AB	USInt	3	D=(A+B)/2
MATHDFCE_SUB_AB2	USInt	4	D(A-B)/2
MATHDFCE_MAX_AB	USInt	5	D=MAX(A,B)
MATHDFCE_DIV_AB	USInt	6	D=A/B
MATHDFCE_NEG_A	USInt	7	D=-A
MATHDFCE_ABS_A	USInt	8	D=|A|
MATHDFCE_LIN_A	USInt	9	D=A*M+O (from v1.33)
SOUNDMASK_OFF	USInt	16#00	
SOUNDMASK_KEYBOARD	USInt	16#01	
SOUNDMASK_ERRORS	USInt	16#02	
SOUNDMASK_WARNINGS	USInt	16#04	
SOUNDMASK_OPERATIONS	USInt	16#08	
SOUNDMASK_STATECHANGE	USInt	16#10	
SOUNDMASK_ERR_WARN_OPER	USInt	16#0E	Errors+Warnings+Operations
NEGRNG_100MAU	USInt	0	
NEGRNG_300MAU	USInt	1	
NEGRNG_700MAU	USInt	2	
LEAKMODE_OFF	USInt	0	
LEAKMODE_STATE	USInt	1	
LEAKMODE_WARNING	USInt	2	
LEAKMODE_ERROR	USInt	3	
KEYLOCK_LOCK	USInt	0	
KEYLOCK_ALLOW_ZERO	USInt	1	
KEYLOCK_UNLOCK	USInt	2	
DETSTATE_IDLE	USInt	0	

Name	*Data Type*	*Value*	*Comment*
DETSTATE_INIT	USInt	1	
DETSTATE_MEAS	USInt	2	
DETSTATE_ZERO	USInt	3	
DETSTATE_SABS	USInt	4	
DETSTATE_SINT	USInt	5	
DETSTATE_UMOD	USInt	6	
DETSTATE_SELF	USInt	7	
DETSTATE_FMOD	USInt	8	
DETSTATE_ASCA	USInt	9	
CMD_READ_NAME	DWord	16#445472	DTr
CMD_READ_OEM	DWord	16#444F72	DOr
CMD_READ_HW	DWord	16#485772	HWr
CMD_READ_MODEL	DWord	16#4D4472	MDr
CMD_READ_FW	DWord	16#535772	SWr
CMD_READ_SN	DWord	16#534E72	SNr
CMD_READ_LAMPAGE	DWord	16#4C4C72	LLr
CMD_READ_STATUS	DWord	16#535972	SYr
CMD_READ_STATUS_OLD	DWord	16#535472	STr
CMD_READ_STATUS_OLDX	DWord	16#535872	SXr
CMD_READ_DISRMT	DWord	16#444872	DHr
CMD_WRITE_DISRMT	DWord	16#444877	DHw
CMD_WRITE_HOLD	DWord	16#484377	HCw
CMD_WRITE_CLRERRS	DWord	16#424577	BEw
CMD_READ_LAMP	DWord	16#4C5072	LPr
CMD_WRITE_LAMP	DWord	16#4C5077	LPw
CMD_WRITE_AUTOZERO	DWord	16#5A5277	ZRw
CMD_WRITE_BEEP	DWord	16#534777	SGw
CMD_READ_IO	DWord	16#465372	FSr
CMD_WRITE_IO	DWord	16#465377	FSw
CMD_READ_WLRANGE	DWord	16#575272	WRr
CMD_READ_WL	DWord	16#574C72	WLr
CMD_WRITE_WL	DWord	16#574C77	WLw
CMD_READ_WLSCAN	DWord	16#575372	WSr
CMD_WRITE_WLSCAN	DWord	16#575377	WSw
CMD_READ_BW	DWord	16#425772	BWr
CMD_WRITE_BW	DWord	16#425777	BWw
CMD_READ_RATE	DWord	16#534672	SFr
CMD_WRITE_RATE	DWord	16#534677	SFw
CMD_READ_TC	DWord	16#544372	TCr
CMD_WRITE_TC	DWord	16#544377	TCw
CMD_READ_MATHDFCE	DWord	16#4F4472	ODr
CMD_WRITE_MATHDFCE	DWord	16#4F4477	ODw
CMD_READ_SOUND	DWord	16#424D72	BMr
CMD_WRITE_SOUND	DWord	16#424D77	BMw
CMD_READ_NEGRNG	DWord	16#4E5272	NRr
CMD_WRITE_NEGRNG	DWord	16#4E5277	NRw
CMD_READ_LEAKMODE	DWord	16#4C4D72	LMr
CMD_WRITE_LEAKMODE	DWord	16#4C4D77	LMw
CMD_READ_RMTLOCK	DWord	16#524B72	RKr
CMD_WRITE_RMTLOCK	DWord	16#524B77	RKw
CMD_READ_ABS	DWord	16#414272	ABr
CMD_READ_ABSX	DWord	16#415872	AXr
CMD_READ_INT	DWord	16#495472	ITr

Name	*Data Type*	*Value*	*Comment*
CMD_READ_INTX	DWord	16#495872	IXr
CMD_READ_SABS	DWord	16#534172	SAr
CMD_READ_SINT	DWord	16#534972	SIr
CMD_READ_SUBS	DWord	16#4D5372	MSr
CMD_WRITE_SUBS	DWord	16#4D5377	MSw
CMD_SUBS_STATUS	DWord	16#535973	SYs
CMD_SUBS_ABS	DWord	16#414273	ABs
CMD_WRITE_ABU	DWord	16#414275	ABu
CMD_SUBS_ABSX	DWord	16#415873	AXs
CMD_WRITE_ABUX	DWord	16#415875	AXu
CMD_SUBS_INT	DWord	16#495473	ITs
CMD_WRITE_ITU	DWord	16#495475	ITu
CMD_SUBS_INTX	DWord	16#495873	IXs
CMD_WRITE_ITUX	DWord	16#495875	IXu
CMD_READ_REPORT	DWord	16#435272	CRr (?)
CMD_READ_XTEST	DWord	16#585872	unsupported (testing ERR1)
CMD_READ_YTEST	DWord	16#585972	unsupported (testing timeout)
ERRCMD_NONE	UInt	0	no error
ERRCMD_UNKNOWN	UInt	1	unknown command, invalid pattern
ERRCMD_VALUE	UInt	2	invalid value range
ERRCMD_BUSY	UInt	3	unit busy
ERRCMD_MODE	UInt	4	invalid mode
ERRCMD_FAILED	UInt	5	operation failed
ERRCMD_BLOCKED	UInt	6	operation blocked
ERRCMD_LENGTH	UInt	10	invalid length or format (reception)
ERRCMD_CRC	UInt	11	invalid checksum (reception)
ERRCMD_IGNORE	UInt	12	command should be ignored (creation)
ERRCMD_TIMEOUT	UInt	13	no response in time (reception)
SCANTYPE_ABS	USInt	0	Scan ABS
SCANTYPE_INT	USInt	1	Scan INT
SCANTYPE_OTHER	USInt	2	Scan other (CCD)

Note: Constant values for commands are created from 3 first command/response letters (e.g. DTr = 16#445472). See www.asciitable.com for details.

List of COM constants (COM_Prot_Table)

Name	*Data Type*	*Value*	*Comment*
BAUDIDX_9600	UInt	6	Index for baudrate 9600bd
BAUDIDX_19200	UInt	7	Index for baudrate 19200bd
BAUDIDX_38400	UInt	8	Index for baudrate 38400bd
BAUDIDX_57600	UInt	9	Index for baudrate 57600bd
BAUDIDX_115200	UInt	11	Index for baudrate 115200bd
PARITYIDX_NONE	UInt	1	Index for no parity
PARITYIDX_EVEN	UInt	2	Index for even parity
PARITYIDX_ODD	UInt	3	Index for odd parity
DATABITSIDX_8	UInt	1	Index for 8 databits
DATABITSIDX_7	UInt	2	Index for 7 databits
STOPBITSIDX_1	UInt	1	Index for 1 stopbit
STOPBITSIDX_2	UInt	2	Index for 2 stopbits

List of BIGProtocol constants (BIGPROT_Prot_Table)

Name	*Data Type*	*Value*	*Comment*
BP_CMD_GET_INFO	Word	16#0000	
BP_CMD_GET_USPEC	Word	16#0006	
BP_CMD_GET_BRDCOUNT	Word	16#0010	
BP_CMD_GET_BRDINFO	Word	16#0012	
BP_CMD_GET_ETHCFG	Word	16#0020	
BP_CMD_SET_ETHCFG	Word	16#0021	
BP_CMD_GET_ETHINFO	Word	16#0022	
BP_CMP_GET_LOGCOUNT	Word	16#0032	
BP_CMD_GET_LOGDATA	Word	16#0034	
BP_CMD_GET_CMPCOUNT	Word	16#0036	
BP_CMD_GET_CMPDATA	Word	16#0038	
BP_CMD_GET_REPORTCOUNT	Word	16#0040	
BP_CMD_GET_REPORTDATA	Word	16#0042	
BP_CMD_MAKE_BEEP	Word	16#0051	
BP_CMD_STOP_SOUND	Word	16#0053	
BP_CMDERR_NONE	UInt	0	no error
BP_CMDERR_UNKNOWN	UInt	1	unknown command, invalid pattern
BP_CMDERR_BUSY	UInt	2	unit busy
BP_CMDERR_SIZE	UInt	3	invalid length or format (reception)
BP_CMDERR_VALUE	UInt	4	invalid value range
BP_CMDERR_MODE	UInt	5	invalid mode
BP_CMDERR_PENDING	UInt	6	pending
BP_CMDERR_FAILED	UInt	7	operation failed
BP_CMDERR_BLOCKED	UInt	8	operation blocked
BP_CMDERR_UNCOMPLETE	UInt	10	uncomplete command (reception)
BP_CMDERR_CRC	UInt	11	invalid crc (reception)
BP_CMDERR_IGNORE	UInt	12	command should be ignored (creation)
BP_CMDERR_TIMEOUT	UInt	13	no response in time (reception)

List of ECP2000 constants (ECP_Prot_Table)

Name	*Data Type*	*Value*	*Comment*
ECP_CMD_GET_STATUS	Word	16#0120	
ECP_CMD_DO_COMMAND	Word	16#0123	
ECP_CMD_GET_DESC	Word	16#0124	
ECP_CMD_GET_GRADCFG	Word	16#0162	
ECP_CMD_SET_GRADCFG	Word	16#0163	
ECP_CMD_GET_CFGA	Word	16#0166	Unsupported
ECP_CMD_SET_CFGA	Word	16#0167	Unsupported
ECP_CMD_GET_PUMPCFG2	Word	16#0168	
ECP_CMD_SET_PUMPCFG2	Word	16#0169	
ECP_CMD_GET_IFCCFG	Word	16#0170	Unsupported
ECP_CMD_SET_IFCCFG	Word	16#0171	Unsupported
ECP_CMD_GET_VALVEMAP	Word	16#0172	
ECP_CMD_SET_VALVEMAP	Word	16#0173	
ECP_CMD_GET_DFLOW	Word	16#0180	Read desired flow
ECP_CMD_SET_DFLOW	Word	16#0181	Write desired flow
ECP_CMD_GET_DPRESS	Word	16#0182	Read desired pressure
ECP_CMD_SET_DPRESS	Word	16#0183	Write desired pressure
ECP_CMD_GET_PRESSLIMITS	Word	16#0184	
ECP_CMD_SET_PRESSLIMITS	Word	16#0185	
ECP_CMD_GET_VALVE	Word	16#0186	
ECP_CMD_SET_VALVE	Word	16#0187	
ECP_CMD_GET_CONC	Word	16#0188	
ECP_CMD_SET_CONC	Word	16#0189	
ECP_CMD_GET_PRES	Word	16#01A0	
ECP_STATUS_INIT	USInt	0	
ECP_STATUS_PREP	USInt	1	
ECP_STATUS_STOPPED	USInt	2	
ECP_STATUS_RUN	USInt	3	
ECP_DOCMD_NONE	USInt	0	
ECP_DOCMD_PUMP	USInt	1	
ECP_DOCMD_STOP	USInt	2	
ECP_DOCMD_PUMPSAFE	USInt	8	
ECP_DOCMD_PURGEMODEON	USInt	101	
ECP_DOCMD_PURGEMODEOFF	USInt	102	
ECP_DOCMD_CLOSEVALVES	USInt	111	
ECP_DOCMD_QUITPROG	USInt	112	
ECP_DOCMD_OPENVALIDVALVE	USInt	113	

List of functions in ECOM_DEVICES library

Name	***Category***	***Comment***
SerialCOM	Communication	FB for serial communication
TCPCom	Communication	FB for TCP communication
BIGPROT_CalcCrc	General	Calculate BP CRC for demanded data
BIGPROT_TestRHeader	General	Test complete received header (size, CRC, …)
Cipher_TO_DecChar	General	Convert cipher to decadic value
DecChars_TO_DInt	General	Convert decadic character (‘0’ to ‘9’) to value
GetArraySize	General	Return size of an array
HexChars_TO_Real	General	Convert hexadecimal characters to real value
HexChars_TO_UDInt	General	Convert hexadecimal characters to integer value
Nibble_TO_HexChar	General	Convert half-byte to hexadecimal character
TOY18_Control	TOY18DAD	FB for basic control of detector TOY18DAD
TOY18_PrintErrorNums	TOY18DAD	Convert detector error array to string (list of numbers)
TOY18_PrintDetStatus	TOY18DAD	Convert detector status code to string
TOY18_Prot	TOY18DAD	FB for detector TOY18DAD protocol
TOY18_CommandID	TOY18DAD-LinkedFC	Return value from command/response header
TOY18_CreateCommandStr	TOY18DAD-LinkedFC	Create command string for demanded command (WPars is needed)
TOY18_DecodeAbsorbance	TOY18DAD-LinkedFC	Decode absorbance response (RPars.Absorbance)
TOY18_DecodeBoolVal	TOY18DAD-LinkedFC	Decode logic response
TOY18_DecodeCmdError	TOY18DAD-LinkedFC	Decode error response (return error)
TOY18_DecodeDetModel	TOY18DAD-LinkedFC	Decode detector model from already read parameters (name, model)
TOY18_DecodeIntensity	TOY18DAD-LinkedFC	Decode intensity response (RPars.Intensity)
TOY18_DecodeResponse	TOY18DAD-LinkedFC	Decode any response
TOY18_DecodeScan	TOY18DAD-LinkedFC	Decode scan response (RPars.Scan)
TOY18_DecodeStatus	TOY18DAD-LinkedFC	Decode status response (states, errors, io, lamp times, intensity)
TOY18_DecodeStatusOld	TOY18DAD-LinkedFC	Decode old status response (state, old errors)
TOY18_DecodeWavelength	TOY18DAD-LinkedFC	Decode wavelength response (RPars.Wavelength)
TOY18_TestChannels	TOY18DAD-LinkedFC	Check response for channels
TOY18_TestCheckSum16	TOY18DAD-LinkedFC	Check response for 16bit checksum
TOY18_TestCheckSum24	TOY18DAD-LinkedFC	Check response for 24bit checksum
ECP_Control	ECP2000	FB for basic control of pump ECP2000
ECP_PrintErrorNums	ECP2000	Convert pump error array to string (list of numbers)
ECP_PrintPumpStatus	ECP2000	Convert pump status code to string
ECP_Prot	ECP2000	FB for pump ECP2000 protocol
ECP_CreateCommand	ECP2000-LinkedFC	Create command data for demanded pump command
ECP_DecodeResponse	ECP2000-LinkedFC	Decode any pump response
ECP_IsError	ECP2000-LinkedFC	Test for error in error array (RPars.Errors)
DBGLog	Debug	Store string into log (MyDebug_DB is needed)
DBGLogBuf	Debug	Store buffer into log (MyDebug_DB is needed)
MyDebug	Debug	FB for storing logs (buffer of strings)
GetSysTicks	SystemTicks	Return millisecond system time (SystemTicks_DB is needed)
SystemTicks	SystemTicks	FB for operation of system ticks