

ONLINE HELP

LECTOR® 620 Image-Based Code Reader

Clever, simple, industrial



SICK
Sensor Intelligence.

Software	Function	Status
Device description (LECTOR620XX.sdd)	Operating and configuring the LECTOR®620	V 1.XX
SOPAS-ET	Operating and configuration software	V 2.38

Copyright

Copyright © 2013

SICK AG, Waldkirch

Auto Ident, Werk Reute

Nimburger Strasse 11

79276 Reute

Germany

Trademarks

Windows 2000™, XP™, Vista™, and Internet Explorer™ are registered trademarks or trademarks of the Microsoft Corporation in the USA and other countries.

Acrobat™ Reader™ is a trademark of Adobe Systems Incorporated.

Table of contents

- 1 Notes on this document.....13**
- 2 Software interface15**
- 3 Online images.....16**
 - 3.1 Online images.....16
 - 3.1.1 Image history18
 - 3.1.2 Code infobox.....19
 - 3.1.3 Statistics20
- 4 Stored images21**
 - 4.1 Saved images21
 - 4.1.1 Image history22
 - 4.1.2 Code infobox.....23
- 5 Parameters25**
 - 5.1 General25
 - 5.1.1 Configuration is password protected25
 - 5.2 Auto setup via function button menu25
 - 5.2.1 Change reading distance26
 - 5.2.2 Change camera settings26
 - 5.2.3 Change code settings26
 - 5.3 Function buttons26
 - 5.3.1 Function26
 - 5.4 Reading configuration.....27
 - 5.4.1 Camera and lighting.....27
 - 5.4.1.1 Reading distance27
 - 5.4.1.1.1 Reading distance.....29
 - 5.4.1.1.2 Auto29
 - 5.4.1.2 Exposure time.....29
 - 5.4.1.2.1 Exposure time29
 - 5.4.1.2.2 Exposure time30
 - 5.4.1.3 Image settings30
 - 5.4.1.3.1 Brightness30
 - 5.4.1.3.2 Brightness30
 - 5.4.1.3.3 Contrast.....31
 - 5.4.1.3.4 Contrast.....31
 - 5.4.1.3.5 Auto31
 - 5.4.1.4 Illumination.....31
 - 5.4.1.4.1 Internal31
 - 5.4.1.4.2 Aiming laser32
 - 5.4.1.4.3 Green feedback spot.....33
 - 5.4.1.4.4 Duration33
 - 5.4.1.5 Image filters.....33
 - 5.4.1.5.1 Noise suppression33
 - 5.4.1.5.2 Dot size34
 - 5.4.1.6 Increase performance.....34
 - 5.4.1.6.1 Setting image frequency manually.....35
 - 5.4.1.6.2 Image frequency input field35
 - 5.4.1.6.3 Image rotation 180°35
 - 5.4.1.6.4 Image region of interest.....35
 - 5.4.2 Object trigger control.....35
 - 5.4.2.1 Starting/Stopping the object trigger36
 - 5.4.2.1.1 Trigger delay36
 - 5.4.2.1.2 Start delay36
 - 5.4.2.1.3 Start by36
 - 5.4.2.1.4 Stop delay37
 - 5.4.2.1.5 Stop by38
 - 5.4.2.1.6 Reading gate length39
 - 5.4.2.1.7 Or39

5.4.2.1.8	Or.....	40
5.4.2.1.9	Pulse	40
5.4.2.1.10	Pause	40
5.4.2.1.11	Of.....	41
5.4.2.1.12	Trigger echo on.....	41
5.4.2.1.13	Reading gate on	41
5.4.2.1.14	Reading gate off.....	41
5.4.2.2	Trigger distribution.....	41
5.4.2.2.1	Distribution	42
5.5	Position.....	42
5.5.1	Coordinates	42
5.5.1.1	X-coordinate	42
5.5.1.2	Y-coordinate	42
5.5.1.3	Z-coordinate	42
5.5.2	Angle.....	42
5.5.2.1	Alpha.....	42
5.5.2.2	Beta	43
5.5.2.3	Gamma	43
5.6	Increment configuration.....	43
5.6.1	Increment	43
5.6.1.1	Increment source.....	43
5.6.1.2	Fixed speed	44
5.6.1.3	System increment resolution	44
5.7	Code configuration	44
5.7.1	General	44
5.7.1.1	Automatic code configuration.....	44
5.7.2	1D Symbologies	44
5.7.2.1	Minimum module width.....	44
5.7.2.2	Codabar.....	45
5.7.2.3	Code 39.....	45
5.7.2.4	UPC/GTIN/EAN	45
5.7.2.5	2/5 Interleaved.....	46
5.7.2.6	Code 93.....	46
5.7.2.7	Code 128 Family.....	46
5.7.2.8	GS1 DataBar	47
5.7.2.9	Pharmacode.....	47
5.7.2.10	Stacked codes	47
5.7.2.10.1	PDF417.....	47
5.7.3	Increase 1D performance	48
5.7.3.1	Code contrast.....	48
5.7.3.2	Code background.....	48
5.7.4	2D code types	48
5.7.4.1	Minimum cell size.....	49
5.7.4.2	Data matrix.....	49
5.7.5	Increase 2D performance	50
5.7.5.1	Code contrast.....	50
5.7.5.2	Code background.....	50
5.7.5.3	Code alignment.....	50
5.7.6	Codabar.....	51
5.7.6.1	General	51
5.7.6.1.1	Multiread	51
5.7.6.1.2	Start/Stop identical	51
5.7.6.1.3	Outputting start/stop.....	52
5.7.6.1.4	Check digit test	52
5.7.6.2	Length.....	52
5.7.6.2.1	Code length	52
5.7.6.2.2	Interval.....	53

5.7.6.2.3	Fixed length.....	53
5.7.7	Code 39	53
5.7.7.1	General	53
5.7.7.1.1	Multiread.....	53
5.7.7.1.2	Transmit start/stop	54
5.7.7.1.3	Full ASCII	54
5.7.7.1.4	Check digit test.....	54
5.7.7.1.5	C32 conversion.....	54
5.7.7.1.6	Hex – ASCII output	55
5.7.7.2	Length	55
5.7.7.2.1	Code length.....	55
5.7.7.2.2	Interval	55
5.7.7.2.3	Fixed length.....	55
5.7.8	UPC/GTIN/EAN	55
5.7.8.1	General	56
5.7.8.1.1	Multiread.....	56
5.7.8.1.2	Add-on	56
5.7.8.2	UPC.....	56
5.7.8.2.1	UPC A.....	56
5.7.8.2.2	UPC E.....	57
5.7.8.2.3	UPC E extended	57
5.7.8.3	EAN.....	57
5.7.8.3.1	GTIN 8 / EAN 8	57
5.7.8.3.2	GTIN 13 / EAN 13.....	58
5.7.9	2/5 interleaved	58
5.7.9.1	General	59
5.7.9.1.1	Multiread.....	59
5.7.9.1.2	Check digit test.....	59
5.7.9.1.3	Check-Digit-Test #2	59
5.7.9.1.4	Check-Digit-Test #3	59
5.7.9.1.5	Check-Digit-Test #4	59
5.7.9.1.6	Check-Digit-Test #5	59
5.7.9.2	Length	59
5.7.9.2.1	Code length.....	60
5.7.9.2.2	Interval	60
5.7.9.2.3	Fixed length.....	60
5.7.10	Code 93	60
5.7.10.1	General	61
5.7.10.1.1	Multiread.....	61
5.7.10.2	Length	61
5.7.10.2.1	Code length.....	61
5.7.10.2.2	Interval	61
5.7.10.2.3	Fixed length.....	61
5.7.11	Code 128 family	62
5.7.11.1	General	62
5.7.11.1.1	Code 128	62
5.7.11.1.2	EAN 128.....	62
5.7.11.1.3	Multiread.....	63
5.7.11.2	Length	63
5.7.11.2.1	Code length.....	63
5.7.11.2.2	Interval	63
5.7.11.2.3	Fixed length.....	64
5.7.11.3	GS1/EAN 128.....	64
5.7.11.3.1	FC1-Value within code.....	64
5.7.11.3.2	FC1-Value on first position.....	64
5.7.12	GS1 DataBar.....	64
5.7.12.1	General	64

5.7.12.1.1	DataBar 14.....	65
5.7.12.1.2	DataBar Expanded.....	65
5.7.12.1.3	DataBar Limited.....	66
5.7.13	Pharmacode.....	66
5.7.13.1	General.....	66
5.7.13.1.1	Multiread.....	67
5.7.13.1.2	Module width.....	67
5.7.13.1.3	Reverse.....	67
5.7.13.2	Not calibratable.....	67
5.7.13.2.1	Fixed length.....	67
5.7.14	Data matrix.....	67
5.7.14.1	General.....	68
5.7.14.1.1	Code format.....	68
5.7.14.1.2	Max. permissible error correction.....	68
5.7.14.1.3	Activate ISO15415 verification.....	68
5.7.14.2	Length.....	68
5.7.14.2.1	Symbol size.....	69
5.7.14.2.2	Fixed length 1.....	69
5.7.14.2.3	Fixed length 2.....	69
5.7.14.2.4	Fixed length 3.....	69
5.7.14.2.5	Fixed length 4.....	70
5.7.14.2.6	Fixed length 5.....	70
5.7.14.2.7	Allow rectangular data fields.....	70
5.7.14.3	Reducing evaluation time.....	70
5.7.14.3.1	Code surrounded by patterns.....	70
5.7.14.3.2	Code surrounded by text.....	70
5.7.14.3.3	Decoding.....	71
5.7.14.4	Increasing robustness.....	71
5.7.14.4.1	Errors in L-pattern.....	71
5.7.14.5	GS1 format.....	71
5.7.14.5.1	Replacing the FNC1 character.....	71
5.7.14.5.2	Replacing the separator.....	73
5.7.14.5.3	Application identifier marking.....	74
5.7.15	PDF 417.....	74
5.7.15.1	General.....	74
5.8	Data processing.....	75
5.8.1	Collection of data.....	75
5.8.1.1	Timeout.....	75
5.8.2	Code summarization.....	75
5.8.2.1	Separate codes depending on position.....	75
5.8.2.2	Code distance.....	75
5.8.2.3	Separate codes dependent on sensor.....	76
5.8.3	Output control.....	76
5.8.3.1	Output control.....	76
5.8.3.1.1	Control.....	76
5.8.3.1.2	Output time.....	76
5.8.3.1.3	Output condition.....	76
5.8.3.1.4	Data output mode.....	77
5.8.3.1.5	Label timeout.....	77
5.8.3.1.6	Condition timeout.....	77
5.8.3.1.7	Delay.....	77
5.8.3.1.8	Output delay.....	77
5.8.3.1.9	Certain numb. of new labels.....	78
5.8.3.1.10	Label timeout active.....	78
5.8.3.1.11	Timeout.....	78
5.8.4	Evaluation conditions.....	78
5.8.4.1	Conditions for Good Read.....	78

- 5.8.4.1.1 Check min. number of valid codes..... 79
- 5.8.4.1.2 Check max. number of valid codes 79
- 5.8.4.2 Evaluation conditions..... 79
- 5.8.5 Match code 80
- 5.8.5.1 Matchcode Teach-in 1 80
- 5.8.5.1.1 Activating teach-in mode 80
- 5.8.5.1.2 Or teach-in via function buttons 81
- 5.8.5.1.3 Teach-in stop by..... 81
- 5.8.5.1.4 Teach-in condition 81
- 5.8.5.1.5 Inverting a condition..... 81
- 5.8.5.1.6 Teach-in code content..... 81
- 5.8.5.1.7 Teach-in code ID (type) 81
- 5.8.5.1.8 Teach-in code length 82
- 5.8.5.2 Match-code Teach-in 2 (Additional) 82
- 5.8.5.2.1 Activate teach-in mode 82
- 5.8.5.2.2 Teach-in stop by..... 83
- 5.8.5.2.3 Teach-in condition 83
- 5.8.5.2.4 Invert condition 83
- 5.8.5.2.5 Teach-in code content..... 83
- 5.8.5.2.6 Teach-in code ID (type) 83
- 5.8.5.2.7 Teach-in code length 83
- 5.8.5.3 General Match code teach-in - settings..... 84
- 5.8.5.3.1 Start teach-in 84
- 5.8.5.3.2 Allowed code types..... 84
- 5.8.5.3.3 Code configuration 84
- 5.8.5.3.4 Save permanent 85
- 5.8.6 Filters/Sorters for the output formatting..... 85
- 5.8.6.1 Filters/Sorters for Output Format1..... 85
- 5.8.6.2 Filters/Sorters for Output Format2 91
- 5.8.7 Output format 96
- 5.8.7.1 Output format 1..... 96
- 5.8.7.2 Output format 2..... 102
- 5.8.7.3 Heartbeat format..... 107
- 5.8.8 Serial input data 108
- 5.8.8.1 Data forwarding..... 108
- 5.8.8.1.1 Rx header 108
- 5.8.8.1.2 Rx terminator 108
- 5.8.8.1.3 Target interface 108
- 5.8.8.1.4 Tx header 109
- 5.8.8.1.5 Tx terminator..... 109
- 5.8.8.2 Auxiliary input/Auxiliary read result 109
- 5.8.8.2.1 Rx header 109
- 5.8.8.2.2 Rx terminator 109
- 5.8.8.2.3 Bar code type..... 109
- 5.8.8.2.4 Code IDs for bar codes..... 110
- 5.8.8.3 EDS text element in output format / element output format ... 110
- 5.8.8.3.1 Rx header 111
- 5.8.8.3.2 Rx terminator 111
- 5.8.8.3.3 Default text 111
- 5.8.9 Application counters 111
- 5.8.9.1 Application counter 111
- 5.8.9.1.1 Resetting counter statuses..... 112
- 5.8.9.1.2 Saving current counter statuses 112
- 5.8.9.1.3 Application counter 112
- 5.8.9.1.4 Value 113
- 5.8.9.1.5 Saving the counter state permanently..... 113
- 5.9 Analysis tools..... 113

5.9.1	Device time.....	113
5.9.1.1	Device time	113
5.9.1.1.1	Time source	113
5.9.1.1.2	Time server IP address	113
5.9.1.1.3	Port.....	114
5.9.1.1.4	Connection timeout	114
5.9.1.1.5	Time difference to GMT	114
5.9.1.1.6	Date (YYYY-MM-DD)	114
5.9.1.1.7	Time (hh:mm:ss)	114
5.9.2	Image diagnosis.....	114
5.9.2.1	Image diagnostic.....	114
5.9.2.1.1	Conditions for image storing.....	115
5.9.2.1.2	Prioritizing image output over the trigger.....	115
5.9.2.1.3	Image selection	115
5.9.2.1.4	Image quality.....	116
5.9.2.2	Save destination	116
5.9.2.2.1	Internal memory (permanent).....	116
5.9.2.2.4	Internal memory (temporary)	117
5.9.2.2.7	MicroSD card.....	117
5.9.2.2.12	SOPAS (PC).....	118
5.9.2.3	Image file path	119
5.9.2.3.1	Folder name	119
5.9.2.3.2	Date	119
5.9.2.3.3	Hour	120
5.9.2.3.4	Subfolder Good Read / No Read	120
5.9.2.3.5	Good Read / No Read file prefix.....	120
5.9.2.3.6	Image file name	120
5.10	Network/Interfaces/IOs	120
5.10.1	Network options.....	121
5.10.1.1	Device ID	121
5.10.1.2	Device name	121
5.10.2	Master/Slave	121
5.10.2.1	Function.....	122
5.10.2.2	Slave list	122
5.10.3	Multiplexer/Server	122
5.10.3.1	Function.....	123
5.10.3.2	Server list	123
5.10.4	Monitoring	123
5.10.4.1	Assign to.....	123
5.10.4.2	Monitored devices	123
5.10.4.3	Max. startup time for the monitored devices.....	123
5.10.5	Serial.....	123
5.10.5.1	Serial Host.....	123
5.10.5.1.1	Output Format.....	124
5.10.5.1.2	With number.....	124
5.10.5.1.3	Multiplexer output.....	124
5.10.5.1.4	Baud rate	124
5.10.5.1.5	Stop bits.....	124
5.10.5.1.6	Data bits / parity	124
5.10.5.1.7	Hardware	124
5.10.5.1.8	Enable heartbeat	124
5.10.5.1.9	Heartbeat interval.....	125
5.10.5.1.10	Restart interval on sending.....	125
5.10.5.1.11	Handling input data	125
5.10.5.2	Serial Aux.....	125
5.10.5.2.1	Output format.....	125
5.10.5.2.2	RDT ID.....	126

5.10.5.2.3	Enable heartbeat.....	126
5.10.5.2.4	Heartbeat interval	126
5.10.5.2.5	Restart interval on sending.....	126
5.10.5.2.6	Handling input data.....	126
5.10.6	Ethernet	127
5.10.6.1	Ethernet General	127
5.10.6.1.1	Addressing Mode.....	127
5.10.6.1.2	IP address	127
5.10.6.1.3	Subnet-Mask.....	127
5.10.6.1.4	Default gateway.....	127
5.10.6.1.5	Speed	128
5.10.6.1.6	Negotiated	128
5.10.6.1.7	MAC-Address.....	128
5.10.6.2	Ethernet Host Port.....	128
5.10.6.2.1	Output Format	128
5.10.6.2.2	Multiplexer output	128
5.10.6.2.3	Server/Client.....	128
5.10.6.2.4	IP port.....	128
5.10.6.2.5	Server Address	129
5.10.6.2.6	Enable heartbeat.....	129
5.10.6.2.7	Heartbeat interval	129
5.10.6.2.8	Restart interval on sending.....	129
5.10.6.3	Ethernet Aux Port	129
5.10.6.3.1	Output Format	129
5.10.6.3.2	Server/Client.....	130
5.10.6.3.3	IP port.....	130
5.10.7	Ethernet/IP	130
5.10.7.1	Ethernet/IP	130
5.10.7.1.1	Activate Ethernet/IP.....	130
5.10.7.1.2	Communication protocol.....	130
5.10.7.1.3	Protocol/Output format.....	130
5.10.7.1.4	Assembly size output on the PLC	130
5.10.7.1.5	Assembly size input on the PLC.....	131
5.10.8	Web server.....	131
5.10.8.1	Web server.....	131
5.10.8.1.1	Activate web server at device startup.....	131
5.10.9	Monitoring.....	131
5.10.9.1	Monitored ports.....	131
5.10.9.1.1	Serial host interface (port 4003)	131
5.10.9.1.2	Serial auxiliary interface (port 4002)	132
5.10.9.1.3	DHCP Fallback Mode.....	132
5.10.10	CAN.....	132
5.10.10.1	CAN.....	132
5.10.10.1.1	Mode	132
5.10.10.1.2	Using device ID as a node ID	132
5.10.10.1.3	Device ID	133
5.10.10.1.4	Data transmission rate	133
5.10.10.1.5	Output format	133
5.10.10.1.6	Enable heartbeat.....	133
5.10.10.1.7	Heartbeat interval	133
5.10.10.1.8	Restart interval on sending.....	133
5.10.10.1.9	Mask for digital input	133
5.10.10.1.10	Enable digital output	134
5.10.10.1.11	COB ID RPDO digital input	134
5.10.10.1.12	COB ID TPDO digital output.....	135
5.10.10.1.13	Enable user IO PDOs	136
5.10.10.1.14	COB ID RPDO user digital input.....	136

5.10.10.1.15	COB ID TPDO user digital output.....	136
5.10.10.1.16	Number of mapped RPDO bytes.....	136
5.10.10.1.17	Number of mapped TPDO bytes.....	136
5.10.10.1.18	Reading result with SDO/PDO.....	136
5.10.10.1.19	Reading result timeout.....	137
5.10.10.1.20	Automatic release after SDO upload.....	137
5.10.10.1.21	Enable command response output.....	137
5.10.10.1.22	Command response timeout.....	137
5.10.10.1.23	Basic COB ID for PDO reading results.....	138
5.10.10.1.24	Transmission type.....	138
5.10.10.1.25	Inhibit time.....	138
5.10.10.1.26	Number of PDOs.....	138
5.10.10.1.27	Enable diagnosis output.....	138
5.10.10.1.28	Diagnostics timeout.....	139
5.10.10.1.29	CANopen heartbeat rate / ms.....	139
5.10.10.1.30	COB ID Emergency Obj.....	139
5.10.10.1.31	Emergency Inhibit Time.....	139
5.10.10.1.32	CANopen Transmit PDOs 1-4.....	139
5.10.10.1.33	CANopen Receive PDOs 1-4.....	140
5.10.11	Fieldbus gateway.....	140
5.10.11.1	Profibus proxy CDF600.....	140
5.10.11.1.1	Slave address.....	140
5.10.11.1.2	Communication protocol.....	141
5.10.11.1.3	Protocol/Output format.....	141
5.10.11.1.4	Using PLC output bit 0.....	141
5.10.11.1.5	Using PLC output bit 1.....	142
5.10.11.1.6	Using PLC input bit 0.....	142
5.10.11.1.7	Using PLC input bit 1.....	142
5.10.11.2	Profibus/DeviceNet/Profinet Gateway CMF400/CDM425.....	143
5.10.11.2.1	Serial auxiliary interface.....	143
5.10.11.2.2	Serial host interface.....	143
5.10.11.3	Profibus DP gateway.....	144
5.10.11.3.1	Using device ID as a Profibus address.....	144
5.10.11.3.2	Device ID.....	144
5.10.11.3.3	Operating mode.....	144
5.10.11.4	DeviceNet gateway.....	144
5.10.11.4.1	Using device ID as a DeviceNet address.....	144
5.10.11.4.2	Device ID.....	144
5.10.11.4.3	Operating mode.....	144
5.10.11.4.4	Data transmission rate.....	145
5.10.11.4.5	Useful input data length.....	145
5.10.11.4.6	Useful output data length.....	145
5.10.11.5	Profinet IO gateway CDM425-PN.....	145
5.10.11.5.1	DHCP.....	145
5.10.11.5.2	IP address.....	145
5.10.11.5.3	Subnet mask.....	145
5.10.11.5.4	Default gateway.....	145
5.10.11.5.5	Station name.....	145
5.10.12	Digital inputs.....	146
5.10.12.1	Sensor / Result 1.....	146
5.10.12.1.1	Control.....	146
5.10.12.1.2	Sensitivity.....	146
5.10.12.1.3	Logic.....	146
5.10.12.1.4	Debouncing.....	147
5.10.12.2	Sensor / Result 2.....	147
5.10.12.2.1	Control.....	147
5.10.12.2.2	Sensitivity.....	147

5.10.12.2.3	Logic	147
5.10.12.2.4	Debouncing.....	147
5.10.12.3	External Input 1	148
5.10.12.3.1	Control.....	148
5.10.12.3.2	Sensitivity.....	148
5.10.12.3.3	Logic	148
5.10.12.3.4	Debouncing.....	148
5.10.12.4	External Input 2	149
5.10.12.4.1	Control.....	149
5.10.12.4.2	Sensitivity.....	149
5.10.12.4.3	Logic	149
5.10.12.4.4	Debouncing.....	149
5.10.13	Digital outputs/Beeper	150
5.10.13.1	Output / Result 1	150
5.10.13.1.1	Active.....	150
5.10.13.1.2	Function (Off).....	150
5.10.13.1.3	Or	151
5.10.13.1.4	Logic	151
5.10.13.1.5	Control.....	151
5.10.13.1.6	Length	152
5.10.13.2	Output/Result 2	152
5.10.13.2.1	Active.....	152
5.10.13.2.2	Function (Off).....	152
5.10.13.2.3	Or	153
5.10.13.2.4	Logic	153
5.10.13.2.5	Control.....	154
5.10.13.2.6	Length	154
5.10.13.3	Output / Result 3	154
5.10.13.3.1	Active.....	154
5.10.13.3.2	Function (Off).....	155
5.10.13.3.3	Or	155
5.10.13.3.4	Logic	155
5.10.13.3.5	Control.....	156
5.10.13.3.6	Length	156
5.10.13.4	Output / Result 4	156
5.10.13.4.1	Active.....	156
5.10.13.4.2	Function (Off).....	157
5.10.13.4.3	Or	157
5.10.13.4.4	Logic	157
5.10.13.4.5	Control.....	158
5.10.13.4.6	Length	158
5.10.13.5	External Output 1	158
5.10.13.5.1	Active.....	158
5.10.13.5.2	Function (Off).....	159
5.10.13.5.3	Or	159
5.10.13.5.4	Logic	159
5.10.13.5.5	Control.....	160
5.10.13.5.6	Length	160
5.10.13.6	External output 2.....	160
5.10.13.6.1	Active.....	160
5.10.13.6.2	Function (Off).....	161
5.10.13.6.3	Or	161
5.10.13.6.4	Logic	161
5.10.13.6.5	Control.....	162
5.10.13.6.6	Length	162
5.10.13.7	Beeper.....	162
5.10.13.7.1	Beeper.....	162

5.10.13.7.2	Volume.....	162
6	Service.....	163
6.1	Operating data	163
6.1.1	Device information.....	163
6.1.1.1	Manufacturer	163
6.1.1.2	Device type.....	163
6.1.1.3	Software version	163
6.1.1.4	Order number.....	163
6.1.1.5	Serial number	163
6.1.2	Operating data	163
6.1.3	Service information.....	163
6.2	System status	164
6.2.1	System information.....	164
6.3	Reading field/Scanning frequency.....	164
7	Analysis.....	165
7.1	Event monitor.....	165
7.1.1	Event monitor.....	165

1 Notes on this document

This document contains the **online help** of device description LECTOR620.sdd (for operation and configuration with SOPAS Single Device or SOPAS-ET).

Used symbols To gain easier access, some information in this documentation is emphasized as follows:



This symbol points out specific features.



This symbol indicates additional settings in the SOPAS-ET configuration software.



This symbol indicates supplementary technical documentation.

Intended use The camera-based LECTOR®620 is an intelligent sensor for the automatic, stationary decoding of codes on moving or still-standing objects. It reads all common 1D codes (barcodes)/ 2D codes (stacked codes/matrix codes). Via its host interface, the LECTOR®620 transmits the reading data to a higher-level computer for further processing.

- Safety information**
- Read the LECTOR®620 operating instructions and familiarize yourself with the device and its functions.
 - To avoid the dazzle caused by integrated illumination, do not look into the reading window when switching the LECTOR®620 on and off.



CAUTION

The accessible radiation of the laser LEDs poses no risk.

Temporary, irritating, optical effects on the human eye (e.g. dazzle, blindness from flash, after-images, impairment of color vision) cannot be completely ruled out, particularly with low ambient brightness. Precautionary measures are not required. Caution - incorrect use may result in the user being exposed to hazardous radiation.

- Do not intentionally look directly into the light sources for long periods of times.
- Observe the currently applicable regulations on photobiological safety of lamps and lamp systems as well as laser protection.



No maintenance is required in order to ensure compliance with risk group RG 1/laser protection class 1.

Further information



For information on startup, operation, and maintenance, see the **operating instructions** for the LECTOR®620.

For more information on the LECTOR®620, see the LECTOR®620 product page on the internet at the SICK Partner Portal, **www.mysick.com**:

- Detailed technical data in the online data sheet
- Scale drawing and 3D CAD scale models in various electronic formats
- EC Declaration of Conformity
- Integration of the Lector™620 in field bus systems
- Overview of the command strings
- Updates of the SOPAS ET configuration software
- Other useful software

You can also obtain assistance from your sales partner (**www.sick.com**).

2 Software interface

Wizards Pressing the AUTO-SETUP button on the WIZARDS tab on the left-hand side of the software interface automatically calls up the Auto Setup wizard for automatic setting of the following parameters:

- READING DISTANCE
- IMAGE SETTINGS
 - BRIGHTNESS
 - CONTRAST
- CODE SETTINGS



The automatic setting of these parameters can also be called up individually ([chapter 5.2 Auto setup via function button menu, page 25](#)).

Context help The CONTEXT HELP tab on the left-hand side of the software interface provides you with information on the currently selected parameters.

3 Online images

The recorded images are displayed on the [Online images](#) tab.

The display enables you to look at the images in detail and assess the recording quality. By modifying the configuration and comparing the recorded images, you can optimize the reading properties and the position of the LECTOR®620.

 The reading results are displayed in [Code infobox](#).

For an evaluation of the current setting for the contrast, the identified codes are displayed with different colors:

- Green (excellent): Contrast between 55 % and 100 %
- Yellow (good): Contrast between 20 % and 55 %
- Red (poor): Contrast less than 20 %

3.1 Online images

The most recently recorded image is displayed in the main view of the [Online images](#) group.

The display enables you to look at the images in detail and assess the quality of the shot. By modifying the configuration and comparing each of the images taken, you can optimize the reading properties and the position of the LECTOR®620.

 The reading results are displayed in the [Code infobox](#).

For an evaluation of the current setting for the contrast, the identified codes are displayed with different colors:

- Green (excellent): Contrast between 55% and 100%
- Yellow (good): Contrast between 20% and 55%
- Red (poor): Contrast less than 20%

Operation If you press the [Operation](#) button, the mode for setting up the LECTOR®620 is stopped, and the operating mode of the LECTOR®620 is activated in accordance with the current configuration.

 In [Image history](#), the recorded images are displayed one after the other. The reading results are displayed in the [Code infobox](#). The reading results for the last readings are combined in the display fields for the [Statistics](#) group.

For an evaluation of the current setting for the contrast, the identified codes are displayed with different colors:

- Green (excellent): Contrast between 55% and 100%
- Yellow (good): Contrast between 20% and 55%
- Red (poor): Contrast less than 20%

Setup Pressing the **Setup** button starts the reading process for testing the configuration. This involves images being taken in free-running mode and displayed in the main view.

Changes to the configuration (e.g., image settings or code configuration) or the position of the LECTOR®620 (e.g., reading distance or angle) are immediately visible in the main view. This makes it possible to complete test readings, as well as to check and optimize the current position and configuration of the LECTOR®620.

For an evaluation of the current setting for the contrast, the identified codes are displayed with different colors:

- Green (excellent): Contrast between 55% and 100%
- Yellow (good): Contrast between 20% and 55%
- Red (poor): Contrast less than 20%

Behavior of the LECTOR®620 during setup:

- Images are accepted into the image sequence at a rate of one per second so that changes can be recorded.
- To evaluate the current contrast settings, the codes that have been detected are highlighted with different colors.
- The code contents and other relevant data for all known codes are displayed in the code info box for the purposes of assessing the reading stability.
- The reading results are transferred to the AUX interface in the configured output format.
- The reading results are not transferred to the host interface.
- External triggers at the digital inputs are ignored.
- The trigger distribution on the CAN interface is deactivated.
- Digital outputs are not used.

Toolbar Pause

You can use the  button to pause the continuous display of recorded images during setup and in operating mode.

In the case of paused continuous display, reading continues; however, the currently displayed image can be looked at in more detail without being overwritten with a new image. The image region of interest can be moved with the mouse (↔).

You can use the  button to continue with the continuous display of recorded images.

Store

You can use the  button to store the currently displayed image on the PC. For each stored image, an xml file containing additional information is stored.

You can choose any image name and storage location.

Enlarge

You can use the  button to enlarge the image in increments.

The image region of interest can be moved with the mouse (↻).

To look at the code structure in more detail and thus find errors in the code, you can also enlarge the image using the mouse scroll wheel.

Reduce

You can use the  button to reduce the image in increments.

You can also reduce the image using the mouse scroll wheel.

Standard size

You can use the  button to display the image in its standard size.

3.1.1 Image history

In *Image history*, the recorded images are displayed one after the other. The images recorded most recently are added on the left.

By comparing and assessing the recorded images, you can analyze, for example,

- whether the trigger is set correctly and the appropriate region of the reading area was recorded
- whether the codes for moving objects were represented in focus and whether the shutter time was set correctly
- how often a code was recorded per reading gate.



To find out the reason for unsuccessful readings, you may find it useful to analyze the images without identified codes (NO READ).

The images displayed in reduced form in *Image history* are displayed by clicking in the main view. The file name and storage location of the images are displayed as a tooltip (mouseover).

You can use the  and  buttons to switch between several sequences.

The number of displayed images depends on the operating mode and the configuration:

- Depending on the configuration of the *Image selection* parameter, either the last 200 individual images are displayed (one individual image per trigger) or 60 images per second are displayed in individual sequences.
- During reading to test the configuration (operating mode *Setup*), the last 30 images recorded are displayed.

3.1.2 Code infobox

The reading results are displayed in the *Code infobox*. To evaluate the readability and read quality, the code contents are displayed along with other relevant data for all known codes. Codes that have been detected several times in one image are listed one below the other.



The columns can be enlarged and their sequence changed using drag and drop.

Code content

The contents of the codes are displayed in the CODE CONTENT column.

By assigning the code contents to the code, you can compare the read contents with the actual contents of the code.

CODE ID

The code type of the code is displayed in the CODE ID column. By assigning the code type to the code contents, you can activate/deactivate certain code types in the code configuration in order to limit multiple contents to one code type, for example.

Code contrast PCS

The contrast of the codes in % is displayed in the CODE CONTRAST PCS column.

A high contrast simplifies identification of a code. By comparing different configurations, you can find the setting for the highest possible contrast.

- From 55% to 100%: Contrast EXCELLENT
- From 20% to 55%: Contrast GOOD
- Less than 20%: Contrast POOR

Module width/height in px

The module width and height of the codes is displayed in pixels in the MODULE WIDTH/HEIGHT IN PX column.

The value can be used to evaluate the reading reliability (reserve) in relation to the image resolution. The higher the value, the more reliable the reading. A value of at least 2/2, is recommended, i.e. every code element is represented by at least 2/2 pixels.

By comparing the two values, you may be able to detect a distortion of the image or of the code.

- The module width is the dimension of the smallest code element (bar of a bar code or dot in a data matrix code) in pixels.
- The module height is the length of a bar of a bar code or the height of a dot in the data matrix code.

3.1.3 Statistics

The display fields of the [Statistics](#) group are used to evaluate the reading results across all triggers since the device was started or since the last reset.

The reading results during setup of the LECTOR®620 are not taken into consideration in the statistics.

Reading gates The number of all triggers since the device was started or since the last reset are displayed in the [Reading gates](#) display field.

The triggers during setup of the LECTOR®620 are not taken into consideration in the statistics.

Good Reads The number of triggers for which the condition for Good Read was met is displayed in the [Good Reads](#) display field.

The triggers during setup of the LECTOR®620 are not taken into consideration in the statistics.

No Reads The number of triggers for which the condition for Good Read was NOT met is displayed in the [No Reads](#) display field.

The triggers during setup of the LECTOR®620 are not taken into consideration in the statistics.

Read rate The proportion of triggers for which the condition for Good Read compared to all triggers is displayed as a % in the [Read rate](#) display field.

The triggers during setup of the LECTOR®620 are not taken into consideration in the statistics.

Reset The [Reset](#) button is used to delete the values of the [Reading gates](#), [Good Reads](#), [No Reads](#), and [Read rate](#) display fields.

4 Stored images

Stored images can be displayed on the [Stored images](#) tab.

The images stored in the LECTOR®620 or on the MicroSD memory card can be transferred to the PC. The data (images, xml files, etc.) stored in the LECTOR®620 or on the MicroSD memory card can also be deleted.



By comparing and assessing the recorded images, you can, for example, analyze which codes were not identified and whether faulty code structures have prevented a reading. For this, the images displayed in reduced form in [Image history](#) are displayed by clicking in the main view. The relevant code data is listed in [Code infobox](#). To classify the recorded images more effectively, you can use the right mouse button to display the date and time of recording.

For an evaluation of the current setting for the contrast, the identified codes are displayed with different colors:

- Green (excellent): Contrast between 55 % and 100 %
- Yellow (good): Contrast between 20 % and 55 %
- Red (poor): Contrast less than 20 %

4.1 Saved images

Images that have been saved are displayed on the main display of the [Saved images](#) group.

The images saved in the LECTOR®620 or on the microSD memory card can be transferred to the PC, where they can be stored in a database, for example, and opened when required. The data (images, .xml files, etc.) saved in the LECTOR®620 or on the microSD memory card can also be deleted.



By comparing and evaluating the images that have been taken, it is possible to analyze, for example, which codes were not identified and whether faulty code structures affected the reading. Click on the smaller images on display in the [Image history](#) to display them in the main view. The relevant code data is listed in the [Code infobox](#). Right-clicking the mouse displays the date and time the images were taken, allowing them to be organized better.

For an evaluation of the current setting for the contrast, the identified codes are displayed with different colors:

- Green (excellent): Contrast between 55% and 100%
- Yellow (good): Contrast between 20% and 55%
- Red (poor): Contrast less than 20%

Copying images The images saved in the LECTOR®620 or on the microSD memory card are transferred to the PC via the [Copying images](#) button.

It is possible to select the storage location on the PC.

Deleting images The data (images, .xml files, etc.) saved in the LECTOR®620 or on the microSD memory card can also be removed by pressing the [Deleting images](#) button.

Loading images The images that have been transferred previously are displayed by pressing the [Loading images](#) button.

Toolbar Store

You can use the  button to store the currently displayed image on the PC. For each stored image, an xml file containing additional information is stored.

You can choose any image name and storage location.

Enlarge

You can use the  button to enlarge the image in increments.

Reduce

You can use the  button to reduce the image in increments.

You can also reduce the image using the mouse scroll wheel.

Standard size

You can use the  button to display the image in its standard size.

4.1.1 Image history

In [Image history](#), the recorded images are displayed one after the other. The images recorded most recently are added on the left.

By comparing and assessing the recorded images, you can analyze, for example,

- how often a code was recorded in total
- how often a code was recorded per reading gate
- whether the codes for moving objects were represented in focus and whether the shutter time was set correctly



To find out the reason for unsuccessful readings, you may find it useful to analyze the images without identified codes (NO READ).

The images displayed in reduced form in [Image history](#) are displayed by clicking in the main view. The file name and storage location of the images are displayed as a tooltip (mouseover).

You can use the  and  buttons to switch between several sequences.

The number of displayed images depends on the operating mode and the configuration:

- Depending on the configuration of the *Image selection* parameter, either the last 200 individual images are displayed (one individual image per trigger) or 60 images per second are displayed in individual sequences.
- During reading to test the configuration (operating mode *Setup*), the last 30 images recorded are displayed.

4.1.2 Code infobox

The reading results are displayed in the *Code infobox*. To evaluate the readability and read quality, the code contents are displayed along with other relevant data for all known codes. Codes that have been detected several times in one image are listed one below the other.



The columns can be enlarged and their sequence changed using drag and drop.

CODE CONTENT

The contents of the codes are displayed in the CODE CONTENT column.

By assigning the code contents to the code, you can compare the read contents with the actual contents of the code.

CODE ID

The code type of the code is displayed in the CODE ID column. By assigning the code type to the code contents, you can activate/deactivate certain code types in the code configuration in order to limit multiple contents to one code type, for example.

Code contrast PCS

The contrast of the codes in % is displayed in the CODE CONTRAST PCS column.

A high contrast simplifies identification of a code. By comparing different configurations, you can find the setting for the highest possible contrast.

- From 55% to 100%: Contrast EXCELLENT
- From 20% to 55%: Contrast GOOD
- Less than 20%: Contrast POOR

Module width/height in px

The module width and height of the codes is displayed in pixels in the MODULE WIDTH/HEIGHT IN PX column.

The value can be used to evaluate the reading reliability (reserve) in relation to the image resolution. The higher the value, the more reliable the reading. A value of at least $2/2$, is recommended, i.e. every code element is represented by at least $2/2$ pixels.

By comparing the two values, you may be able to detect a distortion of the image or of the code.

- The module width is the dimension of the smallest code element (bar of a bar code or dot in a data matrix code) in pixels.
- The module height is the length of a bar of a bar code or the height of a dot in the data matrix code.

5 Parameters

Settings for protecting the configuration can be made on the [Parameters](#) tab.

The PARAMETER folder in the project tree combines all available parameters.

5.1 General

Password protection for the device configuration can be activated in the [General](#) group.

Password protection ensures that only authorized users can enter device parameters. This prevents incorrect settings from being made.

5.1.1 Configuration is password protected

If this parameter is activated, access to the device configuration is password protected. To be able to change parameters, you must log in to the device at the user level of authorized customer or higher. You can use the SOPAS options, which can be found outside of the device configuration, without logging in. Password protection is activated the next time the device is restarted.

5.2 Auto setup via function button menu

The [Auto setup via function button menu](#) group contains the parameters for adjusting the auto-setup wizards.

The AUTO-SETUP function is divided into three individual modules:

- READING DISTANCE
- IMAGE SETTINGS (brightness and contrast)
- CODE SETTINGS (currently only available for DATA MATRIX)

The AUTO-SETUP function can be started in different ways:

- via wizard
- via the pushbuttons on LECTOR®620
- by pressing the AUTO buttons
- via command (SOPAS command)

Commands for starting the AUTO-SETUP function:

- SMN MASSTART: The Auto-Setup process is started.
- sMN MASFINISH: The Auto-Setup process is stopped and the result is transmitted. To return to reading mode, you must end the Auto-Setup process with a stop command.
- sMN MASCANCEL: The Auto-Setup process is stopped. This does, however, cause the result to be discarded. To return to reading mode, you must end the Auto-Setup process with a stop command.



Even if you start the AUTO-SETUP function via a command, the settings for the wizard (standard or advanced) are taken into consideration. The settings can also be configured using commands. A detailed list of commands can be obtained from SICK on request.

5.2.1 Change reading distance

If the *Change reading distance* parameter is activated, the reading distance is modified using the AUTO-SETUP function. The function is called up using the buttons on the LECTOR®620.

5.2.2 Change camera settings

If the *Change camera settings* parameter is activated, the brightness and contrast are modified using the AUTO-SETUP function. The function is called up using the buttons on the LECTOR®620.

5.2.3 Change code settings

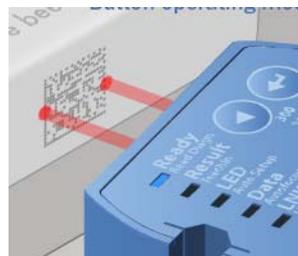
If the *Change code settings* parameter is activated, the code settings for the DATAMATRIX symbology are modified using the AUTO-SETUP function. The function is called up using the buttons on the LECTOR®620.

5.3 Function buttons

The *Function buttons* contains the parameters for defining the functions of the buttons on the LECTOR®620.

5.3.1 Function

You use the *Function* parameter to define the function and behavior of the buttons on the LECTOR®620.



- **NO FUNCTION:** The function buttons on the LECTOR®620 are deactivated. As a result, you can avoid malfunctions during operation caused by pressing the buttons by accident.
- **TRIGGER AND AIMING LASER:** You can use the buttons on the LECTOR®620 to carry out a reading without the LECTOR®620 having to be connected to the PC.
 - You use the left button to open the reading gate. The reading gate remains open until the button is released again or until the condition for closing the reading gate has been met.
 - You use the right button to turn the aiming laser on/off. The aiming laser is automatically turned off after 5 minutes.
- **MENU (SAVE PERMANENT):** The buttons on the LECTOR®620 have a menu function and can be used for parameterization. The changed parameter values are permanently stored in the device. Existing values are overwritten.
- **MENU (SAVE TEMPORARY):** The buttons on the LECTOR®620 have a menu function and can be used for parameterization. Existing values are initially overwritten. The changed parameter values are retained and stored in the device until the device is restarted. As the old parameter values are restored after the restart, this setting is useful for testing parameter changes. The function can also be used to temporarily read another code and reset the settings by restarting the device.

If the menu function is activated, the following functions can be accessed using the buttons on the LECTOR®620:

- **READ DIAGN:** A suitable code in the reading area is read with the current configuration. The reading rate of the last 10 readings is displayed in % via the bar chart on the housing of the LECTOR®620.
- **TEACHIN MATCHCODE:** A suitable code in the reading area is read and stored as a match-code.
- **AUTO-SETUP** (initially only available for DATA MATRIX): Depending on the configuration of the parameters in the [Auto setup via function button menu](#) group, the reading distance, image settings (brightness and contrast), and code settings are modified automatically. For this, a suitable object with contrast changes or edges (e.g. code, text, or other structures) must be placed in the reading area.
- **AUTOFOCUS:** The focus position (reading distance) is modified automatically. For this, a suitable object with contrast changes or edges (e.g. code, text, or other structures) must be placed in the reading area.
- **USERDEFINED:** (currently not implemented)

5.4 Reading configuration

5.4.1 Camera and lighting

All parameters that affect the quality of the images taken are combined on the [Camera and lighting](#) tab.

Optimizing the camera and lighting settings improves the contrast, sharpness and illumination. This results in better code identification and therefore increases the reliability of the reading.



A high image quality increases the time required to save the images taken. The save times can be lowered by reducing the image quality.

5.4.1.1 Reading distance

The distance between the LECTOR®620 and the object is set via the parameters for the [Reading distance](#) group. The reading distance is used to adapt the focal range of the LECTOR®620 to the mounting site.

The value can be entered manually in mm or set automatically via the [Auto](#) button.

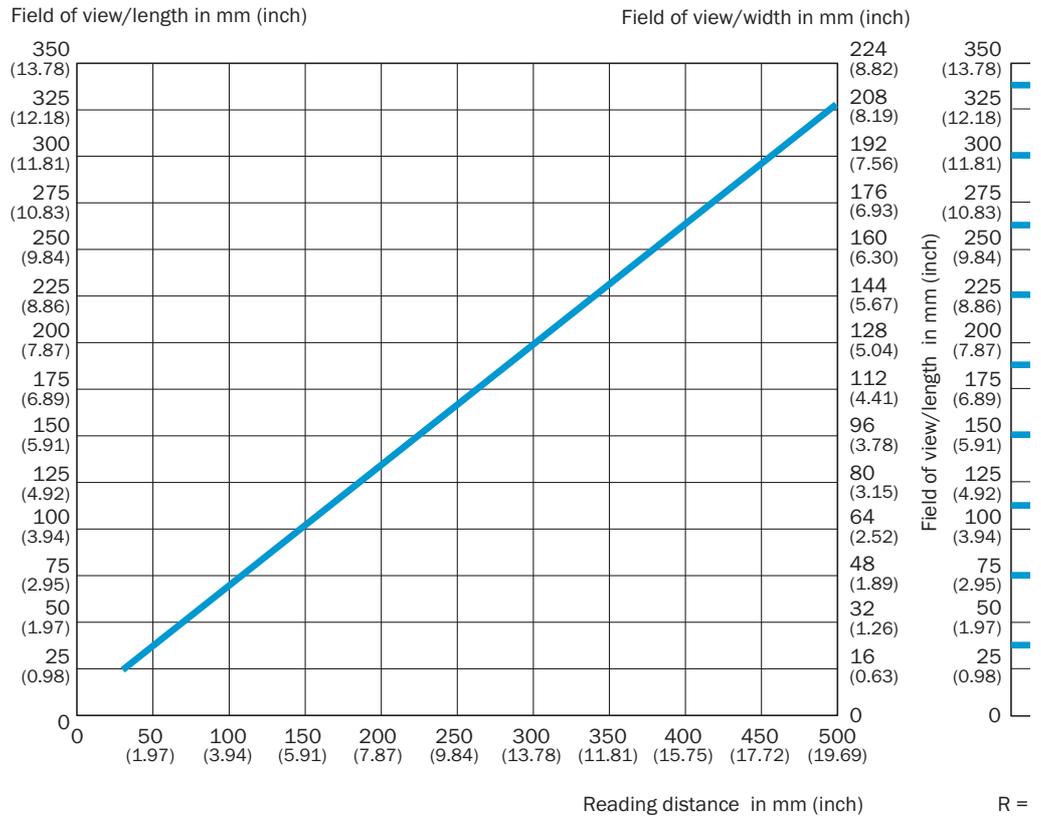


To avoid disruptive reflections on the object surface, the LECTOR®620 should be mounted on a suitable mounting bracket (see installation information).

The reading distance affects the minimum code resolution and the size of the reading area:

- If the reading distance is small, smaller codes can be read.
- A larger reading distance increases the reading area. The reduction in image brightness caused by this is compensated for by the brightness control and, if necessary, by increasing the shutter time.

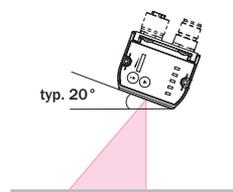
Reference values for reading distance - reading field size - resolution



Maximum reading distance with minimum resolution

Max. reading distance	Minimum resolution
50 mm (1.97 in)	0.10 mm (3.9 mil)
80 mm (3.15 in)	0.15 mm (5.9 mil)
110 mm (4.33 in)	0.20 mm (7.9 mil)
135 mm (5.32 in)	0.25 mm (9.8 mil)
165 mm (6.50 in)	0.30 mm (11.8 mil)
280 mm (11.02 in)	0.50 mm (19.7 mil)
430 mm (16.94 in)	0.75 mm (29.5 mil)

Skew angle, depending on the application



should typically be tilted by 20° out of the perpendicular in relation to the surface of the code to avoid disruptive reflections.

In the case of codes created on metal, e.g., by dot peening, an angle of between 0° (bright field light) and 45° (dark field light) may be advisable.

5.4.1.1.1 Reading distance

The *Reading distance* parameter is used to set the distance between the LECTOR®620 and the object. The reading distance is used to modify the depth of field of the LECTOR®620 at the installation location.

The value can be entered manually in mm or set automatically using the *Auto* button.

 To avoid disruptive reflections on the object surface, the LECTOR®620 should be installed at a suitable fixing bracket (see installation information).

The reading distance influences the minimum code resolution and the size of the reading area:

- If the reading distance is small, smaller codes can be read.
- A larger reading distance increases the reading area. The reduction in image brightness caused by this is compensated for by the brightness control and, if necessary, by increasing the shutter time.

5.4.1.1.2 Auto

Pressing the *Auto* button calls up the Auto Setup wizard for automatic configuration of the reading distance.

For the reading distance to be adjusted automatically, a suitable object with contrast changes or edges (e.g. code, text, or other structures) must be placed in the reading area.

 A prerequisite for automatic configuration of the reading distance is a certain basic brightness at which the structures in the image are visible.

5.4.1.2 Exposure time

The time frame for taking an image is defined in μs via the parameters for the *Exposure time* group.

- A short exposure time can accommodate high object speeds. As this causes the image brightness to decrease, it may be necessary to adjust the signal amplification (*Exposure time*). This does, however, increase the image noise.
- A long exposure time offers a high level of brightness and therefore minimal image noise. This is the optimal setting for stationary or slow-moving objects.

5.4.1.2.1 Exposure time

The time frame for taking an image is defined in μs via the *Exposure time* parameter. The exposure time can be set via the slide control.

- A short exposure time can accommodate high object speeds. As this causes the image brightness to decrease, it may be necessary to adjust the signal amplification (*Brightness*). This does, however, increase the image noise.
- A long exposure time offers a high level of brightness and therefore minimal image noise. This is the optimal setting for stationary or slow-moving objects.

5.4.1.2.2 Exposure time

The time frame for taking an image is defined in μs via the *Exposure time* parameter. The exposure time can be entered in the input field.

- A short exposure time can accommodate high object speeds. As this causes the image brightness to decrease, it may be necessary to adjust the signal amplification (*Exposure time*). This does, however, increase the image noise.
- A long exposure time offers a high level of brightness and therefore minimal image noise. This is the optimal setting for stationary or slow-moving objects.

5.4.1.3 Image settings

The image brightness and contrast settings are made via the parameters in the *Image settings* group.

A good image that displays the code clearly is essential for a reliable reading. Optimizing the image settings results in better code identification and therefore increases the reliability of the reading.

5.4.1.3.1 Brightness

The amplification of the signal is set via the *Brightness* parameter. The signal amplification can be set via the slide control.

The image brightness can be adjusted by amplifying the signal in order to achieve better visibility and decodability. When setting the signal amplification, it should be noted that an overly high setting causes significant image noise and therefore makes it difficult to read the code.



For this to be avoided, the value should be increased in stages until the code can be read successfully.

5.4.1.3.2 Brightness

The amplification of the signal is set via the *Brightness* parameter. The signal amplification value can be entered into the input field.

The image brightness can be adjusted by amplifying the signal in order to achieve better visibility and decodability. When setting the signal amplification, it should be noted that an overly high setting causes significant image noise and therefore makes it difficult to read the code.



For this to be avoided, the value should be increased in stages until the code can be read successfully.

5.4.1.3.3 Contrast

The *Contrast* parameter is used to set the gamma correction in % and adapt it to the code contrast. The gamma correction can be set via the slide control.

If the optical difference between black and white code elements is minimal, the parameter should be set to a high value until the code elements can be clearly distinguished.



To monitor the contrast, the data in the *Code infobox* (CONTRAST) can be evaluated:

- 100%: Max. contrast
- 10%: Code can already be identified

5.4.1.3.4 Contrast

The *Contrast* parameter is used to set the gamma correction in % and adapt it to the code contrast. The percentage value for the gamma correction can be entered into the input field.

If the optical difference between black and white code elements is minimal, the parameter should be set to a high value until the code elements can be clearly distinguished.

To monitor the contrast, the data in the *Code infobox* (CONTRAST) can be evaluated:

- 100%: Max. contrast
- 10%: Code can already be identified

5.4.1.3.5 Auto

Pressing the *Auto* button calls up the Auto Setup wizard for automatic configuration of the brightness and contrast.

For the settings to be adjusted automatically, a suitable object with contrast changes or edges (e.g. code, text, or other structures) must be placed in the reading area.

5.4.1.4 Illumination

The parameters of the *Illumination* group are used to deactivate and configure the illumination elements.

Optimizing the settings leads to an optimum, high-contrast, and consistently illuminated reading area. External influences caused by sun, darkness, room lighting, etc. can therefore be minimized.

5.4.1.4.1 Internal

The *Internal* parameter is used to modify the internal illumination of the LECTOR®620 in line with the application's in situ lighting situation.

- OFF: If external illumination is used, the internal illumination can be deactivated.
- LEFT/RED (type-dependent): This illumination type permits the maximum contrast for codes in the wavelength spectrum between blue and green.

- **RIGHT/BLUE** (type-dependent): This lighting type offers the maximum contrast for codes in the wavelength spectrum between green and red and can improve the reflective properties of codes on metal surfaces.
- **BOTH/RED + BLUE** (type-dependent): The combination of both illumination types offers the maximum contrast for black codes. The reading area is illuminated uniformly with increased light output. This setting is recommended for large reading distances and when the background color is unknown or subject to change.
- **BOTH/INFRARED** (type-dependent): The advantage of the infrared lighting is that the light is invisible to humans, meaning that no disruptive light sources are emitted from the device. The code contrast achieved by the lighting is dependent on the code's material properties and surface condition and must be tested.

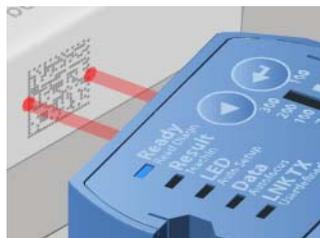


Individual parameter values are not supported by all device types.

Optimal illumination of the reading field and an increased read rate are achieved by setting the internal lighting.

5.4.1.4.2 Aiming laser

For various operating statuses, you can use the *Aiming laser* parameter to mark the optimum position for the codes in the center of the reading area directly on the object using the integrated laser pointer.



As a result, you can easily position the codes manually and achieve optimum alignment.

- **OFF**: Aiming laser is deactivated.
- **LIVE IMAGE, AUTO-SETUP**: Aiming laser is only activated to set up the LECTOR®620.
- **TRIGGER, LIVE IMAGE, AUTO-SETUP**: Aiming laser remains activated as long as the reading gate is open.



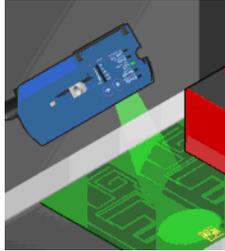
To avoid malfunctions caused by the aiming laser, deactivate it (OFF). For manual reading or if the position changes frequently, the **TRIGGER, LIVE IMAGE, AUTO-SETUP** setting is recommended.



To activate/deactivate the aiming laser using keys on the LECTOR®620, the *Function* parameter must have value **TRIGGER AND AIMING LASER**.

5.4.1.4.3 Green feedback spot

The *Green feedback spot* parameter is used to select the condition causing a green LED spot to be illuminated in the reading area. As a result, an optical notification can be set up for a specific event.



The timeframe for illumination is defined using the *Duration* parameter.

- OFF: The LED spot never illuminates.
- GOOD READ: The green LED spot illuminates if reading was successful.
- NO READ: The green LED spot illuminates if reading was unsuccessful.
- TEACH-IN 1 OK: The green LED spot illuminates if the MATCHCODE TEACH-IN 1 was read in successfully.
- TEACH-IN 1 NOK: The green LED spot illuminates if the MATCHCODE TEACH-IN 1 was not read in successfully.
- TEACH-IN 2 OK: The green LED spot illuminates if the MATCHCODE TEACH-IN 2 was read in successfully.
- TEACH-IN 2 NOK: The green LED spot illuminates if the MATCHCODE TEACH-IN 2 was not read in successfully.
- CONDITION MATCH1: The green LED spot illuminates if the read code corresponds to the matchcode of CONDITION MATCH1.
- CONDITION MULTCODES1: The green LED spot illuminates if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- CONDITION TEACHIN1: The green LED spot illuminates if the read code corresponds to the matchcode from CONDITION TEACHIN1.
- CONDITION TEACHIN2: The green LED spot illuminates if the read code corresponds to the matchcode from CONDITION TEACHIN2.

5.4.1.4.4 Duration

The *Duration* parameter is used to define the timeframe for the optical notification (*Green feedback spot*).

5.4.1.5 Image filters

Parameters for optimizing the camera image in the case of noise or special code markings are combined in the *Image filters* group.

5.4.1.5.1 Noise suppression

Image smoothing is switched on and off via the *Noise suppression* parameter to pick out the code clearly against the background.

Smoothing improves the reading properties, particularly if the image appears to be affected by noise or the code background is unstable due to a high level of amplification (*Brightness*).

Without noise



With noise



5.4.1.5.2 Dot size

You can use the *Dot size* parameter to optically enlarge the code elements in order to adapt the LECTOR®620 to hard-to-read codes (e.g. codes with gaps between the points of the L-pattern).



: difficult to read



: ideal to read

In the case of codes that were created e.g. with inkjet printers, dot peeners, or lasers, the code elements may not be connected in some cases. In this case, the readability of the codes can be improved by enlarging the code elements.



The value should be increased step by step until the code can be read successfully.

5.4.1.6 Increase performance

The parameters for the *Increase performance* group can be used to ensure the readability of the codes at higher object speeds and/or to reduce the evaluation time.

5.4.1.6.1 Setting image frequency manually

If the *Setting image frequency manually* parameter is activated, the image frequency can be specified by the user. If this parameter is deactivated, the reading device operates at maximum image frequency. The REQUEST button can be used to retrieve the current image frequency of a connected device. There may be a deviation between the set image frequency and the actual image frequency used by the reading device. The device adjusts the image frequency to other settings such as the exposure time. A long exposure time significantly reduces the image frequency.

It makes sense to reduce the image frequency for slow applications in which the object containing the code moves past the reading device slowly. Reducing the image frequency increases the time for evaluating individual images. This results in increased reading reliability for each individual image. Reduced image frequency is not suitable for fast applications, as you run the risk of not detecting the code on any image.

5.4.1.6.2 Image frequency input field

The required image frequency can be entered into the input field in Hz, number of images / s. A message appears if invalid values are entered.

5.4.1.6.3 Image rotation 180°

If the *Image rotation 180°* parameter is activated, the image is recorded rotated around 180°.

As the image is processed from the top down, codes in the top part of the image are identified earlier. By rotating the image around 180°, you can therefore reduce the evaluation time for codes that are not positioned centrally. This is particularly true for readings at a standstill.

5.4.1.6.4 Image region of interest

You can use the *Image region of interest* parameter to concentrate the reading area on the required region.

A reduction in the reading area reduces the evaluation time and increases the frame rate. This enables you to achieve higher object speeds.



To change the size of the reading area, you drag and drop the red marking or enter the four percentage values.

5.4.2 Object trigger control

The *Object trigger control* tab contains the parameters for opening and closing the reading gate.

5.4.2.1 Starting/Stopping the object trigger

The *Starting/Stopping the object trigger* group contains the parameters for opening and closing the reading gate.

To ensure that the image is taken at the correct time and the code is securely registered, the electrical properties of the digital inputs must be adjusted to the sensors and triggers that are connected.

Depending on the application, this ensures that the appropriate image region of interest is recorded and that the settings are adjusted to the reading, whether this is in motion or stationary.

5.4.2.1.1 Trigger delay

The *Trigger delay* parameter is used to choose between configuration with units of length or configuration with units of time.

- TRACK CONTROLLED: The values of the *Pause*, *Reading gate length*, *Start delay*, and *Stop delay* parameters are entered in mm. In the event of a track-controlled delay, the reading gate is not opened until the object has continued to move a predefined distance after the trigger signal.
- TIME CONTROLLED: The values of the *Pause*, *Reading gate length*, *Start delay*, and *Stop delay* parameters are entered in ms. In the event of a time-controlled delay, the reading gate is not opened until a predefined time has elapsed after the trigger signal.

5.4.2.1.2 Start delay

You can use the *Start delay* parameter to influence the time for opening the reading gate. The unit can be selected in the *Trigger delay* parameter.

A positive value causes the reading gate to be opened later, e.g. in order to only enter codes that are at the rear of an object to therefore reduce the decoding time.

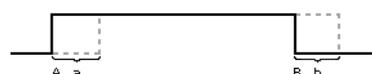
By modifying the trigger, you can adjust the image recording more accurately to the code position.



- A. START DELAY (additional time/distance in which the reading gate remains closed after event A)

5.4.2.1.3 Start by

The *Start by* parameter is used to select the signal source for opening the reading gate.



- A START BY (signal for opening reading gate)

If the reading gate is open, the LECTOR®620 records images with a frequency of 60 Hz and evaluates these by codes. Ideally, at least one code will have been completely recorded and identified on an image. If the reading gate is closed, no images are recorded.

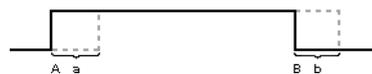
Signal sources for opening the reading gate:

- **SENSOR / RESULT 1:** A signal at digital input 1 (e.g. via a light barrier) opens the reading gate.
- **SENSOR / RESULT 2:** A signal at digital input 2 (e.g. via a light barrier) opens the reading gate.
- **EXTERNAL INPUT 1:** A signal at external digital input 1 (e.g. via a light barrier) opens the reading gate.
- **EXTERNAL INPUT 2:** A signal at external digital input 2 (e.g. via a light barrier) opens the reading gate.
- **SOPAS COMMAND:** A command to open the reading gate is expected e.g. from an external controller (PLC). (A detailed list of the command language is available from SICK on request.)
- **AUTO CYCLE:** The reading gate is opened and closed automatically after a defined cycle (using the *Pulse* and *Pause* parameters) (e.g. if the feed speed stays the same and the code position is constant or for manual code reading)
- **CAN:** The signal for opening and closing the reading gate is expected via the CAN interface. In a network of devices that communicate with each other via the CAN network, the signal for opening the reading gate can be distributed. For this, a suitable trigger signal is received by a device (usually the master) and distributed to the other CAN network.
- **USER DEFINED COMMAND:** You can freely define a command for opening the reading gate, which is e.g. generated by an external controller (PLC) and transmitted to the LECTOR®620.
- **FREE RUNNING:** The reading gate is permanently open. As a result, manual code reading is possible without an external trigger signal (e.g. from a light barrier). Signals for opening/closing the reading gate are ignored. As it is therefore not possible to define a specific point in time for data output, a signal for No READ cannot be output. The time of output and the reading speed depend on the degree of utilization of the LECTOR®620.

5.4.2.1.4 Stop delay

You can use the *Stop delay* parameter to influence the time for closing the reading gate. The unit can be selected in the *Trigger delay* parameter.

A positive value enables the reading gate to be closed later, e.g. to extend the evaluation period for codes at the rear of an object.



- **B. STOP DELAY** (additional time/distance for which the reading gate remains open after event B)

5.4.2.1.5 Stop by

The *Stop by* parameter is used to select the signal source/condition for closing the reading gate.



- B STOP BY (signal for closing reading gate)

If the reading gate is open, the LECTOR®620 records images with a frequency of 60 Hz and evaluates these by codes. Ideally, at least one code will have been completely recorded and identified on an image. If the reading gate is closed, no images are recorded.

Signal sources for closing the reading gate:

- NOT DEFINED: No function for closing the reading gate defined
- TRIGGER SOURCE: The reading gate is closed by the same signal source that opened it. (For example, opening of reading gate as soon as an object breaks a light barrier and closing of reading gate as soon as the object is no longer in front of the light barrier.)
- SENSOR / RESULT 1: A signal at digital input 1 (e.g. via a light barrier) closes the reading gate.
- SENSOR / RESULT 2: A signal at digital input 2 (e.g. via a light barrier) closes the reading gate.
- EXTERNAL INPUT 1: A signal at external digital input 1 (e.g. via a light barrier) closes the reading gate.
- EXTERNAL INPUT 2: A signal at external digital input 2 (e.g. via a light barrier) closes the reading gate.
- SOPAS COMMAND: A standard command to close the reading gate is expected e.g. from an external controller (PLC). (A detailed list of the command language is available from SICK on request.)
- TIMER/TRACKING: The reading gate remains open for a defined time/distance (*Reading gate length*). If the speed of the objects remains the same, and increment pulse is not required (*Increment source*). If the speed changes, the signals of an incremental encoder, for example, can be used to calculate the current speed. As a result, the reading gate can be closed independently of the speed of the object as soon as the object has covered a certain distance.
- GOOD READ: The reading gate is closed as soon as the GOOD READ condition has been met (e.g. the required number of codes was read). As a result, the reading gate is only open for as long as required (direct notification of whether a code was identified).
- USER DEFINED COMMAND: You can freely define a command for closing the reading gate, which is e.g. generated by an external controller (PLC) and transmitted to the LECTOR®620.
- CONDITION MATCH 1: The reading gate is closed as soon as the condition is met.
- CONDITION MULTCODES1: The reading gate is closed as soon as the number of identified codes matches the required number.
- CONDITION TEACHIN1: The reading gate is closed as soon as the *Matchcode Teach-in 1* condition is met.
- CONDITION TEACHIN2: The reading gate is closed as soon as the *Match-code Teach-in 2 (Additional)* condition is met.

5.4.2.1.6 Reading gate length

If the **TIMER/TRACKING** signal source is selected under the *Stop by* parameter, the *Reading gate length* parameter is used to enter the duration for which the reading gate stays open. The unit can be selected in the *Trigger delay* parameter.

5.4.2.1.7 Or

You can use the *Or* parameter to choose a second signal source/condition for closing the reading gate.

For the reading gate to be closed, one of the signal sources/conditions must be met.

- **NOT DEFINED:** No function for closing the reading gate defined
- **TRIGGER SOURCE:** The reading gate is closed by the same signal source that opened it. (For example, opening of reading gate as soon as an object breaks a light barrier and closing of reading gate as soon as the object is no longer in front of the light barrier.)
- **SENSOR / RESULT 1:** A signal at digital input 1 (e.g. via a light barrier) closes the reading gate.
- **SENSOR / RESULT 2:** A signal at digital input 2 (e.g. via a light barrier) closes the reading gate.
- **EXTERNAL INPUT 1:** A signal at external digital input 1 (e.g. via a light barrier) closes the reading gate.
- **EXTERNAL INPUT 2:** A signal at external digital input 2 (e.g. via a light barrier) closes the reading gate.
- **SOPAS COMMAND:** A standard command to close the reading gate is expected e.g. from an external controller (PLC). (A detailed list of the command language is available from SICK on request.)
- **TIMER/TRACKING:** The reading gate remains open for a defined time/distance (*Reading gate length*).
- **GOOD READ:** The reading gate is closed as soon as the GOOD READ condition has been met (e.g. the required number of codes was read). As a result, the reading gate is only open for as long as necessary (direct notification).
- **USER DEFINED COMMAND:** You can freely define a command for closing the reading gate, which is e.g. generated by an external controller (PLC) and transmitted to the LECTOR®620.
- **CONDITION MATCH 1:** The reading gate is closed as soon as the condition is met.
- **CONDITION MULTCODES1:** The reading gate is closed as soon as the number of identified codes matches the required number.
- **CONDITION TEACHIN1:** The reading gate is closed as soon as the *Matchcode Teach-in 1* condition is met.
- **CONDITION TEACHIN2:** The reading gate is closed as soon as the *Match-code Teach-in 2 (Additional)* condition is met.

5.4.2.1.8 Or

You can use the *Or* parameter to choose a third signal source/condition for closing the reading gate.

For the reading gate to be closed, one of the signal sources/conditions must be met.

- NOT DEFINED: No function for closing the reading gate defined
- TRIGGER SOURCE: The reading gate is closed by the same signal source that opened it. (For example, opening of reading gate as soon as an object breaks a light barrier and closing of reading gate as soon as the object is no longer in front of the light barrier.)
- SENSOR / RESULT 1: A signal at digital input 1 (e.g. via a light barrier) closes the reading gate.
- SENSOR / RESULT 2: A signal at digital input 2 (e.g. via a light barrier) closes the reading gate.
- EXTERNAL INPUT 1: A signal at external digital input 1 (e.g. via a light barrier) closes the reading gate.
- EXTERNAL INPUT 2: A signal at external digital input 2 (e.g. via a light barrier) closes the reading gate.
- SOPAS COMMAND: A standard command to close the reading gate is expected e.g. from an external controller (PLC). (A detailed list of the command language is available from SICK on request.)
- TIMER/TRACKING: The reading gate remains open for a defined time/distance (*Reading gate length*).
- GOOD READ: The reading gate is closed as soon as the GOOD READ condition has been met (e.g. the required number of codes was read). As a result, the reading gate is only open for as long as necessary (direct notification).
- USER DEFINED COMMAND: You can freely define a command for closing the reading gate, which is e.g. generated by an external controller (PLC) and transmitted to the LECTOR®620.
- CONDITION MATCH 1: The reading gate is closed as soon as the condition is met.
- CONDITION MULTCODES1: The reading gate is closed as soon as the number of identified codes matches the required number.
- CONDITION TEACHIN1: The reading gate is closed as soon as the *Matchcode Teach-in 1* condition is met.
- CONDITION TEACHIN2: The reading gate is closed as soon as the *Match-code Teach-in 2 (Additional)* condition is met.

5.4.2.1.9 Pulse

If the AUTO CYCLE signal source is selected under the *Start by* parameter, the *Pulse* parameter is used to enter the duration for which the reading gate stays open. The unit can be selected in the *Trigger delay* parameter.

5.4.2.1.10 Pause

If the AUTO CYCLE signal source is selected under the *Start by* parameter, the *Pause* parameter is used to enter the duration for which the reading gate is closed. The unit can be selected in the *Trigger delay* parameter.

5.4.2.1.11 Of

If the CAN signal source is selected under the *Start by* parameter, the *Of* parameter can be used to enter the CAN address of a device from the CAN network. This device receives a suitable trigger signal and distributes it to the other devices in the CAN network.

5.4.2.1.12 Trigger echo on

If the COMMAND signal source is selected under the *Start by* parameter, the command is sent from the LECTOR®620 as a response to the controller if the *Trigger echo on* parameter is activated.

As a result, the controller can check transmitted SOPAS commands.

5.4.2.1.13 Reading gate on

Pressing the *Reading gate on* button manually opens the reading gate.

So that you can test the current settings, all outputs, output strings and image transmission function in accordance with the configuration. The reading gate is closed in accordance with the configuration.



This function is ideally suited to testing operating mode and simulating a machine trigger with a mouse click.

5.4.2.1.14 Reading gate off

Pressing the *Reading gate off* button manually closes the reading gate.

This enables you to conclude the reading operation even if the configured condition for automatically closing the reading gate does not occur.



This function is ideally suited to testing operating mode and simulating a machine trigger with a mouse click.

5.4.2.2 Trigger distribution

The communication from the trigger signal to network users is defined in the *Trigger distribution* group.

In a network with several reading devices, it is often sufficient to use a photoelectric switch or connect a reading device to the control. The trigger signal is then distributed to the network users.

The *Trigger distribution* function allows trigger signals and the number of photoelectric switches in use to be significantly reduced. This leads to a significant reduction in system complexity and cost.

5.4.2.2.1 Distribution

The trigger signal communication is activated/deactivated and assigned to a communication network via the *Distribution* parameter.

- DEACTIVATED: The trigger is not distributed to other network users.
- CAN: In a group of devices that communicate with each other via the CAN network, the signal for opening the reading gate can be distributed. This involves an appropriate trigger signal being received by one of the devices – usually the master – and distributed to the remaining devices in the CAN network

5.5 Position

The coordinates and angles of the LECTOR®620 above the conveying system are defined on the *Position* tab. In the application, this means that objects can be clearly assigned, a defined position can be output, and the objects can be aligned in relation to the reading point.

5.5.1 Coordinates

In the *Coordinates* group, the position of the device above the conveying system is defined in relation to a neutral point. Establishing the device position makes it easier to map the position of an identified object at a later stage.

Looking in the conveying direction, the neutral point can generally be found on the bottom right-hand edge of the conveying system.

5.5.1.1 X-coordinate

The distance between the reading device and the reference point in the X-direction, i.e. along the length of the conveying direction, is entered under the *X-coordinate* parameter.

5.5.1.2 Y-coordinate

The distance between the reading device and the reference point in the Y-direction (at 90° to the conveying direction), is entered under the *Y-coordinate* parameter.

5.5.1.3 Z-coordinate

The distance between the reading device and the reference point in the Z-direction, i.e. the distance above the conveying line, is entered under the *Z-coordinate* parameter.

5.5.2 Angle

The three-dimensional position of the camera is defined in the *Angle* group.

5.5.2.1 Alpha

The device's Alpha angle in relation to the code and trigger position is defined under the *Alpha* parameter.

Example: Setting the angle to 90° or 270°. The reading is taken perpendicular to the conveying direction.

5.5.2.2 Beta

The device's Beta angle in relation to the trigger position is defined under the [Beta](#) parameter.

5.5.2.3 Gamma

The device's Gamma angle in relation to the trigger position is defined under the [Gamma](#) parameter.

5.6 Increment configuration

The [Increment configuration](#) tab contains the parameters for converting the signals for distances and speeds.

The signals are either generated by the configuration software (SOPAS) or CAN interface, or via an incremental encoder on the digital input. Incremental encoders refer to sensors for capturing positional changes (linear) or angle changes (rotating) that can capture the distance and direction or angle change and direction of rotation. They are also known as rotary encoders, incremental rotary encoders, and rotary pulse encoders.

5.6.1 Increment

The [Increment](#) group contains parameters for converting signals for distances and speeds.

The signals are either generated by the configuration software (SOPAS) or via the CAN interfaces or are generated via an incremental encoder on the digital input. Incremental encoders refer to sensors for capturing positional changes (linear) or angle changes (rotating) that can capture the distance and direction or angle change and direction or rotation. They are also known as relative encoders.

5.6.1.1 Increment source

The [Increment source](#) parameter is used to select the source of the incremental signals:

- **FIXED SPEED:** You can use the [Fixed speed](#) parameter to enter a constant value for the feed speed of the belt.
- **SENSOR 1:** The incremental signals are expected from a rotary encoder connected at digital input 1. You configure the digital input using the parameters for the [Sensor / Result 1](#) group. The maximum input frequency at the input is 300 Hz.
- **SENSOR 2:** The incremental signals are expected from a rotary encoder connected at digital input 2. You configure the digital input using the parameters for the [Sensor / Result 2](#) group. The maximum input frequency at the input is 300 Hz.
- **CAN:** The incremental signals are expected via the CAN interface.
- **SOPAS COMMAND:** Corresponding commands are expected as incremental signals. The commands can be generated by an external control (PLC), for example, and transmitted to the LECTOR®620 at defined intervals.

5.6.1.2 Fixed speed

The *Fixed speed* parameter is used to enter a constant value for the feed speed of the conveyor belt in m/sec.

This speed is automatically converted to time/distance.

5.6.1.3 System increment resolution

You use the *System increment resolution* parameter to define the conversion factor for the increment signal.

The value corresponds to the distance covered per increment in mm. For the concrete calculation of speeds and distances, the value must be adapted to the incremental encoder used.

5.7 Code configuration

The *Code configuration* tab contains the decoding settings parameters for individual code types. The code types are each assigned to a code structure.

 As only activated code types are processed, the processing speed can be increased by deactivating the code types that are not being used.

5.7.1 General

A wizard is available via the **AUTO** button in the *General* group, which can be used to set the code configuration automatically.

5.7.1.1 Automatic code configuration

Pressing the **AUTO** button (*Automatic code configuration*) starts the auto-setup wizard for automatic code-type configuration.

To perform the code configuration automatically, a suitable code must be placed in the reading area.

5.7.2 1D Symbologies

The *1D Symbologies* group contains all parameters for configuration of the 1D symbologies (barcodes).

5.7.2.1 Minimum module width

The smallest structure for decoding a code is defined in mm via the *Minimum module width* parameter.

To optimize the decoding process, the value should be adapted as closely as possible to the smallest code element (module width) that actually exists within the code. An average module width must be specified in order to read codes with varying module widths reliably.

5.7.2.2 Codabar

If the [Codabar](#) parameter is activated, decoding is activated for the code type.



As only activated code types are processed, you can increase the processing speed by specifically activating/deactivating the required code types.



- Code name: [Codabar](#) (two-value)
- Character set: 16 alphanumeric characters (10 digits, 6 special characters)
- Structure: 7 code elements (4 bars, 3 gaps); start character (A, B, C or D), code contents, stop character (A, B, C or D)
- Intrinsic reliability: Low
- Space required: Variable
- Area of application: Packet and postal service
- Standard: EN 798 bar-coding

5.7.2.3 Code 39

If the [Code 39](#) parameter is activated, decoding is activated for the code type.



As only activated code types are processed, you can increase the processing speed by specifically activating/deactivating the required code types.



- Code name: [Code 39](#) (two-value)
- Character set: 43 alphanumeric characters (10 digits, 26 letters, 7 special characters)
- Structure: 9 code elements (5 bars, 4 gaps), 3 of which wide and 6 narrow
- Intrinsic reliability: Medium
- Space required: High
- Standard: ISO/IEC 16388
- Area of application: Packet and postal service

5.7.2.4 UPC/GTIN/EAN

Activating the [UPC/GTIN/EAN](#) parameter starts the decoding process for the [UPC A](#), [UPC E](#), [GTIN 8 / EAN 8](#), and [GTIN 13 / EAN 13](#) activated code types.



As only activated code types are processed, the processing speed can be increased by specifically activating/deactivating the necessary code types.

5.7.2.5 2/5 Interleaved

If the [2/5 Interleaved](#) parameter is activated, decoding is activated for the code type.



As only activated code types are processed, you can increase the processing speed by specifically activating/deactivating the required code types.



- Code name: [2/5 Interleaved](#) or ITF (2-value)
- Character set: Numeric characters
- Structure: 5 code elements per character, 2 of which wide and 3 narrow (characters at uneven positions are represented by bars. Characters at even positions are represented by gaps.)
- Intrinsic reliability: Low
- Space required: Low (with up to 18 characters)
- Standard: ISO/IEC 16390
- Area of application: Packet and postal service

5.7.2.6 Code 93

If the [Code 93](#) parameter is activated, decoding is activated for the code type.



As only activated code types are processed, you can increase the processing speed by specifically activating/deactivating the required code types.



- Code name: [Code 93](#) (4-value)
- Character set: 43 alphanumeric characters (10 digits, 26 upper-case letters, 7 special characters: Blanks, -, ., \$, /, + and %)
- Intrinsic reliability: Medium
- Space required: Low
- Standard: ISO/IEC 16388
- Area of application: Packet and postal service

5.7.2.7 Code 128 Family

If the [Code 128 Family](#) parameter is activated, decoding is activated jointly for code types [Code 128](#) and [EAN 128](#). (Code type [Pharmacode](#) must be deactivated.)



As only activated code types are processed, you can increase the processing speed by specifically activating/deactivating the required code types.

5.7.2.8 GS1 DataBar

If the *GS1 DataBar* parameter is activated, decoding is activated for the *DataBar 14*, *DataBar Expanded*, and *DataBar Limited* active code types. This requires the *Pharmacode* code type to be deactivated.

 As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.

5.7.2.9 Pharmacode

If the *Pharmacode* parameter is activated, decoding is activated for the code type.

 All other code types are deactivated.



- Code name: *Pharmacode* (binary)
- Character set: Numeric characters (integers between 3 and 131070)
- Structure: Wide bars: 1, narrow bars: 0 (main code and additional code can have different colors)
- Intrinsic reliability: High
- Space required: Low
- Standard: Barcode standard of the company Laetus
- Area of application: Packaging check in the pharmaceutical industry (agreement with reference code from a code list)

5.7.2.10 Stacked codes

The *Stacked codes* group contains all the parameters for configuring the stacked codes (several barcodes stacked in rows one above the other, with the same start and stop character).



5.7.2.10.1 PDF417

If the *PDF417* parameter is activated, decoding is activated for the code type.

 As only activated code types are processed, you can increase the processing speed by specifically activating/deactivating the required code types.



- Code name: *PDF417*
- Character set: 2710 numeric, 1850 alphanumeric, 1400 ASCII or 1108 ISO characters

- Symbol size: 3 - 90
- Intrinsic reliability: Very high (error correction)
- Space required: Very low
- Standard: ISO/IEC 15438
- Area of application: Packet and postal service

5.7.3 Increase 1D performance

It is possible to optimize the decoding process for 1D codes via the parameters for the [Increase 1D performance](#) group. The parameters can be used to eliminate interference, thereby increasing device performance.

Adapting the settings to suit the application reduces the decoding time and increases both reading reliability and the readability of low-contrast, 1D codes.

5.7.3.1 Code contrast

To find the codes in the reading field, the image taken is searched thoroughly for structures with sufficiently large contrast variations. A code pattern is only expected within these structures.

A minimum value for these contrast variations is defined in % via the [Code contrast](#) parameter. Areas where the contrast is below the defined value are not searched for code patterns.

- To read low-contrast codes reliably regardless, a low value (e.g., 10%) must be set. This does, however, increase the decoding time, as it involves searching a larger image area.
- For high-contrast codes, the value can be increased accordingly. In this case, a correspondingly smaller image area is searched, which reduces the decoding time.

5.7.3.2 Code background

The decoding process can be limited to codes with a light or dark background color using the [Code background](#) parameter, which reduces the decoding time.

- WHITE: Only dark codes on a light background are read.
- BLACK: Only light codes on a dark background (inverted codes) are read.
- BOTH: Both dark codes on a light background and light codes on a dark background (inverted codes) are read.

5.7.4 2D code types

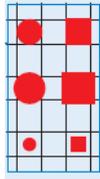
The [2D code types](#) group contains all of the parameters for configuring the 2D code.



The fewer code types are activated in parallel, the faster the decoding process.

5.7.4.1 Minimum cell size

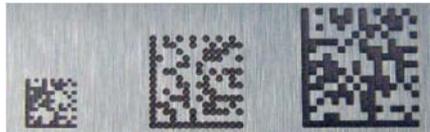
You use the *Minimum cell size* parameter to specify in mm the smallest structure of cells for decoding a code.



To optimize decoding, the value should be tailored as accurately as possible to the smallest code element (cell size) that actually occurs in the code. The cell size should be between 60 % and 105 % of the ideal cell size.



To be able to reliably read codes with different cell sizes, the value for the absolutely smallest cell size must be entered



5.7.4.2 Data matrix

If the *Data matrix* parameter is activated, decoding starts for the relevant code type.

As only activated code types are processed, the processing speed can be increased by specifically activating/deactivating the necessary code types.



- Code name: *Data matrix*
- Max. number of numeric characters: 3116
- Max. number of alphanumeric characters: 2335
- Max. number of ASCII characters: 1982
- Max. number of ISO characters: 1556
- Structure: Symbol with L pattern, rectangular symbols also possible
- Standard: ISO/IEC 16022
- Areas of application: Electronics, automotive, aviation, pharmaceuticals, and food packaging

5.7.5 Increase 2D performance

It is possible to optimize the decoding process for 2D codes via the parameters for the *Increase 2D performance* group. The parameters can be used to eliminate interference, thereby increasing device performance.

Adapting the settings to suit the application reduces the decoding time and increases both reading reliability and the readability of low-contrast, 2D codes.

5.7.5.1 Code contrast

To find the codes in the reading field, the image taken is searched thoroughly for structures with sufficiently large contrast variations. A code pattern is only expected within these structures.

A minimum value for these contrast variations is defined in % via the *Code contrast* parameter. Areas where the contrast is below the defined value are not searched for code patterns.

- To read low-contrast codes reliably regardless, a low value (e.g., 10%) must be set. This does, however, increase the decoding time, as it involves searching a larger image area.
- For high-contrast codes, the value can be increased accordingly. In this case, a correspondingly smaller image area is searched, which reduces the decoding time.

5.7.5.2 Code background

The decoding process can be limited to codes with a light or dark background color using the *Code background* parameter, which reduces the decoding time.

- WHITE: Only dark codes on a light background are read.
- BLACK: Only light codes on a dark background (inverted codes) are read.
- BOTH: Both dark codes on a light background and light codes on a dark background (inverted codes) are read.

5.7.5.3 Code alignment

The decoding process can be limited to mirrored or non-mirrored codes using the *Code alignment* parameter, which reduces the decoding time.

- NORMAL: Only non-mirrored codes can be decoded correctly.
- MIRRORED: Only mirrored codes that have been recorded from below through a glass pane, for example, can be decoded correctly. This required the images taken prior to the decoding process to be mirrored once again.
- BOTH: Both mirrored and non-mirrored codes are read.

5.7.6 Codabar

If the *Codabar* parameter is activated, decoding is activated for the corresponding code type. The configuration tab for the relevant code type can be seen and opened in SOPAS ET.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.



- Code name: *Codabar* (two-value)
- Character set: 16 alphanumeric characters (10 digits, 6 special characters)
- Structure: 7 code elements (4 bars, 3 gaps); start character (A, B, C or D), code content, stop character (A, B, C or D)
- Intrinsic safety: Low
- Spatial requirements: Variable
- Area of application: Parcel and postal service
- Standard: EN 798 bar coding

5.7.6.1 General

Reading processes such as the number of necessary multireads, regulations for dealing with the start and stop characters, or check digit tests are configured under the *General* group.

5.7.6.1.1 Multiread

You use the *Multiread* parameter to define a minimum number of GOOD READ required for a reading to be output as a GOOD READ.

By defining a minimum number of readings, you reduce the likelihood of misinterpretations for codes with poor printing quality or a low contrast, and increase reading reliability.

5.7.6.1.2 Start/Stop identical

If the *Start/Stop identical* parameter is activated, only codes whose start character is the same as the stop character are read.



A code of code type *Codabar* can have one of the four characters (A, B, C or D) as the start/stop character. If only identical start and stop characters are permitted, the reading reliability is increased, and misinterpretations are avoided.

5.7.6.1.3 Outputting start/stop

When the *Outputting start/stop* parameter is activated, the start and stop characters are output along with the code content.



Example for evaluating the reading results.

5.7.6.1.4 Check digit test

This *Check digit test* parameter is used to define the calculation method for the plausibility check on the code contents. A check digit is calculated from the code contents and compared with the check digit in the code. The selected method must match the method used during creation of the code. If the calculated check digit matches the check digit in the code, the reading is regarded as a GOOD READ).



Check digits prevent misinterpretations and increase reading reliability.

5.7.6.2 Length

Codes with a defined number of rows and columns can be excluded from the reading process via the parameters in the *Length* group. Limiting the number of codes to be read reduces the decoding time.

5.7.6.2.1 Code length

You can use the *Code length* parameter to restrict the permitted number of characters in a code.



- FREE: Codes of any code length are read.
- INTERVAL: The *Interval* parameter can be used to define the minimum and maximum code length. Only codes with a code length between these two values are read.
- FIXED: The *Fixed length* parameter is used to define several fixed code lengths. Only codes with one of these defined code lengths are read.

5.7.6.2.2 Interval

The *Interval* parameter can be used to define the minimum and maximum code length. Only codes with a code length between these two values are read.

By defining a minimum and maximum code length, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.6.2.3 Fixed length

The *Fixed length* parameter is used to define several fixed code lengths. Only codes with one of these defined code lengths are read.



By defining certain permitted code lengths, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.7 Code 39

If the *Code 39* parameter is activated, decoding is activated for the corresponding code type. The configuration tab for the corresponding code type can be seen and opened in SOPAS ET.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.



CODE39

- Code name: *Code 39* (two-value)
- Character set: 43 alphanumeric characters (10 digits, 26 letters, 7 special characters)
- Character set: 43 alphanumeric characters (10 digits, 26 letters, 7 special characters)
- Structure: 9 code elements (5 bars, 4 gaps) – 3 wide and 6 narrow
- Intrinsic safety: Average
- Spatial requirements: High
- Standard: ISO/IEC 16388
- Area of application: Parcel and postal service

5.7.7.1 General

Reading processes such as the number of necessary multireads, regulations for dealing with the start and stop characters, or check digit tests are configured under the *General* group.

5.7.7.1.1 Multiread

You use the *Multiread* parameter to define a minimum number of GOOD READ required for a reading to be output as a GOOD READ.

By defining a minimum number of readings, you reduce the likelihood of misinterpretations for codes with poor printing quality or a low contrast, and increase reading reliability.

5.7.7.1.2 Transmit start/stop

If the *Transmit start/stop* parameter is activated, the start and stop characters are transmitted together with the code contents.

As a result, the characters are available for evaluation of the reading results.

5.7.7.1.3 Full ASCII

The *Full ASCII* parameter must be activated if the code contains both digits and other characters.

5.7.7.1.4 Check digit test

This *Check digit test* parameter is used to define the calculation method for the plausibility check on the code contents. A check digit is calculated from the code contents and compared with the check digit in the code. The selected method must match the method used during creation of the code. If the calculated check digit matches the check digit in the code, the reading is regarded as a GOOD READ).



Check digits prevent misinterpretations and increase reading reliability.

5.7.7.1.5 C32 conversion

If the *C32 conversion* parameter is activated, decoding is activated for the code type.



As only activated code types are processed, you can increase the processing speed by specifically activating/deactivating the required code types.



- Code name: *C32 conversion* (two-value)
- Character set: 32 alphanumeric characters (upper case letters A - Z, digits 0 - 9, no special characters)
- Structure: Leading light zone, start character, message characters, check digit, stop character, closing light zone (13 - 16 modules per character)
- Intrinsic reliability: High
- Space required: Variable symbol length
- Standard: ISO/IEC 16388, ANSI/AIM BC5 1995
- Area of application: Article number for pharmaceutical products (Italian pharmaceutical code), drug packaging for human medicine

5.7.7.1.6 Hex – ASCII output

If the *Hex – ASCII output* parameter is activated, every code character is output as a hexadecimal value. If the parameter has deactivated the hexadecimal output, the code contents are output directly as ASCII characters. If the code contains control characters during the ASCII output, these are replaced by the @ character to avoid conflicts with the protocol framework.

Converting the code contents into hexadecimal values enables all characters – including control characters – to be output.

5.7.7.2 Length

Codes with a defined number of characters can be excluded from the reading process via the parameters in the *Length* group.

Limiting the number of codes to be read reduces the decoding time.

5.7.7.2.1 Code length

You can use the *Code length* parameter to restrict the permitted number of characters in a code.

- FREE: Codes of any code length are read.
- INTERVAL: The *Interval* parameter can be used to define the minimum and maximum code length. Only codes with a code length between these two values are read.
- FIXED: The *Fixed length* parameter is used to define several fixed code lengths. Only codes with one of these defined code lengths are read.

5.7.7.2.2 Interval

The *Interval* parameter can be used to define the minimum and maximum code length. Only codes with a code length between these two values are read.

By defining a minimum and maximum code length, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.7.2.3 Fixed length

The *Fixed length* parameter is used to define several fixed code lengths. Only codes with one of these defined code lengths are read.



By defining certain permitted code lengths, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.8 UPC/GTIN/EAN

If the *UPC/GTIN/EAN* parameter is activated, decoding is activated for the *UPC A*, *UPC E*, *GTIN 8 / EAN 8*, and *GTIN 13 / EAN 13* active code types. The configuration tab for the corresponding code types can be seen and opened in SOPAS ET.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.

5.7.8.1 General

Reading processes such as the number of necessary multireads, regulations for dealing with the start and stop characters, or check digit tests are configured under the *General* group.

5.7.8.1.1 Multiread

You use the *Multiread* parameter to define a minimum number of GOOD READ required for a reading to be output as a GOOD READ.

By defining a minimum number of readings, you reduce the likelihood of misinterpretations for codes with poor printing quality or a low contrast, and increase reading reliability.

5.7.8.1.2 Add-on

The *Add-on* parameter is used to configure how the extended code structure is used (additional information usually on magazines).



- NONE: No extended code structure for the code
- ACTIVE: Extended code structure for the code
- REQUIRED: Only codes with an extended code structure are read.

Add-on length The *Add-on length* parameter is used to define the length of the extended code structure.

- 2 DIGITS: A 1
- 5 DIGITS: A 1 B 2 C
- 2 OR 5 DIGITS: A 1 or A 1 B 2 C

5.7.8.2 UPC

The *UPC A*, *UPC E* and *UPC E extended* code types can be activated or deactivated under the *UPC* group.

5.7.8.2.1 UPC A

If the *UPC A* parameter is activated, decoding is activated for the code type.



As only activated code types are processed, you can increase the processing speed by specifically activating/deactivating the required code types.



- Code name: *UPC A* (Universal Product Code) (4-value)
- Character set: 12 numeric characters (check digit: last character)
- Structure: 2 characters for country prefix, 9 characters for manufacturer and article, 1 character as check digit

- Intrinsic reliability: High
- Space required: Low
- Standard: DIN/EN 797
- Area of application: Food and consumer goods in the USA and Canada

5.7.8.2.2 UPC E

If the *UPC E* parameter is activated, decoding is activated for the code type.



As only activated code types are processed, you can increase the processing speed by specifically activating/deactivating the required code types.



- Code name: *UPC E* (Universal Product Code) (4-value)
- Character set: 8 numeric characters
- Structure: System ID (always 0), 2 characters for country prefix, 4 characters for manufacturer and article, 1 character as check digit
- Intrinsic reliability: High
- Space required: Low
- Standard: DIN/EN 797
- Area of application: Food and consumer goods in the USA and Canada

5.7.8.2.3 UPC E extended

If the *UPC E extended* parameter is activated, 8-character codes of code type *UPC E* are automatically converted to 12-character codes. (Code type *Pharmacode* must be deactivated.)

5.7.8.3 EAN

The *GTIN 8 / EAN 8* and *GTIN 13 / EAN 13* code types can be activated or deactivated under the *EAN* group.

5.7.8.3.1 GTIN 8 / EAN 8

If the *GTIN 8 / EAN 8* parameter is activated, decoding starts for the relevant code type.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.



- Code name: *GTIN 8 / EAN 8* (European Article Numbering) (4-value)
- Character set: 8 numerical characters (check digits: last two characters)

- Structure: 2 characters for country prefix (in accordance with GS1 standards), 4 characters for the manufacturer and part (in accordance with GS1 standards), 2 characters as check digits
- Intrinsic safety: High
- Spatial requirements: Low
- Standard: DIN/EN 797
- Areas of application: Foodstuffs and consumer goods in Europe

5.7.8.3.2 GTIN 13 / EAN 13

If the *GTIN 13 / EAN 13* parameter is activated, decoding starts for the relevant code type.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.



- Code name: *GTIN 13 / EAN 13* (European Article Numbering) (4-value)
- Character set: 13 numerical characters (check digits: last two characters)
- Structure: 2 characters for country prefix (in accordance with GS1 standards), 8 characters for the manufacturer and part (in accordance with GS1 standards), 2 characters as check digits
- Intrinsic safety: High
- Spatial requirements: Low
- Standard: DIN/EN 797
- Areas of application: Foodstuffs and consumer goods in Europe

5.7.9 2/5 interleaved

If the *2/5 interleaved* parameter is activated, decoding is activated for the corresponding code type. The configuration tab for the corresponding code type can be seen and opened in SOPAS ET.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.



- Code name: *2/5 interleaved* or ITF (2-value)
- Character set: Numeric characters
- Structure: 5 code elements per character, 2 of which wide and 3 narrow (characters at uneven positions are represented by bars. Characters at even positions are represented by gaps.)
- Intrinsic safety: Low
- Spatial requirements: Low (up to 18 characters)

- Standard: ISO/IEC 16390
- Area of application: Parcel and postal service

5.7.9.1 General

Reading processes such as the number of necessary multireads, regulations for dealing with the start and stop characters, or check digit tests are configured under the [General](#) group.

5.7.9.1.1 Multiread

You use the [Multiread](#) parameter to define a minimum number of GOOD READ required for a reading to be output as a GOOD READ.

By defining a minimum number of readings, you reduce the likelihood of misinterpretations for codes with poor printing quality or a low contrast, and increase reading reliability.

5.7.9.1.2 Check digit test

This [Check digit test](#) parameter is used to define the calculation method for the plausibility check on the code contents. A check digit is calculated from the code contents and compared with the check digit in the code. The selected method must match the method used during creation of the code. If the calculated check digit matches the check digit in the code, the reading is regarded as a GOOD READ).



Check digits prevent misinterpretations and increase reading reliability. The use of a check digit is recommended for this code type.

5.7.9.1.3 Check-Digit-Test #2

[Check-Digit-Test #2](#)

5.7.9.1.4 Check-Digit-Test #3

[Check-Digit-Test #3](#)

5.7.9.1.5 Check-Digit-Test #4

[Check-Digit-Test #4](#)

5.7.9.1.6 Check-Digit-Test #5

[Check-Digit-Test #5](#)

5.7.9.2 Length

Codes with a defined number of characters can be excluded from the reading process via the parameters in the [Length](#) group.

Limiting the number of codes to be read reduces the decoding time.

5.7.9.2.1 Code length

You can use the *Code length* parameter to restrict the permitted number of characters in a code.

- FREE: Codes of any code length are read.
- INTERVAL: The *Interval* parameter can be used to define the minimum and maximum code length. Only codes with a code length between these two values are read.
- FIXED: The *Fixed length* parameter is used to define several fixed code lengths. Only codes with one of these defined code lengths are read.

5.7.9.2.2 Interval

The *Interval* parameter can be used to define the minimum and maximum code length. Only codes with a code length between these two values are read.

By defining a minimum and maximum code length, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.9.2.3 Fixed length

The *Fixed length* parameter is used to define several fixed code lengths. Only codes with one of these defined code lengths are read.



By defining certain permitted code lengths, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.10 Code 93

If the *Code 93* parameter is activated, decoding is activated for the corresponding code type. The configuration tab for the corresponding code type can be seen and opened in SOPAS ET.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.



- Code name: *Code 93* (4-value)
- Character set: 43 alphanumeric characters (10 digits, 26 upper-case letters, 7 special characters: Space, -, ., \$, /, +, and %)
- Intrinsic safety: Average
- Spatial requirements: Low
- Standard: ISO/IEC 16388
- Area of application: Parcel and postal service

5.7.10.1 General

Reading processes such as the number of necessary multireads, regulations for dealing with the start and stop characters, or check digit tests are configured under the *General* group.

5.7.10.1.1 Multiread

You use the *Multiread* parameter to define a minimum number of GOOD READ required for a reading to be output as a GOOD READ.

By defining a minimum number of readings, you reduce the likelihood of misinterpretations for codes with poor printing quality or a low contrast, and increase reading reliability.

5.7.10.2 Length

Codes with a defined number of characters can be excluded from the reading process via the parameters in the *Length* group.

Limiting the number of codes to be read reduces the decoding time.

5.7.10.2.1 Code length

You can use the *Code length* parameter to restrict the permitted number of characters in a code.

- FREE: Codes of any code length are read.
- INTERVAL: The *Interval* parameter can be used to define the minimum and maximum code length. Only codes with a code length between these two values are read.
- FIXED: The *Fixed length* parameter is used to define several fixed code lengths. Only codes with one of these defined code lengths are read.

5.7.10.2.2 Interval

The *Interval* parameter can be used to define the minimum and maximum code length. Only codes with a code length between these two values are read.

By defining a minimum and maximum code length, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.10.2.3 Fixed length

The *Fixed length* parameter is used to define several fixed code lengths. Only codes with one of these defined code lengths are read.



By defining certain permitted code lengths, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.11 Code 128 family

If the *Code 128 family* parameter is activated, decoding is activated for the *Code 128* and *EAN 128* active code types. This requires the *Pharmacode* code type to be deactivated. The configuration tab for the corresponding code types can be seen and opened in SOPAS ET.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.

5.7.11.1 General

Reading processes such as the number of necessary multireads, regulations for dealing with the start and stop characters, or check digit tests are configured under the *General* group.

5.7.11.1.1 Code 128

If the *Code 128* parameter is activated, decoding starts for the relevant code type. This requires the *Pharmacode* code type to be deactivated.



As only activated code types are processed, the processing speed can be increased by specifically activating/deactivating the necessary code types.



- Code name: *Code 128* (4-value)
- Character set: alphanumeric (ASCII) with three switchable character sets and one check digit

Character set A: Digits, upper-case letters, special characters, and ASCII control characters

Character set B: Digits, upper-case and lower-case letters, special characters and escape characters

Character set C: 100 pairs of digits (00 to 99) and FNC1 special characters

- Structure: Start symbol, useful information, check digit, stop symbol (11 modules per symbol, each with 3 bars and 3 gaps)
- Intrinsic safety: Very high (self-checking: Even number of bars, uneven number of gaps)
- Spatial requirements: Largest possible number of characters per centimeter
- Standard: ISO/IEC 15420
- Area of application: All areas, as UCC 128/EAN 128 in retail

5.7.11.1.2 EAN 128

If the *EAN 128* parameter is activated, decoding is activated for the code type. (Code type *Pharmacode* must be deactivated.)



As only activated code types are processed, you can increase the processing speed by specifically activating/deactivating the required code types.



- Code name: [EAN 128](#) (4-value)
- Character set: Alphanumeric (ASCII) with three switchable character sets
Character set A: Digits, upper case letters, and special characters
Character set B: Digits, upper and lower case letters
Character set C: Digits with double density
- Structure: FNC1, characters, check digit
- Intrinsic reliability: High
- Space required: Low
- Standard: ISO/IEC 15417
- Area of application: Marking of commodities

5.7.11.1.3 Multiread

You use the [Multiread](#) parameter to define a minimum number of GOOD READ required for a reading to be output as a GOOD READ.

By defining a minimum number of readings, you reduce the likelihood of misinterpretations for codes with poor printing quality or a low contrast, and increase reading reliability.

5.7.11.2 Length

Codes with a defined number of characters can be excluded from the reading process via the parameters in the [Length](#) group.

Limiting the number of codes to be read reduces the decoding time.

5.7.11.2.1 Code length

You can use the [Code length](#) parameter to restrict the permitted number of characters in a code.

- FREE: Codes of any code length are read.
- INTERVAL: The [Interval](#) parameter can be used to define the minimum and maximum code length. Only codes with a code length between these two values are read.
- FIXED: The [Fixed length](#) parameter is used to define several fixed code lengths. Only codes with one of these defined code lengths are read.

5.7.11.2.2 Interval

The [Interval](#) parameter can be used to define the minimum and maximum code length. Only codes with a code length between these two values are read.

By defining a minimum and maximum code length, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.11.2.3 Fixed length

The *Fixed length* parameter is used to define several fixed code lengths. Only codes with one of these defined code lengths are read.



By defining certain permitted code lengths, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.11.3 GS1/EAN 128

The control characters in the EAN 128 code format can be replaced with alternative, user-specific characters under the *GS1/EAN 128* group. Control characters in the EAN 128 code format are used at the start of the code to identify the code format and within the code to classify the data groups. The user-specific characters are entered via the keyboard or clipboard as hexadecimal values.

5.7.11.3.1 FC1-Value within code

With the parameter *FC1-Value within code* an alternative character for the FC1-character (control character at the beginning of the code) can be defined for the EAN 128 code.

Up to three characters can be defined. Enter data as a decimal character.

5.7.11.3.2 FC1-Value on first position

With the parameter *FC1-Value on first position* an alternative character for the FC1-character (control character at the beginning of the code) can be defined for the EAN 128 code.

Up to three characters can be defined. Enter data as a decimal character.

5.7.12 GS1 DataBar

If the *GS1 DataBar* parameter is activated, decoding is activated for the *DataBar 14*, *DataBar Expanded* and *DataBar Limited* active code types. This requires the *Pharmacode* code type to be deactivated. The configuration tab for the corresponding code types can be seen and opened in SOPAS ET.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.

5.7.12.1 General

Reading processes such as the number of necessary multireads, regulations for dealing with the start and stop characters, or check digit tests are configured under the *General* group.

5.7.12.1.1 DataBar 14

If the *DataBar 14* parameter is activated, decoding starts for the relevant code type.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.



- Code name: *DataBar 14*
- Character set: 14 fixed digits
- Structure: 94 modules divided into 46 elements; code words consist of 15 or 16 modules and are displayed with 4 gaps and 4 lines; the pattern has 14 modules; the lines and gaps are shown with 8 different module widths, i.e. the elements can be between 1X and 8X wide
- Intrinsic safety: High
- Standard: ISO/IEC 2427
- Area of application: Trade

5.7.12.1.2 DataBar Expanded

If the *DataBar Expanded* parameter is activated, decoding starts for the relevant code type.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.



- Code name: GS1 DataBar Expanded (GDBE)
- Character set: Max. 74 digits, max. 41 characters
- Structure: 2 corner characters: 2 modules each, both containing 1 bar and 1 gap; 4-22 data characters: 17 modules each, both containing 4 bars and 4 gaps; 2-11 patterns and 1 check digit
- Intrinsic safety: High
- Spatial requirements: Variable
- Standard: ISO/IEC 646
- Areas of application: Variable-measure foods, coupons

5.7.12.1.3 DataBar Limited

If the *DataBar Limited* parameter is activated, decoding starts for the relevant code type.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.



- Code name: GS1 DataBar Limited (GDBL)
- Character set: 14 fixed digits
- Structure: No quiet zone, the separator consists of a gap (1X) and a line (1X)
- Intrinsic safety: High
- Spatial requirements: Low
- Standard: ISO/IEC 24727
- Areas of application: Trade and industry (pharmaceuticals industry and electronics sector)

5.7.13 Pharmacode

If the *Pharmacode* parameter is activated, decoding is activated for the corresponding code type. The configuration tab for the corresponding code type can be seen and opened in SOPAS ET.



All other code types are deactivated.



- Code name: *Pharmacode* (binary)
 - Character set: Numeric characters (integers between 3 and 131,070)
- Structure: Wide bars: 1, narrow bars: 0 (main code and additional code can have different colors)
- Intrinsic safety: High
 - Spatial requirements: Low
 - Standard: Laetus bar code standard
 - Area of application: Packaging checks in the pharmaceutical industry (agreement with reference code from a code list)

5.7.13.1 General

General (PHARMA)

5.7.13.1.1 Multiread

You use the *Multiread* parameter to define a minimum number of GOOD READ required for a reading to be output as a GOOD READ.

By defining a minimum number of readings, you reduce the likelihood of misinterpretations for codes with poor printing quality or a low contrast, and increase reading reliability.

5.7.13.1.2 Module width

You can use the *Module width* parameter to choose between the automatic and user-defined module width.

User-defined module width

You use the *User-defined module width* parameter to specify in mm the smallest structure for decoding a code.

Missing start and stop characters for codes of code type *Pharmacode* mean that incorrect readings could occur for structures with a similar code. To increase reading reliability, the value should be tailored as accurately as possible to the smallest code element (module width) that actually occurs in the code.

5.7.13.1.3 Reverse

If the *Reverse* parameter is activated, the code is read in reverse direction.

As the codes of code type *Pharmacode* do not have a start or stop character and do not have a check digit at the end, the code contents differ depending on the reading direction. The reading direction must therefore be defined in the configuration.

5.7.13.2 Not calibratable

Not calibratable (PHARMA)

5.7.13.2.1 Fixed length

The *Fixed length* parameter can be used to define a required code length. Only codes with this length are read.

5.7.14 Data matrix

If the *Data matrix* parameter is activated, decoding is activated for the corresponding code type. The configuration tab for the corresponding code type can be seen and opened in SOPAS ET.

As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.



- Code name: *Data matrix*
- Max. number of numeric characters: 3116

- Max. number of alphanumeric characters: 2335
- Max. number of ASCII characters: 1982
- Max. number of ISO characters: 1556
- Structure: Symbol with L pattern, rectangular symbols also possible
- Standard: ISO/IEC 16022
- Areas of application: Electronics, automotive, aviation, pharmaceuticals, and food packaging

5.7.14.1 General

Reading processes such as the code format and symbol size are configured under the [General](#) group.

5.7.14.1.1 Code format

You can use the [Code format](#) parameter to select the format (Data Matrix Code symbology) for data output:

- ECC200: The data is output with the standard data formatting.
- GS1: The data is output with the special GS 1 data formatting. User-specific characters can be defined for the FNC1 character (control character at the start of the code) and for the separator (control character within the code). As a result, after every APPLICATION IDENTIFIER for example, a line break (CR, LF) can be inserted in the data string.

5.7.14.1.2 Max. permissible error correction

The redundancy that may be used for decoding purposes is specified as a percentage via the [Max. permissible error correction](#) parameter. With an error correction of 0 %, no redundancy is used in the code for the reading. Only good-quality codes that have been properly applied are read and identified. With an error correction of 100 %, the code detection is maximized by means of redundancy.

A high value during parameter error correction increases the read rate. Low values reduce the read rate and increase the quality of the reading results.

5.7.14.1.3 Activate ISO15415 verification

If the [Activate ISO15415 verification](#) parameter is activated, the codes are verified for compliance with the ISO/IEC 15415 standard. For this, the code contents are used to calculate values that can be output in the data strings via configuration of the output formats.



As the calculation of the values increases the decoding time, the function should only be activated if required.

5.7.14.2 Length

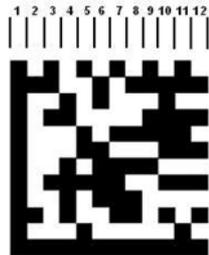
Codes with a defined number of rows and columns can be excluded from the reading process via the parameters in the [Length](#) group.

Limiting the number of codes to be read reduces the decoding time.

5.7.14.2.1 Symbol size

You can use the *Symbol size* parameter to restrict the permitted number of lines and columns of the codes.

The code dimensions are calculated from the symbol size and module size. For this, the symbol size is multiplied by the module size.



To reduce the decoding time, it may make sense to set the actual symbol size of the code and thus exactly specify the expected code size.

- FREE: All AIM/ISO/IEC-specified Data Matrix ECC200 sizes are decoded.
- FIXED: The *Fixed length 1*, *Fixed length 2* and *Fixed length 3* parameters can be used to select fixed symbol sizes. Only codes with one of these defined symbol sizes are read.

5.7.14.2.2 Fixed length 1

The *Fixed length 1* parameter can be used to select a fixed symbol size. Only codes with this symbol size are read.

By defining certain permitted symbol sizes, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.14.2.3 Fixed length 2

The *Fixed length 2* parameter can be used to select a fixed symbol size. Only codes with this symbol size are read.

By defining certain permitted symbol sizes, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.14.2.4 Fixed length 3

The *Fixed length 3* parameter can be used to select a fixed symbol size. Only codes with this symbol size are read.

By defining certain permitted symbol sizes, you restrict the range of codes to be read. This can lead to a reduction in the decoding time.

5.7.14.2.5 Fixed length 4

A fixed symbol size can be selected using the *Fixed length 4* parameter. Only codes with this symbol size are read.

Defining specific permissible symbol sizes restricts the spectrum of codes to be read. This can lead to a reduction in the decoding time.

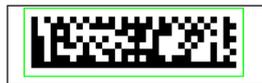
5.7.14.2.6 Fixed length 5

A fixed symbol size can be selected using the *Fixed length 5* parameter. Only codes with this symbol size are read.

Defining specific permissible symbol sizes restricts the spectrum of codes to be read. This can lead to a reduction in the decoding time.

5.7.14.2.7 Allow rectangular data fields

If the *Symbol size* parameter has value FREE, square (e.g. 16 x 16) and rectangular (e.g. 8 x 18) symbols are identified if the *Allow rectangular data fields* parameter is activated.



5.7.14.3 Reducing evaluation time

The decoding time is adapted to the application on site and optimized via the parameters for the *Reducing evaluation time* group.

5.7.14.3.1 Code surrounded by patterns

If the *Code surrounded by patterns* parameter is activated, patterns/lines (e.g. strip conductors on printed circuit board) in the code environment are ignored during evaluation.



As a result, the reading concentrates on the code elements and the decoding time is reduced.



Codes with poor quality are generally ignored if the function is activated.

5.7.14.3.2 Code surrounded by text

If the *Code surrounded by text* parameter is activated, text (e.g. on packaging) in the code environment are ignored during evaluation.



As a result, the reading concentrates on the code elements and the decoding time is reduced.



Codes with poor quality are generally ignored if the function is activated.

5.7.14.3.3 Decoding

The parameter *Decoding* defines if the reading is optimized on time or on aggressive decoding.

- FAST: Fast decoding is recommended when the code quality is good and the decoding time must be very short.
- AGGRESSIVE: Aggressive is useful when decoding time is not that relevant, e.g. stationary applications, but the code is of poor quality, e.g. low contrast, errors in the codes.
- BALANCED: Balanced meets the needs of applications in between.

5.7.14.4 Increasing robustness

The settings in the *Increasing robustness* group are used to increase the readability of codes with poor printing quality. However, these settings reduce the reading reliability.

5.7.14.4.1 Errors in L-pattern

According to the ISO/IEC 16022 standard, the code structure (L-pattern) must be marked without gaps and must extend across the complete code height and width. If the *Errors in L-pattern* parameter is active, however, codes with an incorrect code structure are also read.



During decoding, any gaps in the code structure are ignored. For the code to still be read, there must be a quiet zone of at least 5 times the cell size around the code. This may increase the decoding time.

5.7.14.5 GS1 format

The control characters in the GS1 data matrix code format can be replaced with alternative, user-specific characters under the *GS1 format* group. Control characters in the GS1 data matrix are used at the start of the code to identify the code format and within the code to classify the data groups. The user-specific characters are entered as hexadecimal values via the keyboard, clipboard, or the ADD VARIABLE OR CONSTANT button.

5.7.14.5.1 Replacing the FNC1 character

In the GS1 code format, the *Replacing the FNC1 character* parameter can be used to define user-specific characters for the FNC1 character (control character at the start of the code).

The characters are entered via the keyboard or clipboard.

It is possible to select special characters or enter hexadecimal characters via the  symbol.

Information on editing the characters is available via the  symbol.



For additional information see the F1 help.

Special characters...	Description
NUL 00H	Null
SOH 01H	Start of heading
STX 02H	Start of text
ETX 03H	End of text
EOT 04H	End of transmission
ENQ 05H	Enquiry
ACK 06H	Acknowledge
BEL 07H	Bell
BS 08H	Backspace
HT 09H	Horizontal tab
LF 0AH	NL line feed, new line
VT 0BH	Vertical tab
FF 0CH	NP from feed, new page
CR 0DH	Carriage return
SO 0EH	Shift out
SI 0FH	Shift in
DLE 10H	Data link escape
DC1 11H	Device control 1
DC2 12H	Device control 2
DC3 13H	Device control 3
DC4 14H	Device control 4
NAK 15H	Negative acknowledge
SYN 16H	Synchronous idle
ETB 17H	End of transmission block
CAN 18H	Cancel
EM 19H	End of medium
SUB 1AH	Substitute
ESC 1BH	Escape
FSP1CH	File separator
GSP 1DH	Group separator
RSP 1 EH	Record separator
USP 1FH	Unit separator
SPC 20H	Space
DEL 7FH	Delete
User defined	User defined [HEX]
Nothing	Nothing

5.7.14.5.2 Replacing the separator

In the GS1 code format, the *Replacing the separator* parameter can be used to define user-specific characters for the separator (control character at the start of the code).

The characters are entered via the keyboard or clipboard.

It is possible to select special characters or enter hexadecimal characters via the  symbol.

Information on editing the characters is available via the  symbol.



For additional information see the F1 help.

Special characters...	Description
NUL 00H	Null
SOH 01H	Start of heading
STX 02H	Start of text
ETX 03H	End of text
EOT 04H	End of transmission
ENQ 05H	Enquiry
ACK 06H	Acknowledge
BEL 07H	Bell
BS 08H	Backspace
HT 09H	Horizontal tab
LF 0AH	NL line feed, new line
VT 0BH	Vertical tab
FF 0CH	NP from feed, new page
CR 0DH	Carriage return
SO 0EH	Shift out
SI 0FH	Shift in
DLE 10H	Data link escape
DC1 11H	Device control 1
DC2 12H	Device control 2
DC3 13H	Device control 3
DC4 14H	Device control 4
NAK 15H	Negative acknowledge
SYN 16H	Synchronous idle
ETB 17H	End of transmission block
CAN 18H	Cancel
EM 19H	End of medium
SUB 1AH	Substitute
ESC 1BH	Escape
FSP1CH	File separator
GSP 1DH	Group separator
RSP 1 EH	Record separator

Special characters...	Description
USP 1FH	Unit separator
SPC 20H	Space
DEL 7FH	Delete
User defined	User defined [HEX]
Nothing	Nothing

5.7.14.5.3 Application identifier marking

If the [Application identifier marking](#) parameter is activated, the APPLICATION IDENTIFIER are output in parentheses in the output format.

As a result of the APPLICATION IDENTIFIERS the following values of the code contents are assigned to a defined property. As a result, you can display the data of a code (e.g. expiry date, batch number, date of manufacture etc.) in standardized form.

5.7.15 PDF 417

If the [PDF 417](#) parameter is activated, decoding is activated for the corresponding code type. The configuration tab for the corresponding code type can be seen and opened in SOPAS ET.



As only activated code types are processed, you can increase the processing speed by deactivating the code types that are not being used.



- Code name: [PDF 417](#)
- Character set: 2710 numeric, 1850 alphanumeric, 1400 ASCII or 1108 ISO characters
- Symbol size: 3 to 90
- Intrinsic safety: Very high (error correction)
- Spatial requirements: Very low
- Standard: ISO/IEC 15438
- Area of application: Parcel and postal service

5.7.15.1 General

[General](#) (PDF)

5.8 Data processing

The *Data processing* tab contains the parameters for defining the two output formats, as well as for filtering, sorting, and controlling the data output.

These parameters are used to define the data, conditions, time, and sequence of the output on the communication interfaces.

5.8.1 Collection of data

Data is assigned to an object in the *Collection of data* group. Using the parameters in the *Collection of data* group, object data can be used uniquely in relation to a particular object and the code content can be uniquely assigned to an object.

5.8.1.1 Timeout

The *Timeout* parameter defines the duration before data processing should be started or stopped.

5.8.2 Code summarization

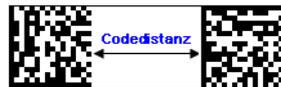
Depending on their position on the object, a distinction can be made between codes of the same type and with the same content using the parameters in the *Code summarization* group.

For this to take place, there must be a definable minimum distance (code distance) between the identical codes on the object.

5.8.2.1 Separate codes depending on position

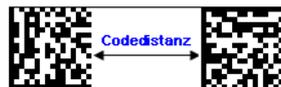
If the *Separate codes depending on position* parameter is activated, the system differentiates between codes of the same code type and with identical code contents depending on their position on the object.

The identical codes on the object must have a definable minimum distance (*Code distance*).



5.8.2.2 Code distance

The *Code distance* parameter is used to define the minimum distance for differentiating between identical codes.



The system only differentiates between codes with the same symbology and identical code contents if their distance on the object is at least the defined *Code distance*.

5.8.2.3 Separate codes dependent on sensor

Separate codes dependent on sensor

5.8.3 Output control

The *Output control* tab contains the parameters used to determine the timings for outputting the *Output format 1* and *Output format 2* data.

5.8.3.1 Output control

The *Output control* group contains the parameters used to define the timeline for data output for *Output format 1* and *Output format 2*.

5.8.3.1.1 Control

Control

5.8.3.1.2 Output time

You can use the *Output time* parameter to define when the reading results are to be output.

- END OF TRIGGER: The reading results of all codes are output as soon as the reading gate is closed. As a result, the data is output at a definable point in time.
- AS SOON AS POSSIBLE: The reading results of a code are output immediately after code identification (even if the reading gate is still open). As a result, data can immediately be processed further.
- NEW LABEL: The reading results of a code are output directly after identification of a code. However, before the same code can be output again, another code must be identified. As a result (e.g. during manual reading), you can prevent a code from being identified multiple times and the data being output again.
- END OF LABEL: The reading results of a code are output as soon as a code leaves the reading area of the LECTOR®620.

In operating mode, image transmission and diagnosis are only available for the END OF TRIGGER setting.

5.8.3.1.3 Output condition

If the AS SOON AS POSSIBLE value is selected for the *Output time* parameter, you can use the *Output condition* parameter to select the condition for data output.

- GOOD READ: The reading results are output if the reading was successful.
- CONDITION MATCH1: The reading results are output if the read code corresponds to the matchcode from CONDITION MATCH1.
- CONDITION MULTCODES1: The reading results are output if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- CONDITION TEACHIN1: The reading results are output if the read code corresponds to the matchcode from CONDITION TEACHIN1.
- CONDITION TEACHIN2: The reading results are output if the read code corresponds to the matchcode from CONDITION TEACHIN2.

5.8.3.1.4 Data output mode

The *Data output mode* parameter is used to define the frequency of data output.

- SINGLE: The reading results are output once.
- MULTIPLE: The reading results are initially output once. After a certain period of time (*Label timeout* or *Condition timeout*), the data is output again. With a suitable setting, you can differentiate between different objects with identical codes.

Label timeout If the AS SOON AS POSSIBLE value is selected for the *Output time* parameter, you can use the *Label timeout* parameter to define the duration up to renewed output of the data. During this period, the data is output again as long as the code is in the reading area.

The unit can be selected in the *Output delay* parameter.

Condition timeout If the AS SOON AS POSSIBLE value is selected for the *Output time* parameter, you can use the *Condition timeout* parameter to define the duration up to renewed output of the data. During this period, the data is output again as long as the condition for data output (*Output condition*) is met.

The unit can be selected in the *Output delay* parameter.

5.8.3.1.5 Label timeout

Label timeout

5.8.3.1.6 Condition timeout

Condition timeout

5.8.3.1.7 Delay

If the END OF TRIGGER value is selected for the *Output time* parameter, the *Delay* parameter can be used to hold back the data for a certain period after the reading gate has been closed. This allows the timings to be adapted to the control (PLC).

The unit can be selected via the *Output delay* parameter.



A long delay time slows down the image output. The image output can speed up if the delay values are low.

5.8.3.1.8 Output delay

The *Output delay* parameter is used to choose between configuration with units of length or configuration with units of time for the *Delay*, *Label timeout*, *Condition timeout*, and *Timeout* parameters

- TRACK CONTROLLED: The entries for the parameters are made in mm.
- TIME CONTROLLED: The entries for the parameters are made in ms.

5.8.3.1.9 Certain numb. of new labels

If the **NEW LABEL** value is selected for the *Output time* parameter, the *Certain numb. of new labels* parameter can be used to define the minimum number of codes (labels) that have to be read before a code is output again. This can be used to ensure that the same object is not read multiple times.

5.8.3.1.10 Label timeout active

If the *Label timeout active* parameter is activated, you can define the time period after which an identical code (label) may be output again.

5.8.3.1.11 Timeout

If the **END OF LABEL** value is selected for the *Output time* parameter, you can use the *Timeout* parameter to define the period after which a code is output again.

The unit can be selected in the *Output delay* parameter.

5.8.4 Evaluation conditions

Conditions that are checked by the device at each reading gate are defined on the *Evaluation conditions* tab. If defined conditions are fulfilled, specific actions can be performed by the device – for example, setting an output signal, ending a reading gate, or outputting an image or a particular data string.

Evaluation conditions can be used to define situations that serve as reference points for flexible process control and data processing in the device software. The use of evaluation conditions shifts the complexity of control tasks from the external controls (PLCs) into the code reader, meaning costs are reduced.

5.8.4.1 Conditions for Good Read

The *Conditions for Good Read* selection list is used to define the condition for which a reading is classed as a **GOOD READ**.

- **USE MIN./MAX. NUMBER OF CODES:** Depending on the configuration of parameters *Check min. number of valid codes* and *Check max. number of valid codes* a reading is classed as a **GOOD READ** if the number of read codes lies between the specified minimum value and the specified maximum value.
- **CONDITION MATCH1:** Reading is rated as a **GOOD READ** if the read code corresponds to **MATCHCODE MATCH 1**.
- **CONDITION MULTCODES1:** Reading is rated as a **GOOD READ** if the number of read codes corresponds to the required number.
- **CONDITION TEACHIN1:** Reading is rated as a **GOOD READ** if the read code corresponds to **MATCHCODE TEACH-IN 1**.
- **CONDITION TEACHIN2:** Reading is rated as a **GOOD READ** if the read code corresponds to **MATCHCODE TEACH-IN 2**.

5.8.4.1.1 Check min. number of valid codes

If the *Check min. number of valid codes* parameter is activated, the number of different codes is determined and compared with a specified minimum value (*Minimum*).

This enables you to check, for example, whether all required codes for an object are available.

Minimum The *Minimum* parameter is used to define the minimum number of different codes that must be identified so that a reading is classed as a GOOD READ. If the specified value is not reached, the reading is classed as invalid (NO READ).

All codes that differ from the other codes in terms of the symbology, code length, code contents, or code positions are counted.

5.8.4.1.2 Check max. number of valid codes

If the *Check max. number of valid codes* parameter is activated, the number of different codes is determined and compared with a specified maximum value (*Maximum*).

This enables you to ensure, for example, that only the required codes for an object are available.

Maximum The *Maximum* parameter is used to define the maximum number of different codes that can be identified so that a reading is classed as a GOOD READ. If the specified value is exceeded, the reading is classed as invalid (NO READ).

All codes that differ from the other codes in terms of the symbology, code length, code contents, or code positions are counted.

5.8.4.2 Evaluation conditions

Conditions that are checked by the device at each reading gate are defined in the *Evaluation conditions* group. If defined conditions are fulfilled, specific actions can be performed by the device – for example, setting an output signal, ending a reading gate, or outputting an image or a particular data string.

Evaluation conditions can be used to define situations that serve as reference points for flexible process control and data processing in the device software. The use of evaluation conditions shifts the complexity of control tasks from the external controls (PLCs) into the code reader, meaning costs are reduced.

General operation Conditions that have already been created can be edited via the  button. A dialog box opens which specifies the condition in detail.

Conditions that have been taught in can be deleted via the  button.

A further new condition can be created via the  button. A dialog box opens which allows you to accurately specify the condition. The condition type and name are accurately defined in the dialog box along with the condition itself.

5.8.5 Match code

The *Match code* tab contains all of the parameters for teaching-in match codes (set codes). Codes read during operation are compared with the contents of the match code. The correlation is displayed.



It is then necessary to ensure the prerequisites for a successful reading are met by setting the parameters correctly on the *Camera and lighting* tab.



To activate a match code that has been taught in, the corresponding condition is selected via parameter *Conditions for Good Read*. If the content of the codes that have been read does not correspond with the activated match code conditions, the reading is evaluated as unsuccessful (NO READ).

5.8.5.1 Matchcode Teach-in 1

The *Matchcode Teach-in 1* group contains the parameters for teaching in matchcode 1.



To activate a taught-in matchcode, the corresponding condition is selected e.g. for the *Conditions for Good Read* parameter.

5.8.5.1.1 Activating teach-in mode

Depending on the settings for the *Start teach-in* parameter, the *Activating teach-in mode* parameter is used to define the signal for preparing or starting the teach-in process for the match code.

The signal can, for example, originate from a key switch at one of the digital inputs or be triggered via a command.

- NOT DEFINED: Match code is not taught in.
- SENSOR 1: The match-code teach-in process is prepared and started via a signal at digital input 1 (e.g. a key switch).
- SENSOR 2: The match-code teach-in process is prepared and started via a signal at digital input 2.
- EXT. INPUT 1 (see note): The match-code teach-in process is prepared and started via a signal at digital external input 1.
- EXT. INPUT 2 (see note): The match-code teach-in process is prepared and started via a signal at digital external input 2.
- SOPAS COMMAND: The match-code read-in process is prepared and started via a corresponding command.



The number of available digital inputs can be expanded via the CDB620/CDM420 connection module in conjunction with the CMC600 parameter memory module. The purpose of the CMC parameter memory module is to act as an input expansion module and convert a digital signal into a command. External digital outputs generally respond more slowly than the internal digital inputs of the LECTOR®620.

5.8.5.1.2 Or teach-in via function buttons

If the *Or teach-in via function buttons* parameter is activated, the matchcode can also be taught in via the buttons on the LECTOR®620. No signal is required in order to prepare teach-in (*Activating teach-in mode*).

5.8.5.1.3 Teach-in stop by

Depending on the setting for the *Start teach-in* parameter, the *Teach-in stop by* parameter is used to define which signal or condition stops teach-in of the matchcode.

- TEACH-IN TRIGGER SOURCE: Teach-in is stopped by a signal (e.g. a key switch) at one of the digital inputs of the LECTOR®620.
- OBJECT TRIGGER: Teach-in stops as soon as the reading gate is closed.
- TAUGHT VALID CODE: Teach-in stops as soon as a matchcode was read successfully.

5.8.5.1.4 Teach-in condition

The *Teach-in condition* parameter defines the condition in which the newly taught-in matchcode content is stored. The selected condition is used as a basis when activating the target/actual comparison. Each code that is read is compared to the read-in code content for the target condition.

The code contents to be compared are assigned to a condition via the *Teach-in condition* parameter. The taught-in match codes can be started and activated at a later time, subject to the condition.

5.8.5.1.5 Inverting a condition

A condition can be inverted using the *Inverting a condition* parameter. Inverting a condition allows all code contents to be approved except the taught-in matchcode content.

With this function, it is possible to prevent a particular product from leaving the production line.

5.8.5.1.6 Teach-in code content

If the *Teach-in code content* parameter is activated, the code content of the matchcode is taught in and included as a condition.

This reduces the likelihood of the code being confused with other codes.

5.8.5.1.7 Teach-in code ID (type)

If the *Teach-in code ID (type)* parameter is activated, the code type of the matchcode is taught in.

This reduces the likelihood of the code being confused with other codes.

5.8.5.1.8 Teach-in code length

If the *Teach-in code length* parameter is activated, the code length of the matchcode is taught in.

This reduces the likelihood of the code being confused with other codes.

5.8.5.2 Match-code Teach-in 2 (Additional)

The *Match-code Teach-in 2 (Additional)* group contains the parameters for teaching in matchcode 2.



To activate a taught-in matchcode, the corresponding condition is selected e.g. for the *Conditions for Good Read* parameter.

5.8.5.2.1 Activate teach-in mode

Depending on the settings for the *Start teach-in* parameter, the *Activate teach-in mode* parameter is used to define the signal for preparing or starting the teach-in process for the match code.

The signal can, for example, originate from a key switch at one of the digital inputs or be triggered via a command.

- NOT DEFINED: Match code is not taught in.
- SENSOR 1: The match-code teach-in process is prepared and started via a signal at digital input 1 (e.g. a key switch).
- SENSOR 2: The match-code teach-in process is prepared and started via a signal at digital input 2.
- EXT. INPUT 1 (see note): The match-code teach-in process is prepared and started via a signal at digital external input 1.
- EXT. INPUT 2 (see note): The match-code teach-in process is prepared and started via a signal at digital external input 2.
- SOPAS COMMAND: The match-code read-in process is prepared and started via a corresponding command.



The number of available digital inputs can be expanded via the CDB620/CDM420 connection module in conjunction with the CMC600 parameter memory module. The purpose of the CMC parameter memory module is to act as an input expansion module and convert a digital signal into a command. External digital outputs generally respond more slowly than the internal digital inputs of the LECTOR®620.

5.8.5.2.2 Teach-in stop by

Depending on the setting for the *Start teach-in* parameter, the *Teach-in stop by* parameter is used to define which signal or condition stops teach-in of the matchcode.

- **TEACH-IN TRIGGER SOURCE:** Teach-in is stopped by a signal (e.g. a key switch) at one of the digital inputs of the LECTOR®620.
- **OBJECT TRIGGER:** Teach-in stops as soon as the reading gate is closed.
- **TAUGHT VALID CODE:** Teach-in stops as soon as a matchcode was read successfully.

5.8.5.2.3 Teach-in condition

The *Teach-in condition* parameter defines the condition in which the newly taught-in match code content is stored. The selected condition is used as a basis when activating the target/actual comparison. Each code that is read is compared to the read-in code content for the target condition.

The code contents to be compared are assigned to a condition via the *Teach-in condition* parameter. The taught-in match codes can be started and activated at a later time, subject to the condition.

5.8.5.2.4 Invert condition

A condition can be inverted using the *Invert condition* parameter. Inverting a condition allows all code contents to be approved except the taught-in match code content.

With this function, it is possible to prevent a particular product from leaving the production line.

5.8.5.2.5 Teach-in code content

If the *Teach-in code content* parameter is activated, the code content of the matchcode is taught in and included as a condition.

This reduces the likelihood of the code being confused with other codes.

5.8.5.2.6 Teach-in code ID (type)

If the *Teach-in code ID (type)* parameter is activated, the code type of the matchcode is taught in.

This reduces the likelihood of the code being confused with other codes.

5.8.5.2.7 Teach-in code length

If the *Teach-in code length* parameter is activated, the code length of the matchcode is taught in.

This reduces the likelihood of the code being confused with other codes.

5.8.5.3 General Match code teach-in - settings

The [General Match code teach-in - settings](#) group contains the parameters for teaching in matchcodes.

The two matchcodes are stored in the [Matchcode Teach-in 1](#) and [Match-code Teach-in 2 \(Additional\)](#) group respectively and can be subsequently modified individually.

5.8.5.3.1 Start teach-in

The [Start teach-in](#) parameter is used to select the signal that triggers automatic teach-in of a matchcode.

- **AUTOMATIC/STATIC:** The reading gate is automatically opened by a signal (e.g. a key switch). The next read code is taught in as a matchcode.
- **MACHINE TRIGGER / DYNAMIC:** Matchcode teach-in is prepared by a signal (e.g. a key switch) at one of the digital inputs of the LECTOR®620. After the reading gate has been opened (usually by a light barrier), the next read code is taught in as a matchcode.

5.8.5.3.2 Allowed code types

You use the [Allowed code types](#) parameter to define which code types are to be taught in as the matchcode.

As every code of a different code type could also be interpreted as a pharmacode, the Pharmacode code type must be selected exclusively.

- **ACTIVATED CODE TYPES:** In accordance with the code configuration, all activated code types are accepted for matchcode teach-in. To ensure that the desired code type can be read, the code configuration must first be checked.
- **ALL TYPES (EXCEPT PHARMACODE):** All code types except Pharmacode are accepted for matchcode teach-in.
- **ONLY PHARMACODE:** Only codes of code type Pharmacode are accepted for matchcode teach-in.

5.8.5.3.3 Code configuration

You use the [Code configuration](#) parameter to define whether and how the teaching in of a matchcode should affect the current code configuration.

- **DON'T CHANGE:** The current code configuration is not changed by a matchcode teach-in. All configured code types can continue to be read.
- **LIMIT TO LAST TAUGHT CODE:** The code configuration is adapted to the taught-in matchcode. The parameters for code type, code length, and code content are limited to the properties of the matchcode. As a result, only codes that match the matchcode can be read.
- **EXPAND BY LAST TAUGHT CODE:** The code configuration is adapted to the taught-in matchcode by also activating the parameters for the taught matchcode. As all other code settings remain unchanged, all configured code types can continue to be read.

5.8.5.3.4 Save permanent

If the *Save permanent* parameter is activated, the taught-in match code are kept even after the LECTOR®620 is switched off.

As the information is stored in the device's internal memory, on the Micro-SD memory card and/or parameter memory module CMC600, the save operation may be delayed.

5.8.6 Filters/Sorters for the output formatting

You can use the graphical interface on the *Filters/Sorters for the output formatting* tab to restrict (filter) data for outputting the reading results and define the sequence in which the data is output (sorted). By combining filters and sorters, you can therefore individually influence data output.

Any number of filters and sorters can be graphically arranged for this between the device and PC. The filters can also be arranged in parallel.

You can add further filters and sorters by pulling down the corresponding symbols. The filters or sorters can be copied by dragging the symbols while holding down the Ctrl pushbutton. To delete a symbol, drag it over to the recycle bin. Selected symbols can also be removed by pressing the ENTF. button.

During filtering, you can use the  button to define the variable and condition for filtering.

Example: (CL) CODELÄNGE = 10: Only codes with a code length of exactly 10 characters are output.

During sorting, you can use the  button to select the variable used to sort the reading results. The option is also available to sort the results in ascending or descending order.

Example: (CL) CODELÄNGE - AUFSTEIGEND: The codes with the shortest code length are output first.

5.8.6.1 Filters/Sorters for Output Format1

You can use the graphical interface of the *Filters/Sorters for Output Format1* group to restrict (filter) data for output of the reading results and define the sequence in which the data is output (sort). By combining filters and sorters, you can therefore individually influence data output.

Any number of filters and sorters can be graphically arranged for this between the device and PC. The filters can also be arranged in parallel.

You can add further filters and sorters by pulling down the corresponding symbols. If you move the symbols while holding down the Ctrl key, you can copy filters or sorters. You can delete the symbols by moving them to the trash. Selected symbols can also be removed by pressing the DEL. key.

During filtering, you can use the  button to define the variable and condition for filtering.

Example: (CL) CODE LENGTH = 10: Only codes with a code length of exactly 10 characters are output.

During sorting, you use the  button to select the variable used to sort the reading results. You can also choose between sorting in ascending and descending order.

Example: (CL) CODE LENGTH - ASCENDING: The codes with the shortest code length are output first.



For more information, see the F1 help.

Items	Description
BC - Code content	Code content of read code
CL - Code length	Code length of read code
X - X pos. absolute	Code distance to trigger position X in mm
Y - Y pos. absolute	Code distance to trigger position Y in mm
Z - Z pos. absolute	Code distance to trigger position Z in mm
CS - Code security	Indicates how 1D codes was read
VAL - Code valid (1 or 0)	Output whether read code is valid/ whether the set condition is fulfilled (e.g. whether number of multireads was fulfilled)
ID - Code type	Code type of read code
ON - Internal object number	Number of actual object (summed up number of scanned objectes)
CC - Code count	Number of read codes within reading gate
VCC - Number of valid codes	Number of valid codes within reading gate

More code-related items...	Description
IWA - Code ID	Code type according to alignment number (ASCII position of code ID which is a alphabetical number of the respective code type)
IN - Increment	Increment value at start of reading gate
FC - Focus position	Reading distance which is set in mm
RSD1DV - Ratio scans/decoded scans value	Ratio between "S1D - Scans auf 1D Codes" to "S1D1 - Decoded scans on 1D codes". Grading to verify the reading stability on 1D codes
VTH - Threshold of CS for code to become valid	Valid if $CS \geq VTH$ which means that number of scans = $VTH * 2$ (for 1D codes only)
S1D - Scans on 1D codes	Number of scans on 1D codes
SD1D - Decoded scans on 1D codes	Number of decoded scans on 1D codes
FCT - Code type full description	Code type, elaborated description
DI - Dimension	Indicates wheter it is a 1D code or a 2D code
X1 - Code corner 1 X	Pixel position of code corner 1X
Y1 - Code corner 1 Y	Pixel position of code corner 1Y
X2 - Code corner 2 X	Pixel position of code corner 2X
Y2 - Code corner 2 Y	Pixel position of code corner 2Y
X3 - Code corner 3 X	Pixel position of code corner 3X

More code-related items...	Description
Y3 - Code corner 3 Y	Pixel position of code corner 3Y
X4 - Code corner 4 X	Pixel position of code corner 4X
Y4 - Code corner 4 Y	Pixel position of code corner 4Y
CPX - X position relative to image origin	Code center X in pixels
CPY - Y position relative to image origin	Code center Y in pixels
AN2 - Tilt angle	Tilt angle for 1D codes and 2D codes relative to the horizontal (angular degree)
MCR - Ratio between image and multicount [in %]	Ratio between images and successful multi-reads
X2D - X 2D	Symbol size (number of modules) in X direction; for 2D codes only
Y2D - Y 2D	Symbol size (number of modules) in Y direction; for 2D codes only
MX2 - Modul X size	Average modul size in X direction in mm; for 2D codes only
MY2 - Modul Y size	Average modul size in Y direction in mm; for 2D codes only
IV2 - Inverse (boolean)	Code was read inverted
MD2 - Mirrored (boolean)	Code was read mirrored
CTG - ISO16022/ISO15415 Symbol contrast grade	Grading for symbol contrast according to ISO/IEC16022 and ISO/IEC15415. Checks the contrast between bright and dark dots in the code.
CTG - ISO16022/ISO15415 Symbol contrast value	Value for symbol contrast according to ISO/IEC16022 and ISO/IEC15415. Checks the contrast between bright and dark dots in the code.

More...	Description
PGV - ISO16022/ISO15415 Print growth value	Grading for print growth according to ISO/IEC16022 and ISO/IEC15415. Checks the difference between the ideal and the actual dot size.
PGG - ISO16022/ISO15415 Print growth grade	Value for print growth according to ISO/IEC16022 and ISO/IEC15415. Checks the difference between the ideal and the actual dot size.
ANUV - ISO16022/ISO15415 Axial non-uniformity value	Value for axial non-uniformity according to ISO/IEC16022 and ISO/IEC15415. Checks the ration between length and width of the code. If the code is stretched or compressed in length or width, it is given a poor rating for its axial non-linearity.
ANUG - ISO16022/ISO15415 Axial non-uniformity grade	Grading for axial non-uniformity according to ISO/IEC16022 and ISO/IEC15415. Checks the ration between length and width of the code. If the code is stretched or compressed in length or width, it is given a poor rating for its axial non-linearity.
UECV - ISO16022/ISO15415 Unused error correction value	Value for unused error correction according to ISO/IEC16022 and ISO/IEC15415. Checks how much redundant data had to be used during reading to decode the data content. The best grade is achieved if redundancy was not required at all.

More...	Description
UECG - ISO16022/ISO15415 Unused error correction grade	Grading for unused error correction according to ISO/IEC16022 and ISO/IEC15415. Checks how much redundant data had to be used during reading to decode the data content. The best grade is achieved if redundancy was not required at all.
OSG - Overall symbol grade	Overall symbol grade according to ISO/IEC16022 and ISO/IEC15415. This grading can be considered as a summary of the criteria. The poorest of all the criteria used is always output.
GNUV - ISO15415 Grid non-uniformity value	Grading for grid non-uniformity according to ISO/IEC15415. A grid is placed over the code using the alternating pattern to locate the scan points for decoding. Grid non-uniformity checks the extent to which the grid deviates from the ideal grid.
GNUG - ISO15415 Grid non-uniformity grade	Value for grid non-uniformity according to ISO/IEC15415. A grid is placed over the code using the alternating pattern to locate the scan points for decoding. Grid non-uniformity checks the extent to which the grid deviates from the ideal grid.
MODV - ISO15415 Modulation value	Value for modulation according to ISO/IEC15415. Checks uniformity of the reflectance of dark and light dots.
MODG - ISO15415 Modulation grade	Grading for modulation according to ISO/IEC15415. Checks uniformity of the reflectance of dark and light dots.
FPDV - ISO15415 Fixed pattern damage value	Value for fixed pattern damage according to ISO/IEC15415. Checks the fundamental characteristics of the code (quiet zone, finder, alternating patterns and reference dots) for defects and calculates an average.
FPDG - ISO15415 Fixed pattern damage grade	Grading for fixed pattern damage according to ISO/IEC15415. Checks the fundamental characteristics of the code (quiet zone, finder, alternating patterns and reference dots) for defects and calculates an average.
EC - ECC level	Specific DMX quality value
ECCW - ECC code words	Number of read code modules for Reed-Solomon-Error correction
DCW - Data code words	Number of code data modules
NEC - Num. ECC correctables	Number of correctable code modules in case of damage
NEE - Num. of ECC errors	Number of incorrect code modules
NES - Num. of ECC erasures	Number of incorrect code modules with known position
MC - Multi Count.	Indicates how often a code was read within a reading gate (Codes are not displayed separately!)
DLSZ - Number of devices	Number of connected devices
DLCS - Device list separated by comma	List of connected devices separated by comma
DLBS - Device list separated by blank	List of connected devices separated by blank

More...	Description
DN - Device number	Device number alternatively parameterized in field bus; default: number is identical to "Device ID"
DIN - Increment read from device	Increment read by device
SID - Symbol ID	Symbol ID

More obj. related items...	Description
FIX1 - Fix defined string 1	User defined string 1
FIX2 - Fix defined string 2	User defined string 2
FIX3 - Fix defined string 3	User defined string 3
TT - Reading gate duration in ms	Reading gate duration in ms
OTL - Trigger length in mm	Trigger length in mm
NZ - Nr. of mismatches	Number of mismatches (codes which do not fulfil the matchcode condition)
NY - Nr. of no matches	Number of objects without any match
GNC - Global nr. of codes	Number of codes
NVRB - Global nr. valid released codes	Number of valid released codes
NC - Number of reading gates	Number of reading gates
DID - Device ID	Device ID
SCGR - Superordinate counter Good Read	Superordinate Good Read counter; remains in case of a new parameterization (reset via power cycle)
SCNR - Superordinate counter No Read	Superordinate No Read counter; remains in case of a new parameterization (reset via power cycle)
DEVN - Device name	Device name
APC01 - Application counter 1	Definable counter 1
APC02 - Application counter 2	Definable counter 2
APC03 - Application counter 3	Definable counter 3
APC04 - Application counter 4	Definable counter 4
APC05 - Application counter 5	Definable counter 5
APC06 - Application counter 6	Definable counter 6
APC07 - Application counter 7	Definable counter 7
APC08 - Application counter 8	Definable counter 8
APC09 - Application counter 9	Definable counter 9
APC10 - Application counter 10	Definable counter 10
TS - Time stamp	Time stamp of reading gate (start reading gate)
RR - Read rate	Ration between Good Reads (SCGR) and number of reading gates
OGA - Distance between two objects	Object to previous object in mm

Match counter...	Description
Match1	Counter how often condition Match1 was fulfilled (object related)
TeachIn1	Counter how often condition TeachIn1 was fulfilled (object related)
TeachIn2	Counter how often condition TeachIn2 was fulfilled (object related)

Superordinate counter....	Description
Match1	Counter how often condition Match1 was fulfilled
TeachIn1	Counter how often condition TeachIn1 was fulfilled
TeachIn2	Counter how often condition TeachIn2 was fulfilled

Special character...	Description
NUL 00H	Null
SOH 01H	Start of heading
STX 02H	Start of text
ETX 03H	End of text
EOT 04H	End of transmission
ENQ 05H	Enquiry
ACK 06H	Acknowledge
BEL 07H	Bell
BS 08H	Backspace
HT 09H	Horizontal tab
LF 0AH	NL line feed, new line
VT 0BH	Vertical tab
FF 0CH	NP from feed, new page
CR 0DH	Carriage return
SO 0EH	Shift out
SI 0FH	Shift in
DLE 10H	Data link escape
DC1 11H	Device control 1
DC2 12H	Device control 2
DC3 13H	Device control 3
DC4 14H	Device control 4
NAK 15H	Negative acknowledge
SYN 16H	Synchronous idle
ETB 17H	End of transmission block
CAN 18H	Cancel
EM 19H	End of medium
SUB 1AH	Substitute
ESC 1BH	Escape
FSP1CH	File separator
GSP 1DH	Group separator
RSP 1EH	Record separator
USP 1FH	Unit separator
SPC 20H	Space
DEL 7FH	Delete
User defined	User defined [HEX]
Nothing	Nothing

5.8.6.2 Filters/Sorters for Output Format2

You can use the graphical interface of the *Filters/Sorters for Output Format2* group to restrict (filter) data for output of the reading results and define the sequence in which the data is output (sort). By combining filters and sorters, you can therefore individually influence data output.

Any number of filters and sorters can be graphically arranged for this between the device and PC. The filters can also be arranged in parallel.

You can add further filters and sorters by pulling down the corresponding symbols. If you move the symbols while holding down the Ctrl key, you can copy filters or sorters. You can delete the symbols by moving them to the trash. Selected symbols can also be removed by pressing the DEL. key.

During filtering, you can use the  button to define the variable and condition for filtering.

Example: (CL) CODE LENGTH = 10: Only codes with a code length of exactly 10 characters are output.

During sorting, you use the  button to select the variable used to sort the reading results. You can also choose between sorting in ascending and descending order.

Example: (CL) CODE LENGTH - ASCENDING: The codes with the shortest code length are output first.



For more information, see the F1 help.

Items	Description
BC - Code content	Code content of read code
CL - Code length	Code length of read code
X - X pos. absolute	Code distance to trigger position X in mm
Y - Y pos. absolute	Code distance to trigger position Y in mm
Z - Z pos. absolute	Code distance to trigger position Z in mm
CS - Code security	Indicates how 1D codes was read
VAL - Code valid (1 or 0)	Output whether read code is valid/ whether the set condition is fulfilled (e.g. whether number of multireads was fulfilled)
ID - Code type	Code type of read code
ON - Internal object number	Number of actual object (summed up number of scanned objectes)
CC - Code count	Number of read codes within reading gate
VCC - Number of valid codes	Number of valid codes within reading gate

More code-related items...	Description
IWA - Code ID	Code type according to alignment number (ASCII position of code ID which is a alphabetical number of the respective code type)

More code-related items...	Description
IN - Increment	Increment value at start of reading gate
FC - Focus position	Reading distance which is set in mm
RSD1DV - Ratio scans/decoded scans value	Ratio between "S1D - Scans auf 1D Codes" to "S1D1 - Decoded scans on 1D codes". Grading to verify the reading stability on 1D codes
VTH - Threshold of CS for code to become valid	Valid if $CS \geq VTH$ which means that number of scans = $VTH * 2$ (for 1D codes only)
S1D - Scans on 1D codes	Number of scans on 1D codes
SD1D - Decoded scans on 1D codes	Number of decoded scans on 1D codes
FCT - Code type full description	Code type, elaborated description
DI - Dimension	Indicates whether it is a 1D code or a 2D code
X1 - Code corner 1 X	Pixel position of code corner 1X
Y1 - Code corner 1 Y	Pixel position of code corner 1Y
X2 - Code corner 2 X	Pixel position of code corner 2X
Y2 - Code corner 2 Y	Pixel position of code corner 2Y
X3 - Code corner 3 X	Pixel position of code corner 3X
Y3 - Code corner 3 Y	Pixel position of code corner 3Y
X4 - Code corner 4 X	Pixel position of code corner 4X
Y4 - Code corner 4 Y	Pixel position of code corner 4Y
CPX - X position relative to image origin	Code center X in pixels
CPY - Y position relative to image origin	Code center Y in pixels
AN2 - Tilt angle	Tilt angle for 1D codes and 2D codes relative to the horizontal (angular degree)
MCR - Ratio between image and multicount [in %]	Ratio between images and successful multi-reads
X2D - X 2D	Symbol size (number of modules) in X direction; for 2D codes only
Y2D - Y 2D	Symbol size (number of modules) in Y direction; for 2D codes only
MX2 - Modul X size	Average modul size in X direction in mm; for 2D codes only
MY2 - Modul Y size	Average modul size in Y direction in mm; for 2D codes only
IV2 - Inverse (boolean)	Code was read inverted
MD2 - Mirrored (boolean)	Code was read mirrored
CTG - ISO16022/ISO15415 Symbol contrast grade	Grading for symbol contrast according to ISO/IEC16022 and ISO/IEC15415. Checks the contrast between bright and dark dots in the code.
CTG - ISO16022/ISO15415 Symbol contrast value	Value for symbol contrast according to ISO/IEC16022 and ISO/IEC15415. Checks the contrast between bright and dark dots in the code.

More...	Description
PGV - ISO16022/ISO15415 Print growth value	Grading for print growth according to ISO/IEC16022 and ISO/IEC15415. Checks the difference between the ideal and the actual dot size.
PGG - ISO16022/ISO15415 Print growth grade	Value for print growth according to ISO/IEC16022 and ISO/IEC15415. Checks the difference between the ideal and the actual dot size.

More...	Description
ANUV - ISO16022/ISO15415 Axial non-uniformity value	Value for axial non-uniformity according to ISO/IEC16022 and ISO/IEC15415. Checks the ration between length and width of the code. If the code is stretched or compressed in length or width, it is given a poor rating for its axial non-linearity.
ANUG - ISO16022/ISO15415 Axial non-uniformity grade	Grading for axial non-uniformity according to ISO/IEC16022 and ISO/IEC15415. Checks the ration between length and width of the code. If the code is stretched or compressed in length or width, it is given a poor rating for its axial non-linearity.
UECV - ISO16022/ISO15415 Unused error correction value	Value for unused error correction according to ISO/IEC16022 and ISO/IEC15415. Checks how much redundant data had to be used during reading to decode the data content. The best grade is achieved if redundancy was not required at all.
UECG - ISO16022/ISO15415 Unused error correction grade	Grading for unused error correction according to ISO/IEC16022 and ISO/IEC15415. Checks how much redundant data had to be used during reading to decode the data content. The best grade is achieved if redundancy was not required at all.
OSG - Overall symbol grade	Overall symbol grade according to ISO/IEC16022 and ISO/IEC15415. This grading can be considered as a summary of the criteria. The poorest of all the criteria used is always output.
GNUV - ISO15415 Grid non-uniformity value	Grading for grid non-uniformity according to ISO/IEC15415. A grid is placed over the code using the alternating pattern to locate the scan points for decoding. Grid non-uniformity checks the extent to which the grid deviates from the ideal grid.
GNUG - ISO15415 Grid non-uniformity grade	Value for grid non-uniformity according to ISO/IEC15415. A grid is placed over the code using the alternating pattern to locate the scan points for decoding. Grid non-uniformity checks the extent to which the grid deviates from the ideal grid.
MODV - ISO15415 Modulation value	Value for modulation according to ISO/IEC15415. Checks uniformity of the reflectance of dark and light dots.
MODG - ISO15415 Modulation grade	Grading for modulation according to ISO/IEC15415. Checks uniformity of the reflectance of dark and light dots.
FPDV - ISO15415 Fixed pattern damage value	Value for fixed pattern damage according to ISO/IEC15415. Checks the fundamental characteristics of the code (quiet zone, finder, alternating patterns and reference dots) for defects and calculates an average.
FPDG - ISO15415 Fixed pattern damage grade	Grading for fixed pattern damage according to ISO/IEC15415. Checks the fundamental characteristics of the code (quiet zone, finder, alternating patterns and reference dots) for defects and calculates an average.
EC - ECC level	Specific DMX quality value

More...	Description
ECCW - ECC code words	Number of read code modules for Reed-Solomon-Error correction
DCW - Data code words	Number of code data modules
NEC - Num. ECC correctables	Number of correctable code modules in case of damage
NEE - Num. of ECC errors	Number of incorrect code modules
NES - Num. of ECC erasures	Number of incorrect code modules with known position
MC - Multi Count.	Indicates how often a code was read within a reading gate (Codes are not displayed separately!)
DLSZ - Number of devices	Number of connected devices
DLCS - Device list separated by comma	List of connected devices separated by comma
DLBS - Device list separated by blank	List of connected devices separated by blank
DN - Device number	Device number alternatively parameterized in field bus; default: number is identical to "Device ID"
DIN - Increment read from device	Increment read by device
SID - Symbol ID	Symbol ID

More obj. related items...	Description
FIX1 - Fix defined string 1	User defined string 1
FIX2 - Fix defined string 2	User defined string 2
FIX3 - Fix defined string 3	User defined string 3
TT - Reading gate duration in ms	Reading gate duration in ms
OTL - Trigger length in mm	Trigger length in mm
NZ - Nr. of mismatches	Number of mismatches (codes which do not fulfil the matchcode condition)
NY - Nr. of no matches	Number of objects without any match
GNC - Global nr. of codes	Number of codes
NVRB - Global nr. valid released codes	Number of valid released codes
NC - Number of reading gates	Number of reading gates
DID - Device ID	Device ID
SCGR - Superordinate counter Good Read	Superordinate Good Read counter; remains in case of a new parameterization (reset via power cycle)
SCNR - Superordinate counter No Read	Superordinate No Read counter; remains in case of a new parameterization (reset via power cycle)
DEVN - Device name	Device name
APC01 - Application counter 1	Definable counter 1
APC02 - Application counter 2	Definable counter 2
APC03 - Application counter 3	Definable counter 3
APC04 - Application counter 4	Definable counter 4
APC05 - Application counter 5	Definable counter 5
APC06 - Application counter 6	Definable counter 6
APC07 - Application counter 7	Definable counter 7
APC08 - Application counter 8	Definable counter 8

More obj. related items...	Description
APC09 - Application counter 9	Definable counter 9
APC10 - Application counter 10	Definable counter 10
TS - Time stamp	Time stamp of reading gate (start reading gate)
RR - Read rate	Ration between Good Reads (SCGR) and number of reading gates
OGA - Distance between two objects	Object to previous object in mm

Match counter...	Description
Match1	Counter how often condition Match1 was fulfilled (object related)
TeachIn1	Counter how often condition TeachIn1 was fulfilled (object related)
TeachIn2	Counter how often condition TeachIn2 was fulfilled (object related)

Superordinate counter...	Description
Match1	Counter how often condition Match1 was fulfilled
TeachIn1	Counter how often condition TeachIn1 was fulfilled
TeachIn2	Counter how often condition TeachIn2 was fulfilled

Special character...	Description
NUL 00H	Null
SOH 01H	Start of heading
STX 02H	Start of text
ETX 03H	End of text
EOT 04H	End of transmission
ENQ 05H	Enquiry
ACK 06H	Acknowledge
BEL 07H	Bell
BS 08H	Backspace
HT 09H	Horizontal tab
LF 0AH	NL line feed, new line
VT 0BH	Vertical tab
FF 0CH	NP from feed, new page
CR 0DH	Carriage return
SO 0EH	Shift out
SI 0FH	Shift in
DLE 10H	Data link escape
DC1 11H	Device control 1
DC2 12H	Device control 2
DC3 13H	Device control 3
DC4 14H	Device control 4
NAK 15H	Negative acknowledge

Special character...	Description
SYN 16H	Synchronous idle
ETB 17H	End of transmission block
CAN 18H	Cancel
EM 19H	End of medium
SUB 1AH	Substitute
ESC 1BH	Escape
FSP1CH	File separator
GSP 1DH	Group separator
RSP 1 EH	Record separator
USP 1FH	Unit separator
SPC 20H	Space
DEL 7FH	Delete
User defined	User defined [HEX]
Nothing	Nothing

5.8.7 Output format

The format of the data strings for outputting the reading results are defined on the [Output format](#) tab.

The flexibility of the output format allows additional information to be added relating to the code and its position, the reading gate, and the individual reading results.

5.8.7.1 Output format 1

The format of the data string for outputting the reading results is defined on the graphical interface of the [Output format 1](#) group. A total of two different formats ([Output format 1](#) and [Output format 2](#)) can be defined. During configuration of the interfaces, one of the two output formats can be assigned.

The output format is graphically displayed and can contain conditions, special characters (orange), variables (blue), or free text. To individually modify the output format, you select the required insertion position with the mouse. The entries are made using the keyboard or special buttons:

You can use the  button or the context menu (right mouse button) to insert, for example, conditions, special characters (e.g., start and stop), or reading result variables.



For additional information see the F1 help.

You can use the  button to modify the properties of a condition. For each condition, you can define which data should be output if the condition is met. You can also define data for the case in which the condition is not met. Conditions can be interleaved.

To check the output format, you can display the reading results on the terminal. To do this, you open the terminal using the  button in the toolbar and establish a connection with the

device (CONNECTION menu on the terminal). For each trigger (reading gate), a data string is transferred and displayed in the terminal.

Data	Description
BC – Code content	Content of the code that has been read
CL – Code length	Length of the code that has been read
X – X pos. absolute	Code distance to trigger position X in mm
Y – Y pos. absolute	Code distance to trigger position Y in mm
Z – Z pos. absolute	Code distance to trigger position Z in mm
CS – Code security	Indicates how a 1D code was read
VAL – Code valid (1 or 0)	Outputs whether the code that has been read is valid, i.e., whether the configured conditions have been met, e.g., queries whether number of multireads has been reached
ID – Code type	Code type of code that has been read
ON – Object number	Number of the current object (total number of scanned objects)
CC – Code count	Number of codes that have been read within reading gate
VCC – Number of valid codes	Number of valid codes within reading gate

More code-related data...	Description
IWA – Code ID	Code type designated according to alignment number (ASCII position of the code ID, i.e., alphabetical numbering of the code types)
IN – Increment	Increment value at the start of the reading gate
FC – Focus position	Reading distance setting in mm
RSD1DV – Ratio between scans and decoded scans	Value that indicates the ratio between "S1D – Number of scans completed on a 1D code" and "SD1D – Number of scans completed successfully on a 1D code"; grading for evaluating the reading stability of 1D codes.
VTH – CS threshold for code to become valid	Valid when $CS \geq VTH$, i.e. number of scans = $VTH * 2$ (for 1D codes only)
S1D – Scans on 1D codes	Number of scans completed on 1D codes
SD1D – Decoded scans on 1D codes	Number of successful scans completed on 1D codes
FCT – Full code type description	Code type, detailed description
DI – Dimension	Indicates whether the code is 1D or 2D
X1 – Code corner 1 X	Pixel position of code corner 1X
Y1 – Code corner 1 Y	Pixel position of code corner 2Y
X2 – Code corner 2 X	Pixel position of code corner 2X
Y2 – Code corner 2 Y	Pixel position of code corner 2Y
X3 – Code corner 3 X	Pixel position of code corner 3X
Y3 – code corner 3 Y	Pixel position of code corner 3Y
X4 – code corner 4 X	Pixel position of code corner 4X
Y4 – code corner 4 Y	Pixel position of code corner 4Y
CPX – X position relative to image origin	Code center X in pixels
CPY – Y position relative to image origin	Code center Y in pixels

More code-related data...	Description
AN2 – Tilt angle	Tilt angle for 1D and 2D codes [in degrees] relative to the horizontal
MCR – Ratio between image and MC [in %]	Relationship between images that have been taken and successful multireads
X2D – X 2D	Symbol size (number of modules) in X direction; for 2D codes only
Y2D – Y 2D	Symbol size (number of modules) in Y direction; for 2D codes only
MX2 – Module X size	Average module size in X direction in mm; for 2D codes only
MY2 – Module Y size	Average module size in Y direction in mm; for 2D codes only
IV2 – Inverse (boolean)	Code was read inversely
MD2 – Mirrored (boolean)	Code was read as a mirror-image
CTG – ISO 16022/ISO15415 Symbol contrast grade	Grading for symbol contrast in accordance with ISO/IEC16022 and ISO/IEC15415. Checks the contrast between dark and light dots in the code.
CTV – ISO16022/ISO15415 Symbol contrast value	Value for symbol contrast in accordance with ISO/IEC16022 and ISO/IEC15415. Checks the contrast between dark and light dots in the code.

More...	Description
PGG – ISO16022/ISO15415 Print growth grade	Grading for print growth in accordance with ISO/IEC16022 and ISO/IEC15415. Checks the difference between the ideal and actual dot size. The value is output in accordance with the values defined in ISO/IEC16022 and ISO/IEC15415.
PGV – ISO16022/ISO15415 Print growth value	Value for print growth in accordance with ISO/IEC16022 and ISO/IEC15415. Checks the difference between the ideal and actual dot size. The value is output in accordance with the values defined in ISO/IEC16022 and ISO/IEC15415.
ANUV – ISO16022/ISO15415 Axial non-uniformity value	Value for axial non-uniformity in accordance with ISO/IEC16022 and ISO/IEC15415. Checks the ratio between the code length and width. If the code is stretched or compressed in terms of its length or width, it is given a poor rating for its axial non-uniformity.
ANUG – ISO16022/ISO15415 Axial non-uniformity grade	Grading for axial non-uniformity in accordance with ISO/IEC16022 and ISO/IEC15415. Checks the ratio between the code length and width. If the code is stretched or compressed in terms of its length or width, it is given a poor rating for its axial non-uniformity.
UECV – ISO16022/ISO15415 Unused error correction value	Value for unused error correction in accordance with ISO/IEC16022 and ISO/IEC15415. Checks how much redundant data had to be used during the reading to decode the data content. The best grade is achieved if redundancy was not required at all.

LECTOR®620

More...	Description
UECG – ISO16022/ISO15415 Unused error correction grade	Grading for unused error correction in accordance with ISO/IEC16022 and ISO/IEC15415. Checks how much redundant data had to be used during the reading to decode the data content. The best grade is achieved if redundancy was not required at all.
OSG – Overall symbol grade	Overall symbol grade in accordance with ISO16022 and ISO15415. This grading can be considered as a summary of the criteria. The poorest of all the criteria used is always output.
GNUV – ISO16022/ISO15415 Grid non-uniformity value	Grading for grid non-linearity in accordance with ISO/IEC16022 and ISO/IEC15415. A grid is placed over the code using the alternating pattern to locate the scan points for decoding. Grid non-uniformity checks the extent to which the grid deviates from the ideal grid and issues this in the form of a value.
GNUG – ISO16022/ISO15415 Grid non-uniformity grading	Value for grid non-uniformity in accordance with ISO/IEC15415. A grid is placed over the code using the alternating pattern to locate the scan points for decoding. Grid non-uniformity checks the extent to which the grid deviates from the ideal grid and issues this in the form of a grading.
MODV – ISO15415 Modulation value	Value for modulation in accordance with ISO/IEC15415. Checks reflective uniformity of dark and light dots and issues this in the form of a value.
MODG – ISO15415 Modulation grade	Grading for modulation in accordance with ISO/IEC15415. Checks the reflective uniformity of dark and light dots and issues this in the form of a grading.
FPDV – ISO15415 Fixed pattern damage value	Value for fixed pattern damage in accordance with ISO/IEC15415. Checks the essential features of the code (quiet zone, finder, alternating patterns, and reference dots) for defects and calculates an average value.
FPDG – ISO15415 Fixed pattern damage grade	Grading for fixed pattern damage in accordance with ISO/IEC15415. Checks the essential features of the code (quiet zone, finder, alternating patterns, and reference dots) for defects and calculates an average grading.
EC – ECC level	Specific DMX quality value
ECCW – ECC code words	Number of redundant code modules for Reed-Solomon error correction
DCW – Data code words	Number of code data modules
NEC – Number of ECC correctables	Number of correctable code modules in case of damage
NEE – Number of ECC errors	Number of incorrect code modules
NES – number of ECC erasures	Number of incorrect code modules with known position
MC – Multiread count	Indicates how often a code was read within a reading gate (codes are not displayed separately!)
DLSZ – Number of devices	Number of connected devices

More...	Description
DLCS – Device list separated by comma	List of all connected devices separated by a comma
DLBS – Device list separated by blank	List of all connected devices separated by a blank space
DN – Device number	Device number alternatively configured in the fieldbus; number is identical to device ID by default
DIN – Increment read by device	Increment read by device
SID – Symbol ID	Symbol ID

More object-related data...	Description
FIX1 – Fix defined string 1	User-defined string 1
FIX2 – Fix defined string 2	User-defined string 2
FIX3 – Fix defined string 3	User-defined string 3
TT – Reading gate duration in ms	Reading gate duration in ms
OTL – Trigger length in mm	Trigger length in mm
NZ – Number of mismatches	Number of mismatches (codes which do not fulfill the match code condition)
NY – Number of no matches	Number of objects without any match
GNC – Global number of codes	Number of codes
NVRB – Global number of valid released codes	Number of valid released codes
NC – Number of reading gates	Number of reading gates
DID – Device ID	Device ID
SCGR – Superordinate counter: Good Read	Superordinate Good Read counter; remains in case of a new parameterization (reset via power cycle)
SCNR – Superordinate counter: No Read	Superordinate No Read counter; remains in case of a new parameterization (reset via power cycle)
DEVN – Device name	Device name
APC01 – Application counter 1	Definable counter 1
APC02 – Application counter 2	Definable counter 2
APC03 – Application counter 3	Definable counter 3
APC04 – Application counter 4	Definable counter 4
APC05 – Application counter 5	Definable counter 5
APC06 – Application counter 6	Definable counter 6
APC07 – Application counter 7	Definable counter 7
APC08 – Application counter 8	Definable counter 8
APC09 – Application counter 9	Definable counter 9
APC10 – Application counter 10	Definable counter 10
TS – Time stamp	Time stamp of reading gate (start reading gate)
RR – Read rate	Ratio between Good Reads (SCGR) and number of reading gates
OGA – Distance between two objects	Distance to previous object in mm

Match counters...	Description
Match1	Counts how often Condition Match1 was fulfilled (object-related)

LECTOR®620

Match counters...	Description
TeachIn1	Counts how often Condition TeachIn1 was fulfilled (object-related)
TeachIn2	Counts how often Condition TeachIn2 was fulfilled (object-related)

Superordinate counters...	Description
Match1	Counts how often Condition Match1 was fulfilled
TeachIn1	Counts how often Condition TeachIn1 was fulfilled
TeachIn2	Counts how often Condition TeachIn2 was fulfilled

Special characters...	Description
NUL 00H	Null
SOH 01H	Start of heading
STX 02H	Start of text
ETX 03H	End of text
EOT 04H	End of transmission
ENQ 05H	Enquiry
ACK 06H	Acknowledge
BEL 07H	Bell
BS 08H	Backspace
HT 09H	Horizontal tab
LF 0AH	NL line feed, new line
VT 0BH	Vertical tab
FF 0CH	NP from feed, new page
CR 0DH	Carriage return
SO 0EH	Shift out
SI 0FH	Shift in
DLE 10H	Data link escape
DC1 11H	Device control 1
DC2 12H	Device control 2
DC3 13H	Device control 3
DC4 14H	Device control 4
NAK 15H	Negative acknowledge
SYN 16H	Synchronous idle
ETB 17H	End of transmission block
CAN 18H	Cancel
EM 19H	End of medium
SUB 1AH	Substitute
ESC 1BH	Escape
FSP1CH	File separator
GSP 1DH	Group separator
RSP 1EH	Record separator
USP 1FH	Unit separator
SPC 20H	Space

Special characters...	Description
DEL 7FH	Delete
User defined	User defined [HEX]
Nothing	Nothing

Wizard Pressing the *Wizard* button launches a dialog box which can be used to select frequently-used standard output formats. The output format selected in the wizard is accepted into the graphic interface of the *Output format 1* group where it is displayed graphically. The output format can be individually adjusted at a later stage.

5.8.7.2 Output format 2

The format of the data string for outputting the reading results is defined on the graphical interface of the *Output format 2* group. A total of two different formats (*Output format 1* and *Output format 2*) can be defined. During configuration of the interfaces, one of the two output formats can be assigned.

The output format is graphically displayed and can contain conditions, special characters (orange), variables (blue), or free text. To individually modify the output format, you select the required insertion position with the mouse. The entries are made using the keyboard or special buttons:

You can use the  button or the context menu (right mouse button) to insert, for example, conditions, special characters (e.g., start and stop), or reading result variables.



For additional information see the F1 help.

You can use the  button to modify the properties of a condition. For each condition, you can define which data should be output if the condition is met. You can also define data for the case in which the condition is not met. Conditions can be interleaved.

To check the output format, you can display the reading results on the terminal. To do this, you open the terminal using the  button in the toolbar and establish a connection with the device (VERBINDUNG menu on the terminal). For each trigger (reading gate), a data string is transferred and displayed in the terminal.

Data	Description
BC – Code content	Content of the code that has been read
CL – Code length	Length of the code that has been read
X – X pos. absolute	Code distance to trigger position X in mm
Y – Y pos. absolute	Code distance to trigger position Y in mm
Z – Z pos. absolute	Code distance to trigger position Z in mm
CS – Code security	Indicates how a 1D code was read
VAL – Code valid (1 or 0)	Outputs whether the code that has been read is valid, i.e., whether the configured conditions have been met, e.g., queries whether number of multireads has been reached
ID – Code type	Code type of code that has been read

Data	Description
ON – Object number	Number of the current object (total number of scanned objects)
CC – Code count	Number of codes that have been read within reading gate
VCC – Number of valid codes	Number of valid codes within reading gate

More code-related data...	Description
IWA – Code ID	Code type designated according to alignment number (ASCII position of the code ID, i.e., alphabetical numbering of the code types)
IN – Increment	Increment value at the start of the reading gate
FC – Focus position	Reading distance setting in mm
RSD1DV – Ratio between scans and decoded scans	Value that indicates the ratio between "S1D – Number of scans completed on a 1D code" and "SD1D – Number of scans completed successfully on a 1D code"; grading for evaluating the reading stability of 1D codes.
VTH – CS threshold for code to become valid	Valid when $CS \geq VTH$, i.e. number of scans = $VTH * 2$ (for 1D codes only)
S1D – Scans on 1D codes	Number of scans completed on 1D codes
SD1D – Decoded scans on 1D codes	Number of successful scans completed on 1D codes
FCT – Full code type description	Code type, detailed description
DI – Dimension	Indicates whether the code is 1D or 2D
X1 – Code corner 1 X	Pixel position of code corner 1X
Y1 – Code corner 1 Y	Pixel position of code corner 2Y
X2 – Code corner 2 X	Pixel position of code corner 2X
Y2 – Code corner 2 Y	Pixel position of code corner 2Y
X3 – Code corner 3 X	Pixel position of code corner 3X
Y3 – code corner 3 Y	Pixel position of code corner 3Y
X4 – code corner 4 X	Pixel position of code corner 4X
Y4 – code corner 4 Y	Pixel position of code corner 4Y
CPX – X position relative to image origin	Code center X in pixels
CPY – Y position relative to image origin	Code center Y in pixels
AN2 – Tilt angle	Tilt angle for 1D and 2D codes [in degrees] relative to the horizontal
MCR – Ratio between image and MC [in %]	Relationship between images that have been taken and successful multireads
X2D – X 2D	Symbol size (number of modules) in X direction; for 2D codes only
Y2D – Y 2D	Symbol size (number of modules) in Y direction; for 2D codes only
MX2 – Module X size	Average module size in X direction in mm; for 2D codes only
MY2 – Module Y size	Average module size in Y direction in mm; for 2D codes only
IV2 – Inverse (boolean)	Code was read inversely
MD2 – Mirrored (boolean)	Code was read as a mirror-image

More code-related data...	Description
CTG – ISO 16022/ISO15415 Symbol contrast grade	Grading for symbol contrast in accordance with ISO/IEC16022 and ISO/IEC15415. Checks the contrast between dark and light dots in the code.
CTV – ISO16022/ISO15415 Symbol contrast value	Value for symbol contrast in accordance with ISO/IEC16022 and ISO/IEC15415. Checks the contrast between dark and light dots in the code.

More...	Description
PGG – ISO16022/ISO15415 Print growth grade	Grading for print growth in accordance with ISO/IEC16022 and ISO/IEC15415. Checks the difference between the ideal and actual dot size. The value is output in accordance with the values defined in ISO/IEC16022 and ISO/IEC15415.
PGV – ISO16022/ISO15415 Print growth value	Value for print growth in accordance with ISO/IEC16022 and ISO/IEC15415. Checks the difference between the ideal and actual dot size. The value is output in accordance with the values defined in ISO/IEC16022 and ISO/IEC15415.
ANUV – ISO16022/ISO15415 Axial non-uniformity value	Value for axial non-uniformity in accordance with ISO/IEC16022 and ISO/IEC15415. Checks the ratio between the code length and width. If the code is stretched or compressed in terms of its length or width, it is given a poor rating for its axial non-uniformity.
ANUG – ISO16022/ISO15415 Axial non-uniformity grade	Grading for axial non-uniformity in accordance with ISO/IEC16022 and ISO/IEC15415. Checks the ratio between the code length and width. If the code is stretched or compressed in terms of its length or width, it is given a poor rating for its axial non-uniformity.
UECV – ISO16022/ISO15415 Unused error correction value	Value for unused error correction in accordance with ISO/IEC16022 and ISO/IEC15415. Checks how much redundant data had to be used during the reading to decode the data content. The best grade is achieved if redundancy was not required at all.
UECG – ISO16022/ISO15415 Unused error correction grade	Grading for unused error correction in accordance with ISO/IEC16022 and ISO/IEC15415. Checks how much redundant data had to be used during the reading to decode the data content. The best grade is achieved if redundancy was not required at all.
OSG – Overall symbol grade	Overall symbol grade in accordance with ISO16022 and ISO15415. This grading can be considered as a summary of the criteria. The poorest of all the criteria used is always output.
GNUV – ISO16022/ISO15415 Grid non-uniformity value	Grading for grid non-linearity in accordance with ISO/IEC16022 and ISO/IEC15415. A grid is placed over the code using the alternating pattern to locate the scan points for decoding. Grid non-uniformity checks the extent to which the grid deviates from the ideal grid and issues this in the form of a value.

More...	Description
GNUG – ISO16022/ISO15415 Grid non-uniformity grading	Value for grid non-uniformity in accordance with ISO/IEC15415. A grid is placed over the code using the alternating pattern to locate the scan points for decoding. Grid non-uniformity checks the extent to which the grid deviates from the ideal grid and issues this in the form of a grading.
MODV – ISO15415 Modulation value	Value for modulation in accordance with ISO/IEC15415. Checks reflective uniformity of dark and light dots and issues this in the form of a value.
MODG – ISO15415 Modulation grade	Grading for modulation in accordance with ISO/IEC15415. Checks the reflective uniformity of dark and light dots and issues this in the form of a grading.
FPDV – ISO15415 Fixed pattern damage value	Value for fixed pattern damage in accordance with ISO/IEC15415. Checks the essential features of the code (quiet zone, finder, alternating patterns, and reference dots) for defects and calculates an average value.
FPDG – ISO15415 Fixed pattern damage grade	Grading for fixed pattern damage in accordance with ISO/IEC15415. Checks the essential features of the code (quiet zone, finder, alternating patterns, and reference dots) for defects and calculates an average grading.
EC – ECC level	Specific DMX quality value
ECCW – ECC code words	Number of redundant code modules for Reed-Solomon error correction
DCW – Data code words	Number of code data modules
NEC – Number of ECC correctables	Number of correctable code modules in case of damage
NEE – Number of ECC errors	Number of incorrect code modules
NES – number of ECC erasures	Number of incorrect code modules with known position
MC – Multiread count	Indicates how often a code was read within a reading gate (codes are not displayed separately!)
DLSZ – Number of devices	Number of connected devices
DLCS – Device list separated by comma	List of all connected devices separated by a comma
DLBS – Device list separated by blank	List of all connected devices separated by a blank space
DN – Device number	Device number alternatively configured in the fieldbus; number is identical to device ID by default
DIN – Increment read by device	Increment read by device
SID – Symbol ID	Symbol ID

More object-related data...	Description
FIX1 – Fix defined string 1	User-defined string 1
FIX2 – Fix defined string 2	User-defined string 2
FIX3 – Fix defined string 3	User-defined string 3

More object-related data...	Description
TT – Reading gate duration in ms	Reading gate duration in ms
OTL – Trigger length in mm	Trigger length in mm
NZ – Number of mismatches	Number of mismatches (codes which do not fulfill the match code condition)
NY – Number of no matches	Number of objects without any match
GNC – Global number of codes	Number of codes
NVRB – Global number of valid released codes	Number of valid released codes
NC – Number of reading gates	Number of reading gates
DID – Device ID	Device ID
SCGR – Superordinate counter: Good Read	Superordinate Good Read counter; remains in case of a new parameterization (reset via power cycle)
SCNR – Superordinate counter: No Read	Superordinate No Read counter; remains in case of a new parameterization (reset via power cycle)
DEVN – Device name	Device name
APC01 – Application counter 1	Definable counter 1
APC02 – Application counter 2	Definable counter 2
APC03 – Application counter 3	Definable counter 3
APC04 – Application counter 4	Definable counter 4
APC05 – Application counter 5	Definable counter 5
APC06 – Application counter 6	Definable counter 6
APC07 – Application counter 7	Definable counter 7
APC08 – Application counter 8	Definable counter 8
APC09 – Application counter 9	Definable counter 9
APC10 – Application counter 10	Definable counter 10
TS – Time stamp	Time stamp of reading gate (start reading gate)
RR – Read rate	Ratio between Good Reads (SCGR) and number of reading gates
OGA – Distance between two objects	Distance to previous object in mm

Match counters...	Description
Match1	Counts how often Condition Match1 was fulfilled (object-related)
TeachIn1	Counts how often Condition TeachIn1 was fulfilled (object-related)
TeachIn2	Counts how often Condition TeachIn2 was fulfilled (object-related)

Superordinate counters...	Description
Match1	Counts how often Condition Match1 was fulfilled
TeachIn1	Counts how often Condition TeachIn1 was fulfilled
TeachIn2	Counts how often Condition TeachIn2 was fulfilled

Special characters...	Description
NUL OOH	Null

Special characters...	Description
SOH 01H	Start of heading
STX 02H	Start of text
ETX 03H	End of text
EOT 04H	End of transmission
ENQ 05H	Enquiry
ACK 06H	Acknowledge
BEL 07H	Bell
BS 08H	Backspace
HT 09H	Horizontal tab
LF 0AH	NL line feed, new line
VT 0BH	Vertical tab
FF 0CH	NP from feed, new page
CR 0DH	Carriage return
SO 0EH	Shift out
SI 0FH	Shift in
DLE 10H	Data link escape
DC1 11H	Device control 1
DC2 12H	Device control 2
DC3 13H	Device control 3
DC4 14H	Device control 4
NAK 15H	Negative acknowledge
SYN 16H	Synchronous idle
ETB 17H	End of transmission block
CAN 18H	Cancel
EM 19H	End of medium
SUB 1AH	Substitute
ESC 1BH	Escape
FSP1CH	File separator
GSP 1DH	Group separator
RSP 1 EH	Record separator
USP 1FH	Unit separator
SPC 20H	Space
DEL 7FH	Delete
User defined	User defined [HEX]
Nothing	Nothing

Wizard Pressing the [Wizard](#) button launches a dialog box which can be used to select frequently-used standard output formats. The output format selected in the wizard is accepted into the graphic interface of the [Output format 2](#) group where it is displayed graphically. The output format can be individually adjusted at a later stage.

5.8.7.3 Heartbeat format

[Heartbeat format](#)

5.8.8 Serial input data

The *Serial input data* tab contains the parameters for handling input data. The following options are available for handling input data from the serial aux and host interfaces:

- DATA FORWARDING
- AUXILIARY INPUT
- EDS TEXT ELEMENT IN OUTPUT FORMAT

The data can be transferred to the sensor via the serial aux/serial host interfaces.

The telegram identifier must differentiate between STX and ETX SOPAS commands, as other SOPAS commands can be sent to the sensor. It is recommended to use the default values for headers (*Rx header*) and terminators (*Rx terminator*) for the received data.

5.8.8.1 Data forwarding

This option can be used if you intend to send data from an external device such as a hand-held scanner to a host **in addition to** the reading result. In this case, the data is not interpreted by the sensor, but is also output at the specified interface. The digital outputs are **not controlled** and the data is not processed according to evaluation conditions.

Possible destinations are:

- Serial aux interface
- Serial host interface
- Ethernet aux interface
- Ethernet host interface

5.8.8.1.1 Rx header

The *Rx header* parameter is used to define a **header for the received data**. This header is used by the sensor to identify the start of the input data. The identification must differ from the SOPAS command. It is recommended to use the default setting.

The header is restricted to two characters and must be entered via the context menu or the  button. It is possible to enter control characters and constants. The input field is highlighted in red if invalid characters are entered.

5.8.8.1.2 Rx terminator

The *Rx terminator* parameter is used to define a **terminator for the received data**. This terminator is used by the sensor to identify the end of the input data.

The input is restricted to two characters and must be entered via the context menu or the  button. It is possible to enter control characters and constants. The input field is highlighted in red if invalid characters are entered.

5.8.8.1.3 Target interface

The *Target interface* parameter determines which interface is used to **output the data**.

The following sensor interfaces can be used as a data output for data forwarding:

- Serial aux interface
- Serial host interface
- Ethernet aux interface
- Ethernet host interface

5.8.8.1.4 Tx header

A header for the received data is defined at the output via the *Tx header* parameter. *Rx header* and *Rx terminator* are removed once the data is received. The string is reframed with *Tx header* and *Tx terminator* for the data output.

The input is restricted to two characters and must be entered via the context menu or the  button. It is possible to enter control characters and constants. The input field is highlighted in red if invalid characters are entered.

5.8.8.1.5 Tx terminator

A terminator for the received data is defined at the output via the *Tx terminator* parameter. *Rx header* and *Rx terminator* are removed once the data is received. The string is reframed with *Tx header* and *Tx terminator* for the data output.

The input is restricted to two characters and must be entered via the context menu or the  button. It is possible to enter control characters and constants. The input field is highlighted in red if invalid characters are entered.

5.8.8.2 Auxiliary input/Auxiliary read result

This option can be used if you intend for data from an external device such as a hand-held scanner to be output **instead of** the specific sensor reading. The auxiliary input must be made when the sensor's reading gate is closed.

The external device data is entered in the sensor's database and output immediately at **all configured interfaces** – including the fieldbus interfaces – in the same way as a specific reading result.

This generates **the same format** as is used in the case of a specific result. The data is interpreted by the sensor and can also control the digital outputs and, if necessary, the beeper. The data can also be checked using **evaluation conditions** (e.g., at Match 1).

5.8.8.2.1 Rx header

The *Rx header* parameter is used to define a **header for the received data**. This header is used by the sensor to identify the start of the input data. The identification must differ from the SOPAS command. It is recommended to use the default setting.

The header is restricted to two characters and must be entered via the context menu or the  button. It is possible to enter control characters and constants. The input field is highlighted in red if invalid characters are entered.

5.8.8.2.2 Rx terminator

The *Rx terminator* parameter is used to define a **terminator for the received data**. This terminator is used by the sensor to identify the end of the input data.

The input is restricted to two characters and must be entered via the context menu or the  button. It is possible to enter control characters and constants. The input field is highlighted in red if invalid characters are entered.

5.8.8.2.3 Bar code type

The *Bar code type* parameter determines which bar code type is used to enter the received data in the sensor. The selected bar code type must also be enabled in the sensor.

For the USE FIRST CHARACTER FOR IDENTIFICATION setting, the first character after the *Rx header* must correspond to the code ID for the required bar code.

Examples:

- STX > E 123456 ETX: Input via code ID "e" as 2/5 interleaved, content = 123456
- STX > B ABCDEF ETX: Input via code ID "b" as code 39, content = abcdef



For a list of valid bar code IDs, see [Code IDs for bar codes](#)

5.8.8.2.4 Code IDs for bar codes

- A, CODABAR
- B, CODE39
- C, UPC
- D, EAN
- E, INTERLEAVED25
- G, MSI CODE
- I, CODE93
- J, CODE128
- L, PLESSY
- N, EAN 128
- O, PHARMA
- P, GS1 DataBar Expanded
- R, PDF417
- S, QR CODE
- T, TELEPEN
- U, GS1 DataBar Limited
- W, MAXICODE
- Y, GS1_DATABAR
- Z, AZTEC
- <, CODE32 (6 digit C39 decodes as C32)

5.8.8.3 EDS TEXT ELEMENT in output format / element output format

This option can be used if you intend for data from an external device such as a **weighing scale** to be **joined** with the sensor data. This allows the reading result and the external data to be sent to the host **in a string**.

For this to take place, the external data must be received by the sensor **while the reading gate is open**. If the sensor does not receive any data during the reading gate interval, a default text is sent so that the host is aware that no data has been received. It makes the most sense to use this option with data output at the PSDI end.

The EDS text element can be positioned freely as a variable in its output format and can be formatted by double-clicking on a fixed length.

The output takes place at **all configured interfaces**, including the fieldbus interfaces.

5.8.8.3.1 Rx header

The *Rx header* parameter is used to define a **header for the received data**. This header is used by the sensor to identify the start of the input data. The identification must differ from the SOPAS command. It is recommended to use the default setting.

The header is restricted to two characters and must be entered via the context menu or the  button. It is possible to enter control characters and constants. The input field is highlighted in red if invalid characters are entered.

5.8.8.3.2 Rx terminator

The *Rx terminator* parameter is used to define a **terminator for the received data**. This terminator is used by the sensor to identify the end of the input data.

The input is restricted to two characters and must be entered via the context menu or the  button. It is possible to enter control characters and constants. The input field is highlighted in red if invalid characters are entered.

5.8.8.3.3 Default text

The *Default text* parameter is used to define a default text for the EDS variable. The default text is only output for EDS if no valid data was received during the reading gate.

The EDS variable is described with the default text at the start of the reading gate. During the reading gate, the EDS variable can be overwritten by an external device with the required data – several times, if necessary. At the end, the most recently assigned value or the default text is output.

To ensure that data is output, the EDS variable (object-related, EDS = external data string) must be inserted. The option is available to format the EDS variable by double-clicking, e.g., on a fixed length.

5.8.9 Application counters

Events such as successful readings (GOOD READ) and unsuccessful readings (NO READ) are counted and saved on the *Application counters* tab. Depending on the setting, the device stores the data temporarily in the RAM or permanently in the flash memory.

Quantitative conditions are defined using counters and can be used, for example, for output formatting. Counting functions can be used to control processes, for example object bundling is initiated upon reaching a defined counter reading.

5.8.9.1 Application counter

Events such as successful readings (GOOD READ) and unsuccessful readings (NO READ) are counted and saved in the *Application counter* group. Depending on the setting, the device stores the data temporarily in the RAM or permanently in the flash memory.

Quantitative conditions are defined using counters and can be used, for example, for output formatting. Counting functions can be used to control processes, for example object bundling is initiated upon reaching a defined counter reading.

5.8.9.1.1 Resetting counter statuses

Pressing the [Resetting counter statuses](#) button resets all counters.

To reset the permanent flash memory in addition to this, the [Saving current counter statuses](#) button must also be pressed.

5.8.9.1.2 Saving current counter statuses

Pressing the [Saving current counter statuses](#) button saves all values to the device's flash memory. This means the values are available without a permanent supply voltage, i.e., the values are still saved after restarting the device.

5.8.9.1.3 Application counter

Various events can be selected as application counters.

Counter	Description
Deactivated	No event counting
Input 1	Number of input 1 activations
Input 2	Number of input 2 activations
Ext. input 1	Number of external input 1 activations
Ext input 2	Number of external input 2 activations
Output 1	Number of output 1 activations
Output 2	Number of output 2 activations
Output 3	Number of output 3 activations
Output 4	Number of output 4 activations
Ext. output 1	Number of external output 1 activations
Ext. output 2	Number of external output 2 activations
Condition match 1	Number of codes read that correspond to the match code from <code>BEDINGUNG MATCH 1</code>
Condition MultCodes1	Number of codes read that correspond to the number specified in <code>BEDINGUNG MULTCODES1</code>
Condition TeachIn1	Number of codes read that correspond to the match code from <code>BEDINGUNG TEACHIN1</code>
Condition TeachIn2	Number of codes read that correspond to the match code from <code>BEDINGUNG TEACHIN2</code>
(NZ) Number of mismatches	Number of codes read that do not fulfill the match conditions
(NY) Number of no matches	Number of objects without a match
(GNC) Global number of codes	Number of codes read
(NRVB) Global number of valid released codes	Number of valid released codes
(NC) ANZ. LESETORE	Number of reading gates
(SCGR) Superordinate counter: Good Read	Superordinate Good Read counter; remains in case of a new parameterization (reset via power cycle)
(SCNR) Superordinate counter: No Read	Superordinate No Read counter; remains in case of a new parameterization (reset via power cycle)
(RR) Read rate	Ratio between Good Reads (SCGR) and number of reading gates

5.8.9.1.4 Value

The *Value* field displays the current counter state.

5.8.9.1.5 Saving the counter state permanently

When the *Saving the counter state permanently* parameter is activated, the counter status is automatically backed up in the flash memory every 6 minutes.

An additional manual backup can take place at any time via the *Saving current counter statuses* button.

The automatic backup function allows the counter status availability to be ensured – even in the event of a brownout or device restart.

5.9 Analysis tools

The *Analysis tools* folder contains parameter groups that are used to diagnose and analyze the reading results.

5.9.1 Device time

Synchronization of the device time is defined on the *Device time* tab.

By entering a time source that the device can use to synchronize itself, it is possible to obtain an exact analysis of the reading results after the recording period.

5.9.1.1 Device time

Synchronization of the device time is defined via the parameters in the *Device time* group.

By entering a time source that the device can use to synchronize itself, it is possible to obtain an exact analysis of the reading results after the recording period.

The actual time on the connected reading device can be retrieved via the REQUEST TIME button.

5.9.1.1.1 Time source

The *Time source* parameter defines where the reading device obtains a time specification for the synchronization.

- TIME SERVER: The exact date and time are requested from a time server at regular intervals.
- MANUAL INPUT: A time entered on the user interface is used once to synchronize the device time. The synchronization is triggered by the user via a button. When the device is switched off, the device time is automatically reset to the most recent manually entered value.



It is recommended to use a time server for synchronization purposes as this setting does not require the device time to be manually reentered after restarting the device.

5.9.1.1.2 Time server IP address

The *Time server IP address* parameter is used to enter the server's IP address from which the reading device obtains the time during synchronization. The user can initiate the first synchronization process using the RETRIEVE TIME FROM TIME SERVER button.

5.9.1.1.3 Port

The *Port* parameter is used to set the port to be used for the connection to the time server.

5.9.1.1.4 Connection timeout

The *Connection timeout* parameter is used to define a timeout in ms for establishing the connection between the device and time server. After a connection interruption or restart, the device attempts to establish a connection to the server at predefined intervals.

5.9.1.1.5 Time difference to GMT

A deviation from Greenwich Mean Time (GMT) is set via the *Time difference to GMT* parameter. The time difference is entered in minutes.

If a time server with a GMT time zone is used, yet the reading device is to be used in a different time zone, it is recommended and indeed necessary to set the deviation from GMT.

5.9.1.1.6 Date (YYYY-MM-DD)

The *Date (YYYY-MM-DD)* parameter is used to enter the date to be accepted by the reading device when setting its device time. The device time synchronization is triggered by the user via the SET DEVICE TIME button.

5.9.1.1.7 Time (hh:mm:ss)

The *Time (hh:mm:ss)* parameter is used to enter the time to be accepted by the reading device when setting its device time. The device time synchronization is triggered by the user via the SET DEVICE TIME button.

5.9.2 Image diagnosis

The *Image diagnosis* tab is used to determine which images are transmitted or stored by the reading device for diagnosis purposes. It is even possible to define time delays and the conditions for transmitting or backing up the images.

The file path, storage medium, and image names can also be defined.

One of the purposes of saving images is to support the user during the commissioning process by making it clear to see what the device is detecting. A further function of saving images is to help analyze the device in the event of a failed reading (reading result: NO READ). The saved images make it easier to identify causes of failure such as the deterioration of code markings.

5.9.2.1 Image diagnostic

The *Image diagnostic* group contains the parameters for image output.

Depending on your requirements, the selection of images to be stored can be restricted. For example, you can store only images from unsuccessful readings in order to investigate possible causes of failed/incorrect readings.

5.9.2.1.1 Conditions for image storing

The *Conditions for image storing* parameter is used to select the condition on which images are stored for diagnosis.

- GOOD READ: Only images of a successful code reading are stored.
- NO READ: Only images of an unsuccessful code reading are stored.
- ALL: All recorded images are stored.
- CONDITION MATCH 1: The recorded images are only stored if the read code corresponds to the matchcode from CONDITION MATCH 1.
- CONDITION MULTCODES1: The recorded images are only stored if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- CONDITION TEACHIN1: The recorded images are only stored if condition *Matchcode Teach-in 1* is met.
- CONDITION TEACHIN2: The recorded images are only stored if condition *Match-code Teach-in 2 (Additional)* is met.

5.9.2.1.2 Prioritizing image output over the trigger

The *Prioritizing image output over the trigger* parameter defines a processing sequence for outputting images and accepting an object trigger. If this parameter is activated, the diagnostics images are transmitted in full before the device accepts and processes an object trigger for the subsequent reading process. If this parameter is deactivated, a trigger signal is processed by the device in any case and an unfinished image output is interrupted.

To set up the reading device, the priority of the image output can be used to determine what the device detects within a reading gate. If this parameter is activated, you can check the object trigger interval and length of the reading process. In applications where image output is more important than continuous object/code detection, it is only recommended to prioritize the image output for normal operation in exceptional circumstances.

5.9.2.1.3 Image selection

You use the *Image selection* parameter to select the images to be stored for each reading gate.

- LAST GOOD READ / LAST IMAGE: If the code reading was successful, the last image on which one or more codes were read is stored. If the code reading was not successful, the most recently recorded image is stored. (Recommended for applications for which the code does not move out of the reading gate during reading.)
- FIRST IMAGE: The first recorded image is stored. (Recommended for applications for which the code is in motion during reading and for which triggering takes place to the code, i.e. the first image already contains a code.)
- IMAGE SEQUENCE: All recorded images are stored as a sequence. (Recommended for applications for which the reading gate is opened by an object and the position of the code on the object is not known exactly.) The recording of image sequences is used to check triggering and recording quality for moving objects. The images are numbered top left. If image numbers are missing, not all recordings can be decoded and the settings (e.g. code settings) must be modified accordingly.

5.9.2.1.4 Image quality

You use the *Image quality* parameter to modify the file size or quality (resolution) of the stored image files. The setting has no effect on the reading speed of the LECTOR®620.

For each image, an xml file containing the data of the *Code infobox* is stored.

- Quality level 6 (max. resolution): File size with full image region of interest 354 kB
- Quality level 5: File size with full image region of interest 90 KB
- Quality level 4: File size with full image region of interest 41 KB
- Quality level 3: File size with full image region of interest 24 KB
- Quality level 2: File size with full image region of interest 16 KB
- Quality level 1: File size with full image region of interest 12 KB

5.9.2.2 Save destination

The *Save destination* group contains the parameters for saving the image files. The save destination determines the media that is used to store the images.



The storage media selection affects the storage times. Saving the data to the internal memory or onto a microSD card takes significantly longer than other storage options. It is therefore recommended to select the INTERNAL MEMORY (PERMANENT) storage option or the SOPAS (PC) storage option at high reading speeds. Only one storage option should be activated at high reading speeds.

5.9.2.2.1 Internal memory (permanent)

If the *Internal memory (permanent)* parameter is activated, the recorded images are permanently stored in the internal memory of the LECTOR®620. The images are retained even after the device has been switched off.

The available internal memory size is 120 MB. The stored images can be transferred to the PC and can be manually deleted.

5.9.2.2.2 Counter

The *Counter* parameter has the fixed value TRIGGER NO. and cannot be changed.

The stored images are sequentially numbered. The number is increased by 1 each time the reading gate is opened. After the device is restarted, numbering starts again at 1. If the memory is full, no further images are stored.

5.9.2.2.3 Modulo

5.9.2.2.4 Internal memory (temporary)

If the *Internal memory (temporary)* parameter is activated, the recorded images are stored in the internal memory of the LECTOR®620. When the LECTOR®620 is switched off, the stored images are deleted.

The available internal memory size is 15 MB. Therefore around 40 images can be stored. The stored images can be transferred to the PC and can be manually deleted.

5.9.2.2.5 Counter

The *Counter* parameter is used to define the numbering for the stored images.

- **CYCLIC NO:** The stored images are sequentially numbered up to a definable value (*Modulo*). As soon as this value has been reached, numbering starts again from 1, thereby overwriting existing images. After the device is restarted or parameters are changed, numbering also starts again from 1. In the case of cyclic numbering, the most recently recorded images are stored.
- **TRIGGER NO:** The stored images are sequentially numbered. The number is increased in each case by 1 as soon as the reading gate is opened. After the device is restarted, numbering starts again at 1. If the memory is full, no further images are stored. In the case of numbering via the trigger, the images recorded first are stored.

5.9.2.2.6 Modulo

The *Modulo* parameter is used to define the maximum value for the counter.

As soon as this value has been reached, numbering starts again from 1, thereby overwriting existing images. This enables you to restrict the maximum memory required for the stored images.

5.9.2.2.7 MicroSD card

If the *MicroSD card* parameter is activated, the recorded images are permanently stored on the memory card of the LECTOR®620.

For this, a suitable memory card (MicroSD up to 32 GB, file system FAT) must be available in the LECTOR®620.

You can use memory cards to easily increase the storage space. The stored image files can be analyzed/processed further on any PC.



For many MicroSD cards, the maximum number of files per folder is restricted to 10000.

5.9.2.2.8 Counter

The *Counter* parameter is used to define the numbering for the stored images.

- **CYCLIC NO:** The stored images are sequentially numbered up to a definable value (*Modulo*). As soon as this value has been reached, numbering starts again from 1, thereby overwriting existing images. After the device is restarted or parameters are changed, numbering also starts again from 1. In the case of cyclic numbering, the most recently recorded images are stored.
- **TRIGGER NO:** The stored images are sequentially numbered. The number is increased in each case by 1 as soon as the reading gate is opened. After the device is restarted, numbering starts again at 1. If the memory is full, no further images are stored. In the case of numbering via the trigger, the images recorded first are stored.

5.9.2.2.9 Modulo

The *Modulo* parameter is used to define the maximum value for the counter.

As soon as this value has been reached, numbering starts again from 1, thereby overwriting existing images. This enables you to restrict the maximum memory required for the stored images.

5.9.2.2.10 Remove SD card

If you press the *Remove SD card* button, any write or read operations are concluded and the memory card is de-energized. As a result, the memory card can be removed safely and you do not need to worry about losing data or destroying the card.

5.9.2.2.11 Check SD card

The working order of the memory card is checked by pressing the *Check SD card* button.

5.9.2.2.12 SOPAS (PC)

If the *SOPAS (PC)* parameter is activated, the recorded images are stored on the connected PC.

For this, the SOPAS configuration software must be running during recording. The recorded images are shown in the main view of the *Online images* tab.

5.9.2.2.13 Select directory

You can use the *Select directory* button to define the storage location for the recorded images on the connected PC.

5.9.2.2.14 Modulo

The *Modulo* parameter is used to define the maximum value for the counter.

As soon as this value has been reached, numbering starts again from 1, thereby overwriting existing images. This enables you to restrict the maximum memory required for the stored images.

5.9.2.2.15 Counter

The *Counter* parameter is used to define the numbering for the stored images.

- **CYCLIC NO:** The stored images are sequentially numbered up to a definable value (*Modulo*). As soon as this value has been reached, numbering starts again from 1, thereby overwriting existing images. After the device is restarted or parameters are changed, numbering also starts again from 1. In the case of cyclic numbering, the most recently recorded images are stored.
- **TRIGGER NO:** The stored images are sequentially numbered. The number is increased in each case by 1 as soon as the reading gate is opened. After the device is restarted, numbering starts again at 1. If the memory is full, no further images are stored. In the case of numbering via the trigger, the images recorded first are stored.

5.9.2.3 Image file path

The *Image file path* group contains the parameters for the storage location and naming of the image files.

5.9.2.3.1 Folder name

The *Folder name* parameter is used to select the standard name for the image folder.

- **SERIAL NUMBER:** The image files are stored in a folder that has the same name as the LECTOR®620 serial number.
- **DEVICE NAME:** The image files are stored in a folder that has the same name as the value of the *Name* parameter. If you assign suitable device names, you can easily differentiate between the image files of several similar devices (e.g. in a network).

Name You can use the *Name* parameter to define the device name for the LECTOR®620. The image files of the LECTOR®620 are stored in a folder of the same name. If you assign suitable device names, you can easily differentiate between the image files of several similar devices (e.g. in a network).

5.9.2.3.2 Date

If the *Date* parameter is activated, the diagnosis images are sorted by their recording date and stored in a subfolder with the appropriate time stamp. A new subfolder is created for each day.

 Storing data according to the recording date requires the device time to be set accurately (see *Device time*).

It makes sense to store data according to the recording date if you have a high volume of image files.

5.9.2.3.3 Hour

If the *Hour* parameter is activated, the diagnosis images are sorted by their recording time and stored in a subfolder with the appropriate time stamp. A new subfolder is created for each complete hour.



Storing data according to the recording time requires the device time to be set accurately (see *Device time*).

It makes sense to store data according to the recording time if you have a high volume of image files.

5.9.2.3.4 Subfolder Good Read / No Read

If the *Subfolder Good Read / No Read* parameter is activated, the images are stored in two different subfolders depending on the read result (GOODREAD or NOREAD) to enable easier analysis.

5.9.2.3.5 Good Read / No Read file prefix

If the *Good Read / No Read file prefix* parameter is activated, the images are only given a name supplement depending on the read result (GOODREAD or NOREAD) to enable easier analysis.

5.9.2.3.6 Image file name

You can use the *Image file name* parameter to define the file name for the image files. The image files are also given a sequential number, and if the *Good Read / No Read file prefix* parameter is activated, a name supplement (GOODREAD or NOREAD).

By assigning suitable file names, you can clearly assign the image files (e.g. to a project or device).

5.10 Network/Interfaces/IOs

Settings for using the CAN bus interface can be made on the *Network/Interfaces/IOs* tab. With the CDF600 connection module, SICK reading devices have a CAN interface that can be used in the following ways:

- The CAN interface is used to integrate the device into a SICK network. In the SICK network, only SICK reading devices can be networked via CAN bus. This type of device group is used to scan an object from different perspectives and transmit the reading result to a host. As far as the host is concerned, the device group functions as an individual device. The master/slave or multiplexer/server operating modes are available for the SICK network. Both operating modes can be combined.
- The reading device can be integrated into a CAN fieldbus system with no further specification via CANopen. The SICK reading device operates as a slave under a CANopen client (PLC) and can communicate with devices from other manufacturers.

The *Network/Interfaces/IOs* folder in the project tree contains all parameters for configuring and activating serial interfaces, fieldbus interfaces, digital inputs and outputs, and beepers.

5.10.1 Network options

Parameters for identifying the reading devices in the network are combined in the *Network options* group.

5.10.1.1 Device ID

The *Device ID* parameter assigns the device a unique number in the network. The assigned ID is used as a CAN node for the CAN bus.



If the device is connected to a SICK network via CAN, device IDs 1 to 63 can be assigned. If the device is connected to a CANopen fieldbus, device IDs 1 to 127 can be assigned. Duplicated IDs must not be assigned.

The device must be clearly identified on the network to enable communication via CAN bus and to assign reading results to devices.

5.10.1.2 Device name

A name is assigned to the device via the *Device name* parameter. The device name appears in SOPAS in the project tree.

It is not compulsory to assign a device name, although it can offer a better overview of all devices in the system during the configuration process.

5.10.2 Master/Slave

In the *Master/Slave* group, the device is configured as a SICK network participant in a master/slave operating mode. A maximum of 10 slaves can be connected to a master. The master sends the trigger signal to the slaves via CAN bus at the start of the reading process. The slaves send the reading results to the master, which then sends the reading result in the configured output format to the PLC at the end of the reading process. The PLC only receives a No Read for an unsuccessful reading result from the master if none of the slave devices have sent a successful reading result (Good Read) to the master.

The SICK network in the master/slave operating mode is typically used to read objects that can accommodate the code in various positions. Several reading devices can be used to search for the existing code on all sides of the object. The master only transmits the reading from the device with the successful reading result to the PLC. The unsuccessful readings (No Reads) from the devices where no code was available are not sent to the PLC. This increases the processing speed of the control system.



For additional information see the F1 help.

Wizard Pressing the START WIZARD button launches a dialog box via which the device can be configured in full as a SICK network participant in the master/slave operating mode. The wizard provides the parameters required for configuring the CAN interface, device network functions, output control, and object trigger control. The values entered into the wizard are accepted into the relevant parameter groups and tabs once the configuration is complete.

5.10.2.1 Function

The device function within the SICK network in master/slave operating mode is specified via the *Function* parameter.

- **STAND ALONE:** The reading device is not integrated into a SICK network and operates as a single device station in the system. The device communicates directly with the host.
- **SLAVE:** As a slave, the reading device receives trigger signals from the master for the start and end of the reading process. The slave sends the reading result to the master.
- **MASTER:** As a master, the reading device distributes trigger signals to the slaves for the start and end of the reading process. The master receives the trigger signals directly from a photoelectric switch or in series via the PLC. Once it receives the reading results from the slaves, the master sends these to the PLC in accordance with the configured output format.

5.10.2.2 Slave list

The slave's device IDs that are connected to the master device are entered via the *Slave list* parameter. The device IDs are separated by commas. Up to 10 slaves can be entered for one master.



The input field is highlighted in red if the input is invalid.

5.10.3 Multiplexer/Server

In the *Multiplexer/Server* group, the device is configured as a SICK network participant in a multiplexer/server operating mode. A maximum of 31 devices can be connected to a multiplexer. The servers receive trigger signals for the start and end of the reading process via their respective photoelectric switches which are connected to the digital input (sensor 1). This allows them to be distributed in a system at different positions. The servers transmit the reading results to the multiplexer, which forwards them to the PLC in a configured output format along with the server's device ID. As the servers have their own trigger sources and operate independently of each other, the reading results are processed and forwarded in the multiplexer in the sequence in which they are entered.

In the multiplexer/server operating mode, a SICK network of 32 reading devices can be managed without auxiliary devices. The operating mode is typically inserted in systems that have a large distribution of reading devices which function independently of each other.



For additional information see the F1 help.

Wizard Pressing the START WIZARD button launches a dialog box via which the device can be configured in full as a SICK network participant in the multiplexer/server operating mode. The wizard provides the parameters required for configuring the CAN interface, device network functions, output format, output control, and object trigger control. The values entered into the wizard are accepted into the relevant parameter groups and tabs once the configuration is complete.

5.10.3.1 Function

The device function within the SICK network in multiplexer/server operating mode is specified via the *Function* parameter.

- **SERVER:** The reading device sends its reading results to the multiplexer. In relation to the object trigger and reading process, the device functions independently with its own photoelectric switch at the digital input (sensor 1).
- **MULTIPLEXER:** The multiplexer sends the server's reading results to the PLC in a configured output format. The device ID is included with the reading results so that they can be assigned to devices. In addition to its multiplexer function, the device can also be used as a reading device with its own photoelectric switch.

5.10.3.2 Server list

The server's device IDs that are connected to the multiplexer device are entered via the *Server list* parameter. The device IDs are separated by commas. Up to 31 servers can be entered for one multiplexer.



The input field is highlighted in red if the input is invalid.

5.10.4 Monitoring

The CAN network monitoring function is activated and configured in the *Monitoring* group.

5.10.4.1 Assign to

The monitoring function in the CAN network for the reading device is activated or deactivated via the *Assign to* parameter.

- **DEACTIVATED:** The device monitors no further reading devices.
- **CAN:** The reading device monitors further reading devices on the CAN bus. The monitoring device detects the participants within reach and the devices that have failed.

5.10.4.2 Monitored devices

The participant's device IDs that are to be monitored by the reading device are entered via the *Monitored devices* parameter. The device IDs are separated by commas.



The input field is highlighted in red if the input is invalid.

5.10.4.3 Max. startup time for the monitored devices

The *Max. startup time for the monitored devices* parameter defines an interval in seconds for the startup time for the monitored devices. A participant is considered within reach when the interval between the signals that have been sent is within the specified period.

5.10.5 Serial

The *Serial* tab contains all of the parameters for configuring the serial host and aux interfaces. The serial host interface is typically used to communicate with the control (PLC). The serial aux interface is used to configure the reading device separately.

5.10.5.1 Serial Host

The *Serial Host* group contains the parameters for configuring the serial Host interface. The serial Host interface is typically used for communication with the controller (PLC).

5.10.5.1.1 Output Format

You use the *Output Format* parameter to select the output format for the interface.

- No OUTPUT: There is no data output via this interface.
- OUTPUT FORMAT 1: In the event of output via this interface, the data is formatted in line with the configuration in the *Output format 1* group.
- OUTPUT FORMAT 2: In the event of output via this interface, the data is formatted in line with the configuration in the *Output format 2* group.
- CUSTOMER OUTPUT FORMAT: Via this interface, the data is output based on customer-specific formatting. The formatting is defined via the *With number* parameter.

5.10.5.1.2 With number

The *With number* parameter is used to specify the number of the customer-specific output format.

5.10.5.1.3 Multiplexer output

The data output can be switched on or off via the *Multiplexer output* parameter.

5.10.5.1.4 Baud rate

You can use the *Baud rate* parameter to adapt the maximum transmission speed of the LECTOR®620 to the host.

The value is specified in Baud/s. This value is used to define the number of signal changes per second that can take place via the interface. Depending on the protocol, up to four bits per signal change can be transmitted (e.g. at 2400 Bauds up to 9600 bit/second).

5.10.5.1.5 Stop bits

The *Stop bits* parameter is used to define whether 1 (typical) or 2 characters are used at the end of a transmitted character as the pause (stop bits). The setting must match the setting on the host.

5.10.5.1.6 Data bits / parity

You use the *Data bits / parity* parameter to define the number of data bits (8 or 7) and the corresponding parity. The setting must match the setting on the host.

5.10.5.1.7 Hardware

The *Hardware* parameter is used to select the protocol (RS232 or RS244) for the serial interface.

5.10.5.1.8 Enable heartbeat

If the *Enable heartbeat* parameter is activated, a signal is output via the interface at regular intervals (*Heartbeat interval*). The availability of the LECTOR®620 can therefore be checked via the interface.

5.10.5.1.9 Heartbeat interval

The *Heartbeat interval* parameter is used to define the time between the heartbeat signals in s.

5.10.5.1.10 Restart interval on sending

If the *Restart interval on sending* parameter is activated, the time between the heartbeat signals starts again after output of the reading results.

5.10.5.1.11 Handling input data

The *Handling input data* parameter is used to define how the sensor uses data received at the serial host interface.

- NO INPUT DATA: No data is received at the interface.
- DATA FORWARDING: The incoming data is forwarded to another interface. Possible target interfaces are serial aux, serial host, Ethernet aux, and Ethernet host.
- AUXILIARY INPUT: The incoming data is output to all configured interfaces in the same way as a specific reading result.
- EDS TEXT ELEMENT IN OUTPUT FORMAT: The incoming data is assigned to the EDS variable, which can be used in an output string.



For further details and settings, see [Serial input data](#).

5.10.5.2 Serial Aux

The *Serial Aux* group contains the parameters for configuring the serial Aux interface. The serial Aux interface is typically used to parameterize and analyze the LECTOR®620.

As the configuration of the interface on the LECTOR®620 cannot be changed, the settings must be modified on the connected device (e.g. PC or controller).

5.10.5.2.1 Output format

The output format for the interface is selected via the *Output format* parameter.

- NO OUTPUT: There is no data output via this interface.
- READING DIAGNOSIS: This interface is used to output a diagnosis string for each reading gate.
- MONITOR SERIAL HOST INTERFACE: The aux interface is used to output the same data that is output via the host interface. This makes it possible to check the data traffic for the host interface.
- RDT400 PROTOCOL: A predefined data string is output via this interface. As a result, the reading results of several devices can be combined and evaluated using a remote diagnostic tool (RDT).
- OUTPUT FORMAT 1: Data that is output via this interface is formatted in line with the configuration in the [Output format 1](#) group.
- OUTPUT FORMAT 2: Data that is output via this interface is formatted in line with the configuration in the [Output format 2](#) group.

For the reading diagnosis, the following information is combined in a diagnosis string for each reading gate:

- TT: Length of the reading gate in ms
- OTL: Trigger length in mm
- CC: Number of codes per reading gate
- FC: Focus length in mm
- Code content
- Code type

For 1D code types

- CL: Code length
- RSD1DV: Ratio of scans to decoded scans (reading reliability) in %

For 2D code types

- SZ: Symbol size
- RES: Dot resolution in pixels
- CTV: Symbol contrast in %
- UECV: Decodability (unused error correction) in %
- MC: Number of multireads per code

5.10.5.2.2 RDT ID

RDT ID

5.10.5.2.3 Enable heartbeat

If the *Enable heartbeat* parameter is activated, a signal is output via the interface at regular intervals (*Heartbeat interval*). The availability of the LECTOR®620 can therefore be checked via the interface.

5.10.5.2.4 Heartbeat interval

The *Heartbeat interval* parameter is used to define the time between the heartbeat signals in s.

5.10.5.2.5 Restart interval on sending

If the *Restart interval on sending* parameter is activated, the time between the heartbeat signals starts again after output of the reading results.

5.10.5.2.6 Handling input data

The *Handling input data* parameter is used to define how the sensor uses data received at the serial aux interface.

- NO INPUT DATA: No data is received at the interface.
- DATA FORWARDING: The incoming data is forwarded to another interface. Possible target interfaces are serial aux, serial host, Ethernet aux, and Ethernet host.
- AUXILIARY INPUT: The incoming data is output to all configured interfaces in the same way as a specific reading result.

- EDS TEXT ELEMENT IN OUTPUT FORMAT: The incoming data is assigned to the EDS variable, which can be used in an output string.



For further details and settings, see [Serial input data](#).

5.10.6 Ethernet

The [Ethernet](#) tab contains all of the parameters for configuring the Ethernet interface. The device address and the configuration data for the host and aux ports can be entered on the tab.

5.10.6.1 Ethernet General

The [Ethernet General](#) group contains general parameters for configuring the Ethernet interfaces (Host and Aux).

5.10.6.1.1 Addressing Mode

You use the [Addressing Mode](#) parameter to define whether the IP address for the LECTOR®620 is to be defined manually (STATIC) or by a server (DHCP).

5.10.6.1.2 IP address

The [IP address](#) parameter is used to manually define the Ethernet address for the LECTOR®620. A prerequisite for this is a connection with the LECTOR®620 in the SERVICE user level.



It is possible to use the connection wizard to change the [IP address](#) and [Subnet-Mask](#) parameters in the AUTHORIZED CLIENT user level.

The Ethernet address must match the setting for the [Subnet-Mask](#) parameter and must be adapted in line with the network.

5.10.6.1.3 Subnet-Mask

The [Subnet-Mask](#) parameter is used to define the bit mask or prefix mask for network protocol IPv4 and IPv6. A prerequisite for this is a connection with the LECTOR®620 in the SERVICE user level.



It is possible to use the connection wizard to change the [IP address](#) and [Subnet-Mask](#) parameters in the AUTHORIZED CLIENT user level.

The setting must be adapted to the network.

5.10.6.1.4 Default gateway

The [Default gateway](#) parameter can be used to specify the IP address of a gateway for implementation of the network protocol. A prerequisite for this is a connection with the LECTOR®620 in the SERVICE user level.



It is possible to use the connection wizard to change the [IP address](#) and [Subnet-Mask](#) parameters in the AUTHORIZED CLIENT user level.

5.10.6.1.5 Speed

The *Speed* can be used to adapt the maximum transmission speed of the LECTOR®620 to the speed of the network.

To ensure a reliable connection, you are advised to automatically determine the speed of a directly connected PC (AUTO).

5.10.6.1.6 Negotiated

The current speed is displayed in the *Negotiated* display field.

For this, there must be a connection between the LECTOR®620 and the PC.

5.10.6.1.7 MAC-Address

The *MAC-Address* display field shows a unique number for devices with an Ethernet interface. This number can be used to uniquely identify the LECTOR®620.

5.10.6.2 Ethernet Host Port

The *Ethernet Host Port* group contains the parameters for configuring the Ethernet Host interface. The Ethernet Host interface is typically used for communication with the controller

5.10.6.2.1 Output Format

You use the *Output Format* parameter to select the output format for the interface.

- NO OUTPUT: There is no data output via this interface.
- OUTPUT FORMAT 1: In the event of output via this interface, the data is formatted in line with the configuration in the *Output format 1* group.
- OUTPUT FORMAT 2: In the event of output via this interface, the data is formatted in line with the configuration in the *Output format 2* group.

5.10.6.2.2 Multiplexer output

The data output can be switched on or off via the *Multiplexer output* parameter.

5.10.6.2.3 Server/Client

The *Server/Client* parameter is used to select the function of the LECTOR®620 within a network.

- SERVER: The LECTOR®620 performs the function of the server within a network.
- CLIENT: Within a network, the LECTOR®620 is subordinate to a server. The server is defined via the *Server Address* parameter.

5.10.6.2.4 IP port

The *IP port* parameter is used to adapt the port number for the Ethernet Host interface (default value: 2112) to the connected PC or the controller (PLC).

5.10.6.2.5 Server Address

The *Server Address* parameter is used to define the IP address of the server to which the LECTOR®620 is assigned within a network.

5.10.6.2.6 Enable heartbeat

If the *Enable heartbeat* parameter is activated, a signal is output via the interface at regular intervals (*Heartbeat interval*). The availability of the LECTOR®620 can therefore be checked via the interface.

5.10.6.2.7 Heartbeat interval

The *Heartbeat interval* parameter is used to define the time between the heartbeat signals in s.

5.10.6.2.8 Restart interval on sending

If the *Restart interval on sending* parameter is activated, the time between the heartbeat signals starts again after output of the reading results.

5.10.6.3 Ethernet Aux Port

The *Ethernet Aux Port* group contains the parameters for configuring the Ethernet Aux interface. The Ethernet Aux interface is typically used to parameterize and analyze the LECTOR®620.

5.10.6.3.1 Output Format

You use the *Output Format* parameter to select the output format for the interface.

- No OUTPUT: There is no data output via this interface.
- READING DIAGNOSIS: This interface is used to output a diagnosis string for each reading gate.
- OUTPUT FORMAT 1: In the event of output via this interface, the data is formatted in line with the configuration in the *Output format 1* group.
- OUTPUT FORMAT 2: In the event of output via this interface, the data is formatted in line with the configuration in the *Output format 2* group.

For the reading diagnosis, the following information is combined in a diagnosis string for each reading gate:

- TT: Length of the reading gate in ms
- OTL: Trigger length in mm
- CC: Number of codes per reading gate
- FC: Focus length in mm
- Code content
- Code type

for 1D symbologies

- CL: Code length
- RSD1DV: Proportion of scans to decoded scans in (reading reliability) in %

for 2D symbologies

- SZ: Symbol size
- RES: Resolution of a cell in pixels
- CTV: Symbol contrast in %
- UECV: Decodability (unused error correction) in %
- MC: Number of multireads per code

5.10.6.3.2 Server/Client

The *Server/Client* parameter has the fixed value CLIENT and cannot be changed.

5.10.6.3.3 IP port

The *IP port* parameter is used to adapt the port number for the Ethernet Aux interface (default value: 2111) to the connected PC or the controller (PLC).

5.10.7 Ethernet/IP

The device communication via Ethernet/IP can be activated and configured on the *Ethernet/IP* tab.

5.10.7.1 Ethernet/IP

The device communication via Ethernet/IP can be activated and configured in the *Ethernet/IP* group.

5.10.7.1.1 Activate Ethernet/IP

The device communication via Ethernet/IP is activated/deactivated via the *Activate Ethernet/IP* parameter. Once Ethernet/IP has been activated, the reading device must be restarted.

5.10.7.1.2 Communication protocol

A communication mode for exchanging data between the reading device and host is selected via the *Communication protocol* parameter.

- WITHOUT HANDSHAKE: The protocol type outputs the telegram directly. The mode does not function with handshakes. A repeat request is not processed.
- WITH HANDSHAKE: The protocol type outputs the telegram directly. The mode functions with handshakes. A repeat request is processed.



If several data telegrams are sent at the same time, the "With handshake" mode must be used to prevent data loss.

5.10.7.1.3 Protocol/Output format

An output format is assigned to the interface via the *Protocol/Output format* parameter. Output formats are defined under DATA PROCESSING.

5.10.7.1.4 Assembly size output on the PLC

The *Assembly size output on the PLC* parameter identifies the size of the incoming telegrams from the PLC in bytes. Values can be specified between 8 and 500 bytes.

5.10.7.1.5 Assembly size input on the PLC

The [Assembly size input on the PLC](#) parameter identifies the size of the outgoing telegrams to the PLC in bytes. Values can be specified between 8 and 500 bytes.

5.10.8 Web server

The device web server is activated and deactivated on the [Web server](#) tab.

The web server in the reading device provides statistical data on reading processes via Ethernet TCP/IP. The statistical data can be retrieved by the user via a web browser.



The device IP address is specified on the Ethernet tab.

5.10.8.1 Web server

The device web server is activated and deactivated in the [Web server](#) group.

The web server in the reading device provides statistical data on reading processes via Ethernet TCP/IP. The statistical data can be retrieved by the user via a web browser.



The device IP address is specified on the Ethernet tab.

5.10.8.1.1 Activate web server at device startup

The device web server is activated and deactivated via the [Activate web server at device startup](#) parameter. Once the web server is activated, it starts automatically every time the device is restarted. The device must be restarted when the web server is activated for the first time.

5.10.9 Monitoring

The [Monitoring](#) tab is used to activate ports for a TCP device connection which monitors the device communication via the serial host or aux interfaces.



A TCP connection to the device is only possible if an output format is activated at the monitored interface. The output format is activated on the [Serial](#) tab.

The function is used to monitor the data exchange between PLC and device via serial interfaces on a terminal.

5.10.9.1 Monitored ports

The [Monitored ports](#) group is used to activate ports for a TCP device connection which monitors the device communication via the serial host or aux interfaces.



A TCP connection to the device is only possible if an output format is activated at the monitored interface. The output format is activated on the [Serial](#) tab.

The function is used to monitor the data exchange between PLC and device via serial interfaces on a terminal.

5.10.9.1.1 Serial host interface (port 4003)

If the [Serial host interface \(port 4003\)](#) parameter is activated, the serial host interface can be monitored via TCP and port 4003.

5.10.9.1.2 Serial auxiliary interface (port 4002)

If the [Serial auxiliary interface \(port 4002\)](#) parameter is activated, the serial aux interface can be monitored via TCP and port 4002.

5.10.9.1.3 DHCP Fallback Mode

The [DHCP Fallback Mode](#) parameter is used to select an action that is performed if automatic IP assignment was not possible.

- **STATIC IP:** A static IP value is set as soon as it is established that DHCP was not possible.
- **REPEAT DHCP:** The process is repeated until an IP is assigned via DHCP.

5.10.10 CAN

The CAN interface is configured on the [CAN](#) tab.

5.10.10.1 CAN

The CAN interface is activated and configured in the [CAN](#) group. There are two options available for using the CAN interface:

- The CAN interface is used to integrate the device into a SICK network. In the SICK network, only SICK reading devices can be networked via CAN bus. This type of device group is used to scan an object from different perspectives and transmit the reading result to a host. As far as the host is concerned, the device group functions as an individual device. The master/slave or multiplexer/server operating modes are available for the SICK network. Both operating modes can be combined.
- The reading device can be integrated into a CAN fieldbus system with no further specification via CANopen. The SICK reading device operates as a slave under a CANopen client (PLC) and can communicate with devices from other manufacturers.

5.10.10.1.1 Mode

Use of the CAN interface is determined via the [Mode](#) parameter.

- **INACTIVE:** Communication via the CAN interface is deactivated.
- **SICK NETWORK:** The device is integrated into a SICK network. In the SICK network, only SICK reading devices can be networked via CAN bus. This type of device group is used to scan an object from different perspectives and transmit the reading result to a host. As far as the host is concerned, the device group functions as an individual device.
- **CANOPEN:** The device is integrated into a CAN fieldbus system with no further specification. The SICK reading device operates as a slave under a CANopen client (PLC) and can communicate with devices from other manufacturers.



The master/slave or multiplexer/server operating modes are available for the SICK network. The operating modes are configured on the [Network/Interfaces/IOs](#) tab.

5.10.10.1.2 Using device ID as a node ID

When the relevant parameter is activated, the assigned device ID is used as a CAN node for the CAN bus.

5.10.10.1.3 Device ID

The *Device ID* parameter displays the device ID that was assigned under *Network/Interfaces/IOs*. The device is assigned a unique number in the network in the form of the device ID. The assigned ID is used as a CAN node for the CAN bus.

The device must be clearly identified on the network to enable communication via CAN bus and to assign reading results to devices.

5.10.10.1.4 Data transmission rate

The maximum transmission rate can be adjusted via the *Data transmission rate* parameter. The value is provided in bits/s.



All devices in the CAN network must be operated at the data transmission rate. The maximum length of cable for the network is dependent on the data transmission rate.

5.10.10.1.5 Output format

An output format is assigned to the interface via the *Output format* parameter.



Output formats are defined under *Data processing*.

5.10.10.1.6 Enable heartbeat

If the *Enable heartbeat* parameter is activated, a signal is output via the interface at regular intervals (*Heartbeat interval*). The availability of the reading device can therefore be checked via the interface.

5.10.10.1.7 Heartbeat interval

The *Heartbeat interval* parameter is used to define the time between the heartbeat signals in seconds.

5.10.10.1.8 Restart interval on sending

If the *Restart interval on sending* parameter is activated, the time between the heartbeat signals starts again after output of the reading results.

5.10.10.1.9 Mask for digital input

A filter for the digital input bits that is used to integrate the device into a second fieldbus for assigning the data is entered via the *Mask for digital input* parameter.

Example: All bits are active when 0xFF is entered.



To assign the digital CAN input bits, see *COB ID RPDO digital input*.

5.10.10.1.10 Enable digital output

If the *Enable digital output* parameter is activated, the device's reading diagnosis data is monitored by the control via the CANopen interface. The reading device initiates monitoring via process data objects (PDOs). The device sends a PDO with identifier 0x180 and CAN node to indicate that the control should load an entry.

5.10.10.1.11 COB ID RPDO digital input

The *COB ID RPDO digital input* parameter defines the CAN identifier for the receive process data objects (RPDOs) received on the reading device.



For additional information see the F1 help.

The CAN input bits on the reading device are to be assigned to the CANopen object dictionary as follows:

- Byte 0: 0x6200 / 01
- Byte 1: 0x6200 / 02

Assignment of the digital CAN input bit for the 0x6200 object on the reading device:

Bit	Assignment	Name	Remark
Byte 0, Bit 0	Fix	Trigger	LSB
Byte 0, Bit 1	Fix	Sensor - Idle	
Byte 0, Bit 2	Fix	TeachIn1	
Byte 0, Bit 3	Fix	TeachIn2	
Byte 0, Bit 4	Fix	External Output_1	Physical Output 1 of CDF600
Byte 0, Bit 5	Fix	External Output_2	Physical Output 2 of CDF600
Byte 0, Bit 6	Fix	Digital Output_1 (Result_1)	
Byte 0, Bit 7	Fix	Digital Output (Result_2)	
Byte 1, Bit 0	Soft	PLC_Out_08	
Byte 1, Bit 1	Soft	PLC_Out_09	
Byte 1, Bit 2	Soft	PLC_Out_10	
Byte 1, Bit 3	Soft	PLC_Out_11	
Byte 1, Bit 4	Fix	Distance_Config_0	LSB
Byte 1, Bit 5	Fix	Distance_Config_1	
Byte 1, Bit 6	Fix	Distance_Config_2	
Byte 1, Bit 7	Fix	Distance_Config_3	MSB



To activate the switching functions, the Fieldbus input value must be set with the corresponding parameters. If, for example, the trigger bit is to be used for controlling the trigger, the Fieldbus input value must be set as the start of the object trigger (see *Start by*).

5.10.10.1.12 COB ID TPDO digital output

The *COB ID TPDO digital output* parameter defines the CAN identifier for the transmit process data objects (TPDOs) sent by the reading device.

For the first four transmit process data objects (TPDOs), communication and mapping can be configured by accessing objects 0x1800 to 0x1803 and 0x1A00 to 0x1A03. This task is normally performed by a CANopen control. The configuration is saved in the device's parameter set.

Objects TPDO6 to TPDO12 are used to output the reading result and cannot be configured.



It is recommended to use TPDO2 to TPDO4 for user-defined applications. Object TPDO1 is used for synchronization. TPDO1 is used to signal the availability of data that can be retrieved from the service data object client (SDO client) via an SDO upload access.



For additional information see the F1 help.

Assignment of the digital CAN output bit for the 0x6000 object on the reading device:

Bit	Assignment	Name	Remark
Byte 0, Bit 0	Fix	Device Ready	
Byte 0, Bit 1	Fix	System Ready	Only supported in the SICK network
Byte 0, Bit 2	Fix	Good Read	
Byte 0, Bit 3	Fix	No Read	
Byte 0, Bit 4	Fix	Status External Output_1	Physical Output 1 of CDF600
Byte 0, Bit 5	Fix	Status External Output_2	Physical Output 2 of CDF600
Byte 0, Bit 6	Fix	Status Output_1 (Result_1)	
Byte 0, Bit 7	Fix	Status Output (Result_2)	
Byte 1, Bit 0	Fix	External Input 1	Physical Input 1 of CDF600
Byte 1, Bit 1	Fix	External Input 2	Physical Input 2 of CDF600
Byte 1, Bit 2	Fix	Input 1 (Sensor 1)	
Byte 1, Bit 3	Fix	Input 2 (Sensor 2)	
Byte 1, Bit 4	Soft	Defined by sensor configuration	
Byte 1, Bit 5	Soft	Defined by sensor configuration	
Byte 1, Bit 6	Soft	Defined by sensor configuration	
Byte 1, Bit 7	Soft	Defined by sensor configuration	

5.10.10.1.13 Enable user IO PDOs

If the *Enable user IO PDOs* parameter is activated, user-defined process data objects (PDOs) can be sent and received by a script program.

The user-defined PDOs are entries in the object dictionary that are reserved for scripts and configured in the object dictionary. The reserved PDOs are used to allow the script to receive status changes from the CAN input modules and control CAN output modules by sending a PDO. This is how the communication bandwidth of the device is extended in terms of its software. The reading device can use the extended communication bandwidth for control tasks within its surrounding area, even if the required number of switching inputs and outputs for the control process is not available.

5.10.10.1.14 COB ID RPDO user digital input

The *COB ID RPDO user digital input* parameter defines the CAN identifier for the user-defined receive process data objects (RPDOs) received on the reading device.

For the first four receive process data objects (RPDOs), communication and mapping can be configured by accessing objects 0x1400 to 0x1403 and 0x1600 to 0x1603. The configuration is saved in the device's parameter set. The other RPDOs cannot be configured.

5.10.10.1.15 COB ID TPDO user digital output

The *COB ID TPDO user digital output* parameter defines the CAN identifier for the user-defined transmit process data objects (TPDOs) sent by the reading device.

5.10.10.1.16 Number of mapped RPDO bytes

The *Number of mapped RPDO bytes* parameter defines the scope of the receive process data objects (RPDOs) that can be received by the script program.

5.10.10.1.17 Number of mapped TPDO bytes

The *Number of mapped TPDO bytes* parameter defines the scope of the transmit process data objects (TPDOs) that can be sent by the script program.

5.10.10.1.18 Reading result with SDO/PDO

The *Reading result with SDO/PDO* parameter determines how the reading result is transmitted within the CANopen protocol.

- **WITH SDO:** The reading result is stored in the object dictionary in the configured output format. The reading device sends a process data object (PDO) with a status bit set to DATA AVAILABLE to indicate that the reading results are available. A CANopen client, typically the PLC, can load the data via the service data object domain (SDO domain). Once the upload is complete, the CANopen client issues an enabling signal for the reading results in the object dictionary.
- **WITH PDO:** The reading result is transmitted via several consecutive process data objects (PDOs). The configured reading result is restricted to 50 characters for this transmission. It is possible to configure the number of PDOs used for the transmission and the identifier of the first PDO sent. All other PDOs involved in data transmission use continuous identifiers.
- **DEACTIVATED:** No reading results are transferred.



The output format is configured on the *Output format* tab or under the *Data processing* folder.

5.10.10.1.19 Reading result timeout

The *Reading result timeout* parameter defines an interval in ms within which a reading result in the object dictionary is made available for upload by an SDO client. Once this interval has elapsed, the reading result is discarded.

When transmitting the reading results with the service data object (SDO), the reading results are stored in the object dictionary and downloaded by the SDO client. A timeout must be specified for the SDO client upload.

5.10.10.1.20 Automatic release after SDO upload

If the *Automatic release after SDO upload* parameter is activated, the reading results are automatically released by the SDO client after the upload.

When transmitting the reading results with the service data object (SDO), the reading results are stored in the object dictionary and downloaded by the SDO client. To enter the following reading result, a release must take place after the SDO upload. If the release takes place automatically, the SDO client does not require an application for release purposes. Data transmission is relatively insecure, as the process after automatic release does not enable a repeated data request by the SDO client.

5.10.10.1.21 Enable command response output

If the *Enable command response output* parameter is activated, the reading device responds to the SOPAS commands sent by the PLC.

SOPAS commands from the control are entered into object 0x2200/0. The reading device responds via an entry in object 0x2020/4 and sends a process data object (PDO) with identifier 0x180 and the CAN node number to the PLC. The PLC loads the command response entered in the object dictionary via the SDO domain.

SOPAS commands constitute a command language that can be used to control the functions of the reading device and the object trigger control. A detailed list of the command language can be obtained from SICK on request.

5.10.10.1.22 Command response timeout

The *Command response timeout* parameter defines an interval in ms within which a response to a SOPAS command is made available in the object dictionary for upload by an SDO client. Once this interval has elapsed, the command response is discarded.

When transmitting the responses to SOPAS commands, the device response is stored in the object dictionary and downloaded by the SDO client. A timeout must be specified for the SDO client upload.

5.10.10.1.23 Basic COB ID for PDO reading results

The *Basic COB ID for PDO reading results* parameter defines the CAN identifier for the first transmit process data object (TPDO) used to transmit the reading result.

When transmitting the reading results with process data objects (PDOs), the reading result is distributed to several PDOs sent in sequence and sent to the PLC. The identifier for the first PDO must be configured; subsequent PDOs have continuous identifiers.

5.10.10.1.24 Transmission type

The processing and triggering procedures for process data objects (PDOs) are configured via the *Transmission type* parameter. The CANopen parameter defines the processing and triggering time in relation to the SYNC object. The SYNC object is for synchronizing the CAN nodes in the CAN bus and, together with the TIME STAMP object, ensures that the time is uniform consistent across the network. The parameter must be entered in accordance with the CANopen standard.

OxFE example: The device processes and transmits PDOs as soon as new data is available.

There are three different transfer types:

- **ASYNCHRONOUS:** PDOs bear no reference to the SYNC object. PDOs are processed as soon as they have been received and sent immediately after they have been triggered. The trigger can be controlled by an event or a send request.
- **SYNCHRONOUS:** PDOs are processed with temporal reference to the SYNC object. It is possible to set the number of SYNC objects before processing takes place.
- **ACYCLIC SYNCHRONOUS:** PDOs are received with no reference to the SYNC object but are not processed until after the next SYNC object. The PDO is triggered by an event with no reference to the SYNC object but is not transmitted until after the next SYNC object.

5.10.10.1.25 Inhibit time

The *Inhibit time* parameter defines an interval in ms that must lie between the repeated transmission of process data objects (PDOs).

5.10.10.1.26 Number of PDOs

The *Number of PDOs* parameter is used to determine the number of process data objects (PDOs) that transmit the reading result.

When transmitting reading results with PDOs, the reading result data is distributed to several PDOs. The PDOs are transmitted in sequence. The configured reading result is restricted to 50 characters for this type of transmission.

5.10.10.1.27 Enable diagnosis output

If the *Enable diagnosis output* parameter is activated, the device's reading diagnosis data is monitored by the control via the CANopen interface. The reading device initiates monitoring via process data objects (PDOs). The device sends a PDO with identifier 0x180 and CAN node to indicate that the control should load an entry.

5.10.10.1.28 Diagnostics timeout

The *Diagnostics timeout* parameter defines an interval in ms within which reading diagnosis data is made available in the object dictionary for upload by the control. Once this interval has elapsed, the reading diagnostic data is discarded.

Alternatively, the timeout values can be modified directly in the object dictionary via objects 0x3010/01 to 0x3010/02. The changes are not accepted until the device has been restarted.

When monitoring the reading diagnostic data, the data is stored in the object dictionary and downloaded by the control. A timeout must be specified for the upload.

5.10.10.1.29 CANopen heartbeat rate / ms

The *CANopen heartbeat rate / ms* parameter defines an interval in ms that lies between the CANopen heartbeat objects.

The heartbeat protocol is used in CANopen networks to check the CAN nodes. The CANopen heartbeat protocol stipulates that every CAN node must send out heartbeat objects at cyclical intervals and receive them from other bus participants. The transmission cycle must be configured.

5.10.10.1.30 COB ID Emergency Obj

The *COB ID Emergency Obj* parameter defines the CAN identifier for emergency objects in the CANopen protocol.

CANopen protocol requires emergency objects to transmit error codes. The error codes are defined in communication protocol DS-301 and in device profile DSP-40x.

5.10.10.1.31 Emergency Inhibit Time

The *Emergency Inhibit Time* parameter defines an interval in ms that must lie between the repeated transmission of emergency objects.

5.10.10.1.32 CANopen Transmit PDOs 1-4

The process data objects (PDOs) sent by the device are configured in the *CANopen Transmit PDOs 1-4* group. The values entered must correspond to the CANopen specification.

- PREDEF. CONN.: A YES value indicates that the identifier for the Predefined Connection Set should be used for the CANopen specification. The configured COB identifiers are ignored.
- MAP OBJ.1-8: Up to eight objects can be mapped for each TPDO. If the number of mapping objects specified is lower, the last entries in the table are ignored.



Validation of the values entered does not take place via the SOPAS ET user interface. If invalid or inconsistent values have been entered, an incorrect entry is created during PDO initialization.

5.10.10.1.33 CANopen Receive PDOs 1-4

The process data objects (PDOs) received by the device are configured in the [CANopen Receive PDOs 1-4](#) group. The values entered must correspond to the CANopen specification.

- PREDEF. CONN.: A YES value indicates that the identifier for the Predefined Connection Set should be used for the CANopen specification. The configured COB identifiers are ignored.
- MAP OBJ.1-8: Up to eight objects can be mapped for each RPDO. If the number of mapping objects specified is lower, the last entries in the table are ignored.

 Validation of the values entered does not take place via the SOPAS ET user interface. If invalid or inconsistent values have been entered, an incorrect entry is created during PDO initialization.

5.10.11 Fieldbus gateway

All settings for connecting the reading device to a gateway are made on the [Fieldbus gateway](#) tab. The gateway can be configured directly via the reading device. The gateway and reading device communicate via the serial interface.

The reading device is incorporated into the required fieldbus via a gateway.

5.10.11.1 Profibus proxy CDF600

The [Profibus proxy CDF600](#) group contains the parameters for establishing a connection between the reading gate and the Profibus proxy CDF600.

A typical cable layout when using the CDF600 is shown below.



5.10.11.1.1 Slave address

The slave address of the Profibus participant on which the reading device is connected in the Profibus is displayed under the [Slave address](#) parameter. The slave address is set via the rotary switch on the CDF600.

5.10.11.1.2 Communication protocol

The communication modes for exchanging data between the reading device and the fieldbus master are selected via the *Communication protocol* parameter.

- **OPERATING MODE BMV10:** This mode is compatible with the data communication via the SICK bus connection module BMV10/BMH10. For this, the reading device is connected to a Profibus proxy CDF600 while other participants in the fieldbus communicate via the bus connection module BMV10. No output states are transferred. The fieldbus module's digital switching inputs and outputs can be used for the reading device regardless.
- **CMF400 OPERATING MODE WITHOUT HANDSHAKE:** This mode is compatible with the data communication of the fieldbus module CMF400-1x01. For this, the reading device is connected to a Profibus proxy CDF600 while other participants in the fieldbus communicate via the fieldbus module CMF400. This mode operates without byte handshake and is used for test purposes during data transmission. The timeout of the handshake is deactivated.
- **CMF400 OPERATING MODE:** This mode is compatible with the data communication of the fieldbus module CMF400-1x01. It is used for exchanging data during normal operation when used in existing systems. For this, the reading device is connected to a Profibus proxy CDF600 while other participants in the fieldbus communicate via the fieldbus module CMF400.
- **OPERATING MODE CDF600:** This mode is used for exchanging data during normal operation in new reading device installations. The fieldbus module operates as a proxy.
- **OPERATING MODE CDF600 WITHOUT HANDSHAKE:** This mode functions without a byte handshake and is used for testing data transmission. The timeout of the handshake is deactivated.



Individual parameter values are not supported by all device types.

The communication modes without handshake are used when the application does not require monitoring.

5.10.11.1.3 Protocol/Output format

An output format is assigned to the interface via the *Protocol/Output format* parameter.



Output formats are defined under *Data processing*.

5.10.11.1.4 Using PLC output bit 0

Use of the PLC output is determined via the *Using PLC output bit 0* parameter.

- **NO FUNCTION:** The PLC output is not used.
- **OBJECT TRIGGER:** A digital signal is sent via the PLC output. To use this function, the OBJECT TRIGGER CONTROL parameter must be set to the FIELDBUS INPUT value. The OBJECT TRIGGER CONTROL parameter is located under READING CONFIGURATION -> OBJECT TRIGGER CONTROL.
- **SET/RESET EXTERNAL OUTPUT 1/2:** External output 1 or 2 is reset. To use this function, the EXTERNAL OUTPUT 1 or EXTERNAL OUTPUT 2 parameter must be set to the FIELDBUS INPUT value. The EXTERNAL OUTPUT 1/2 parameters can be found under NETWORK/INTERFACES/IOS -> DIGITAL OUTPUTS/BEEPER.
- **DISTANCE CONFIGURATION BIT 0/1:** The DYNAMIC FOCUS parameter is controlled via the PLC output. To use this function, the DYN. CON. MODE parameter must be set to the FIELDBUS value.

The parameter is used in connection with the CMF400 operating mode to assign properties to PLC outputs. As there is no opportunity to use triggers during the CMF400 communication mode, instructions can be issued via the output properties.

5.10.11.1.5 Using PLC output bit 1

Use of the PLC output is determined via the [Using PLC output bit 1](#) parameter.

- NO FUNCTION: The PLC output is not used.
- OBJECT TRIGGER: A digital signal is sent via the PLC output. To use this function, the OBJECT TRIGGER CONTROL parameter must be set to the FIELD BUS INPUT value. The OBJECT TRIGGER CONTROL parameter is located under READING CONFIGURATION -> OBJECT TRIGGER CONTROL.
- SET/RESET EXTERNAL OUTPUT 1/2: External output 1 or 2 is reset. To use this function, the EXTERNAL OUTPUT 1 or EXTERNAL OUTPUT 2 parameter must be set to the FIELD BUS INPUT value. The EXTERNAL OUTPUT 1/2 parameters can be found under NETWORK/INTERFACES/IOS -> DIGITAL OUTPUTS/BEEPER.
- DISTANCE CONFIGURATION BIT 0/1: The DYNAMIC FOCUS parameter is controlled via the PLC output. To use this function, the DYN. CON. MODE parameter must be set to the FIELD BUS value.

The parameter is used in connection with the CMF400 operating mode to assign properties to PLC outputs. As there is no opportunity to use triggers during the CMF400 communication mode, instructions can be issued via the output properties.

5.10.11.1.6 Using PLC input bit 0

Use of the PLC input is determined via the [Using PLC input bit 0](#) parameter.

- NO FUNCTION: The PLC input is not used.
- DEVICE READY
- GOOD READ
- NO READ
- STATUS OF INPUT SENSOR 1
- STATUS OF INPUT SENSOR 2
- STATUS OF EXTERNAL INPUT 1
- STATUS OF EXTERNAL INPUT 2

The parameter is used in connection with the CMF400 operating mode to assign properties to PLC inputs. As there is no opportunity to use triggers during the CMF400 communication mode, instructions can be issued via the input properties.

5.10.11.1.7 Using PLC input bit 1

Use of the PLC input is determined via the [Using PLC input bit 1](#) parameter.

- NO FUNCTION: The PLC input is not used.
- DEVICE READY
- GOOD READ

- NO READ
- STATUS OF INPUT SENSOR 1
- STATUS OF INPUT SENSOR 2
- STATUS OF EXTERNAL INPUT 1
- STATUS OF EXTERNAL INPUT 2

The parameter is used in connection with the CMF400 operating mode to assign properties to PLC inputs. As there is no opportunity to use triggers during the CMF400 communication mode, instructions can be issued via the input properties.

5.10.11.2 Profibus/DeviceNet/Profinet Gateway CMF400/CDM425

Via the parameters for the *Profibus/DeviceNet/Profinet Gateway CMF400/CDM425* group, gateways are selected for interfaces that are used to integrate the reading device into the required fieldbus system or Industrial Ethernet standard. The selected gateway must correspond to the features of the connection module in use.



It is recommended to connect the gateway to the host interface and only use the auxiliary interface if 2 gateways are connected to the connection module CDM4xx.

5.10.11.2.1 Serial auxiliary interface.

The fieldbus gateway used on the serial auxiliary interface via the connection module CDM4xx is specified under the *Serial auxiliary interface* parameter.

- NONE: There is no connection module connected to the fieldbus gateway.
- PROFIBUS DP: The connection module CMD4xx is fitted with a Profibus CMF card that acts as a gateway into the Profibus DP fieldbus system.
- DEVICENET: The connection module CMD4xx is fitted with a DeviceNet CMF card that acts as a gateway into the DeviceNet fieldbus system.



To use the interface, the switch on the gateway must be in the AUX position.

5.10.11.2.2 Serial host interface.

The fieldbus gateway used on the serial host interface via the connection module CDM4xx is specified under the *Serial host interface* parameter.

- NONE: There is no connection module connected to the fieldbus gateway.
- PROFIBUS DP: The connection module CMD4xx is fitted with a Profibus CMF card that acts as a gateway into the Profibus DP fieldbus system.
- DEVICENET: The connection module CMD4xx is fitted with a DeviceNet CMF card that acts as a gateway into the DeviceNet fieldbus system.
- PROFINET IO: The device is connected to the connection module CDM425 with Profinet IO gateway, which integrates the reading device into the Industrial Ethernet standard Profinet IO.



To use the interface, the switch on the gateway must be in the AUX position.

5.10.11.3 Profibus DP gateway

The *Profibus DP gateway* group contains the parameters for the Profibus settings.

5.10.11.3.1 Using device ID as a Profibus address

If the *Using device ID as a Profibus address* parameter is activated, the device ID is used as a Profibus address.

5.10.11.3.2 Device ID

The device ID is displayed under the *Device ID* parameter. If the device ID is activated for use as a fieldbus address, a new device ID can be assigned.

5.10.11.3.3 Operating mode

A data transmission mode is selected via the *Operating mode* parameter.

- **MODE1: CMF STANDARD FORMAT / BYTE HANDSHAKE / DIGITAL IO:** In this transmission mode, acknowledging the telegrams ensures that no telegrams are lost. This mode is recommended for high object frequencies and long telegrams.
- **MODE2: CMF STANDARD FORMAT / NO BYTE HANDSHAKE / DIGITAL IO:** No telegrams are acknowledged in this transmission mode. It is possible to lose telegrams if the PLC cycle time is too slow. The mode should only be used for low object frequencies and short telegrams.
- **MODE 3: BMV-COMPATIBLE/NO DIGITAL IO:** The mode is compatible with BMV/H10-0111. The mode should only be selected if it is necessary to replace a BMV/H.



Individual parameter values are not supported by all device types.

5.10.11.4 DeviceNet gateway

The *DeviceNet gateway* group contains the parameters for DeviceNet settings.

5.10.11.4.1 Using device ID as a DeviceNet address

If this parameter is activated, the device ID is used as a DeviceNet address.

5.10.11.4.2 Device ID

The device ID is displayed under the *Device ID* parameter. If the device ID is activated for use as a fieldbus address, a new device ID can be assigned.

5.10.11.4.3 Operating mode

A data transmission mode is selected via the *Operating mode* parameter.

- **MODE1: CMF STANDARD FORMAT / BYTE HANDSHAKE / DIGITAL IO:** In this transmission mode, acknowledging the telegrams ensures that no telegrams are lost. This mode is recommended for high object frequencies and long telegrams.
- **MODE2: CMF STANDARD FORMAT / NO BYTE HANDSHAKE / DIGITAL IO:** No telegrams are acknowledged in this transmission mode. It is possible to lose telegrams if the PLC cycle time is too slow. The mode should only be used for low object frequencies and short telegrams.
- **MODE 3: BMV-COMPATIBLE/NO DIGITAL IO:** The mode is compatible with BMV/H10-0111. The mode should only be selected if it is necessary to replace a BMV/H.



Individual parameter values are not supported by all device types.

5.10.11.4.4 Data transmission rate

The maximum transmission rate can be adjusted via the *Data transmission rate* parameter.

The value is provided in bits/s.

5.10.11.4.5 Useful input data length

The length of the input data for the PLC or master is determined via the *Useful input data length* parameter.

5.10.11.4.6 Useful output data length

The length of the output data to the PLC or master is determined via the *Useful output data length* parameter.

5.10.11.5 Profinet IO gateway CDM425-PN

The *Profinet IO gateway CDM425-PN* group contains the parameters for the Profinet IO settings.

5.10.11.5.1 DHCP

If the *DHCP* parameter is activated, a DHCP server from the corporate network automatically provides the gateway with an IP address, subnet mask, and default gateway address. The addresses must not be entered manually.

5.10.11.5.2 IP address

The Ethernet address for the Profinet IO gateway is set manually via the *IP address* parameter.

5.10.11.5.3 Subnet mask

The bitmask or prefix mask for network protocol IPv4/IPv6 is manually defined via the *Subnet mask* parameter.

5.10.11.5.4 Default gateway

The default gateway address is defined manually via the *Default gateway* parameter. The default gateway redirects all network activity that does not belong to this subnetwork into another subnetwork.

5.10.11.5.5 Station name

A name for the PROFINET IO gateway is defined via the *Station name* parameter.

The following naming conventions apply:

- The station name consists of letters, digits, a dash or period, and must not exceed 127 characters.
- A single component of the device name, i.e., a string of characters between two periods, must not exceed 63 characters.
- With the exception of hyphens, special characters must not be used.
- The device name must not start or end with a hyphen or period.

- The device name must not begin with digits.
- The device name must not be in the form of n.n.n.n (where n is a number between 0 and 999).
- The device name must not start with the character sequence "port-xyz-" (where x,y,z are numbers between 0 and 9).

5.10.12 Digital inputs

The *Digital inputs* tab contains all of the parameters for configuring the digital inputs.

These parameters define the signal that causes the input to respond along with its time delay.



The required input signal function must be configured accordingly in another area. The use of input signals as object triggers, for example, is set on the *Object trigger control* tab.

5.10.12.1 Sensor / Result 1

The *Sensor / Result 1* group contains the parameters for modifying digital input 1.

5.10.12.1.1 Control

The *Control* parameter is used to choose between configuration with units of length or configuration with units of time.

- TRACK CONTROLLED: The configuration is based on track increments in mm that are either generated by the configuration software (SOPAS) or via the CAN interfaces or are generated via an incremental encoder on the digital input.
- TIME CONTROLLED: Configuration is based on time information in ms.

5.10.12.1.2 Sensitivity

The *Sensitivity* parameter is used to choose between edge control and level control.

- EDGE: Depending on the configuration of the *Logic* parameter, the digital input reacts to the change between Low (0) and High (1). A permanently present signal is ignored.
- LEVEL: Depending on the configuration of the *Logic* parameter, the digital input reacts to a permanently present signal.

5.10.12.1.3 Logic

The *Logic* parameter is used to select the polarity for the *Sensitivity* parameter.

- NOT INVERTED: The digital input reacts to a rising edge, i.e. to the change from Low (0) to High (1) or a permanent High signal (1).
- INVERTED: The digital input reacts to a falling edge, i.e. to the change from High (1) to Low (0) or a permanent Low signal (0).

5.10.12.1.4 Debouncing

Depending on the configuration of the *Control* parameter, the *Debouncing* parameter is used to specify the distance in mm or time in ms for which the signal must at least be present at the external input. If the value is not achieved, the signal is not identified.

Depending on the application, this value must be modified in accordance with the required reading pulse rate.

5.10.12.2 Sensor / Result 2

The *Sensor / Result 2* group contains the parameters for modifying digital input 2.

5.10.12.2.1 Control

The *Control* parameter is used to choose between configuration with units of length or configuration with units of time.

- TRACK CONTROLLED: The configuration is based on track increments in mm that are either generated by the configuration software (SOPAS) or via the CAN interfaces or are generated via an incremental encoder on the digital input.
- TIME CONTROLLED: Configuration is based on time information in ms.

5.10.12.2.2 Sensitivity

The *Sensitivity* parameter is used to choose between edge control and level control.

- EDGE: Depending on the configuration of the *Logic* parameter, the digital input reacts to the change between Low (0) and High (1). A permanently present signal is ignored.
- LEVEL: Depending on the configuration of the *Logic* parameter, the digital input reacts to a permanently present signal.

5.10.12.2.3 Logic

The *Logic* parameter is used to select the polarity for the *Sensitivity* parameter.

- NOT INVERTED: The digital input reacts to a rising edge, i.e. to the change from Low (0) to High (1) or a permanent High signal (1).
- INVERTED: The digital input reacts to a falling edge, i.e. to the change from High (1) to Low (0) or a permanent Low signal (0).

5.10.12.2.4 Debouncing

Depending on the configuration of the *Control* parameter, the *Debouncing* parameter is used to specify the distance in mm or time in ms for which the signal must at least be present at the external input. If the value is not achieved, the signal is not identified.

Depending on the application, this value must be modified in accordance with the required reading pulse rate.

5.10.12.3 External Input 1

The connection module CDB620 or CDM420 in conjunction with the parameter memory module CMC600 can be used to extend the number of available digital inputs. The parameter memory module CMC is used for input extension and converts a digital signal into a command. External digital input react more slowly than the internal digital inputs of the LECTOR®620.

The *External Input 1* group contains the parameters for modifying external digital input 1.

5.10.12.3.1 Control

The *Control* parameter is used to choose between configuration with units of length or configuration with units of time.

- TRACK CONTROLLED: The configuration is based on track increments in mm that are either generated by the configuration software (SOPAS) or via the CAN interfaces or are generated via an incremental encoder on the digital input.
- TIME CONTROLLED: Configuration is based on time information in ms.

5.10.12.3.2 Sensitivity

The *Sensitivity* parameter is used to choose between edge control and level control.

- EDGE: Depending on the configuration of the *Logic* parameter, the digital input reacts to the change between Low (0) and High (1). A permanently present signal is ignored.
- LEVEL: Depending on the configuration of the *Logic* parameter, the digital input reacts to a permanently present signal.

5.10.12.3.3 Logic

The *Logic* parameter is used to select the polarity for the *Sensitivity* parameter.

- NOT INVERTED: The digital input reacts to a rising edge, i.e. to the change from Low (0) to High (1) or a permanent High signal (1).
- INVERTED: The digital input reacts to a falling edge, i.e. to the change from High (1) to Low (0) or a permanent Low signal (0).

5.10.12.3.4 Debouncing

Depending on the configuration of the *Control* parameter, the *Debouncing* parameter is used to specify the distance in mm or time in ms for which the signal must at least be present at the external input. If the value is not achieved, the signal is not identified.

Depending on the application, this value must be modified in accordance with the required reading pulse rate.

5.10.12.4 External Input 2

The connection module CDB620 or CDM420 in conjunction with the parameter memory module CMC600 can be used to extend the number of available digital inputs. The parameter memory module CMC is used for input extension and converts a digital signal into a command. External digital input react more slowly than the internal digital inputs of the LECTOR®620.

The *External Input 2* group contains the parameters for modifying external digital input 1.

5.10.12.4.1 Control

The *Control* parameter is used to choose between configuration with units of length or configuration with units of time.

- TRACK CONTROLLED: The configuration is based on track increments in mm that are either generated by the configuration software (SOPAS) or via the CAN interfaces or are generated via an incremental encoder on the digital input.
- TIME CONTROLLED: Configuration is based on time information in ms.

5.10.12.4.2 Sensitivity

The *Sensitivity* parameter is used to choose between edge control and level control.

- EDGE: Depending on the configuration of the *Logic* parameter, the digital input reacts to the change between Low (0) and High (1). A permanently present signal is ignored.
- LEVEL: Depending on the configuration of the *Logic* parameter, the digital input reacts to a permanently present signal.

5.10.12.4.3 Logic

The *Logic* parameter is used to select the polarity for the *Sensitivity* parameter.

- NOT INVERTED: The digital input reacts to a rising edge, i.e. to the change from Low (0) to High (1) or a permanent High signal (1).
- INVERTED: The digital input reacts to a falling edge, i.e. to the change from High (1) to Low (0) or a permanent Low signal (0).

5.10.12.4.4 Debouncing

Depending on the configuration of the *Control* parameter, the *Debouncing* parameter is used to specify the distance in mm or time in ms for which the signal must at least be present at the external input. If the value is not achieved, the signal is not identified.

Depending on the application, this value must be modified in accordance with the required reading pulse rate.

5.10.13 Digital outputs/Beeper

The *Digital outputs/Beeper* tab contains all of the parameters for configuring the digital outputs.

These parameters define which events cause an output signal to be sent.



The required output signal function must be configured accordingly in another area. The use of output signals as object triggers, for example, is set on the *Application counters* tab.

5.10.13.1 Output / Result 1

The *Output / Result 1* group contains the parameters for modifying digital output 1.

5.10.13.1.1 Active

The *Active* parameter is used to select the event for which the digital output is activated.

- SOPAS COMMAND: The output is activated by a command. As a result, the defined statuses can be displayed via a connected control lamp as a result of a controller (PLC) command.
- DEVICE READY: The output is activated as soon as the LECTOR®620 is operational.
- GOOD READ: The output is activated in the event of a successful reading.
- NO READ: The output is activated in the event of an unsuccessful reading.
- TEACH-IN 1 OK: The output is activated if the MATCHCODE TEACH-IN 1 was taught in successfully.
- TEACH-IN 1 NOK: The output is activated if the MATCHCODE TEACH-IN 1 was not taught in successfully.
- TEACH-IN 2 OK: The output is activated if the MATCHCODE TEACH-IN 2 was taught in successfully.
- TEACH-IN 2 NOK: The output is activated if the MATCHCODE TEACH-IN 2 was not taught in successfully.
- CONDITION MATCH 1: The output is activated if the read code corresponds to the matchcode from CONDITION MATCH 1.
- CONDITION MULTCODES1: The output is activated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- CONDITION TEACHIN1: The output is activated if the read code corresponds to the matchcode from CONDITION TEACHIN1.
- CONDITION TEACHIN2: The output is activated if the read code corresponds to the matchcode from CONDITION TEACHIN2.

5.10.13.1.2 Function (Off)

The *Function (Off)* parameter is used to select the event for which the digital output is deactivated.

- TIMER/TRACKING: The output is deactivated after a certain period of time (*Length*).
- NEXT OBJECT TRIGGER: The output is deactivated as soon as the next reading gate is opened.
- SOPAS COMMAND: The output is deactivated by a command. As a result, the defined statuses can be displayed via a connected control lamp as a result of a controller (PLC) command.

- GOOD READ: The output is deactivated in the event of a successful reading.
- NO READ: The output is deactivated in the event of an unsuccessful reading.
- TEACH-IN 1 START: The output is deactivated as soon as the processing for teaching in the MATCHCODE TEACH-IN 1 starts.
- TEACH-IN 2 START: The output is deactivated as soon as the processing for teaching in the MATCHCODE TEACH-IN 2 starts.
- CONDITION MATCH 1: The output is deactivated if the read code corresponds to the match-code from CONDITION MATCH 1.
- CONDITION MULTCODES1: The output is deactivated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- CONDITION TEACHIN1: The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN1.
- CONDITION TEACHIN2: The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN2.

5.10.13.1.3 Or

You can use the *Or* parameter to choose a second condition for deactivating the digital output.

One of the conditions must be met for the digital output to be deactivated.

- --: Second condition not selected.
- CONDITION MATCH 1: The output is deactivated if the read code corresponds to the match-code from CONDITION MATCH 1.
- CONDITION MULTCODES1: The output is deactivated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- CONDITION TEACHIN1: The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN1.
- CONDITION TEACHIN2: The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN2.

5.10.13.1.4 Logic

The *Logic* parameter is used to select the polarity of the digital output.

- NOT INVERTED: On activation, the digital output switches from Low (0) to High (1).
- INVERTED: On activation, the digital output switches from High (1) to Low (0).

5.10.13.1.5 Control

The *Control* parameter is used to choose between configuration with units of length or configuration with units of time.

- TRACK CONTROLLED: The entries for the *Length* parameter are made in mm.
- TIME CONTROLLED: The entries for the *Length* parameter are made in ms.

5.10.13.1.6 Length

If the **TIMER/TRACKING** condition is selected under the *Function (Off)* parameter, you can use the *Length* parameter to enter the duration for which the digital output remains activated. The unit can be selected in the *Control* parameter.

5.10.13.2 Output/Result 2

The *Output/Result 2* group contains the parameters for adjusting digital output 2.

5.10.13.2.1 Active

The event that activates the digital output is selected via the *Active* parameter.

- **SOPAS COMMAND:** The output is activated via a command. This allows defined statuses to be displayed by a connected indicator lamp via a corresponding command from the control (PLC).
- **DEVICE READY:** The output is activated as soon as the LECTOR®620 is ready for use.
- **GOOD READ:** The output is activated when the read is successful.
- **NO READ:** The output is activated when the read is unsuccessful.
- **TEACH-IN 1 OK:** The output is activated when the MATCH CODE TEACH-IN 1 has been successfully taught in.
- **TEACH-IN 1 NOK:** The output is activated if the MATCH CODE TEACH-IN 1 has not been successfully taught in.
- **TEACH-IN 2 OK:** The output is activated when the MATCH CODE TEACH-IN 2 has been successfully taught in.
- **TEACH-IN 2 NOK:** The output is activated if the MATCH CODE TEACH-IN 2 has not been successfully taught in.
- **EXTERNAL LIGHTING:** External lighting is connected to the output and can be switched on and off by the device.
- **MATCH 1 CONDITION:** The output is activated when the code that has been read corresponds to the MATCH 1 CONDITION match code.
- **MULTICODES1 CONDITION:** The output is activated when the number of codes that have been read corresponds to the number specified by the MULTICODES1 CONDITION.
- **TEACH-IN1 CONDITION:** The output is activated when the code that has been read corresponds to the TEACH-IN1 CONDITION match code.
- **TEACH-IN2 CONDITION:** The output is activated when the code that has been read corresponds to the TEACH-IN2 CONDITION match code.



The behavior of the external lighting and the switch type is set under *Illumination*.

5.10.13.2.2 Function (Off)

The *Function (Off)* parameter is used to select the event for which the digital output is deactivated.

- **TIMER/TRACKING:** The output is deactivated after a certain period of time (*Length*).
- **NEXT OBJECT TRIGGER:** The output is deactivated as soon as the next reading gate is opened.

- **SOPAS COMMAND:** The output is deactivated by a command. As a result, the defined statuses can be displayed via a connected control lamp as a result of a controller (PLC) command.
- **GOOD READ:** The output is deactivated in the event of a successful reading.
- **NO READ:** The output is deactivated in the event of an unsuccessful reading.
- **TEACH-IN 1 START:** The output is deactivated as soon as the processing for teaching in the MATCHCODE TEACH-IN 1 starts.
- **TEACH-IN 2 START:** The output is deactivated as soon as the processing for teaching in the MATCHCODE TEACH-IN 2 starts.
- **CONDITION MATCH 1:** The output is deactivated if the read code corresponds to the match-code from CONDITION MATCH 1.
- **CONDITION MULTCODES1:** The output is deactivated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- **CONDITION TEACHIN1:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN1.
- **CONDITION TEACHIN2:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN2.

5.10.13.2.3 Or

You can use the *Or* parameter to choose a second condition for deactivating the digital output.

One of the conditions must be met for the digital output to be deactivated.

- **---**: Second condition not selected.
- **CONDITION MATCH 1:** The output is deactivated if the read code corresponds to the match-code from CONDITION MATCH 1.
- **CONDITION MULTCODES1:** The output is deactivated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- **CONDITION TEACHIN1:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN1.
- **CONDITION TEACHIN2:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN2.

5.10.13.2.4 Logic

The *Logic* parameter is used to select the polarity of the digital output.

- **NOT INVERTED:** On activation, the digital output switches from Low (0) to High (1).
- **INVERTED:** On activation, the digital output switches from High (1) to Low (0).

5.10.13.2.5 Control

The *Control* parameter is used to choose between configuration with units of length or configuration with units of time.

- TRACK CONTROLLED: The entries for the *Length* parameter are made in mm.
- TIME CONTROLLED: The entries for the *Length* parameter are made in ms.

5.10.13.2.6 Length

If the TIMER/TRACKING condition is selected under the *Function (Off)* parameter, you can use the *Length* parameter to enter the duration for which the digital output remains activated. The unit can be selected in the *Control* parameter.

5.10.13.3 Output / Result 3

The *Output / Result 3* group contains the parameters for modifying digital output 3.



Digital outputs 3 and 4 are not available via the 15-pin D-Sub connection on connection modules CDB620 and CDM420 and can only be used in conjunction with a 17-pole line with open ends.

5.10.13.3.1 Active

The *Active* parameter is used to select the event for which the digital output is activated.

- SOPAS COMMAND: The output is activated by a command. As a result, the defined statuses can be displayed via a connected control lamp as a result of a controller (PLC) command.
- DEVICE READY: The output is activated as soon as the LECTOR®620 is operational.
- GOOD READ: The output is activated in the event of a successful reading.
- NO READ: The output is activated in the event of an unsuccessful reading.
- TEACH-IN 1 OK: The output is activated if the MATCHCODE TEACH-IN 1 was taught in successfully.
- TEACH-IN 1 NOK: The output is activated if the MATCHCODE TEACH-IN 1 was not taught in successfully.
- TEACH-IN 2 OK: The output is activated if the MATCHCODE TEACH-IN 2 was taught in successfully.
- TEACH-IN 2 NOK: The output is activated if the MATCHCODE TEACH-IN 2 was not taught in successfully.
- CONDITION MATCH 1: The output is activated if the read code corresponds to the matchcode from CONDITION MATCH 1.
- CONDITION MULTCODES1: The output is activated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- CONDITION TEACHIN1: The output is activated if the read code corresponds to the matchcode from CONDITION TEACHIN1.
- CONDITION TEACHIN2: The output is activated if the read code corresponds to the matchcode from CONDITION TEACHIN2.

5.10.13.3.2 Function (Off)

The *Function (Off)* parameter is used to select the event for which the digital output is deactivated.

- **TIMER/TRACKING:** The output is deactivated after a certain period of time (*Length*).
- **NEXT OBJECT TRIGGER:** The output is deactivated as soon as the next reading gate is opened.
- **SOPAS COMMAND:** The output is deactivated by a command. As a result, the defined statuses can be displayed via a connected control lamp as a result of a controller (PLC) command.
- **GOOD READ:** The output is deactivated in the event of a successful reading.
- **NO READ:** The output is deactivated in the event of an unsuccessful reading.
- **TEACH-IN 1 START:** The output is deactivated as soon as the processing for teaching in the MATCHCODE TEACH-IN 1 starts.
- **TEACH-IN 2 START:** The output is deactivated as soon as the processing for teaching in the MATCHCODE TEACH-IN 2 starts.
- **CONDITION MATCH 1:** The output is deactivated if the read code corresponds to the match-code from CONDITION MATCH 1.
- **CONDITION MULTCODES1:** The output is deactivated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- **CONDITION TEACHIN1:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN1.
- **CONDITION TEACHIN2:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN2.

5.10.13.3.3 Or

You can use the *Or* parameter to choose a second condition for deactivating the digital output.

One of the conditions must be met for the digital output to be deactivated.

- **--:** Second condition not selected.
- **CONDITION MATCH 1:** The output is deactivated if the read code corresponds to the match-code from CONDITION MATCH 1.
- **CONDITION MULTCODES1:** The output is deactivated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- **CONDITION TEACHIN1:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN1.
- **CONDITION TEACHIN2:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN2.

5.10.13.3.4 Logic

The *Logic* parameter is used to select the polarity of the digital output.

- **NOT INVERTED:** On activation, the digital output switches from Low (0) to High (1).
- **INVERTED:** On activation, the digital output switches from High (1) to Low (0).

5.10.13.3.5 Control

The *Control* parameter is used to choose between configuration with units of length or configuration with units of time.

- TRACK CONTROLLED: The entries for the *Length* parameter are made in mm.
- TIME CONTROLLED: The entries for the *Length* parameter are made in ms.

5.10.13.3.6 Length

If the TIMER/TRACKING condition is selected under the *Function (Off)* parameter, you can use the *Length* parameter to enter the duration for which the digital output remains activated. The unit can be selected in the *Control* parameter.

5.10.13.4 Output / Result 4

The *Output / Result 4* group contains the parameters for modifying digital output 4.



Digital outputs 3 and 4 are not available via the 15-pin D-Sub connection on connection modules CDB620 and CDM420 and can only be used in conjunction with a 17-pole line with open ends.

5.10.13.4.1 Active

The *Active* parameter is used to select the event for which the digital output is activated.

- SOPAS COMMAND: The output is activated by a command. As a result, the defined statuses can be displayed via a connected control lamp as a result of a controller (PLC) command.
- DEVICE READY: The output is activated as soon as the LECTOR®620 is operational.
- GOOD READ: The output is activated in the event of a successful reading.
- NO READ: The output is activated in the event of an unsuccessful reading.
- TEACH-IN 1 OK: The output is activated if the MATCHCODE TEACH-IN 1 was taught in successfully.
- TEACH-IN 1 NOK: The output is activated if the MATCHCODE TEACH-IN 1 was not taught in successfully.
- TEACH-IN 2 OK: The output is activated if the MATCHCODE TEACH-IN 2 was taught in successfully.
- TEACH-IN 2 NOK: The output is activated if the MATCHCODE TEACH-IN 2 was not taught in successfully.
- CONDITION MATCH 1: The output is activated if the read code corresponds to the matchcode from CONDITION MATCH 1.
- CONDITION MULTCODES1: The output is activated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- CONDITION TEACHIN1: The output is activated if the read code corresponds to the matchcode from CONDITION TEACHIN1.
- CONDITION TEACHIN2: The output is activated if the read code corresponds to the matchcode from CONDITION TEACHIN2.

5.10.13.4.2 Function (Off)

The *Function (Off)* parameter is used to select the event for which the digital output is deactivated.

- **TIMER/TRACKING:** The output is deactivated after a certain period of time (*Length*).
- **NEXT OBJECT TRIGGER:** The output is deactivated as soon as the next reading gate is opened.
- **SOPAS COMMAND:** The output is deactivated by a command. As a result, the defined statuses can be displayed via a connected control lamp as a result of a controller (PLC) command.
- **GOOD READ:** The output is deactivated in the event of a successful reading.
- **NO READ:** The output is deactivated in the event of an unsuccessful reading.
- **TEACH-IN 1 START:** The output is deactivated as soon as the processing for teaching in the MATCHCODE TEACH-IN 1 starts.
- **TEACH-IN 2 START:** The output is deactivated as soon as the processing for teaching in the MATCHCODE TEACH-IN 2 starts.
- **CONDITION MATCH 1:** The output is deactivated if the read code corresponds to the match-code from CONDITION MATCH 1.
- **CONDITION MULTCODES1:** The output is deactivated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- **CONDITION TEACHIN1:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN1.
- **CONDITION TEACHIN2:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN2.

5.10.13.4.3 Or

You can use the *Or* parameter to choose a second condition for deactivating the digital output.

One of the conditions must be met for the digital output to be deactivated.

- **--:** Second condition not selected.
- **CONDITION MATCH 1:** The output is deactivated if the read code corresponds to the match-code from CONDITION MATCH 1.
- **CONDITION MULTCODES1:** The output is deactivated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- **CONDITION TEACHIN1:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN1.
- **CONDITION TEACHIN2:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN2.

5.10.13.4.4 Logic

The *Logic* parameter is used to select the polarity of the digital output.

- **NOT INVERTED:** On activation, the digital output switches from Low (0) to High (1).
- **INVERTED:** On activation, the digital output switches from High (1) to Low (0).

5.10.13.4.5 Control

The *Control* parameter is used to choose between configuration with units of length or configuration with units of time.

- TRACK CONTROLLED: The entries for the *Length* parameter are made in mm.
- TIME CONTROLLED: The entries for the *Length* parameter are made in ms.

5.10.13.4.6 Length

If the TIMER/TRACKING condition is selected under the *Function (Off)* parameter, you can use the *Length* parameter to enter the duration for which the digital output remains activated. The unit can be selected in the *Control* parameter.

5.10.13.5 External Output 1

The connection module CDB620 or CDM420 in conjunction with the parameter memory module CMC600 can be used to extend the number of available digital outputs. The parameter memory module CMC is used for output extension and converts a digital signal into a command. External digital outputs react more slowly than the internal digital outputs of the LECTOR®620.

The *External Output 1* group contains the parameters for modifying external digital output 1.

5.10.13.5.1 Active

The *Active* parameter is used to select the event for which the digital output is activated.

- SOPAS COMMAND: The output is activated by a command. As a result, the defined statuses can be displayed via a connected control lamp as a result of a controller (PLC) command.
- DEVICE READY: The output is activated as soon as the LECTOR®620 is operational.
- GOOD READ: The output is activated in the event of a successful reading.
- NO READ: The output is activated in the event of an unsuccessful reading.
- TEACH-IN 1 OK: The output is activated if the MATCHCODE TEACH-IN 1 was taught in successfully.
- TEACH-IN 1 NOK: The output is activated if the MATCHCODE TEACH-IN 1 was not taught in successfully.
- TEACH-IN 2 OK: The output is activated if the MATCHCODE TEACH-IN 2 was taught in successfully.
- TEACH-IN 2 NOK: The output is activated if the MATCHCODE TEACH-IN 2 was not taught in successfully.
- CONDITION MATCH 1: The output is activated if the read code corresponds to the matchcode from CONDITION MATCH 1.
- CONDITION MULTCODES1: The output is activated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- CONDITION TEACHIN1: The output is activated if the read code corresponds to the matchcode from CONDITION TEACHIN1.
- CONDITION TEACHIN2: The output is activated if the read code corresponds to the matchcode from CONDITION TEACHIN2.

5.10.13.5.2 Function (Off)

The *Function (Off)* parameter is used to select the event for which the digital output is deactivated.

- **TIMER/TRACKING:** The output is deactivated after a certain period of time (*Length*).
- **NEXT OBJECT TRIGGER:** The output is deactivated as soon as the next reading gate is opened.
- **SOPAS COMMAND:** The output is deactivated by a command. As a result, the defined statuses can be displayed via a connected control lamp as a result of a controller (PLC) command.
- **GOOD READ:** The output is deactivated in the event of a successful reading.
- **NO READ:** The output is deactivated in the event of an unsuccessful reading.
- **TEACH-IN 1 START:** The output is deactivated as soon as the processing for teaching in the MATCHCODE TEACH-IN 1 starts.
- **TEACH-IN 2 START:** The output is deactivated as soon as the processing for teaching in the MATCHCODE TEACH-IN 2 starts.
- **CONDITION MATCH 1:** The output is deactivated if the read code corresponds to the match-code from CONDITION MATCH 1.
- **CONDITION MULTCODES1:** The output is deactivated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- **CONDITION TEACHIN1:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN1.
- **CONDITION TEACHIN2:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN2.

5.10.13.5.3 Or

You can use the *Or* parameter to choose a second condition for deactivating the digital output.

One of the conditions must be met for the digital output to be deactivated.

- **--:** Second condition not selected.
- **CONDITION MATCH 1:** The output is deactivated if the read code corresponds to the match-code from CONDITION MATCH 1.
- **CONDITION MULTCODES1:** The output is deactivated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- **CONDITION TEACHIN1:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN1.
- **CONDITION TEACHIN2:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN2.

5.10.13.5.4 Logic

The *Logic* parameter is used to select the polarity of the digital output.

- **NOT INVERTED:** On activation, the digital output switches from Low (0) to High (1).
- **INVERTED:** On activation, the digital output switches from High (1) to Low (0).

5.10.13.5.5 Control

The *Control* parameter is used to choose between configuration with units of length or configuration with units of time.

- TRACK CONTROLLED: The entries for the *Length* parameter are made in mm.
- TIME CONTROLLED: The entries for the *Length* parameter are made in ms.

5.10.13.5.6 Length

If the TIMER/TRACKING condition is selected under the *Function (Off)* parameter, you can use the *Length* parameter to enter the duration for which the digital output remains activated. The unit can be selected in the *Control* parameter.

5.10.13.6 External output 2

The number of available digital outputs can be expanded via the CDB620/CDM420 connection module in conjunction with the CMC600 parameter memory module. The purpose of the CMC parameter memory module is to expand the output and convert a digital signal into a command. External digital outputs generally respond more slowly than the internal digital outputs of the LECTOR®620.

The *External output 2* group contains the parameters for adjusting external digital output 2.

5.10.13.6.1 Active

The *Active* parameter is used to select the event for which the digital output is activated.

- SOPAS COMMAND: The output is activated by a command. As a result, the defined statuses can be displayed via a connected control lamp as a result of a controller (PLC) command.
- DEVICE READY: The output is activated as soon as the LECTOR®620 is operational.
- GOOD READ: The output is activated in the event of a successful reading.
- NO READ: The output is activated in the event of an unsuccessful reading.
- TEACH-IN 1 OK: The output is activated if the MATCHCODE TEACH-IN 1 was taught in successfully.
- TEACH-IN 1 NOK: The output is activated if the MATCHCODE TEACH-IN 1 was not taught in successfully.
- TEACH-IN 2 OK: The output is activated if the MATCHCODE TEACH-IN 2 was taught in successfully.
- TEACH-IN 2 NOK: The output is activated if the MATCHCODE TEACH-IN 2 was not taught in successfully.
- CONDITION MATCH 1: The output is activated if the read code corresponds to the matchcode from CONDITION MATCH 1.
- CONDITION MULTCODES1: The output is activated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- CONDITION TEACHIN1: The output is activated if the read code corresponds to the matchcode from CONDITION TEACHIN1.
- CONDITION TEACHIN2: The output is activated if the read code corresponds to the matchcode from CONDITION TEACHIN2.

5.10.13.6.2 Function (Off)

The *Function (Off)* parameter is used to select the event for which the digital output is deactivated.

- **TIMER/TRACKING:** The output is deactivated after a certain period of time (*Length*).
- **NEXT OBJECT TRIGGER:** The output is deactivated as soon as the next reading gate is opened.
- **SOPAS COMMAND:** The output is deactivated by a command. As a result, the defined statuses can be displayed via a connected control lamp as a result of a controller (PLC) command.
- **GOOD READ:** The output is deactivated in the event of a successful reading.
- **NO READ:** The output is deactivated in the event of an unsuccessful reading.
- **TEACH-IN 1 START:** The output is deactivated as soon as the processing for teaching in the MATCHCODE TEACH-IN 1 starts.
- **TEACH-IN 2 START:** The output is deactivated as soon as the processing for teaching in the MATCHCODE TEACH-IN 2 starts.
- **CONDITION MATCH 1:** The output is deactivated if the read code corresponds to the match-code from CONDITION MATCH 1.
- **CONDITION MULTCODES1:** The output is deactivated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- **CONDITION TEACHIN1:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN1.
- **CONDITION TEACHIN2:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN2.

5.10.13.6.3 Or

You can use the *Or* parameter to choose a second condition for deactivating the digital output.

One of the conditions must be met for the digital output to be deactivated.

- **--:** Second condition not selected.
- **CONDITION MATCH 1:** The output is deactivated if the read code corresponds to the match-code from CONDITION MATCH 1.
- **CONDITION MULTCODES1:** The output is deactivated if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- **CONDITION TEACHIN1:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN1.
- **CONDITION TEACHIN2:** The output is deactivated if the read code corresponds to the match-code from CONDITION TEACHIN2.

5.10.13.6.4 Logic

The *Logic* parameter is used to select the polarity of the digital output.

- **NOT INVERTED:** On activation, the digital output switches from Low (0) to High (1).
- **INVERTED:** On activation, the digital output switches from High (1) to Low (0).

5.10.13.6.5 Control

The *Control* parameter is used to choose between configuration with units of length or configuration with units of time.

- TRACK CONTROLLED: The entries for the *Length* parameter are made in mm.
- TIME CONTROLLED: The entries for the *Length* parameter are made in ms.

5.10.13.6.6 Length

If the TIMER/TRACKING condition is selected under the *Function (Off)* parameter, you can use the *Length* parameter to enter the duration for which the digital output remains activated. The unit can be selected in the *Control* parameter.

5.10.13.7 Beeper

The *Beeper* group contains the parameters for adjusting the acoustic signal.

As with the digital outputs, it is possible to select an event that activates the beeper. As soon as this event occurs, it is fed back acoustically by the beeper.

5.10.13.7.1 Beeper

The *Beeper* parameter is used to select the event for which the beeper is activated.

- GOOD READ: The beeper sounds if the reading was successful.
- NO READ: The beeper sounds if the reading was unsuccessful.
- TEACH-IN 1 OK: The beeper sounds if the MATCHCODE TEACH-IN 1 was taught in successfully.
- TEACH-IN 1 NOK: The beeper sounds if the MATCHCODE TEACH-IN 1 was not taught in successfully.
- TEACH-IN 2 OK: The beeper sounds if the MATCHCODE TEACH-IN 2 was taught in successfully.
- TEACH-IN 2 NOK: The beeper sounds if the MATCHCODE TEACH-IN 2 was not taught in successfully.
- CONDITION MATCH 1 : The beeper sounds if the read code corresponds to the matchcode from CONDITION MATCH 1.
- CONDITION MULTCODES1: The beeper sounds if the number of read codes corresponds to the required number from CONDITION MULTCODES1.
- CONDITION TEACHIN1: The beeper sounds if the read code corresponds to the matchcode from CONDITION TEACHIN1.
- CONDITION TEACHIN2: The beeper sounds if the read code corresponds to the matchcode from CONDITION TEACHIN2.

5.10.13.7.2 Volume

The *Volume* parameter can be used to deactivate the beeper (OFF) or set the volume of the beeper (LOUD or SOFT).

As a result, the beeper can be adapted to background noise.

6 Service

Operating data along with status and version information for diagnostic and analysis purposes can be retrieved from the [Service](#) folder.

6.1 Operating data

The data relating to the device, operation, and service and maintenance is combined on the [Operating data](#) tab.

6.1.1 Device information

The identification data of the LECTOR®620 is indicated via the parameters of the [Device information](#) group.

This data is important for service work.

6.1.1.1 Manufacturer

The manufacturer of the LECTOR®620 is indicated in the [Manufacturer](#) display field.

6.1.1.2 Device type

The device type of the LECTOR®620 is indicated in the [Device type](#) display field.

6.1.1.3 Software version

The version of the installed firmware is indicated in the [Software version](#) display field.

6.1.1.4 Order number

The order number of the LECTOR®620 is indicated in the [Order number](#) display field.

6.1.1.5 Serial number

The serial number of the LECTOR®620 is indicated in the [Serial number](#) display field.

6.1.2 Operating data

Notes on how to operate the device are displayed under [Operating data](#).

- POWER-ON COUNTER: Displays how many times the device has been switched on.
- OPERATING HOURS: Displays the total number of device operating hours.
- POWER-ON TIME: Displays the current operating hours since the last switch-on.

6.1.3 Service information

Service and maintenance data is displayed in the [Service information](#) group. Maintenance intervals can be entered.

- LAST USER: User name of the last user to be logged in
- LAST CONFIGURATION AT: Date and time of the last configuration saved in the device
- LAST MAINTENANCE: Date on which maintenance was last completed
- NEXT MAINTENANCE: Date on which maintenance should next be completed

6.2 System status

Device status information that has been recorded is displayed on the [System status](#) tab. This purpose of this data is to analyze the connections to the device and identify any system errors that arise.

6.2.1 System information

The [System information](#) group displays the messages from the device log file that can be used for connection analysis and diagnostic purposes.

6.3 Reading field/Scanning frequency

Reading field diagrams and scanning frequency diagrams are displayed on the [Reading field/Scanning frequency](#) tab.

The diagrams are important for the application layout and help with the correct alignment and mounting of the device.

7 Analysis

Analysis

7.1 Event monitor

Input signals and output signals can be monitored via the diagrams on the *Event monitor* tab.

7.1.1 Event monitor

Input signals and output signals can be monitored via the diagrams in the *Event monitor* group.

Using the signal visualization, events such as triggering or switching outputs can be tested and the logic and function of signal sequences checked.



For additional information see the F1 help.

Symbol	Function
	Select signal that has to be monitored.
	Open a previously recorded monitoring diagram.
	Save the current monitoring diagram.
	Measure distances within the diagram.
	Start monitoring.
	Interrupt monitoring.
	Stop monitoring.
	Record monitoring.
	Enlarge diagram.
	Set diagram zoom level to 100%.
	Scale down diagram.
	Indicate time axis in diagram.
	Indicate increment axis in diagram.
	Activate/deactivate raster in diagram.
	Indicate values in diagram.

Australia

Phone +61 3 9457 0600
1800 334 802 - tollfree
E-Mail sales@sick.com.au

Belgium/Luxembourg

Phone +32 (0)2 466 55 66
E-Mail info@sick.be

Brasil

Phone +55 11 3215-4900
E-Mail sac@sick.com.br

Canada

Phone +1 905 771 14 44
E-Mail information@sick.com

Česká republika

Phone +420 2 57 91 18 50
E-Mail sick@sick.cz

China

Phone +86 4000 121 000
E-Mail info.china@sick.net.cn
Phone +852-2153 6300
E-Mail ghk@sick.com.hk

Danmark

Phone +45 45 82 64 00
E-Mail sick@sick.dk

Deutschland

Phone +49 211 5301-301
E-Mail info@sick.de

España

Phone +34 93 480 31 00
E-Mail info@sick.es

France

Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Great Britain

Phone +44 (0)1727 831121
E-Mail info@sick.co.uk

India

Phone +91-22-4033 8333
E-Mail info@sick-india.com

Israel

Phone +972-4-6881000
E-Mail info@sick-sensors.com

Italia

Phone +39 02 27 43 41
E-Mail info@sick.it

Japan

Phone +81 (0)3 3358 1341
E-Mail support@sick.jp

Magyarország

Phone +36 1 371 2680
E-Mail office@sick.hu

Nederland

Phone +31 (0)30 229 25 44
E-Mail info@sick.nl

Norge

Phone +47 67 81 50 00
E-Mail sick@sick.no

Österreich

Phone +43 (0)22 36 62 28 8-0
E-Mail office@sick.at

Polska

Phone +48 22 837 40 50
E-Mail info@sick.pl

România

Phone +40 356 171 120
E-Mail office@sick.ro

Russia

Phone +7-495-775-05-30
E-Mail info@sick.ru

Schweiz

Phone +41 41 619 29 39
E-Mail contact@sick.ch

Singapore

Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Slovenija

Phone +386 (0)1-47 69 990
E-Mail office@sick.si

South Africa

Phone +27 11 472 3733
E-Mail info@sickautomation.co.za

South Korea

Phone +82 2 786 6321/4
E-Mail info@sickkorea.net

Suomi

Phone +358-9-25 15 800
E-Mail sick@sick.fi

Sverige

Phone +46 10 110 10 00
E-Mail info@sick.se

Taiwan

Phone +886 2 2375-6288
E-Mail sales@sick.com.tw

Türkiye

Phone +90 (216) 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates

Phone +971 (0) 4 88 65 878
E-Mail info@sick.ae

USA/México

Phone +1(952) 941-6780
1 (800) 325-7425 - tollfree
E-Mail info@sickusa.com

More representatives and agencies
at www.sick.com