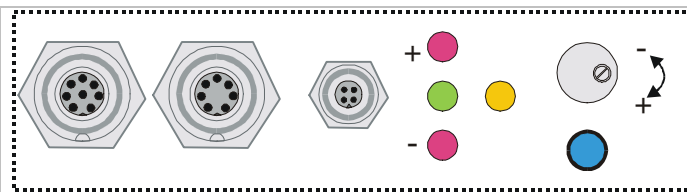


RS232 interface protocol - L-LAS-TB-MS-Scope V4.0

- Standard RS232 serial interface, no hardware handshake
- 3-line connection: GND, TX0, RX0
- Speed: 9600 Baud, 19200 Baud, 38400 Baud, 57600 Baud, or 115200 Baud
- 8 data-bits
- NO parity bit
- 1 stop-bit
- Binary mode.

RS232 CONNECTOR to PC/PLC:



PLC /
POWER

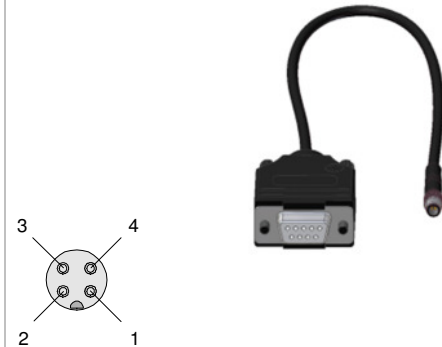
SLAVE /
SPI

PC /
RS-232

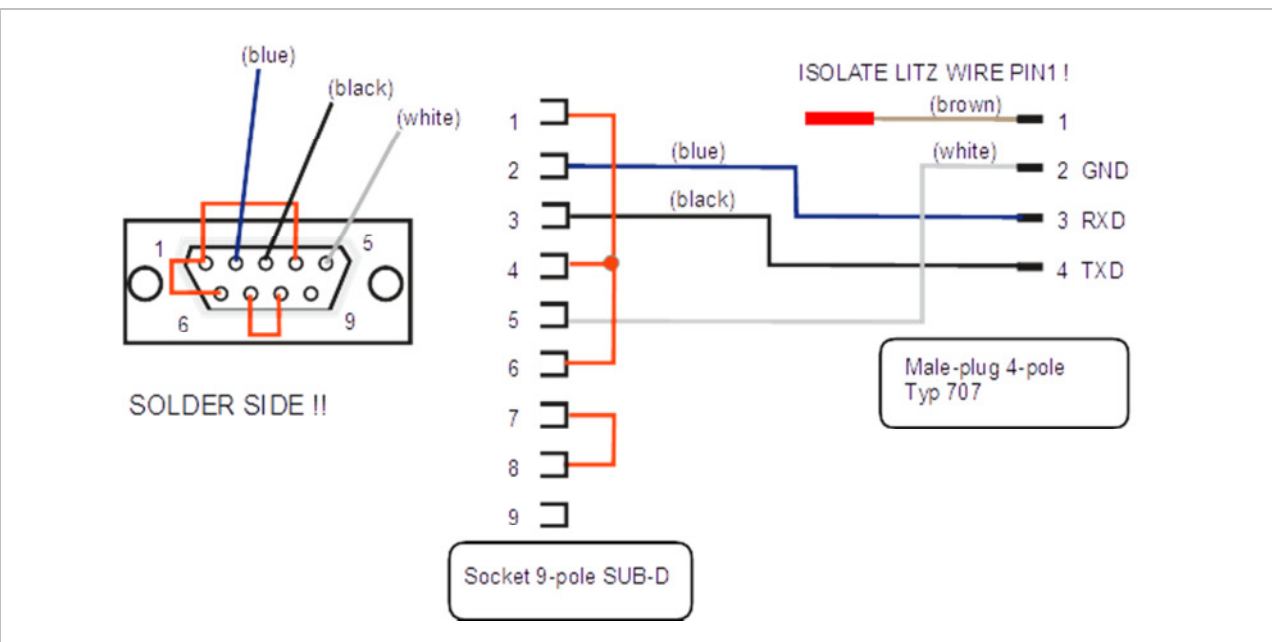
PC-RS-232:

1 (brown)	+UB
2 (white)	GND
3 (blue)	RXD
4 (black)	TXD

4-pole M5 female connector type Binder 707
 connecting cable: cab-las4/PC



INTERNAL SUB-D9 JUMPER:



COMMUNICATION METHOD:

METHOD:

The sensor control unit always behaves passively. Data exchange therefore is initiated by the PC (or PLC). The PC sends a data package ("frame") either with or without appended data, to which the sensor control unit responds with a frame that matches the request. The data package comprises a **HEADER** and the optional **DATA**.

HEADER

- 1. Byte** : Synchronisation byte <SYNC> (85_{dec} = 0x55_{hex})
- 2. Byte** : Order byte <ORDER>
3. Byte : Argument <ARG LO>
4. Byte : Argument <ARG HI>
5. Byte : Data length <LEN LO>
6. Byte : Data length <LEN HI>
7. Byte : Checksum header <CRC8 HEAD>
8. Byte : Checksum data <CRC8 DATA>

The first byte is a synchronisation byte and always is 85_{dec} (55_{hex}). The second byte is the so-called order byte <ORDER>, it determines the action that should be performed (send data, save data, etc.).

A 16-bit value <ARG> follows as the third and fourth byte. Depending on the order the argument is assigned a corresponding value. The fifth and sixth byte represent a 16-bit value <LEN>. This value states the number of appended data bytes. Without appended data <LEN=0>, the maximum data length is 512 bytes <LEN=512>. The seventh byte is formed with the CRC8 checksum over all data bytes.

The eight byte is the CRC8 checksum for the header and is formed from bytes 0 up to and incl. 7.

The header always has a total length of 8 bytes. The complete frame may contain between 8 and 520 bytes.

RS-232-FRAME:

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header	Byte9 Data	Byte10 Data	...	Byte n+7 Data	Byte n+8 Data
0x55	<ORDER>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	Data1 (lo byte)	Data1 (hi byte)	...	Data n/2 (lo byte)	Data n/2 (hi byte)

ORDER-TABLE:

<ORDER>	Meaning of the 2.nd byte <order>:	ORDER-TABLE
0	NOP	no operation
1	Send parameter from PC to L-LAS-RAM	PC ⇒ L-LAS-RAM
2	Get parameter from L-LAS-RAM	L-LAS-RAM ⇒ PC
3	Send parameter from PC to EEPROM	PC ⇒ L-LAS-EEPROM
4	Get parameter from EEPROM of L-LAS	L-LAS-EEPROM ⇒ PC
5	Echo check: Get echo of L-LAS	first word=0x00AA=170dec
6	Activate teach at L-LAS, store in RAM	PC ⇒ L-LAS-RAM
7	Get software version info of L-LAS	L-LAS ⇒ PC
8	Get measured values from L-LAS-RAM	L-LAS-RAM ⇒ PC
9	Get video-buffer info from L-LAS	L-LAS-RAM ⇒ PC
11	Reset maximum/minimum values at analog-output	PC ⇒ L-LAS-RAM
190	Change RS232-baud-rate (L-LAS-RAM)	PC ⇒ L-LAS-RAM

CRC8 checksum

The so-called "Cyclic Redundancy Check" or CRC is used to verify data integrity. This algorithm makes it possible to detect individual bit errors, missing bytes, and faulty frames. For this purpose a value - the so-called checksum - is calculated over the data (bytes) to be checked and is transmitted together with the data package. Calculation is performed according to an exactly specified method based on a generator polynomial. The length of the checksum is 8 bit (= 1 byte). The generator polynomial is:

$$X^8 + X^5 + X^4 + X^1$$

To verify the data after they have been received, CRC calculation is performed once again. If the sent and the newly calculated CRC values are identical, the data are without error.

The following pseudo code can be used for checksum calculation:

calcCRC8 (data[], table[])

Input: data[], n data of unsigned 8bit

table[], 256 table entries of unsigned 8bit

Output: crc8, unsigned 8bit

```

crc8 := AAhex
for I := 1 to n do
    idx := crc8 EXOR data[ i ]
    crc8 := table[ idx ]
endfor
return crc8

```

table[]

0	94	188	226	97	63	221	131	194	156	126	32	163	253	31	65
157	195	33	127	252	162	64	30	95	1	227	189	62	96	130	220
35	125	159	193	66	28	254	160	225	191	93	3	128	222	60	98
190	224	2	92	223	129	99	61	124	34	192	158	29	67	161	255
70	24	250	164	39	121	155	197	132	218	56	102	229	187	89	7
219	133	103	57	186	228	6	88	25	71	165	251	120	38	196	154
101	59	217	135	4	90	184	230	167	249	27	69	198	152	122	36
248	166	68	26	153	199	37	123	58	100	134	216	91	5	231	185
140	210	48	110	237	179	81	15	78	16	242	172	47	113	147	205
17	79	173	243	112	46	204	146	211	141	111	49	178	236	14	80
175	241	19	77	206	144	114	44	109	51	209	143	12	82	176	238
50	108	142	208	83	13	239	177	240	174	76	18	145	207	45	115
202	148	118	40	171	245	23	73	8	86	180	234	105	55	213	139
87	9	235	181	54	104	138	212	149	203	41	119	244	170	72	22
233	183	85	11	136	214	52	106	43	117	151	201	74	20	246	168
116	42	200	150	21	75	169	247	182	232	10	84	215	137	107	53

APPENDED-DATA: PARAMETER-SET

The sensors of the *L-LAS-TB-MS series* operate with the following parameters that are sent to the sensor or read from the sensor as appended data in the stated sequence:

DATA-FRAME: <parameter-set>		
Para	Meaning	Comment
1	EVAL-MODE	Evaluation-mode (0=MASTER,1=SLAVE,2=WIDTH,3=CENTER,4=DIFF)
2	TRIG-MODE	External-trigger-mode:(0=CONTINUOUS, 1=IN0 L/H, 2=IN0 HI, 3=SAVE-VTHD, 4=OUTPUT-WIDTH, 5=LIGHT-ON)
3	ANA-MODE	Analog-mode (0=DIRECT,1=MAXIMA,2=MINIMA,3=MAX_MIN)
4	HW-MODE	Enable/disable TOL-potentiometer and button at housing (DISABLE-ALL=0, ENABLE-ALL=1,ENABLE-BTN=2, ENABLE POTI=3)
5	MEAS-MODE	Measurement-mode (0=RELATIVE, 1=ABSOLUTE)
6	AVERAGE	Average-setting (1,2,4,5,16,32,64,128 or 256)
7	POLARITY	Polarity for OUT0, OUT1 und OUT2 (0=DIRECT, 1=INVERT)
8	ANA-ZOOM	Analog-output-zoom-mode: output (0=DIRECT, 1=ZOOMx1, 2=ZOOMx2, 3=ZOOMx4, 4=ZOOMx8, 5=ZOOMx16, 6=WIN_5V, 7=WIN_10V)
9	RS-232-MODE	RS232 mode: (0=STAT,1=IN0-L/H,2=IN0-HI[3-byte],3=CONT[3-byte])
10	RS-232-BAUD	Baudrate: (0=9600,1=19200,2=38400,3=57600,4=115200) baud
11	UM-TOL-UP (low-word)	Upper-tolerance-value [µm] low-byte
12	UM-TOL-UP (hi-word)	Upper-tolerance-value [µm] high-byte
13	UM-TOL-LO (low-word)	Lower-tolerance-value [µm] low-byte
14	UM-TOL-LO (hi word)	Lower-tolerance-value [µm] high-byte
15	UM-SETPOINT (low-word)	Setpoint-value [µm] low-byte
16	UM-SETPOINT (hi-word)	Setpoint-value [µm] high-byte
17	POWER MASTER	Laser intensity at master (0 ... 1000)
18	VIDEO-THD-FIX-MA	Video-threshold-master-fix (0 ... 100)
19	VIDEO-THD-AUTO-MA	Video-threshold-master-auto (0 ... 100)
20	TEACH-VAL-MA	Teach-value-master [pixel]
21	EVAL-BEG-MA	Evaluation start-pixel-master (1 ... E_END - 1)
22	EVAL-END-MA	Evaluation end-pixel-master (E_BEG+1 ... SUBPIXEL)
23	OP-MODE-MA	Receiver-operation-mode-master (0=LOW-GAIN, 1=HIGH-GAIN)
24	VTHD-MODE-MA	Video-threshold-mode master (0=FIX, 1=AUTO)
25	VIDEO-SMOOTH-MA	Smooth of video-signal-master (0, 2, 4, 8, 16, 32 or 64)
26	PARA9-MA	Free parameter-master
27	POWER SLAVE	Laser intensity at slaver (0 ... 1000)
28	UM-TEACH-SL (low-word)	Teach-value-slave [µm] low-byte
29	UM-TEACH-SL (hi-word)	Teach-value-slave [µm] high-byte
30	VIDEO-THD-FIX-SL	Laser intensity at slave (0 ... 1000)
31	VIDEO-THD-AUTO-SL	Video-threshold-slave-fix (0 ... 100)
32	TEACH-VAL-SL	Teach-value-slave [pixel]
33	EVAL-BEG-SL	Evaluation start-pixel-slave (1 ... E_END - 1)
34	EVAL-END-SL	Evaluation end-pixel-slave (E_BEG+1 ... SUBPIXEL)
35	OP-MODE-SL	Receiver-operation-mode-slave (0=LOW-GAIN, 1=HIGH-GAIN)
36	VTHD-MODE-SL	Video-threshold-mode-slave (0=FIX, 1=AUTO)
37	VIDEO-SMOOTH-SL	Smooth of video-signal-slave (0, 2, 4, 8, 16, 32 or 64)
38	PARA9-SL	Free parameter-slave
39	UM-TEACH-SL (low-word)	Teach-value-slave [µm] low-byte
40	UM-TEACH-SL (hi-word)	Teach-value-slave [µm] high-byte

RS232 data-transfer-examples:

< ORDER = 5 > : READ CONNECTION OK from sensor.

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85	5	0	0	0	0	170	60
ARG=0				LEN=0			

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85	5	170	0	0	0	170	178
ARG=170				LEN=SERIAL-NO			

< ORDER = 7 > : Read FIRMWARE-STRING from sensor.

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	7	0	0	0	0	170	82
ARG=0				LEN=0			

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header	Byte9 Data	Byte10 Data	Byte11 Data	Byte12 Data
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	ASCII	ASCII	ASCII	ASCII
85 (dec)	7	0	0	72	0	xxx	82	L	-	L	A
ARG=0				LEN=72							

Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data	Byte21 Data	Byte22 Data	Byte23 Data	Byte24 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
S	-	T	B	-	8	-	C	O	N	1	-

Byte25 Data	Byte26 Data	Byte27 Data	Byte28 Data	Byte29 Data	Byte30 Data	Byte31 Data	Byte32 Data	Byte33 Data	Byte34 Data	Byte35 Data	Byte36 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
2	X		V	4	.	0	.	0		0	2

Byte37 Data	Byte38 Data	Byte39 Data	Byte40 Data	Byte41 Data	Byte42 Data	Byte43 Data	Byte44 Data	Byte45 Data	Byte46 Data	Byte47 Data	Byte48 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
/	S	e	p	/	1	4					

Byte49 Data	Byte50 Data	Byte51 Data	Byte52 Data	Byte53 Data	Byte54 Data	Byte55 Data	Byte56 Data	Byte57 Data	Byte58 Data	Byte59 Data	Byte60 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII

Byte61 Data	Byte62 Data	Byte63 Data	Byte64 Data	Byte65 Data	Byte66 Data	Byte67 Data	Byte68 Data	Byte69 Data	Byte70 Data	Byte71 Data	Byte72 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII

Byte73 Data	Byte74 Data	Byte75 Data	Byte76 Data	Byte77 Data	Byte78 Data	Byte79 Data	Byte80 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII

< ORDER = 1 > : SEND PARAMETER-SET TO RAM of the sensor

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header	Byte9 Data	Byte10 Data	Byte11 Data	Byte12 Data
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	Para1 (lo byte)	Para1 (hi byte)	Para2 (lo byte)	Para2 (hi byte)
85 (dec)	1	0	0	46	0	xxx	81	2	0	0	0
ARG=0				LEN=46				EMODE=2		TRGMODE=0	

Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data	Byte21 Data	Byte22 Data	Byte23 Data	Byte24 Data
Para3	Para3	Para4	Para4	Para5	Para5	Para6	Para6	Para7	Para7	Para8	Para8
0	0	0	0	1	0	2	0	1	0	0	0
ANAMODE=0		HWMODE=0		MEASMODE=1		AVERAGE=2		POLARITY=1		ANAZOOM=0	

Byte25 Data	Byte26 Data	Byte27 Data	Byte28 Data	Byte29 Data	Byte30 Data	Byte31 Data	Byte32 Data	Byte33 Data	Byte34 Data	Byte35 Data	Byte36 Data
Para9	Para9	Para10	Para10	Para11	Para11	Para12	Para12	Para13	Para13	Para14	Para14
0	0	4	0	232	3	0	0	3	0	0	0
RS232MODE=0		RS232BAUD=4		UM_TOL_HI=1000				UM_TOL_LOW=1000			

Byte37 Data	Byte38 Data	Byte39 Data	Byte40 Data	Byte41 Data	Byte42 Data	Byte43 Data	Byte44 Data	Byte45 Data	Byte46 Data	Byte47 Data	Byte48 Data
Para15	Para15	Para16	Para16	Para17	Para17	Para18	Para18	Para19	Para19	Para20	Para20
16	39	0	0	144	1	250	0	70	0	8	0
UM_SETPOINT=10000				POWER_MA=400		VTHD_FIX_MA=25		VTHD_AUT_MA=70		TVAL_PIX_MA=2048	

Byte49 Data	Byte50 Data	Byte51 Data	Byte52 Data	Byte53 Data	Byte54 Data	Byte55 Data	Byte56 Data	Byte57 Data	Byte58 Data	Byte59 Data	Byte60 Data
Para21	Para21	Para22	Para22	Para23	Para23	Para24	Para24	Para25	Para25	Para26	Para26
1	0	16	0	0	0	0	0	2	0	0	0
E_BEG_MA=1		E_END_4096		OP_MODE_MA=0		VTHDMODE_MA=0		SMOOTH_MA=2		PARA9_MA=0	

Byte49 Data	Byte50 Data	Byte51 Data	Byte52 Data	Byte53 Data	Byte54 Data	Byte55 Data	Byte56 Data	Byte57 Data	Byte58 Data	Byte59 Data	Byte60 Data
Para27	Para27	Para28	Para28	Para29	Para29	Para30	Para30	Para31	Para31	Para32	Para32
136	19	0	0	144	1	25	0	70	19	08	0
UM_TEACH_MA=5000				POWER_SL=400		VTHD_FIX_SL=25		VTHD_AUT_SL=70		TVAL_PIX_MA=2048	

Byte61 Data	Byte62 Data	Byte63 Data	Byte64 Data	Byte65 Data	Byte66 Data	Byte67 Data	Byte68 Data	Byte69 Data	Byte70 Data	Byte71 Data	Byte72 Data
Para33	Para33	Para34	Para34	Para35	Para35	Para36	Para36	Para37	Para37	Para38	Para38
1	0	16	0	0	0	0	0	2	0	0	0
E_BEG_SL=1		E_END_SL=4096		OP_MODE_SL=0		VTHDMODE_SL=0		SMOOTH_SL=2		PARA9_SL=0	

Byte73 Data	Byte74 Data	Byte75 Data	Byte76 Data
Para39	Para39	Para40	Para40
136	19	0	0
UM_TEACH_SL=5000			

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	1	0	0	0	0	170	81
ARG=0				LEN=0			

< ORDER = 2 > : READ PARAMETER-SET FROM RAM of the sensor

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	2	0	0	0	0	170	185
ARG=0				LEN=0			

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header	Byte9 Data	Byte10 Data	Byte11 Data	Byte12 Data
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	Para1 (lo byte)	Para1 (hi byte)	Para2 (lo byte)	Para2 (hi byte)
85 (dec)	2	0	0	46	0	xxx	185	144	1	0	0
ARG=0				LEN=46				POWER=400		PMODE=0	

Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data	Byte21 Data	Byte22 Data	Byte23 Data	Byte24 Data
Para3	Para3	Para4	Para4	Para5	Para5	Para6	Para6	Para7	Para7	Para8	Para8
0	0	0	0	1	0	2	0	1	0	0	0
ANAMODE=0		HWMODE=0		MEASMODE=1		AVERAGE=2		POLARITY=1		ANAZOOM=0	

Byte25 Data	Byte26 Data	Byte27 Data	Byte28 Data	Byte29 Data	Byte30 Data	Byte31 Data	Byte32 Data	Byte33 Data	Byte34 Data	Byte35 Data	Byte36 Data
Para9	Para9	Para10	Para10	Para11	Para11	Para12	Para12	Para13	Para13	Para14	Para14
0	0	4	0	232	3	0	0	3	0	0	0
RS232MODE=0		RS232BAUD=4		UM_TOL_HI=1000				UM_TOL_LOW=1000			

Byte37 Data	Byte38 Data	Byte39 Data	Byte40 Data	Byte41 Data	Byte42 Data	Byte43 Data	Byte44 Data	Byte45 Data	Byte46 Data	Byte47 Data	Byte48 Data
Para15	Para15	Para16	Para16	Para17	Para17	Para18	Para18	Para19	Para19	Para20	Para20
16	39	0	0	144	1	250	0	70	0	8	0
UM_SETPOINT=10000				POWER_MA=400		VTHD_FIX_MA=25		VTHD_AUT_MA=70		TVAL_PIX_MA=2048	

Byte49 Data	Byte50 Data	Byte51 Data	Byte52 Data	Byte53 Data	Byte54 Data	Byte55 Data	Byte56 Data	Byte57 Data	Byte58 Data	Byte59 Data	Byte60 Data
Para21	Para21	Para22	Para22	Para23	Para23	Para24	Para24	Para25	Para25	Para26	Para26
1	0	16	0	0	0	0	0	2	0	0	0
E_BEG_MA=1		E_END_MA=4096		OP_MODE_MA=0		VTHDMODE_MA=0		SMOOTH_MA=2		PARA9_MA=0	

Byte49 Data	Byte50 Data	Byte51 Data	Byte52 Data	Byte53 Data	Byte54 Data	Byte55 Data	Byte56 Data	Byte57 Data	Byte58 Data	Byte59 Data	Byte60 Data
Para27	Para27	Para28	Para28	Para29	Para29	Para30	Para30	Para31	Para31	Para32	Para32
136	19	0	0	144	1	25	0	70	19	08	0
UM_TEACH_MA=5000				POWER_SL=400		VTHD_FIX_SL=25		VTHD_AUT_SL=70		TVAL_PIX_MA=2048	

Byte61 Data	Byte62 Data	Byte63 Data	Byte64 Data	Byte65 Data	Byte66 Data	Byte67 Data	Byte68 Data	Byte69 Data	Byte70 Data	Byte71 Data	Byte72 Data
Para33	Para33	Para34	Para34	Para35	Para35	Para36	Para36	Para37	Para37	Para38	Para38
1	0	16	0	0	0	0	0	2	0	0	0
E_BEG_SL=1		E_END_SL=4096		OP_MODE_SL=0		VTHDMODE_SL=0		SMOOTH_SL=2		PARA9_SL=0	

Byte73 Data	Byte74 Data	Byte75 Data	Byte76 Data
Para39	Para39	Para40	Para40
136	19	0	0
UM_TEACH_SL=5000			

< ORDER = 8 > : READ MEASUREMENT DATA from sensor

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	8	0	0	0	0	170	118
ARG=0				LEN=0			

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header	Byte9 Data	Byte10 Data	Byte11 Data	Byte12 Data
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	Raw1 (lo byte)	Raw1 (hi byte)	Raw2 (lo byte)	Raw2 (hi byte)
85 (dec)	8	0	0	40	0	xxxx	118	155	8	64	9
ARG=0				LEN=34				MA_PIXEL = 2203		SL_PIXEL = 2368	

Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data	Byte21 Data	Byte22 Data	Byte23 Data	Byte24 Data
Raw3	Raw3	Raw4	Raw4	Raw5	Raw5	Raw6	Raw6	Raw7	Raw7	Raw8	Raw8
219	17	142	178	0	0	14	86	0	0	128	92
M_VAL_PIX = 4571		UM_VALUE = 45710				UM_VALUE_MA = 22030				UM_VALUE_SL	

Byte25 Data	Byte26 Data	Byte27 Data	Byte28 Data	Byte29 Data	Byte30 Data	Byte31 Data	Byte32 Data	Byte33 Data	Byte34 Data	Byte35 Data	Byte36 Data
Raw9	Raw9	Raw10	Raw10	Raw11	Raw11	Raw12	Raw12	Raw13	Raw13	Raw14	Raw14
0	0	1	0	1	0	0	0	249	3	195	80
= 23680		EDGE MA = 1		EDGE SL = 1		IN_STATE = 0		POTI= 1017		UM_MAX_VAL	

Byte37 Data	Byte38 Data	Byte39 Data	Byte40 Data	Byte41 Data	Byte42 Data
Raw15	Raw15	Raw16	Raw16	Raw17	Raw17
0	0	0	0	0	0
= 50000		UM_MIN_VAL = 0			

< ORDER = 11 > : RESET MAX/MIN VALUE OF ANALOG-OUTPUT at sensor

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	11	0	0	0	0	170	118
ARG=0				LEN=0			

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	190	0	0	0	0	170	195
ARG=0				LEN=0			

< ORDER = 6 > : INITIATE TEACH-PROCEDURE at sensor (RAM)

The actual measurement value is set as new TEACH-IN value

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85	6	0	0	0	0	170	101
ARG=0				LEN=0			

DATA FRAME Sensor → PC
New TEACH-value is sent back

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header	Byte9 Data	Byte10 Data
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	Para1 (lo byte)	Para1 (hi byte)
85 (dec)	7	0	0	1	0	xxx	195	232	3
ARG=0				LEN=1				TVAL_MA=1000	

Byte11 Data	Byte12 Data	Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data
Raw9	Raw9	Raw10	Raw10	Raw11	Raw11	Raw12	Raw12	Raw13	Raw13
232	3	136	19	0	0	136	19	0	0
TVAL_SL= 1000		UM_TVAL_MA = 5000				UM_TVAL_SL = 5000			

< ORDER = 190 > : CHANGE BAUDRATE at sensor (RAM)

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	190	1	0	0	0	170	14
ARG=1				LEN=0			

New baud rate is set by <ARG> value:

ARG=0: baud rate = 9600

ARG=1: baud rate = 19200

ARG=2: baud rate = 38400

ARG=3: baud rate = 57600

ARG=4: baud rate = 115200

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	190	0	0	0	0	170	195
ARG=0				LEN=0			

< ORDER = 9 > : GET VIDEO-DATA INFORMATION of sensor

ATTENTION: Only 256 pixel of the CMOS line-sensor are transferred!

The <ARG> value determines the source of the VIDEO-DATA-INFORMATION

ARG = 0 : CMOS-VIDEO-RAM-DATA (first 128-pixel = MASTER, second 128-pixel = SLAVE)

ARG = 1 : EE-VIDEO-THRESHOLD-DATA (first 128-pixel = MASTER, second 128-pixel = SLAVE)

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	9	0	0	0	0	170	185
ARG=0				LEN=0			

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header	Byte9 Data	Byte10 Data	Byte11 Data	Byte12 Data
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	Para1 (lo byte)	Para1 (hi byte)	Para2 (lo byte)	Para2 (hi byte)
85 (dec)	9	0	0	0	1	xxx	185	200	0	220	0
ARG=0				LEN=256				PIX1=200		PIX2=220	

Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data	Byte21 Data	Byte22 Data	Byte23 Data	Byte24 Data
Para3	Para3	Para4	Para4	Para5	Para5	Para6	Para6	Para7	Para7	Para8	Para8
240	0	0	1	44	1	124	1	0	2	88	2
PIX3=240		PIX4=256		PIX5=300		PIX6=380		PIX7=512		PIX8=600	

Byte25 Data	Byte26 Data	Byte27 Data	Byte28 Data	Byte29 Data	Byte30 Data	Byte31 Data	Byte32 Data	Byte33 Data	Byte34 Data	Byte35 Data	Byte36 Data
Para9	Para9	Para10	Para10	Para11	Para11	Para12	Para12	Para13	Para13	Para14	Para14
168	2	170	2	188	2	188	2	198	2	208	2
PIX9=680		PIX10=682		PIX11=700		PIX12=700		PIX13=710		PIX14=720	

Byte37 Data	Byte38 Data	Byte39 Data	Byte40 Data	Byte41 Data	Byte42 Data	Byte43 Data	Byte44 Data	Byte45 Data	Byte46 Data	Byte47 Data	Byte48 Data
Para15	Para15	Para16	Para16	Para17	Para17	Para18	Para18	Para19	Para19	Para20	Para20
34	3	32	3	32	3	22	3	19	3	20	3
PIX15=802		PIX16=800		PIX17=800		PIX18=790		PIX19=787		PIX20=788	

●
●
●

Byte49 Data	Byte50 Data	Byte51 Data	Byte52 Data	Byte53 Data	Byte54 Data	Byte55 Data	Byte56 Data	Byte57 Data	Byte58 Data	Byte59 Data	Byte60 Data
Para251	Para251	Para252	Para252	Para253	Para253	Para254	Para254	Para255	Para255	Para256	Para256
124	1	44	1	0	1	240	0	220	0	200	0
PIX251=380		PIX252=300		PIX253=256		PIX254=240		PIX255=220		PIX256=200	