General information:

Serial interfaces enable digital communication with computers or higher ranking control systems.
An RS 232 (optional USB) interface permits connection of one controller per computer interface. The RS485 interfaces enable the connection of max. 32 participants in one data bus. Here, the controllers must be set to different addresses for differentiation.(Controller configuration level). Several Interfaces upon request.

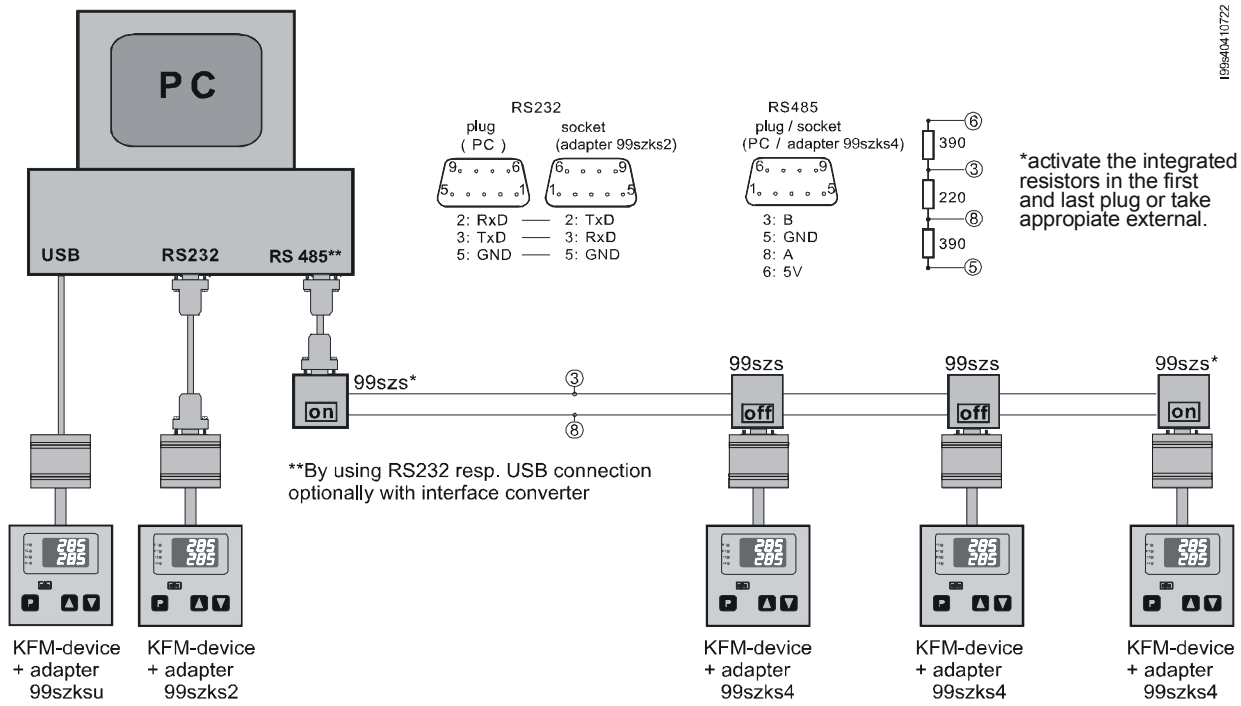
Technical data:

Interface:	direct connection - RS232, USB	bus connection - RS485
Connection:	serial, asynchronous 2 wire (+GND)	serial, asynchronous 2 wire (+GND)
Transfer medium:	twisted and screened cable	twisted and screened cable
Bus line length:	-	1000m
Branch connection length:	15m	2m
Max. number of controllers:	1	31
Transmission direction:	-	data flow control
Transfer rate:	9600, 19200, 38400 Bit / s	9600, 19200, 38400 Bit / s

Connection lines:

Cable junction for KFM devices is done by adapters which are linked to the service interface.
By direct connection: RS232(99szks2), alternatively USB(99szksu)
By bus connection: RS485(99szks4)
To establish the bus connection, providing that there is no interface RS485 on PC resp. at the PLC existing, an interface converter RS232 resp. USB to RS485 is necessary.
Use screened lines to connect the interfaces (e.g. KFM 99szl.).
Place the screening on the controller earthing terminal.
Connect the RS485 line at the beginning (PC or interface converter) and the end (last controller) with d- sub - plugs with integrated resistors (f.e. type 99szs) or appropriate external resistors.

Wiring example:



Transmission protocol:

Data is transmitted according to the KFM protocol 2.0 which is laid out in line with the ISO 1745.

Data format:

Each field of the data frame comprises a start bit, the 7 bit ASCII value,
a parity bit for even parity and a stop bit.
Transmission rate: 9600, 19200 or 38400 Bit/s (depending on type resp. setting).

Data frame (telegram) :

The data frame commences and ends with a control character (see table) and comprises of 2 bytes for the controller address (ADR) and 4 bytes for the parameter code (code) as well as up to 6 bytes for the number value, i.e.: number value on the left from the decimal point: up to 4 bytes; 1 byte for the decimal point, number value on the right of the decimal point: 1 byte.
1 byte = 1 digit or letter or control character = 1 ASCII value

Data integrity:

The data frames for parameter transmission are safeguarded with a "BCC" sign, i.e.the transmitted data is supplemented by a check sum consisting of the logic linking (EXOR) of all transmitted characters between STX (excl.) and ETX (incl.).
The controller acknowledges a successful transmission with "ACK". A faulty transmission is confirmed by "NAK".

Examples: The computer requests data

EOT	ADR	ADR	Code	Code	Code	Code	ENQ

the controller response:

STX	Code	Code	Code	Code	=	val.	val.	val.	val.	val.	ETX	BCC

The computer sends data. To this effect, the controller must be in operational state, because the simultaneous entry of data via interface and controller keyboard is inadmissible.

EOT	ADR	ADR	STX	Code	Code	Code	Code	=	val.	val.	val.	val.	val.	ETX	BCC

controllers response to a successful transmission: ACK

controllers response to a faulty transmission: NAK

Control characters:

control characters	value(HEX)	meaning
STX	02	start of text
ETX	03	end of text
EOT	04	end of transmission
ENQ	05	enquire
ACK	06	acknowledge
NAK	15	not acknowledge
=	3D	value allocation

Permitted characters for code and value:

ASCII-code	value (HEX)	description	ASCII-Zeichen	value(HEX)	description
"0"	30	zero	"9"	39	nine
"1"	31	one	"A"	41	ten (code)
"2"	32	two	"B"	42	eleven (code)
"3"	33	three	"C"	43	twelve (code)
"4"	34	four	"D"	44	thirteen (code)
"5"	35	five	"E"	45	fourteen (code)
"6"	36	six	"F"	46	fifteen (code)
"7"	37	seven	"."	2E	dec.point (value)
"8"	38	eight	"-"	2D	minus sign(val.)

general information: *Only for older devices without online- functionality:*
Transmission parameters are distinguished between online parameters and offline parameters, depending on their processing in the KFM – devices.
Online- parameters allow data transmission while the controller is in normal operating mode. In contrast, **offline**- parameters cannot be transmitted while the controller continues normal operation. So the controller has to be stopped by sending "10FE = 7708" (the display shows "ConF"). After the transmission the controller has to be switched-over to the normal operating mode by sending "10FF = 7708".

(The parameter code is exemplary for channel 1. For example: 113A = travel time of actuator channel 1. For the other channels change the number at the second place to the true channel number, for example 123A = travel time of actuator channel 2).
Parameters are available depending on type and design.

display	purpose	value range	CODE (HEX)	type
Operating indication:				
-	status- / control words 1..5 (word structure see page 6)		1001..05	online
IST1	actual value 1..6		1010..15	online*
ISTM1.1..10.2	actual value external modules 1..20		6250..64	online*
Y(1)..5	controller output channel 1..5	-100...100	1020..24	online*
Y	active controller output (e.g. 99g8.)	-100...100	102A	online*
D.W.	difference actual value 1 – actual value 2		1052	online*
M.W.	average actual value 1 / 2		1051	online*
Setpoint level:				
(1)SP	(internal) set point value channel 1	Lo...Hi *(see level 2)*	1(1)00	online
(1)SP2	second set point value channel 1	Lo...Hi *(see level 2)*	1(1)01	online
(1..5SP) **(rSP)**	active set point value channel 1..5, also active ramp- / program set point value		1030..34	online*
SP..	actual program step set point		3002	online*
SPB	bus setpoint	Lo...Hi *(see level 2)*	1060..64	online
SP-F	switch over SP/SPE	0 = SP, 1 = SPE	111C	online
2SP	current sequential controller set point value		103F	online*
P-CY	number of program cycles**	0...20	0148	online
“	number of program cycles module softw.**	0...1000	2650	online
A-CY	current program cycle module software**	0...1000	3003	online
Pro	actual program status	0=off,1=on,2=stop	3001	online
d15	density	500...1500	0152	offline
Parameter level 1:				
FUE	guide controller on/off	0=off,1=on	014D	online
(1)P(1)..4	proportional band XP1..4 channel 1	0.0...999.9	1(1)03..06	online
(1)I(1)..4	integral action time Tn1..4 channel 1	0.0...999.9	1(1)07..0A	online
(1)d(1)..4	derivative time Tv1..4 channel 1	0.0..99.9/0.00..99.99	1(1)0B..0E	online
(1)Sh	neutral zone Xsh channel 1	0.05...1,0	1(1)0F	online
(1)SA1..2	switching interval 1..2 channel 1	0...range(bLo/Hi) *(see*	1(1)13..14	online
(1)Sd1..2	switching difference 1..2 channel 1	0...range(bLo/Hi) *level 2)*	1(1)15..16	online
SA1..8	switching interval addit.contact 1..8	0.0...range	2000..07	online
Sd1..8	switching diff. addit.contact 1..8	0.1...range	2008..0F	online

* (parameter codes that can only be **send** by the controller)

** (transmission only with deactivated program function)

display	purpose	value range	CODE (HEX)	type
Parameter level 2:				
Unit	display unit °C / °F	0=°C, 1=°F	013F	offline
0bLo	min. val. range of actual val. 0 (diff/ aver.)	-999...bHi	1129	offline
UNIT	viscosity	0=cst, 1=cP	0151	offline
0bHi	max. val. range of actual val. 0 (diff/ aver.)	blo...4000	112A	offline
1..6bLo	min. value range input 1..6	-999...bHi	010C..11	offline
M1.1..10.2bLo	min. value range ext. modules input 1..20	-999...bHi	E4D0..E4	offline
1..6bHi	max. value range input 1..6	blo...4000	0112..17	offline
M1.1..10.2bHi	max. value range ext. modules input 1..20	blo...4000	E4E0..E4	offline
(1..3)SLo	min. value range signal output	-999...Shi	012A..2C	offline
(1..3)SHi	max. value range signal output	SLo...4000	0130..32	offline
0nst	decimal point actual value 0 (diff/ aver.)	0...2 (dep. on the range)	1128	offline
FLo	low limit set point value guide controller	0... Fhi	1130	online
FHi	high limit set point value guide controller	Flo...400	1131	online
1..6nst	decimal point input 1..6	0...2 (dep. on the range)	0118..1d	offline
M1.1..10.2nst	decimal point ext. modules input 1..20	0...2 (dep. on the range)	E550..64	offline
1 Lo	low limit set point value	-999...bHi	112E	offline
1 Hi	high limit set point value	blo...4000	112F	offline
DT	allowed deviation actual val.(dt control)	0...400	1146	offline
dSPL	lower display indication	0=OFF,1=SP, 2=rSP,3=Y,4=°C,5=°F, 6=bar,7=%,8=Ist1,9=Ist2...	0140	offline
DSP1..4	indication display line 1..4	0=OFF, 1=1 SP, 10=IST1, 11=IST2	0164..67	offline
EIN1..4	unit of measurement display line 1..4	3=m3_h, 4=C, 5=F, 6=%, 7=bar, 8=mbar, 9=mPas, 10=cSt,1=KGm3,12=mm	0168..6B	offline
Pr-S	number of program steps **	0...20	0149	offline
SP.1 .. 20	1...20. program setpoint val., program 1**	Lo...Hi	4101..14	online
H' 1.. 20	1...20. holding time, program 1**	0...6000	3101..14	online
Configuration level:				
ConF	type of controller		013C	offline
Cod1	code number	0...9999	0142	offline
Cod2	code 2..4	0...9999	0161..63	offline
LNG	language selection	0=DEUTSCH, 1=ENGLISH, 2=USER DEF, 3=OFF	8800	offline
Ist1..6	correction actual value 1..6	blo...bHi (+/-)	0124..29	offline
IstM1.1..10.2	correction actual value ext. modules 1..20	M.blo...M.bHi (+/-)	E4C0..D4	offline
Ain1..6	type of measuring input 1..6	0=4-20, 1=2-10, 2=0-20, 3=0-10, 5=rtd, 20=n100	011E..23	offline
AinM1.1..10.2	type of measuring input ext. modules 1..20	0=4-20, 1=2-10, 2=0-20, 3=0-10, 5=rtd, 20=n100	E4B0..C4	offline
SP-F	switch-over of the ext. setpoint via menu / bin. input (SP/SPE)	-2=AUS, -1=SPEB(bin.), 0=SPEM(menu), 1=SP2	014F	offline
YE	switch-over SPE / YE	0=SPE,1=YE	114E	offline
SPE	function of the ext. setpoint	2=AbS,3=Add,4=Sub	112D	offline

* (parameter codes that can only be **send** by the controller)

** (transmission only with deactivated program function)

display	purpose	value range	CODE (HEX)	type
configuration level (continued):				
REL_	switch.behaviour of the first step,step contr.	0=stat, 1=rel	1144	offline
(1) Y"	travel time of actuator channel 1	6...600	1(1)3A	offline
(1) TE	switch-on delay per step, step controller	0...600	1(1)43	offline
Cy"	cycle time	2...120	013D	offline
(1) TP	pause time step controller	0...60	1(1)45	offline
(1)out	type of output signal 0 / 4...20mA	0=0-20,1=4-20	1(1)3B	offline
(1)out	output direction di / in	0=in(in),1=(in)di,2=diin,3=didi	1(1)3C	Offline
dSLo	valve drop. minimum	0...50	1122	offline
out	limitation selection (min / max)	0=Lo, 1=Hi	1127	offline
dSHi	valve drop. maximum	50...100	1123	offline
(1) ib	integration range limit channel 1	0...100	1(1)40	offline
(1)YLo	low limit control output	0...Yhi	1(1)38	offline
(1)YHl	limitation control output	-100...100	1(1)41	offline
(1)YHi	high limit control output	YLo...100	1(1)39	offline
(1) TY	control output slope	0...100	1(1)42	offline
(1) DB	damping range	0...100	1(1)25	offline
(1) D"	damping value	0...100	1(1)26	offline
Gr1..2	gradient 1..2	0...100	1132..33	offline
rF1..2	waiting window value 1..2	0.1...999.9	1134..35	offline
td	dead range	0.0...10.0	113D	offline
Sout(1..3)	signal output 0/4...20mA	0=0-20,1=4-20	0136..38	offline
Sou1..5	assignment signal output	11=Ist1, 12=Ist2, 21=SP	0155..5A	offline
(1)Y_S	output reaction at meas.fault (relais)	0=off,1=K1,2=K2	1(1)3E	offline
(1)Y_S	output reaction at meas.fault (Y)	YLo...YHi (continuous)	1(1)3F	offline
(1)YAP	operating point	YLO...Yhi	1(1)37	offline
YH	switch over control val. on / off	0=off,1=on	1148	offline
YH	external control value	0...100	1149	offline
d.SP	max. deviation actual value	0.1...200.0	0147	offline
t"	tolerance period act.value (ser. interface)	1...100	014E	offline
rEL1..8	function selection additional contact 1..8	0=LC A, 1=LC E, 2=Su A, 3=Su E, 4=So A, 5=So E, 6=St A, 7=US A, 8=US E, 11=OFF, 12=ON	2010..17	offline
rEL1..8	input selection additional contact 1..8	1...6=Ist1...6, 11=1Y...	2018..1F	offline
rEL1..8	channel / setpoint selection additional contact 1..8	1...4=1..4SP, 11=rSP...	2020..27	offline
rEL1..8	condition relay 1...8 for measuring line fault	0=SiA,1=SiE	2028..2F	offline
Adr	controller address	1...255	0141	offline
BAUD	baud rate (series 92..)	0=9600	2629	offline
	baud rate (series 902..)	0=9600, 1=19200, 2=38400	2629	offline

* (parameter codes that can only be **send** by the controller)

** (transmission only with deactivated program function)

parameter for series 9..

read: (from KFM device)

status word 1 (8 Ascii-characters, code 1001)

char. 8	char. 7	char. 6	char. 5	char. 4	char. 3	char. 2	char. 1

Ascii- *character* 1..7: status measuring input 1 .. 7
0 = error-free measurement; 1 = fault at the resp. input

status word 2 (0 - 40 Ascii-characters depending on existing binary inputs, code 1002)

char. 40	char. 39	..	..	..	..	char. 2	char. 1

Ascii-character 1..40: status binary input 1 .. 40
0 = binary input deactivated; 1 = binary input activated

status word 3 (0 - 40 Ascii-characters depending on existing additional contacts, code 1005)

char. 40	char. 39	..	..	..	..	char. 2	char. 1

Ascii-character 1..40: status additional contact 1..40
0 = contact deactivated; 1 = contact activated

write: (to KFM device)

control word 1 (2 Ascii-characters, code 1004)

character 2				character 1			
bit 4	bit 3	bit 2	bit 1	bit 8	bit 7	bit 6	bit 5

bit 1 .. 4: control bus-setpoint 1 .. 4
0 = bus-setpoint deactivated, internal sepoint (SP) active
1 = bus-setpoint active (SPB)

control word 2 (10 Ascii-characters, code 1005)

character 10				..	..	..	..	character 1			
bit 40	bit 39	bit 38	bit 37	..	..	..	..	bit 4	bit 3	bit 2	bit 1

bit 1 .. 40: control additional contact 1 .. 40, if the contact is configured to “BUS“
0 = contact deactivated
1 = contact activated

parameter for malfunction indicator type 821, 822

CODE (HEX)	parameter*	range
100F	status word binary input 1-16	0, 1

Logical organization of the hexadecimal status word with the resp. LEDs:

figure 7				figure 6				figure 5				figure 4				figure 3				figure 2				figure 1				figure 0			
4	3	2	1	8	7	6	5	12	11	10	9	16	15	14	13	4	3	2	1	8	7	6	5	12	11	10	9	16	15	14	13

figure 7..4 = status LED 1 to 16 (0 = LED off, 1 = LED active)
figure 3..0 = flashing status LED 1 to 16 (0 = LED constantly on, 1 = LED flashing)

example: LEDs 1,6,8,11,16 active (0001 1010 0100 1000),
LEDs 6,8,16 flashing (0000 1010 0000 1000)
=> status word 1 = "1A48 0A08" (as ascii-character string)**

parameter for tableau type 8219sbtm

CODE (HEX)	parameter*	range
0901	status word binary input 1-16 of I/O-unit 1	0, 1
0902	status word binary input 1-16 of I/O-unit 2	0, 1
0903	status word binary input 1-16 of I/O-unit 3	0, 1
0904	status word binary input 1-16 of I/O-unit 4	0, 1

Logical organization of the hexadecimal status word 1-4 with the resp. LEDs:

addr	addr	,	figure 7				figure 6				figure 5				figure 4				figure 3				figure 2				figure 1				figure 0			
			4	3	2	1	8	7	6	5	12	11	10	9	16	15	14	13	4	3	2	1	8	7	6	5	12	11	10	9	16	15	14	13

addr — double-digit address of the resp. I/O-unit (e.g. malfunction indicator address 04)

, — separator (comma) between address and status word

figure 7..4 — status LED 1 to 16 (0 = LED off, 1 = LED active)
figure 3..0 — flashing status LED 1 to 16 (0 = LED constantly on, 1 = LED flashing)

example: LEDs 2,5,7,10,15 active (0010 0101 0010 0100),
LEDs 5,7,10 flashing (0000 0101 0010 0000)
=> status word 1 = "04, 2524 0520" (as ascii-character string)**

note: In case of an interruption between tableau and I/O-unit, the address 00 will be send.

* (Parameters are available depending on type and design.)

** Transmission as hexadecimal number, e.g. 1 (ASCII)= 31 (hex) , see also section "data format", page 2!

programming example in “C” (extract):

```
void send_data_frame( void)
{
int i;
char antwort=' ', z_buff[80];
unsigned char bcc;

printf( "\n\ndata ----> controller");
for ( i=0; i<=strlen( Ist); i++ )                    // send data - frame
    {
    if ( i==0 )
        {
        sende_byte( 0x04);                           // send 'EOT'
        sende_byte( adresse[0]);                     // send 1. adress-byte
        sende_byte( adresse[1]);                     // send 2. adress-byte
        sende_byte( 0x02);                           // send 'STX'
        sende_byte( code[0]);                        // send 1. code-byte
        bcc=code[0];
        sende_byte( code[1]);                        // send 2. code-byte
        bcc = bcc^code[1];
        sende_byte( code[2]);                        // send 3. code-byte
        bcc = bcc^code[2];
        sende_byte( code[3]);                        // send 4. code-byte
        bcc = bcc^code[3];
        sende_byte( EQL);                            // send '='
        bcc = bcc^EQL;
        }
    sende_byte( Ist[i]);                             // send data
    bcc = bcc^Ist[i];
    }//for
sende_byte( 0x03);                                   // send 'ETX'
bcc = bcc^0x03;
sende_byte( bcc & 0x00ff);                           // send BCC-byte

for ( i=1; i<=400; i++ )
    {
    if ( (inportb ( com+LSR) & 0x01) ) antwort=inportb( com+RBR);
    if ( antwort==NAK ) { printf( "\nOut of Range !"); break;}
    if ( antwort==ACK ) { printf( "\nOK !"); break; }
    delay( 1);
    }//for
if ( i==401 ) printf( "\nNo response !");
};
```