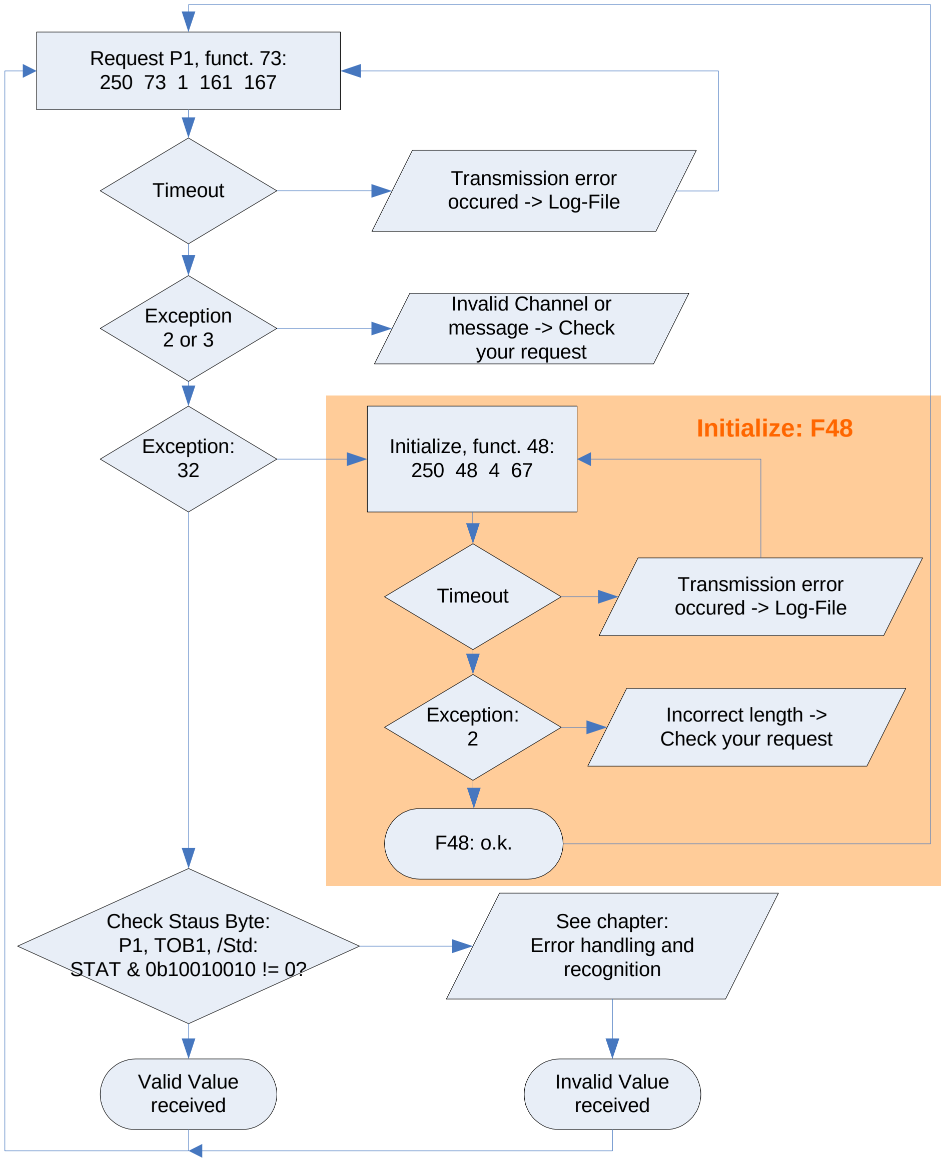
5 Description of Keller bus functions

This section describes the functions of the bus protocol for Series 30 transmitters (device *Class.Group 5.20-XX.XX and 5.21-XX.XX*) using the Keller bus functions (not MODBUS).
Note that all numbers are shown as decimal (not as hex, contradictory to what was described in the MODBUS-chapters)!

Overview:
F30: Read out calibration (scaling) and information floating-point values
F31: Write calibration floating-point values
F32: Read out configurations
F33: Write configurations
F48: Initialise devices, whereby the device ID is returned
F66: Write bus address
F69: Read out serial number
F73: Read out current pressure and temperature values in floating-point format
F74: Read out current pressure and temperature values in integer format
F95: Zeroing functions

5.1 Example: read pressure value with exception handling



A simple example for reading out a pressure value.
Because only one transmitter is connected, the „transparent“ address 250 is used.
To read out pressure functions 73 and 48 are necessary. When the slave replies with error no. 32 (device just recently started up, power on), then this has to be confirmed with function 48. This is helpful to detect current supply interruptions (on the power supply circuit).
Enhancement:
During start-up the device group can be checked via function F48 to ensure that this version is supported.
Further information is available:

- F30: pressure and temperature range
- F69 serial number

Some examples:

	address	request	response	received value
Read P1	250	250 73 1 161 167	250 73 63 109 186 172 0 26 27	0.9286296 bar
	1	1 73 1 80 214	1 73 63 109 177 83 0 231 97	0.9284870 bar
Read P2	1	1 73 2 81 150	1 73 63 109 178 242 0 119 232	0.9285117 bar
Read TOB1	250	250 73 4 162 103	250 73 65 201 184 0 0 224 204	25.21484 °C
	1	1 73 4 83 22	1 73 65 202 81 128 0 95 54	25.28979 °C
Initialize	1	1 48 52 0 1 48 52 0	1 48 5 20 1 48 5 21	FW=5.20-5.50, Buffer=10, device already initialized

5.2 Function 30: Read coefficient

Request:

DevAddr	30	Nr.	CRC16_H	CRC16_L

Response:

DevAddr	30	B3	B2	B1	B0	CRC16_H	CRC16_L

Exception errors:

2 if no. > 111
3 if message length incorrect
32 if device is not yet initialised

Note:
Every coefficient can be read in IEEE754 format (floating-point format 4-byte B0 .. B3) via this function.
→ Information on IEEE754: see appendix.

5.2.1 Calibration values

No.	Read / Write	Description of coefficient	Unit
53	R/W	Threshold value of the roof function, (must be >0 if sqrt is used) version5.20-XX.XX only	bar
64	R/W	Offset of pressure sensor P1, default 0.0	bar
65	R/W	Gain factor of pressure sensor P1, default 1.0	
66	R/W	Offset of pressure sensor P2, default 0.0	bar
67	R/W	Gain factor of pressure sensor P2, default 1.0	
68	R/W	Offset of analogue output	
69	R/W	Gain factor of analogue output	
70	R/W	Offset of CH0, default 0.0	
71	R/W	Gain factor of CH0, default 1.0	
72	R/W	Offset of temperature sensor T version 5.21-XX.XX only	°C
74	R/W	Offset of temperature sensor TOB1 version 5.21-XX.XX only	°C
76	R/W	Offset of temperature sensor TOB2 version 5.21-XX.XX only	°C
72	R/W	Upper threshold value for switching output 1** version 5.20-XX.XX only	
73	R/W	Lower threshold for switching output 1** version 5.20-XX.XX only	
78	R/W	Upper threshold value for switching output 2** version 5.20-XX.XX only	
79	R/W	Lower threshold for switching output 2** version 5.20-XX.XX only	
121	R/W	Gain Conductivity Range 1 version 5.21-XX.XX only	
122	R/W	Gain Conductivity Range 2 version 5.21-XX.XX only	
123	R/W	Gain Conductivity Range 3 version 5.21-XX.XX only	
124	R/W	Gain Conductivity Range 4 version 5.21-XX.XX only	
126	R/W	Conductivity Temperature Coefficient (default 0.022-> water) version 5.21-XX.XX only	
127	R/W	Conductivity Cell Constant (default 1.00) version 5.21-XX.XX only	
100… 111	R/W	free coefficients for customer use	

** no longer supported for devices 5.20-10.XX and newer.
The calibration values can be read and written.

5.2.2 Information values

No.	Read / Write	Description of the coefficient	Unit
80	R	Minimum pressure of sensor P1	bar
81	R	Maximum pressure of sensor P1	bar
82	R	Minimum pressure of sensor P2	bar
83	R	Maximum pressure of sensor P2	bar
84	R	Minimum temperature of temperature sensor	°C
85	R	Maximum temperature of temperature sensor	°C
86	R	Minimum temperature of sensor P1	°C
87	R	Maximum temperature of sensor P1	°C
88	R	Minimum temperature of sensor P2	°C
89	R	Maximum temperature of sensor P2	°C
90	R	Minimum value of channel CH0	
91	R	Maximum value of channel CH0	
92	R	Pressure for minimum analogue signal * version 5.20-XX.XX only	bar
93	R	Pressure for maximum analogue signal * version 5.20-XX.XX only	bar
94	R	Minimum analogue signal* version 5.20-XX.XX only	mA , V
95	R	Maximum analogue signal* version 5.20-XX.XX only	mA , V
99	R	Calibration date: B0:day B1:month B2:B3:year	
120	R	Conductivity hardware version and firmware version (C.G/Y.W)	-

* Required for scaling the analogue output (see below)

The information values are readable only.
The information for No. 94 and 95 may be in mA or V, according to whether the device possesses a voltage output or a current output (see DAC_CTRL function 33).

5.2.3 Scaling of channels CH0, P1 and P2

CH0, P1 and P2 are linearly scalable with zero point and gain factor: **Value = gain factor * value + offset**
Standard values: Offset = 0.0, gain factor = 1.0
T, TOB1 and TOB2 are only scalable with zero point. Standard value: Offset = 0.0
It is also possible to influence the offset values via function 95 (see function 95).
The gain factor should be used **for calibration purposes only**, and not to alter pressure units. The latter operation should always be carried out by the master! In order to represent other pressure units via the analogue output, the unit conversion must be taken into account when scaling the analogue output.

5.2.4 Scaling the analogue output

The analogue output on the Series 30 pressure transmitters can be programmed via the interface. As the two routes *sensor-signal → digital transformation and digital value → analogue signal* are calibrated independently at the factory, the analogue output can be set to different pressures or pressure units **without** requiring recalibration. For this purpose, KELLER offers the free CCS30 software, which provides a convenient means of carrying out this scaling with a PC.

To programme the scaling of the analogue output yourself, proceed as follows:

Function 32 enables you to ascertain whether the device possesses an analogue output. The coefficients required for calculation can be read out using function 30. A new scaling can be programmed using function 31.

Read-out of pressure range for the analogue output:
The following coefficients (K[No.]) must be read out using function 30 in order to calculate the lower and upper limit of the analogue output:

```
A = (K[92] – K[68]) / K[69]
B = (K[93] – K[68]) / K[69]
```

Setting a new pressure range for the analogue output:
`K[68] and K[69]` must be calculated and written into the device using function 31:

```
K[68] = K[92] - ((K[93] – K[92]) / (B – A)) * A
K[69] = (K[93] - K[92]) / (B – A)
```

Whereby:
`K[x]:` Coefficient with the corresponding number [x] → see function 30
`A:` Pressure in bar at which the signal `K[94]` is to be output
`B:` Pressure in bar at which the signal `K[95]` is to be output
Other pressure units are to be converted into bar.

5.3 Function 31: Write coefficient

Request:

DevAddr	31	Nr.	B3	B2	B1	B0	CRC16_H	CRC16_L

Response:

DevAddr	31	0	CRC16_H	CRC16_L

Exception errors:

2 If write access is not allowed
3 If message length is incorrect
32 If device has not yet been initialised

Note:
Information on scaling of the channels: See functions 73 and 95. Information on which channels are active: See function 32 CFG_P, CFG_T and CFG_CH0.

5.4 Function 32: Read configuration

Request:

DevAddr	32	Nr.	CRC16_H	CRC16_L

Response:

DevAddr	32	Dates	CRC16_H	CRC16_L

Exception errors:

2 If no. is not available
3 If message length is incorrect
32 If device has not yet been initialised

Remark:
See description function 33

5.5 Function 33: Write configuration

Request:

DevAddr	33	Nr.	Dates	CRC16_H	CRC16_L

Response:

DevAddr	33	0	CRC16_H	CRC16_L

Exception errors:

2 If write access is not allowed
3 If message length is incorrect
32 If device has not yet been initialised

Remark:
With functions 32 and 33 one can read and write some configuration of the device. This functions provide a single byte access and replace function 100 / 101 for the devices with firmware *Class.Group -Year.Week* = 5.20-5.50 und earlier.

Description:

Nr.	Name	Description	Read	Write
0	CFG_P	Active pressure channels (high priority): Bit 1: P1 Bit 2: P2	✔	✘
1	CFG_T	Active Temperature channels (low priority): Bit 3: T (Temperature sensor) Bit 4: TOB1 (Temperature of pressure sensor P1) Bit 5: TOB2 (Temperature of pressure sensor P2) Bit 7 Con (Conductivity sensor) version 5.21-XX.XX only	✔	✘
2	CFG_CH0	version 5.20-XX.XX only Configuration of CH0 (Calculated channel): Byte value (decimal) 0: inactive 1: Difference P1 – P2 2: Difference P2 – P1 3: Square root calculation sqrt(P1), set coefficient Nr. 53 > 0 4: Square root calculation sqrt(P2) , set coefficient Nr. 53 > 0 5: Square root calculation sqrt(P1 – P2) , set coefficient Nr. 53 > 0 6: Square root calculation sqrt(P2 – P1) , set coefficient Nr. 53 > 0 11: Absolute value abs(P1) 12: Absolute value abs(P1 – P2) 13: Line pressure compensated differential pressure (only at factory, add. parameters) 14: straight line curve fitting of P1 (only if P2 is not active, 5.20-12.24 and earlier)	✔	✔ Device has to be restarted (power Off/On). Please note that for some settings there are more configurations needed.

3	CNT_T	Temperature measurement interval in seconds. version 5.20-XX.XX only	✓	✓
4	CNT_TCOMP LP-FILTER	Value of Bit 0 ... 3 (LowNibble): CNT_TCOMP version 5.20-XX.XX only After CNT_T * CNT_TCOMP seconds a temperature compensation will be performed. Value of Bit 4 ... 7 (HighNibble):Low pass filter for P1 and P2. *LowpassFilter* = $2^{Bit\,4\,...\,B7}$ The formula for the low pass filter is given as: $P_{n+1} = \frac{(2^{LowpassFiter} - 1) \cdot P_{n-1} + P_n}{2^{LowpassFiter}}$ where: P_{n+1}: new filtered value P_n: actual measured value P_{n-1}: old filtered value	✓	✓
5		--		
6		--		
7	FILTER	Filter setting for one conversion: Bit 0: Adaptive filter for P1 and P2 (on / off) Bit 1: Low pass filter for T, TOB1 and TOB2 (on / off) Bit 2.. 4: Over sampling ration OSR = $2^{(8+Bit\,2\,...\,4)}$ version 5.20-XX.XX only Bit 5.. 6: Amount of samples per averaging: 0..3 = 1, 2, 4 or 8 values version 5.20-XX.XX only Factory settings see FILTER_ORG. version 5.20-XX.XX only	✓	✓
8		--		
9	DAC	Analogue output: version5.20-XX.XX only Bit 0: Milli Amperes output (4 .. 20mA) Bit 1: Voltage output Bit 4 = 1: P1 is linked to the analogue output Bit 4 = 0: CH0 is linked to the analogue output Scaling see function 30/31	✓	(✓ : Bit4) 5.20-10.40 and earlier. Change of Bit 4 allowed to link the analogue output either to CH0 or P1
10	UART	UART settings: Bit 0 .. 3: Baud rate Baud rate Value = 0: 9'600baud Baud rate Value = 1: 115'200baud Bit 4: Parity selection. 0: no Parity, 1: Parity enable Bit 5: Parity mode. 0: odd parity, 1: even parity Bit 6: Stop bit. 0: 1 Stop bit, 1: 2 Stop bits version 5.21-XX.XX only	✓	✓ Device has to be restarted (power Off/On). version 5.20-XX-XX only version 5.21-XX.XX generates a restart -> reinitialised
11	FILTER_ORG	Factory setting for filter value. version 5.20-XX.XX only	✓	✗
12	STAT	Status of the measurement. See function 73 for details.	✓	✗
13	DEV_ADDR	Device address. Range: 1 .. 255.	✓	✓
14	P-Mode	Type of sensor and calibration. version 5.20-12.XX and later only and 5.21-XX.XX Bit 0 ... 3 (LowNibble) P1: 0: PR (relative), 1: PA (absolute), 2: PAA (absolute) Bit 4 ... 7 (HighNibble) P2: 0: PR (relative), 1: PA (absolute), 2: PAA (absolute)	✓	✗
15	SPS	Samples per Second: version 5.21-XX.XX only Range: 5-7 5: 120 SPS 6: 240 SPS 7: 480 SPS	✓	✓
20	SDI-12	version 5.21-14.XX and later only Switch to SDI-12 Mode 0x99	✗	✓
28	ConOn	version 5.21-XX.XX only 0x01: Conductivity powered 0x00: Conductivity unpowered	✓	✓
31	ConRange	version 5.21-XX.XX only 0x01: Conductivity Range 1: 0…200uS (0.2mS) 0x02: Conductivity Range 2: 0…2'000uS (2mS) 0x03: Conductivity Range 3: 0…20'000uS (20mS) 0x04: Conductivity Range 4: 0…200'000uS (200mS)	✓	✓
32	ConTempCompMode	Conductivity Temperature Compensation Method after DIN/EN 27 888 version 5.21-XX.XX only 0x01: Linear Temperature Compensation @ 25°C 0x02: Linear Temperature Compensation @ 20°C 0x03: Non-linear Temperature Compensation (Table)@ 25°C	✓	✓
33	SDI-12_available	version 5.21-13-35 and 5.21-14.35 and later only	✓	✗

		0x01: Switch to SDI-12 possible 0x00: SDI-12 Wire not available		

5.6 Function 48 : Initialise and release

Request:

DevAddr	48	CRC16_H	CRC16_L

Response:

DevAddr	48	Class	Group	Year	Week	BUF	STAT	CRC16_H	CRC16_L

Exception error:

3 If message length is incorrect

Note:
Each time the device is switched on by applying the supply voltage or after a break in the power supply, the device must be initialised via this function. Calling a different function will lead to **exception error 32**.

The following information is returned:

Class Device ID code
5: Series 30 digital pressure transmitter (33, 35, 36, 39)

Group Subdivision within a device class
1: Series 30 transmitter from 1999 or later
20: Series 30 transmitter from 2002 or later
21: Series 30 transmitter version X2
The differences between these devices are defined in italics in the functions.

Year, Week Firmware version

BUF Length of the internal receive buffer

STAT Status information
0: Device addressed for first time after switching on.
1: Device was already initialised

5.7 Function 66 : Write and read new device address

Request:

DevAddr	66	NewAddr	CRC16_H	CRC16_L

Response:

DevAddr	66	ActAddr	CRC16_H	CRC16_L

Exception error:

3 If message length is incorrect
32 If device is not yet initialised

Note:
This function programmes the device addresses to NewAddr. The address is returned in ActAddr as confirmation. It is to be ensured that the new address NewAddr is not already in use by another bus user.

Permissible addresses: 1 .. 255. Address 250 is transparent. This means that every device, irrespective of the set address, will respond to address 250. Consequently, *transparent* DevAddr = 250 may only be used in stand-alone operating mode!

For the purpose of **reading the device address** when the address is not known, for example, the value 250 is transferred as DevAddr and the value 0 is transferred as NewAddr. The current address is then returned in response.

5.8 Function 69 : Read serial number

Request:

DevAddr	69	CRC16_H	CRC16_L

Response:

DevAddr	69	SN3	SN2	SN1	SN0	CRC16_H	CRC16_L

Exception errors:

3 If message length is incorrect
32 If device is not yet initialised.

Note:
The serial number is allocated at the factory. It consists of 4 bytes unsigned integer and is calculated as follows:

$$SN = 256^3 * SN3 + 256^2 * SN2 + 256 * SN1 + SN0$$

5.9 Function 73 : Read value of a channel (floating point)

Request:

DevAddr	73	CH	CRC16_H	CRC16_L

Response:

DevAddr	73	B3	B2	B1	B0	STAT	CRC16_H	CRC16_L

Exception errors:

2 If CH > 5 version 5.20-XX.XX and if CH > 11 version 5.21-XX.XX
3 If message length is incorrect
32 If device is not yet initialised

Note:
A device can measure up to five signals (channels):
version 5.20-XX.XX: Two independent pressure sensors P1 and P2, as well as the temperatures of pressure sensors TOB1 and TOB2 and an additional temperature sensor T. The temperatures of the pressure sensors (TOB1, TOB2) are required for temperature compensation of the pressure signal.
version 5.21-XX.XX: One pressure sensor P1, as well as the temperatures of pressure sensors TOB1 and an additional temperature sensor T. Optional a conductivity sensor is available and ConTc and ConRaw can be measured as well. The temperature of the pressure sensor (TOB1) is required for temperature compensation of the pressure signal.
Please use function 32 to get the information which Channels are active.
CH0 is a calculated channel whose mode of functioning is defined in function 32 / 33.
On a standard pressure transmitter, only channels P1 and TOB1 are available. You can read out which channels are active via function 32.

The measured value is returned in IEEE754 format (4-byte B0 ... B3).

CH	Name	Description	Unit
0	CH0	Calculated channel (see function 32,33 version 5.20-XX.XX only)	*
1	P1	Pressure from pressure sensor 1	bar
2	P2	Pressure from pressure sensor 2	bar
3	T	Additional temperature sensor	°C
4	TOB1	Temperature of pressure sensor 1	°C
5	TOB2	Temperature of pressure sensor 2	°C
10	ConTc	Conductivity Temperature Compensated version 5.21-XX.XX only	mS/cm
11	ConRaw	Conductivity Raw Value version 5.21-XX.XX only	mS/cm

* Dependent on definition in function 32.

The **STAT** byte contains the current status.

Bit position	.7	.6	.5	.4	.3	.2	.1	.0
Name	/STD	ERR2	TOB2	TOB1	T	P2	P1	CH0

A set **/STD** bit indicate whether the transmitter is in Power-up mode, otherwise it is in Standard mode.

A set **ERR2** bit denotes that a computation error has occurred in the calculation process for the analogue output. This occurs if the analogue Signal is in saturation (depends on the scaling). version5.20-XX.XX only

A set **CH0, P1**, **P2**, **T**, **TOB1**, **TOB2** bit indicates that a measuring or computation error has occurred in the channel concerned.

For details in error-handling see chapter **Error handling and recognition!**

Function 74 : Read value of a channel (32bit integer)

Request:

DevAddr	74	CH	CRC16_H	CRC16_L

Response:

DevAddr	74	B3	B2	B1	B0	STAT	CRC16_H	CRC16_L

Exception errors:

2 If CH > 5

3 If message length is incorrect

4 *Class.Group -Year.Week* = 5.20-5.50 and earlier: If a channel is in overflow/underflow/inactive state or the data is invalid.
For newer versions: overflow/underflow/inactive state is showed by value. See Chapter 6.3

32 If device has not yet been initialised

Note:

Same as function 73, but values as 4-byte integer (long) B0 .. B3, where B3 is MSByte. The resolution is reduced to 0.1mbar.

Unit: CH0: $*10^{-5}$
P1 and P2: Pascal (1Pa = 10^{-5} bar).
T, TOB1 and TOB2: 0.01°C
ConTC and ConRaw are not available with F74

Status-Byte (STAT): See function 73.
For details in error-handling see chapter **Error handling and recognition!**

5.10 Function 95 : Commands for setting the zero point

Requests:

Request a:

DevAddr	95	CMD	CRC16_H	CRC16_L

Request b with set point:

DevAddr	95	CMD	*B3*	*B2*	*B1*	*B0*	CRC16_H	CRC16_L

where B3:B0: Floating-point number IEEE754 format (4-byte B0 ... B3) for the set point.

Response:

DevAddr	95	0	CRC16_H	CRC16_L

Exception errors:

1 If in Power-up mode
If data value is incorrect (NaN, INF, NINF) *Class.Group 5.21-XX.XX only*
2 If CMD invalid
3 If message length incorrect
32 If device is not yet initialised

Note:
The following actions can be carried out with this function:

CMD	Meaning
0	Set zero point of P1
1	Reset zero point of P1 to standard value
2	Set zero point of P2
3	Reset zero point of P2 to standard value
4	--
5	--
6	Set zero point of CH0
7	Reset zero point of CH0 to standard value
8	Set zero point of T version 5.21-XX.XX only
9	Reset zero point of T to standard value version 5.21-XX.XX only
10	Set zero point of TOB1 version 5.21-XX.XX only
11	Reset zero point of TOB1 to standard value version 5.21-XX.XX only
12	Set zero point of TOB2 version 5.21-XX.XX only
13	Reset zero point of TOB2 to standard value version 5.21-XX.XX only

CMD 0, 2, 6, 8, 10, 12:
Zero point values for pressure channels P1, P2 and the calculated channel CH0 and T, TOB1, TOB2. These values can also be read via function 30 and written via function 31.
Request a: The zero point is calculated such that the current measured value = 0.0.
Request b: The zero point is calculated such that the current measured value equals the set point (B3:B0).

CMD 1, 3, 7, 9, 11, 13: Reset zero point to factory setting
The zero point values are reset to 0.

5.11 Function 100 : Read configuration

Request:

DevAddr	100	Index	CRC16_H	CRC16_L

Response:

DevAddr	100	PARA0	PARA1	PARA2	PARA3	PARA4	CRC16_H	CRC16_L

Exception errors:

2 If index > 8
3 If message length is incorrect
32 If device is not yet initialised

Note:
This function supplies the information about the configuration of the device. Please use Function 32 instead of this function for devices of *Class.Group 5.20-5.24 and earlier*. With function 32/33 you have access to a single parameter instead of all five parameters.
A pressure transmitter can read two independent pressure sensors (P1 and P2), plus the temperatures of the respective pressure sensors (TOB1 and TOB2) and an independent temperature (T).

Index	**Para0**	**Para1**	**Para2**	**Para3**	**Para4**
0		UART	FILTER_ORG		
2	CFG_P	CFG_T	CFG_CH0 *	CNT_T	High Nibble | Low Nibble* LP-Filter | CNT_TCOMP
3			FILTER		DAC *

*: Not for *Class.Group 5.21-XX.XX*

For details see description in function 32 / 33. For change the configuration use function 33.

6 Appendix

6.1 Interface converter

The serial RS232 interface or the USB interface can be used for connection to a PC. KELLER offers converters for this purpose. Various other products are commercially available, however. The following requirements apply when working with KELLER software:

- The converter must control transmit / receive switch-over automatically.
- KELLER converters feature a hardware echo, i.e. the transmitted message is received again immediately as an echo. This echo is required by some KELLER software programmes.

6.2 Floating-point format IEEE754

As data transmission is effected byte-wise (8-bit data), the floating-point values are represented as follows:
B0: Bit 0..7; B1: Bit 8..15, B2: Bit 16..23, B3: Bit 24..31

Representation in accordance with IEEE754:

B3 DATA H (Reg. 0)	**B2** DATA L (Reg. 0)	**B1** DATA H (Reg. 1)	**B0** DATA L (Reg. 1)	
b01000001 (0x41)	b00101001 (0x29)	b00000010 (0x02)	b11011110 (0xDE)	Valid Number
b01111111 (0x7F)	b10000000 (0x80)	b00000000 (0x00)	b00000000 (0x00)	∞ / Overflow
b11111111 (0xFF)	b10000000 (0x80)	b00000000 (0x00)	b00000000 (0x00)	-∞ / Underflow
bx1111111 (0xFF)	b11111111 (0xFF)	b11111111 (0xFF)	b11111111 (0xFF)	Not a Number

1 bit Sign + 8 bit Exponent + 23 bit Mantis = 32 bit

Calculation of the value transmitted:

$$V = (-1)^S \cdot (1.0 + \frac{M}{2^{23}}) \cdot 2^{E-127}$$

0 = 0
10000010 = 130
01010010000001011011110 = 2687710

$-1^{0} * (1.0 + 2687710/8388608) * 2^{130-127} = 10.5631999969482421875$

These values directly show the value in the requested unit [bar] or [°C].
⇨ 10.5632 bar

Usage of Keller software:
If you use the DLL which is available from KELLER, you do not need to carry out conversion, as this is encapsulated in the DLL. If you wish to address the devices directly, however, you must convert the individual bytes into a floating-point value.
To obtain a floating-point value from the individual bytes, proceed as follows:

1. Define data structure in which an array of 4 bytes and a 32-bit floating-point value is defined at the same memory location.
2. Write the bytes into the byte array.
3. Read out the floating-point value.

You do not need to carry out any actions, therefore, as the computer attends to interpretation. Some microcontrollers have a different data structure for floating-point values. In such cases, adaptation is necessary.

6.3 Error handling and recognition

The electronic unit can read five signals: pressure values from two pressure sensors (**P1, P2**), temperature values from both of these pressure sensors (**TOB1, TOB2**) and temperature value from one additional temperature sensor (**T**). Additionally, out of these values, other values can be calculated (**CH0**). These values are described as **channels** in this documentation.
To check if the channels are active one may use function F32 (or MODBUS addresses 0x0204-0x0206) (refer to according chapter).

Measuring range
The signals are being measured with an analogue to digital converter (ADC). The measuring range is limited upwards and downwards.
For the pressure signals P1 and P2 the limitation is depending on the compensated pressure range, which can be read out with function 30 (as well as MODBUS function 3 [range 0x03xx] from version 5.20-10.XX and later and 5.21-XX.XX).
Readable range: (pressure range minimum – 10%) up to (pressure range maximum + 10%)
Is the pressure below or above this range, then the bit in the Status-Byte will be set. The measured value itself is no longer valid when the Status-Bit is set.

Dependencies
P1 and P2 are temperature-compensated and therefore they are depending on temperature. Channel CH0 is always depending on P1 and / or P2 and on a temperature channel. Is there a fault in one of these dependencies so the depending channel will follow accordingly.

6.3.1 What is new in version 5.20-10.40 (and later) and 5.21-XX.XX

Version **5.20-10.40** (available end of 2010) and later will display additionally to the Status-Bit also an error in the value itself. The format follows the special indications according to floating point arithmetic: **NaN**, **+Inf** (ovl), **-Inf** (uvl). See chapter floating-point Format IEEE754.

	CH0	P1 / P2	TOB1 / TOB2
NaN	Dependency error (P1, P2 = NaN or +/-Inf)	+/-Inf in compensating T-channel	-
+Inf	-	ADC value out of range (Overflow)	ADC value out of range (Overflow) or T > 300 °C
-Inf	-	ADC value out of range (Underflow)	ADC value out of range (Underflow) or T < - 70°C
0.000	Sqrt: (p < Pcutoff)	-	-

The values NaN, +Inf and –Inf are defined in the IEEE754 standard and are described in chapter Floating-point format IEEE754.

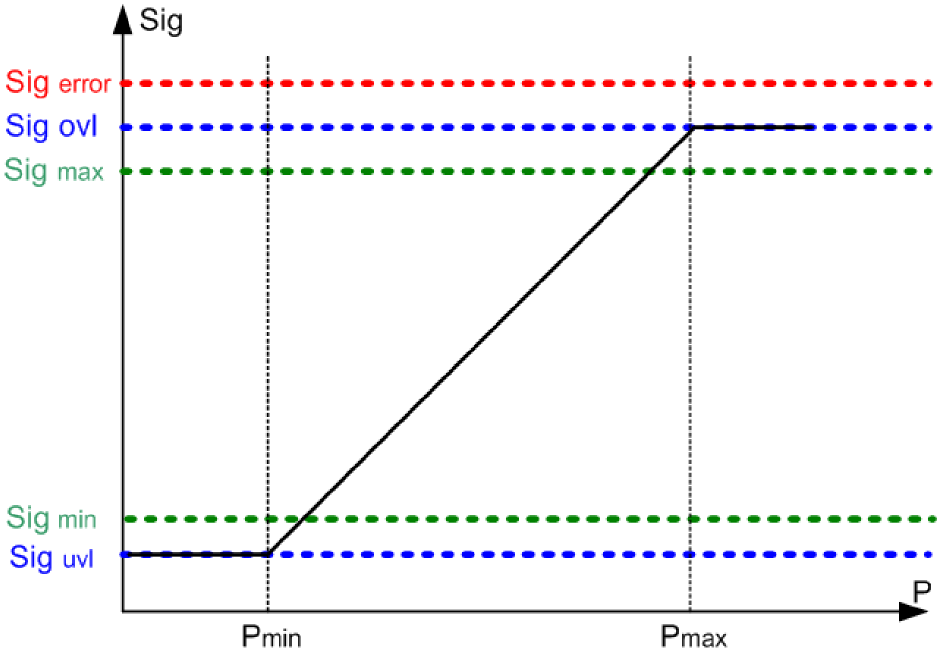
NaN is also shown, when the requested channel is not active. In this case, the Staus-Bit is not set.

Special Integer Values (F74, F3):

	32bit signed integer	16bit signed integer
NaN, +Inf	2147483647 (0x7FFFFFFF)	32767 (0x7FFF)
-Inf	-2147483648 (0x80000000)	-32768 (0x8000)

6.3.2 Analogue Output

The analogue output has to transfer the error states described in the former chapter and therefore additional analogue states were introduced. The graph below shows the transfer function between pressure (in this example the analogue output is a function of P – however it could also be some other mathematical function using CH0). The black line shows the transfer function for a positive DAC-gain. The Sig_{error} is independent of any scaling function and represents an internal error state (source has NaN, see table above).



The following table shows possible analogue values for different hardware types:

	0-10 V, (0..5V)	0-2.5 V	4-20 mA	Dig. representative
Sig error	11.6 V	2.9 V	22.5 mA	NaN
Sig ovl	11 V	2.75 V	21.8 mA	+Inf
Sig max	10 V	2.5 V	20 mA	value
Sig min	0 V	0 V	4 mA	value
Sig uvl	-1 V	-0.3 V	3.3 mA	-Inf

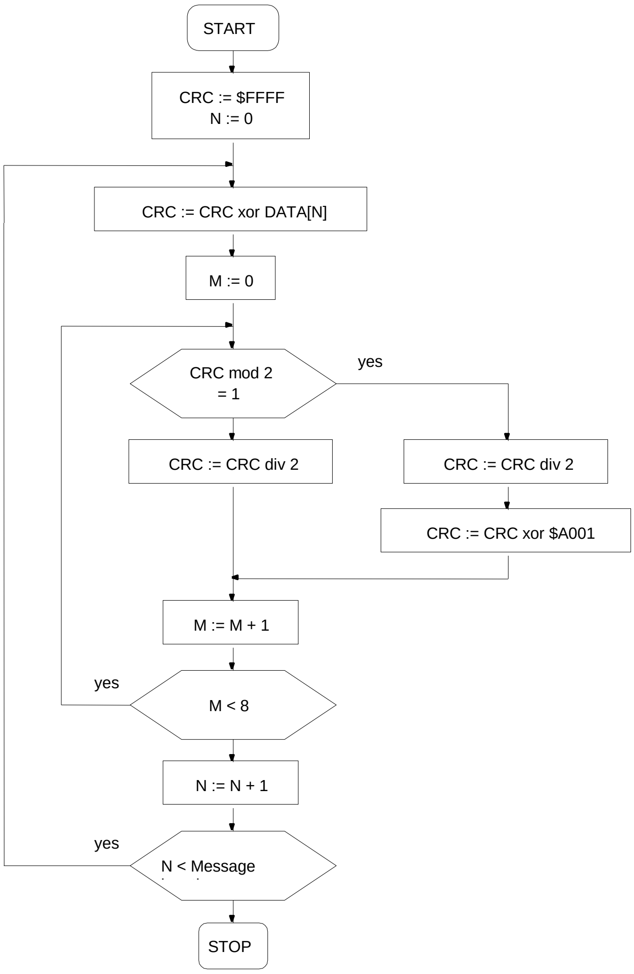
6.4 Calculation of the CRC16 checksum

The checksum can either be calculated or derived from a table.
Here is an example of CRC16 calculation in C:

```
//////////////////////////////////////////////////////////////////////////
// CRC-16 calculation in C
//
// Calculation of CRC-16 checksum over an amount of bytes in the serial buffer.
// The calculation is done without the 2byte from crc16 (receive-mode).
// SC_Buffer[]: Byte-Buffer for the serial interface. Type: unsigned char (8bit)
// SC_Amount : Amount of Bytes which should be transmitted or are received (without CRC16)
//
//////////////////////////////////////////////////////////////////////////
void CalcCRC16(unsigned char* CRC_H, unsigned char* CRC_L)
{
    // locals
    unsigned int Crc;
    unsigned char  n, m, x;

    // initialisation
    Crc= 0xFFFF;
    m= SC_Amount;
    x= 0;

    // loop over all bits
    while(m>0)
    {
        Crc^= SC_Buffer[x];
        for(n=0; n<8; n++)
        {
           if(Crc&1)
           {
               Crc>>= 1;
               Crc^= 0xA001;
           }
           else
               Crc>>= 1;
        }
        m--;
        x++;
    }
    // result
    *CRC_H= (Crc>>8)&0xFF;
    *CRC_L= Crc&0xFF;
}// end CalcCRC16
```



This results in the following calculation for function 48 with device address 250: CRC16_H= 4, CRC16_L= 67.

Examples showing use based on a table are to be found in the MODBUS documentation at:
http://www.modbus.org

6.5 Changes

- **Document version 2.1**, 19. October 2005: New function 32 and 33 for device *Class.Group -Year.Week = 05.24* and earlier, F75: cancelled.
- **Document version 2.2**, 16. August 2006:
  Description for function 32/33 revised. Function 101 added. Formula for scaling of the analogue output corrected.
- **Document version 3.0**, 20. December 2010:
  Added Modbus (chapter 4) support for *Class.Group=5.20-10.XX*
  Modbus Implementation of Functions (3,6,8 and 16) with support of all functions covered by the Keller Bus protocol.
  Added chapter error handling and recognition
  Modified chapter "Floating-point format IEEE754"
- **Document version 3.1**, 14. August 2012:
  Added changements of *Class.Group=5.20-12.28*
- **Document version 3.2**, 12. August 2013:
  New device family Class.Group=5.21-XX.XX added.
- **Document version 3.3**, 1. November 2014:
  Commands for Conductivy Sensor added in Class.Group=5.21-XX.XX.
- **Document version 3.4**, 1. June 2015:
  New commands for Conductivity Sensor added in Class.Group=5.21-XX.XX.
- **Document version 3.5**, 27. January 2017:
  Default added for Baudrate and Bus Address

6.6 Software versions

An overview of the released versions for Class.Group 5.20-XX.XX:

Version Year.Week	Date of production	Major changements
2.40	2002..2003	Base version
3.50	2003 .. 2006	- CH0: add option Line pressure compensation
5.50	2006 .. 2010	- Hardware redesign to increase robustness under EMC - add Low pass Filter and adaptive Filter for pressure channels - Switch2 option cancelled - only two temperature channels possible: TOB1 and TOB2 or T - sqrt calculation without scaling factor - add function F32 and F33 for configuration access known issue: CRC of exception code using MODBUS F3 is wrong
10.40	2011…2012	- Improved error handling: see error handling and recognition - Switch option: cancelled - CH0: Curve fitting added (CFG_CH0= 14), SF6 calculation: cancelled (CFG_CH0=7..10) MODBUS protocol: - CRC16 in case of an exception corrected - F3: access to all registers as in the KELLER protocol (Information and config values) - Serial buffer increased from 10 to 13Bytes → readout of 2 values possible in one task - F6, F8, F16 added: Configuration an calibration also with MODBUS known issue: - MODBUS F3 0x0010 ..0x001A: not usable - CH0 scale (Coeff 70 and 71) must be default (Nr.70 = 0.00, Nr. 71 = 1.00)
12.28	Aug. 2012..	- CH0 scaling: corrected - MODBUS F3: 0x020E, 0x020F: Read firmware version added Int 16Bit values 0x0010 .. 0x0015 compatible with version 5.20-5.50 0x0020 .. 0x002A: integer 32bit for process values added

An overview of the released versions for Class.Group 5.21-XX.XX:

Main differences between Group 20 and the new added Group 21.

- Group 21 has no analogue output only digital communication
- Group 21 has a SDI-12 communication interface and is able to work as SDI-12 device
- Group 21 has a higher ADC resolution for pressure and temperature channel
- Group 21 has a larger buffer size for RS485 communication.

Version Year.Week	**Date of production**	**Major revision**
13.33	August 2013	Base version
13.35	August 2013	Implementation of HW identification
14.35	August 2014	Implementation of the conductivity sensor
15.45	November 2015	Adaption of threshold values of conductivity

6.7 Support

We are pleased to offer you support in implementing the protocol. Use our free PC-software CCS30 for communication and configuration. Also divers for LabView, C#, etc. are available on our website: http://www.keller-druck.com

KELLER AG für Druckmesstechnik
St. Gallerstrasse 119 • CH-8404 Winterthur
Tel: ++41 52 235 25 25
http://www.keller-druck.com