Photoelectric Sensors
True-Color-Sensor BFS 33M with seriell interface

Interface Description

True-Color-Sensor with seriell interface

Ordercode: **BFS000L**
Typecode: **BFS 33M-GSS-F01-PU-02**

Document number: 893395 E | Edition E15 Replaces edition 1211| Subject to modification

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

Content

Chapters:

Figures:

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

Tables:

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

1. General information

This description specifies how the BFS000L/BFS000L sensor can be controlled via its serial interface. It explains the employed protocol and shows the most essential commands. Additionally it is described how the measuring values are to be interpreted and worked with, using one's own routines.

We assume that the reader knows how to build up a serial communication via the standard COM ports or even with low-level functions within a microcontroller system and its peripherals.

This manual is especially for those users who cannot use the available WIN32-DLL. A significant part of the functionality of the BFS000L/BFS000L system is encapsulated inside of the DLL. This leads to that the effort for the communication arising from the manual is much higher than it can be expected from the DLL commands.

Please observe by all means, that the serial interface nearly allows 'everything', i.e. functions and instructions that would e.g. line up the device from new as well. For that reason do not use undocumented instruction numbers or storage addresses in any case. This description only gives instructions relevant to the end-user.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

2. Safety Instructions

These photoelectric sensors may not be used in applications in which the safety of persons depends on functioning of the device (not a safety component as defined by the EU Machine Directive).

Read the manual carefully before commissioning.

LED Class 1 per DIN EN 60825-1:2003-10.
Exempt Group per IEC 62471:2006-07.
DO NOT LOOK DIRECTLY INTO THE LIGHT BEAM!
Risk of glare and irritation!

The sensor should be installed such that it is not possible to look directly into the light source during operation.

The CE Mark confirms that our products conform to the requirements of the EC Directives 2004/108/EG (EMV) and the EMC Law.
In our EMC Laboratory, which is accredited by the DATech for Testing of Electromagnetic Compatibility, we have verified that Balluff products meet the EMC requirements of EN 60947-5-2.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

3. The serial interface / interface parameters

The serial data transmission is effected with 115200 Baud. One start bit, 8 data bits and one stop bit are transferred. There are no hardware handshake signals (RTS/CTS). Thus, the hardware of the host needs to be quick enough and equipped with sufficiently large FIFOs. It may be helpful NOT to set the FIFO value of the receiver buffer to the maximum!

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

3.1. The transmission protocol

This is a mere master/slave protocol, with BFS000L/BFS000L System generally acting as the slave. Blocks of variable length are transferred. The master receives **one** block in sequence to each command, which terminates a communication sequence.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

3.2. Structure of the blocks

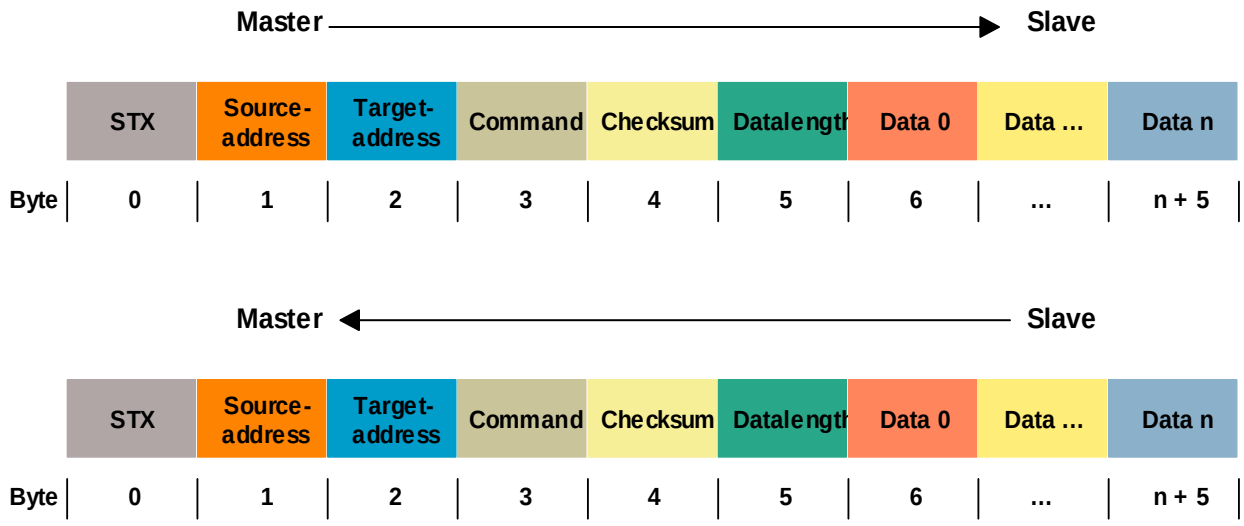

Figure 1 Structure of the blocks

The blocks transferred from the master to the slave and vice versa are constructed in the same way.

3.2.1. Byte 0, STX

The start signal at the beginning of each block (02h).

3.2.2. Byte 1, Sourceaddress

Here, the user's own participant address is always transferred. The host address is 0.

3.2.3. Byte 2, Targetaddress

The target address, device addresses possible within the range from1 to 253.

Commands to address 255 are carried out but not responded to by all bus participants.

Address 254 may only be used when exactly one participant is connected to the bus. All devices respond to address 254. This address is envisaged for the commissioning, but – to simplify matters – should also be utilized in systems connected to the standard interface of the host PC.

3.2.4. Byte 3, Command

This byte contains the command number, with possible values ranging from 0 to 239, corresponding to the defined commands. In this byte, the slave returns the previously received

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

command number, corresponding to acknowledge. If the slave has recognized an error in the checksum, it returns an ***NAK (F8h)*** in the command byte.

3.2.5. Byte 4, Checksum

The checksum is constructed in a way that the low byte arising from the sum of **all bytes of the block <u>including the checksum itself</u>** adds up to 0**.**

For data that you would like to send yourself, the checksum can be calculated as follows:

- To begin with, initialize the complete block, i.e. header of 6 byte and the data bytes possibly belonging to the command. For the checksum byte, a value of 0 is inserted first!
- Now, add together <u>all</u> bytes which belong to the prepared block. Please observe that the single bytes should be interpreted as unsigned numbers!
- The, establish the two's complement of the previously calculated sum. Like this, the sum is accordingly negated! The lowest byte of the result constitutes the correct checksum and can be entered in byte 4. After this, the complete block can be transferred.

The correct checksum of a received block can be verified e.g. as follows:

- Add together <u>all</u> bytes of the received header and all possibly belonging data bytes (the number of which arises from byte 5 of the header) as unsigned numbers.
- When the lowest byte of the previously calculated sum features a value of 0, you have received a valid block which you can work with. Otherwise, the received block is invalid!

3.2.6. Byte 5, Datalength

Contains the number of data bytes that are transferred in the following. Depending on the command, this may be 0 to 255 bytes maximum.

3.2.7. Byte 6 und following, Data 0 to Data n

Data bytes, e.g. containing the parameters of a command or slave data requested by the master. The number of these bytes relates to the value indicated in ***Datalength***. If values of more than one byte (short, long, etc.) are supposed to be transferred with this data packet, the LSB is sent first with each.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

4. Instruction manual

We already pointed out that, in case of 16 or 32 bit values, the low byte always needs to be transferred before the high byte. This corresponds to the so called INTEL notation also usual in Windows PC programs. In addition with the BFS000L System very often floating point numbers are used. Those match the C data type FLOAT, are 32 bit wide and comply with the format defined in IEEE-754.

If not differently indicated all following parameters are always integer values. All parameters where FLOAT values have to be considered theses are explicitly mentioned.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

4.1. Change or read gain, Command 3

This command can be used to change the gain to a new value or to read the actual gain from the sensor. In both cases the command returns the actual gain setting of the sensor. When changing the gain the new setting becomes active immediately. However it will only be saved permanently if the command for saving the remanent parameters (see **chapter 4.7**) is used before the next power-up of the sensor.

The equivalent DLL function for the command described here is:

iPR126_ChangeGAIN(unsigned short * pusGAIN, BOOL bReadOnly, unsigned int uiPR126Address);

Command from Host: CMD_CHANGE_GAIN (3) **Number of data bytes from Host: 4**	
Data 1..2	**Change,** change or read gain
Data 3..4	**Gain,** value the gain should be changed to

Table 1 Change or read gain

The parameter ***Change*** defines if the gain should be changed (value unequal 0) or if the actual gain should be read (value equal). If the value of parameter ***Change*** is unequal 0 the gain will be changed to the value given with parameter ***Gain***.

The answer from the sensor for the command described above is shown in the following table.

Answer from Sensor **Number of data bytes from Sensor: 4**	
Data 1..2	**Change,** corresponds to value when sending command
Data 3..4	**Gain,** actual gain on sensor

Table 2 Answer for changing or reading gain

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

4.2. Change or read status auto-gain, Command 23

This command can be used to change the status of the automatic gain setting or to read the actual setting of the status of the automatic gain setting. In both cases the command returns the actual setting of the status of the automatic gain setting from the sensor. When changing the Auto-Gain the new setting becomes active immediately. However it will only be saved permanently if the command for saving the remanent parameters (see **chapter 4.7**) is used before the next power-up of the sensor.

The equivalent DLL function for the command described here is:

iPR126_ChangeAutoGainEnableState(unsigned short * pusState, BOOL bReadOnly, unsigned int uiPR126Address);

Command from Host: CMD_CHANGE_STATUS_ENABLE_AUTO_GAIN (23) **Number of data bytes from Host: 4**	
Data 1..2	**Change,** change or read status automatic gain setting
Data 3..4	**EnableAutoGain,** enable Auto-Gain (>= 1) or disable (= 0)

Table 3 Change or read status auto-gain

The parameter ***Change*** defines if the status of the automatic gain setting should be changed (value unequal 0) or the actual status of the automatic gain setting should be read (value equal 0). If the value of parameter ***Change*** is unequal 0 the automatic gain setting will be enabled if the value of parameter ***EnableAutoGain*** is greater than 0, otherwise the automatic gain setting is disabled.

The answer from the sensor for the command described above is shown in the following table.

Answer from Sensor **Number of data bytes from Sensor: 4**	
Data 1..2	**Change,** corresponds to value when sending command
Data 3..4	**EnableAutoGain,** actual status of auto-gain setting on sensor

Table 4 Answer for changing or reading status auto-gain

The automatic gain setting is enabled if the value of parameter ***EnableAutoGain*** is greater than 0, otherwise it is disabled.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

4.3. Change or read normalization, Command 30

This command can be used to change the normalization setting of the sensor or to read the actual normalization setting from the sensor. In both cases the command returns the actual normalization setting of the sensor. When changing the normalization the new setting becomes active immediately. However it will only be saved permanently if the command for saving the remanent parameters (see **chapter 4.7**) is used before the next power-up of the sensor.

The equivalent DLL function for the command described here is:

iPR126_ChangeNormalization(struct PR126DLL_CHANGE_NORMALIZATION * pstrChangeNormalization, unsigned int uiPR126Address);

Command from Host: CMD_CHANGE_NORMALISATION (30) **Number of data bytes from Host: 10**	
Data 1..2	**Change,** change or read normalization setting
Data 3..6	**IEEE754-32-Bit Float** **Factor,** value must be -1 in any case!!!
Data 7..10	**IEEE754-32-Bit Float** **YGoal,** Tristimulus Y target value that should be used for normalization

Table 5 Change or read normalization

The answer from the sensor for the command described above is shown in the following table.

Answer from Sensor **Number of data bytes from Sensor: 10**	
Data 1..2	**Change,** corresponds to value when sending command
Data 3..6	**IEEE754-32-Bit Float** **Factor,** internal factor for normalization
Data 7..10	**IEEE754-32-Bit Float** **YGoal,** internal Tristimulus Y value for normalization

Table 6 Answer for changing or reading normalization

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

4.4. Change or read number cycles averaging, Command 39

This command can be used to change the number of cycles for averaging or to read the actual number of cycles for averaging from the sensor. In both cases the command returns the actual number of cycles for averaging of the sensor. When changing the averaging the new setting becomes active immediately. However it will only be saved permanently if the command for saving the remanent parameters (see **chapter 4.7**) is used before the next power-up of the sensor.

The equivalent DLL function for the command described here is:

iPR126_ChangeAveraging(signed long * pslNbrAveragingCycles, BOOL bReadOnly, unsigned int uiPR126Address);

Command from Host: CMD_CHANGE_NO_CYCLES_AVERAGING (39) **Number of data bytes from Host: 6**	
Data 1..2	**Change,** change or read number of cycles for averaging
Data 3..6	**NbrAveragingCycles,** value for number of cycles for averaging to be set on sensor

Table 7 Change or read number cycles averaging

The parameter ***Change*** defines if the number of cycles for averaging should be changed (value unequal 0) or if the actual number of cycles for averaging should be read (value equal 0). If the value of parameter ***Change*** is unequal 0 the number of cycles for averaging will be changed to the value given with parameter ***NbrAveragingCycles***. In any case the value of parameter ***NbrAveragingCycles*** must be greater than 0 because otherwise the change will not be completed.

The answer from the sensor for the command described above is shown in the following table.

Answer from Sensor **Number of data bytes from Sensor: 6**	
Data 1..2	**Change,** corresponds to value when sending command
Data 3..6	**NbrAveragingCycles,** actual number of cycles for averaging on sensor

Table 8 Answer for changing or reading number cycles averaging

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

4.5. Read number products, Command 43

This command can be used to read the available number of products from the sensor.

The equivalent DLL function for the command described here is:

iPR126_ReadNumberProducts(unsigned short * pusNumberProducts, unsigned int uiPR126Address);

Command from Host: CMD_READ_NO_PRODUCTS (43) **Number of data bytes from Host: 4**	
Data 1..4	Value of all bytes must be 0 in any case!!!

Table 9 Read number products

The answer from the sensor for the command described above is shown in the following table.

Answer from Sensor **Number of data bytes from Sensor: 4**	
Data 1..2	Value is irrelevant!!!
Data 3..4	**NbrPossibleProducts,** actual number of available products on sensor

Table 10 Answer for read number products

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

4.6. Read status sensor, Command 44

This command can be used to read the actual status and CIELab mode data from a BFS000L sensor.

The equivalent DLL function for the command described here is:

iPR126_ReadCIELabFullState(struct PR126DLL_CIELAB_FULL_STATE * pstrCIELabFullState, unsigned int uiPR126Address);

Command from: CMD_READ_CIELAB_FULL_STATE (44) **Number of data bytes from Host: 0**
The command does not need additional data bytes

Table 11 Read status sensor

The answer from the sensor for the command described above is shown in the following table.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

Answer from Sensor **Number of data bytes from Sensor: 98**	
Data 1..4	**StateBits,** bit coded status sensor (see below)
Data 5..8	**IEEE754-32-Bit Float** **Actual_dE_1,** Product no. 1 – actual product error (dE)
…	…
Data 33..36	**IEEE754-32-Bit Float** **Actual_dE_8,** Product no. 8 – actual product error (dE)
Data 37..40	**IEEE754-32-Bit Float** **Actual_dE_ab_1,** Product no. 1 – for future options
…	…
Data 65..68	**IEEE754-32-Bit Float** **Actual_dE_ab_8,** Product no. 8 – for future options
Data 69..72	**IEEE754-32-Bit Float** **X,** actual normalized Tristimulus X measurement value
Data 73..76	**IEEE754-32-Bit Float** **Y,** actual normalized Tristimulus Y measurement value
Data 77..80	**IEEE754-32-Bit Float** **Z,** actual normalized Tristimulus Z measurement value
Data 81..84	**IEEE754-32-Bit Float** **CIELab L,** actual CIELab L measurement value
Data 85..88	**IEEE754-32-Bit Float** **CIELab a,** actual CIELab a measurement value
Data 89..92	**IEEE754-32-Bit Float** **CIELab b,** actual CIELab b measurement value
Data 93..96	**IEEE754-32-Bit Float** **Temperature,** actual temperature sensor
Data 97..98	**ActualGain,** actual gain sensor

Table 12 Answer for reading status sensor

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

With this command the sensor returns the actual status and the actual measurement results for the CIELab mode.

The parameter ***StateBits*** contains the bit coded information from the sensor described in the following. If not differently indicated the description refers to the active state of the bit (the bit is set). Bit positions not indicated here are used for internal purpose and should not be considered.

Attention! The bit positions are given 0 based.

- Bit-No.5, ***OVERLOAD_RED,*** is set if red sensor channel was overdriven during last measurement.
- Bit-No.6, ***OVERLOAD_GREEN,*** is set if green sensor channel was overdriven during last measurement.
- Bit-No.7, ***OVERLOAD_BLUE,*** is set if blue sensor channel was overdriven during last measurement.
- Bit-No.8, ***IPARAMS_CHANGED,*** is set if internal parameters of sensor have been changed since switching on power but have not been changed remanent on sensor.
- Bit-No.9, ***IPARAMS_UPDATE_ACTIVE,*** is set if internal process for saving remanent parameters has been started and is still in progress
- Bit-No.10, ***HSO1_ACTIVE,*** corresponds to status of digital output “Out1”.
- Bit-No.11, ***HSO2_ACTIVE,*** corresponds to status of digital output “Out2”.
- Bit-No.12, ***HSO3_ACTIVE,*** corresponds to status of digital output “Out3”.
- Bit-No.13, ***VIRTUAL_HSO4_ACTIVE,*** is set if product “Background” has been determined as “best fitting” product.
- Bit-No.14, ***ACTIVE_MEASURETYPE,*** is set with active work mode “Precise” and is reset with active work mode “Best Fit”.
- Bit-No.19, ***AUTOGAIN_ACTIVE,*** is set if automatic gain setting is active on sensor.
- Bit-No.27, ***ENVIRONMENTAL_COMPENSATION_ACTIVE,*** is set if environment compensation is active on sensor.

The parameters ***Actual_dE_1...Actual_dE_8*** contain the actual products errors (dE) for the products during the last measurement. The value of one of those parameters is -1 if the according product has not been enabled or if the product number is higher than the number of available products on the sensor.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

4.7. Save remanent parameter, Command 13

This command stores all internal parameters permanently on a BFS000L sensor. You should not use this function more often than really needed. Instead please do all you changes first which affect remanent parameters (up to then, they are only changed in RAM) and then finally save all parameters permanently by the use of this command.

The remanent parameters should only be modified if they lead to noticeable modifications! The remanent parameters are saved in FLASH which can be rewritten approximately 5000 times! Permanent overwriting during a normal process can quickly exceed this number and can cause damage of the remanent memory!

There is no equivalent DLL function available.

Command from: CMD_UPDATE_REMANENT_FLASH_PARAMS (13) **Number of data bytes from Host: 0**
The command does not need additional data bytes

Table 13 Save remanent parameter

The answer from the sensor for the command described above is shown in the following table.

Answer from Sensor **Number of data bytes from Sensor: 2**	
Data 1..2	**StatusSave,** status saving parameter

Table 14 Answer for saving remanent parameter

The parameter ***StatusSave*** contains the status of saving the remanent parameters. The available return values are described in the following.

- **CMD_NO_ERROR** (0x00), saving of remanent parameters started successfully, ***this does not mean that saving the remanent parameters has been finished***
- **UPDATE_ALREADY_IN_PROGRESS** (0x0A), a previously started saving of the remanent parameters has been started and is not finished, ***saving the remanent parameters has NOT been started***

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

After the command finished with the return value *CMD_NO_ERROR* the sensor internally starts a process for saving the remanent parameters. While the internal saving process is active the status bit 9 (see ***chapter 4.6*** – parameter *StateBits*) is set. The internal process could not be started if it is already running on the sensor. In that case the command will return the value *UPDATE_ALREADY_IN_PROGRESS*.

After sending this command, it is good practice to read the status of the sensor (see ***chapter 4.6***) continuously until the status bit 9 has been reset by the sensor. If the status bit 9 is reset the internal process for saving the remanent parameters has been finished.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

4.8. Change or read product parameter, Command 16

The command can be used to either change the product parameters of a certain product or to read the actual parameters of a certain product. In both cases the command returns the actual parameters of the product. The allowed range of product numbers for this command can be read with the command 43 (see **chapter 4.5**). When changing the product parameters the new settings become active immediately. However they will only be saved permanently if the command for saving the remanent parameters (see **chapter 4.7**) is used before the next power-up of the sensor.

There is no equivalent DLL function available.

Command from Host: CMD_CHANGE_PRODUCTS (16) **Number of data bytes from Host: 86**	
Data 1..2	**Change,** change or read product parameter for product number
Data 3..4	**ProductNumber,** product number the product parameters should be changed or read (0..NbrPossibleProducts-1)
Data 5..6	**ProductEnabled,** product enabled (= 1) or disabled (= 0) for product determination on sensor

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

Data 7..10	**IEEE754-32-Bit Float** **Spare01,** for future option – use default value = 1.0
Data 11..14	**IEEE754-32-Bit Float** **Spare02,** for future option – use default value = 1.0
Data 15..18	**IEEE754-32-Bit Float** **Spare03,** for future option – use default value = 1.0
Data 19..22	**IEEE754-32-Bit Float** **Spare04,** for future option – use default value = 1.0
Data 23..26	**IEEE754-32-Bit Float** **Spare05,** for future option – use default value = 1.0
Data 27..30	**IEEE754-32-Bit Float** **Spare06,** for future option – use default value = 1.0
Data 31..34	**IEEE754-32-Bit Float** **Spare07,** for future option – use default value = 1.0
Data 35..38	**IEEE754-32-Bit Float** **Spare08,** for future option – use default value = 1.0
Data 39..42	**IEEE754-32-Bit Float** **Spare09,** for future option – use default value = 1.0
Data 43..46	**IEEE754-32-Bit Float** **TargetCIELab_L,** target value CIELab L for product
Data 47..50	**IEEE754-32-Bit Float** **TargetCIELab_a,** target value CIELab a for product
Data 51..54	**IEEE754-32-Bit Float** **TargetCIELab_b,** target value CIELab b for product
Data 55..58	**IEEE754-32-Bit Float** **Spare13,** for future option – use default value = 0.0
Data 59..62	**IEEE754-32-Bit Float** **Spare14,** for future option – use default value = 0.0

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

Data 63..66	**IEEE754-32-Bit Float** **MaxAllowedDeltaE,** maximum allowed Delta E (dE) for product (for 'Precise' mode, otherwise used default = 0.0)
Data 67..70	**IEEE754-32-Bit Float** **Spare16,** for future option – use default value = 0.0
Data 71..74	**IEEE754-32-Bit Float** **Spare17,** for future option – use default value = 0.0
Data 75..78	**IEEE754-32-Bit Float** **Spare18,** for future option – use default value = 1.0
Data 79..82	**IEEE754-32-Bit Float** **Spare19,** for future option – use default value = 1.0
Data 83..86	**IEEE754-32-Bit Float** **Spare20,** for future option – use default value = 1.0

Table 15 Change or read product parameter

The parameter ***Change*** defines if the product parameters should be changed (value unequal 0) or if the actual product parameters should be read (value equal 0). The parameter ***ProductNumber*** defines the product number the parameter should be changed or read. The allowed range for that parameter is 0...NbrPossibleProducts-1 (see **chapter 4.5**). The parameter ***ProductEnabled*** defines if the product number is enabled (=1) or disabled (=0) for the product determination on the sensor. For the test on the sensor only 'enabled' products will be considered. The target CIELab values for the product are given with the product parameters ***TargetCIELab_L***, ***TargetCIELab_a*** and ***TargetCIELab_b***. The parameter ***MaxAllowedDeltaE*** defines the maximum allowed Delta E between the actual CIELab sensor values and the defined CIELab Target values that must be met for that product. The parameter does have only an impact for the 'Precise' mode (see **chapter 4.9**). Otherwise the default value of 0.0 should be used.

All other parameters for this command are provided for future options and should remain at their given default values.

The answer from the sensor for the command described above is shown in the following table.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

Answer from Sensor **Number of data bytes from Sensor: 86**	
Data 1..2	**Change,** corresponds to value when sending command
Data 3..4	**ProductNumber,** corresponds to value when sending command
Data 5..6	**ProductEnabled,** product enabled (= 1) or disabled (= 0) for product determination on sensor

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

Data 7..10	**IEEE754-32-Bit Float** **Spare01,** for future option – do not use
Data 11..14	**IEEE754-32-Bit Float** **Spare02,** for future option – do not use
Data 15..18	**IEEE754-32-Bit Float** **Spare03,** for future option – do not use
Data 19..22	**IEEE754-32-Bit Float** **Spare04,** for future option – do not use
Data 23..26	**IEEE754-32-Bit Float** **Spare05,** for future option – do not use
Data 27..30	**IEEE754-32-Bit Float** **Spare06,** for future option – do not use
Data 31..34	**IEEE754-32-Bit Float** **Spare07,** for future option – do not use
Data 35..38	**IEEE754-32-Bit Float** **Spare08,** for future option – do not use
Data 39..42	**IEEE754-32-Bit Float** **Spare09,** for future option – do not use
Data 43..46	**IEEE754-32-Bit Float** **TargetCIELab_L,** target value CIELab L for product
Data 47..50	**IEEE754-32-Bit Float** **TargetCIELab_a,** target value CIELab a for product
Data 51..54	**IEEE754-32-Bit Float** **TargetCIELab_b,** target value CIELab b for product
Data 55..58	**IEEE754-32-Bit Float** **Spare13,** for future option – do not use
Data 59..62	**IEEE754-32-Bit Float** **Spare14,** for future option – do not use

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

Data 63..66	**IEEE754-32-Bit Float** **MaxAllowedDeltaE,** maximum allowed Delta E (dE) for product (for 'Precise' mode, ignore otherwise)
Data 67..70	**IEEE754-32-Bit Float** **Spare16,** for future option – do not use
Data 71..74	**IEEE754-32-Bit Float** **Spare17,** for future option – do not use
Data 75..78	**IEEE754-32-Bit Float** **Spare18,** for future option – do not use
Data 79..82	**IEEE754-32-Bit Float** **Spare19,** for future option – do not use
Data 83..86	**IEEE754-32-Bit Float** **Spare20,** for future option – do not use

Table 16 Answer for changing or reading product parameter

With this command the sensor returns the actual product parameter for the given product number.

The parameters ***TargetCIELab_L***, ***TargetCIELab_a*** and ***TargetCIELab_b*** contain the actual product settings for the target CIELab values of the product. The parameter ***MaxAllowedDeltaE*** returns the maximum allowed Delta E between the actual CIELab sensor values and the target CIELab values defined for the product. The parameter does have only an impact for the 'Precise' mode (see **chapter 4.9**). Otherwise the given value can be ignored.

All other values are provided for future options and should be ignored.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

4.9. Change or read measure type, Command 34

This command can be used to change the actual measure type setting or to read the actual measure type setting from a BFS000L sensor. In both cases the command returns the actual measure type setting. When changing the measure type the new setting becomes active immediately. However it will only be saved permanently if the command for saving the remanent parameters (see **chapter 4.7**) is used before the next power-up of the sensor.

In 'Best Fit' mode the sensor does not consider any defined product tolerances. The sensor always internally processes the parameter of all active products and as a result returns the product number to which the actual sensor values best fit. For the determination of the best fitting product the Delta-E between the actual sensor values and the defined target values for any saved and active product are calculated. If more than one of the saved products shows the exact same variation to the actual values, then the lowest of those products numbers is returned. In this mode always one products will be detected as best fitting product, independent from the given product parameters. The mode 'Best Fit' is best suited for the selection of objects out of several (known products).

In 'Precise' mode a specific product number as test result is only given if the actual values of the sensor are within the given tolerances for that product. If the actual values are inside of the tolerances of multiple products no clearly correlation to a product is possible. If the actual sensor values are outside the tolerances of any saved product, no product is detected. The 'Precise' mode is best suited e.g. for quality inspection of test objects, because here the tolerances for color and intensity must fit in order to detect an object of that kind.

There is no equivalent DLL function available.

Command from Host: CMD_CHANGE_MEASURE_TYPE (34) **Number of data bytes from Host: 4**	
Data 1..2	**Change,** change or read measure type
Data 3..4	**MeasureType,** value for measure type to be set on sensor

Table 17 Change or read measure type

The parameter ***Change*** defines if the measure type should be changed (value unequal 0) or if the actual measure type should be read (value equal 0). If the value of parameter ***Change*** is unequal 0 the measure type will be changed to the value given with parameter ***MeasureType***. The measure type will be changed to the 'Best Fit' mode for a parameter value of 0 and will be changed to the 'Precise' mode when using a parameter value of 1. Any other value will change the measure type to the default mode ('Precise').

The answer from the sensor for the command described above is shown in the following table.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

Answer from Sensor **Number of data bytes from Sensor: 4**	
Data 1..2	**Change,** corresponds to value when sending command
Data 3..4	**MeasureType,** actual measure type used on sensor

Table 18 Answer for changing or reading measure type

With the answer the sensor returns the actual measure type used by the sensor. For a value of 0 the ‘Best Fit’ mode is used, otherwise the ‘Precise’ mode is used by the sensor.

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com