Optoelektronische Sensoren
True-Color-Sensor BFS 33M mit serieller Schnittstelle

Befehlsbeschreibung

True-Color-Sensor mit serieller Schnittstelle

Bestellcode: **BFS000L**
Typenbezeichnung: **BFS 33M-GSS-F01-PU-02**

Dokumentnummer: 893393 D | Ausgabe E15 ersetzt Ausgabe 1211| Änderungen vorbehalten

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

Inhalt

Kapitel:

Abbildungen:

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

Tabellen:

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. Allgemeines

Diese Beschreibung bezieht sich auf die Möglichkeiten zur Steuerung des Sensors BFS 33M-GSS-F01-PU-02 über dessen serielle Schnittstelle. Das verwendete Protokoll wird erläutert und die Befehle dargestellt. Zusätzlich wird beschrieben, wie die Messwerte zu interpretieren sind und wie sie mit eigenen Routinen weiterverarbeitet werden können.

Es wird vorausgesetzt, dass dem Leser bekannt ist, wie er von seiner Plattform aus eine serielle Kommunikation über die Standard COM-Ports aufbauen kann.

Diese Dokumentation richtet sich an diejenigen Anwender die die zur Verfügung stehende WIN32-DLL nicht nutzen können. Ein nicht unerheblicher Teil der Funktionalität des BFS 33M-GSS-F01-PU-02 Sensors ist dabei innerhalb der DLL gekapselt. Daraus ergibt sich, dass der Kommunikationsaufwand der sich aus dieser Dokumentation hier ergibt deutlich größer ist als es die DLL erwarten lässt.

Es sei dringend darauf hingewiesen, dass über die serielle Schnittstelle „alles“ mit dem Sensor möglich ist, also auch Funktionen und Befehle, die das Gerät z.B. neu abgleichen würden. Deshalb sollten auf keinen Fall undokumentierte Befehlsnummern oder Speicheradressen verwendet werden. Es werden ausschließlich die Befehle beschrieben, die für den Endanwender relevant sind.

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. Sicherheitshinweise

Diese optoelektronischen Sensoren dürfen nicht in Anwendungen eingesetzt werden, in denen die Sicherheit von Personen von der Gerätefunktion abhängt (kein Sicherheitsbauteil gemäß EU-Maschinenrichtlinie).

Vor Inbetriebnahme ist die Betriebsanleitung sorgfältig zu lesen.

LED Klasse 1 nach DIN EN 60825-1:2003-10.
Freie Gruppe nach IEC 62471:2006-07.
NICHT IN DEN LICHTSTRAHL BLICKEN!
Gefahr von Blendung und Irritation!

Der Sensor ist so zu montieren, dass auch während des Betriebs kein direkter Blick in die Lichtquelle möglich ist.

Mit dem CE-Zeichen bestätigen wir, dass unsere Produkte den Anforderungen der EG-Richtlinien 2004/108/EG (EMV) und des EMV-Gesetzes entsprechen.
In unserem EMV-Labor, das von der DATech für Prüfungen der elektromagnetischen Verträglichkeit akkreditiert ist, wurde der Nachweis erbracht, dass die Balluff-Produkte die EMV-Anforderungen der Norm EN 60947-5-2 erfüllen.

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. Die Serielle Schnittstelle

3.1. Die Schnittstellenparameter

Die serielle Datenübertragung erfolgt mit 115200 Baud. Es wird ein Startbit, 8 Datenbits und ein Stoppbit übertragen. Es existieren keine Hardware_Handshakesignale (RTS/CTS). Die Hardware des Host muss also schnell genug sein und mit ausreichend großen FIFOs bestückt sein. Dabei kann es hilfreich sein, die FIFO Größe des Empfangspuffers NICHT auf das Maximum einzustellen!

3.2. Das Übertragungsprotokoll

Es handelt sich um ein reines Master/Slave Protokoll, wobei das BFS 33M-GSS-F01-PU-02 Sensor als Slave agiert. Es werden Blöcke mit variabler Länge übertragen. Auf jeden Befehl erhält der Master **einen** Block als Antwort, damit ist eine Kommunikationssequenz abgeschlossen.

3.3. Blockaufbau

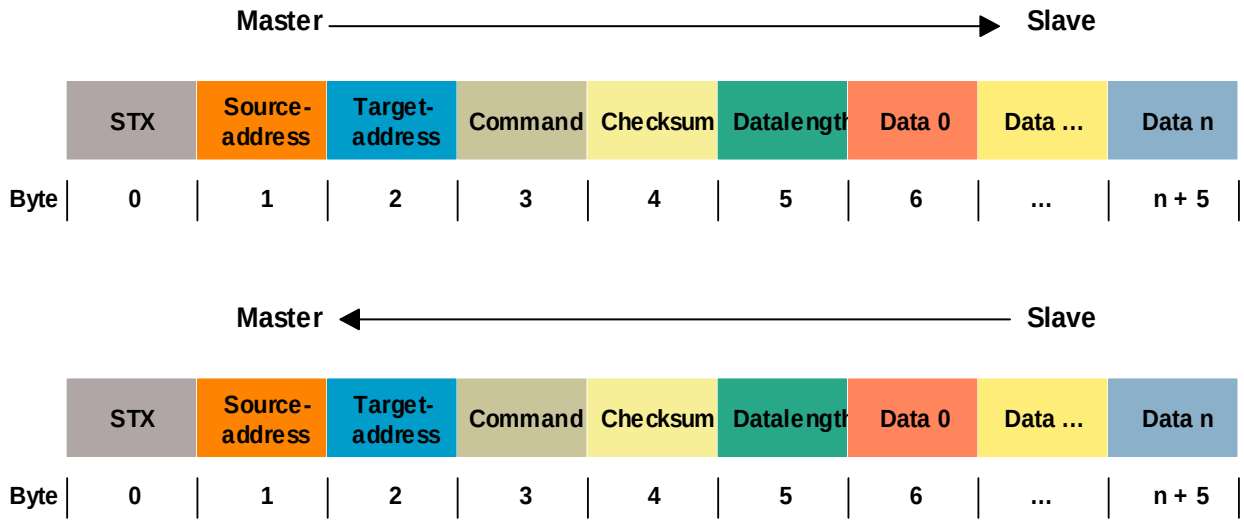

Abbildung 1: Aufbau der Blöcke

Die Blöcke vom Master zum Slave sowie die vom Slave zum Master sind gleich aufgebaut.

3.3.1. Byte 0, STX

Das Startzeichen an jedem Blockanfang (02h).

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

3.3.2. Byte 1, Sourceaddress
Hier wird immer die eigene Teilnehmeradresse übertragen, der Host hat die Adresse 0.

3.3.3. Byte 2, Targetaddress
Die Zieladresse, Geräteadressen sind im Bereich von 1 bis 253 möglich.

Befehle an die Adresse 255 werden von allen Busteilnehmern ausgeführt, aber von keinem beantwortet.

Die Adresse 254 darf nur benutzt werden, wenn nur ein Teilnehmer am Bus hängt. Auf die Adresse 254 antwortet jedes Gerät. Diese Adresse ist für die Inbetriebnahme vorgesehen, sollte der Einfachheit halber auch in Systemen verwendet werden die an einer Standardschnittstelle des Host-PC angeschlossen sind.

3.3.4. Byte 3, Command
Dieses Byte enthält die Befehlsnummer, hier sind Werte von 0 bis 239 möglich, entsprechend den definierten Befehlen. Der Slave sendet in diesem Byte jeweils die vorher an ihn übermittelte Befehlsnummer zurück. Das entspricht einem Acknowledge. Sollte der Slave einen Checksummenfehler erkannt haben, sendet er im Command Byte ein ***NAK (F8h)*** zurück.

3.3.5. Byte 4, Checksum
Die Checksumme ist so aufgebaut, dass das Lowbyte aus der Summe **von allen Bytes des Blocks incl. der Checksumme selbst** immer den Wert 0 ergibt**.**

Für Daten die Sie selbst versenden wollen kann die Checksumme z.B. folgendermaßen berechnet werden:

- Zuerst den kompletten Block initialisieren, also den 6-Byte großen Header und die eventuell zum Kommando gehörenden Datenbytes. Für das Checksummenbyte wird zunächst der Wert 0 eingesetzt.
- Nun alle Bytes die zu dem präparierten Block gehören aufaddieren. Bitte beachten Sie bitte, dass die einzelnen Bytes als vorzeichenlose Zahl interpretiert werden sollten!
- Bilden Sie nun das 2-er Komplement der zuvor ermittelten Summe. Das entspricht dem negieren der Summe! Das niederwertigste Byte des Ergebnisses ist die korrekte Checksumme und kann nun in Byte 4 eingetragen werden. Danach können Sie den ganzen Block versenden.

Für einen empfangenen Block können Sie die korrekte Checksumme z.B. folgendermaßen überprüfen:

- Addieren Sie alle Bytes des empfangenen Headers und alle eventuell dazu gehörenden Datenbytes (deren Zahl ergibt sich aus Byte 5 des Headers) als vorzeichenlose Zahlen auf.

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

- Wenn das niederwertigste Byte der zuvor gebildeten Summe den Wert 0 hat haben Sie einen gültigen Block empfangen und können ihn weiter verarbeiten. Ansonsten ist der empfangene Block ungültig!

3.3.6. Byte 5, Datalength

Enthält die Anzahl der Datenbytes, die im Folgenden übertragen werden. Je nach Befehl können das 0 bis maximal 255 Bytes sein.

3.3.7. Byte 6 und folgende, Data 0 bis Data n

Datenbytes, die z.B. Parameter zu einem Befehl oder auch Daten des Slaves enthalten, die vom Master angefragt wurden. Die Anzahl dieser Bytes entspricht dem in ***Datalength*** angegebenen Wert. Sollen innerhalb dieses Datenpaketes Werte übertragen werden, die mehr als ein Byte belegen (short, long, usw.) so wird jeweils das LSB zuerst gesendet.

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. Befehlsbeschreibung

Bei 16- oder 32-Bit Werten muss immer zuerst das Low-Byte und dann das High-Byte übertragen werden. Das entspricht der sogenannten INTEL-Schreibweise die auch in PC-Programmen unter Windows üblich ist. Zusätzlich werden beim BFS 33M-GSS-F01-PU-02 Sensor auch häufig Gleitkommazahlen verwendet. Diese entsprechen der C-Datentype FLOAT, sind 32-Bit groß und entsprechen dem durch die IEEE-754 definierten Format.

Sofern nicht anders angegeben sind alle im Folgenden beschriebenen Parameter immer Integer-Werte. Da wo die genannten FLOAT-Werte berücksichtigt werden müssen ist das auch explizit aufgeführt.

4.1. Verstärkung ändern oder einlesen, Befehl 3

Mit diesem Befehl wird die Verstärkung auf einen neuen Wert eingestellt bzw. die aktuelle Verstärkung des Sensors eingelesen. Das Kommando liefert in beiden Fällen immer die aktuelle Verstärkung des Sensors. Beim Ändern wird der neue Wert für die Verstärkung zwar sofort übernommen. Die Einstellung wird jedoch nur remanent auf dem Sensor gespeichert falls das entsprechende Kommando (siehe **Kapitel 4.7**) vor dem nächsten Ausschalten des Sensors verwendet wurde.

Das DLL-äquivalent zu dem hier beschriebenen Kommando ist die Funktion.

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

Befehl vom Host: CMD_CHANGE_GAIN (3) **Anzahl der Datenbytes vom Host: 4**	
Data 1..2	**Change,** Verstärkung ändern oder einlesen
Data 3..4	**Gain,** Verstärkung die eingestellt werden soll

Tabelle 1: Verstärkung ändern oder einlesen

Der Parameter ***Change*** bestimmt ob die Verstärkung geändert (Wert ungleich 0) oder die aktuelle Verstärkung eingelesen (Wert gleich 0) werden soll. Falls der Parameter ***Change*** einen Wert ungleich 0 hat dann wird die Verstärkung auf den mit Parameter ***Gain*** vorgegebenen Wert eingestellt.

Auf das zuvor beschriebene Kommando antwortet der Sensor entsprechend der folgenden Tabelle.

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

Antwort des Sensors **Anzahl der Datenbytes vom Sensor: 4**	
Data 1..2	**Change,** Wert entspricht dem beim Senden des Kommandos
Data 3..4	**Gain,** aktuelle Verstärkung des Sensors

Tabelle 2: Antwort für Verstärkung ändern oder einlesen

4.2. Status Enable Auto-Gain ändern oder einlesen, Befehl 23

Mit diesem Befehl wird der Status der automatischen Verstärkungseinstellung geändert oder der aktuelle Status der automatischen Verstärkungseinstellung des Sensors eingelesen. Das Kommando liefert in beiden Fällen immer die aktuelle Einstellung der automatischen Verstärkungseinstellung des Sensors. Beim Ändern wird der neue Wert für den Auto-Gain zwar sofort übernommen. Die Einstellung wird jedoch nur remanent auf dem Sensor gespeichert falls das entsprechende Kommando (siehe **Kapitel 4.7**) vor dem nächsten Ausschalten des Sensors verwendet wurde.

Das DLL-äquivalent zu dem hier beschriebenen Kommando ist die Funktion.

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

Befehl vom Host: CMD_CHANGE_STATUS_ENABLE_AUTO_GAIN (23) **Anzahl der Datenbytes vom Host: 4**	
Data 1..2	**Change,** Status Auto-Gain ändern oder einlesen
Data 3..4	**EnableAutoGain,** Auto-Gain aktivieren (>= 1) oder deaktivieren (= 0)

Tabelle 3: Status Auto-Gain ändern oder einlesen

Der Parameter ***Change*** bestimmt ob der Status der automatischen Verstärkungseinstellung geändert (Wert ungleich 0) oder der aktuelle Status eingelesen (Wert gleich 0) werden soll. Falls der Parameter ***Change*** einen Wert ungleich 0 hat dann wird die automatische Verstärkungseinstellung aktiviert wenn der Wert des Parameters ***EnableAutoGain*** größer 0 ist, andernfalls wird diese deaktiviert.

Auf das zuvor beschriebene Kommando antwortet der Sensor entsprechend der folgenden Tabelle.

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

Antwort des Sensors **Anzahl der Datenbytes vom Sensor: 4**	
Data 1..2	**Change,** Wert entspricht dem beim Senden des Kommandos
Data 3..4	**EnableAutoGain,** aktuelle Einstellung der automatischen Verstärkungseinstellung des Sensors

Tabelle 4: Antwort für Status Auto-Gain ändern oder einlesen

Die automatische Verstärkungseinstellung ist aktiv falls der Parameter ***EnableAutoGain*** einen Wert größer 0 hat, andernfalls ist diese inaktiv.

4.3. Normalisierung ändern oder einlesen, Befehl 30

Mit diesem Befehl wird die Normalisierung des Sensors geändert oder die aktuelle Einstellung der Normalisierung vom Sensor eingelesen. Das Kommando liefert in beiden Fällen die aktuelle Einstellung der Normalisierung des Sensors. Beim Ändern wird der neue Wert für die Normalisierung zwar sofort übernommen. Die Einstellung wird jedoch nur remanent auf dem Sensor gespeichert falls das entsprechende Kommando (siehe **Kapitel 4.7**) vor dem nächsten Ausschalten des Sensors verwendet wurde.

Das DLL-äquivalent zu dem hier beschriebenen Kommando ist die Funktion.

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

Befehl vom Host: CMD_CHANGE_NORMALISATION (30) **Anzahl der Datenbytes vom Host: 10**	
Data 1..2	**Change,** Normalisierung ändern oder einlesen
Data 3..6	**IEEE754-32-Bit Float** **Factor,** muss immer den Wert -1 haben!!!
Data 7..10	**IEEE754-32-Bit Float** **YGoal,** Tristimulus Y Zielwert auf den normalisiert werden soll

Tabelle 5: Normalisierung ändern oder einlesen

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

Auf das zuvor beschriebene Kommando antwortet der Sensor entsprechend der folgenden Tabelle.

Antwort des Sensors **Anzahl der Datenbytes vom Sensor: 10**	
Data 1..2	**Change,** Wert entspricht dem beim Senden des Kommandos
Data 3..6	**IEEE754-32-Bit Float** **Factor,** interner Faktor der Normalisierung
Data 7..10	**IEEE754-32-Bit Float** **YGoal,** interner Tristimulus Y Wert der Normalisierung

Tabelle 6: Antwort für Normalisierung ändern oder einlesen

4.4. Anzahl Zyklen Mittelwertbildung ändern oder einlesen, Befehl 39

Mit diesem Befehl kann die Anzahl von Zyklen für die Mittelwertbildung geändert oder der aktuelle Wert des Sensors eingelesen werden. Das Kommando liefert in beiden Fällen immer die aktuelle Einstellung der Zyklen für die Mittelwertbildung des Sensors. Beim Ändern wird der neue Wert für die Mittelwertbildung zwar sofort übernommen. Die Einstellung wird jedoch nur remanent auf dem Sensor gespeichert falls das entsprechende Kommando (siehe **Kapitel 4.7**) vor dem nächsten Ausschalten des Sensors verwendet wurde.

Das DLL-äquivalent zu dem hier beschriebenen Kommando ist die Funktion.

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

Befehl vom Host: CMD_CHANGE_NO_CYCLES_AVERAGING (39) **Anzahl der Datenbytes vom Host: 6**	
Data 1..2	**Change,** Anzahl Zyklen Mittelwertbildung ändern oder einlesen
Data 3..6	**NbrAveragingCycles,** Anzahl von Zyklen Mittelwertbildung die auf dem Sensor eingestellt werden soll

Tabelle 7: Anzahl Zyklen Mittelwertbildung ändern oder einlesen

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

Der Parameter ***Change*** bestimmt ob die Anzahl der Zyklen für die Mittelwertbildung geändert (Wert ungleich 0) oder die aktuelle Anzahl der Zyklen für die Mittelwertbildung eingelesen (Wert gleich 0) werden soll. Falls der Parameter ***Change*** einen Wert ungleich 0 hat dann wird die Anzahl der Zyklen für die Mittelwertbildung auf dem Sensor auf den mit dem Parameter ***NbrAveragingCycles*** vorgegebenen Wert geändert. Der Parameters muss auf jeden Fall einen Wert größer 0 haben da ansonsten die Änderung nicht ausgeführt wird.

Auf das zuvor beschriebene Kommando antwortet der Sensor entsprechend der folgenden Tabelle.

Antwort des Sensors **Anzahl der Datenbytes vom Sensor: 6**	
Data 1..2	**Change,** Wert entspricht dem beim Senden des Kommandos
Data 3..6	**NbrAveragingCycles,** aktuelle eingestellte Anzahl von Zyklen für die Mittelwertbildung

Tabelle 8: Antwort für Zyklen Mittelwertbildung ändern oder einlesen

4.5. Anzahl Produkte einlesen, Befehl 43

Mit diesem Befehl kann die auf dem Sensor verfügbare Anzahl von Produkten eingelesen werden

Das DLL-äquivalent zu dem hier beschriebenen Kommando ist die Funktion.

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

Befehl vom Host: CMD_READ_NO_PRODUCTS (43) **Anzahl der Datenbytes vom Host: 4**	
Data 1..4	Wert aller Bytes muss immer 0 sein!!!

Tabelle 9: Anzahl Produkte einlesen

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

Auf das zuvor beschriebene Kommando antwortet der Sensor entsprechend der folgenden Tabelle.

Antwort des Sensor **Anzahl der Datenbytes vom Sensor: 4**	
Data 1..2	Wert ist irrelevant!!!
Data 3..4	**NbrPossibleProducts,** aktuelle Einstellung der Anzahl von verfügbaren Produkten

Tabelle 10: Antwort für Anzahl Produkte einlesen

4.6. Status Sensor einlesen, Befehl 44

Mit diesem Befehl wird der aktuelle Status mit den Daten des CIELab Modus vom Sensor eingelesen.

Das DLL-äquivalent zu dem hier beschriebenen Kommando ist die Funktion.

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

Befehl vom Host: READ_CIELAB_FULL_STATE (44) **Anzahl der Datenbytes vom Host: 0**
Diese Kommando benötigt keine weiteren Datenbytes

Tabelle 11: Status Sensor einlesen

Auf das zuvor beschriebene Kommando antwortet der Sensor entsprechend der folgenden Tabelle.

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

Antwort des Sensors **Anzahl der Datenbytes vom Sensor: 98**	
Data 1..4	**StateBits,** bitkodierter Zustand des Sensors
Data 5..8	**IEEE754-32-Bit Float** **Actual_dE_1,** Produktnummer 1 – aktueller Produktfehler des Produktes
…	…
Data 33..36	**IEEE754-32-Bit Float** **Actual_dE_8,** Produktnummer 8 – aktueller Produktfehler des Produktes
Data 37..40	**IEEE754-32-Bit Float** **Actual_dE_ab_1,** Produktnummer 1 – für zukünftige Erweiterungen, z.Zt. ohne Funktion
…	…
Data 65..68	**IEEE754-32-Bit Float** **Actual_dE_ab_8,** Produktnummer 8 – für zukünftige Erweiterungen, z.Zt. ohne Funktion
Data 69..72	**IEEE754-32-Bit Float** **X,** aktueller normierter Tristimulus Messwert X
Data 73..76	**IEEE754-32-Bit Float** **Y,** aktueller normierter Tristimulus Messwert Y
Data 77..80	**IEEE754-32-Bit Float** **Z,** aktueller normierter Tristimulus Messwert Z
Data 81..84	**IEEE754-32-Bit Float** **CIELab L,** aktueller CIELab Messwert L
Data 85..88	**IEEE754-32-Bit Float** **CIELab a,** aktueller CIELab Messwert a
Data 89..92	**IEEE754-32-Bit Float** **CIELab b,** aktueller CIELab Messwert b

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 93..96	**IEEE754-32-Bit Float** **Temperatur,** aktuelle Temperatur des Sensors
Data 97..98	**ActualGain,** aktuell eingestellte Verstärkung des Sensors

Tabelle 12: Antwort für Einlesen Status Sensor

Mit diesem Kommando liefert der Sensor den aktuellen Status und die aktuellen Messergebnisse im CIELab Modus.

Der Parameter ***StateBits*** liefert bitcodiert den aktuellen Zustand des Sensors. Wenn nicht anders angegeben bezieht sich die Beschreibung der Bits immer auf den aktiven Zustand (das Bit ist gesetzt). Die nicht beschriebenen Bitpositionen sind für die interne Verwendung reserviert und sollten nicht berücksichtigt werden.

Achtung! Die Bitpositionen sind 0-basierend angegeben.

- Bit-Nr.5, ***OVERLOAD_RED,*** ist gesetzt falls bei der letzten Messung beim roten Sensorkanal eine Übersteuerung vorlag.
- Bit-Nr.6, ***OVERLOAD_GREEN,*** ist gesetzt falls bei der letzten Messung beim grünen Sensorkanal eine Übersteuerung vorlag.
- Bit-Nr.7, ***OVERLOAD_BLUE,*** ist gesetzt falls bei der letzten Messung beim blauen Sensorkanal eine Übersteuerung vorlag.
- Bit-Nr.8, ***IPARAMS_CHANGED,*** ist gesetzt falls interne Parameter des Sensors seit dem Einschalten verändert wurden, diese aber noch nicht remanent auf dem Sensor gespeichert sind.
- Bit-Nr.9, ***IPARAMS_UPDATE_ACTIVE***, ist gesetzt wenn der interne Prozess zum Speichern der remanenten Parameter gestartet aber noch nicht beendet wurde
- Bit-Nr.10, ***HSO1_ACTIVE,*** entspricht dem Zustand des digitalen Ausgangs ‚Out1'.
- Bit-Nr.11, ***HSO2_ACTIVE,*** entspricht dem Zustand des digitalen Ausgangs ‚Out2'.
- Bit-Nr.12, ***HSO3_ACTIVE,*** entspricht dem Zustand des digitalen Ausgangs ‚Out3'.
- Bit-Nr.13, ***VIRTUAL_HSO4_ACTIVE,*** ist gesetzt wenn das Produkt ‚Background' als am besten passendes Produkt vom Sensor erkannt wurde.
- Bit-Nr.14, ***ACTIVE_MEASURETYPE,*** ist gesetzt bei aktiver Betriebsart ‚Precise' und ist zurückgesetzt bei aktiver Betriebsart ‚Best Fit'.
- Bit-Nr.19, ***AUTOGAIN_ACTIVE,*** ist gesetzt wenn die automatische Verstärkungseinstellung des Sensors aktiv ist.
- Bit-Nr.27, ***ENVIRONMENTAL_COMPENSATION_ACTIVE,*** ist gesetzt wenn die Umgebungskompensation auf dem Sensor aktiv ist.

Die Parameter ***Actual_dE_1 .. Actual_dE_8*** liefern die Produktfehler (dE) der Produkte während der letzten Messung des Sensors. Der Wert eines der Parameter ist -1 falls das entsprechende Produkt nicht freigegeben ist oder falls die Produktnummer höher als die verfügbare Anzahl von Produkten ist.

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. Parameter remanent speichern, Befehl 13

Der Befehl speichert alle internen Parameter remanent auf dem BFS000L Sensor. Das Kommando sollte nicht öfter wie tatsächlich notwendig verwendet werden. Anstelle dessen sollten alle Einstellungen die remanente Parameter betreffen erst einmal geändert werden (bis dahin existieren die Änderungen nur im RAM), und dann final mit Hilfe dieses Kommandos remanent auf dem Sensor gespeichert werden.

Änderungen an den remanenten Parametern sollten nur dann durchgeführt werden, wenn sich dadurch auch tatsächlich etwas ändert. Die remanenten Daten werden im FLASH gesichert das ca. 5000-mal neu beschrieben werden kann. Beim ständigen Überschreiben aus dem normalen Prozessablauf heraus kann diese Zahl sehr schnell überschritten werden, was dann zur Zerstörung des remanenten Speichers führen kann!

Es existiert kein DLL-äquivalent zu dem hier beschriebenen Kommando.

Befehl vom Host: READ_UPDATE_REMANENT_FLASH_PARAMS (13) **Anzahl der Datenbytes vom Host: 0**
Diese Kommando benötigt keine weiteren Datenbytes

Tabelle 13: Remanente Parameter speichern

Auf das zuvor beschriebene Kommando antwortet der Sensor entsprechend der folgenden Tabelle.

Antwort des Sensor **Anzahl der Datenbytes vom Sensor: 2**	
Data 1..2	**StatusSave,** Status speichern Parameter

Tabelle 14: Antwort für Speichern remanente Parameter

Der Parameter ***StatusSave*** enthält den Status des Speichervorgangs der remanenten Parameter. Die verfügbaren Rückgabewerte sind im Folgenden näher beschrieben.

- **CMD_NO_ERROR** (0x00), der interne Vorgang zum Speichern der remanenten Parameter wurde erfolgreich gestartet, **dies bedeutet jedoch nicht das das Speichern der remanenten Parameter auf dem Sensor schon beendet wurde**
- **UPDATE_ALREADY_IN_PROGRESS** (0x0A), ein vorher gestarteter Vorgang zum Speichern der remanenten Parameter ist noch nicht beendet, **ein neuer Vorgang zum Speichern der remanenten Parameter konnte nicht gestartet werden**

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

Wenn das Kommando mit dem Rückgabewert *CMD_NO_ERROR* beendet wurde, dann startet auf dem Sensor ein interner Prozess zum Speichern der remanenten Parameter. Solange dieser interne Prozess aktiv ist wird vom Sensor das Statusbit 9 (siehe **Kapitel 4.6** – Parameter StateBits) gesetzt. Dieser interne Prozess kann nicht gestartet werden falls dieser bereits auf dem Sensor läuft. In diesem Falle liefert das Kommando den Rückgabewert *UPDATE_ALREADY_IN_PROGRESS*.

Nach dem Senden des Kommandos wird es empfohlen den Status des Sensors (siehe **Kapitel 4.6**) kontinuierlich abzufragen bis das Status-Bit 9 vom Sensor zurückgesetzt wurde. Ist das Status Bit 9 zurückgesetzt dann ist der interne Prozess des Sensors zum Speichern der remanenten Parameter beendet.

4.8. Produkt Parameter ändern oder einlesen, Befehl 16

Der Befehl dient zum Ändern bzw. Einlesen der Produktparameter eines spezifischen Produktes des Sensors. Das Kommando liefert in beiden Fällen immer die aktuellen Produkt Parameter des angegebenen Produktes. Beim Ändern werden die neuen Produkt Parameter zwar sofort übernommen. Die Einstellung wird jedoch nur remanent auf dem Sensor gespeichert falls das entsprechende Kommando (siehe **Kapitel 4.7**) vor dem nächsten Ausschalten des Sensors verwendet wurde.

Es existiert kein DLL-äquivalent zu dem hier beschriebenen Kommando.

Befehl vom Host: CMD_CHANGE_PRODUCTS (16) **Anzahl der Datenbytes vom Host: 86**	
Data 1..2	**Change,** Produkt Parameter ändern oder einlesen
Data 3..4	**ProductNumber,** Produktnummer für die Produkt Parameter geändert oder eingelesen werden sollen (0..NbrPossibleProducts-1)
Data 5..6	**ProductEnabled,** Produkt aktiv (= 1) oder inaktiv (= 0) bei Auswertung Produkte auf 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,** für zukünftige Erweiterungen – Defaultwert (1.0) verwenden
Data 11..14	**IEEE754-32-Bit Float** **Spare02,** für zukünftige Erweiterungen – Defaultwert (1.0) verwenden
Data 15..18	**IEEE754-32-Bit Float** **Spare03,** für zukünftige Erweiterungen – Defaultwert (1.0) verwenden
Data 19..22	**IEEE754-32-Bit Float** **Spare04,** für zukünftige Erweiterungen – Defaultwert (1.0) verwenden
Data 23..26	**IEEE754-32-Bit Float** **Spare05,** für zukünftige Erweiterungen – Defaultwert (1.0) verwenden
Data 27..30	**IEEE754-32-Bit Float** **Spare06,** für zukünftige Erweiterungen – Defaultwert (1.0) verwenden
Data 31..34	**IEEE754-32-Bit Float** **Spare07,** für zukünftige Erweiterungen – Defaultwert (1.0) verwenden
Data 35..38	**IEEE754-32-Bit Float** **Spare08,** für zukünftige Erweiterungen – Defaultwert (1.0) verwenden
Data 39..42	**IEEE754-32-Bit Float** **Spare09,** für zukünftige Erweiterungen – Defaultwert (1.0) verwenden
Data 43..46	**IEEE754-32-Bit Float** **TargetCIELab_L,** Sollwert CIELab L für Produkt
Data 47..50	**IEEE754-32-Bit Float** **TargetCIELab_a,** Sollwert CIELab a für Produkt
Data 51..54	**IEEE754-32-Bit Float** **TargetCIELab_b,** Sollwert CIELab b für Produkt

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 55..58	**IEEE754-32-Bit Float** **Spare13,** für zukünftige Erweiterungen – Defaultwert (0.0) verwenden
Data 59..62	**IEEE754-32-Bit Float** **Spare14,** für zukünftige Erweiterungen – Defaultwert (0.0) verwenden
Data 63..66	**IEEE754-32-Bit Float** **MaxAllowedDeltaE,** maximal erlaubte Abweichung Delta E für Produkt (für ‚Precise' Mode, andernfalls Default = 0.0)
Data 67..70	**IEEE754-32-Bit Float** **Spare16,** für zukünftige Erweiterungen – Defaultwert (0.0) verwenden
Data 71..74	**IEEE754-32-Bit Float** **Spare17,** für zukünftige Erweiterungen – Defaultwert (0.0) verwenden
Data 75..78	**IEEE754-32-Bit Float** **Spare18,** für zukünftige Erweiterungen – Defaultwert (1.0) verwenden
Data 79..82	**IEEE754-32-Bit Float** **Spare19,** für zukünftige Erweiterungen – Defaultwert (1.0) verwenden
Data 83..86	**IEEE754-32-Bit Float** **Spare20,** für zukünftige Erweiterungen – Defaultwert (1.0) verwenden

Tabelle 15: Produkt Parameter ändern oder einlesen

Der Parameter ***Change*** bestimmt ob die Produkt Parameter des Produktes geändert und gelesen (Wert ungleich 0) oder nur gelesen (Wert gleich 0) werden sollen. Der Parameter ***ProductNumber*** definiert die Produktnummer dessen Produkt Parameter geändert oder gelesen werden sollen. Der erlaubte Wertebereich des Parameters ist dabei 0…NbrPossibleProducts-1 (siehe **Kapitel 4.5**). Der Parameter ***ProductEnabled*** bestimmt ob die Produktnummer für die Produktauswertung auf dem Sensor aktiviert (= 1) oder deaktiviert (= 0) werden soll. Für den Test auf dem Sensor werden nur Produkte berücksichtigt die auch aktiv sind. Die CIELab Sollwerte des Produktes werden mit den Parametern ***TargetCIELab_L***, ***TargetCIELab_a*** und ***TargetCIELab_b*** übermittelt. Der Parameter ***MaxAllowedDeltaE*** bestimmt die maximal erlaubte Abweichung (Toleranz) Delta-E zwischen den aktuellen CIELab Sensorwerten und den definierten CIELab Sollwerten die für das Produkt erreicht werden soll. Der Parameter wird nur in der ‚Precise' Betriebsart verwendet (siehe **Kapitel 4.9**). Andernfalls sollte der Defaultwert 0.0 verwendet werden.

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

Alle anderen Parameter sind für zukünftige Erweiterungen vorgesehen. Für diese Parameter sollte immer der vorgegebene Defaultwert verwendet werden.

Auf das zuvor beschriebene Kommando antwortet der Sensor entsprechend der folgenden Tabelle.

Antwort des Sensors **Anzahl der Datenbytes vom Sensor: 86**	
Data 1..2	**Change,** Wert entspricht dem beim Senden des Kommandos
Data 3..4	**ProductNumber,** Wert entspricht dem beim Senden des Kommandos
Data 5..6	**ProductEnabled,** Produkt aktiv (= 1) oder inaktiv (= 0) bei Auswertung Produkte auf 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,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 11..14	**IEEE754-32-Bit Float** **Spare02,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 15..18	**IEEE754-32-Bit Float** **Spare03,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 19..22	**IEEE754-32-Bit Float** **Spare04,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 23..26	**IEEE754-32-Bit Float** **Spare05,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 27..30	**IEEE754-32-Bit Float** **Spare06,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 31..34	**IEEE754-32-Bit Float** **Spare07,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 35..38	**IEEE754-32-Bit Float** **Spare08,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 39..42	**IEEE754-32-Bit Float** **Spare09,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 43..46	**IEEE754-32-Bit Float** **TargetCIELab_L,** aktueller Sollwert CIELab L für Produkt
Data 47..50	**IEEE754-32-Bit Float** **TargetCIELab_a,** aktueller Sollwert CIELab a für Produkt
Data 51..54	**IEEE754-32-Bit Float** **TargetCIELab_b,** aktueller Sollwert CIELab b für Produkt

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 55..58	**IEEE754-32-Bit Float** **Spare13,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 59..62	**IEEE754-32-Bit Float** **Spare14,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 63..66	**IEEE754-32-Bit Float** **MaxAllowedDeltaE,** maximal erlaubte Abweichung Delta E für Produkt (für ‚Precise' Mode, andernfalls ignorieren)
Data 67..70	**IEEE754-32-Bit Float** **Spare16,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 71..74	**IEEE754-32-Bit Float** **Spare17,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 75..78	**IEEE754-32-Bit Float** **Spare18,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 79..82	**IEEE754-32-Bit Float** **Spare19,** für zukünftige Erweiterungen – Wert nicht verwenden
Data 83..86	**IEEE754-32-Bit Float** **Spare20,** für zukünftige Erweiterungen – Wert nicht verwenden

Tabelle 16: Antwort für Einlesen Produkt Parameter

Mit diesem Kommando liefert der Sensor die aktuellen Produkt Parameter für die verwendete Produktnummer.

Die Parameter ***TargetCIELab_L***, ***TargetCIELab_a*** und ***TargetCIELab_b*** enthalten die aktuellen CIELab Sollwerte des Produktes. Der Parameter ***MaxAllowedDeltaE*** liefert die maximal erlaubte Abweichung (Toleranz) Delta E zwischen den aktuellen CIELab Sensorwerten und den CIELab Sollwerten der verwendeten Produktnummer. Der Parameter hat nur in der ‚Precise' Betriebsart (siehe **Kapitel 4.9**) eine Bedeutung. Andernfalls kann der Wert ignoriert werden.

Alle weiteren Parameter sind für zukünftige Erweiterungen vorgesehen und können ignoriert werden.

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. Betriebsart ändern oder einlesen, Befehl 34

Der Befehl dient zum Ändern bzw. Einlesen der Betriebsart des Sensors. Das Kommando liefert in beiden Fällen immer die aktuell eingestellte Betriebsart des Sensors. Beim Ändern wird die neue Betriebsart zwar sofort übernommen. Die Einstellung wird jedoch nur remanent auf dem Sensor gespeichert falls das entsprechende Kommando (siehe **Kapitel 4.7**) vor dem nächsten Ausschalten des Sensors verwendet wurde.

Im ‚Best Fit' Modus berücksichtigt der Sensor keine vorgegebenen Produkttoleranzen. Der Sensor verarbeitet intern immer die Parameter aller aktiven Produkte und gibt als Ergebnis immer die Produktnummer zurück der die aktuellen Istwerte am nächsten kommen. Zu Ermittlung des am besten passenden Produktes wird das Delta-E zwischen aktuellen Istwerten des Sensors und den Sollwerten jedes gespeicherten Produktes berechnet. Sollten mehrere gespeicherte Produkte die exakt gleiche Abweichung zu den Istwerten haben dann wird die niedrigste der Produktnummern zurückgegeben. In diesem Mode wird unabhängig von den eingestellten Produktparametern immer eines der Produkte als ‚am besten' passendes Produkt erkannt! Die Betriebsart ‚Best Fit' eignet sich zu Auswahl eines Objektes aus mehreren möglichen (und bekannten).

Im ‚Precise' Modus wird nur dann eine bestimmte Produktnummer als Ergebnis ausgegeben wenn die aktuellen Istwerte des Sensors innerhalb der vorgegebenen Toleranzen für das Produkt liegen. Liegen die Istwerte innerhalb der Toleranzen von mehreren Produkten dann ist keine eindeutige Produktzuordnung möglich. Falls die Istwerte außerhalb der Toleranzen aller gespeicherter Produkte liegen dann ist ebenfalls keine Produktzuordnung möglich. Die Betriebsart ‚Precise' eignet sich z.B. zur Qualitätssicherung von Objekten da hier die Toleranzen für Farbe und Intensität eingehalten werden müssen damit ein Objekt erkannt wird.

Es existiert kein DLL-äquivalent zu dem hier beschriebenen Kommando.

Befehl vom Host: CMD_CHANGE_MEASURE_TYPE (34) **Anzahl der Datenbytes vom Host: 4**	
Data 1..2	**Change,** Betriebsart ändern oder einlesen
Data 3..4	**MeasureType,** Wert für Betriebsart die auf dem Sensor eingestellt werden soll

Tabelle 17: Betriebsart ändern oder einlesen

Der Parameter ***Change*** definiert ob die Betriebsart geändert und gelesen (Wert ungleich 0) oder nur gelesen (Wert = 0) werden soll. Falls der Parameter ***Change*** einen Wert ungleich 0 hat dann wird die Betriebsart auf den mit Parameter ***MeasureType*** angegebenen Wert geändert. Die Betriebsart wird auf ‚Best Fit' geändert falls der Parameter einen Wert von 0 hat und wird auf ‚Precise' geändert falls der Parameter einen Wert von 1 hat. Jeder andere Wert stellt die Betriebsart auf den Defaultwert ‚Precise' ein.

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

Auf das zuvor beschriebene Kommando antwortet der Sensor entsprechend der folgenden Tabelle.

Antwort des Sensors **Anzahl der Datenbytes vom Sensor: 4**	
Data 1..2	**Change,** Wert entspricht dem beim Senden des Kommandos
Data 3..4	**MeasureType,** aktuelle eingestellte Betriebsart des Sensors

Tabelle 18: Antwort für Betriebsart ändern oder einlesen

Mit der Antwort für das Kommando liefert der Sensor die aktuell eingestellte Betriebsart des Sensors. Hat der Parameter ***MeasureType*** den Wert 0 dann ist die Betriebsart ‚Best Fit' aktiv, andernfalls wird die Betriebsart ‚Precise' vom Sensor verwendet.

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