

Intelligente servoelektrische Greifmodule der WSG-Serie
Befehlssatzreferenz

Firmware Version 4.0
Dezember 2016



www.weiss-robotics.com

Inhalt

1	Einleitung	4
1.1	Aufbau einer Verbindung zum Greifmodul	4
1.2	Kommunikation mit dem Greifmodul.....	5
1.2.1	Befehlsformat.....	5
1.2.2	Antwortnachrichten vom Greifmodul.....	6
1.3	Asynchrone Befehle	7
1.4	Schnittstellenoptionen	8
1.4.1	Deaktivieren der CRC-Prüfsumme	8
1.4.2	Deaktivieren des Fehlers beim unangekündigten Beenden einer TCP-Verbindung.....	9
2	WSG-Commander	10
3	Befehlsreferenz	12
3.1	Verbindungsverwaltung	12
3.1.1	Loop (06h)	12
3.1.2	Ankündigung der Verbindungstrennung (07h)	13
3.2	Bewegungssteuerung	14
3.2.1	Referenzierungsfahrt (20h).....	14
3.2.2	Vorpositionieren der Finger (21h).....	16
3.2.3	Stop (22h).....	19
3.2.4	Auslösen eines Fast Stop (23h)	20
3.2.5	Quittieren eines Fast Stop oder einer Fehlerbedingung (24h)	21
3.2.6	Teil greifen (25h)	22
3.2.7	Teil loslassen (26h).....	24
3.3	Bewegungskonfiguration	25
3.3.1	Setzen der Beschleunigung (30h).....	25
3.3.2	Auslesen der Beschleunigung (31h).....	26
3.3.3	Setzen der Greifkraft (32h)	27
3.3.4	Auslesen der Greifkraft (33h).....	28
3.3.5	Setzen der Soft Limits (34h)	29
3.3.6	Auslesen der Soft Limits (35h)	31
3.3.7	Aufheben der Soft Limits (36h)	32
3.3.8	Overdrive-Modus (37h).....	33
3.3.9	Kraftsensor tarieren (38h).....	35
3.4	Systemzustandsbefehle	36
3.4.1	Systemzustand auslesen (40h).....	36
3.4.2	Greifzustand auslesen (41h)	38
3.4.3	Greifstatistik auslesen (42h)	40
3.4.4	Öffnungsweite auslesen (43h)	41

3.4.5	Fingergeschwindigkeit auslesen (44h)	42
3.4.6	Greifkraft auslesen (45h)	43
3.4.7	Temperatur auslesen (46h).....	44
3.5	Systemkonfiguration	45
3.5.1	Systeminformation auslesen (50h)	45
3.5.2	Device Tag setzen (51h)	46
3.5.3	Device Tag auslesen (52h).....	47
3.5.4	Systemgrenzen auslesen (53h)	48
3.6	Fingerschnittstelle.....	49
3.6.1	Finger 1 Informationen auslesen (60h).....	49
3.6.2	Finger 1 Zustandsflags auslesen (61h)	50
3.6.3	Finger 1 Stromversorgung ein-/ausschalten (62h)	51
3.6.4	Finger 1 Daten auslesen (63h)	52
3.6.5	Finger 2 Informationen auslesen (70h).....	53
3.6.6	Finger 2 Zustandsflags auslesen (71h)	54
3.6.7	Finger 2 Stromversorgung ein-/ausschalten (72h)	55
3.6.8	Finger 2 Daten auslesen (73h)	56
Anhang A.	Statuscodes	57
Anhang B.	Systemzustandsflags.....	59
Anhang C.	Greifzustände	62
Anhang D.	Beispielcode zum Berechnen der CRC-Prüfsumme.....	63

1 Einleitung

Die Greifmodule der WSG-Serie können unter Verwendung eines Binärprotokolls über verschiedene Standardschnittstellen gesteuert werden. Dieses Handbuch beschreibt detailliert sowohl das Protokoll selbst als auch die notwendigen Befehle zum Steuern der Greifmodule. Zum Einstieg in das Kommunikationsprotokoll empfehlen wir die Verwendung der mitgelieferten Software „WSG Commander“ für Microsoft Windows.

 **Soweit nicht anders angegeben, werden in diesem Handbuch die folgenden Notationen verwendet:**

- Hexadezimale Werte werden mit dem Buchstaben “h” am Ende dargestellt, z. B. 12h, wohingegen dezimale Werte nicht explizit markiert sind.
- Die Datenübertragung erfolgt im „Little Endian“-Format, d. h. das niederwertigste Byte eines Datenwortes wird zuerst übertragen. Dies bedeutet, dass z. B. für einen Ganzzahlwert 1234h, der durch zwei aufeinanderfolgende Bytes repräsentiert wird, zunächst das niederwertige Byte 34h und dann das höherwertige Byte 12h übertragen wird.
- Gleitkommawerte werden als 32-Bit-Gleitkommawerte einfacher Präzision nach IEEE 754 codiert. Hierbei wird für die Bitfolge der Werte folgendes Schema angewandt:
D31: Vorzeichen
D30...23: Exponent
D22...0: Mantisse
- Jede Aufzählung von Werten beginnt mit dem Index 0, d. h. ein Array von n Elementen hat einen Indexbereich von 0 bis n-1.

 **Die folgenden Datentypen werden im Befehlssatz verwendet:**

- Integer: Ganzzahlwert der Länge 8, 16 oder 32 Bit
- Float: Gleitkommawert der Länge 32 Bit nach IEEE 754
- String: ASCII-codierter Text, der keinerlei Sonderzeichen enthalten darf
- Bitvektor: Vektor aus einzelnen Bits, wobei jedes Bit eine spezielle Bedeutung hat
- Enum: Aufzählungstyp. Vergleichbar mit Integer, wobei jedem Wert eine spezielle Bedeutung zugeordnet ist.

1.1 Aufbau einer Verbindung zum Greifmodul

Die Greifmodule der WSG-Serie verfügen über verschiedene Schnittstellen, die jeweils über das in diesem Handbuch beschriebene, binäre Befehlsprotokoll angesprochen werden können. Je nach Typ

sind als Schnittstellen RS-232, Ethernet TCP/IP, Ethernet UDP/IP und/oder CAN-Bus verfügbar. Die Feldbusschnittstellen für Profibus, Profinet und Modbus/TCP unterscheiden sich grundlegend von diesen Schnittstellen und werden im Handbuch „WSG Feldbusschnittstelle“ separat behandelt.

Auswahl und Konfiguration der Befehlsschnittstelle erfolgen über die webbasierte Konfigurationsoberfläche des Greifmoduls. Hierzu muss das Modul mit einem lokalen Firmennetzwerk oder direkt mit der Netzwerkschnittstelle eines Computers oder Laptops verbunden werden. Weiterführende Informationen befinden sich in der Montage- und Bedienungsanleitung des Greifmoduls.


Ist das Greifmodul verbunden, kann mit Hilfe eines gängigen Web-Browsers z.B. durch Eingabe der vorkonfigurierten IP-Adresse <http://192.168.1.20> die Web-Oberfläche des Moduls aufgerufen werden. Sobald die Oberfläche erfolgreich geladen wurde, können im Menüpunkt *“Settings -> Command Interface”* Einstellungen zur Befehlsschnittstelle vorgenommen werden.

1.2 Kommunikation mit dem Greifmodul

Unabhängig von der gewählten Schnittstelle kommunizieren die Greifmodule mit der angeschlossenen Steuerung über binär codierte Nachrichten (Ausnahme: Feldbusschnittstellen, vgl. Handbuch „WSG Feldbusschnittstelle“). Die folgenden Abschnitte beschreiben das allgemeine Format dieser Nachrichten.


1.2.1 Befehlsformat

Die folgende Tabelle beschreibt das grundlegende Befehlsformat. Alle Nachrichten starten mit einer Präambel, die den Anfang der neuen Nachricht signalisiert. Ein Identifikationscode beschreibt den Inhalt der Nachricht. Er wird als Befehls-ID verwendet und unterscheidet zwischen den verschiedenen Befehlen des Moduls. Der aus zwei Bytes bestehende Längenwert legt die Anzahl an Bytes der anschließend übertragenen Nutzdaten fest. Zum Schluss folgt eine CRC-Prüfsumme, um die Integrität der Daten zu gewährleisten.

 **Für die Berechnung der CRC-Prüfsumme befindet sich ein Codebeispiel in Kapitel Anhang D.**

Sollte die Berechnung der Prüfsumme Probleme verursachen oder aus anderen Gründen nicht erwünscht sein, kann sie optional über die Web-Oberfläche des Greifmoduls deaktiviert werden (vgl. Kapitel 1.4.1).

Um die Integrität einer vom Greifmodul empfangenen Antwortnachricht zu überprüfen, muss die CRC-Prüfsumme nach dem selben Verfahren über die gesamte Nachricht inklusive Präambel und angehängter Prüfsumme neu berechnet werden. Sind die Daten unverfälscht, so ist die berechnete Prüfsumme 0.

 **Für erste Schritte mit dem Befehlsprotokoll wird die Verwendung des „Custom Command Editor“ der mitgelieferten Software „WSG Commander“ empfohlen (vgl. Kapitel 2). Dieser ermöglicht das interaktive Erstellen und Versenden von Nachrichten an das Greifmodul bei gleichzeitiger Darstellung der übertragenen Bytesequenz.**


Byte	Symbol	Beschreibung
0..2	Präambel	Signalisiert den Anfang einer neuen Nachricht durch die fest stehende Bytesequenz AAAAAAh
3	Befehls-ID	Repräsentiert die ID des auszuführenden Befehls.
4..5	Länge der Nutzdaten	Länge der in der Nachricht enthaltenen Nutzdaten in Bytes. Kann für manche Pakete den Wert 0 annehmen.
6..n	Nutzdaten	Nutzdaten (optional)
n+1..n+2	Prüfsumme	<p>CRC-Prüfsumme über die gesamte Nachricht, inklusive Präambel. Vgl. Kapitel Anhang D für ein Beispiel zur Berechnung der Prüfsumme.</p> <p> Hinweis: Sofern die Prüfsummenberechnung deaktiviert wurde (vgl. Kapitel 1.4.1), wird die Prüfsumme vom Greifmodul nicht ausgewertet und der Wert der Prüfsumme kann beliebig gewählt werden. Es muss jedoch immer eine Prüfsumme mit gesendet werden. Nachrichten mit fehlender Prüfsumme werden vom Greifmodul als ungültig verworfen.</p>

Table 1: Struktur der Datenpakete

Beispiel 1: Nachricht mit ID = 1, keine Nutzdaten:

AAh AAh AAh 01h 00h 00h E8h 10h



Beispiel 2: Nachricht mit ID = 2, zwei Bytes Nutzdaten (12h, 34h), Prüfsumme 666Dh:

AAh AAh AAh 01h 02h 00h 12h 34h 6Dh 66h

1.2.2 Antwortnachrichten vom Greifmodul

Jeder Befehl, der vom Greifmodul empfangen wurde, wird mit einer standardisierten Nachricht beantwortet, die den empfangenen Befehl quittiert und eventuelle Rückgabewerte zurück liefert. Diese Antwortnachrichten entsprechen dem folgenden Format:

Byte	Symbol	Beschreibung
0..2	Präambel	Signalisiert den Anfang einer neuen Nachricht durch die fest stehende Bytesequenz AAAAAAh

3	Befehls-ID	Repräsentiert die ID des zu beantwortenden Befehls.
4..5	Länge der Nutzdaten	Länge der in der Nachricht enthaltenen Nutzdaten in Bytes. Diese beträgt $n - 4$, d. h. beispielsweise 2 für eine Nachricht nur mit Statuscode oder 6 für eine Nachricht, die neben dem Statuscode noch einen befehlspezifischen 4-Byte-Nutzdatenwert zurück liefert.
6..7	Statuscode	Statuscode, vgl. Kapitel Anhang A
8..n	Nutzdaten	Befehlsspezifische Nutzdaten (z. B. Rückgabewerte, optional). Nur verfügbar, wenn der Wert des Statuscodes E_SUCCESS ist.
n+1..n+2	Prüfsumme	CRC-Prüfsumme über die gesamte Nachricht inklusive Präambel. Vgl. Kapitel Anhang D für Informationen zur Berechnung.  Hinweis: Auch bei deaktivierter Prüfsummenberechnung (vgl. Kapitel 1.4.1) wird den Antwortnachrichten, die vom Greifmodul gesendet werden, immer eine korrekte Prüfsumme angehängt. Die Auswertung obliegt dem Client (z.B. Anlagensteuerung).  Die Berechnung der CRC-Prüfsumme über die gesamte Nachricht inklusive Prüfsumme muss den Wert 0 ergeben.

Beispiel 1: Quittierung eines erfolgreich ausgeführten Befehls (Statuscode E_SUCCESS bzw. 0000h) ohne weitere Rückgabewerte (hier: „Homing“-Befehl):

AAh AAh AAh 20h 02h 00h 00h 00h B3h FDh

Beispiel 2: Quittierung eines fehlerhaften Befehls (Befehls-ID 90h ist unbekannt, weshalb das Greifmodul E_CMD_UNKNOWN bzw. 000Eh als Statuscode zurück liefert):

AAh AAh AAh 90h 02h 00h 0Eh 00h FDh 02h

Beispiel 3: Quittierung eines erfolgreich ausgeführten „Beschleunigung auslesen“-Befehls, der einen 4-Byte-Gleitkommawert zurück liefert (hier: 150.0 mm/s² oder 00h 00h 16h 43h):

AAh AAh AAh 30h 06h 00h 00h 00h 00h 16h 43h DCh CBh

1.3 Asynchrone Befehle

Für den Fall, dass das Ergebnis eines Befehls nicht sofort verfügbar ist (z. B. bei Bewegungsbefehlen oder Referenzierungsfahrt), liefert das Greifmodul einen Statuscode zurück, der ausdrückt dass die

Nachricht zwar korrekt empfangen und die Ausführung des Befehls gestartet wurde, ein Ergebnis jedoch erst mit zeitlicher Verzögerung in einer weiteren Nachricht geliefert werden kann.

Die unmittelbare Antwort eines solchen Befehls enthält den Statuscode E_CMD_PENDING, um den Empfang des Befehls zu quittieren. Sobald die Ausführung des Befehls beendet wurde, folgt eine zweite Antwortnachricht, die das Ergebnis des Befehls enthält.

Beispiel: Quittierung des Empfangs eines Vorpositionierbefehls (21h, vgl. Kapitel 3.2.2) mit Statuscode E_CMD_PENDING (001Ah):

AAh AAh AAh 21h 02h 00h 1Ah 00h 67h CBh


Nachdem die Zielposition erreicht wurde, sendet das Greifmodul eine weitere Antwortnachricht mit dem Ergebnis (hier: Statuscode E_SUCCESS bzw. 0000h):

AAh AAh AAh 21h 02h 00h 00h 00h 28h 04h

1.4 Schnittstellenoptionen

Abhängig von der Art der Schnittstelle stehen neben der Basiskonfiguration einige zusätzliche Einstellungen zur Verfügung, die das Verhalten des Greifmoduls beeinflussen. Die folgenden Abschnitte geben einen Überblick über diese Einstellungen.

1.4.1 Deaktivieren der CRC-Prüfsumme

 **Diese Einstellung ist für fortgeschrittene Anwender gedacht. Es wird empfohlen, die Auswertung der Prüfsumme aktiviert zu lassen, um die Integrität der ausgetauschten Nachrichten auf allen Schnittstellen zu gewährleisten.**

Wie in den vorhergehenden Abschnitten beschrieben, wird jeder Nachricht eine CRC-Prüfsumme angehängt um die Integrität der Daten beim Austausch von Nachrichten über das binäre Befehlsprotokoll zu gewährleisten.

In manchen Fällen kann es jedoch von Nutzen sein, die Auswertung der Prüfsumme zu deaktivieren, z. B. zu Testzwecken während Integration des Greifmoduls oder auch dauerhaft, wenn die übergeordnete Steuerung eine Berechnung der Prüfsumme nicht zulässt und/oder die Integrität der Daten durch unterliegende Übertragungsprotokolle der gewählten Schnittstelle gewährleistet ist (z. B. bei Verwendung von TCP-Netzwerkverbindungen).

Die Deaktivierung der CRC-Prüfsumme erfolgt über die Web-Oberfläche des Greifmoduls im Menüpunkt „Settings -> Command Interface“ durch Entfernen des Häkchens bei „Disable CRC Checksum“. Wenn die Auswertung der CRC-Prüfsumme deaktiviert ist, wird das Greifmodul den Wert der Prüfsummen-Bytes in allen empfangenen Nachrichten ignorieren. Dennoch müssen die entsprechenden Bytes an jede Nachricht angehängt werden. Sie dürfen jedoch beliebige Werte annehmen.

Es wird jedoch nur die Auswertung der Prüfsumme für auf dem Greifmodul eingehende Nachrichten deaktiviert. Vom Greifmodul gesendete Antwortnachrichten werden in jedem Fall mit einer korrekten Prüfsumme versehen, unabhängig davon, welche Einstellung gewählt ist. Ob und inwieweit die Auswertung der Prüfsumme in einer übergeordneten Anlagensteuerung möglich bzw. notwendig ist, liegt im Ermessen des Anwenders. Es wird empfohlen, die Auswertung der Prüfsumme für empfangene Nachrichten sowohl am Greifmodul zu aktivieren (Standardeinstellung) als auch auf der übergeordneten Anlagensteuerung zu implementieren.

1.4.2 Deaktivieren des Fehlers beim unangekündigten Beenden einer TCP-Verbindung

 **Diese Einstellung ist für fortgeschrittene Anwender gedacht. Abgebrochene Verbindungen können auf eine fehlerhafte Kommunikation mit dem Greifmodul hindeuten. Aus diesem Grund sollte die im Folgenden beschriebene Einstellung nur im äußersten Ausnahmefall deaktiviert werden.**

Bei der Verwendung verbindungsorientierter Übertragungsprotokolle wie TCP müssen Verbindungsabbrüche und unangekündigtes, einseitiges Beenden der Verbindung als Fehler betrachtet werden. Aus diesem Grund wird auf dem Greifmodul ein FAST STOP ausgelöst, sobald eine TCP-Verbindung ohne vorherige Ankündigungsnachricht (vgl. Kapitel 3.1.2) beendet wird.

In manchen Fällen kann es notwendig sein, diese Funktion zu deaktivieren. Die entsprechende Einstellung kann über die Web-Oberfläche im Menüpunkt „*Settings -> Command Interface*“ vorgenommen werden. Sofern TCP als Schnittstelle ausgewählt wurde, kann das entsprechende Häkchen entfernt werden.

2 WSG-Commander

△ Die Software „WSG Commander“ ist als Werkzeug zur einfachen Evaluierung und Inbetriebnahme des Greifmoduls kostenlos im Lieferumfang enthalten. Sie dient ausschließlich zu Testzwecken und sollte nicht in produktiven Umgebungen eingesetzt werden!

Die Software „WSG Commander“ für Microsoft Windows® ist im Lieferumfang des Greifmoduls enthalten und soll dazu dienen, sich mit dem Protokoll und dem Befehlssatz vertraut zu machen. Das Programm ermöglicht das Senden einfacher Befehle an das Greifmodul sowie die Auswertung der übertragenen Daten über das sog. „Command Log“. Darüber hinaus ist mit dem „Custom Command Editor“ ein Werkzeug integriert, mit dem eigene Nachrichten einfach byteweise zusammengesetzt und an das Greifmodul gesendet werden können.

Das Programm kann entweder von der beiliegenden Produkt-CD installiert oder direkt von der Web-Oberfläche des Greifmoduls aus dem Menüpunkt „Help -> Documentation“ herunter geladen werden.



Figure 1: WSG Commander

Hauptfenster

Um eine Verbindung zum Greifmodul aufzubauen, wählen Sie im Hauptfenster den Menüpunkt *Gripper | Connect*. Im anschließenden Dialog können die Schnittstelle ausgewählt sowie eventuell notwendige Verbindungseinstellungen (Adresse, Datenrate etc.) vorgenommen werden. Bitte beachten Sie, dass das Greifmodul eventuell erst über die Web-Oberfläche für die entsprechenden Verbindungseinstellungen konfiguriert werden muss.

Der WSG Commander unterstützt Verbindungen über die folgenden Schnittstellen: RS-232, CAN-Bus via ESD-Karten, Ethernet TCP/IP und UDP/IP.

Custom Command Editor

Neben der vordefinierten Steuerung über das Hauptfenster können im „Custom Command Editor“ auch eigene Befehle byteweise zusammengesetzt und gesendet werden. Zum Öffnen des Command Editors wählen Sie im Hauptfenster den Menüpunkt *Commands | Command Editor*.

Im Feld „Command ID“ kann die ID des zu sendenden Befehls entweder aus dem Auswahlnenü gewählt oder direkt von Hand eingegeben werden. Es werden sowohl dezimale als auch hexadezimale Werte mit vorangestelltem „0x“ (z.B. „0x20“) akzeptiert. Ebenso können die einzelnen Datenbytes für die Nutzdaten im Feld „Payload“ entweder als dezimale oder als hexadezimale Werte eingegeben werden. Auch können Gleitkommawerte mit einem „f“ am Ende (z.B. „150.0f“) oder Text in Anführungszeichen eingegeben werden. Die eingegebenen Daten werden in die hellblau hinterlegte Bytesequenz („Command Packet Decoding“) konvertiert, wobei die jeweils passende CRC-Prüfsumme automatisch generiert wird. Mit einem Klick auf den Button „Send“ wird der Befehl an das Greifmodul gesendet.

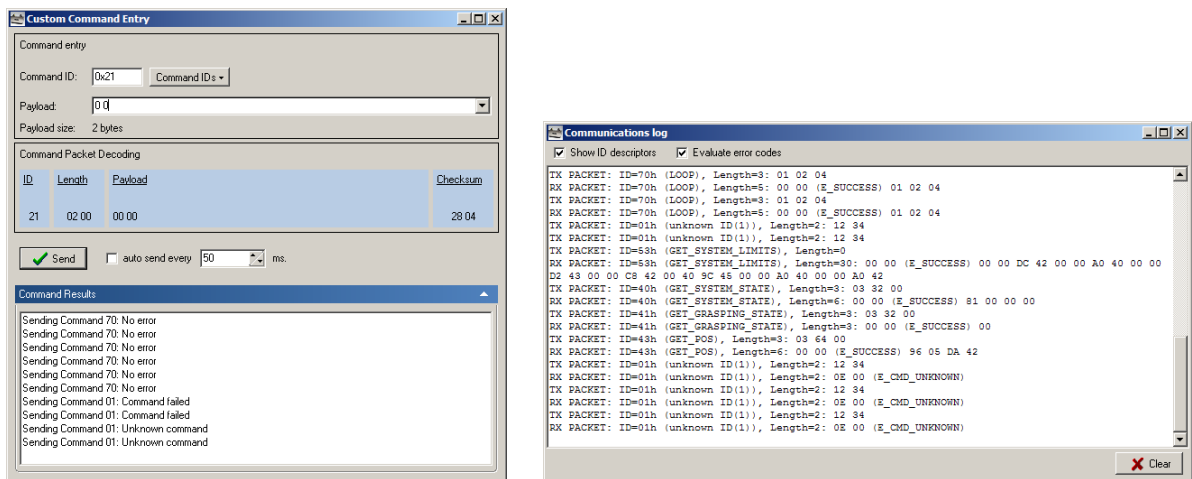


Figure 2: Custom Command Editor (links) und Communication Log (rechts)

Communication Log

Um die Kommunikation zwischen Greifmodul und WSG Commander zu verfolgen, kann das „Communication Log“ verwendet werden. Es kann vom Hauptfenster aus über den Menüpunkt *View | Command log* geöffnet werden. Über entsprechende Einstellungen ist es möglich, Befehls-IDs und Statuscodes automatisch decodieren zu lassen.

Wenn ein oder mehrere Bytes im Communication Log markiert werden, erscheint ein Popup-Menü und die markierten Bytes können in einen Gleitkomma- oder Ganzzahlwert bzw. ASCII-Text konvertiert werden.

3 Befehlsreferenz

Im Folgenden werden die einzelnen Befehle des Greifmoduls im Detail beschrieben.

3.1 Verbindungsverwaltung

3.1.1 Loop (06h)

„Loop-back“-Befehl, der eine Kopie der gesendeten Befehlsparameter zurück sendet. Dieser Befehl dient zum Testen der Befehlsschnittstelle.

Befehls-ID: 06h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0..n	LOOPDATA	Integer	Beliebige Nutzdaten <i>Bis zu 256 Bytes an beliebigen Nutzdaten können gesendet werden.</i>

Rückgabewerte:

Die als LOOPDATA gesendete Bytefolge wird identisch in den Nutzdaten der Antwortnachricht zurück geliefert. Es ist zu beachten, dass den Nutzdaten in der Antwortnachricht noch zwei Bytes für den Statuscode vorangestellt werden (vgl. Beschreibung des Formats der Antwortnachrichten in Kapitel 1.2.2).

Mögliche Statuscodes:


E_CMD_FORMAT_ERROR: Nachricht zu lang (mehr als 256 Bytes an Nutzdaten).

E_INVALID_PARAMETER: Fehlerhafter Parameter.

E_CMD_PENDING: Kein Fehler, Befehl wartet auf Abarbeitung.

3.1.2 Ankündigung der Verbindungstrennung (07h)

Signalisiert dem Greifmodul, dass die Verbindung in Kürze beendet wird. Dieser Befehl ist nur bei Verwendung von Ethernet TCP/IP-Verbindungen erforderlich und bewirkt, dass nach dem anschließenden Abbau der Verbindung kein FAST STOP-Fehlerzustand ausgelöst wird. Beim Empfang dieses Befehls werden alle laufenden Fingerbewegungen des Greifmoduls unmittelbar gestoppt.

 **Wird dem Greifmodul die Trennung der Verbindung angekündigt, wartet es auf das Ende der Verbindung und akzeptiert keine weiteren Befehle mehr. Alle danach eintreffenden Nachrichten werden mit dem Statuscode E_ACCESS_DENIED beantwortet. Erst nach regulärem Abbau und Neuaufbau der Verbindung können neue Befehle gesendet werden.**

Befehls-ID: 07h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_NO_PARAM_EXPECTED: Der Befehl erwartet keine Parameter aber mindestens ein Parameter wurde gesendet.


E_NOT_AVAILABLE: Befehl wurde bei einer nicht-verbindungsorientierten Schnittstelle verwendet.

3.2 Bewegungssteuerung


3.2.1 Referenzierungsfahrt (20h)

Führt eine Referenzierungsfahrt („Homing“) durch, um die Position der Greiffinger zu referenzieren. Dieser Befehl muss nach dem Starten des Greifmoduls vor allen anderen Bewegungsbefehlen durchgeführt werden. Die Richtung der Referenzierungsfahrt (Referenzierung außen oder innen) kann entweder explizit vorgegeben werden oder nach Voreinstellung über die Konfiguration auf der Web-Oberfläche.

Während der Referenzierungsfahrt bewegen sich die Greiffinger in der vorgegebenen Richtung bis an ihren mechanischen Endanschlag. Die Position, an der die Fingerbewegung blockiert wird, wird bei allen folgenden Bewegungsbefehlen als Ursprung für die Bestimmung der Fingerposition verwendet.

 **Die höchste Positioniergenauigkeit wird erreicht, wenn die Referenzierungsfahrt in der Richtung durchgeführt wird, in die später auch gegriffen werden soll.**

 **Während der Referenzierungsfahrt werden die eingestellten Soft Limits ignoriert!**

 **Hindernisse, die die freie Bewegung der Greiffinger während des Referenzierens blockieren, können dazu führen, dass der Ursprung für die Positionsberechnung nicht korrekt gesetzt werden kann und das Greifmodul in der Folge fehlerhafte Fingerbewegungen durchführt!**

Befehls-ID: 20h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0	DIRECTION	Enum	Richtung der Referenzierungsfahrt <i>0: Verwende voreingestellte Richtung aus der Systemkonfiguration (kann über die Web-Oberfläche angepasst werden)</i> <i>1: Referenzierung in positiver Bewegungsrichtung (außen)</i> <i>2: Referenzierung in negativer Bewegungsrichtung (innen)</i>

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

Unmittelbare Statuscodes:

E_ACCESS_DENIED: Greifmodul ist im FAST STOP-Zustand.

E_ALREADY_RUNNING: Greifmodul führt aktuell bereits einen Bewegungsbefehl aus (kann durch STOP-Befehl abgebrochen werden).

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

E_INVALID_PARAMETER: Ungültiger Befehlsparameter.

E_CMD_PENDING: Kein Fehler, Befehl wird ausgeführt.

Statuscodes nach der Durchführung des Befehls:

E_SUCCESS: Befehl erfolgreich.

E_CMD_ABORTED: Referenzierungsfahrt abgebrochen.

E_AXIS_BLOCKED: Achse wurde in Gegenrichtung blockiert.

E_TIMEOUT: Befehl konnte nicht innerhalb des vorgegebenen Zeitfensters durchgeführt werden.


3.2.2 Vorpositionieren der Finger (21h)


Bewegt die Greiffinger zu einer vorgegebenen Öffnungsposition. Dieser Befehl ist dafür vorgesehen, die Finger vor einem Greifbefehl an eine bestimmte Stelle vorzupositionieren. Um ein Teil zu greifen oder loszulassen, müssen im Unterschied dazu die Befehle *Teil greifen* (25h, siehe Kapitel 3.2.6) und *Teil loslassen* (26h, siehe Kapitel 3.2.7) verwendet werden.

Es kann ausgewählt werden, ob die Bewegung der Finger relativ zur aktuellen Fingerposition oder an eine absolut vorgegebene Fingerposition erfolgen soll. Der Befehl wird asynchron ausgeführt, d. h. das Greifmodul sendet bei korrektem Empfang und Verarbeitung der Nachricht unmittelbar eine Antwortnachricht mit dem Status E_CMD_PENDING. Wenn die Zielposition wie vorgesehen erreicht oder der Bewegungsbefehl durch einen Fehler (z. B. blockierte Achse) abgebrochen wurde, folgt eine weitere Quittierungsnachricht, die E_SUCCESS oder einen anderen, zum aufgetretenen Fehler passenden Statuscode zurück liefert (vgl. Beschreibung asynchroner Befehle in Kapitel 1.3).


Vorgegebene Geschwindigkeits- und Positionswerte, die sich außerhalb der physischen Grenzen des Greifmoduls befinden, werden auf den höchst- bzw. niedrigstmöglichen Wert korrigiert. Dies umfasst nicht nur die absoluten Endwerte, sondern ggf. auch Werte innerhalb der Grenzen, die mit den aktuellen Voreinstellungen nicht erreicht werden können (z. B. aufgrund zu niedrig eingestellter Beschleunigung).

Es bietet sich an, die physischen Grenzen des Greifmoduls zuerst auszulesen (Befehl *Systemgrenzen auslesen*, vgl. Kapitel 3.5.4) und die Bewegungsparameter vor dem Absenden entsprechend zu prüfen, um sicher zu stellen dass die Bewegung wie gewünscht durchgeführt werden kann.

 Um die aktuelle Position der Greiffinger zu bestimmen kann der Befehl **Aktuelle Öffnungsweite auslesen** (43h, siehe Kapitel 3.4.4) verwendet werden.

 Der Befehl **Vorpositionieren** schätzt die Kontaktkraft beim Blockieren der Achse immer über den Motorstrom ab („Force Approximation Mode“), unabhängig davon ob Kraftmessfinger installiert sind oder nicht.


 Um Teile zu greifen oder loszulassen müssen die Befehle **Teil greifen** (25h, vgl. Kapitel 3.2.6) bzw. **Teil loslassen** (26h, vgl. Kapitel 3.2.7) verwendet werden.

 Das Greifmodul muss referenziert sein und darf sich nicht im Zustand **FAST STOP** befinden, um einen Bewegungsbefehl starten zu können.

Befehls-ID: 21h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0	FLAGS	Bitvektor	D7..D2: nicht verwendet, auf 0 zu setzen D1: Stop bei Blockierung <i>1: Stop bei Blockierung</i>

			<p>Wenn eine Blockierung in Bewegungsrichtung der Finger festgestellt wird, liefert der Befehl den Statuscode E_AXIS_BLOCKED zurück und der Antrieb wird gestoppt.</p> <p><i>0: Klemmen bei Blockierung</i></p> <p>Wenn eine Blockierung in Bewegungsrichtung der Finger festgestellt wird, liefert der Befehl den Statuscode E_AXIS_BLOCKED zurück. Der Antrieb wird aber nicht gestoppt, statt dessen wird mit begrenzter Kraft (50% der Nenngreifkraft) weiter versucht, die Endposition zu erreichen.</p> <p> Sobald die Blockierung gelöst wird, schnappen die Greiffinger an die vorgegebene Endposition!</p> <p>D0: Bewegungsart</p> <p><i>1: Relative Bewegung</i></p> <p>Die vorgegebene Greifweite wird als Distanz relativ zur aktuellen Position der Greiffinger behandelt.</p> <p><i>0: Absolute Bewegung</i></p> <p>Die vorgegebene Greifweite wird als absolute Zielposition behandelt (geschlossene Greiffinger bei einer Greifweite von 0 mm).</p>
1..4	WIDTH	Float	Öffnungsweite in mm
5..8	SPEED	Float	Verfahrgeschwindigkeit in mm/s

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

Unmittelbare Statuscodes:

E_ACCESS_DENIED: Greifmodul ist im Fehlerzustand FAST STOP.

E_NOT_INITIALIZED: Greifmodul ist nicht referenziert. Es muss erst eine Referenzierungsfahrt durchgeführt werden.

E_ALREADY_RUNNING: Greifmodul führt aktuell bereits einen Bewegungsbefehl aus (kann durch STOP-Befehl abgebrochen werden).

E_RANGE_ERROR: Soft Limits sind aktiviert und die vorgegebene Zielposition liegt außerhalb des durch Soft Limits definierten Bereichs.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

E_CMD_PENDING: Kein Fehler, Befehl wird ausgeführt.

Statuscodes nach der Durchführung des Befehls:

E_SUCCESS: Befehl erfolgreich.

E_INSUFFICIENT_RESOURCES: Speicherüberlauf

E_AXIS_BLOCKED: Achse ist blockiert.


E_RANGE_ERROR: Eine Grenze (Soft Limit oder mechanische Anschlagposition) wurde während des Verfahrens erreicht. Das Greifmodul wurde gestoppt.

E_TIMEOUT: Befehl konnte nicht innerhalb des vorgegebenen Zeitfensters durchgeführt werden

E_CMD_ABORTED: Der Bewegungsbehl wurde abgebrochen, z. B. durch einen Stop-Befehl.

3.2.3 Stop (22h)

Stoppt unmittelbar jede laufende Fingerbewegung. Der Befehl setzt das Flag SF_AXIS_STOPPED. Der Zustand muss nicht quittiert werden, er wird automatisch beim nächsten empfangenen Bewegungsbefehl aufgehoben.

 Um das Greifmodul im Fall eines Fehlers zu stoppen sollte statt dessen der Befehl Fast Stop (23h, vgl. Kapitel 3.2.4) verwendet werden.

Befehls-ID: 22h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_NO_PARAM_EXPECTED: Der Befehl erwartet keine Parameter aber mindestens ein Parameter wurde gesendet.

E_TIMEOUT: Befehl konnte nicht innerhalb des vorgegebenen Zeitfensters durchgeführt werden.

3.2.4 Auslösen eines Fast Stop (23h)

Dieser Befehl ähnelt einem „Notstopp“ und stoppt unmittelbar jede laufende Fingerbewegung und verhindert die Ausführung weiterer Bewegungsbefehle. Der FAST STOP-Zustand kann nur mit Hilfe des Befehls *Quittieren eines Fast Stops* (24h, vgl. Kapitel 3.2.5) wieder aufgehoben werden.

Alle während des aktiven FAST STOP-Zustands gesendeten Bewegungsbefehle liefern als Ergebnis den Statuscode E_ACCESS_DENIED.

Der aktive FAST STOP-Zustand wird über die Systemzustandsflags signalisiert und im Log des Greifmoduls vermerkt. Allgemein sollte der Befehl zur Reaktion auf unvorhergesehene Fehler verwendet werden.

 **Um die aktive Bewegung der Greiffinger ohne das Auslösen eines Fehlerzustands zu stoppen, kann der Befehl Stop (vgl. Kapitel 3.2.3) verwendet werden.**

Befehls-ID: 23h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

3.2.5 Quittieren eines Fast Stop oder einer Fehlerbedingung (24h)

Quittiert den FAST STOP-Zustand nach dem Beheben der entsprechenden Fehlerursache. Anschließend ist das Greifmodul wieder im normalen Betriebszustand und Bewegungsbefehle werden wieder akzeptiert.

Befehls-ID: 24h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0..2	ACK_KEY	String	ASCII-Textfolge „ack“, entsprechend der Bytesequenz 61h 63h 6Bh)

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Falscher Parameter ACK_KEY.

3.2.6 Teil greifen (25h)

Greifen eines Teils mit vorgegebener nominaler Größe und Greifgeschwindigkeit. Der Befehl bewegt die Greiffinger mit der vorgegebenen Greifgeschwindigkeit an die Position, die durch die nominale Teilgröße festgelegt wird und versucht, das Teil mit der voreingestellten Kraft zu greifen. Falls die Kraft innerhalb einer vordefinierten Strecke („Clamping Travel“), gemessen ab der nominalen Teilgröße, aufgebracht werden kann, wird das Teil als korrekt gegriffen erkannt (Greifzustand HOLDING). Falls sich die Greiffinger über diese Strecke hinaus bewegen, ohne die voreingestellte Greifkraft aufbringen zu können, wird kein Teil erkannt (Greifzustand NO PART).

Die Strecke, um welche sich die Greiffinger über die nominale Greifposition hinaus bewegen dürfen, um die Greifkraft aufzubringen, wird als „Clamping Travel“ bezeichnet und kann auf der Web-Oberfläche festgelegt werden. Sofern kein Teil gefunden wird, liefert der Befehl den Statuscode E_CMD_FAILED zurück.

Für empfindliche Greifteile sollte die Greifgeschwindigkeit entsprechend reduziert werden, um den durch die Masse der Greiffinger entstehenden Aufschlagimpuls beim Auftreffen der Greiffinger auf das Greifteil zu begrenzen.

Der Greifzustand („Gripper State“) reflektiert den Zustand des laufenden Greifprozesses. Er kann mittels des Befehls *Greifzustand auslesen* (41h, vgl. Kapitel 3.4.2) abgefragt werden.

Es ist zu beachten, dass es nicht möglich ist, auf einen Greifbefehl hin einen weiteren Greifbefehl zu senden, der die Greiffinger in die entgegengesetzte Richtung bewegt. Allgemein sollte auf einen Greifbefehl hin immer ein Befehl *Teil loslassen* (42h, vgl. Kapitel 3.2.7) gesendet werden, bevor der nächste Greifbefehl gesendet wird. Es ist jedoch möglich, nach einem Greifbefehl noch einmal in die gleiche Richtung „nachzugreifen“, wenn z. B. die vorgegebene nominale Teilgröße zu groß gewählt war.

Befehls-ID: 25h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0..3	WIDTH	Float	Nominale Größe des zu greifenden Teils in mm.
4..7	SPEED	Float	Greifgeschwindigkeit in mm/s

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

Unmittelbare Statuscodes:

E_ACCESS_DENIED: Greifmodul ist im Zustand FAST STOP.

E_ALREADY_RUNNING: Greifmodul führt aktuell bereits einen Bewegungsbefehl aus (kann durch STOP-Befehl abgebrochen werden).

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

E_RANGE_ERROR: Parameter WIDTH verletzt die voreingestellten Soft Limits.

E_CMD_PENDING: Kein Fehler, Befehl wird ausgeführt.

Statuscodes nach der Durchführung des Befehls:

E_SUCCESS: Befehl erfolgreich.

E_CMD_ABORTED: Greifen abgebrochen.

E_CMD_FAILED: Kein Teil gefunden.

E_TIMEOUT: Befehl konnte nicht innerhalb des vorgegebenen Zeitfensters durchgeführt werden.

3.2.7 Teil loslassen (26h)

Öffnen der Greiffinger und Loslassen des gegriffenen Teils.

Befehls-ID: 26h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0..3	OPENWIDTH	Float	Öffnungsweite in in mm (absolute Zielposition).
4..7	SPEED	Float	Öffnungsgeschwindigkeit in mm/s

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

Unmittelbare Statuscodes:

E_ACCESS_DENIED: Greifmodul ist im Zustand FAST STOP.

E_ALREADY_RUNNING: Greifmodul führt aktuell bereits einen Bewegungsbefehl aus (kann durch STOP-Befehl abgebrochen werden).

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

E_RANGE_ERROR: Parameter OPENWIDTH verletzt die voreingestellten Soft Limits.

E_CMD_PENDING: Kein Fehler, Befehl wird ausgeführt.

Statuscodes nach der Durchführung des Befehls:

E_SUCCESS: Befehl erfolgreich.


E_CMD_ABORTED: Loslassen abgebrochen.

E_TIMEOUT: konnte nicht innerhalb des vorgegebenen Zeitfensters durchgeführt werden.

3.3 Bewegungskonfiguration

3.3.1 Setzen der Beschleunigung (30h)

Legt den Beschleunigungswert fest, mit dem Bewegungsbefehle ausgeführt werden.

-  Beim Starten des Greifmoduls wird ein voreingestellter Standardwert für die Beschleunigung verwendet. Dieser Standardwert kann über die Web-Oberfläche des Greifmoduls angepasst werden. Der Beschleunigungswert, der mit diesem Befehl gesetzt wird, ist nur für die aktuelle Sitzung gültig und geht bei Neustart des Moduls verloren.

Befehls-ID: 30h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0..3	ACC	Float	Beschleunigung in mm/s ² . Der Wert wird auf das zulässige Minimum bzw. Maximum des Greifmoduls begrenzt, wenn außerhalb der Systemgrenzen.

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Länge der übermittelten Nutzdaten ist nicht korrekt.

3.3.2 Auslesen der Beschleunigung (31h)

Liefert den aktuell eingestellten Beschleunigungswert zurück.

Befehls-ID: 31h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0..3	ACC	Float	Beschleunigung in mm/s ²



Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_NO_PARAM_EXPECTED: Der Befehl erwartet keine Parameter aber mindestens ein Parameter wurde gesendet.

3.3.3 Setzen der Greifkraft (32h)

Legt die Greifkraft fest, die für die folgenden Greifbefehle (*Teil greifen*, 25h, siehe Kapitel 3.2.6) verwendet werden soll. Die Greifkraft legt die maximale Kraft fest, die bei mechanischem Kontakt mit dem Greifteil aufgebracht werden soll.

-  **Beim Starten des Greifmoduls wird ein voreingestellter Standardwert für die Greifkraft verwendet. Dieser Standardwert kann über die Web-Oberfläche des Greifmoduls angepasst werden. Der Kraftwert, der mit diesem Befehl gesetzt wird, ist nur für die aktuelle Sitzung gültig und geht bei Neustart des Moduls verloren.**
-  **Hinweis: Die Greifkraft ist definiert als das Produkt aus der effektiven Greifkraft mal der Anzahl der Greiffinger.**

Befehls-ID: 32h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0..3	FORCE	Float	Greifkraft in Newton. Der Wert wird auf das zulässige Minimum bzw. Maximum des Greifmoduls begrenzt, wenn außerhalb der Systemgrenzen.

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Länge der übermittelten Nutzdaten ist nicht korrekt.

3.3.4 Auslesen der Greifkraft (33h)

Liefert die aktuell eingestellte Kraftbegrenzung zurück, die zuvor mittels des Befehls *Setzen der Kraftbegrenzung* (32h, vgl. Kapitel 3.3.3) eingestellt wurde.

 **Hinweis: Die Greifkraft ist definiert als das Produkt aus der effektiven Greifkraft mal der Anzahl der Greiffinger.**

Befehls-ID: 33h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0..3	ACC	Float	Greifkraft in Newton

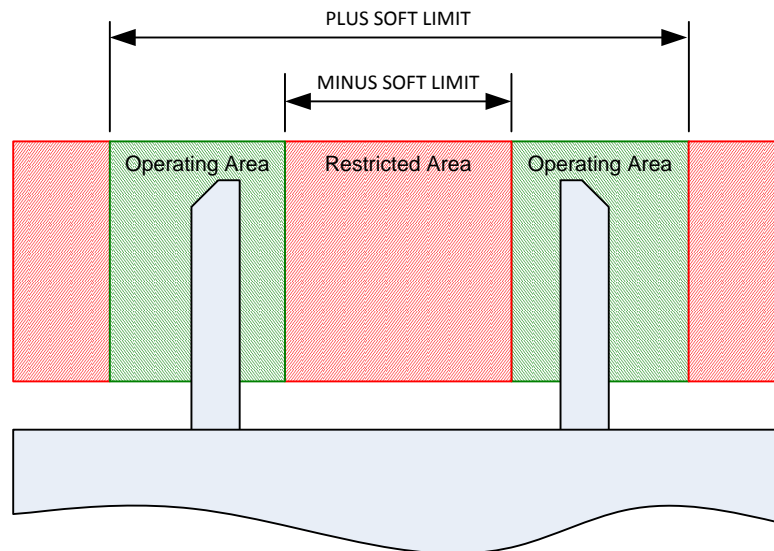
Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_NO_PARAM_EXPECTED: Der Befehl erwartet keine Parameter aber mindestens ein Parameter wurde gesendet.

3.3.5 Setzen der Soft Limits (34h)

Legt die Soft Limits fest sowohl in positiver als auch in negativer Bewegungsrichtung. Mit Hilfe von Soft Limits kann festgelegt werden, dass sich die Greiffinger nur innerhalb eines festgelegten Bereichs bewegen lassen. Wenn Soft Limits aktiviert sind, liefert das Greifmodul einen Fehler bei allen Bewegungsbefehlen, die eine Zielposition außerhalb des festgelegten Bereichs anzufahren versuchen und stellt sicher, dass sich die Greiffinger nicht in den ausgeschlossenen Bereich bewegen lassen. Wenn sich die Finger dennoch außerhalb des festgelegten Bereichs bewegen, wird ein quittierungspflichtiger Fast Stop ausgelöst.



- ⚠ Die Breite der Greiffinger wird vom Greifmodul nicht berücksichtigt. Die Öffnungsweite bezieht sich immer auf die Innenseite der Grundbacken.
- i Wenn sich die Greiffinger beim Aktivieren der Soft Limits außerhalb des erlaubten Bereichs befinden, wird das entsprechende Systemzustandsflag gesetzt und es sind ausschließlich Bewegungsbefehle erlaubt, die die Greiffinger aus dem ausgeschlossenen Bereich heraus bewegen.
- 👉 Die eingestellten Soft Limits gelten nur für die aktuelle Sitzung. Bei einem Neustart des Greifmoduls werden die Soft Limits zurückgesetzt.

Befehls-ID: 34h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0..3	LIMIT_MINUS	Float	Soft Limit in negativer Bewegungsrichtung (mm).
4..7	LIMIT_PLUS	Float	Soft Limit in positiver Bewegungsrichtung (mm).

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

3.3.6 Auslesen der Soft Limits (35h)

Liefert die aktuell eingestellten Soft Limits zurück. Wenn keine Soft Limits eingestellt sind, wird der Statuscode E_NOT_AVAILABLE zurück geliefert.

Befehls-ID: 35h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0..3	LIMIT_MINUS	Float	Soft Limit in negativer Bewegungsrichtung (mm).
4..7	LIMIT_PLUS	Float	Soft Limit in positiver Bewegungsrichtung (mm).

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_NOT_AVAILABLE: Keine Soft Limits gesetzt.

E_INSUFFICIENT_RESOURCES: Speicherüberlauf

E_NO_PARAM_EXPECTED: Der Befehl erwartet keine Parameter aber mindestens ein Parameter wurde gesendet.

3.3.7 Aufheben der Soft Limits (36h)

Löscht die zuvor eingestellten Soft Limits.

Befehls-ID: 36h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Keine Rückgabewerte.


Mögliche Statuscodes:


E_SUCCESS: Befehl erfolgreich.


E_NO_PARAM_EXPECTED: Der Befehl erwartet keine Parameter aber mindestens ein Parameter wurde gesendet.


3.3.8 Overdrive-Modus (37h)

Aktiviert bzw. deaktiviert den Overdrive-Modus. Während im Normalmodus für die eingestellte Greifkraft ausschließlich Werte zulässig sind, die die Nenngreifkraft des Greifmoduls nicht überschreiten, können im Overdrive-Modus auch höhere Greifkräfte (bis zum jeweiligen Overdrive-Limit) vorgegeben werden.

 **Der Overdrive-Modus ist nicht auf allen Greifmodulen der WSG-Serie verfügbar. Weiterführende Informationen entnehmen Sie bitte der Montage- und Betriebsanleitung des Greifmoduls.**

 **Verwenden Sie den Overdrive-Modus mit Vorsicht. Sofern der Overdrive-Modus aktiviert ist und eine Greifkraft vorgegeben wird, die die Nenngreifkraft des Greifmoduls überschreitet, erhöht sich die Leistungsaufnahme des Greifmoduls. Abhängig vom Greifzyklus kann dies zu Überhitzung und in Folge zu erzwungener Abschaltung der Antriebseinheit des Greifmoduls führen. In Einzelfällen kann das Greifmodul durch Überlastung auch beschädigt werden.**

 **Wenn der Overdrive-Modus nicht aktiviert ist und Greifkräfte vorgegeben werden, die die Nenngreifkraft des Greifmoduls überschreiten, wird die Greifkraft automatisch auf die Nenngreifkraft begrenzt.**

 **Das Aktivieren bzw. Deaktivieren des Overdrive-Modus wird im System Log des Greifmoduls vermerkt.**

Befehls-ID: 37h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0	FLAGS	Bitvektor	D7...D1: nicht verwendet, auf 0 zu setzen D0: Overdrive-Modus <i>1: Overdrive-Modus aktivieren</i> Maximal einzustellende Greifkraft entspricht dem Overdrive-Limit <i>0: Overdrive-Modus deaktivieren</i> Maximal einzustellende Greifkraft entspricht der Nenngreifkraft.

Rückgabewerte:

Keine Rückgabewerte.



Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

3.3.9 Kraftsensor tarieren (38h)

Tariert den angeschlossenen Kraftmessfinger, der für die Kraftregelung verwendet wird.

-  **Kraftmessfinger sind nicht für alle Greifmodule der WSG-Serie verfügbar. Weiterführende Informationen entnehmen Sie bitte der Montage- und Betriebsanleitung des Greifmoduls.**
-  **Dieser Befehl ist nur dann zulässig, wenn die Kraftregelung des Greifmoduls nicht aktiv ist (d. h. der Greifzustand darf nicht auf HOLDING stehen, wenn der Befehl ausgeführt wird).**

Befehls-ID: 38h

Verfügbarkeit

Dieser Befehl ist verfügbar ab Firmware-Version 1.1.0.

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_NOT_AVAILABLE: Kein Kraftsensor angeschlossen.

E_ACCESS_DENIED: Befehl ist nicht erlaubt, während die Kraftregelung aktiv ist.

E_NO_PARAM_EXPECTED: Der Befehl erwartet keine Parameter aber mindestens ein Parameter wurde gesendet.



3.4 Systemzustandsbefehle

3.4.1 Systemzustand auslesen (40h)

Liefert den aktuellen Systemzustand („System State“) zurück. Der Befehl unterstützt das automatische Senden von Update-Nachrichten entweder in festgelegten Zeitabständen oder bei Änderung der Zustandsflags.

Sofern keine automatischen Update-Nachrichten gewünscht werden (FLAGS'0=0), liefert der Befehl unmittelbar genau eine Antwortnachricht mit dem aktuellen Systemzustand zurück.

Eine Übersicht über die Bedeutung der einzelnen Zustandsflags befindet sich in Kapitel Anhang B.

-  **Sofern das Senden von Update-Nachrichten ausschließlich bei Änderung der Zustandsflags aktiviert ist, wird das vorgegebene Zeitintervall zwischen zwei Nachrichten dennoch berücksichtigt, auch dann wenn die Änderungsrate höher ist als im Parameter PERIOD_MS angegeben.**
-  **Die Zustandsflags des Systemzustands dienen ausschließlich der Information und sind nicht dafür geeignet, den Greifprozess zu steuern. Zu diesem Zweck darf ausschließlich der Greifzustand verwendet werden (vgl. Kapitel 3.4.2).**

Befehls-ID: 40h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0	FLAGS	Bitvektor	D7...D2: Nicht verwendet, auf 0 zu setzen D1: Update-Nachrichten bei Änderung: 1: Update nur bei Änderung 0: Ständige Updates im vorgegebenen Zeitraster D0: Automatisches Senden von Update-Nachrichten: 1: Automatische Updates aktivieren 0: Automatische Updates deaktivieren
1..2	PERIOD_MS	Integer	Periodendauer für das Versenden automatischer Update-Nachrichten in Millisekunden (ms). Der minimale zulässige Wert beträgt 10 ms.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0..3	SSTATE	Bitvektor	Systemzustand. Eine Übersicht über die Bedeutung der einzelnen Flags befindet sich in Kapitel Anhang B.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).


3.4.2 Greifzustand auslesen (41h)

Liefert den aktuellen Greifzustand („Gripper State“) zurück. Der Greifzustand kann verwendet werden, um den Greifprozess zu überwachen. Die Zustandswerte werden als Ganzzahl codiert. Folgende Zustände sind möglich:

- **Idle (0)**
Der Greifprozess ist im Leerlaufzustand und bereit, neue Befehle zu empfangen.
- **Grasping (1)**
Die Greiffinger bewegen sich zur vorgegebenen Schließungsweite, um ein Teil zu greifen. Es wurde aber noch kein Greifteil erkannt.
- **No part found (2)**
Die Greiffinger haben sich geschlossen, es wurde aber an der vorgegebenen Position kein Greifteil erkannt bzw. die eingestellte Greifkraft konnte nicht innerhalb der definierten Grenzen („Clamping Travel“, vgl. Kapitel 3.2.6) aufgebracht werden. Dieser Zustand bleibt bestehen, bis der nächste Bewegungsbefehl empfangen wird.
- **Part lost (3)**
Ein Teil wurde zunächst korrekt gegriffen, ging aber dann verloren bevor ein Öffnungsbefehl empfangen wurde. Dieser Zustand bleibt bestehen, bis der nächste Bewegungsbefehl empfangen wird.
- **Holding (4)**
Ein Teil wurde erfolgreich gegriffen und wird nun mit der eingestellten Greifkraft gehalten.
- **Releasing (5)**
Die Greiffinger bewegen sich zur eingestellten Öffnungsweite, um ein Teil loszulassen.
- **Positioning (6)**
Die Greiffinger bewegen sich auf Referenzierungsfahrt oder zum Vorpositionieren.
- **Error (7)**
Ein Fehler ist während des Greifprozesses aufgetreten oder ein Fast Stop wurde ausgelöst. Dieser Zustand bleibt erhalten, bis der nächste Bewegungsbefehl empfangen wird. Sofern der FAST STOP-Zustand aktiv ist, muss dieser zunächst quittiert werden.

Der Befehl unterstützt das automatische Senden von Update-Nachrichten entweder in festgelegten Zeitabständen oder bei Änderung des Greifzustandes.

Sofern keine automatischen Update-Nachrichten gewünscht werden (FLAGS'0=0), liefert der Befehl unmittelbar genau eine Antwortnachricht mit dem aktuellen Greifzustand zurück.

 **Sofern das Senden von Update-Nachrichten ausschließlich bei Änderung des Greifzustands aktiviert ist, wird das vorgegebene Zeitintervall zwischen zwei Nachrichten dennoch berücksichtigt, auch dann wenn die Änderungsrate höher ist als im Parameter PERIOD_MS angegeben.**

Befehls-ID: 41h**Befehlsparameter:**

Byte	Symbol	Datentyp	Beschreibung
0	FLAGS	Bitvektor	D7...D2: Nicht verwendet, auf 0 zu setzen D1: Update-Nachrichten bei Änderung: 1: Update nur bei Änderung 0: Ständige Updates im vorgegebenen Zeitraster D0: Automatisches Senden von Update-Nachrichten: 1: Automatische Updates aktivieren 0: Automatische Updates deaktivieren
1..2	PERIOD_MS	Integer	Periodendauer für das Versenden automatischer Update-Nachrichten in Millisekunden (ms). Der minimale zulässige Wert beträgt 10 ms.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0	GSTATE	Enum	Greifzustand: 0: Idle 1: Grasping 2: No part found 3: Part lost 4: Holding 5: Releasing 6: Positioning 7: Error 8 to 255: reserviert

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

3.4.3 Greifstatistik auslesen (42h)

Liefert die aktuelle Greifstatistik zurück. Die Greifstatistik besteht aus der Anzahl der ausgeführten Greifbefehle, der Anzahl gefundener sowie der Anzahl verlorener Greifteile.

Befehls-ID: 40h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0	FLAGS	Bitvektor	D7...D1: Nicht verwendet, auf 0 zu setzen D0: Statistik zurücksetzen: 1: Statistik nach dem Lesen der Werte zurücksetzen 0: Nicht zurücksetzen

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0..3	TOTAL	Integer	Anzahl insgesamt empfangener Greifbefehle
4..5	NO_PART	Integer	Anzahl von Greifbefehlen, bei denen kein Greifteil gefunden wurde.
6..7	LOST_PART	Integer	Anzahl von Greifbefehlen, bei denen das Greifteil vor Öffnung der Finger verloren wurde.

Mögliche Statuscodes:


E_SUCCESS: Befehl erfolgreich.


E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

3.4.4 Öffnungsweite auslesen (43h)

Liefert die aktuelle Öffnungsweite (Position der Greiffinger) zurück. Der Befehl unterstützt das automatische Senden von Update-Nachrichten entweder in festgelegten Zeitabständen oder bei Änderung des Positionswertes um mindestens 0,01 mm.

Sofern keine automatischen Update-Nachrichten gewünscht werden (FLAGS'0=0), liefert der Befehl unmittelbar genau eine Antwortnachricht mit der aktuellen Öffnungsweite zurück.

 Falls das Greifmodul noch nicht referenziert wurde, liefert der Befehl eine Öffnungsweite von 0 zurück und liefert keine Update-Nachrichten bei geänderter Öffnungsweite.

 Sofern das Senden von Update-Nachrichten ausschließlich bei Änderung der Öffnungsweite aktiviert ist, wird das vorgegebene Zeitintervall zwischen zwei Nachrichten dennoch berücksichtigt, auch dann wenn die Änderungsrate höher ist als im Parameter PERIOD_MS angegeben.

 Der Befehl liefert die Distanz zwischen den Greiffingern zurück, nicht ihre absolute Position!

Befehls-ID: 43h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0	FLAGS	Bitvektor	D7..D2: Nicht verwendet, auf 0 zu setzen D1: Update-Nachrichten bei Änderung: 1: Update nur bei Änderung 0: Ständige Updates im vorgegebenen Zeitraster D0: Automatisches Senden von Update-Nachrichten: 1: Automatische Updates aktivieren 0: Automatische Updates deaktivieren
1..2	PERIOD_MS	Integer	Periodendauer für das Versenden automatischer Update-Nachrichten in Millisekunden (ms). Der minimale zulässige Wert beträgt 10 ms.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0	WIDTH	Float	Öffnungsweite in Millimetern (mm).

Mögliche Statuscodes:


E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

3.4.5 Fingergeschwindigkeit auslesen (44h)

Liefert die aktuelle Geschwindigkeit der Greiffinger zurück. Der Befehl unterstützt das automatische Senden von Update-Nachrichten entweder in festgelegten Zeitabständen oder bei Änderung des Geschwindigkeitswertes um mindestens 0,05 mm/s.

Sofern keine automatischen Update-Nachrichten gewünscht werden (FLAGS'0=0), liefert der Befehl unmittelbar genau eine Antwortnachricht mit der aktuellen Geschwindigkeit zurück.

 **Sofern das Senden von Update-Nachrichten ausschließlich bei Änderung der Geschwindigkeit aktiviert ist, wird das vorgegebene Zeitintervall zwischen zwei Nachrichten dennoch berücksichtigt, auch dann wenn die Änderungsrate höher ist als im Parameter PERIOD_MS angegeben.**

 **Der Geschwindigkeitswert bezieht sich auf die relative Geschwindigkeit zwischen den Greiffingern.**

Befehls-ID: 44h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0	FLAGS	Bitvektor	D7..D2: Nicht verwendet, auf 0 zu setzen D1: Update-Nachrichten bei Änderung: 1: Update nur bei Änderung 0: Ständige Updates im vorgegebenen Zeitraster D0: Automatisches Senden von Update-Nachrichten: 1: Automatische Updates aktivieren 0: Automatische Updates deaktivieren
1..2	PERIOD_MS	Integer	Periodendauer für das Versenden automatischer Update-Nachrichten in Millisekunden (ms). Der minimale zulässige Wert beträgt 10 ms.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0	SPEED	Float	Fingergeschwindigkeit in mm/s.

Mögliche Statuscodes:


E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

3.4.6 Greifkraft auslesen (45h)

Liefert die aktuelle Greifkraft zurück. Der Befehl unterstützt das automatische Senden von Update-Nachrichten entweder in festgelegten Zeitabständen oder bei Änderung des Kraftwertes um mindestens 0,05 N.

Sofern keine automatischen Update-Nachrichten gewünscht werden (FLAGS'0=0), liefert der Befehl unmittelbar genau eine Antwortnachricht mit der aktuellen Greifkraft zurück.

 **Sofern das Senden von Update-Nachrichten ausschließlich bei Änderung der Greifkraft aktiviert ist, wird das vorgegebene Zeitintervall zwischen zwei Nachrichten dennoch berücksichtigt, auch dann wenn die Änderungsrate höher ist als im Parameter PERIOD_MS angegeben.**

 **Hinweis: Die Greifkraft ist definiert als das Produkt aus der effektiven Greifkraft mal der Anzahl der Greiffinger.**

Befehls-ID: 45h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0	FLAGS	Bitvektor	D7..D2: Nicht verwendet, auf 0 zu setzen D1: Update-Nachrichten bei Änderung: 1: Update nur bei Änderung 0: Ständige Updates im vorgegebenen Zeitraster D0: Automatisches Senden von Update-Nachrichten: 1: Automatische Updates aktivieren 0: Automatische Updates deaktivieren
1..2	PERIOD_MS	Integer	Periodendauer für das Versenden automatischer Update-Nachrichten in Millisekunden (ms). Der minimale zulässige Wert beträgt 10 ms.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0	FORCE	Float	Greifkraft in Newton (N).

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

3.4.7 Temperatur auslesen (46h)

Liefert die aktuelle Temperatur im Gehäuse des Greifmoduls zurück. Dieser Wert kann verwendet werden, um die Betriebsbedingungen zu überprüfen.

Befehls-ID: 46h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0..1	TEMP	Integer	Temperatur in 0.1 °C-Schritten, e.g. FFF5h = -1.0 °C FFFFh = -0.1 °C 0000h = 0.0 °C 000Ah = 1.0 °C 00C8h = 20.0 °C 01F4h = 50.0 °C

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_NO_PARAM_EXPECTED: Der Befehl erwartet keine Parameter aber mindestens ein Parameter wurde gesendet.

E_INSUFFICIENT_RESOURCES: Speicherüberlauf

3.5 Systemkonfiguration

3.5.1 Systeminformation auslesen (50h)

Liefert Informationen über das Greifmodul zurück. Der Befehl ist auf allen Produkten von Weiss Robotics verfügbar, die das binäre Befehlsprotokoll unterstützen.

Befehls-ID: 50h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0	TYPE	Enum	Gerätetyp: 0: unknown 1: WSG 50 2: WSG 32 (3: Force-Torque Sensor KMS 40) (4: Tactile Sensing Module WTS) 5: WSG 25 6: WSG 70
1	HWREV	Integer	Hardware-Revision
2..3	FW_VERSION	Integer	Firmware-Version: D15..12 Hauptversionsnummer D11..8 Unterversionsnummer 1 D7..4 Unterversionsnummer 2 D3..0 0 für Release-Versionen, 1..15 für Vorabversionen
4..7	SN	Integer	Seriennummer

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_NO_PARAM_EXPECTED: Der Befehl erwartet keine Parameter aber mindestens ein Parameter wurde gesendet.

E_INSUFFICIENT_RESOURCES: Speicherüberlauf

3.5.2 Device Tag setzen (51h)

Setzt das Device Tag. Hierbei handelt es sich um eine allgemeine Zeichenkette, die z. B. verwendet werden kann, um das Greifmodul zu identifizieren und in Umgebungen mit mehreren Greifmodulen von anderen zu unterscheiden. Die maximal zulässige Länge liegt bei 64 Zeichen (ASCII). Sonderzeichen dürfen nicht verwendet werden.

Befehls-ID: 51h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0..n	TAG	String	Device Tag Textstring (ASCII). Maximale Länge von 64 Zeichen.

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_OVERRUN: Tag-Text ist zu lang.

E_INVALID_PARAMETER: Tag enthält unzulässige Zeichen.

E_INSUFFICIENT_RESOURCES: Speicherüberlauf.

3.5.3 Device Tag auslesen (52h)

Liefert das Device Tag zurück. Falls kein Device Tag gesetzt ist, liefert der Befehl den Statuscode E_NOT_AVAILABLE zurück.

Befehls-ID: 52h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0..n	TAG	String	Device Tag Textstring (ASCII).

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_NO_PARAM_EXPECTED: Der Befehl erwartet keine Parameter aber mindestens ein Parameter wurde gesendet.

E_NOT_AVAILABLE: Kein Device Tag gesetzt.

E_INSUFFICIENT_RESOURCES: Speicherüberlauf.

3.5.4 Systemgrenzen auslesen (53h)

Liefert die zulässigen Grenzen für Öffnungsweite (Hub), Geschwindigkeit, Beschleunigung und Greifkraft zurück. Diese Werte können verwendet werden, um die Zulässigkeit der Parameter für Bewegungsbefehle zu überprüfen.

Befehls-ID: 53h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0..3	STROKE	Float	Hub des Greifmoduls in mm
4..7	MIN_SPEED	Float	Minimale Geschwindigkeit in mm/s
8..11	MAX_SPEED	Float	Maximale Geschwindigkeit in mm/s
12..15	MIN_ACC	Float	Minimale Beschleunigung in mm/s ²
16..19	MAX_ACC	Float	Maximale Beschleunigung in mm/s ²
20..23	MIN_FORCE	Float	Minimale Greifkraft in N
24..27	NOM_FORCE	Float	Nenngreifkraft in N
28..31	OVR_FORCE	Float	Maximale Greifkraft im Overdrive-Modus in N ¹

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.


E_NO_PARAM_EXPECTED: Der Befehl erwartet keine Parameter aber mindestens ein Parameter wurde gesendet.

E_INSUFFICIENT_RESOURCES: Speicherüberlauf

¹ Overdrive mode is not supported by all WSG grippers. Please refer to the User's Manual for further information.

3.6 Fingerschnittstelle

Einige Greifmodule der WSG-Serie verfügen über eine integrierte, elektrische Fingerschnittstelle („Sensor Port“) in den Grundbacken, über die passende Sensorfinger verbunden werden können.

 **Die elektrische Fingerschnittstelle ist nicht auf allen Modellen der WSG-Serie verfügbar. Weitere Informationen entnehmen Sie bitte der Montage- und Betriebsanleitung des Greifmoduls.**

3.6.1 Finger 1 Informationen auslesen (60h)

Liefert Informationen über den angeschlossenen Finger zurück. Der Befehl kann verwendet werden, um die Art der angeschlossenen Finger sowie die Länge der vom Sensor bereitgestellten Binärdaten zu bestimmen.

Befehls-ID: 60h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0	TYPE	Enum	Fingertyp: 0: Allgemeiner Finger oder kein Sensorfinger installiert 1: WSG-FMF 2: WSG-DSA 3..255: reserviert
1..2	SIZE	Integer	Länge der Daten (Anzahl Bytes), die vom Sensor geliefert werden und die vom Befehl <i>Finger 1 Daten auslesen</i> (63h, siehe Kapitel 3.6.4) zurück geliefert werden. Fall der Finger das Auslesen von Daten nicht unterstützt, wird der Wert 0000h zurück geliefert.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

E_INSUFFICIENT_RESOURCES: Speicherüberlauf

3.6.2 Finger 1 Zustandsflags auslesen (61h)

Liefert die Zustandsflags für Finger 1 zurück.

Befehls-ID: 61h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0..1	FLAGS	Bitvektor	<i>Finger-Zustandsflags</i>
			Bit Index: Name Beschreibung
			Bit 0: POWER_ON Falls gesetzt, ist der Finger eingeschaltet
			Bit 1: CONFIG_AVAIL Der verbundene Finger stellt Konfigurationsdaten bereit (d. h. ein intelligenter Sensorfinger ist angeschlossen).
			Bit 2: COMM_OPEN Eine Befehlsschnittstelle ist geöffnet
			Bit 3..7: (reserviert)
			Bit 8: POWER_FAULT Ein Überstromfehler wurde erkannt
			Bit 9: COMM_FAULT Ein Kommunikationsfehler ist aufgetreten
Bit 10..15 (reserviert)			

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

E_INDEX_OUT_OF_BOUNDS: Finger index is out of bounds.


E_INSUFFICIENT_RESOURCES: Speicherüberlauf

3.6.3 Finger 1 Stromversorgung ein-/ausschalten (62h)

Schält die Stromversorgung für Finger 1 ein oder aus. Der Befehl kann mit kundenspezifischer Hardware verwendet werden, um das Verhalten des Fingers zu beeinflussen.

Das Einschalten des Fingers wird als asynchroner Befehl ausgeführt, da die Steuerung eine Zeit lang wartet, bis die Stromversorgung tatsächlich eingeschaltet ist. In diesem Fall wird zunächst eine Antwortnachricht mit dem Statuscode E_CMD_PENDING zurück geliefert, bevor nach etwa 500 ms eine weitere Nachricht mit dem Statuscode E_SUCCESS zurück geliefert wird.

Das Ausschalten der Stromversorgung wird unmittelbar durchgeführt, d.h. es wird keine Statusnachricht mit E_CMD_PENDING zurück geliefert.

 **Der Befehl kann nur ausgeführt werden, wenn der angeschlossene Finger vom Typ „generic“ ist. Vordefinierte Sensorfinger (z. B. WSG-FMF, WSG-DSA) werden beim Starten des Greifmoduls automatisch erkannt und in die Steuerung eingebunden und können nicht ein- oder ausgeschaltet werden.**

Befehls-ID: 62h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0	ON/OFF	Enum	Der Wert dieses Bytes muss auf 0 gesetzt werden, um die Stromversorgung auszuschalten oder auf 1, um die Stromversorgung einzuschalten. Die Stromversorgung kann nur für Finger des Typs „generic“ gesteuert werden. Alle Finger werden beim Einschalten des Greifmoduls automatisch mit eingeschaltet.

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

Unmittelbare Statuscodes:

E_SUCCESS: Befehl erfolgreich (beim Ausschalten der Stromversorgung).

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

E_CMD_FAILED: Ein Überstromfehler ist beim Einschalten des Fingers aufgetreten.

E_CMD_PENDING: Stromversorgung wurde eingeschaltet, warte auf das Einschalten des Fingers.

Statuscodes nach der Durchführung des Befehls (nur beim Einschalten der Stromversorgung):

E_SUCCESS: Stromversorgung erfolgreich eingeschaltet.

3.6.4 Finger 1 Daten auslesen (63h)

Liest für vordefinierte Sensorfinger (z. B. WSG-FMF, WSG-DSA) die Sensordaten aus. Die Länge der fingerspezifischen Sensordaten kann mittels des Befehls *Finger 1 Informationen auslesen* (60h, siehe Kapitel 3.6.1) abgefragt werden.

 **Inhalt und Länge der zurück gelieferten Daten hängen vom Typ des angeschlossenen Sensorfingers ab. Bitte beachten Sie dazu die Betriebsanleitung des Sensorfingers.**

Befehls-ID: 63h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Fingerspezifische Daten.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

E_IO_ERROR: Beim Zugriff auf den Sensorfinger ist ein Kommunikationsfehler aufgetreten.

E_NOT_AVAILABLE: Der angeschlossene Finger unterstützt das Auslesen von Daten nicht..

3.6.5 Finger 2 Informationen auslesen (70h)

Liefert Informationen über den angeschlossenen Finger zurück. Der Befehl kann verwendet werden, um die Art der angeschlossenen Finger sowie die Länge der vom Sensor bereitgestellten Binärdaten zu bestimmen.

Befehls-ID: 70h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0	TYPE	Enum	Fingertyp: 0: Allgemeiner Finger oder kein Sensorfinger installiert 1: WSG-FMF 2: WSG-DSA 3..255: reserviert
1..2	SIZE	Integer	Länge der Daten (Anzahl Bytes), die vom Sensor geliefert werden und die vom Befehl <i>Finger 2 Daten auslesen</i> (73h, siehe Kapitel 3.6.8) zurück geliefert werden. Fall der Finger das Auslesen von Daten nicht unterstützt, wird der Wert 0000h zurück geliefert.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

E_INSUFFICIENT_RESOURCES: Speicherüberlauf

3.6.6 Finger 2 Zustandsflags auslesen (71h)

Liefert die Zustandsflags für Finger 2 zurück.

Befehls-ID: 71h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Byte	Symbol	Datentyp	Beschreibung
0..1	FLAGS	Bitvektor	<i>Finger-Zustandsflags</i>
			Bit Index: Name Beschreibung
			Bit 0: POWER_ON Falls gesetzt, ist der Finger eingeschaltet
			Bit 1: CONFIG_AVAIL Der verbundene Finger stellt Konfigurationsdaten bereit (d. h. ein intelligenter Sensorfinger ist angeschlossen).
			Bit 2: COMM_OPEN Eine Befehlsschnittstelle ist geöffnet
			Bit 3..7: (reserviert)
			Bit 8: POWER_FAULT Ein Überstromfehler wurde erkannt
			Bit 9: COMM_FAULT Ein Kommunikationsfehler ist aufgetreten
Bit 10..15 (reserviert)			

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).


E_INSUFFICIENT_RESOURCES: Speicherüberlauf

3.6.7 Finger 2 Stromversorgung ein-/ausschalten (72h)

Schält die Stromversorgung für Finger 2 ein oder aus. Der Befehl kann mit kundenspezifischer Hardware verwendet werden, um das Verhalten des Fingers zu beeinflussen.

Das Einschalten des Fingers wird als asynchroner Befehl ausgeführt, da die Steuerung eine Zeit lang wartet, bis die Stromversorgung tatsächlich eingeschaltet ist. In diesem Fall wird zunächst eine Antwortnachricht mit dem Statuscode E_CMD_PENDING zurück geliefert, bevor nach etwa 500 ms eine weitere Nachricht mit dem Statuscode E_SUCCESS zurück geliefert wird.

Das Ausschalten der Stromversorgung wird unmittelbar durchgeführt, d.h. es wird keine Statusnachricht mit E_CMD_PENDING zurück geliefert.

 **Der Befehl kann nur ausgeführt werden, wenn der angeschlossene Finger vom Typ „generic“ ist. Vordefinierte Sensorfinger (z. B. WSG-FMF, WSG-DSA) werden beim Starten des Greifmoduls automatisch erkannt und in die Steuerung eingebunden und können nicht ein- oder ausgeschaltet werden.**

Befehls-ID: 72h

Befehlsparameter:

Byte	Symbol	Datentyp	Beschreibung
0	ON/OFF	Enum	Der Wert dieses Bytes muss auf 0 gesetzt werden, um die Stromversorgung auszuschalten oder auf 1, um die Stromversorgung einzuschalten. Die Stromversorgung kann nur für Finger des Typs „generic“ gesteuert werden. Alle Finger werden beim Einschalten des Greifmoduls automatisch mit eingeschaltet.

Rückgabewerte:

Keine Rückgabewerte.

Mögliche Statuscodes:

Unmittelbare Statuscodes:

E_SUCCESS: Befehl erfolgreich (when disabling power).

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

E_CMD_FAILED: Ein Überstromfehler ist beim Einschalten des Fingers aufgetreten.

E_CMD_PENDING: Stromversorgung wurde eingeschaltet, warte auf das Einschalten des Fingers.

Statuscodes nach der Durchführung des Befehls (nur beim Einschalten der Stromversorgung):

E_SUCCESS: Stromversorgung erfolgreich eingeschaltet.

3.6.8 Finger 2 Daten auslesen (73h)

Liest für vordefinierte Sensorfinger (z. B. WSG-FMF, WSG-DSA) die Sensordaten aus. Die Länge der fingerspezifischen Sensordaten kann mittels des Befehls *Finger 2 Informationen auslesen* (70h, siehe Kapitel 3.6.5) abgefragt werden.

 **Inhalt und Länge der zurück gelieferten Daten hängen vom Typ des angeschlossenen Sensorfingers ab. Bitte beachten Sie dazu die Betriebsanleitung des Sensorfingers.**

Befehls-ID: 73h

Befehlsparameter:

Keine Parameter.

Rückgabewerte:

Fingerspezifische Daten.

Mögliche Statuscodes:

E_SUCCESS: Befehl erfolgreich.

E_CMD_FORMAT_ERROR: Fehler im Befehlsformat (z. B. Länge der Nachricht).

E_IO_ERROR: Beim Zugriff auf den Sensorfinger ist ein Kommunikationsfehler aufgetreten.

E_NOT_AVAILABLE: Der angeschlossene Finger unterstützt das Auslesen von Daten nicht.

Anhang A. Statuscodes

Alle Befehle werden vom Greifmodul mit einem Statuscode quittiert. Die verfügbaren Statuscodes werden in der folgenden Tabelle beschrieben.

Statuscode	Symbolname	Beschreibung
0	E_SUCCESS	Kein Fehler aufgetreten, Befehl erfolgreich
1	E_NOT_AVAILABLE	Funktion oder Daten nicht verfügbar
2	E_NO_SENSOR	Kein Messumformer angeschlossen
3	E_NOT_INITIALIZED	Gerät nicht initialisiert
4	E_ALREADY_RUNNING	Datenerfassung wird bereits ausgeführt
5	E_FEATURE_NOT_SUPPORTED	Die Funktion ist nicht verfügbar
6	E_INCONSISTENT_DATA	Einer oder mehrere Parameter sind inkonsistent
7	E_TIMEOUT	Zeitüberschreitung
8	E_READ_ERROR	Fehler beim Lesen von Daten
9	E_WRITE_ERROR	Fehler beim Schreiben von Daten
10	E_INSUFFICIENT_RESOURCES	Nicht genügend Speicher vorhanden
11	E_CHECKSUM_ERROR	Prüfsummenfehler
12	E_NO_PARAM_EXPECTED	Parameter übergeben, obwohl keiner erwartet
13	E_NOT_ENOUGH_PARAMS	Zu wenige Parameter für den Befehl übergeben
14	E_CMD_UNKNOWN	Unbekannter Befehl
15	E_CMD_FORMAT_ERROR	Fehler im Befehlsformat
16	E_ACCESS_DENIED	Zugriff verweigert
17	E_ALREADY_OPEN	Schnittstelle ist bereits geöffnet
18	E_CMD_FAILED	Fehler während der Ausführung eines Befehls
19	E_CMD_ABORTED	Befehlsausführung vom Benutzer abgebrochen
20	E_INVALID_HANDLE	Ungültiges Handle
21	E_NOT_FOUND	Gerät oder Datei nicht gefunden

22	E_NOT_OPEN	Gerät oder Datei nicht geöffnet
23	E_IO_ERROR	Ein-/Ausgabefehler
24	E_INVALID_PARAMETER	Ungültiger Parameter
25	E_INDEX_OUT_OF_BOUNDS	Index außerhalb des zulässigen Bereichs
26	E_CMD_PENDING	Der Befehl wurde noch nicht vollständig ausgeführt. Eine Rückmeldung mit Statuscode folgt nach Ausführung des Befehls.
27	E_OVERRUN	Datenüberlauf
28	E_RANGE_ERROR	Bereichsfehler
29	E_AXIS_BLOCKED	Achse blockiert
30	E_FILE_EXISTS	Datei existiert bereits

Anhang B. Systemzustandsflags

Die Systemzustandsflags sind in einem 32 Bit breiten Bitvektor angeordnet, der über den Befehl Systemzustand auslesen (40h, siehe Kapitel 3.4.1) ausgelesen werden kann. Die einzelnen Bits und ihre Bedeutung sind in untenstehender Tabelle aufgelistet.

Bit Nr.	Flag-Name	Beschreibung
D31..21	reserviert	Diese Bits werden aktuell nicht verwendet, können aber in zukünftig veröffentlichten Firmware-Versionen verwendet werden.
D20	SF_SCRIPT_FAILURE	Skript Fehler. Während der Ausführung eines Skripts ist ein Fehler aufgetreten und das Skript wurde abgebrochen. Das Flag wird zurückgesetzt, sobald ein Skript gestartet wird.
D19	SF_SCRIPT_RUNNING	Ein Skript wird momentan ausgeführt. Das Flag wird zurückgesetzt wenn das Skript normal beendet wurde, ein Skriptfehler auftrat oder es vom Benutzer manuell beendet wurde.
D18	SF_CMD_FAILURE	Befehl gescheitert. Der letzte Befehl hat einen Fehler zurückgegeben.
D17	SF_FINGER_FAULT	Fingerfehler. Der Status mindestens eines Fingers ist weder „operating“ noch „not connected“. Bitte prüfen Sie die Fingerzustandsflags für eine genauere Fehlerbeschreibung.
D16	SF_CURR_FAULT	Zu hoher Motorstrom. Der Motor hat die maximal zulässige Wärmeverlustleistung erreicht. Das Flag wird automatisch zurückgesetzt, sobald sich der Motor erholt hat. Danach kann der dadurch bedingte Fast Stop quittiert werden.
D15	SF_POWER_FAULT	Spannungsversorgung fehlerhaft. Die Versorgungsspannung ist außerhalb des zulässigen Bereichs. Bitte prüfen Sie die angeschlossene Stromversorgung.

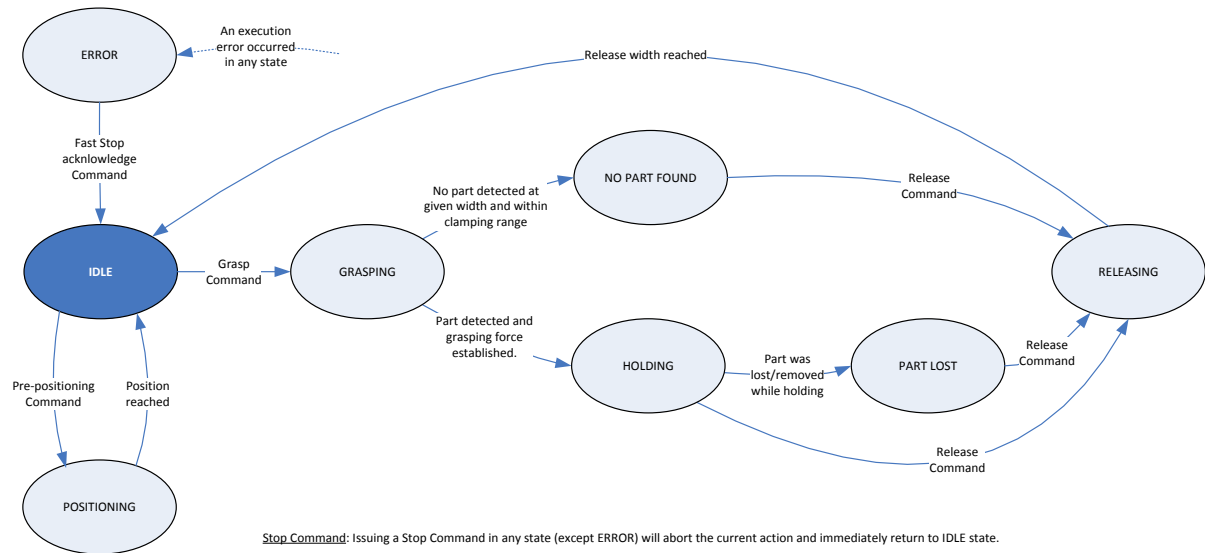
D14	SF_TEMP_FAULT	<p>Temperaturfehler.</p> <p>Der Temperatursensor auf der Steuerungsplatine im Inneren des Greifmoduls hat eine kritische Temperatur erreicht. Alle Bewegungsbefehle werden unterbunden, bis die Temperatur wieder unter die kritische Grenze gefallen ist.</p>
D13	SF_TEMP_WARNING	<p>Temperaturwarnung.</p> <p>Der Temperatursensor auf der Steuerungsplatine im Inneren des Greifmoduls nähert sich der kritischen Temperatur.</p>
D12	SF_FAST_STOP	<p>Fast Stop.</p> <p>Das Greifmodul wurde aufgrund eines Fehlers gestoppt. Um das Flag zurückzusetzen und die Bewegungsbefehle wieder zu aktivieren, muss der Fehlerzustand quittiert werden.</p>
D11..10	reserviert	<p>Diese Bits werden aktuell nicht verwendet, können aber in zukünftig veröffentlichten Firmware-Versionen verwendet werden.</p>
D9	SF_FORCECNTL_MODE	<p>Kraft geregelter Modus.</p> <p>Die Kraftregelung ist momentan aktiv unter Verwendung des installierten Kraftmessfingers (WSG-FMF). Wenn dieses Flag nicht gesetzt ist, wird die Greifkraft auf Basis des Motorstroms näherungsweise geregelt.</p>
D8	SF_OVERDRIVE_MODE	<p>Overdrive Modus².</p> <p>Das Greifmodul befindet sich im Overdrive-Modus und die Greifkraft kann bis auf den Wert der Overdrive-Kraftgrenze erhöht werden. Wenn dieses Bit nicht gesetzt ist, kann die Greifkraft nicht höher als die Nenngreifkraft des Moduls gesetzt werden.</p>

² Der Overdrive Modus wird nicht von allen Greifmodulen der WSG-Serie unterstützt. Bitte beachten Sie die Montage- und Bedienungsanleitung für weiterführende Informationen.

D7	SF_TARGET_POS_REACHED	<p>Zielposition erreicht.</p> <p>Wird gesetzt, sobald die Zielposition erreicht wurde. Das Flag ist nicht mit SF_MOVING synchronisiert, so dass eine Verzögerung zwischen dem Zurücksetzen von SF_MOVING und dem Setzen von SF_TARGET_POS auftreten kann.</p>
D6	SF_AXIS_STOPPED	<p>Achse gestoppt.</p> <p>Ein vorheriger Bewegungsbefehl wurde durch den Stop-Befehl abgebrochen. Das Flag wird durch den nächsten Bewegungsbefehl wieder zurückgesetzt.</p>
D5	SF_SOFT_LIMIT_PLUS	<p>Soft Limit in positiver Richtung erreicht.</p> <p>Die Finger haben die definierten Soft Limits in positiver Bewegungsrichtung erreicht. Eine weitere Bewegung in diese Richtung ist nicht erlaubt. Das Flag wird gelöscht, wenn sich die Finger wieder von der Position entfernen.</p>
D4	SF_SOFT_LIMIT_MINUS	<p>Soft Limit in negativer Richtung erreicht.</p> <p>Die Finger haben die definierten Soft Limits in negativer Bewegungsrichtung erreicht. Eine weitere Bewegung in diese Richtung ist nicht erlaubt. Das Flag wird gelöscht, wenn sich die Finger wieder von der Position entfernen.</p>
D3	SF_BLOCKED_PLUS	<p>Achse ist in positiver Bewegungsrichtung blockiert.</p> <p>Wird gesetzt, wenn die Achse in positiver Bewegungsrichtung blockiert ist. Das Flag wird zurückgesetzt, wenn die Blockade gelöst ist oder ein Stop-Befehl erteilt wurde.</p>
D2	SF_BLOCKED_MINUS	<p>Achse ist in negativer Bewegungsrichtung blockiert.</p> <p>Wird gesetzt, wenn die Achse in negativer Bewegungsrichtung blockiert ist. Das Flag wird zurückgesetzt, wenn die Blockade gelöst ist oder ein Stop-Befehl erteilt wurde.</p>
D1	SF_MOVING	<p>Die Finger sind gerade in Bewegung.</p> <p>Das Flag wird gesetzt, sobald eine Bewegung gestartet wurde (z.B. Befehl <i>Vorpositionieren</i>) und wird automatisch zurückgesetzt, wenn die Bewegung stoppt.</p>
D0	SF_REFERENCED	<p>Finger sind referenziert.</p> <p>Ist dieses Flag gesetzt, dann ist das Greifmodul referenziert und akzeptiert Bewegungsbefehle.</p>

Anhang C. Greifzustände

Das folgende Diagramm stellt die verfügbaren Greifzustände („Gripper States“) und zugehörigen Zustandsübergänge im Normalbetrieb beim Ausführen von Bewegungsbefehlen dar.



Anhang D. Beispielcode zum Berechnen der CRC-Prüfsumme

Der folgende, in ANSI C geschriebene Beispielcode illustriert, wie die Prüfsumme für die Kommunikation mit dem Greifmodul berechnet wird.

```
#include <stdio.h>
#include <stdlib.h>

typedef struct
{
    unsigned short length;    //!< Length of the message's payload in bytes
                              // (0, if the message has no payload)
    unsigned char id;        //!< ID of the message
    unsigned char *data;     //!< Pointer to the message's payload
} TMESSAGE;    //!< command message format

//! Status codes
typedef enum
{
    E_SUCCESS = 0,           //!< No error
    E_NOT_AVAILABLE,        //!< Device, service or data is not available
    E_NO_SENSOR,            //!< No sensor connected
    E_NOT_INITIALIZED,      //!< The device is not initialized
    E_ALREADY_RUNNING,      //!< Service is already running
    E_FEATURE_NOT_SUPPORTED, //!< The asked feature is not supported
    E_INCONSISTENT_DATA,    //!< One or more dependent parameters mismatch
    E_TIMEOUT,              //!< Timeout error
    E_READ_ERROR,           //!< Error while reading from a device
    E_WRITE_ERROR,          //!< Error while writing to a device
    E_INSUFFICIENT_RESOURCES, //!< No memory available
    E_CHECKSUM_ERROR,       //!< Checksum error
    E_NO_PARAM_EXPECTED,    //!< No parameters expected
    E_NOT_ENOUGH_PARAMS,    //!< Not enough parameters
    E_CMD_UNKNOWN,          //!< Unknown command
    E_CMD_FORMAT_ERROR,     //!< Command format error
    E_ACCESS_DENIED,        //!< Access denied
    E_ALREADY_OPEN,         //!< The interface is already open
    E_CMD_FAILED,           //!< Command failed
    E_CMD_ABORTED,          //!< Command aborted
    E_INVALID_HANDLE,       //!< invalid handle
    E_NOT_FOUND,            //!< device not found
    E_NOT_OPEN,             //!< device not open
    E_IO_ERROR,             //!< I/O error
    E_INVALID_PARAMETER,    //!< invalid parameter
    E_INDEX_OUT_OF_BOUNDS,  //!< index out of bounds
    E_CMD_PENDING,          //!< Command execution needs more time
    E_OVERRUN,              //!< Data overrun
    E_RANGE_ERROR,          //!< Range error
    E_AXIS_BLOCKED,         //!< Axis is blocked
    E_FILE_EXISTS           //!< File already exists
} TStat;

#define SER_MSG_NUM_HEADER_BYTES 3    //!< number of header bytes
#define SER_MSG_HEADER_BYTE 0xAA     //!< header byte value

const unsigned short CRC_TABLE[256] = {
    0000h, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7,
    0x8108, 0x9129, 0xa14a, 0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef,
    0x1231, 0x0210, 0x3273, 0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,
    0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff, 0xe3de,
    0x2462, 0x3443, 0x0420, 0x1401, 0x64e6, 0x74c7, 0x44a4, 0x5485,
    0xa56a, 0xb54b, 0x8528, 0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,
    0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695, 0x46b4,
    0xb75b, 0xa77a, 0x9719, 0x8738, 0xf7df, 0xe7fe, 0xd79d, 0xc7bc,
    0x48c4, 0x58e5, 0x6886, 0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,
    0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a, 0xb92b,
```

```

0x5af5, 0x4ad4, 0x7ab7, 0x6a96, 0x1a71, 0x0a50, 0x3a33, 0x2a12,
0xdbfd, 0xcbdc, 0xfbbf, 0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,
0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60, 0x1c41,
0xedaе, 0xfd8f, 0xcdec, 0xddcd, 0xad2a, 0xbd0b, 0x8d68, 0x9d49,
0x7e97, 0x6eb6, 0x5ed5, 0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,
0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59, 0x8f78,
0x9188, 0x81a9, 0xb1ca, 0xaleb, 0xd10c, 0xc12d, 0xf14e, 0xe16f,
0x1080, 0x00a1, 0x30c2, 0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f, 0xf35e,
0x02b1, 0x1290, 0x22f3, 0x32d2, 0x4235, 0x5214, 0x6277, 0x7256,
0xb5ea, 0xa5cb, 0x95a8, 0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
0xa7db, 0xb7fa, 0x8799, 0x97b8, 0xe75f, 0xf77e, 0xc71d, 0xd73c,
0x26d3, 0x36f2, 0x0691, 0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a, 0xa9ab,
0x5844, 0x4865, 0x7806, 0x6827, 0x18c0, 0x08e1, 0x3882, 0x28a3,
0xcb7d, 0xdb5c, 0xeb3f, 0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3, 0x3a92,
0xfd2e, 0xed0f, 0xdd6c, 0xcd4d, 0xbdaa, 0xad8b, 0x9de8, 0x8dc9,
0x7c26, 0x6c07, 0x5c64, 0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,
0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9, 0x9ff8,
0x6e17, 0x7e36, 0x4e55, 0x5e74, 0x2e93, 0x3eb2, 0x0ed1, 0x1ef0
};

/*****
/*!
Calculates the CRC checksum of an array by using a table.
The start value for calculating the CRC should be set to 0xFFFF.

@param *data points to the byte array from which checksum should
        be calculated
@param size size of the byte array
@param crc value calculated over another array and start value
        of the crc16 calculation

@return CRC16 checksum
*/
*****/

static unsigned short checksum_update_crc16( unsigned char *data,
        unsigned int size, unsigned short crc )
{
    unsigned long c;
    /* process each byte prior to checksum field */
    for ( c=0; c < size; c++ )
    {
        crc = CRC_TABLE[ ( crc ^ *( data ++ ) ) & 0x00FF ] ^ ( crc >> 8 );
    }
    return( crc );
}

/*****
/*!
Builds a data packet from the given message.
You have to free the returned buffer, if you do not use it anymore.

@param *msg Pointer to the source message
@param *size Returns the size of the created buffer

@return buffer containing the bitwise packet data or NULL in case
        of an error.
*/
*****/

static unsigned char *msg_build( TMESSAGE * msg, unsigned int *size )
{

```



```

unsigned char *buf;
unsigned short chksum;
unsigned int c, len;

len = MSG_NUM_HEADER_BYTES + 3 + 2 + msg->length;

buf = malloc( len );
if ( !buf )
{
    *size = 0;
    return( NULL );
}

// Assemble the message header:
for ( c=0; c<MSG_NUM_HEADER_BYTES; c++ ) buf[c] = MSG_HEADER_BYTE;
buf[ MSG_NUM_HEADER_BYTES ] = msg->id; // Message ID
buf[ MSG_NUM_HEADER_BYTES + 1 ] = lo( msg->length ); // Msg. length low byte
buf[ MSG_NUM_HEADER_BYTES + 2 ] = hi( msg->length ); // Msg. length high byte

// Copy payload to buffer:
if ( msg->length ) memcpy( &buf[ MSG_NUM_HEADER_BYTES + 3 ], msg->data, msg->length );

// Calculate the checksum over the header, include the preamble:
chksum = checksum_update_crc16( buf, MSG_NUM_HEADER_BYTES + 3 + msg->length, 0xFFFF );

// Add checksum to message:
buf[ MSG_NUM_HEADER_BYTES + 3 + msg->length ] = lo( chksum );
buf[ MSG_NUM_HEADER_BYTES + 4 + msg->length ] = hi( chksum );

*size = len;
return( buf );
}

/*****
/*!
Send a message to an open file handle

@param *file Handle of an open file to which the message should be sent
@param *msg Pointer to the message that should be sent

@return E_SUCCESS, if successful, otherwise error code
*/
*****/

TStat msg_send( FILE * file, TMESSAGE * msg )
{
    unsigned int c, size;

    // Convert message into byte sequence:
    unsigned char *buf = msg_build( msg, &size );
    if ( !buf ) return( E_INSUFFICIENT_RESOURCES );

    // Transmit buffer:
    c = fwrite( buf, size, 1, file );

    // Free allocated memory:
    free( buf );
    if ( c != 1 ) return( E_WRITE_ERROR );

    return( E_SUCCESS );
}

```



www.weiss-robotics.com

© Weiss Robotics GmbH & Co. KG. Alle Rechte vorbehalten.

Die in diesem Dokument angegebenen technischen Daten können zum Zwecke der Produktverbesserung ohne Vorankündigung geändert werden. Warenzeichen sind Eigentum des jeweiligen Eigentümers. Unsere Produkte sind nicht für den Einsatz in lebenserhaltenden Systemen oder für Systeme, bei denen ein Fehlverhalten zu Personenschäden führen könnte, vorgesehen.