



Benutzerhandbuch

ASCII

Handshake-Mechanismus

Hilscher Gesellschaft für Systemautomation mbH

www.hilscher.com

DOC100212UM03DE | Revision 3 | Deutsch | 2011-05 | Freigegeben | Öffentlich

Inhaltsverzeichnis

1	EINLEITUNG	3
1.1	Über dieses Handbuch	3
1.1.1	Änderungsübersicht.....	3
1.1.2	Bezug auf Firmware und Software	4
1.1.3	Konventionen in diesem Handbuch.....	5
1.2	Rechtliche Hinweise	6
1.2.1	Copyright	6
1.2.2	Wichtige Hinweise	6
1.2.3	Haftungsausschluss	7
1.2.4	Gewährleistung.....	7
1.2.5	Exportbestimmungen	8
1.2.6	Eingetragene Warenzeichen	8
2	ASCII HANDSHAKE-MECHANISMUS.....	9
2.1	Einleitung	9
2.2	E/A-Datenstruktur zur Datenübergabe von und zur Steuerung	10
2.2.1	Ausgabedaten-Struktur - Daten von der übergeordneten Steuerung nach ASCII	10
2.2.2	Eingabedaten-Struktur - Daten von ASCII zur übergeordneten Steuerung	10
2.3	Handshake und Initialisierung der E/A-Kommunikation.....	11
2.3.1	Aufbau der Synchronisationsregister in den E/A-Daten.....	13
2.3.2	Initialisierung der Kommunikation	15
2.3.3	Verarbeitungsbestätigung zwischen übergeordneter Steuerung und ASCII.....	16
3	FEHLERCODES	19
4	ANHANG	20
4.1	Abbildungsverzeichnis	20
4.2	Tabellenverzeichnis	20
4.3	Kontakte.....	21

1 Einleitung

1.1 Über dieses Handbuch

Dieses Dokument beschreibt den Handshake-Mechanismus der seriellen ASCII Kommunikation.

Das ASCII Protokoll zur seriellen Datenübertragung wird auf netTAP NT 50, netTAP NT 100 und netBRICK NB 100 -Gateway-Geräten eingesetzt.

1.1.1 Änderungsübersicht

Index	Datum	Kapitel	Änderungen
1	2010-02-19	Alle	Erstellt
2	2010-03-18	2.3	Beschreibung in Abschnitt <i>Handshake und Initialisierung der E/A-Kommunikation</i> erweitert: Bitpaar für ASCII Telegramm senden bzw. empfangen und <i>Zusammenhang ASCII senden/empfangen und ASCII Betriebsart</i>
3	2011-05-30	1.1.2	Abschnitt <i>Bezug auf Firmware und Software</i> : netTAP NT 50 und netBRICK NB 100 ergänzt

Tabelle 1: Änderungsübersicht

1.1.2 Bezug auf Firmware und Software

Firmware

Firmware-Datei	Firmware-Version
N5ASCxxx.NXF bzw. N5xxxASC.NXF netTAP NT 50 Firmwares mit ASCII Protokoll	ab 1.0.x.x
NTxxxASC.NXF netTAP NT 100 Firmwares mit ASCII Protokoll	ab 1.3.x.x
NBASCxxx.NXF netBRICK NB 100 Firmwares mit ASCII Protokoll	ab 1.4.x.x

Tabelle 2: Bezug auf Firmware

Software

Software	Software-Version
SYCONnet netX setup.exe	ab 1.310.x.x

Tabelle 3: Bezug auf Software

1.1.3 Konventionen in diesem Handbuch

Handlungsanweisungen, ein Ergebnis eines Handlungsschrittes bzw. Hinweise sind wie folgt gekennzeichnet:

Handlungsanweisungen:

➤ <Anweisung>

oder

1. <Anweisung>

2. <Anweisung>

Ergebnisse:

➤ <Ergebnis>

Hinweise:



Wichtig: <Wichtiger Hinweis>



Hinweis: <Hinweis>



<Hinweis, wo Sie weitere Informationen finden können>

1.2 Rechtliche Hinweise

1.2.1 Copyright

© Hilscher, 2010-2011, Hilscher Gesellschaft für Systemautomation mbH

Alle Rechte vorbehalten.

Die Bilder, Fotografien und Texte der Begleitmaterialien (Benutzerhandbuch, Begleittexte, Dokumentation etc.) sind durch deutsches und internationales Urheberrecht sowie internationale Handels- und Schutzbestimmungen geschützt. Sie sind ohne vorherige schriftliche Genehmigung nicht berechtigt, diese vollständig oder teilweise durch technische oder mechanische Verfahren zu vervielfältigen (Druck, Fotokopie oder anderes Verfahren), unter Verwendung elektronischer Systeme zu verarbeiten oder zu übertragen. Es ist Ihnen untersagt, Veränderungen an Copyrightvermerken, Kennzeichen, Markenzeichen oder Eigentumsangaben vorzunehmen. Darstellungen werden ohne Rücksicht auf die Patentlage mitgeteilt. Die in diesem Dokument enthaltenen Firmennamen und Produktbezeichnungen sind möglicherweise Marken bzw. Warenzeichen der jeweiligen Inhaber und können warenzeichen-, marken- oder patentrechtlich geschützt sein. Jede Form der weiteren Nutzung bedarf der ausdrücklichen Genehmigung durch den jeweiligen Inhaber der Rechte.

1.2.2 Wichtige Hinweise

Das Benutzerhandbuch, Begleittexte und die Dokumentation wurden mit größter Sorgfalt erarbeitet. Fehler können jedoch nicht ausgeschlossen werden. Eine Garantie, die juristische Verantwortung für fehlerhafte Angaben oder irgendeine Haftung kann daher nicht übernommen werden. Sie werden darauf hingewiesen, dass Beschreibungen in dem Benutzerhandbuch, den Begleittexte und der Dokumentation weder eine Garantie, noch eine Angabe über die nach dem Vertrag vorausgesetzte Verwendung oder eine zugesicherte Eigenschaft darstellen. Es kann nicht ausgeschlossen werden, dass das Benutzerhandbuch, die Begleittexte und die Dokumentation nicht vollständig mit den beschriebenen Eigenschaften, Normen oder sonstigen Daten der gelieferten Produkte übereinstimmen. Eine Gewähr oder Garantie bezüglich der Richtigkeit oder Genauigkeit der Informationen wird nicht übernommen.

Wir behalten uns das Recht vor, unsere Produkte und deren Spezifikation, sowie zugehörige Benutzerhandbücher, Begleittexte und Dokumentationen jederzeit und ohne Vorankündigung zu ändern, ohne zur Anzeige der Änderung verpflichtet zu sein. Änderungen werden in zukünftigen Manuals berücksichtigt und stellen keine Verpflichtung dar; insbesondere besteht kein Anspruch auf Überarbeitung gelieferter Dokumente. Es gilt jeweils das Manual, das mit dem Produkt ausgeliefert wird.

Die Hilscher Gesellschaft für Systemautomation mbH haftet unter keinen Umständen für direkte, indirekte, Neben- oder Folgeschäden oder Einkommensverluste, die aus der Verwendung der hier enthaltenen Informationen entstehen.

1.2.3 Haftungsausschluss

Die Software wurde von der Hilscher Gesellschaft für Systemautomation mbH sorgfältig erstellt und getestet und wird im reinen Ist-Zustand zur Verfügung gestellt. Es kann keine Gewährleistung für die Leistungsfähigkeit und Fehlerfreiheit der Software für alle Anwendungsbedingungen und -fälle und die erzielten Arbeitsergebnisse bei Verwendung der Software durch den Benutzer übernommen werden. Die Haftung für etwaige Schäden, die durch die Verwendung der Hard- und Software oder der zugehörigen Dokumente entstanden sein könnten, beschränkt sich auf den Fall des Vorsatzes oder der grob fahrlässigen Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen vorhersehbaren Schaden begrenzt.

Es ist strikt untersagt, die Software in folgenden Bereichen zu verwenden:

- für militärische Zwecke oder in Waffensystemen;
- zum Entwurf, zur Konstruktion, Wartung oder zum Betrieb von Nuklearanlagen;
- in Flugsicherungssystemen, Flugverkehrs- oder Flugkommunikationssystemen;
- in Lebenserhaltungssystemen;
- in Systemen, in denen Fehlfunktionen der Software körperliche Schäden oder Verletzungen mit Todesfolge nach sich ziehen können.

Sie werden darauf hingewiesen, dass die Software nicht für die Verwendung in Gefahrumgebungen erstellt worden ist, die ausfallsichere Kontrollmechanismen erfordern. Die Benutzung der Software in einer solchen Umgebung geschieht auf eigene Gefahr; jede Haftung für Schäden oder Verluste aufgrund unerlaubter Benutzung ist ausgeschlossen.

1.2.4 Gewährleistung

Obwohl die Hard- und Software mit aller Sorgfalt entwickelt und intensiv getestet wurde, übernimmt die Hilscher Gesellschaft für Systemautomation mbH keine Garantie für die Eignung für irgendeinen Zweck, der nicht schriftlich bestätigt wurde. Es kann nicht gewährleistet werden, dass die Hard- und Software Ihren Anforderungen entspricht, die Verwendung der Software unterbrechungsfrei und die Software fehlerfrei ist. Eine Garantie auf Nichtübertretung, Nichtverletzung von Patenten, Eigentumsrecht oder Freiheit von Einwirkungen Dritter wird nicht gewährt. Weitere Garantien oder Zusicherungen hinsichtlich Marktgängigkeit, Rechtsmangelfreiheit, Integrierung oder Brauchbarkeit für bestimmte Zwecke werden nicht gewährt, es sei denn, diese sind nach geltendem Recht vorgeschrieben und können nicht eingeschränkt werden. Gewährleistungsansprüche beschränken sich auf das Recht, Nachbesserung zu verlangen.

1.2.5 Exportbestimmungen

Das gelieferte Produkt (einschließlich der technischen Daten) unterliegt den gesetzlichen Export- bzw. Importgesetzen sowie damit verbundenen Vorschriften verschiedener Länder, insbesondere denen von Deutschland und den USA. Die Software darf nicht in Länder exportiert werden, in denen dies durch das US-amerikanische Exportkontrollgesetz und dessen ergänzender Bestimmungen verboten ist. Sie verpflichten sich, die Vorschriften strikt zu befolgen und in eigener Verantwortung einzuhalten. Sie werden darauf hingewiesen, dass Sie zum Export, zur Wiederausfuhr oder zum Import des Produktes unter Umständen staatlicher Genehmigungen bedürfen.

1.2.6 Eingetragene Warenzeichen

Windows® 2000, Windows® XP, Windows® Vista und Windows® 7 sind eingetragene Warenzeichen der Microsoft Corporation.

Alle anderen erwähnten Marken sind Eigentum Ihrer jeweiligen rechtmäßigen Inhaber.

2 ASCII Handshake-Mechanismus

2.1 Einleitung

Die übergeordnete Steuerung kommuniziert über ein zyklisch ablaufendes Protokoll mit dem netTAP- bzw. netBRICK-Gerät. ASCII hingegen arbeitet azyklisch (auftragsorientiert). Um beide Kommunikationsarten miteinander zu verbinden, ist ein Handshake-Mechanismus notwendig. In der Übergabestruktur an das ASCII Protokoll sind dazu Synchronisationsregister enthalten.

Das folgende Bild zeigt, dass über die Signalzuordnung Daten vom ASCII Protokoll zum Protokoll an X2 und vom Protokoll an X2 nach ASCII zugeordnet werden können. In der Zuordnung sind die Synchronisationsregister als auch die eigentlichen Nutzdaten enthalten.

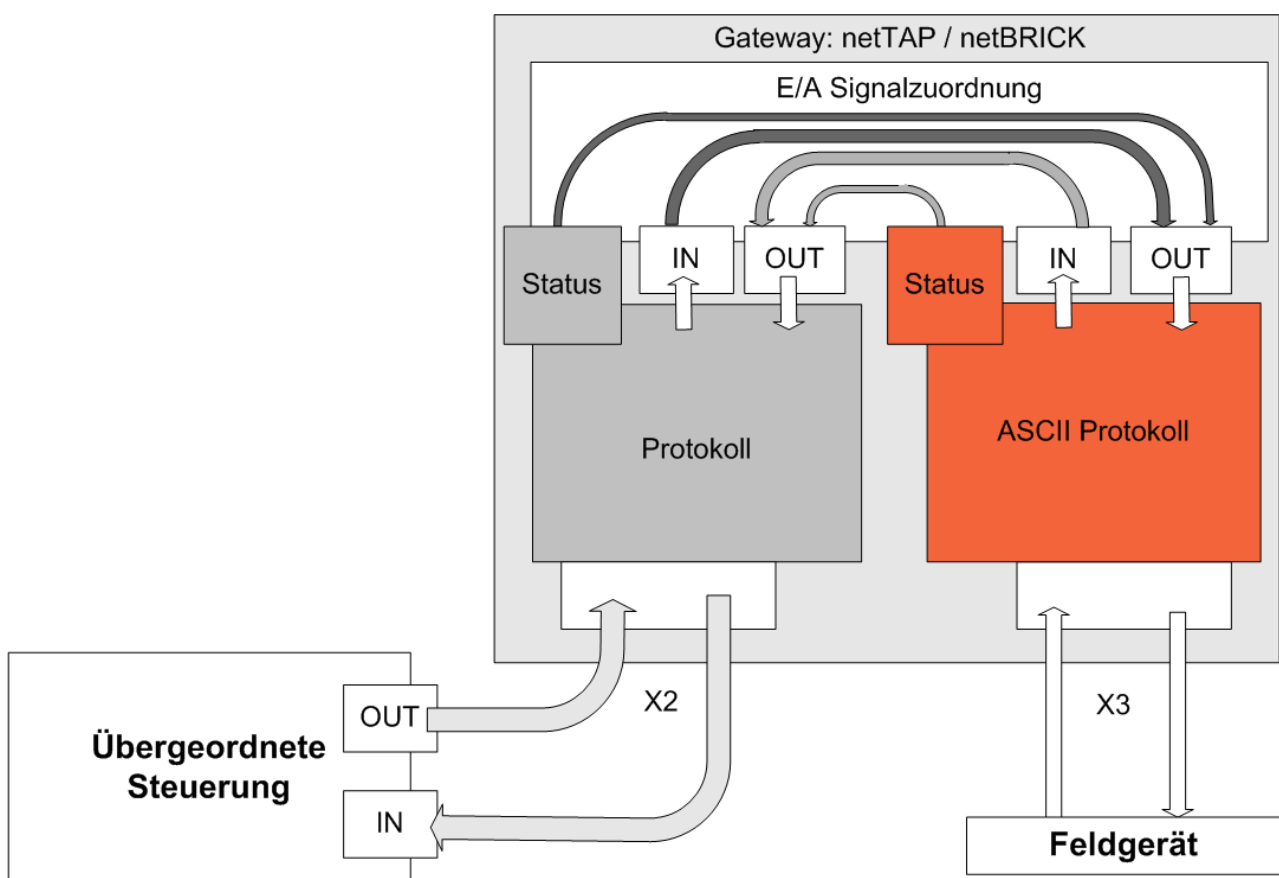


Abbildung 1: Prinzipielle Zuordnung der Daten zwischen den Protokollen

Die Zuordnung der Daten erfolgt mit der Konfigurationssoftware SYCON.net.

2.2 E/A-Datenstruktur zur Datenübergabe von und zur Steuerung

2.2.1 Ausgabedaten-Struktur - Daten von der übergeordneten Steuerung nach ASCII

Datentyp	Beschreibung	Signal für Signalzuordnung in SYCON.net
UINT32	Ausgabe-Synchronisationsregister	Application handshake flags
UINT32	Anzahl der Sendenutzdatendaten in Bytes	Byte count of OutData
Byte Array[0 ... 1023]	Ausgabe-Nutzdaten für die übergeordnete Steuerung und Sendenutzdatendaten für ASCII	OutData.UNSIGNED8_0000 OutData.UNSIGNED8_0001 ... OutData.UNSIGNED8_1023

Tabelle 4: ASCII - Transferdatenstruktur von der Steuerung

2.2.2 Eingabedaten-Struktur - Daten von ASCII zur übergeordneten Steuerung

Datentyp	Beschreibung	Signal für Signalzuordnung in SYCON.net
UINT32	Eingabe-Synchronisationsregister	Protocol handshake flags
UINT32	Anzahl der Empfangsnutzdatendaten in Bytes	Byte count of InData
UINT32	Fehlerregister zur Übergabe von Fehlerzuständen für Empfangsfehler. Die Fehlercodes sind in Abschnitt <i>Fehlercodes</i> auf Seite 19 beschrieben.	Error number in case of receive error
UINT32	Fehlerregister zur Übergabe von Fehlerzuständen für Sendefehler. Die Fehlercodes sind in Abschnitt <i>Fehlercodes</i> auf Seite 19 beschrieben.	Error number in case of transmit error
Byte Array[0 ... 1023]	Empfangsnutzdaten für ASCII und Eingabe-Nutzdaten für die übergeordnete Steuerung.	InData.UNSIGNED8_0000 InData.UNSIGNED8_0001 ... InData.UNSIGNED8_1023

Tabelle 5: ASCII - Transferdatenstruktur zur übergeordneten Steuerung

2.3 Handshake und Initialisierung der E/A-Kommunikation

Zwischen der übergeordneten Steuerung und dem ASCII Protokoll wird der Datenaustausch in einem Übergabeverfahren in den E/A-Daten organisiert.

Die Grundidee der Übergabe ist, dass es für jede Aktion in den beiden Synchronisationsregistern ein Bitpaar gibt, das zur Synchronisation dient. Ein Bit dient zur Anforderung der Aktion (CMD), das andere zur Bestätigung (ACK). Eines der beiden liegt im Ausgabe-Synchronisationsregister, eines im Eingabe-Synchronisationsregister. Es wird jeweils von jedem Kommunikationspartner auf das Eingabe-Synchronisationsregister nur lesend auf das Ausgabe-Synchronisationsregister nur schreibend zugegriffen.

Eine Aktion wird angefordert, indem ein Kommando-Bit ungleich dem Quittungs-Bit gesetzt wird. Die andere Seite bestätigt die Aktion, indem sie das entsprechende Quittungs-Bit wieder gleich dem Kommando-Bit setzt.

ASCII Telegramm senden

Ein Bitpaar wird für das Senden von ASCII Telegrammen verwendet. Die übergeordnete Steuerung steuert damit, wann ein ASCII Telegramm versendet werden soll. Die Bits dieses Bitpaares heißen APP_HS_TX_CMD und PROT_HS_TX_ACK und werden in den folgenden Abschnitten beschrieben:

- APP_HS_TX_CMD in Tabelle 7: ASCII - Synchronisationsregister zu ASCII auf Seite 13
- PROT_HS_TX_ACK in Tabelle 8: ASCII - Synchronisationsregister zur Steuerung auf Seite 14
- Die Verwendung ist beschrieben in Abschnitt Übergeordnete Steuerung in Richtung ASCII auf Seite 16

ASCII Telegramm empfangen

Ein anderes Bitpaar wird für das Empfangen von ASCII Telegrammen verwendet. Die übergeordnete Steuerung erkennt damit, dass ein ASCII Telegramm empfangen wurde. Die Bits dieses Bitpaares heißen APP_HS_RX_ACK und PROT_HS_RX_CMD und werden in den folgenden Abschnitten beschrieben:

- APP_HS_RX_ACK in Tabelle 7: ASCII - Synchronisationsregister zu ASCII auf Seite 13
- PROT_HS_RX_CMD in Tabelle 8: ASCII - Synchronisationsregister zur Steuerung auf Seite 14
- Die Verwendung ist beschrieben in Abschnitt ASCII in Richtung übergeordneter Steuerung auf Seite 17

Zusammenhang zwischen ASCII Telegramm senden/empfangen und den Betriebsarten

ASCII bietet vier Betriebsarten. Die folgende Tabelle zeigt den Zusammenhang zwischen ASCII Telegramm senden/ASCII Telegramm empfangen und den vier Betriebsarten aus Sicht der übergeordneten Steuerung:

Betriebsart	Bitpaar	Beschreibung
Modus 'Nur Senden'	APP_HS_TX_CMD und PROT_HS_TX_ACK	Übergeordnete Steuerung in Richtung ASCII auf Seite 16
Modus 'Nur Empfangen'	APP_HS_RX_ACK und PROT_HS_RX_CMD	ASCII in Richtung übergeordneter Steuerung auf Seite 17
Client Mode (erst senden, dann empfangen)	APP_HS_TX_CMD und PROT_HS_TX_ACK APP_HS_RX_ACK und PROT_HS_RX_CMD	Übergeordnete Steuerung in Richtung ASCII auf Seite 16 ASCII in Richtung übergeordneter Steuerung auf Seite 17
Server Modus (erst empfangen, dann senden)	APP_HS_RX_ACK und PROT_HS_RX_CMD APP_HS_TX_CMD und PROT_HS_TX_ACK	ASCII in Richtung übergeordneter Steuerung auf Seite 17 Übergeordnete Steuerung in Richtung ASCII auf Seite 16

Tabelle 6: Zusammenhang ASCII senden/empfangen und ASCII Betriebsart

2.3.1 Aufbau der Synchronisationsregister in den E/A-Daten

2.3.1.1 Datenrichtung von der übergeordneter Steuerung nach ASCII

Aufbau des Synchronisationsregisters der Steuerung in Richtung ASCII:

Bit-Nr.	Name und Beschreibung
0	APP_HS_TX_CMD Kommando Ausgabe-Nutzdaten von der Steuerung an ASCII senden aktivieren. Wird von ASCII automatisch geprüft, ob neue geschriebene Daten vorliegen.
1	APP_HS_RX_ACK In der Steuerung von ASCII empfangene Eingabe-Nutzdaten bestätigen. Das Bit wird aus ASCII automatisch abgefragt, ob der vorherige Schreibauftrag quittiert wurde.
2 ... 5	Nicht verwendet und reserviert
6	APP_HS_TX_ENABLE_CMD Freigabe der Ausgabe-Nutzdatenübertragung von der Steuerung an ASCII. Ist dieses Bit nicht gesetzt, wird ASCII ein über das Bit APP_HS_TX_CMD angeforderte Kommando nicht auswerten.
7	APP_HS_RX_ENABLE_CMD Freigabe der Eingabe-Nutzdatenübertragung von ASCII an die Steuerung. Ist dieses Bit nicht gesetzt, wird ASCII kein Kommando über das Bit PROT_HS_RX_CMD absetzen.
8 ... 31	Nicht verwendet und reserviert

Tabelle 7: ASCII - Synchronisationsregister zu ASCII

2.3.1.2 Datenrichtung von ASCII zur übergeordneten Steuerung

Aufbau des Synchronisationsregisters von ASCII in Richtung Steuerung:

Bit-Nr.	Name und Beschreibung
0	PROT_HS_TX_ACK Ein von der übergeordneten Steuerung aktiviertes Kommando Ausgabe-Nutzdaten wird mit diesem Bit quittiert. Es wird von ASCII automatisch bedient.
1	PROT_HS_RX_CMD Kommando Eingabe-Nutzdaten von ASCII an die übergeordnete Steuerung senden aktivieren. Es wird ASCII automatisch geprüft.
2	Nicht verwendet und reserviert
3	PROT_HS_RUN_IND ASCII meldet Bereitschaft und signalisiert das Ende seiner Initialisierung. Dieses Bit wird von ASCII unabhängig vom Steuerbit APP_HS_RX_ENABLE_CMD gesetzt.
4	PROT_HS_TX_ERROR_IND Zur Anzeige, ob ein Sendefehler aufgetreten ist. Wird von ASCII geschrieben. Das Bit wird mit dem nächsten fehlerfreien Sendetelegramm wieder auf 0 zurückgesetzt. 0: kein Sendefehler 1: Sendefehler aufgetreten. Der Fehlercode wird in „Fehlerregister zur Übergabe von Fehlerzuständen für Sendefehler“ angegeben. Die Fehlercodes sind in Abschnitt <i>Fehlercodes</i> auf Seite 19 beschrieben.
5	PROT_HS_RX_ERROR_IND Zur Anzeige, ob Empfangsfehler aufgetreten ist. Wird von ASCII geschrieben. Das Bit wird mit dem nächsten fehlerfreien Empfangstelegramm wieder auf 0 zurückgesetzt. 0: kein Empfangsfehler 1: Empfangsfehler. Der Fehlercode wird in „Fehlerregister zur Übergabe von Fehlerzuständen für Empfangsfehler“ angegeben. Die Fehlercodes sind in Abschnitt <i>Fehlercodes</i> auf Seite 19 beschrieben.
6	PROT_HS_TX_ENABLE_ACK Die Freigabe der Ausgabe-Nutzdatenübertragung von der Steuerung in Richtung ASCII wird quittiert. Das Bit wird von ASCII automatisch bedient.
7	PROT_HS_RX_ENABLE_ACK Die Freigabe der Eingabe-Nutzdatenübertragung von ASCII in Richtung der Steuerung wird quittiert. Das Bit wird von ASCII automatisch bedient.
8 ... 31	Nicht verwendet und reserviert

Tabelle 8: ASCII - Synchronisationsregister zur Steuerung

2.3.2 Initialisierung der Kommunikation

Start der Kommunikation

Schritt	Aktion: Start der Kommunikation, Initialisierung erfolgt von der übergeordneten Steuerung aus.	Handshake-Sendebyte der übergeordneten Steuerung	Handshake-Empfangsbyte der übergeordneten Steuerung
		7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
0	Speichersituation nach einem Geräte-Reset. ASCII meldet sich nicht bereit.	0 0 x x x x 0 0	0 0 0 0 0 0 0 0
1	ASCII meldet sich bereit.	0 0 x x x x 0 0	0 0 0 0 1 0 0 0
2	Die übergeordnete Steuerung eröffnet die Kommunikation mit ASCII. Durch setzen des Bits 6 und 7 wird ASCII erlaubt die Kommunikation mit der übergeordneten Steuerung aufzunehmen.	1 1 x x x x 0 0	0 0 0 0 1 0 0 0
3	ASCII empfängt die Handshakeflags der übergeordneten Steuerung. Diese lösen folgende Aktionen aus: Auf Grund von Bit 6 wird die Senderichtung zur übergeordneten Steuerung freigeschaltet. Auf Grund von Bit 7 wird die Empfangsrichtung für ASCII frei geschaltet.	1 1 x x x x 0 0	
4	Der Empfang des Handshakebytes wird von ASCII wie folgt bestätigt. Damit kann der Datenversand an die übergeordnete Steuerung beginnen.		1 1 0 0 1 0 0 0
5	Nachdem die übergeordnete Steuerung die Bestätigung von ASCII über die Sende- und Empfangsbereitschaft erhalten hat, kann auch die übergeordnete Steuerung Daten an ASCII senden.		1 1 0 0 1 0 0 0

Tabelle 9: ASCII - Kommunikationsinitialisierung



Hinweis: Nach der Freigabe der Kommunikation kann die Datenkommunikation zu jedem beliebigen Zeitpunkt starten.

2.3.3 Verarbeitungsbestätigung zwischen übergeordneter Steuerung und ASCII

Für jede gesendete Date, von ASCII an die übergeordnete Steuerung und umgekehrt, wird vom Empfänger eine Empfangsbestätigung im jeweiligen Synchronisationsregister erwartet. Solange diese Empfangsbestätigung nicht vorliegt, werden keine weiteren Daten an den Empfänger versendet.

Dieser Handshake-Ablauf wird im folgenden für beide Richtungen beschrieben.



Hinweis: In den folgenden Tabellen wird ein „x“ für eine nicht definierte Bitposition und ein „X“ für eine definierte, hier aber unbedeutende Bitposition verwendet.

2.3.3.1 Übergeordnete Steuerung in Richtung ASCII

Der Wert des mit X gekennzeichneten Bits ist für diesen Vorgang ohne Bedeutung.

Schritt	Aktion: Die übergeordnete Steuerung sendet Daten an ASCII	Handshake-Sendebyte der übergeordneten Steuerung / Empfangsbyte von ASCII	Handshake-Empfangsbyte der übergeordnete Steuerung / Sendebyte von ASCII
		7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
0	Der Datenzustand in den Handshake-Sende und Empfangspuffern sei wie nach der Kommunikationsinitialisierung. Bit 0 des Handshakebytes hat sowohl im Sende als auch im Empfangspuffer den Wert 0. → Ein Datenversand an ASCII kann eingeleitet werden.	1 1 x x x x X 0	1 1 x x 1 x X 0
1	Die übergeordnete Steuerung stellt die Sendedaten in den Sendebuffer und setzt im Handshakebyte das Bit 0 im Sendebuffer auf 1	1 1 x x x x X 1	
2	Die Daten werden an ASCII übertragen	1 1 x x x x X 1	
3	Solange eine Ungleichheit des Bits 0 des Handshakebytes im Sende und Empfangspuffer der übergeordnete Steuerung besteht, dürfen keine neuen Daten zur Versendung bereitgestellt werden.	1 1 x x x x X 1	1 1 x x 1 x X 0
4	ASCII erkennt an der Ungleichheit des Handshake-Bit 0 im Sende und Empfangspuffers, das neue Daten von der übergeordnete Steuerung eingetroffen sind.	1 1 x x x x X 1	1 1 x x 1 x X 0
5	ASCII entnimmt die von der übergeordnete Steuerung gesendeten Daten aus seinem Empfangspuffer und bestätigt diesen Vorgang durch setzen des Bit 0 des Handshakebytes in seinem Sendebuffer auf 1.		1 1 x x 1 x X 1
6	Der Zustand des Handshakebytes von ASCII wird an die übergeordnete Steuerung übertragen.		1 1 x x 1 x X 1
7	Die übergeordnete Steuerung erkennt an der Identität des Bits 0 des Handshakebytes im Sende und Empfangspuffer, dass ASCII die gesendeten Daten empfangen hat. → Es können neue Daten an ASCII übertragen werden.	1 1 x x x x X 1	1 1 x x 1 x X 1
8	Die übergeordnete Steuerung stellt neue Daten für ASCII in seinen Sendebuffer und setzt dabei das Bit 0 des Handshakebytes auf den Wert 0.	1 1 x x x x X 0	
9	Die Daten werden an ASCII übertragen.	1 1 x x x x X 0	

Schritt	Aktion: Die übergeordnete Steuerung sendet Daten an ASCII	Handshake-Sendebyte der übergeordneten Steuerung / Empfangsbyte von ASCII	Handshake-Empfangsbyte der übergeordnete Steuerung / Sendebyte von ASCII
		7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
10	Solange eine Ungleichheit im Bit 1 des Handshakebytes zwischen dem Sende und Empfangsbuffers besteht, dürfen keine weiteren Daten zur Versendung bereit gestellt werden.	1 1 x x x x x 0	1 1 x x 1 x x 1
11	ASCII erkennt an der Ungleichheit des Handshake-Bit 0 im Sende- und Empfangsbuffers, das neue Daten von der übergeordnete Steuerung eingetroffen sind.	1 1 x x x x x 0	1 1 x x 1 x x 1
12	ASCII entnimmt die Daten aus seinem Empfangsbuffer und setzt das Handshakebit 0 in seinem Sendebuffer auf den Wert des Handshake-Bits 0 in seinem Empfangsbuffer.		1 1 x x 1 x x 0
13	Die Daten des Sende-Handshakebuffers von ASCII werden an die übergeordnete Steuerung übertragen.		1 1 x x 1 x x 0
14	Die übergeordnete Steuerung erkennt an dem Wechsel des Handshake-Bits 0 in seinem Empfangsbuffers auf 0, dass ASCII die Daten Empfangen hat.		1 1 x x 1 x x 0
15	Die Handshake-Bits 0 im Sende und Empfangsbuffers haben den selben Wert. Damit ist der Ausgangszustand wie unter Schritt 0 erreicht und der Vorgang kann erneut ablaufen.	1 1 x x x x x 0	1 1 x x 1 x x 0

Tabelle 10: ASCII – Synchronisationsregister: ASCII Telegramm senden

2.3.3.2 ASCII in Richtung übergeordneter Steuerung

Der Wert eines mit x gekennzeichneten Bits ist für diesen Vorgang ohne Bedeutung.

Schritt	Aktion: ASCII sendet Daten an die übergeordnete Steuerung	Handshake-Sendebyte der übergeordneten Steuerung / Empfangsbyte von ASCII	Handshake-Empfangsbyte der übergeordneten Steuerung / Sendebyte von ASCII
		7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
0	Datenzustand vor Beginn des Sendevorgangs. Das Handshake-Bit 1 im Empfangs und im Sendebuffer hat denselben Zustand → Ein Datenversand an die übergeordnete Steuerung kann eingeleitet werden.	1 1 x x x x 0 X	1 1 x x 1 x 0 X
1	ASCII hat Daten zum Versand bereitgestellt und setzt im Handshake-Byte das Bit 1.		1 1 x x 1 x 1 X
2	Die Daten werden an die übergeordnete Steuerung übertragen.		1 1 x x 1 x 1 X
3	Solange die Ungleichheit im Bit 1 des Handshakebytes im Sende und Empfangspuffer von ASCII besteht, wird ASCII keine weiteren Daten zur Versendung an die übergeordnete Steuerung bereitstellen.	1 1 x x x x 0 X	1 1 x x 1 x 1 X
4	Die übergeordnete Steuerung erkennt an der Ungleichheit im Bit 1 des Handshake-Bytes zwischen Sende und Empfangsbuffers, dass neue Daten von ASCII eingetroffen sind.	1 1 x x x x 0 X	1 1 x x 1 x 1 X
5	Sind die Daten in der übergeordnete Steuerung angekommen und aus dem Empfangspuffer abgeholt worden, wird dieses von der übergeordnete Steuerung durch Setzen des Bit 1 in Ihrem Handshake-Sendebyte bestätigt.	1 1 x x x x 1 X	

Schritt	Aktion: ASCII sendet Daten an die übergeordnete Steuerung	Handshake-Sendebyte der übergeordneten Steuerung / Empfangsbyte von ASCII	Handshake-Empfangsbyte der übergeordneten Steuerung / Sendebyte von ASCII
		7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
6	Das Handshake-Sendebyte der übergeordneten Steuerung wird an ASCII übertragen.	1 1 x x x x 1 X	
7	ASCII erkennt, dass das Handshake-Bit 1 im Sendepuffer und Empfangspuffer den selben Wert hat. → Die übergeordnete Steuerung hat die zuvor gesendeten Daten in Empfang genommen und ist zum Empfang weiterer Daten bereit.	1 1 x x x x 1 X	1 1 x x 1 x 1 X
8	ASCII setzt bei der Einstellung neuer Sendedaten das Handshake-Bit 1 in seinem Sendebuffer auf 0.		1 1 x x 1 x 0 X
9	Die Daten werden an die übergeordnete Steuerung übertragen.		1 1 x x 1 x 0 X
10	Solange eine Ungleichheit im Bit 1 des Handshakebytes zwischen dem Sende und Empfangspuffer besteht, dürfen keine weiteren Daten zur Versendung bereit gestellt werden.	1 1 x x x x 1 X	1 1 x x 1 x 0 X
11	Die übergeordnete Steuerung erkennt an der Ungleichheit des Bit 1 in seinem Handshake-Sende- und Empfangs-Buffer, dass neue Daten von ASCII eingetroffen sind.	1 1 x x x x 1 X	1 1 x x 1 x 0 X
12	Die übergeordnete Steuerung holt die neuen Daten aus dem Empfangspuffer und setzt zur Empfangsbestätigung das Bit 1 im Handshake-Byte seines Sendepuffers auf 0 (den selben Wert wie im Empfangspuffer)	1 1 x x x x 0 X	1 1 x x 1 x 0 X
13	Die Änderung im Handshake-Byte der übergeordneten Steuerung wird an ASCII übertragen.	1 1 x x x x 0 X	1 1 x x 1 x 0 X
14	ASCII erkennt am Wechsel des Bit 1 in Handshake-Byte in seinem Empfangspuffer, dass die Daten von der übergeordnete Steuerung entgegengenommen wurden.		
15	Die Handshake-Bits 1 haben bei ASCII im Sende und Empfangbuffer den selben Wert. Damit ist für ASCII der Zustand wie im Schritt 0 gegeben und der Vorgang kann von vorne beginnen.	1 1 x x x x 0 X	1 1 x x 1 x 0 X

Tabelle 11: ASCII – Synchronisationsregister: ASCII Telegramm empfangen

3 Fehlercodes

Fehlercode	Name und Beschreibung
0000 0000 hex 0 dez	TLR_S_OK Kein Fehler
C000 0007 hex 3.221.225.479 dez	TLR_E_INVALID_PACKET_LEN Ungültige Paketlänge
C000 0180 hex 3.221.225.857 dez	TLR_E_BUS_OFF ASCII wurde von Host Applikation in den Zustand Stop gesetzt.
C07E 0001 hex 3.229.483.009 dez	TLR_E_ASCII_COMMAND_INVALID Ungültiges Kommando empfangen
C07E 0002 hex 3.229.483.010 dez	TLR_E_ASCII_STACK_DATA_SIZE_INVALID Ungültige Datenlänge
C07E 0003 hex 3.229.483.011 dez	TLR_E_ASCII_STACK_BUSY ASCII Protokoll ist beschäftigt. Die serielle Schnittstelle ist in Verwendung, d. h. ein Sende- oder Empfangsvorgang läuft.
C07E 0004 hex 3.229.483.012 dez	TLR_E_ASCII_STACK_PACKET_TOO_LONG ASCII senden: Paket zu lang ASCII empfangen: Das empfangene Telegramm ist zu lang.
C07E 0005 hex 3.229.483.013 dez	TLR_E_ASCII_STACK_DATA_OVERLAPPED ASCII empfangen: Ein weiteres Telegramm wurde über die serielle Schnittstelle empfangen, bevor das letzte Empfangstelegramm von der übergeordneten Steuerung quittiert wurde. Die vorhergehenden Daten wurden im internen Puffer des ASCII Protokolls überschrieben.
C07E 0006 hex 3.229.483.014 dez	TLR_E_ASCII_STACK_RESPONCE_TIMEOUT Antwortzeitlimit abgelaufen
C07E 0007 hex 3.229.483.015 dez	TLR_E_ASCII_STACK_WAITING_RESPONCE ASCII wartet auf Antwort, d. h. auf den Empfang von Daten auf der seriellen Schnittstelle.
C07E 0008 hex 3.229.483.016 dez	TLR_E_ASCII_STACK_LED_NOT_SUPPORTED LED nicht unterstützt
C07E 0009 hex 3.229.483.017 dez	TLR_E_ASCII_STACK_MSG_MODE_DISABLED Message-Modus ist deaktiviert

Tabelle 12: Fehlercodes ASCII

4 Anhang

4.1 Abbildungsverzeichnis

Abbildung 1: Prinzipielle Zuordnung der Daten zwischen den Protokollen	9
------------------------------------------------------------------------	---

4.2 Tabellenverzeichnis

Tabelle 1: Änderungsübersicht	3
Tabelle 2: Bezug auf Firmware	4
Tabelle 3: Bezug auf Software	4
Tabelle 4: ASCII - Transferdatenstruktur von der Steuerung	10
Tabelle 5: ASCII - Transferdatenstruktur zur übergeordneten Steuerung	10
Tabelle 6: Zusammenhang ASCII senden/empfangen und ASCII Betriebsart	12
Tabelle 7: ASCII - Synchronisationsregister zu ASCII	13
Tabelle 8: ASCII - Synchronisationsregister zur Steuerung	14
Tabelle 9: ASCII - Kommunikationsinitialisierung	15
Tabelle 10: ASCII – Synchronisationsregister: ASCII Telegramm senden	17
Tabelle 11: ASCII – Synchronisationsregister: ASCII Telegramm empfangen	18
Tabelle 12: Fehlercodes ASCII	19

4.3 Kontakte

Hauptsitz

Deutschland

Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Telefon: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

Support

Telefon: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

Niederlassungen

China

Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Telefon: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support

Telefon: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

Frankreich

Hilscher France S.a.r.l.
69500 Bron
Telefon: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support

Telefon: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

Indien

Hilscher India Pvt. Ltd.
New Delhi - 110 025
Telefon: +91 11 40515640
E-Mail: info@hilscher.in

Italien

Hilscher Italia srl
20090 Vimodrone (MI)
Telefon: +39 02 25007068
E-Mail: info@hilscher.it

Support

Telefon: +39 02 25007068
E-Mail: it.support@hilscher.com

Japan

Hilscher Japan KK
Tokyo, 160-0022
Telefon: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support

Telefon: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Korea

Hilscher Korea Inc.
Suwon, 443-734
Telefon: +82 (0) 31-695-5515
E-Mail: info@hilscher.kr

Schweiz

Hilscher Swiss GmbH
4500 Solothurn
Telefon: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support

Telefon: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA

Hilscher North America, Inc.
Lisle, IL 60532
Telefon: +1 630-505-5301
E-Mail: info@hilscher.us

Support

Telefon: +1 630-505-5301
E-Mail: us.support@hilscher.com