

	DIN IEC 62055-52 (VDE 0418-5-52)	DIN
	Diese Norm ist zugleich eine VDE-Bestimmung im Sinne von VDE 0022. Sie ist nach Durchführung des vom VDE-Präsidium beschlossenen Genehmigungsverfahrens unter der oben angeführten Nummer in das VDE-Vorschriftenwerk aufgenommen und in der „etz Elektrotechnik + Automation“ bekannt gegeben worden.	VDE
ICS 91.140.50 Einsprüche bis 2006-08-31 Entwurf Messung der elektrischen Energie – Zählersysteme mit Inkassofunktion – Teil 52: Standardtransferspezifikation – Protokoll der Bitübertragungsschicht für einen virtuellen Zweiweg-Tokenträger für die direkte lokale Verbindung (IEC 13/1371/CD:2006) Electricity metering – Payment systems – Part 52: Standard transfer Specification – Physical Layer Protocol for a two-way virtual token carrier for direct local connection (IEC 13/1371/CD:2006) Compteurs d'énergie électrique – Systèmes à paiement – Partie 52: Spécification de transfert standard – Protocole de couche physique pour un porteur identifiant virtuel à double voie pour la connexion directe locale (CEI 13/1371/CD:2006) Anwendungswarnvermerk Dieser Norm-Entwurf wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt. Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfes besonders zu vereinbaren. Stellungnahmen werden erbeten – vorzugsweise als Datei per E-Mail an dke@vde.com in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.dke.de/stellungnahme abgerufen werden – oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE, Stresemannallee 15, 60596 Frankfurt am Main. Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevante Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen. Gesamtumfang 96 Seiten DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE		

– Entwurf –

E DIN IEC 62055-52 (VDE 0418-5-52):2006-07

Beginn der Gültigkeit

Diese Norm gilt ab ...

Nationales Vorwort

Das internationale Dokument IEC 13/1371/CD:2006 „Electricity metering – Payment systems – Part 52: Standard transfer Specification – Physical Layer Protocol for a two-way virtual token carrier for direct local connection“ (CD, en: Committee Draft) ist unverändert in diesen Norm-Entwurf übernommen worden. Dieser Norm-Entwurf enthält eine noch nicht autorisierte deutsche Übersetzung.

Um Zweifelsfälle in der Übersetzung auszuschließen, ist die englische Originalfassung des CD entsprechend der diesbezüglich durch die IEC erteilten Erlaubnis beigefügt. Die Nutzungsbedingungen für den deutschen Text des Norm-Entwurfes gelten gleichermaßen auch für den englischen IEC-Text.

Das internationale Dokument wurde vom TC 13 „Equipment for electrical energy measurement and load control“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet und den nationalen Komitees zur Stellungnahme vorgelegt.

Die IEC und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Da der Abstimmungszeitraum für einen FDIS bzw. Schluss-Entwurf prEN nur 2 Monate beträgt, und dann keine sachlichen Stellungnahmen mehr abgegeben werden können, sondern nur noch eine „JA/NEIN“-Entscheidung möglich ist, wobei eine „NEIN“-Entscheidung fundiert begründet werden muss, wird bereits der CD als DIN-Norm-Entwurf veröffentlicht, um die Stellungnahmen aus der Öffentlichkeit frühzeitig berücksichtigen zu können.

Für diesen Norm-Entwurf ist das nationale Arbeitsgremium K 461 „Elektrizitätszähler“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (<http://www.dke.de>) zuständig.

Die IEC weist darauf hin, dass die Einhaltung dieser Internationalen Norm hinsichtlich der Fortschreibung für die Verschlüsselungscodeverwaltung und die Ablage der Protokolle, auf denen die aktuelle Norm IEC 62055-41 beruht, Patentrechte berühren können. [Siehe IEC 62055-41, Anhang C.1.] Die IEC nimmt keine Stellung zur Rechtmäßigkeit, zur Gültigkeit und zum Aufgabengebiet dieser Wartungsdienste.

Der Anbieter dieser Wartungsdienste hat der IEC zugesichert, dass er bereit ist, über Dienstleistungen zu vernünftigen und nicht diskriminierenden Geschäftsbedingungen mit Antragstellern in der ganzen Welt zu verhandeln. In diesem Zusammenhang ist die Erklärung des Anbieters dieser Wartungsdienste bei der IEC registriert. Informationen sind erhältlich bei:

Anschrift: The STS Association, P.O. Box 868, Ferndale 2160, Republic of South Africa

Tel: +27 11 789 1384

Fax: +27 11 789 1385

Email: email@sts.org.za

Website: <http://www.sts.org.za>

Die Internationale Norm IEC 62055-52 wurde von der Arbeitsgruppe 15 des Technischen Komitees 13 „Equipment for electrical energy measurement and load control“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Der Text dieser Norm basiert auf folgenden Schriftstücken:

FDIS	Abstimmungsbericht
XX/XX/FDIS	XX/XX/RVD

Vollständige Informationen über die Abstimmung zur Annahme dieser Norm können dem in vorstehender Tabelle angeführten Abstimmungsbericht entnommen werden.

Die vorliegende Publikation wurde nach den ISO/IEC-Richtlinien, Teil 2, erarbeitet.

Das IEC-Komitee hat entschieden, dass der Inhalt dieser Publikation bis zu dem auf der IEC-Website unter „<http://webstore.iec.ch>“ mit den Daten zu dieser Publikation angegebenen Datum¹⁾ (maintenance result date) unverändert bleiben soll. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees die Publikation

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

Das Protokoll in dieser Internationalen Norm beruht auf dem Übertragungsprotokoll nach IEC 62056-21 und wurde durch Herausnahme einiger Einzelheiten aus dem Protokoll nach IEC 62056-21 vereinfacht, die für die gegenwärtigen Anforderungen an den Datenaustausch zwischen dem VTC07-Client und dem Inkassozähler-Server nicht erforderlich sind.

IEC 62055-52 ist ein ergänzendes Schriftstück zu IEC 62055-41 und sollte im Zusammenhang damit gelesen werden.

Die Normenreihe IEC 62055 behandelt Zählersysteme mit Inkassofunktion, umfasst die Kundeninformationssysteme, Kassensysteme, Tokenträger, Inkassozähler und die entsprechenden Schnittstellen, die zwischen diesen Entitäten existieren. Zum Zeitpunkt der Erarbeitung dieser Norm bestand IEC 62055 mit dem allgemeinen Titel *Messung der elektrischen Energie – Zählersysteme mit Inkassofunktion* aus den folgenden Teilen:

Teil 21: *Aufbau für die Normung*

Teil 31: *Besondere Anforderungen – Inkasso-Wirkverbrauchszähler (Genauigkeitsklassen 1 und 2)*

Teil 41: *Standardtransferspezifikation – Protokoll der Anwendungsschicht für Einweg-Tokenträgersysteme*

Teil 51: *Standardtransferspezifikation – Protokoll der Bitübertragungsschicht für Einweg-Tokenträger auf Magnetkarten und als Zahlwort*

Teil 52: *Standardtransferspezifikation – Protokoll der Bitübertragungsschicht für einen virtuellen Zweiweg-Tokenträger für die direkte lokale Verbindung*

In der Reihe *Teil 4x* werden Protokolle der Anwendungsschicht und in der Reihe *Teil 5x* werden Protokolle der Bitübertragungsschicht festgelegt.

Die STS-Association ist für die Erarbeitung von Normen für den Bereich der Standardtransferspezifikation eine Typ-D-Verbindung mit IEC/TC 13/WG 15 eingegangen und ist somit für die Fortschreibung aller derartiger IEC-Normen verantwortlich, die im Ergebnis dieser Verbindung erarbeitet werden könnten.

Die STS-Association ist bei der IEC als Registrierungsstelle für die Bereitstellung von Wartungsdiensten zur Unterstützung der STS eingetragen (nähere Angaben sind in IEC 62055-41, C.1, enthalten).

¹⁾ Die Nationalen Komitees werden gebeten zu beachten, dass für diese Publikation das Datum gilt.

– Entwurf –

E DIN IEC 62055-52 (VDE 0418-5-52):2006-07

Nationaler Anhang NA (informativ)

Zusammenhang mit Europäischen und Internationalen Normen

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Eine Information über den Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ist nachstehend wiedergegeben.

Tabelle NA.1

Europäische Norm	Internationale Norm	Deutsche Norm	Klassifikation im VDE-Vorschriftenwerk
–	IEC 60050-300	–	–
–	IEC 62051:1999 ersetzt durch IEC/TR 62051-1:2004	–	–
–	IEC 62055-21:2005	–	–
EN 62055-31:2005	IEC 62055-31:2005	DIN EN 62055-31 (VDE 0418-5-31):2006-08	VDE 0418-5-31
–	IEC 13/1333A/CD	E DIN IEC 62055-41 (VDE 0418-5-41):2005-06	VDE 0418-5-41
–	IEC 13/1334A/CD	E DIN IEC 62055-51 (VDE 0418-5-51):2005-06	VDE 0418-5-51
EN 62056-21:2002	IEC 62056-21:2002	DIN EN 62056-21:2003-01	–
–	ISO/IEC 646	–	–
EN ISO/IEC 7498-1:1995	ISO/IEC 7498-1:1994	DIN EN ISO/IEC 7498-1:1995-12	–
–	STS 15-001:200X	–	–
–	STS 18-001:200X	–	–

Nationaler Anhang NB (informativ)

Literaturhinweise

DIN EN 62055-31 (VDE 0418-5-31), *Messung der elektrischen Energie – Zählersysteme mit Inkassofunktion – Teil 31: Besondere Anforderungen – Elektronische Inkasso-Wirkverbrauchsähler (Klassen 1 und 2)*

E DIN IEC 62055-41 (VDE 0418-5-41), *Messung der elektrischen Energie – Zählersysteme mit Inkassofunktion – Teil 41: Standardtransferspezifikation – Protokoll der Anwendungsschicht für Einweg-Tokenträgersysteme*

— Entwurf —

E DIN IEC 62055-52 (VDE 0418-5-52):2006-07

E DIN IEC 62055-51 (VDE 0418-5-51), *Messung der elektrischen Energie – Zählersysteme mit Inkassofunktion – Teil 51: Standardtransferspezifikation – Protokoll der Bitübertragungsschicht für Einweg-Zahlwort- und Magnetkarten-Tokenträger*

DIN EN 62056-21, *Messung der elektrischen Energie – Zählerstandsübertragung, Tarif- und Laststeuerung – Teil 21: Datenübertragung für festen und mobilen Anschluss*

DIN EN ISO/IEC 7498-1, *Informationstechnik – Kommunikation Offener Systeme – Teil 1: Basis-Referenzmodell: Basismodell*

**Messung der elektrischen Energie – Zählersysteme mit Inkassofunktion – Teil 52:
Standardtransferspezifikation – Protokoll der Bitübertragungsschicht für einen
virtuellen Zweiweg-Tokenträger für die direkte lokale Verbindung**

Inhalt

	Seite
Einleitung.....	9
1 Anwendungsbereich	10
2 Normative Verweisungen.....	10
3 Begriffe und Abkürzungen	11
3.1 Begriffe.....	11
3.2 Abkürzungen.....	11
3.3 Notation und Terminologie.....	12
3.4 Nummerierungsvereinbarungen	12
4 Protokoll-Referenzmodell	13
5 POS_to-Token_Carrier_Interface: Protokoll der Bitübertragungsschicht.....	14
6 Token_Carrier_to_Meter_Interface: Protokoll der Bitübertragungsschicht	14
6.1 TCDU	14
6.1.1 Allgemeines	14
6.1.2 Token_Data	14
6.1.3 Authentication_Result.....	14
6.1.4 Validation_Result.....	15
6.1.5 Token_Result.....	15
6.2 Physikalische Verbindung und Signalschnittstellen.....	15
6.2.1 Alternativen für Schnittstellen	15
6.2.2 Alternative 1: Optische Schnittstelle	15
6.2.3 Alternative 2: Linienstrom-Schnittstelle	15
6.2.4 Alternative 3: Spannungsschnittstelle.....	15
6.3 Zeichenübertragung.....	15
6.3.1 Übertragungsart.....	15
6.3.2 Übertragungsformat.....	15
6.3.3 Übertragungsgeschwindigkeit.....	16
6.3.4 Zeichencodierung	16
6.4 Festlegungen der Nachrichtensyntax	17
6.4.1 Allgemeines	17
6.4.2 Nachricht ID_Request.....	17
6.4.3 Nachricht ID_Response.....	18
6.4.4 Nachricht Read_Command	18
6.4.5 Nachricht Write_Command.....	18

	Seite
6.4.6 Nachricht Break_Command	18
6.4.7 ACK: Nachricht Acknowledge	18
6.4.8 NAK: Nachricht Negative_Acknowledge	19
6.4.9 Nachricht Data	19
6.5 Festlegungen der Nachrichtfelder	19
6.6 Funktionen des Protokolls der Bitübertragungsschicht	21
6.6.1 Flussdiagramm des Serverprotokolls	21
6.6.2 Funktion ID_Request_Processing	22
6.6.3 Funktion Read_Command_Processing	23
6.6.4 Funktion Write_Command_Processing	24
6.6.5 Funktion Break_Command_Processing	25
6.6.6 Funktion Token_Lockout	26
6.7 Anforderungen an die Zeitsteuerung des Servers	27
6.7.1 Zeitliche Steuerung der Nachrichten- und Zeichenzwischenräume	27
6.7.2 Zeitbezogene Übertragungsfehler	28
6.7.3 Zeitliche Steuerung der Nachrichtenausführung	28
6.8 Register_Table	29
6.8.1 Register_Table-Schnittstellenklasse	29
6.8.2 Registerschnittstellenklasse	30
6.8.3 Vordefinierte Register und Register_ID-Werte	31
6.9 Begleitspezifikationen und Instantiierungen der Register_Table	41
7 Fortschreibung der STS-Entitäten und zugehörigen Dienste	41
7.1 Allgemeines	41
7.2 Fortschreibung der Register_Table	41
7.3 Fortschreibung der Register	42
7.4 Fortschreibung der Table_ID	42
7.5 Fortschreibung der FOIN	42
7.6 Fortschreibung der Protocol_Version	42
7.7 Fortschreibung des Server_Status	42
7.8 Fortschreibung des Token_Status	43
7.9 Fortschreibung der Software_Version	43
Anhang A (normativ) Zustandsdiagramme des Servers für die Verarbeitung von Anforderungsnachrichten	44
Literaturhinweise	48
 Bild 1 – Bitübertragungsschichten des STS-Protokollstapels	13
Bild 2 – Zeichenübertragungsformat	15
Bild 3 – Flussdiagramm des Serverprotokolls	21
Bild 4 – Zeitliches Verhalten für die Nachrichtenzwischenräume	27
Bild 5 – Zeitbezogene Übertragungsfehler	28

– Entwurf –

E DIN IEC 62055-52 (VDE 0418-5-52):2006-07

	Seite
Bild A.1 – Zustandsdiagramm des Servers für die Funktion ID_Request_Processing.....	44
Bild A.2 – Zustandsdiagramm des Servers für die Funktion Read_Command_Processing	45
Bild A.3 – Zustandsdiagramm des Servers für die Funktion Write_Command_Processing	46
Bild A.4 – Zustandsdiagramm des Servers für die Funktion Break_Command_Processing	47
Tabelle 1 – Datenelemente in der TCDU	14
Tabelle 2 – Bitcodierung eines 7-Bit-Zeichencode	16
Tabelle 3 – Beispiel für die Zeichencodierung einer 14-Bit-Binärzahl	16
Tabelle 4 – Beispiel für die Zeichencodierung einer Hexadezimalzahl mit 4 Ziffern	17
Tabelle 5 – Beispiel für die Zeichencodierung einer Dezimalzahl mit 4 Ziffern	17
Tabelle 6 – Festlegungen der Nachrichtenfelder	19
Tabelle 7 – Vom Server unterstützte Anforderungsnachrichten	22
Tabelle 8 – Vom Server unterstützte Antwortnachrichten	22
Tabelle 9 – Vom Server unterstützte Funktionen.....	22
Tabelle 10 – Anforderungen an die Zeitsteuerung des Servers für die Antwort auf eine Anforderungsnachricht.....	27
Tabelle 11 – Anforderungen an die Zeitsteuerung für die Zeichenzwischenräume	27
Tabelle 12 – Rückgewinnungswartedauer bei Übertragungsfehler	28
Tabelle 13 – Vordefinierte Register und Register_ID-Werte	31
Tabelle 14 – Definierte Werte für Protocol_Version.....	32
Tabelle 15 – Festgelegte Werte für Server_Status.....	35
Tabelle 16 – Definierte Werte für Token_Status.....	39
Tabelle 17 – Entitäten/Dienste, die einen Fortschreibungsdienst erfordern.....	41

Einleitung

Die STS (Standardtransferspezifikation) ist ein gesichertes Nachrichtenprotokoll, welches die Übertragung von Informationen zwischen der Einrichtung des Kassenplatzes (POS) und den Inkassozählern zulässt und berücksichtigt mehrere Nachrichtentypen, wie Guthaben, Konfigurationssteuerung, Anzeige- und Prüfanweisungen. Ferner werden Einrichtungen und Regeln festgelegt, die eine sichere Verwaltung (Generierung, Speicherung, Wiedergewinnung und Transport) der innerhalb des Systems verwendeten kryptografischen Schlüssel ermöglicht.

Das nationale Elektrizitätsversorgungsunternehmen in Südafrika (Eskom) entwickelte zuerst und veröffentlichte 1993 die Standardtransferspezifikation und übertrug 1998 die Eigentumsrechte an die STS-Association zur Verwaltung und Weiterentwicklung.

Vor der Entwicklung der STS wurden eine Vielzahl von firmenspezifischen Inkassozählern und Verkaufssystemen entwickelt, die untereinander jedoch nicht kompatibel waren. Bei den Hauptanwendern entwickelte sich daraus ein bestimmter Bedarf, vereinheitlichte Lösungen anzustreben, indem die Betriebsprobleme angesprochen wurden, die bei den unterschiedlichen Typen der Inkassozähler und Kassenplätze gleichzeitig zu bearbeiten sind. Es wurde eine Standardtransferspezifikation (STS) entwickelt, die die Anwendung und Zusammenarbeit der Inkassozähler und Verkaufseinrichtungen von mehreren Herstellern in einer Installation aus Vorkassezählern und Verkaufseinrichtung zulassen sollte.

Der Token_Carrier (Tokenträger) ist das physikalische Medium, welches zur Übertragung der Information von einem Verkaufsort oder dem Verwaltungssystem zum Inkassozähler oder vom Inkassozähler zum Verkaufsort oder zum Verwaltungssystem verwendet wird. In diesem Teil von IEC 62055 wird ein virtueller Tokenträger festgelegt, wie er sich in einer direkten lokalen Verbindung zwischen einer Verwaltungseinrichtung (Client) und einem Inkassozähler-Server darstellt. Von der STS Association wurde der Kennzeichnungscode 07 zugewiesen und er wird allgemein als VTC07 bezeichnet. Die Nationalen Komitees oder die STS Association dürfen neue Tokenträger als neue Arbeitsthemen vorschlagen.

Das hauptsächliche Entwurfsziel bei der Festlegung des Protokolls war die Anforderung zur Verringerung der Komplexität der Software, die zur Umsetzung des Protokolls im Inkassozähler erforderlich ist. Dies bezieht sich direkt auf eine kleinere Speichergröße, was als Kosteneinsparung oder Fähigkeit zur Aufnahme neuer Softwareeinzelheiten bei einer bestimmten Speichergröße betrachtet werden kann.

Obwohl die hauptsächliche Realisierung der STS in der Elektroversorgungsindustrie liegt, ist sie ebenso für die Verwaltung von anderen Versorgungsdienstleistungen, wie Wasser und Gas geeignet. Zukünftige Überarbeitungen der STS können andere Tokenträgertechnologien, wie Smartcards und Speicherschlüssel mit Zweiweg-Funktionalität zulassen und eine Echtzeituhr und komplexe Tarife im Inkassozähler berücksichtigen.

1 Anwendungsbereich

In diesem Teil von IEC 62055 wird ein Protokoll der Bitübertragungsschicht der Standardtransferspezifikation für die Übertragung von Guthabeneinheiten und weiterer Verwaltungsinformationen zwischen einem Client (üblicherweise eine tragbare Einheit) und einem Server (ein STS-konformer Inkasso-Elektrizitätszähler) festgelegt, die üblicherweise über eine lokale Verbindung erfolgt. Dieser Teil ergänzt das Protokoll der Anwendungsschicht, das in IEC 62055-41 festgelegt wurde, und sollte gemeinsam mit dieser Norm gelesen werden.

Seine Anwendung ist für eine Reihe von Zählern vorgesehen, die von verschiedenen Herstellern entwickelt worden sind, sowie zur Sicherung der Kompatibilität zwischen diesen Produkten und anderen Anfrageeinrichtungen.

Festgelegt wird ein Client/Server-Übertragungsprotokoll, mit dem:

- STS-konforme Token von einer Client-Einrichtung an einen Inkassozähler-Server übertragen werden;
- nach der Tokenübertragung und -ausführung das Ergebnis aus dem Inkassozähler ausgelesen wird;
- Verwaltungsdaten von der Client-Einrichtung an den Inkassozähler-Server übertragen werden;
- Verwaltungsdaten vom Inkassozähler-Server gelesen und diese gleich an die Client-Einrichtung übertragen werden.

ANMERKUNG Obwohl sie für Inkassosysteme in der Elektrizitätsversorgung entwickelt worden ist, enthält diese Norm auch Bestimmungen für Token, die bei anderen Versorgungsdienstleistungen, wie Wasser und Gas, angewendet werden.

2 Normative Verweisungen

Die folgenden zitierten Dokumente sind für die Anwendung dieses Dokuments erforderlich. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

IEC 60050-300, *International Electrotechnical Vocabulary (IEV) – Electrical and electronic measurements and measuring instruments – Part 311: General terms relating to measurements – Part 312: General terms relating to electrical measurements – Part 313: Types of electrical measuring instruments – Part 314: Specific terms according to the type of instrument*

IEC 62051:1999, *Electricity metering – Glossary of terms*

IEC 62055-31:2005, *Electricity metering – Payment systems – Part 31: Particular requirements – Static payment meters for active energy (classes 1 and 2)*

IEC 62055-41, *Electricity metering – Payment systems – Part 41: Standard Transfer Specification – Application Layer Protocol for one-way token carrier systems²⁾*

IEC 62055-51, *Electricity metering – Payment systems – Part 51: Standard Transfer Specification – Physical Layer Protocol for one-way numeric and magnetic card token carriers³⁾*

IEC 62056-21:2002, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 21: Direct local data exchange*

ISO/IEC 646, *7-Bit coded character set for information interchange*

STS 18-001:200x, *Standard Transfer Specification – Companion Specification – Token carriers: Mechanical and electrical interface for application with IEC 62055-52⁴⁾*

²⁾ Zu veröffentlichen.

³⁾ Zu veröffentlichen.

⁴⁾ Zu veröffentlichen.

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieser Internationalen Norm gelten die Begriffe aus IEC 60050-300, IEC 62051, IEC 62055-31, IEC 62055-41 und die folgenden.

Wenn es einen Unterschied zwischen den Begriffen in dieser Internationalen Norm und denjenigen in den zitierten IEC-Normen gibt, dann ist denen in dieser Internationalen Norm aufgeführten Begriffen der Vorrang zu geben.

Der in dieser Norm gebrauchte Begriff ASCII steht für den mit 7 Bit codierten Zeichensatz, wie er in ISO/IEC 646 definiert ist.

3.2 Abkürzungen

ACK (Acknowledge) [ASCII]
positive Rückmeldung

APDU (Application_Protocol_Data_Unit)
Dateneinheit des Anwendungsprotokolls

ASCII (American Standard Code for Information Interchange)
amerikanischer Standardcode für den Informationsaustausch

BCC (Block Check Character)
Blockprüfzeichen

Char (Character)
Zeichen

CR (Carriage Return) [ASCII]
Wagenrücklauf

DL (Data Length)
Datenlänge

DCTK (Decoder_Common_Transfer_Key)
einheitlicher Transferschlüssel für Decoder

ETX (End Of Text) [ASCII]
Endemarkierung

FOIN (Function_Object_Identification_Number)
Kennzeichnungsnummer des Funktionsobjektes

GPRS (General Packet Radio Service)
Allgemeiner Paketfunkdienst

GSM (Global System For Mobile Communication)
Europäisches System für digitale Mobilfunkdienste

hex (hexadecimal)
hexadezimaler Zahlensystem

ID (Identification, Identifier)
Kennzeichnung, Kennung

ISDN (Integrated Services Digital Network)
Diensteintegrierendes Digitalnetz

ISO (International Standards Organisation)
Internationale Normungsorganisation

LAN (Local Area Network)
lokales Netz

LF (Line Feed)
Zeilenvorschub

E DIN IEC 62055-52 (VDE 0418-5-52):2006-07

ms (Milli-second)
Millisekunde

NAK (Negative Acknowledgement) [ASCII]
negative Rückmeldung

PLC (Power Line Carrier)
Träger auf Hochspannungsleitungen

POS (Point_Of_Sale)
Kassenplatz

PSTN (Public Switched Telephone Network)
öffentliches Fernsprechwählnetz

Ref (Reference clause)
Bezugsabschnitt

RID (Register Identifier Code)
Registerkennungscode

SOH (Start Of Header) [ASCII]
Anfang des Kopfes

STS (Standard Transfer Specification)
Standardtransferspezifikation

STX (Start Of Text) [ASCII]
Startmarkierung

TCDU (Token_Carrier_Data_Unit)
Dateneinheit des Tokenträgers

VTC07 (Virtual-Token_Carrier_Type_07)
virtueller Tokenträger Typ 07

WAN (Wide Area Network)
Weitverkehrsnetz

3.3 Notation und Terminologie

In dieser Internationalen Norm gelten die folgenden Regeln für Benennungsvereinbarungen:

- Entitätennamen, Datenelementnamen, Namen für Funktionen und Prozesse werden als übergeordnete Objektklassen behandelt und erhalten Namen in Hinsicht auf Substantive, die mit einem Großbuchstaben beginnen und mit einem Unterstrich verbunden werden, um sie als einzelne Entität kenntlich zu machen. Beispiele sind: `Supply_Group_Code` als ein Datenelementname und `Credit_Transfer` als Name für einen Prozess.
- Direkte (spezifische) Verweisung auf eine benannte Objektklasse verwendet die Form mit Großbuchstaben, während eine allgemeine (nicht-spezifische) Verweisung Kleinbuchstaben ohne verbindenden Unterstrich verwendet. Ein Beispiel für eine direkte Verweisung ist: „Der `Supply_Group_Code` steht mit einer Gruppe von Inkassozählern in Verbindung“, während ein Beispiel für eine indirekte Verweisung ist: „Ein Versorgungsgruppencode steht im Zusammenhang mit einem Verkaufsschlüssel“.
- Für weitere Begriffe werden die allgemein akzeptierten Formen, wie `PSTN` für `Public Switched Telephone Network` verwendet.

3.4 Nummerierungsvereinbarungen

In dieser Norm erfolgt die Darstellung der Zahlen in Binärzeichenfolgen nach der Vereinbarung, dass das niederwertigste Bit rechts und das hochwertigste Bit links steht.

Die Nummerierung der Binärstellen beginnt an der Binärstelle 0, die dem niederwertigsten Bit in einer Binärzahl entspricht.

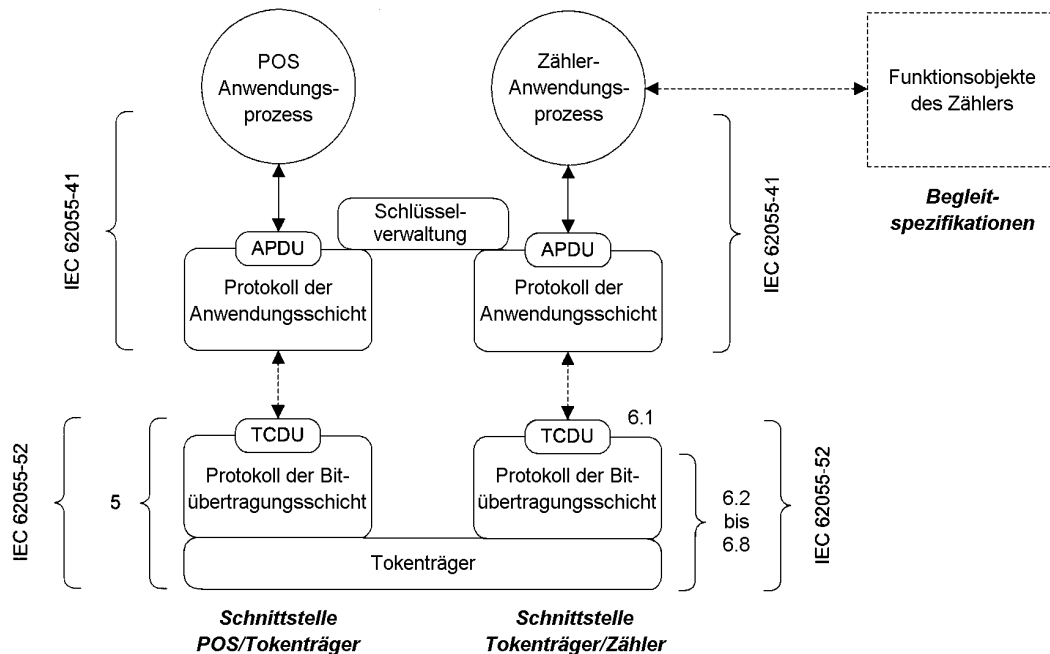
Zahlen werden üblicherweise im Dezimalformat angegeben, sofern nicht anders dargestellt. Jede Ziffer ohne besonderen Hinweis bedeutet Dezimalformat.

Die Werte von Binärziffern sind 0 oder 1.

Die Werte von Dezimalziffern reichen von 0 bis 9.

Die Werte von Hexadezimalziffern reichen von 0 bis 9 und A bis F und werden üblicherweise mit einem „hex“ gekennzeichnet.

4 Protokoll-Referenzmodell



Legende:

APDU Dateneinheit des Anwendungsprotokolls; Datenschnittstelle zum Protokoll der Anwendungsschicht

TCDU Dateneinheit des Tokenträgers; Datenschnittstelle zum Protokoll der Bitübertragungsschicht

Die Verweisungen auf die betreffenden Abschnitsnummern in dieser Internationalen Norm sind neben jedem Kästchen angegeben.

Bild 1 – Bitübertragungsschichten des STS-Protokollstapels

Die STS ist ein gesichertes Datenübertragungsprotokoll zwischen einem POS und einem Inkassozähler mit einem Tokenträger als Übertragungsmedium. Das Protokoll der Anwendungsschicht befasst sich mit den Token, den Verschlüsselungsprozessen und Funktionen und ist in IEC 62055-41 festgelegt, während sich das Protokoll der Bitübertragungsschicht mit der eigentlichen Codierung der Tokendaten auf verschiedene Typen der Tokenträger befasst.

In diesem Teil von IEC 62055 wird das Protokoll der Bitübertragungsschicht festgelegt, welches sich mit der eigentlichen Codierung der Tokendaten auf einem virtuellen Tokenträger befasst und welches aus einer direkten lokalen Verbindung zwischen einem Client und einem Server (Inkassozähler) besteht, die ein serielles Übertragungsprotokoll anwendet und in Verbindung mit dem in IEC 62055-41 festgelegten Protokoll der Anwendungsschicht arbeitet (siehe Bild 1).

Beispiele für andere Arten virtueller Tokenträger sind: PSTN-Modem, ISDN-Modem, GSM-Modem, GPRS-Modem, Funkmodem, PLC-Modem, Infrarot-, LAN- und WAN-Verbindungen und die direkte lokale Verbindung, die zukünftig in weiteren Teilen der Normenreihe IEC 62055-5x festgelegt werden kann.

E DIN IEC 62055-52 (VDE 0418-5-52):2006-07

IEC 62055-41, Abschnitt 5, enthält eine umfangreichere Beschreibung des STS-Referenzmodells und der Datenflüsse vom POS_Application_Process (Kassenplatz-Anwendungsprozess) zum Meter_Application_Process (Zähleranwendungsprozess).

Durch das Protokoll wird eine allgemeine Nachrichtenstruktur für das Schreiben und Lesen festgelegt, die es einem Client ermöglicht, Daten von einem Inkassozähler zu lesen oder an ihn zu schreiben, indem er auf eine virtuelle Registertabelle als eine logische Schnittstelle zu den eigentlichen Registern oder Funktionen Bezug nimmt. In dieser Norm wird eine Register_Table-Schnittstellenklasse (siehe 6.8.1) und eine Register-Schnittstellenklasse (siehe 6.8.2) für ein Meter_Function_Object (Funktionsobjekt des Zählers) festgelegt, welches in einer Begleitspezifikation bestimmt wird. Die Begleitspezifikationen sind für IEC 62055-52 nicht normativ und werden von der STS-Association verwaltet (siehe 6.9).

5 POS_to-Token_Carrier_Interface: Protokoll der Bitübertragungsschicht

Die Client-Schnittstelle zum virtuellen Tokenträger wird in dieser Norm nicht im Einzelnen festgelegt, aber sie muss allgemein die Anforderungen ergänzen, die in den betreffenden Teilen des Abschnitts 6 angegeben sind.

In der Praxis ist die Client-Einrichtung üblicherweise eine tragbare Einheit, die mit einer direkten lokalen Verbindung an den Inkassozähler angeschlossen wird, jedoch ist es für die Verbindung auch möglich, dass sie mit geeigneten zwischengeschalteten Modems, die über jedes geeignete Übertragungsmedium verknüpft sind, zu einem abgesetzten Verwaltungssystem ausgedehnt werden kann. Eine derartige ausgedehnte Fernverbindung wird in dieser Norm nicht ausdrücklich behandelt, aber im Wesentlichen bedeutet dies einfach eine Verlängerung des physikalischen Mediums.

6 Token_Carrier_to_Meter_Interface: Protokoll der Bitübertragungsschicht

6.1 TCDU

6.1.1 Allgemeines

Die TCDU ist die Datenschnittstelle zwischen dem Protokoll der Bitübertragungsschicht und dem Protokoll der Anwendungsschicht und besteht aus den folgenden Datenelementen:

Tabelle 1 – Datenelemente in der TCDU

Element	Format	Bezug
Token_Data	66 Bit binär	6.1.2
Authentication_Result	Boolesch	6.1.3
Validation_Result	Boolesch	6.1.4
Token_Result	Boolesch	6.1.5

6.1.2 Token_Data

Dies ist das 66-Bit-Binärformat der Tokendaten wie sie aus Token_Carrier entschlüsselt wurden. Es ist dasselbe Datenelement, wie es für die TCDU am POS_to-Token_Carrier_Interface (POS-Tokenträger-Schnittstelle) dargestellt wurde (siehe IEC 62055-4.1, 6.4.3 bis 6.4.5).

6.1.3 Authentication_Result

Dies ist ein Statusanzeiger für das Protokoll der Bitübertragungsschicht, um das Ergebnis der ersten Authentifikationsprüfungen mitzuteilen. Zur Festlegung der Werte von Authentication_Result (Ergebnis der Authentifikationsprüfung) siehe auch IEC 62055-41, 7.1.3.

Diese Datenelemente sind auch im Register Token_Status (siehe 6.8.3.7) für den Zugriff durch den Client enthalten.

6.1.4 Validation_Result

Dies ist ein Statusanzeiger für das Protokoll der Bitübertragungsschicht, um das Ergebnis der ersten Authentifikationsprüfungen mitzuteilen. Zur Festlegung der Werte von Validation_Result (Ergebnis der Berechtigungsprüfung) siehe auch IEC 62055-41, 7.1.4.

Diese Datenelemente sind auch im Register Token_Status (siehe 6.8.3.7) für den Zugriff durch den Client enthalten.

6.1.5 Token_Result

Dies ist ein Statusanzeiger aus dem Meter_Application_Process, um das Ergebnis nach der Verarbeitung des Token mitzuteilen, damit das Protokoll der Bitübertragungsschicht die entsprechenden Aktionen ausführen kann. Zur Festlegung der Werte von Token_Result (Ergebnis der Tokenverarbeitung) siehe auch IEC 62055-41, 7.1.5.

Diese Datenelemente sind auch im Register Token_Status (siehe 6.8.3.7) für den Zugriff durch den Client enthalten.

6.2 Physikalische Verbindung und Signalschnittstellen

6.2.1 Alternativen für Schnittstellen

Der Hersteller des Inkassozählers darf eine der in 6.2.2 bis 6.2.4 angegebenen Alternativen praktisch realisieren, die jedoch zwischen Hersteller und Versorgungsunternehmen zu vereinbaren ist.

Bei künftigen Überarbeitungen dieser Norm können weitere Alternativen beschrieben sein.

6.2.2 Alternative 1: Optische Schnittstelle

Die physikalische Verbindung und die Signalschnittstellen müssen den in IEC 62056-21, 4.3, angegebenen Anforderungen entsprechen.

6.2.3 Alternative 2: Linienstrom-Schnittstelle

Die physikalische Verbindung und die Signalschnittstellen müssen den in IEC 62056-21, 4.1, angegebenen Anforderungen entsprechen.

6.2.4 Alternative 3: Spannungsschnittstelle

Die physikalische Verbindung und die Signalschnittstellen müssen den in STS 18-001 angegebenen Anforderungen entsprechen.

6.3 Zeichenübertragung

6.3.1 Übertragungsart

Asynchrone serielle Übertragung – Halbduplex.

6.3.2 Übertragungsformat

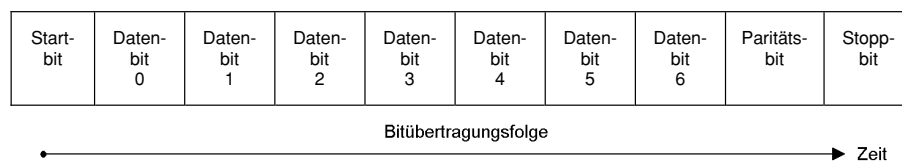


Bild 2 – Zeichenübertragungsformat

– Entwurf –

E DIN IEC 62055-52 (VDE 0418-5-52):2006-07

Wie in Bild 2 dargestellt umfasst das Zeichenübertragungsformat 1 Startbit, 7 Datenbits, 1 gerades Paritätsbit und 1 Stoppbit.

In Tabelle 2 ist die Bitcodierung der 7-Bit-ASCII-Zeichen angegeben.

Tabelle 2 – Bitcodierung eines 7-Bit-Zeichencodes

Bit	Wert	Beschreibung
Startbit	0	Kennzeichnet den Anfang eines seriellen Zeichenbitstroms
Datenbit 0	0 – 1	Niederwertigstes Bit des Zeichencodes
Datenbit 1	0 – 1	Zeichencode
Datenbit 2	0 – 1	Zeichencode
Datenbit 3	0 – 1	Zeichencode
Datenbit 4	0 – 1	Zeichencode
Datenbit 5	0 – 1	Zeichencode
Datenbit 6	0 – 1	Höchstwertigstes Bit des Zeichencodes
Paritätsbit	0	Wenn die Datenbits 0 bis 6 eine gerade Anzahl binärer Einsen enthalten
	1	Wenn die Datenbits 0 bis 6 eine ungerade Anzahl binärer Einsen enthalten
Stoppbit	1	Kennzeichnet das Ende eines seriellen Zeichenbitstroms

6.3.3 Übertragungsgeschwindigkeit

Baudrate = 2 400 Bit je Sekunde.

Der Server darf keine automatische Baudrateneinstellung oder -umschaltung unterstützen.

6.3.4 Zeichencodierung

Das Format der Anwendungsdaten ist in den Spezifikationen festgelegt, die für das spezielle Datenelement gelten. Für die Anwendung dieser Spezifikation müssen alle Anwendungsdatenelemente, die über einen virtuellen Tokenträger VTC07 zu übermitteln sind, vor der Zeichencodierung im binären, hexadezimalen oder dezimalen Format vorliegen.

Daten im Binärformat sind wie folgt zu codieren:

Daten im Binärformat werden in Halbbytes zu je 4 Bit aufgeteilt und jedes Halbbyte wird vor der Übertragung in seine 7-Bit-ASCII-Darstellung konvertiert. Das höchstwertigste Halbbyte wird zuerst übertragen. Während des Empfangs durch den Client wird ein umgekehrter Vorgang angewendet, um die Daten wieder in ihrem ursprünglichen Format zu erlangen. Für den Fall, dass das Datenformat keine vollständige Anzahl von Halbbytes hat, muss die Binärzahl rechts ausgerichtet und links mit führenden binären Nullen aufgefüllt werden, um eine vollständige Anzahl von Halbbytes zu erhalten. Ein Beispiel ist in Tabelle 3 angegeben.

Tabelle 3 – Beispiel für die Zeichencodierung einer 14-Bit-Binärzahl

Operation	Ergebnis	7-Bit-ASCII
Startwert der Binärzahl	11001011011110	
Aufteilung in Halbbytes mit je 4 Bit	11 0010 1101 1110	
Linke Seite mit Nullen auffüllen, um eine ganzzahlige Anzahl von Halbbytes zu erhalten	0011 0010 1101 1110	
Zu konvertierendes und zu übertragenes 1. Halbbyte	0011	0110011
Zu konvertierendes und zu übertragenes 2. Halbbyte	0010	0110010
Zu konvertierendes und zu übertragenes 3. Halbbyte	1101	1000100
Zu konvertierendes und zu übertragenes 4. Halbbyte	1110	1000101

Daten im Hexadezimalformat sind wie folgt zu codieren:

Daten im Hexadezimalformat werden in Hexadezimalziffern aufgeteilt und jede Ziffer wird vor der Übertragung in seine 7-Bit-ASCII-Darstellung konvertiert. Die höchstwertigste Ziffer wird zuerst übertragen. Während des Empfangs durch den Client wird ein umgekehrter Vorgang angewendet, um die Daten wieder in ihrem ursprünglichen Format zu erlangen. Ein Beispiel ist in Tabelle 4 angegeben.

Tabelle 4 – Beispiel für die Zeichencodierung einer Hexadezimalzahl mit 4 Ziffern

Operation	Ergebnis	7-Bit-ASCII
Startwert der Hexadezimalzahl	12AF	
Aufteilung in 4 Hexadezimalziffern	1 2 A F	
Zu konvertierende und zu übertragene 1. Ziffer	1	0110001
Zu konvertierende und zu übertragene 2. Ziffer	2	0110010
Zu konvertierende und zu übertragene 3. Ziffer	A	1000001
Zu konvertierende und zu übertragene 4. Ziffer	F	1000110

Daten im Dezimalformat sind wie folgt zu codieren:

Daten im Dezimalformat werden in Dezimalziffern aufgeteilt und jede Ziffer wird vor der Übertragung in seine 7-Bit-ASCII-Darstellung konvertiert. Die höchstwertigste Ziffer wird zuerst übertragen. Während des Empfangs durch den Client wird ein umgekehrter Vorgang angewendet, um die Daten wieder in ihrem ursprünglichen Format zu erlangen. Ein Beispiel ist in Tabelle 5 angegeben.

Tabelle 5 – Beispiel für die Zeichencodierung einer Dezimalzahl mit 4 Ziffern

Operation	Ergebnis	7-Bit-ASCII
Startwert der Dezimalzahl	3059	
Aufteilung in 4 Dezimalziffern	3 0 5 9	
Zu konvertierende und zu übertragene 1. Ziffer	3	0110011
Zu konvertierende und zu übertragene 2. Ziffer	0	0110000
Zu konvertierende und zu übertragene 3. Ziffer	5	0110101
Zu konvertierende und zu übertragene 4. Ziffer	9	0111001

6.4 Festlegungen der Nachrichtensyntax

6.4.1 Allgemeines

Nachfolgend werden die verschiedenen unterstützten Nachrichten angegeben. Die Bezugsnummern unter jedem Feld beziehen sich auf die Liste in 6.5 mit den ausführlichen Festlegungen für jedes Feld.

Nach der Codierung der Daten (siehe 6.3.4) repräsentiert jedes Feld ein oder mehr 7-Bit-ASCII-Zeichen.

6.4.2 Nachricht ID_Request

Der Client sendet die Nachricht ID_Request (Anforderung der Kennung), um den Server aufzufordern, eine Kennungsinformation zu erzeugen, die dem Client hilft den Servertyp zu bestimmen, mit dem er sich befasst, und ist üblicherweise die erste Nachricht, die der Server in einer Übertragungssitzung empfängt.

/	?	/	CR	LF
(1)	(2)	(3)	(4)	(5)

Zu weiteren Angaben über die Verarbeitung dieser Nachricht siehe auch 6.6.2.

6.4.3 Nachricht ID_Response

Der Server sendet die Nachricht ID_Response (Rückmeldung der Kennung) an den Client als Antwort auf den Empfang der Nachricht ID_Request vom Client. Der Server sendet die Kennung seines Inkassozähler-typs, indem er den Herstellercode und den Code der Softwareversion des Servers sendet.

I	M	X	V	CR	LF
(1)	(22)	(6)	(7)	(4)	(5)

Zu weiteren Angaben über die Verarbeitung dieser Nachricht siehe auch 6.6.2.

6.4.4 Nachricht Read_Command

Der Client liest einen Datensatz aus einem bestimmten Register (siehe 6.8.2) im Inkassozähler, indem er an den Server ein Read_Command (Lesebefehl) sendet.

SOH	R	STX	RID	DL	ETX	BCC
(8)	(9)	(10)	(11)	(12)	(13)	(14)

ANMERKUNG BCC wird aus den Feldern (9) bis (13) berechnet.

Zu weiteren Angaben über die Verarbeitung dieser Nachricht siehe auch 6.6.3.

6.4.5 Nachricht Write_Command

Der Client schreibt einen Datensatz an ein bestimmtes Register (siehe 6.8.2) im Inkassozähler, indem er an den Server ein Write_Command (Schreibbefehl) sendet.

SOH	W	STX	RID	(	D	)	ETX	BCC
(8)	(15)	(10)	(11)	(16)	(17)	(18)	(13)	(14)

ANMERKUNG BCC wird aus den Feldern (15) bis (13) berechnet.

Zu weiteren Angaben über die Verarbeitung dieser Nachricht siehe auch 6.6.4.

6.4.6 Nachricht Break_Command

Der Client annulliert alle vorherigen Nachrichten, die vom Server bis jetzt noch nicht ausgeführt wurden, indem er an den Server ein Break_Command (Abbruchbefehl) sendet.

SOH	B	ETX	BCC
(8)	(19)	(13)	(14)

ANMERKUNG BCC wird aus den Feldern (19) bis (13) berechnet.

Zu weiteren Angaben über die Verarbeitung dieser Nachricht siehe auch 6.6.5.

6.4.7 ACK: Nachricht Acknowledge

Der Server gibt eine positive Rückmeldung zurück, indem er die Nachricht Acknowledge sendet.

ACK
(20)

Zu weiteren Angaben über die Verarbeitung dieser Nachricht siehe auch 6.6.4 und 6.6.5.

6.4.8 NAK: Nachricht Negative_Acknowledge

Der Server gibt eine negative Rückmeldung zurück, indem er die Nachricht Negative_Acknowledge sendet.

NAK
(21)

Zu weiteren Angaben über die Verarbeitung dieser Nachricht siehe auch 6.6.1 bis 6.6.5 und 6.7.2.

6.4.9 Nachricht Data

Der Server sendet eine Nachricht Data (Daten) als Antwort auf ein Read_Command vom Client.

STX	(	D	)	ETX	BCC
(10)	(16)	(17)	(18)	(13)	(14)

ANMERKUNG BCC wird aus den Feldern (16) bis (13) berechnet.

Zu weiteren Angaben über die Verarbeitung dieser Nachricht siehe auch 6.6.3.

6.5 Festlegungen der Nachrichtfelder

Jedes Nachrichtenfild besteht aus einem oder mehr 7-Bit-ASCII-Zeichen. In der nachstehenden Tabelle ist der mögliche Wert für jedes Zeichen in einem Feld angegeben.

Tabelle 6 – Festlegungen der Nachrichtfelder

Feld	Feld-Nr.	Zeichen-anzahl	Zeichen-wert	Beschreibung	Bezug
/	(1)	1	47	Startzeichen ASCII „/“; Schrägstrich	6.4.2 6.4.3
?	(2)	1	63	Zeichen für Befehl ID_Request ASCII „?“; Fragezeichen	6.4.2
!	(3)	1	33	Endezeichen ASCII „!“; Ausrufezeichen	6.4.2
CR	(4)	1	13	Beendigungszeichen ASCII CR; Wagenrücklauf	6.4.2 6.4.3
LF	(5)	1	10	Beendigungszeichen ASCII LF; Zeilenvorschub	6.4.2 6.4.3
X	(6)	2	48 - 57	Herstellercode ASCII „0“ bis „9“ Bereich 00 bis 99 Dezimale Darstellung des Herstellercodes mit zwei Ziffern, wie es in IEC 62055-41, 6.1.2.3.2 festgelegt ist	6.4.3
V	(7)	4	48 - 57 und 65 - 70	Software_Version ASCII „0“ bis „9“ und „A“ bis „F“ Darstellung der Softwareversion des Inkassozählers mit vier Ziffern. Die Interpretation der Ziffern ist hersteller-spezifisch (siehe auch 6.8.3.5).	6.4.3
SOH	(8)	1	1	Kopfzeichen ASCII SOH; Start of header	6.4.4 6.4.5
R	(9)	1	82	Zeichen für Befehl Read ASCII „R“	6.4.4
STX	(10)	1	2	Zeichen für Rahmenanfang ASCII STX; Start of text	6.4.4 6.4.5

– Entwurf –

E DIN IEC 62055-52 (VDE 0418-5-52):2006-07

Feld	Feld-Nr.	Zeichen-anzahl	Zeichen-wert	Beschreibung	Bezug
RID	(11)	4	48 - 57 und 65 - 70	Register_ID ASCII „0“ bis „9“ und „A“ bis „F“ Feld Registerkennung im Hexadezimalformat mit vier Ziffern. Es ist die Kennung des Registers, aus dem die Daten zu lesen sind oder an das Daten zu schreiben sind. Zur Festlegung der Register_Table siehe auch 6.8.	6.4.4 6.4.5
DL	(12)	1	48 - 57 und 65 - 70	Datenlänge ASCII „0“ bis „9“ und „A“ bis „F“ Der Server muss den Wert dieses Feldes entsprechend der jeweiligen Festlegung für jedes Register (siehe 6.8.2) in einer Instanz Register_Table (siehe 6.8.1) interpretieren. ANMERKUNG In älteren Inkassozählern wurde dieses Steuerdatenelement zur Festlegung der Byteanzahl verwendet, die aus einem Register während der Ausführung der Nachricht Read_Command zu lesen sind. Es war auch auf einen Höchstwert von 10 beschränkt.	6.4.4
ETX	(13)	1	3	Zeichen für Rahmenende ASCII ETX; End of text	6.4.4 6.4.5 6.4.6 6.4.9
BCC	(14)	1	0 - 127	Blockprüfzeichen Dies ist ein berechneter Wert und kann zu einem beliebigen Wert innerhalb des kompletten 7-Bit-ASCII-Bereiches führen. Das BCC wird aus dem ersten Zeichen unmittelbar nach dem ersten SOH- oder STX-Zeichen in der Nachricht und bis einschließlich dem ETX-Zeichen, welches den Nachrichtenrahmen abschließt, berechnet. Das BCC wird durch Ausführung eines zeichenweisen logischen XOR über 7 Bit berechnet und unmittelbar hinter das ETX-Zeichen gestellt.	6.4.4 6.4.5 6.4.6 4.4.9
W	(15)	1	87	Zeichen für Befehl Write ASCII „W“	6.4.5
(	(16)	1	40	Zeichen für Datenblockbeginn ASCII „(“; linke runde Klammer	6.4.5 6.4.9
D	(17)	X	48 - 57 und 65 - 70	Datensatz ASCII „0“ bis „9“ und „A“ bis „F“ Datenfeld, welches die zu schreibenden (Nachricht Write_Command) oder zu lesenden (Nachricht Read_Command) Daten darstellt. ANMERKUNG In älteren Inkassozählern war dieser Datensatz auf maximal 10 Bytes beschränkt (20 Zeichen).	6.4.5 6.4.9
)	(18)	1	41	Zeichen für Datenblockende ASCII „)“; rechte runde Klammer	6.4.5 6.4.9
B	(19)	1	66	Zeichen für den Befehl Break ASCII „B“	6.4.6
ACK	(20)	1	6	Zeichen: Acknowledge ASCII ACK; Bestätigung	6.4.7
NAK	(21)	1	21	Zeichen: negative Acknowledge ASCII NAK; negative Rückmeldung	6.4.8
M	(22)	1	77	Herstellerangaben folgen ASCII „M“ Zeigt an, dass in den nächsten 6 Zeichen der Nachricht Einzelheiten des Herstellers enthalten sind.	6.4.3

Für jede Registerinstanz müssen das D-Feld (17) und das DL-Feld (12) in einer Begleitspezifikation festgelegt werden (siehe 6.8.1 und 6.9).

6.6 Funktionen des Protokolls der Bitübertragungsschicht

6.6.1 Flussdiagramm des Serverprotokolls

Das Flussdiagramm des Serverprotokolls wird in Bild 3 dargestellt.

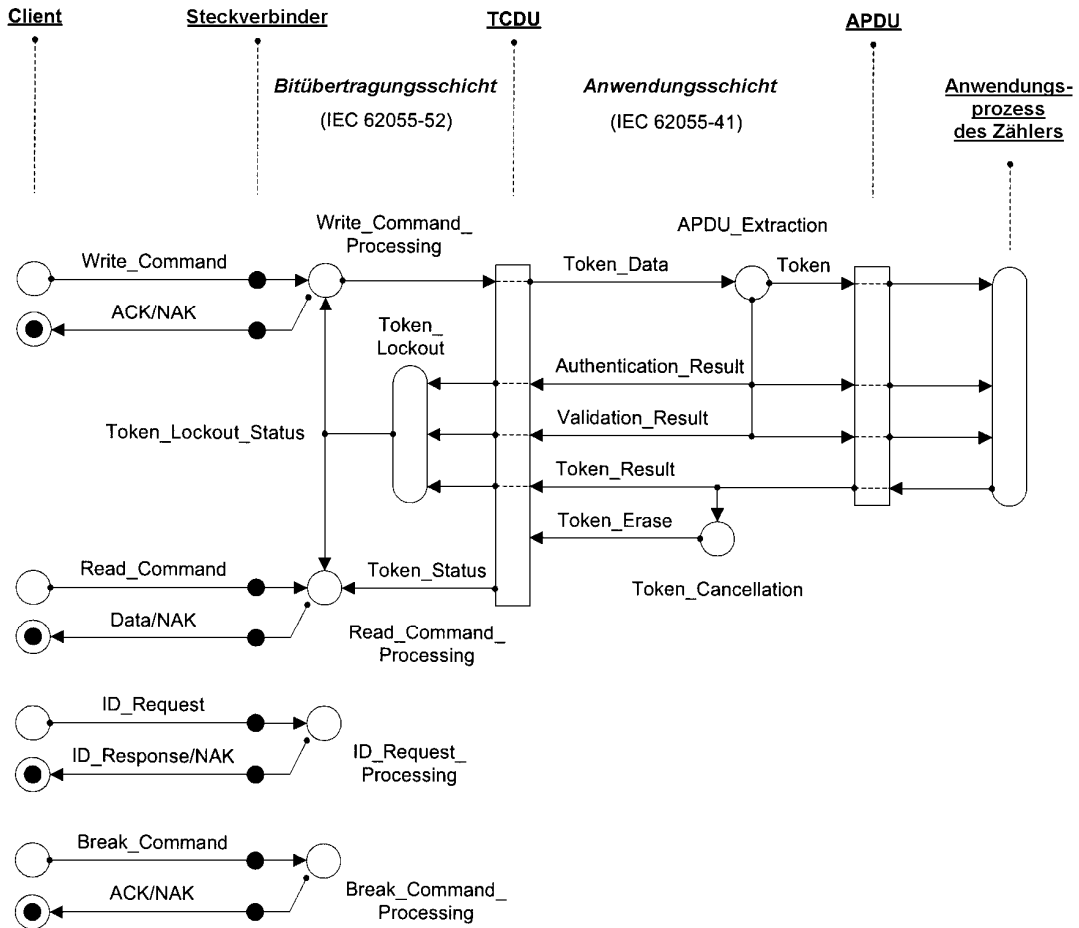


Bild 3 – Flussdiagramm des Serverprotokolls

Der Client darf Nachrichten in beliebiger Reihenfolge senden und der Server muss jede Nachricht in der Reihenfolge verarbeiten wie sie empfangen wurden; dies gilt nicht für die Nachricht **Break_Command** (siehe 6.6.5).

Jede Übertragungssitzung muss ein Nachrichtenpaar darstellen, die als Anforderung vom Client begonnen und mit einer Antwort vom Server beendet wird.

Der Client kann durch Zeitüberschreitungen (siehe 6.6.2 bis 6.6.6 und 6.7) Fehlerbedingungen der Sitzung beheben.

Der Server muss die in Tabelle 7 angegebenen Anforderungsnachrichten unterstützen.

Tabelle 7 – Vom Server unterstützte Anforderungsnachrichten

Anforderungsnachricht	Bezug
ID_Request	6.4.2
Read_Command	6.4.4
Write_Command	6.4.5
Break_Command	6.4.6

Der Server muss die in Tabelle 8 angegebenen Antwortnachrichten unterstützen.

Tabelle 8 – Vom Server unterstützte Antwortnachrichten

Antwortnachricht	Bezug
ID_Response	6.4.3
ACK	6.4.7
NAK	6.4.8
Data	6.4.9

Der Server muss die in Tabelle 9 angegebenen Funktionen unterstützen.

Tabelle 9 – Vom Server unterstützte Funktionen

Funktion	Bezug
ID_Request_Processing	6.6.2
Read_Command_Processing	6.6.3
Write_Command_Processing	6.6.4
Break_Command_Processing	6.6.5
Token_Lockout	6.6.6

6.6.2 Funktion ID_Request_Processing

Die Funktion ID_Request_Processing (Verarbeitung der Kennungsanforderung) ist im Protokoll der Bitübertragungsschicht des Inkassozähler-Servers lokalisiert, welcher die ID_Request-Nachrichten (siehe 6.4.2) von einem Client empfängt und verarbeitet.

Der Server muss eine ID_Request-Nachricht entsprechend dem in Anhang A, Bild A.1, angegebenen Serverzustandsdiagramm verarbeiten.

Der Server muss unter jeder der folgenden Bedingungen mit einer NAK antworten:

- In einem der empfangenen Zeichen entspricht der Wert des Paritätsbits nicht 6.3.2; der Server muss im Register Server_Status auch den Wert für Parity_Error (Paritätsfehler) setzen.
- Die maximal zulässige Zeit zwischen den Zeichen wurde nach den in Tabelle 11 angegebenen Werten überschritten; der Server muss im Register Server_Status auch den Wert für Character_Timeout_Error (Zeichen-Zeitüberschreitungsfehler) setzen.
- Der Client hat mehr Zeichen gesendet als der Server empfangen könnte; der Server muss im Register Server_Status auch den Wert für Character_Overflow_Error (Zeichenüberschreitungsfehler) setzen.
- Die empfangene Nachrichtensyntax und -struktur entspricht der Festlegung, wie sie in 6.4.2 angegeben ist; der Server muss im Register Server_Status auch den Wert für Message_Syntax_Error (Syntaxfehler in der Nachricht) setzen.
- Beim Empfang der Nachricht ist ein Fehler aufgetreten, der in dieser Norm nicht festgelegt ist; der Server muss im Register Server_Status auch den Wert für Undefined_Transmission_Error (undefinierter Übertragungsfehler) setzen.

Wenn keine der vorstehenden Bedingungen wahr ist, dann muss der Server im Register `Server_Status` den Wert für `Message_Accepted` (Nachricht angenommen) setzen und anschließend mit der Ausführung des Befehls fortsetzen.

Für jede der vorstehenden Bedingungen muss der Wert des Registers `Server_Status` (siehe 6.8.3.4) auf den entsprechenden Wert gesetzt werden, der in Tabelle 15 angegeben ist.

Nach der Ausführung des Befehls muss der Server dem Client (siehe 6.4.3) mit einer Nachricht `ID_Response` antworten.

6.6.3 Funktion `Read_Command_Processing`

Die Funktion `Read_Command_Processing` (Verarbeitung des Lesebefehls) ist im Protokoll der Bitübertragungsschicht des Inkassozähler-Servers lokalisiert, welcher die Nachrichten `Read_Command` (siehe 6.4.4) von einem Client empfängt und verarbeitet.

Um Informationen von einem bestimmten Register im Inkassozähler zu „lesen“, darf der Client eine Nachricht `Read_Command` mit dem jeweiligen `Register_ID`-Wert im `RID`-Feld entsprechend den Festlegungen in der `Register_Table` (siehe 6.8) senden.

Der Server muss eine Nachricht `Read_Command` nach dem in Anhang A, Bild A.2, angegebenen Zustandsdiagramm verarbeiten.

Der Server muss unter jeder der folgenden Bedingungen mit einer NAK antworten:

- In einem der empfangenen Zeichen entspricht der Wert des Paritätsbits nicht 6.3.2; der Server muss im Register `Server_Status` auch den Wert für `Parity_Error` setzen.
- Die maximal zulässige Zeit zwischen den Zeichen wurde nach den in Tabelle 11 angegebenen Werten überschritten; der Server muss im Register `Server_Status` auch den Wert für `Character_Timeout_Error` setzen.
- Der Client hat mehr Zeichen gesendet als der Server empfangen könnte; der Server muss im Register `Server_Status` auch den Wert für `Character_Overflow_Error` setzen.
- Die empfangene Nachrichtensyntax und -struktur entspricht nicht der Festlegung, wie sie in 6.4.4 angegeben ist; der Server muss im Register `Server_Status` auch den Wert für `Message_Syntax_Error` setzen.
- Das `BCC`-Feld in der Nachricht (siehe 6.4.4) stimmt nicht mit dem berechneten Wert aus der empfangenen Nachricht überein; der Server muss im Register `Server_Status` auch den Wert für `BCC_Error` (Fehler des Blockprüfzeichens) setzen.
- Beim Empfang der Nachricht ist ein Fehler aufgetreten, der in dieser Norm nicht festgelegt ist; der Server muss im Register `Server_Status` auch den Wert für `Undefined_Transmission_Error` setzen.

Für jede der vorstehenden Bedingungen muss der Wert des Registers `Server_Status` (siehe 6.8.3.4) auf den entsprechenden Wert gesetzt werden, der in Tabelle 15 angegeben ist.

Wenn keine der vorstehenden Bedingungen wahr ist, dann muss der Server fortsetzen, um den Datensatz von dem Register zu „lesen“ der im `RID`-Feld der Nachricht angegeben ist.

Der Server muss während des „Lesens“ aus dem Register unter jeder der folgenden Bedingungen mit einer NAK antworten:

- Die `Register_ID`, wie sie im `RID`-Feld der Nachricht empfangen wurde, entspricht nicht einem im Server unterstützten Register; der Server muss im Register `Server_Status` auch den Wert für `Register_ID_Invalid` (ungültige Kennung des Registers) setzen.
- Das Register ist belegt und kann zu diesem Zeitpunkt die Anforderung zum „Lesen“ nicht bearbeiten; der Server muss im Register `Server_Status` auch den Wert für `Register_Busy` (Register belegt) setzen.
- Das Register hat nicht das `R`-Attribut, das als wahr definiert ist (siehe 6.8.2), folglich kann es nicht „gelesen“ werden; der Server muss im Register `Server_Status` auch den Wert für `Register_Read_Protected` (Register ist lesegeschützt) setzen.

– Entwurf –

E DIN IEC 62055-52 (VDE 0418-5-52):2006-07

- Das Register wird unterstützt, jedoch ist die Funktion, mit der es in Verbindung steht, im Inkassozähler deaktiviert worden; der Server muss im Register Server_Status auch den Wert für Function_Disabled (Funktion deaktiviert) setzen.
- Während der Phase des „Lesens“ des Registers tritt ein Fehler auf, der in dieser Norm nicht definiert ist; der Server muss im Register Server_Status auch den Wert für Undefined_Reading_Error (undefinierter Lesefehler) setzen.

Wenn keine der vorstehenden Bedingungen wahr ist, dann muss der Server im Register Server_Status den Wert für Message_Accepted setzen und anschließend mit der Ausführung des „Lesens“ aus dem Register fortsetzen.

Für jede der vorstehenden Bedingungen muss der Wert des Registers Server_Status (siehe 6.8.3.4) auf den entsprechenden Wert gesetzt werden, der in Tabelle 15 angegeben ist.

Nach der Ausführung des Befehls muss der Server mit der Datennachricht an den Client (siehe 6.4.9) fortsetzen.

Zusätzliche Verarbeitungen bezüglich der Tokeneingabe werden in 6.8.3.7 für das Register Token_Status angegeben.

6.6.4 Funktion Write_Command_Processing

Die Funktion Write_Command_Processing (Verarbeitung des Schreibbefehls) ist im Protokoll der Bit-übertragungsschicht des Inkassozähler-Servers lokalisiert, welcher die Nachrichten Write_Command (siehe 6.4.5) von einem Client empfängt und verarbeitet.

Um Informationen in einem bestimmten Register im Inkassozähler zu „schreiben“, darf der Client eine Nachricht Write_Command mit dem Datensatz im D-Feld und dem entsprechenden Register_ID-Wert im RID-Feld entsprechend den Festlegungen in der Register_Table (siehe 6.8) senden.

Der Server muss eine Nachricht Write_Command nach dem in Anhang A, Bild A.3, angegebenen Server-Zustandsdiagramm verarbeiten.

Der Server muss unter jeder der folgenden Bedingungen mit einer NAK antworten:

- In einem der empfangenen Zeichen entspricht der Wert des Paritätsbits nicht 6.3.2; der Server muss im Register Server_Status auch den Wert für Parity_Error setzen.
- Die maximal zulässige Zeit zwischen den Zeichen wurde nach den in Tabelle 11 angegebenen Werten überschritten; der Server muss im Register Server_Status auch den Wert für Character_Timeout_Error setzen.
- Der Client hat mehr Zeichen gesendet als der Server empfangen könnte; der Server muss im Register Server_Status auch den Wert für Character_Overflow_Error setzen.
- Die empfangene Nachrichtensyntax und -struktur entspricht nicht der Festlegung, wie sie in 6.4.5 angegeben ist; der Server muss im Register Server_Status auch den Wert für Message_Syntax_Error setzen.
- Das BCC-Feld in der Nachricht (siehe 6.4.5) stimmt nicht mit dem berechneten Wert aus der empfangenen Nachricht überein; der Server muss im Register Server_Status auch den Wert für BCC_Error setzen.
- Beim Empfang der Nachricht ist ein Fehler aufgetreten, der in dieser Norm nicht festgelegt ist; der Server muss im Register Server_Status auch den Wert für Undefined_Transmission_Error setzen.

Für jede der vorstehenden Bedingungen muss der Wert des Registers Server_Status (siehe 6.8.3.4) auf den entsprechenden Wert gesetzt werden, der in Tabelle 15 angegeben ist.

Wenn keine der vorstehenden Bedingungen wahr ist, dann muss der Server mit dem „Schreiben“ des Datensatzes an das Register fortsetzen, das im RID-Feld der Nachricht angegeben ist.

Der Server muss während des „Schreibens“ an das Register unter jeder der folgenden Bedingungen mit einer NAK antworten:

- Die Register_ID, wie sie im RID-Feld der Nachricht empfangen wurde, entspricht nicht einem im Server unterstützten Register; der Server muss im Register Server_Status auch den Wert für Register_ID_Invalid setzen.
- Das Register ist belegt und kann zu diesem Zeitpunkt den Befehl zum „Schreiben“ nicht bearbeiten; der Server muss im Register Server_Status auch den Wert für Register_Busy setzen.
- Das Register hat nicht das W-Attribut, das als wahr definiert ist (siehe 6.8.2), folglich kann es nicht „beschrieben“ werden; der Server muss im Register Server_Status auch den Wert für Register_Write_Protected (Register ist schreibgeschützt) setzen.
- Das Register wird unterstützt, jedoch ist die Funktion, mit der es in Verbindung steht, im Inkassozähler deaktiviert worden; der Server muss im Register Server_Status auch den Wert für Function_Disabled setzen.
- Im Falle des „Schreibens“ an das Register Binary-Token_Entry (Eingabe eines binären Token) (siehe 6.8.3.6) oder an jedes funktionell gleichwertige Register, welches in einer Begleitspezifikation festgelegt sein kann, und wenn die Anzeige Token_Lockout_Status (Token-Sperrzustand) (siehe 6.6.6) aus der Funktion Token_Lockout (Token Sperre) wahr ist; der Server muss im Register Server_Status auch den Wert für Token_Lockout setzen.
- Während der Phase des „Schreibens“ an das Register tritt ein Fehler auf, der in dieser Norm nicht definiert ist; der Server muss im Register Server_Status auch den Wert für Undefined_Writing_Error (undefinierter Schreibfehler) setzen.

Wenn keine der vorstehenden Bedingungen wahr ist, dann muss der Server mit einer ACK antworten und den Wert für Message_Accepted im Register Server_Status setzen; anschließend ist mit der Ausführung des „Schreibens“ an das Register fortzusetzen.

Die zusätzliche Verarbeitung, die sich speziell auf das Laden von Token in den Inkassozähler bezieht, wird in 6.8.3.6 beschrieben.

Eine ACK- oder NAK-Antwort darf nicht vom Ergebnis nach der Ausführung einer Write_Command-Nachricht abhängen, sondern muss lediglich anzeigen, dass der Empfang der Nachricht gelungen ist.

Der Server darf nicht auf die Ausführung der Nachricht warten, sondern muss die ACK so bald wie möglich zurückgeben, wenn alle vorstehenden Bedingungen erfüllt sind.

Nach der Ausführung der Write_Command-Nachricht muss eine anwendungsspezifische Rückmeldung an den Client so erfolgen, dass in einem geeignet definierten Register entsprechende Datenelemente gesetzt werden, die vom Client gelesen werden können.

6.6.5 Funktion Break_Command_Processing

Die Funktion Break_Command_Processing (Verarbeitung des Abbruchbefehls) ist im Protokoll der Bit-übertragungsschicht des Inkassozähler-Servers lokalisiert, welcher die Nachrichten Break_Command (siehe 6.4.6) von einem Client empfängt und verarbeitet.

Der Server muss eine Nachricht Break_Command nach dem in Anhang A, Bild A.4, angegebenen Server-Zustandsdiagramm verarbeiten.

Der Server muss unter jeder der folgenden Bedingungen mit einer NAK antworten:

- In einem der empfangenen Zeichen entspricht der Wert des Paritätsbits nicht 6.3.2; der Server muss im Register Server_Status auch den Wert für Parity_Error setzen.
- Die maximal zulässige Zeit zwischen den Zeichen wurde nach den in Tabelle 11 angegebenen Werten überschritten; der Server muss im Register Server_Status auch den Wert für Character_Timeout_Error setzen.
- Der Client hat mehr Zeichen gesendet als der Server empfangen könnte; der Server muss im Register Server_Status auch den Wert für Character_Overflow_Error setzen.

– Entwurf –

E DIN IEC 62055-52 (VDE 0418-5-52):2006-07

- Die empfangene Nachrichtensyntax und -struktur entspricht nicht der Festlegung, wie sie in 6.4.6 angegeben ist; der Server muss im Register `Server_Status` auch den Wert für `Message_Syntax_Error` setzen.
- Das BCC-Feld in der Nachricht (siehe 6.4.6) stimmt nicht mit dem berechneten Wert aus der empfangenen Nachricht überein; der Server muss im Register `Server_Status` auch den Wert für `BCC_Error` setzen.
- Beim Empfang der Nachricht ist ein Fehler aufgetreten, der in dieser Norm nicht festgelegt ist; der Server muss im Register `Server_Status` auch den Wert für `Undefined_Transmission_Error` setzen.

Wenn keine der vorstehenden Bedingungen wahr ist, dann muss der Server mit einer ACK antworten und im Register `Server_Status` den Wert für `Message_Accepted` setzen; anschließend setzt er mit der Ausführung des „Break“-Befehls fort.

Für jede der vorstehenden Bedingungen muss der Wert des Registers `Server_Status` (siehe 6.8.3.4) auf den entsprechenden Wert gesetzt werden, der in Tabelle 15 angegeben ist.

Eine ACK- oder NAK-Antwort darf nicht vom Ergebnis nach der Ausführung einer `Break_Command`-Nachricht abhängen, sondern muss lediglich anzeigen, dass der Empfang der Nachricht gelungen ist.

Der Server darf nicht auf die Ausführung der Nachricht warten, sondern muss die ACK so bald wie möglich zurückgeben, wenn alle vorstehenden Bedingungen erfüllt sind.

Eine `Break_Command`-Nachricht muss immer die höchste Priorität besitzen und die nächste in der Reihe werden, die vom Server auszuführen sind, und zwar unabhängig von allen weiteren Nachrichten, die in einer Schlange auf die Abarbeitung warten können.

Bei der Ausführung einer `Break_Command`-Nachricht muss der Server alle anstehenden Nachrichten beenden und auf den Abschluss der Ausführung aller gegenwärtig aktiven Prozesse warten, die vom Client durch vorherige Befehlsmeldungen ausgelöst worden sind. Hinsichtlich der Behandlung von Fehlerbedingungen siehe auch 6.7.1 bis 6.7.3.

6.6.6 Funktion `Token_Lockout`

Die Funktion `Token_Lockout` (Tokensperre) ist im Protokoll der Bitübertragungsschicht des Inkassozähler-Servers lokalisiert und überwacht die Ergebnisse der Tokenverarbeitung im Protokoll der Anwendungsschicht oder dem `Meter_Application_Process` anhand der Felder `Validation_Result`, `Authentication_Result` und `Token_Result` in der TCDU (siehe Bild 3). Das Überwachungsergebnis wird durch `Token_Lockout_Status` angezeigt.

Die in diesem Protokoll festgelegte `Token_Lockout`-Funktion darf nur auf die Verarbeitung von Token wirken, die an der Schnittstelle für virtuelle Tokenträger, wie in dieser Norm definiert, eingegeben worden sind, und darf keinen Einfluss auf die Verarbeitung von Token haben, welche möglicherweise in den Inkassozähler mit Hilfe vorhandener anderer Schnittstellen für Tokenträger eingegeben werden.

Für die Anwendung dieser Funktion werden Kurzcodes (en: short-codes) nicht genauso wie Token behandelt. Kurzcodes dürfen keinen Einfluss auf die Funktion `Token_Lockout` haben und gültige Kurzcodes dürfen nicht von einer Sperrbedingung beeinträchtigt werden.

Für den Fall, dass Token aufgrund einer der Tokenrückweisungsbedingungen nach IEC 62055-41, 8.2, hintereinander zurückgewiesen werden, muss nach einer derartigen Rückweisung die weitere Eingabe eines folgenden Token für eine bestimmte Dauer gesperrt werden.

Innerhalb von mindestens 10 aufeinanderfolgenden Rückweisungen muss eine maximale Sperrzeit von etwa 60 s bis 120 s erreicht werden.

Ein empfohlenes Verfahren für die Sperrdauer besteht darin, mit einem kleinen Wert zu beginnen und diesen bei jeder folgenden Rückweisung größer werden zu lassen. Diese Herangehensweise ist zu bevorzugen, weil es für die Kunden, die auf Schwierigkeiten bei der Tokeneingabe unter schlechten Übertragungsbedingungen stoßen, leichter zu tolerieren ist.

Es dürfen auch andere Verfahren realisiert werden, jedoch muss die maximale Sperrdauer etwa 60 s bis 120 s betragen.

Bei der ersten Annahme eines Tokens der Klasse 0 oder der Klasse 2 nach den Bedingungen in IEC 62055-41, 8.2, muss die Sperrbedingung aufgehoben und der Normalbetrieb für den Tokeneingabevorgang wieder hergestellt werden.

Der verbleibende Wert der Sperrdauer in Sekunden muss im Register Token_Lockout_Time_Remaining (Restdauer der Tokensperre) (siehe 6.8.3.8) zur Verfügung stehen und kann mit dem entsprechend gesetzten Wert der Register_ID im RID-Feld mit einer Read_Command-Nachricht gelesen werden.

6.7 Anforderungen an die Zeitsteuerung des Servers

6.7.1 Zeitliche Steuerung der Nachrichten- und Zeichenzwischenräume

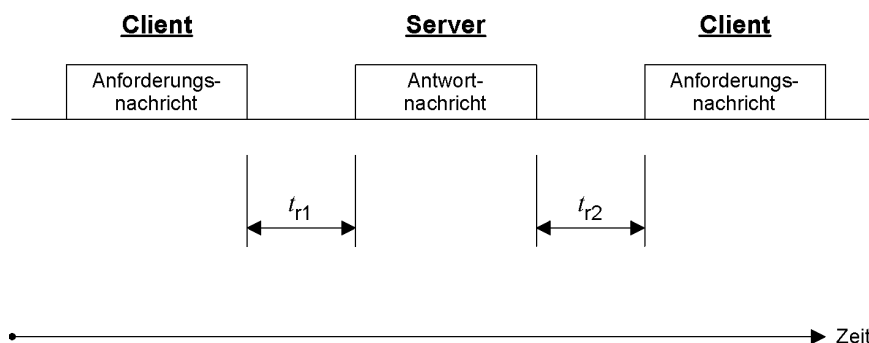


Bild 4 – Zeitliches Verhalten für die Nachrichtenzwischenräume

Der Server muss auf eine Anforderungsnachricht innerhalb der Zeitvorgaben antworten, die für t_{r1} in Tabelle 10 angegeben ist.

Der Server muss nach dem Senden der letzten Antwortnachricht innerhalb der Zeitvorgaben bereit zum Empfang einer Anforderungsnachricht sein, die für t_{r2} in Tabelle 10 angegeben ist.

Tabelle 10 – Anforderungen an die Zeitsteuerung des Servers für die Antwort auf eine Anforderungsnachricht

t_{r1}	t_{r2}
$20 \text{ ms} \leq t_{r1} \leq 1 \text{ 500 ms}$	$20 \text{ ms} \leq t_{r2} \leq 1 \text{ 500 ms}$

Wenn der Server auf eine Anforderung innerhalb der in Tabelle 10 angegebenen zeitlichen Anforderungen an die Nachrichtenzwischenräume nicht antwortet, dann kann der Client annehmen, dass ein Fehler aufgetreten ist.

Die Dauer zwischen zwei Zeichen in einer beliebigen Zeichenfolge muss innerhalb der Zeitvorgaben liegen, die für t_a in Tabelle 11 angegeben sind.

Tabelle 11 – Anforderungen an die Zeitsteuerung für die Zeichenzwischenräume

t_a
$t_a \leq 1 \text{ 500 ms}$

Wenn der Server feststellt, dass die in Tabelle 11 angegebenen Grenzwerte für die Zeitsteuerung der Zeichenzwischenräume überschritten worden sind, dann stellt dies entsprechend der Festlegung in 6.7.2 einen Übertragungsfehler dar.

6.7.2 Zeitbezogene Übertragungsfehler

Ein Übertragungsfehler tritt am Server auf, wenn er in der Nachricht einen Fehler feststellt, während er noch damit beschäftigt ist, diese vom Client zu empfangen und eine der folgenden Bedingungen erfüllt:

- In einem der empfangenen Zeichen entspricht der Wert des Paritätsbits nicht 6.3.2; diese Bedingung führt dazu, dass im Register Server_Status der Wert für Parity_Error gesetzt wird.
- Die maximal zulässige Zeit zwischen den Zeichen wurde nach den in Tabelle 11 angegebenen Werten überschritten; diese Bedingung führt dazu, dass im Register Server_Status der Wert für Character_Timeout_Error gesetzt wird.
- Der Client hat mehr Zeichen gesendet als der Server empfangen könnte; diese Bedingung führt dazu, dass im Register Server_Status der Wert für Character_Overflow_Error gesetzt wird.
- Die empfangene Nachrichtensyntax und -struktur entspricht nicht den Festlegungen in 6.4; diese Bedingung führt dazu, dass im Register Server_Status der Wert für Message_Syntax_Error gesetzt wird.
- Das BCC-Feld in der Nachricht (siehe 6.4) stimmt nicht mit dem berechneten Wert aus der empfangenen Nachricht überein; diese Bedingung führt dazu, dass im Register Server_Status der Wert für BCC_Error gesetzt wird.
- Während der Empfangsphase der Nachricht ist ein Fehler aufgetreten, der in dieser Norm nicht festgelegt ist; diese Bedingung führt dazu, dass im Register Server_Status der Wert für Undefined_Transmission_Error gesetzt wird.

Die Werte für die Fehlercodes im Register Server_Status werden in Tabelle 15 angegeben.

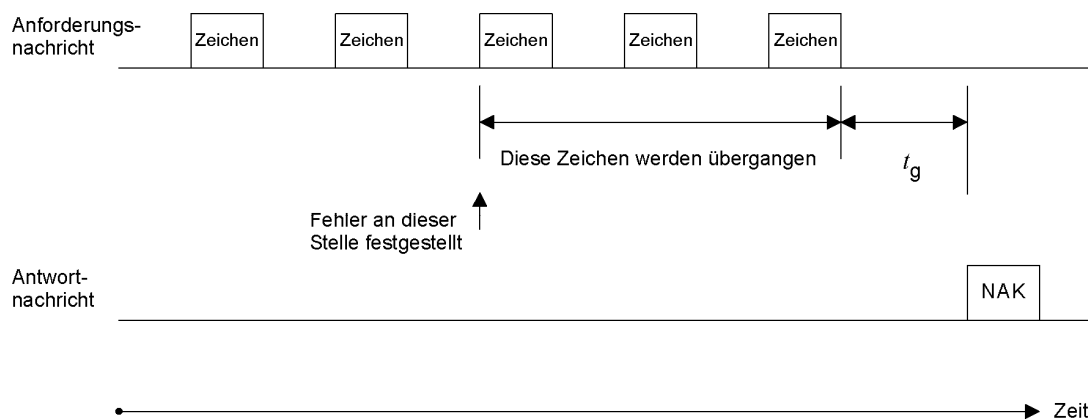


Bild 5 – Zeitbezogene Übertragungsfehler

Wenn der Server während des Empfangs einer Anforderungsnachricht einen Übertragungsfehler feststellt, dann muss er den Rest der Nachricht übergangen.

Anschließend muss der Server für die in Tabelle 12 angegebene Dauer t_g warten, während der keine Zeichen vor der Übertragung einer NAK-Antwort empfangen werden.

Tabelle 12 – Rückgewinnungswartedauer bei Übertragungsfehler

t_g
1 500 ms

6.7.3 Zeitliche Steuerung der Nachrichtenausführung

Die zeitliche Steuerung der Nachrichtenausführung ist in dieser Norm nicht festgelegt, jedoch sollte der Server innerhalb der Zeitvorgaben, die durch 6.7.1 und 6.7.2 auferlegt sind, bereit zum Empfang einer neuen Anforderungsnachricht sein.

Für den Fall, dass es für den Server notwendig ist, die durch 6.7.1 und 6.7.2 auferlegten Zeitvorgaben für die Ausführung einer Nachricht zu überschreiten, kann er den Wert Register_Busy im Server_Status-Register

verwenden (siehe 6.8.3.4). Hinsichtlich der Anwendung des Zustandscodes Register_Busy ist in 6.6.3 bis 6.6.4 ein Beispiel angegeben.

6.8 Register_Table

6.8.1 Register_Table-Schnittstellenklasse

Attribute	Bereich	Beschreibung
Name	{Register_Table_Name}	Der registrierte Name dieses Function_Object in der Begleit-spezifikation.
Table_ID	FOIN	Function_Object_Identification_Number, wie in der Begleit-spezifikation festgelegt.
Datenelemente		Externe Datenschnittstelle zur Instanz Function_Object. (aus Sicht der Token_Carrier)
Register_Array	Anordnung von einfachen oder komplexen Registern; Menge zwischen 1 und 65 536. Die Datenelemente werden je Registerinstanz definiert (siehe 6.8.2).	Die Anordnung der Registerinstanzen ist in der Begleitspezifikation definiert. Die Tabelle darf bis zu 65 536 Einträge haben. Jede Registerinstanz darf einen einfachen oder komplexen Datensatz-aufbau haben.
Verfahren		Äußere Serviceschnittstelle zur Instanz Function_Object. (aus Sicht der Token_Carrier)
kein	Die Verfahren sind je Registerinstanz definiert (siehe 6.8.2). Die Register_Table hat keine Operations-verfahren.	Die Tabelle ist virtuell und darf nicht zugänglich sein.
Operation		Interne Funktionalität der Register_Table
keine	Die Operationen sind je Registerinstanz definiert (siehe 6.8.2). Die Register_Table hat keine Operations-funktionalität.	Die Tabelle definiert bloß eine Liste von Registerdefinitionen, von denen jede eine logische Schnittstelle zu den tatsächlichen Registern des Inkassozähler zur Verfügung stellt.
Zuordnung		Unterstützungsdienste, die von anderen Function_Objects zur Verfügung gestellt werden
keine	Die Zuordnungen sind je Registerinstanz definiert (siehe 6.8.2).	Die Register_Table verwendet keine anderen festgelegten Dienste für ihr Funktionieren.

Diese Schnittstellenklasse liefert das übergeordnete Format für die Festlegung einer Register_Table als eine Instanz eines Meter_Function_Object, wie er in der Begleitspezifikation festgelegt ist (siehe 6.9).

Jede Instanz einer Register_Table ist gemeinsam mit jedem Register in einer Begleitspezifikation festgelegt.

Jede Register_Table ist eindeutig mit ihrer FOIN gekennzeichnet, die von der STS Association zugewiesen wird und wie sie in der Begleitspezifikation registriert ist.

Jede in der Register_Table definierte Registerinstanz (siehe 6.8.2) stellt eine logische Schnittstelle zu einem tatsächlichen Register oder einer Funktion im Inkassozähler zur Verfügung.