



Datenübertragungsprotokoll für stationäre Anlagen (LLE57)

(noch nicht vollständig realisiert!)

LLE[®]-MARK IV Industries GmbH

**Stockfeldstr. 1
D-76437 Rastatt**

**Telefon: (07222) 1001-0
Telefax: (07222) 1001-155**

Version:	3.0
Datum:	21.11.2000
Verfasser:	Dipl. Ing. Ch. Beck Abt. Entwicklung
LLE-Dokument:	N01201AD
Verteiler-Index:	-

Inhalt:

1. Einleitung.....	3
2. Allgemeiner Telegrammaufbau	4
2.1 Austauschzeichen	4
2.2 Telegrammrahmen bei Aufruftelegrammen.....	4
3. Definition des Datenfelds.....	6
3.1 Grundsätzlicher Aufbau des Datenfelds.....	6
3.2 Datensatz für Zeichensatzauswahl	6
3.3 Datensatz für Textausgabe.....	7
3.3.1 Datensatzbeschreibung	7
3.3.2 Anwendung der Ausgabesteuerung	10
3.3.3 Verwendung von Lauftexten und Blinken.....	10
3.3.4 Verwendung von Rückfalltexten beim „Timeout“	11
3.3.5 Beispiel für ein Telegramm mit Textausgabe:	12
3.4 Datensatz für Graphikausgabe	13
3.5 Datensatz zur Ansteuerung des Ausgabeports	15
3.6 Datensatz zur Uhrzeit- / Datumsübertragung.....	16
3.7 Datensatz für Wechselverkehrszeichen	17
4. Definition von Antworttelegrammen	18

Änderungen:

Datum	Version	Beschreibung	Bearbeiter
15.05.1997	1.0	Erstellung des Dokuments	CB
20.03.1996	1.0	Erstellung	UL
30.06.1997	1.1	Erklärung der Grafikübertragung	UL
27.01.1999	2.0	Überarbeitung + Definition der Antworttelegramme	CB
28.08.1999	2.1	"Datensatz zur Ansteuerung des Ausgabeports" (Kapitel 3.6) hinzugefügt, bisheriges Kapitel 3.6 jetzt Kapitel 3.7, empfohlener Schnittstellenparameter "gerade Parität" in "keine Parität" geändert	CB
08.02.2000	2.2	Bisher undokumentierte Funktion "Transparentsteuerung bei Fontauswahl" in Dokument eingefügt.	CB
23.03.2000	2.3	Portierung von AmiPro auf Word	CB
16.11.2000	3.0	Einführen des Begriffs „Feld“, Definition von Lauftexten und Blinken als Feldattribut, Definition eines Uhrzeittelegramms, Einführung von Slaveadresse 0xFF	CB

1. Einleitung

Das in diesem Dokument beschriebene Übertragungsprotokoll "LLE57" wurde in unserem Hause bereits in zahlreichen stationären Anwendungen eingesetzt. Es ist an das genormte Übertragungsprotokoll TC57 angelehnt und soll eine gewisse Kompatibilität garantieren. Aus diesem Grund existieren Telegrammbytes, die zum Verfassungszeitpunkt dieses Dokuments keine Bedeutung haben und auf feste Werte zu setzen sind.

Das Protokoll basiert auf dem Master-Slave-Prinzip. Der steuernde Rechner (im Folgenden "Leitstelle" genannt), ist der Master, alle Anzeigen sind Slaves. Nur der Master darf zu jeder Zeit Telegramme versenden ("Aufruftelegramme"). Slaves dürfen nur dann ein Telegramm absetzen ("Antworttelegramme"), wenn sie ein für sie bestimmtes gültiges Aufruftelegramm empfangen haben (Auswertung des Adreßbytes und der Prüfsumme im Aufruftelegramm, Slaveadresse ungleich 0xFF).

Antwortet ein Slave nicht auf ein für ihn bestimmtes Aufruftelegramm, so wiederholt der Master das Telegramm nach einer Timeoutzeit noch bis zu zwei mal. Bleibt jedes der drei Aufruftelegramme unbeantwortet, so kann von einem Defekt des Slave-Rechners oder von einer Unterbrechung der Übertragungsstrecke ausgegangen werden.

Die Datenübertragung erfolgt in der Regel bitseriell und asynchron im Halbduplexbetrieb mit den folgenden Parametern:

- **8 Datenbits**
- **keine Parität**
- **1 Stopbit**

Von den Anzeigen werden diverse Baudraten unterstützt. Empfohlen wird eine Baudrate von **9600 Baud**.

Bei zu übertragenden Werten die aus mehreren Bytes bestehen, wird das niederwertigste Byte zuerst geschickt.

Die Leitstelle muß mindestens alle 5 Minuten ein Telegramm an jede Anzeige schicken, da sonst die Anzeige ein Defekt der Leitstelle bzw. der Übertragungsstrecke annimmt und die Anzeige je nach Kundenwunsch ganz dunkel schaltet oder einen vordefinierten Text ausgibt.

Alle Text- und Graphiktelegramme verwenden X/Y-Koordinaten zur Positionierung auf der Anzeige. Kommen zeilenorientierte Anzeigen oder Anzeigen mit Blockschrift zum Einsatz (z.B. LCD-Anzeigen), so werden diese trotzdem wie eine Vollmatrixanzeige angesprochen. Es ist Aufgabe der Leitstelle, die Texte bzw. Graphiken so zu platzieren, daß es zu einer korrekten Anzeige kommt. Spezielle Zeichensätze, die an die Zeilen- bzw. Blockstruktur angepaßt sind, vereinfachen oftmals diese Aufgabe.

Im folgenden wird folgende Konvention getroffen:

Alle Zahlen, welche ein vorgestelltes "0x" besitzen, sind hexadezimale Zahlen. Alle anderen Zahlen sind dezimal zu interpretieren.

2. Allgemeiner Telegrammaufbau

2.1 Austauschzeichen

Um zuverlässig den Telegrammbeginn bzw. das Telegrammende erkennen zu können, darf das Rahmenzeichen 0x7E im Nutzdatenstrom nicht vorkommen. Um trotzdem alle Bitkombinationen übertragen zu können, ist das Austauschzeichen 0x7D vorgesehen. Das zu übertragende Zeichen muß dann mit 0x20 XOR-verknüpft werden.

Folgende Umsetzungen müssen im Nutzdatenstrom vor der Übertragung durchgeführt werden:

zu übertragendes Nutzdatenzeichen:	wird bei der Übertragung ersetzt durch die beiden Zeichen:
0x7D	0x7D 0x5D
0x7E	0x7D 0x5E

Alle oben **nicht** aufgeführte Zeichen bleiben bei der Übertragung unverändert.

Der Empfänger kann durch Umkehren der Zeichenersetzung die eigentlichen Nutzdaten wiedergewinnen.

Bei der Verwendung von Austauschzeichen verlängern sich die Telegramme. Zu beachten ist, daß alle Telegrammlängenangaben (auch im Telegramm selbst) sich ausschließlich auf die Nutzdaten **vor** dem Einfügen von Austauschzeichen beziehen!

2.2 Telegrammrahmen bei Aufruftelegrammen

Alle Aufruftelegramme im LLE57-Protokoll haben einen identischen Aufbau des Telegrammrahmens. Die Antworttelegramme sind vereinfacht, um eine unnötig große Belastung der Übertragungsstrecke zu vermeiden. Sie werden in Kapitel 4 beschrieben.

Alle in der folgenden Tabelle aufgeführten Bytes müssen in jedem Aufruftelegramm in der angegebenen Reihenfolge gesendet werden. Alle Zeichen bis auf die Rahmenzeichen müssen, wenn erforderlich, der in Kapitel 2.1 beschriebenen Prozedur unterworfen werden.

Funktion:	Wert/Wertebereich:	Bemerkungen:
Rahmenzeichen	0x7E	Dieses Zeichen wird nicht nach dem in Kapitel 2.1 beschriebenen Algorithmus ausgetauscht!
Startzeichen	0x68	
Längenbyte	0x04 ... 0xFF	Die Länge wird über das komplette Telegramm ab dem Steuerbyte (einschließlich) bis ausschließlich der Prüfsumme gebildet.
Kopie Längenbyte	0x04 ... 0xFF	Dieser Wert muß identisch mit dem davorstehenden Längenbyte sein.
Startzeichen	0x68	
Steuerbyte	0x43	Kennung für Datentelegramm

Slaveadresse	0x01 ... 0xFE	Über dieses Byte wird der Empfänger des Aufruftelegramms definiert. Die Slaveadresse kann in der Anzeige auf dem Anzeigenrechner eingestellt werden. Die Slaveadresse 0xFF spricht alle Anzeigen an! Empfängt ein Anzeigenrechner ein Telegramm mit der Slaveadresse 0xFF, so darf er kein Antworttelegramm absetzen! Die Slaveadressen 0x00 ist reserviert.
OSI3-Byte	0x00	Reserve, wird nicht verwendet
Informationsbyte	0x00 ... 0x3F	Anzahl der Datensätze im Datenfeld
Es folgt das Datenfeld mit den Datensätzen (max. 63 Datensätze mit insgesamt max. 251 Bytes, vergl. Kapitel 3). Es sind Werte von 0x00 ... 0xFF zulässig.		
Prüfsumme	0x00 ... 0xFF	Die Prüfsumme wird durch Addition aller Bytes im Telegramm ab dem Steuerbyte (einschließlich) bis ausschließlich der Prüfsumme selbst modulo 0x100 gebildet.
Stopzeichen	0x16	
Rahmenzeichen	0x7E	Dieses Zeichen wird nicht nach dem in Kapitel 2.1 beschriebenen Algorithmus ausgetauscht!

3. Definition des Datenfelds

3.1 Grundsätzlicher Aufbau des Datenfelds

Als erstes Byte im Datenfeld kommt einmalig folgende Information:

Funktion:	Wert/Wertebereich:	Bemerkungen:
Funktionsgruppe	0xB9	Es folgen Datensätze mit Text-, Zeichensatz-, oder Graphikinformationen etc. (vergl. Kapitel 3.2 bis 3.6).
	oder	
	0x04	Es folgen Datensätze für Wechselverkehrszeichen mit Textreferenznummern (vergl. Kapitel 3.7).

Alle nun folgenden Datensätze (siehe Kapitel 3.2 bis Kapitel 3.7) können in einem Datenfeld mehrfach wiederholt werden. Dabei dürfen innerhalb eines Telegramms nur Datensätze einer Funktionsgruppe verwendet werden.

Zur Ansteuerung von Abfahrts-, Fahrplan-, bzw. Werbeanzeigen in Vollmatrix-, Zeilenmatrix- und Blockmatrixstruktur werden praktisch ausschließlich Datensätze nach der Funktionsgruppe 0xB9 verwendet.

3.2 Datensatz für Zeichensatzauswahl

Dieser Datensatz ist nur gültig, wenn als Funktionsgruppe am Anfang des Datenfelds 0xB9 angegeben ist.

Der Datensatz wählt den aktiven Zeichensatz in der Anzeige aus. Dieser gilt bis auf weiteres für alle nachfolgenden "Datensätze für Textausgabe". Er dient sowohl zum einmaligen Einstellen eines Zeichensatzes wie auch zum Umschalten zwischen verschiedenen Zeichensätzen.

Nach einem Reset der Anzeige ist der Zeichensatz im Anzeigenrechner undefiniert. Daher wird empfohlen, in jedem Telegramm, in dem Datensätze für Textausgabe verwendet werden, zuerst einen Datensatz für die Zeichensatzauswahl zu senden.

Alle in der folgenden Tabelle aufgeführten Bytes müssen in der angegebenen Reihenfolge gesendet werden, um einen Zeichensatz auszuwählen. Alle Zeichen unterliegen, wenn erforderlich, der in Kapitel 2.1 beschriebenen Austauschprozedur.

Funktion:	Wert/Wertebereich:	Bemerkungen:
Datensatzlänge	0x03	Die Länge des Datensatzes ist auf 3 Bytes festgelegt
interne Adresse	0x01	
Datensatzkennung	0x32	Datensatzkennung für Zeichensatzauswahl

Zeichensatz- nummer + Spacing+ Transparenz	Bitpositionen: 76543210	
	7-----	=1: Textzeichen sind transparent, darunterliegende Zeichen bleiben sichtbar (diese Einstellung ist bevorzugt zu verwenden!). =0: Textzeichen sind nicht transparent, darunterliegende Zeichen werden verdeckt. Der Bereich der Verdeckung (immer rechteckig) ist abhängig von der editierten Größe des entsprechenden Textzeichens, d.h. der Bereich kann größer sein als die sichtbare Größe des Zeichens (z.B. nicht sichtbare Unterlänge).
	65 ----- Wertebereich: 0x0 bis 0x3	Wert gibt Leerspalten zwischen 2 Zeichen + 1 an („Spacing“), d.h. bei einem Wert von 0 wird eine Leerspalte zwischen 2 Zeichen eingefügt, bei einem Wert von z.B. 3 werden vier Leerspalten eingefügt.
	43210 ----- Wertebereich: 0x00 ... 0x1F	Nummer des Zeichensatzes in der Anzeige. Eine Übersicht über die in der Anzeige verfügbaren Zeichensätze und den entsprechenden Zeichensatznummern ist anzeigenabhängig bei LLE anzufordern.

3.3 Datensatz für Textausgabe

3.3.1 Datensatzbeschreibung

Dieser Datensatz ist nur gültig, wenn als Funktionsgruppe am Anfang des Datenfelds 0xB9 angegeben ist.

Der Datensatz beinhaltet neben dem darzustellenden Text ein Byte für die Ausgabesteuerung sowie Informationen zur Positionierung des Textes auf der Anzeige und der Art der Ausgabe (z.B. Blinkend, Lauftext etc.).

Alle in der folgenden Tabelle aufgeführten Bytes müssen in der angegebenen Reihenfolge gesendet werden, um einen Text an die Anzeige zu senden bzw. diesen auf der Anzeige sichtbar zu machen. Alle Zeichen unterliegen, wenn erforderlich, der in Kapitel 2.1 beschriebenen Austauschprozedur.

Funktion:	Wert/Wertebereich:	Bemerkungen:
Datensatzlänge	0x02 ... 0xFB	Die Datensatzlänge gibt die Anzahl der Bytes ab dem nächsten bis einschließlich dem letzten zu diesem Datensatz gehörenden Byte an. Für die maximale Datensatzlänge muß auch die maximale Größe des Datenfeldes (vergl. Kapitel 2.2) beachtet werden.
interne Adresse	0x01	

Datensatz- kennung	0x30	Datensatzkennung für Textausgabe
Textsteuerbyte	<p>Bitpositionen: 76543210</p> <p>7654 ----- Wertebereich: 0x1 ... 0xF</p> <p>3 ----- (Wert immer 0)</p> <p>210 -----</p> <p>Wert 0x0:</p> <p>Wert 0x1:</p> <p>Wert 0x2:</p> <p>Wert 0x3:</p> <p>Wert 0x4:</p> <p>Wert 0x5:</p> <p>Wert 0x6, 0x7:</p>	<p>Anzeigenspeicher, in welcher der Text geschrieben wird und der für die Ausgabesteuerung relevant ist. Es können in der Anzeige prinzipiell bis zu 14 verschiedene Anzeigenspeicher angesprochen werden. Achtung: Der Wertebereich dieses Parameters ist projektabhängig. Normalerweise sind 9 Ausgabematrizen verfügbar (0x1-0x9)!</p> <p>keine Funktion</p> <p>Ausgabesteuerung (Anwendung siehe Kapitel 3.3.2):</p> <p>Der Anzeigenspeicher wird vor dem Beschreiben gelöscht. Der Text wird in den Anzeigenspeicher geschrieben. Danach wird der Speicher zur Anzeige gebracht.</p> <p>Der Anzeigenspeicher wird vor dem Beschreiben nicht gelöscht. Der neue Text wird zusätzlich zu dem vorhandenen Text in den Anzeigenspeicher geschrieben. Danach wird der Speicher zur Anzeige gebracht.</p> <p>Der Anzeigenspeicher wird vor dem Beschreiben nicht gelöscht. Der neue Text wird zusätzlich zu dem vorhandenen Text in den Anzeigenspeicher geschrieben. Der Speicher wird noch nicht zur Anzeige gebracht.</p> <p>Der Anzeigenspeicher wird vor dem Beschreiben gelöscht. Der Text wird in den Anzeigenspeicher geschrieben. Der Speicher wird noch nicht zur Anzeige gebracht.</p> <p>Der Anzeigenspeicher wird auf "gelb" gesetzt (alle Anzeigenelemente aktiv). Der Speicher wird noch nicht zur Anzeige gebracht.</p> <p>Der Anzeigenspeicher wird auf Schachbrettmuster gesetzt (abwechselnd in X- und Y-Richtung ein Anzeigenelement aktiv, das nächste inaktiv usw.). Der Speicher wird noch nicht zur Anzeige gebracht.</p> <p>reserviert</p> <p>Wichtig: Für die Behandlung von Lauftexten und blinkenden Texten bitte Kapitel 3.3.3 beachten.</p>

Horizontale Feldposition	0x0000 ... 0xFFFF	Y-Koordinate der linken oberen Ecke eines zum nachfolgenden Text gehörenden virtuellen Textfeldes (2 Bytes!). Der Koordinatenursprung (0/0) befindet sich in der linken oberen Ecke der Anzeige. Positive Y-Koordinaten führen nach unten.
Vertikale Feldposition	0x0000 ... 0xFFFF	X-Koordinate der linken oberen Ecke eines zum nachfolgenden Text gehörenden virtuellen Textfeldes (2 Bytes!). Der Koordinatenursprung (0/0) befindet sich in der linken oberen Ecke der Anzeige. Positive X-Koordinaten führen nach rechts.
Feldbreite	0x0000 ... 0xFFFF	Breite eines zum nachfolgenden Text gehörenden virtuellen Textfeldes in Pixeln (2 Bytes!).
Feldhöhe	0x00 .. 0xFF	Höhe eines zum nachfolgenden Text gehörenden virtuellen Textfeldes in Pixeln
Feldattribut	Wert 0x00 Wert 0x10 Wert 0x20	links und oben bündig: Nachfolgender Text wird links und oben bündig in das Feld geschrieben. Der Text wird gegebenenfalls unten pixelgenau und rechts zeichengenau abgeschnitten. („zeichengenau“ bedeutet, daß ab dem ersten nicht mehr vollständig ins Feld passenden Zeichen die Textausgabe unterdrückt wird). Blinken: (bitte Kapitel 3.3.3 beachten) Der nachfolgende Text wird wie unter Feldattribut 0x00 beschrieben, jedoch blinkend dargestellt. Lauftext: (bitte Kapitel 3.3.3 beachten) Der nachfolgende Text läuft beginnend mit einem komplett dunklen Feld von rechts nach links durch, bis das Feld wieder komplett dunkel ist. Danach beginnt der Zyklus aufs Neue. Der Text wird oben bündig im Feld ausgegeben und pixelgenau an den Feldgrenzen abgeschnitten.
Textlänge	0x00 ... 0x7F	Anzahl Zeichen im folgenden Textstring
Text (beliebige ASCII-Zeichen), max. Länge: 127 Zeichen		

3.3.2 Anwendung der Ausgabesteuerung

Die folgende Tabelle enthält Beispiele über die Anwendung der Ausgabesteuerung in Verbindung mit Datensätzen zur Textausgabe (Textsteuerbyte, Bit 2 bis 0). Hierbei wird die Möglichkeit für die Verwendung von mehreren Anzeigenspeichern außer acht gelassen.

Ziel	Datensatz	Wert der Ausgabesteuerbits
1 Text soll dargestellt werden	mit Text	0x0
2 Texte sollen dargestellt werden	mit 1. Text	0x3
	mit 2. Text	0x1
3 Texte sollen dargestellt werden	mit 1. Text	0x3
	mit 2. Text	0x2
	mit 3. Text	0x1
n Texte sollen dargestellt werden (n > 3)	mit 1. Text	0x3
	mit 2. Text	0x2

	mit (n-1). Text	0x2
	mit n. Text	0x1

Bei der Verwendung von Lauftexten und Blinken ist Kapitel 3.3.3 zu beachten.

Die Verwendung von mehreren Anzeigenspeichern (vergl. obere 4 Bits des Textsteuerbytes) ermöglicht die zeitliche Trennung der eigentlichen Datenübertragung und der Anzeige der Daten. Zum Beschreiben eines Ausgabespeichers „im Hintergrund“ dienen die Ausgabesteuerungswerte 0x2 und 0x3. Zum Anzeigen eines „im Hintergrund“ bereits beschriebenen Anzeigenspeichers dient ein Datensatz zur Textausgabe mit einem Leerzeichen (Space) im Textteil und einem Ausgabesteuerungswert 0x1.

3.3.3 Verwendung von Lauftexten und Blinken

Jedem Lauftext bzw. blinkenden Text ist im Anzeigenrechner ein Prozess zugeordnet, der sich um die Verwaltung des Textes kümmert (Aktivierung des Textes, Ausgabe, Terminierung etc.)

Jeder Prozess kann mehreren Anzeigenspeichern zugeordnet sein. Ein Prozess wird bei seiner Erzeugung genau einem Anzeigenspeicher zugeordnet. Ein bereits existierender Prozess wird immer dann zusätzlich einem anderen Anzeigenspeicher zugeordnet, wenn im entsprechenden Telegramm zur Beschreibung des Anzeigenspeichers ein Text mit den gleichen Feldkoordinaten, Feldgrößen, dem gleichen Feldattribut und dem identischen Text gesendet wird. Dies gilt auch für einen Anzeigenspeicher, auf dem aktuell ein Prozess läuft, der jedoch im Hintergrund neu beschrieben wird. Durch das Löschen des Speichers wird die Zuordnung des Prozesses zum Anzeigenspeicher zwar aufgehoben (siehe unten), durch das Übertragen des gleichen Textes mit den identischen Feldinformationen jedoch wird einfach die Zuordnung des noch existierenden Prozesses wieder hergestellt. Der Prozess läuft nach Ausgabe des Speichers einfach weiter. Somit ist es möglich, z.B. einen Anzeigenspeicher mehrfach identisch zu beschreiben und unmerklich für den Betrachter zur Anzeige zu bringen.

Wird zwischen Anzeigenspeichern, welchen der gleiche Prozess zugeordnet ist, umgeschaltet, so läuft der Prozess ohne zu stocken auf dem nächsten Speicher weiter. Wird auf einen Speicher umgeschaltet, welchem der Prozess nicht zugeordnet ist, so wird der Prozess angehalten (der Text verschwindet, an der Stelle kann der neue Speicher einen

anderen Text anzeigen). Wird ein Anzeigenspeicher, welchem der Prozess zugeordnet ist, wieder aktiv, so startet der Prozess neu (Text läuft in das schwarze Feld von rechts ein).

Wird ein Anzeigenspeicher gelöscht (siehe Ausgabesteuerung), so wird die Zuordnung des Prozesses zum entsprechenden Anzeigenspeicher aufgehoben.

Wird ein Anzeigenspeicher zur Anzeige gebracht, werden alle Prozesse, die zu diesem Zeitpunkt keine Zuordnung zu irgendeinem Anzeigenspeicher tragen, gelöscht. Prozesse, die keine Zuordnung zum aktuell auszugebenden Anzeigenspeicher tragen, jedoch andere Zuordnungen, werden lediglich angehalten. Alle Prozesse die bereits laufen und weder gelöscht noch angehalten werden müssen, laufen ohne stocken weiter. Prozesse, die noch nicht laufen, jedoch der aktuell auszugebenden Seite zugeordnet sind, werden gestartet.

3.3.4 Verwendung von Rückfalltexten beim „Timeout“

Auf Kundenwunsch kann die Betriebssoftware des Anzeigenrechners so konfiguriert werden, daß der letzte Anzeigenspeicher (in der Regel Anzeigenspeicher 0x9) in einem batteriegepufferten RAM auf dem Anzeigenrechner abgelegt wird. So besteht die Möglichkeit, Daten in den Speicher zu schreiben, die dann auch nach einem Stromausfall oder einem Reset der Anzeige noch im Anzeigenrechner verfügbar sind. Bekommt der Anzeigenrechner länger als eine vordefinierte Zeit (z.B. 10 Minuten) keine Telegramme mehr von der Leitstelle, gibt er dann automatisch den Inhalt des gepufferten Anzeigenspeichers auf der Anzeige aus.

Eine typische Information für einen solchen Rückfalltext ist z.B. "Außer Betrieb". Lauftexte und Blinken wird auf diesem Anzeigenspeicher nicht unterstützt! Texte mit diesen Feldattributen werden wie mit Feldattribut 0x00 (links und oben bündig) ausgegeben!

3.3.5 Beispiel für ein Telegramm mit Textausgabe:

Telegrammrahmen (Kopf):

0x7E	Rahmenzeichen	
0x68	Startzeichen	
0x1A	Längenbyte = 26 -----	
0x1A	Kopie Längenbyte	
0x68	Startzeichen	
0x43	Steuerbyte: Kennung für Datentelegramm	1
0x01	Adresse des Anzeigenrechners	2
0x00	OSI3-Byte, wird ignoriert	3
0x02	Anzahl der folgenden Datensätze:	4

Datenfelder:

0xB9	Funktionsgruppe		5
0x03	Datensatzlänge = 3 -----		6
0x01	interne Adresse	1	7
0x32	Datenkennung = Fontauswahl:	2	8
0x82	Zeichensatznummer, Spacing, Transparenz	3	9
0x10	Datensatzlänge = 16 -----		10
0x01	interne Adresse	1	11
0x30	Datensatzkennung = Text	2	12
0x01	Textsteuerbyte	3	13
0x0B	horizontale Feldposition	4	14
0x00	"	5	15
0x01	vertikale Feldposition	6	16
0x00	"	7	17
0x5A	Feldbreite	8	18
0x00	"	9	19
0x10	Feldhöhe	10	20
0x00	Feldattribut	11	21
0x04	Textlänge: 4 Zeichen	12	22
0x41	A	13	23
0x42	B	14	24
0x43	C	15	25
0x44	D	16	26

Telegrammrahmen (Abschluß):

0x7D	Austauschzeichen	Prüfsumme = 0x7D, dieser Wert ist jedoch reserviert!
0x5D	0x7D XOR 0x20	
0x16	Stopzeichen	
0x7E	Rahmenzeichen	

3.4 Datensatz für Graphikausgabe

Dieser Datensatz ist nur gültig, wenn als Funktionsgruppe am Anfang des Datenfelds 0xB9 angegeben ist.

Der Datensatz beinhaltet ein Byte für die Ausgabesteuerung, Informationen zur Positionierung der Graphik auf der Anzeige sowie Bitmapdaten.

Alle in der folgenden Tabelle aufgeführten Bytes müssen in der angegebenen Reihenfolge gesendet werden, um eine Graphik an die Anzeige zu senden bzw. diese auf der Anzeige sichtbar zu machen. Alle Zeichen unterliegen, wenn erforderlich, der in Kapitel 2.1 beschriebenen Austauschprozedur.

Funktion:	Wert/Wertebereich:	Bemerkungen:
Datensatzlänge	0x02 ... 0xFB	Die Datensatzlänge gibt die Anzahl der Bytes ab dem nächsten bis einschließlich dem letzten zu diesem Datensatz gehörenden Byte an. Für die maximale Datensatzlänge muß auch die maximale Größe des Datenfeldes (vergl. Kapitel 2.2) beachtet werden.
interne Adresse	0x01	
Datensatzkennung	0x31	Datensatzkennung für Graphikausgabe
Textsteuerbyte	siehe Kapitel 3.3	siehe Kapitel 3.3.1
Horizontale Position	0x0000 ... 0xFFFF	Y-Koordinate der linken oberen Ecke der Graphik (2 Bytes!). Der Koordinatenursprung (0/0) ist in der linken oberen Ecke der Anzeige. Positive Y-Koordinaten führen nach unten. Werden Pixel über den Rand der Anzeige geschrieben, so werden sie entsprechend abgebrochen.
Vertikale Position	0x0000 ... 0xFFFF	X-Koordinate der linken oberen Ecke der Graphik (2 Bytes!). Der Koordinatenursprung (0/0) ist in der linken oberen Ecke der Anzeige. Positive X-Koordinaten führen nach rechts. Werden Zeichen über den Rand der Anzeige geschrieben, so werden sie pixelgenau abgebrochen
Höhe der Graphik in Pixel	0x0000 ... 0xFFFF	Ausdehnung der Graphik in Y-Richtung (2 bytes!).
Breite der Graphik in Pixel	0x0000 ... 0xFFFF	Ausdehnung der Graphik in X-Richtung (2 bytes!).
Pixeldaten (max. Länge: 127 Bytes)		

Alle Pixeldaten einer Zeile werden in aufeinanderfolgenden Bytes übertragen. Jede Zeile beginnt mit einem ganzen Byte. Bit 0 in einem Byte entspricht dem linkesten Pixel der Pixelgruppe, auf welche sich das Byte bezieht, Bit 7 dem rechtesten.

Wenn die Anzahl der auszugebenden Spalten (Breite der Graphik) kein ganzzahliges Vielfaches von 8 ergibt, werden die entsprechenden Bits im letzten Byte jeder Zeile ignoriert.

Beispiel für eine Matrix von 10 Spalten und 3 Zeilen:

Bitwertigkeit:	Nummer des Bytes der Pixeldaten
01234567 01234567	

*****	**-----	0, 1
*****	**-----	2, 3
*****	**-----	4, 5

*: Für die Ausgabe der Graphik relevante Bits

–: Für die Ausgabe der Graphik nicht relevante Bits

3.5 Datensatz zur Ansteuerung des Ausgabeports

Dieser Datensatz steuert eine Leistungsverstärker-Platine, welche auf einem LLE-Stationär-Anzeigenrechner aufgesteckt ist und bis zu 8 Schaltausgänge besitzt. Da sich bis zu 4 Verstärkerplatinen kaskadieren lassen, können bis zu 32 Schaltausgänge bedient werden.

Der Datensatz ist nur gültig, wenn als Funktionsgruppe am Anfang des Datenfelds 0xB9 angegeben ist.

Der Datensatz beinhaltet die Adresse des Schaltausgangs und den digitalen Wert, den der Ausgang annehmen soll.

Alle in der folgenden Tabelle aufgeführten Bytes müssen in der angegebenen Reihenfolge gesendet werden, um einen Ausgang auf einer Leistungsverstärker-Platine auf den gewünschten Wert zu schalten. Alle Zeichen unterliegen, wenn erforderlich, der in Kapitel 2.1 beschriebenen Austauschprozedur.

Funktion:	Wert/Wertebereich:	Bemerkungen:
Datensatzlänge	0x04	Die Länge des Datensatzes ist auf 4 Bytes festgelegt
interne Adresse	0x01	
Datensatzkennung	0x40	Datensatzkennung zur Ansteuerung des Ausgabeports
Portadresse	0x00 ... 0x1F	Adresse des Ports, dessen Wert geändert werden soll
neuer Ausgabewert	0x00 oder 0x01	neuer logischer Zustand des adressierten Ports. Die eigentliche Funktion, welche dem Wert entspricht ist platinenabhängig. In der Regel bedeutet jedoch, daß ein Wert 0x01 den an den Port angeschlossenen Verbraucher aktiviert und ein Wert 0x00 den Verbraucher deaktiviert.

3.6 Datensatz zur Uhrzeit- / Datumsübertragung

Dieser Datensatz korrigiert die Uhrzeit und das Datum auf dem Anzeigenrechner. Da die LLE-Stationär-Anzeigenrechner batteriegepufferte Echtzeituhren besitzen, ist es ausreichend, das Telegramm nur ein mal pro Tag zu senden.

Das Telegramm stellt die Uhr sekundengenau. Bei einem stark belasteten Datenbus kann es jedoch zu Verzögerungen beim Absetzen bzw. beim Auswerten des Telegramms kommen, so daß die Uhrzeit nicht exakt gestellt wird. Es wird daher empfohlen, das Telegramm nur zu Zeiten mit geringer Busbelastung (z.B. in der Nacht) zu senden. Weiterhin sollte das Telegramm nicht im gleichen Rahmen mit anderen Datensätzen verschickt werden, um die Sendezeit zu minimieren!

Unter Umständen ist es sinnvoll, das Telegramm an alle Anzeigen gleichzeitig zu senden (unter Verwendung der Slaveadresse 0xFF).

Der Datensatz ist nur gültig, wenn als Funktionsgruppe am Anfang des Datenfelds 0xB9 angegeben ist.

Alle in der folgenden Tabelle aufgeführten Bytes müssen in der angegebenen Reihenfolge gesendet werden, um Datum und Uhrzeit im Anzeigenrechner sekundengenau zu stellen. Alle Zeichen unterliegen, wenn erforderlich, der in Kapitel 2.1 beschriebenen Austauschprozedur.

Funktion:	Wert/Wertebereich:	Bemerkungen:
Datensatzlänge	0x09	Die Länge des Datensatzes ist auf 9 Bytes festgelegt
interne Adresse	0x01	
Datensatzkennung	0x50	Datensatzkennung zum Korrigieren der Uhrzeit und des Datums auf dem Anzeigenrechner
Jahr	0x0000 ... 0xFFFF	Jahr (0000 bis 65536, 2 Bytes!)
Monat	0x01 ... 0x0C	Monat (1 bis 12)
Tag	0x01 ... 0x1F	Tag (1 bis 31)
Stunden	0x00 ... 0x17	Stunden (0 bis 23)
Minuten	0x00 ... 0x3B	Minuten (0 bis 59)
Sekunden	0x00 ... 0x3B	Sekunden (0 bis 59)

3.7 Datensatz für Wechselverkehrszeichen

Dieser Datensatz ist nur gültig, wenn als Funktionsgruppe am Anfang des Datenfelds 0x04 angegeben ist.

Der Datensatz beinhaltet verschiedene Bytes zur Steuerung der Anzeige sowie Referenznummern zur Anzeige von vordefinierten Texten.

Alle in der folgenden Tabelle aufgeführten Bytes müssen in der angegebenen Reihenfolge gesendet werden, um ein Wechselverkehrszeichen zu steuern. Alle Zeichen unterliegen, wenn erforderlich, der in Kapitel 2.1 beschriebenen Austauschprozedur.

Funktion:	Wert/Wertebereich:	Bemerkungen:
Datensatzlänge	0x02 ... 0xFB	Die Datensatzlänge gibt die Anzahl der Bytes ab dem nächsten bis einschließlich dem letzten zu diesem Datensatz gehörenden Byte an. Für die maximale Datensatzlänge muß auch die maximale Größe des Datenfeldes (vergl. Kapitel 2.2) beachtet werden.
interne Adresse	0x01	
Datensatzkennung	0x32	Datensatzkennung für Datentelegramm Wechselverkehrszeichen
Der weitere Aufbau des Telegramms ist noch in Bearbeitung!		

4. Definition von Antworttelegrammen

Antworttelegramme benutzen aus Effektivitätsgründen nicht den Telegrammrahmen der Aufruftelegramme, sondern haben einen einfacheren Aufbau. Ein Slave sendet nur dann ein Antworttelegramm, wenn er ein für ihn adressiertes Aufruftelegramm korrekt empfangen hat (Auswertung der Längenbytes und der Checksumme, Slaveadresse nicht 0xFF). Ein inhaltlicher Fehler eines Aufruftelegramms wird nicht an die Leitstelle signalisiert!

Alle Zeichen unterliegen, wenn erforderlich, der in Kapitel 2.1 beschriebenen Austauschprozedur.

Ein Antworttelegramm hat folgenden Aufbau:

Funktion:	Wert/Wertebereich:	Bemerkungen:
Rahmenzeichen	0x7E	Dieses Zeichen wird nicht nach dem in Kapitel 2.1 beschriebenen Algorithmus ausgetauscht!
Slaveadresse	0x01 ... 0xFE	Über dieses Byte identifiziert sich der Sender des Antworttelegramms (Slave). Die Slaveadresse kann in der Anzeige auf dem Anzeigenrechner eingestellt werden. Die Slaveadressen 0x00 und 0xFF sind reserviert.
Statusbyte	0x00: 0x0F:	Status des Slave: Letztes Aufruftelegramm wurde korrekt empfangen, Anzeigenrechner O.K. Letztes Telegramm wurde korrekt empfangen. Anzeigenrechner wurde seit dem Empfang des vorletzten Aufruftelegramms neu gestartet, der Rechner hatte daher keine Daten mehr in den Anzeigenspeichern verfügbar (Ausnahme der letzte Anzeigenspeicher, der auf Kundenwunsch in ein Batteriegepuffertes RAM gelegt werden kann). Der Status wird nach dem ersten Versenden zurückgesetzt.
Prüfsumme	0x00 ... 0xFF	Die Prüfsumme wird durch Addition der Slaveadresse und des Statusbytes modulo 0x100 gebildet.
Rahmenzeichen	0x7E	Dieses Zeichen wird nicht nach dem in Kapitel 2.1 beschriebenen Algorithmus ausgetauscht!