

# Implementierung einer Zugnummern-Meldeanlage für eine FREMO-Betriebsstelle

Bodo Mertins, 17.11.2013 (Vorläufige Version!)

## 0. Einleitung

Es entsteht derzeit vermehrter Informationsbedarf über die Zugnummern-Meldeanlage des FREMO. Es gibt einige Interessenten, die ihre Betriebsstellen an dieses System anbinden möchten und noch nicht wissen, wie das zu bewerkstelligen ist. Insbesondere an diese Personen richtet sich dieser Text. Es wird die Beherrschung einer beliebigen Programmiersprache und die Fähigkeit zu einfacher Netzwerkprogrammierung vorausgesetzt. Ich werde hier nicht auf technische Details wie Stellwerks-Anzeigen oder Eingabemöglichkeiten eingehen.

Je nach Stellwerks-Architektur, kann es einige Unterschiede geben, so ist z.B. bei einem ESTW die Ein- und Ausgabe integriert, während es in anderen Stellwerken getrennte Anzeigen und ein Eingabe-Terminal (DAT/EKA) gibt. Wenn ich also von Eingaben schreibe, kann das auf die eine oder andere Art geschehen.

Auch benötigt eine Zugnummern-Meldeanlage Informationen über die Betriebsstelle, mindestens die Stellung der Hauptsignale und der wichtigsten Weichen. Diese Informationen sind in einem rechnerbasierten Stellwerk schon vorhanden und müssen bei anderen Stellwerken irgendwie (darauf gehe ich nicht weiter ein!), z.B. per Busabfrage zugeführt werden.

Interne Details, wie z.B. Blinkflags wieder zurückzusetzen, habe ich der Übersichtlichkeit weggelassen!

Die Textausgabe am Eingabe-Terminal und am Zugnummerndrucker spare ich zunächst noch aus!

**Ich bitte um Kritik, Ergänzungen und Korrekturen der Wissenden...**

## 1. Zweck der Zugnummern-Meldeanlage

Die FREMO-Zugnummern-Meldeanlage dient der Beschleunigung der Abwicklung von Zugfahrten zwischen den Betriebsstellen.

**Voraussetzung für den Einsatz ist ein funktionierender Streckenblock!**

Durch den Einsatz der Blocktechnik entfallen die Telefongespräche zum Rückmelden der Züge. Durch die Zugnummern-Meldeanlage entfallen auch noch die Anbietet-/Annahme- und die Abmelde-Gespräche. Es wird eigentlich nur noch im Störfall telefoniert. Dadurch ergibt sich ein immenser Zeitvorteil in der Zugfolge, denn abgesehen von den Wartezeiten bis zum Abnehmen - wer kann den schon in z.B. sechsfacher Geschwindigkeit telefonieren ;-)

## 2. Aufbau der Zugnummern-Meldeanlage

### 2.1 Zugnummernrechner

Der Aufbau der ZN-Anlage ist einfach, jede Betriebsstelle besitzt einen Zugnummernrechner. Das kann ein eigener Rechner sein oder vom ggfs. vorhandenen Stellwerksrechner übernommen werden. Der ZN-Rechner meldet sich über eine Netzwerkverbindung an einem Server an, der den ZLV-Bus (Zug-Lenk und Verfolgungs-Bus) simuliert.

Dazu mehr in **Anlage 1**. Prinzipiell werden von den Betriebsstellen Info-Telegramme auf den ZLV-Bus gesendet, die dann von allen anderen Betriebsstellen empfangen und ausgewertet werden.

### 2.2 Unterstationsnummer

Da auf dem ZLV-Bus unterschieden werden muss, welche Betriebsstelle ein Telegramm gesendet hat und es außerdem für die Telegramm-Inhalte ein eindeutiges Bezeichnungs-System geben muss, bekommt jede Betriebsstelle eine eigene Unterstationsnummer von 01 bis 99.

Diese Nummern werden von Moritz Hebert verwaltet und vergeben.

### 2.3 Zugnummernfelder

#### 2.3.1 Allgemeines

Die wichtigsten Strukturen einer ZN-Anlage sind "Felder". Diese Felder beinhalten Informationen (i.d.R. Zugnummern), und tauschen über den Bus deren Inhalt mit anderen Betriebsstellen aus.

Felder sind, abstrakt gesehen, Anzeigemöglichkeiten für ZLV-Bus-Telegramme, sie bestehen aus einem Feldtyp (siehe 2.3.2), einer Bezeichnung (siehe 2.3.3) und einem Inhalt (siehe 2.3.4). Mehrere Betriebsstellen können ein Feld verwenden, d.h. nichts anderes, als ein ZLV-Bus-Telegramm mit dem Inhalt für das Feld von mehreren Betriebsstellen verarbeitet und angezeigt wird, dazu später mehr.

#### 2.3.2 Felder-Typen

Es gibt prinzipiell vier Arten von Feldern, die im jeweiligen Stellwerk der Betriebsstelle angezeigt werden sollten.

1. **Gleisfelder**: Jedes Hauptgleis einer Betriebsstelle sollte ein Gleisfeld bekommen. Diese Felder zeigen, welcher Zug sich gerade in welchem Gleis befindet (oder gar keiner). Es können Eingaben und Löschungen erfolgen. Außerdem können die Nummern in benachbarte Felder "fortgeschaltet" werden (siehe ###).

2. **Spiegelfelder**: Diese Felder sind Gleisfelder einer anderen (meist benachbarten) Betriebsstelle. Hier sollten keine Veränderungen möglich sein, sondern es wird nur beobachtet, was in der anderen Betriebsstelle vorgeht. Wenn man eine Bereichsübersicht programmiert, auf der viele Betriebsstellen dargestellt werden, gibt es nur Spiegelfelder!

3. **Anbietefelder**: Über diese Felder wird die Zugfolge geregelt (Anbieten/Annahmen). Sie liegen normalerweise neben den Gleisfeldern für Streckengleise bei eingleisigen Strecken oder Gleiswechselbetrieb. Bei zweigleisigen Strecken im reinen Richtungsbetrieb kommen diese Felder normalerweise nicht vor.

4. **Voranzeigefelder**: Diese Felder liegen normalerweise neben den

Gleisfeldern der Streckengleise bei zweigleisigen Strecken im Richtungsbetrieb und dienen der Vorplanung des Fahrdienstleiters. Sie zeigen die Zugnummer, die als nächstes im zugehörigen Streckengleis auftauchen wird, wie das funktioniert, wird später erklärt. Bei eingeleisigen Strecken ist der Nutzen dieser Felder gering, da man einen vergleichbaren Informationsgehalt auch den Anbietefeldern entnehmen kann. Verboten sind sie aber nicht!

### 2.3.3 Bezeichnung der Felder

Alle Felder haben eine systemweit eindeutige Bezeichnung. Diese ist eine sechsstellige Nummer, zuerst die zweistellige Unterstationsnummer der Betriebsstelle, zu der das Feld gehört, dann eine vierstellige freie Nummer, die z.B. die Gleisnummer darstellt.

In der Realität kann die vierstellige Nummer auch noch führende Leerzeichen enthalten, man könnte Gleis "420042" von "42 42" unterscheiden. Da wir nicht wissen, ob alle Betriebsstellen derzeit damit klarkommen sollte auf diese Option verzichtet werden, also immer "420042" verwenden!

Es gibt Felder, auf die von zwei Betriebsstellen zugegriffen wird, z.B. die Streckengleise zwischen den Betriebsstellen. In der Realität und in den früheren FREMO-Implementierungen wurde sich vor einem Treffen auf eine Nummer geeinigt. Dabei hat dann eine Betriebsstelle quasi auf ein Feld mit fremder Unterstationsnummer zugegriffen. Diese Option funktioniert immer, es hat sich jedoch bei neueren Implementierungen durchgesetzt, dass diese Gleise zwei Bezeichnungen haben, die eigene und die des Nachbarn.

Zu einem Streckengleisfeld gehörende Anbieter- oder Voranzeigefelder können (müssen aber nicht) die gleiche Bezeichnung wie das Gleisfeld haben. In der Realität ist es meiner Erfahrung nach meistens die gleiche Nummer, es gibt aber beide Varianten! Wenn es z.B. bei einer Strecke noch Blocksignale gibt, kann es aus Sicht mindestens einer beteiligten Betriebsstelle nicht die gleiche Bezeichnung sein. Die Bezeichnung der Felder darf aus dem Grund gleich sein, weil aus dem Telegramm immer eindeutig hervorgeht, ob es für ein Gleis- ein Anbieter- oder ein Vorschaufeld gilt.

Beispiel: Wir betrachten zwei Betriebsstellen mit den Unterstationsnummern 42 und 43 (im FREMO sind sie meistens nicht fortlaufend, da Arrangementabhängig!)

#### Aus Sicht von Unterstation 42:

Streckengleis	420100	(alternativ 430020)
Anbietefeld	420100	(alternativ 430021)
Voranzeigefeld	420100	(alternativ 430022)

#### Aus Sicht von Unterstation 43:

Streckengleis	430020	(alternativ 420100)
Anbietefeld	430021	(alternativ 420100)
Voranzeigefeld	430022	(alternativ 420100)

Wenn nur eine Bezeichnung verwendet werden soll, muss man sich vorher

mit dem Nachbarn einigen! Besser ist, alle Beteiligten veröffentlichen nur ihre Bezeichnungen und die Doppelvariante wird verwendet.

Beide Betriebsstellen müssen dabei beide Namen kennen und auf dem ZLV-Bus ankommende Telegramme mit beiden Bezeichnungen akzeptieren. Beim Senden von Telegrammen wird jeweils nur die "eigene" Bezeichnung verwendet! Damit haben wir eine symmetrische Bezeichnung.

## 2.3.4 Feldinhalte

Der Inhalt eines Feldes ist immer (maximal) sechsstellig. Es gibt es prinzipiell drei Möglichkeiten:

### 2.3.4.1 Zugnummern

Zugnummern stehen rechtsbündig, vorne mit Leerzeichen aufgefüllt. Würde man mit Vornullen auffüllen, wäre das ein rein optischer Unterschied.

Zugnummern sind maximal 5-stellig. Die sechste (linke) Stelle ist die Zuglenkziffer, die bei Stellwerken mit Zuglenkung verwendet werden kann, um z.B. automatische Gleiswahl zu gewährleisten.

Bei der Auswertung ankommender ZLV-Telegramme müssen Zugnummern aus dem Telegramm mit den gespeicherten Inhalten verglichen werden (dazu später mehr).

Alle Zugnummern mit dem gleichen Vergleichswert gelten als "gleich".

Die folgende Tabelle enthält jeweils Zugnummern in mehreren Formaten und ihre Wirkungen auf Anzeige und Zuglenkziffer:

Gespeicherte bzw. übertragene Zug-Nr.	Anzeige	Vergleichswert	Lenkziffer
0 (rein theoretisch!)	0	00000	keine
000 (rein theoretisch!)	0	00000	keine
000000 (rein theoretisch!)	0	00000	0
0 0 (rein theoretisch!)	0	00000	0
4 0 (rein theoretisch!)	0	00000	4
400000 (rein theoretisch!)	0	00000	4
(Leeres Feld)		Leer	keine
1	1	00001	keine
001	1	00001	keine
000001	0 1	00001	0
0 1	0 1	00001	0
400001	4 1	00001	4
4 1	4 1	00001	4
12	1	00012	keine
012	1	00012	keine
000012	0 1	00012	0
0 12	0 1	00012	0
400012	4 1	00012	4
4 12	4 1	00012	4
123	123	00123	keine
0123	123	00123	keine
000123	0 123	00123	0
0 123	0 123	00123	0
400123	4 123	00123	4
4 123	4 123	00123	4
1234	1234	01234	keine
001234	0 1234	01234	0
0 1234	0 1234	01234	0
401234	4 1234	01234	4
4 1234	4 1234	01234	4
12345	12345	12345	keine
012345	0 12345	12345	0
412345	4 12345	12345	4

Anmerkung: Es wäre theoretisch auch ein "L" als Zuglenkziffer möglich (LZB-geführter Zug), hat das schon jemand?

Für Zugnummern gibt es ein wichtiges Grundprinzip: Es kann nur einen geben! D.h. eine Zugnummer darf im ganzen System in allen Betriebsstellen nur in einem Gleisfeld auftreten, Anbiete- und Voranzeigefelder sind davon ausgenommen. Wenn also ein Zug in ein Gleis eingegeben wird, werden alle anderen Betriebsstellen davon über den ZLV-Bus in Kenntnis gesetzt und löschen die Nummer, falls sie bei sich noch irgendwo vorkommt. Bei einer Zugnummernfortschaltung gibt es nur ein Telegramm für das neue Feld. Das Löschen der Nummer im bisherigen Feld erfolgt immer implizit / automatisch, es gibt in diesem Fall kein Lösch-Telegramm!!!!

#### 2.3.4.2 Fehlernummern

Es gibt einen Sonderfall für Zugnummern: Falls bei einer Zugnummernfortschaltung keine Nummer im Startfeld eingetragen ist, erzeugt der ZN-Rechner eine Fehlernummer.

Diese hat keine Zuglenkziffer (Leerzeichen), ein "F" für Fehler, die 2-stellige Unterstationsnummer und eine 2-stellige Fehlerzählernummer, die danach hochgezählt wird und nach 99 wieder bei 00 anfängt (z.B. F4201). Diese Nummern werden dann wie normale Zugnummern weiterverarbeitet.

Als Sonder-Sonderfall gilt, dass eine neue Fehlernummer im ersten Feld, in dem sie erzeugt wurde, blinkend dargestellt wird!

#### 2.3.4.3 Verfügbarkeitshinweise

Es gibt bei Gleisfeldern **zusätzlich** zur Zugnummer die Möglichkeit einen von drei Verfügbarkeitshinweisen zu speichern:

00000 bedeutet "Gleis gesperrt"

0F0F0 bedeutet "Gleis nicht verfügbar" oder "belegt ohne Zugnummer"

FFFFF bedeutet "Gleis für elektrische Traktion gesperrt"

Verfügbarkeitshinweise werden nie fortgeschaltet. Wenn in einem Gleisfeld ein Verfügbarkeitshinweis und eine Zugnummer gespeichert sind, wird der Verfügbarkeitshinweis von der Zugnummer verdeckt. Ist die Zugnummer gelöscht oder fortgeschaltet, wird der Verfügbarkeitshinweis wieder sichtbar!

### 2.3.5 Funktionen des ZN-Rechners

#### 2.3.5.1 Empfang und Verarbeitung von ZLV-Telegrammen

Der ZN-Rechner bekommt Telegramme vom ZLV-Bus und verarbeitet diese, darin könnte z.B. die Information stecken, dass im Nachbarbahnhof ein Zug ausgefahren ist und dessen Zugnummer jetzt in das Streckengleisfeld eingetragen werden muss (siehe ###).

#### 2.3.5.2 Eingaben des Fahrdienstleiters verarbeiten und als ZLV-Telegramm senden

Der Fahrdienstleiter muss Eingaben in Felder oder Löschungen aus Feldern durchführen können. Sind die Eingabemöglichkeiten ins Stellwerk integriert (z.B. ESTW), dann wird die eingegebene Zugnummer direkt in das zugehörige Feld eingetragen **und** ein ZLV-Bus-Telegramm gesendet, das alle

anderen Betriebsstellen davon informiert (siehe ###).

### 2.3.5.3 Überwachung der Fortschaltkriterien und Erzeugung der Fortschalttelegramme

Der eigentliche Sinn der ZN-Anlage ist die automatische Fortschaltung von Zugnummern von einem Gleisfeld in das nächste auf seinem Weg. Um das zu erreichen, bekommt der ZN-Rechner (wie das auch immer realisiert sein mag) Informationen über die Stellung der Hauptsignale und der wichtigsten Weichen. Fällt ein Hauptsignal von einem Zugfahrtbegriff aus Halt, ist das der Auslöser für eine Fortschaltung. Wird ein Signal vom Bediener zurück auf Halt gestellt, müssen Vorkehrungen getroffen werden, damit die Fortschaltung verhindert wird. Beispielsweise wird bei Betätigung der HaGT ein kurzer Timer gestartet, der Fortschaltungen unterbindet. Rechnerbasierte Stellwerke haben die Information i.d.R implizit.

Es gibt auch noch einige Ausnahme-Situationen für Fortschaltkriterien. Bei Zugdeckungssignalen etwa wird die Fortschaltung anhand der Gleisbesetzmeldungen erzeugt, da es keinen Haltfall gibt. Näher will ich darauf hier nicht eingehen.

Wenn nun eine nötige Fortschaltung erkannt wird, verfolgt der ZN-Rechner in seinem Modell der Betriebsstelle den Fahrweg vom Signal rückwärts bis zum ersten Gleisfeld und vorwärts bis zum nächsten Gleisfeld hinter dem Signal. Das kann anhand eines Spurplans oder tabellenbasiert geschehen. Damit sind die Informationen über die Quelle und das Ziel der Fortschaltung vorhanden. Jetzt gibt es noch drei Fälle zu unterscheiden:

1. das Quell-Gleisfeld ist leer. In diesem Fall wird im Ziel-Gleisfeld eine Fehlernummer erzeugt (siehe ###).
2. das Quell-Gleisfeld ist gefüllt aber das Ziel-Gleisfeld enthält schon eine Nummer die überschrieben wird. In diesem Fall wird ein "Überschreiben"-Telegramm erzeugt (siehe ###).
3. das Quell-Gleisfeld ist gefüllt, das Ziel-Gleisfeld ist leer (Regelfall!) In diesem Fall wird ein normales Fortschaltungs-Telegramm erzeugt (siehe ###).

Mit der Erzeugung und dem Senden der Telegramme werden auch die internen Feldinhalte verändert.

## 3. Die Telegrammtypen und deren Verarbeitung bzw. Erzeugung

In diesem Kapitel werde ich quasi Rückwärts die Telegramme und deren Verwendung erklären. Auf das genaue Format (in Abhängigkeit vom Server) gehe ich erst in Anlage 2 ein!

Telegramme werden normalerweise gesendet, wenn der ZN-Rechner aufgrund einer Bedienhandlung oder einer Fortschaltung Feldinhalte verändern möchte.

Wenn ein Telegramm intern (durch Eingabe oder Fortschaltung) erzeugt wird, wird es gesendet und derselben Routine übergeben, die empfangene Telegramme verarbeitet.

Bei der Verwendung einer getrennten DAT/EKA geschieht erfolgt dies

automatisch durch "echten" Empfang.

Wenn ein Telegramm vom ZLV-Bus empfangen wird, ist es sinnvoll, es in einer Schleife o.ä. allen Feldern zur Verarbeitung zur Verfügung zu stellen. Jedes Feld kann aus Telegrammtyp und -inhalt eindeutig für sich entscheiden, ob es das Telegramm verarbeiten muss!!! Diese Vorgehensweise wird in den folgenden Kapiteln vorausgesetzt, alle Massnahmen in den "Empfangen"-Kapiteln müssen für alle Felder des jeweiligen Typs erfolgen.

Jedes Telegramm enthält die Unterstationsnummer der sendenden Betriebsstelle (2-stellig) und den Telegrammtyp (2-stellig). Weitere Parameter variieren.

Anhand der Unterstationsnummer kann entschieden werden, ob das Telegramm selbst erzeugt wurde. Es gibt Fälle, wo das von Interesse ist.

Ich werde teilweise Pseudocode in Annäherung von C verwenden!

Achtung: die Numerierung der Kapitel ist hier nicht mehr fortlaufend, sondern die zweite Zahl enthält den Telegrammtyp der Spezifikation, um besser nachschlagen zu können.

Es werden nicht alle möglichen Telegrammtypen verwendet!

Hier werden ab jetzt folgende Abkürzungen verwendet:

Telegramminhalte:

UST = Unterstationsnummer

TYP = Telegrammtyp

ZN = Zugnummer

FN = Fehlernummer

ZN/FN Kann Zugnummer oder Fehlernummer enthalten

VH = Verfügbarkeitshinweis

GNR = Gleisfeld-Nummer in einem Telegramm

ANR = Anbietefeld-Nummer in einem Telegramm

VNR = Voranzeigefeld-Nummer in einem Telegramm

Variablen eines Feldes:

GBNR = Gleisfeld-Bezeichnungs-Nummer

GMEM = Gleisfeld-ZN/FN-Speicher

GVHMEM = Gleisfeld-VH-Speicher

ABNR = Anbietefeld-Bezeichnungs-Nummer

AMEM = Anbietefeld-ZN-Speicher

VBNR = Voranzeigefeld-Bezeichnungs-Nummer

VMEM = Voranzeigefeld-ZN/FN-Speicher

Außerdem definiere ich als

"Gleistreffer", wenn die GNR des Telegrammes mit der GBNR des

verarbeitenden Gleisfeldes übereinstimmt  
 "Anbietetreffer", wenn die ANR des Telegrammes mit der ABNR des  
 verarbeitenden Anbietefeldes übereinstimmt  
 "Voranzeigetrefeffer", wenn die VNR des Telegrammes mit der VBNR des  
 verarbeitenden Voranzeigefeldes übereinstimmt  
 "Inhalttreffer", wenn die ZN/FN des Telegrammes mit dem GMEM, AMEM  
 oder VMEM des verarbeitenden Gleis- Anbierte- oder Vorschaufeldes  
 übereinstimmt (**siehe 2.3.4.1!!!**)

```

void Gleisfeld_Setze_Inhalt(ZN_FN_VH)
{ // 5-stellig ist richtig - sicherheitshalber tolerant!
  if (ZN_FN_VH==" 00000" || (ZN_FN_VH=="000000"))
    GVHMEM=" 00000";
  else if ((ZN_FN_VH==" 0F0F0" || (ZN_FN_VH=="0F0F0F"))
    GVHMEM=" 0F0F0";
  else if ((ZN_FN_VH==" FFFFF" || (ZN_FN_VH=="FFFFFF"))
    GVHMEM=" FFFFF";
  else
    GMEM=ZN_FN_VH;

  if (korrespondierendes_Anbietefeld_vorhanden)
  { // Wenn der angebotene Zug im Gleis steht, Anbietefeld löschen
    if (ZN_FN_VH==AMEM)
      AMEM=" ";
  }

  if (korrespondierendes_Anbietefeld_vorhanden)
  { // Wenn das Gleis zu einem Voranzeigefeld besetzt wird, löschen
    if (VMEM!=" ")
      SendeTelegramm(UST, "06", VMEM, VBNR); Sicherheitshalber
      VMEM=" ";
  }
}

void Gleisfeld_Loesche_Inhalt()
{
  Gleisfeld_Setze_Inhalt(" ");
  switch (GVHMEM)
  { // VH ggfs. wieder sichtbar machen!
    case " 00000":SendeTelegramm(UST, "10", " 00000", GBNR);break;
    case " 0F0F0":SendeTelegramm(UST, "10", " 0F0F0", GBNR);break;
    case " FFFFF":SendeTelegramm(UST, "10", " FFFFF", GBNR);break;
  }
}

```

## 3.02 Regel-Fortschaltungs-Telegramm

**3.02.1 Parameter:** UST, TYP=02, ZN/FN, GNR

**3.02.2 Senden, wenn...**

eine Fortschaltung von einem besetzten Quellfeld auf ein freies Zielfeld (auch wenn dieses evtl. einen Verfügbarkeitshinweis enthält) erfolgt. Das Quellfeld wird gelöscht, ein evtl. vorhandener VH wird ggfs. mit einem weiteren gesendeten Telegramm (TYP=10) wieder sichtbar gemacht (siehe 3.10)!

**3.02.3 Empfangen, für alle Gleisfelder...**

```

if (Gleistrefeffer)
  Gleisfeld_Setze_Inhalt(ZN_FN);
else if (Inhalttreffer)
  Gleisfeld_Loesche_Inhalt();

```

## 3.03 Fehlernummer-Erzeugungs-Telegramm

**3.03.1 Parameter:** UST, TYP=03, FN, GNR

**3.03.2 Senden, wenn...**

eine Fortschaltung von einem leeren Quellfeld auf ein freies oder besetztes Zielfeld erfolgt. Es wird im Zielfeld eine (blinkende) Fehlernummer erzeugt!

### 3.03.3 Empfangen, für alle Gleisfelder...

```
if (Gleistreffer)
    Gleisfeld_Setze_Inhalt(FN);
else if (Inhaltstreffer)
    Gleisfeld_Loesche_Inhalt();
```

## 3.04 Überschreib-Fortschaltungs-Telegramm

**3.04.1 Parameter:** UST, TYP=04, ZN/FN, GNR

### 3.04.2 Senden, wenn...

eine Fortschaltung von einem leeren Quellfeld auf ein besetztes Zielfeld erfolgt. Das Quellfeld wird gelöscht, ein evtl. vorhandener VH wird ggfs. mit einem weiteren gesendeten Telegramm (TYP=10) wieder sichtbar gemacht (siehe 3.10)!

### 3.04.3 Empfangen, für alle Gleisfelder...

```
if (Gleistreffer)
    Gleisfeld_Setze_Inhalt(ZN_FN);
else if (Inhaltstreffer)
    Gleisfeld_Loesche_Inhalt();
```

## 3.05 Voranzeige-Telegramm

**3.05.1 Parameter:** UST, TYP=05, ZN/FN, VNR

### 3.05.2 Senden, wenn...

ein Signal auf Fahrt geht, dessen Haltfall zu einer Fortschaltung führt. Dabei wird die Fortschaltung im Voranzeigefeld "vorhergesagt".

### 3.05.3 Empfangen, für alle Voranzeigefelder...

```
if (Voranzeigetreffer)
    VMEM=ZN_FN;
```

## 3.06 Voranzeige-Rücknahme-Telegramm

**3.06.1 Parameter:** UST, TYP=06, ZN/FN, VNR

### 3.06.2 Senden, wenn...

ein Signal auf Halt geht (manuell oder automatisch), dessen Fahrtstellung zu einem Voranzeigetelegramm geführt hat.

### 3.06.3 Empfangen, für alle Voranzeigefelder...

```
// Würde wohl auch ohne "Inhaltstreffer" gehen...
if (Voranzeigetreffer && Inhaltstreffer)
    VMEM="";
```

## 3.10 Einwahl/Eingabe-Telegramm

**3.10.1 Parameter:** UST, TYP=10, ZN/FN/VH, GNR

### 3.10.2 Senden, wenn...

eine Benutzereingabe in ein Gleisfeld erzeugt. Das kann eine Zugnummer oder ein Verfügbarkeitshinweis sein.

### 3.10.3 Empfangen, für alle Gleisfelder...

```
if (Gleistreffer)
    Gleisfeld_Setze_Inhalt(ZN_FN_VH);
else if (Inhaltstreffer)
    Gleisfeld_Loesche_Inhalt();
```

## 3.20 VH-Löschen-Telegramm

**3.20.1 Parameter:** UST, TYP=20, VH, GNR

**3.20.2 Senden, wenn...**

eine VH durch den Benutzer gelöscht wird.

**3.20.3 Empfangen, für alle Gleisfelder...**

```
if (Gleistreffer && Inhaltstreffer)
    GVHMEM="    ";
```

## 3.21 ZN/FN-Löschen-Telegramm

**3.21.1 Parameter:** UST, TYP=21, ZN/FN

**3.21.2 Senden, wenn...**

eine Benutzereingabe in ein Gleisfeld erzeugt. Das kann eine Zugnummer oder ein Verfügbarkeitshinweis sein.

**3.21.3 Empfangen, für alle Gleisfelder...**

```
if (Inhaltstreffer) // Ohne Prüfung der GBNR
    Gleisfeld_Loesche_Inhalt();
```

## 3.30 ZN/FN-Löschen-Telegramm

**3.30.1 Parameter:** UST, TYP=30, neue ZN, bisherige ZN/FN

**3.30.2 Senden, wenn...**

eine ZN/FN durch eine neue ZN ersetzt werden soll, unabhängig davon, wo der Zug genau steht (z.B. nach Richtungswechsel oder wenn ein Fortschaltfehler korrigiert werden soll).

**3.30.3 Empfangen, für alle Gleisfelder...**

```
if (Bisherige_ZN_FN==GMEM)
    GMEM=Neue_ZN;
else if (Neue_ZN==GMEM)
    Gleisfeld_Loesche_Inhalt();
```

## 3.51 Anbieten-Telegramm

**3.51.1 Parameter:** UST, TYP=51, ZN, ANR

**3.51.2 Senden, wenn...**

ein Zug der Nachbarbetriebsstelle angeboten werden soll.

**3.51.3 Empfangen, für alle Anbietefelder...**

```
if (Anbietetreffer)
{
    AMEM=ZN;
    Blinken_Einschalten();
    if (UST==EingeneUST)
        SelbstAngeboten_Merker_setzen();
    else
        Summer();
}
```

## 3.52 Annehmen-Telegramm

**3.52.1 Parameter:** UST, TYP=52, ZN, ANR

**3.52.2 Senden, wenn...**

ein angebotener Zug von der Nachbarbetriebsstelle angenommen werden soll (Nur wenn nicht selbst angeboten!!!).

**3.52.3 Empfangen, für alle Anbietefelder...**

```
if (Anbietetreffer && Inhaltstreffer &&
```

```

(!SelbstAngeboten)
{
    AMEM=ZN; // Sollte wohl auch ohne gehen...
    Blinken_Ausschalten();
    SelbstAngeboten_Merker_loeschen();
}

```

## 3.53 Anbietefeld-Löschen-Telegramm

**3.53.1 Parameter:** UST, TYP=53, ZN, ANR

**3.53.2 Senden, wenn...**

ein angebotener Zug von der Nachbarbetriebsstelle abgelehnt werden soll ("... Nein warten!") oder, wenn ein Angebot zurückgenommen werden soll ("... kommt nicht!").

**3.53.3 Empfangen, für alle Anbietefelder...**

```

if (Anbietetreffer && Inhaltstreffer)
{
    AMEM="          ";
    Blinken_Ausschalten();
    SelbstAngeboten_Merker_loeschen();
}

```

## 3.92 Anmelde-Telegramm

**3.92.1 Parameter:** UST, TYP=92, Bahnhofs-Kürzel

**3.92.2 Senden, wenn...**

eine Betriebsstelle sich am Server anmeldet. Wird beim neuen Server durch ein "ID:"-Paket ersetzt

**3.92.3 Empfangen, Ignorieren!**

# Anlage 1: Kommunikation mit dem Server

Es gibt beim FREMO zwei ZN-Server, eine Konsolenanwendung und den neuen Server mit Informations-Oberfläche zum Debuggen.

Der alte Server wurde Defaultmäßig auf dem TCP-Port 50099 angesprochen. Es war nötig, vor einem Treffen festzulegen auf welchem Rechner bzw. unter welcher IP er laufen würde.

Der neue Server ist das Ergebnis von Problemen, die bei einem Treffen aufgetreten sind. Es gab eine Version eines ZN-Clients, die sich nicht an die übliche Länge der Pakete gehalten hat. Ein toleranter Hilf-Proxi zwischen Client und Server konnte das Problem dann umgehen. Da das keine Dauerlösung sein konnte, habe ich den neuen Server geschrieben, der einerseits den alten Server einschliesslich des Proxys ersetzen kann, andererseits einen zweiten Port für ein etwas abgewandeltes Protokoll unterstützt. Außerdem kann dieser Server per UDP im Netzwerk gefunden werden. Damit ergibt sich die Möglichkeit auch mit einem Router bzw. DHCP-Server die IPs automatisch zu verteilen.

Für beide Server hat sich ein automatisches timergesteuertes Reconnect

bewährt!

### **Algorithmus zum (neuen) ZN-Server suchen/finden:**

UDPBroadcast("255.255.255.255",57111,"ZNSERVER?")

Wenn ein Antwortpaket mit "ZNSERVERPORT:" beginnt, folgt ohne Trennzeichen der TCP-Port des ZN-Servers im ASCII-Format. Die IP-Adresse des Servers ist implizit im Header enthalten.

Beispiel:

Gesendet "ZNSERVER?"  
Empfangen "ZNSERVERPORT:50100"

Weitere Pakete werden ignoriert, es sollte nur einen Server geben!  
Es wird nur der Port für das neue Format übertragen (Derzeit fest 50100)

Ursprünglich wollte ich auch die IP-Adresse als ASCII-Text übertragen, davon bin ich abgegangen, da ein System mehr als eine IP-Adresse haben kann

und ich nicht entscheiden kann/will, welche gesendet werden muss.

Wenn der Server jedoch auf das Broadcast-Paket antwortet, sende ich automatisch die richtige Adresse!

Diese Header-Daten sind funktional so wichtig, dass ich davon ausgehe, dass jede Programmiersprache den Zugriff darauf ermöglicht!

### **Anmeldung am ZN-Server:**

Für beide Server gilt: Nach Öffnen der TCP-Verbindung wird ein Anmelde-Paket gesendet.

Beim alten Server ist das UST(2-stellig)+"9201"+Bahnhofsname (max. 14 Stellen). Das Paket wird mit Leerzeichen auf 20 Stellen aufgefüllt, es gibt keinen Zeilentrenner!

Beim neuen Server wird ein "ID:"+ Bahnhofskürzel (3-Stellig) +","+Bahnhofsname (Länge egal aber vernünftig bleiben!)+<CR><LF> gesendet

### **Paketformate zum/vom ZN-Server:**

Die Kommunikation mit dem alten Server erfolgt über die ZLV-Telegramme in Anlehnung an die Originalspezifikation der Bahn.

Dabei werden Telegramme zum Server mit 20 Stellen (hinten mit Leerzeichen auffüllen) **ohne Zeilentrenner** gesendet!

Die Telegramme vom Server haben 20 Stellen <CR><LF>!

Beim neuen Server wurde das für beide Richtungen vereinheitlicht:

Die ursprünglichen Telegramme werden in ein übergeordnetes Protokoll eingebettet, das für die meisten Pakete so aussieht:

"ZLV:"+Telegramm+<CR><LF>, wobei die Länge des Telegramms nach Bedarf variieren darf, üblich sind 16-20 Byte.

Die Regeln des neuen Protokolls sind einheitlich:

- Keine vorgegebene Paketlänge
- Jedes Paket wird mit <CR><LF> abgeschlossen (Eingang und Ausgang)
- Jedes Paket geht an alle Clients außer dem Sender
- Jedes Paket beginnt mit einem Pakettyp, der mit Doppelpunkt von den Nutzdaten getrennt wird:
  - ZLV: für ZN-Bus-Pakete
  - UHR: für Uhrzeit-Pakete
  - ID: für ein Begrüßungspaket (bisher Telegramm 92) (Beispiel: ID:WTL,Welsetal)

Ein Beispielpaket für eine die Eingabe eines Zuges 4711 in Gleis 201 in Welsetal sähe dann so aus:

ZLV:0410 4711040201<CR><LF>

## Anlage 2: Liste der Paketformate

In dieser Liste wird nur das neue Format aufgeführt!

UST ist die 2-stellige Unterstationsnummer

ZN / FN / VH sind jeweils 6-stellig

GBNR / ABNR / VBNR sind jeweils 6-stellig

Telegrammtyp 02:	"ZLV: "+UST+"02"+ZN/FN	+GBNR	+<CR><LF>
Telegrammtyp 03:	"ZLV: "+UST+"03"+FN	+GBNR	+<CR><LF>
Telegrammtyp 04:	"ZLV: "+UST+"04"+ZN/FN	+GBNR	+<CR><LF>
Telegrammtyp 05:	"ZLV: "+UST+"05"+ZN/FN	+VBNR	+<CR><LF>
Telegrammtyp 06:	"ZLV: "+UST+"06"+ZN/FN	+VBNR	+<CR><LF>
Telegrammtyp 10:	"ZLV: "+UST+"10"+ZN/FN/VH	+GBNR	+<CR><LF>
Telegrammtyp 20:	"ZLV: "+UST+"20"+VH	+GBNR	+<CR><LF>
Telegrammtyp 21:	"ZLV: "+UST+"21"+ZN/FN	+GBNR	+<CR><LF>
Telegrammtyp 30:	"ZLV: "+UST+"30"+ZN-neu	+ZN/FN-alt	+<CR><LF>
Telegrammtyp 51:	"ZLV: "+UST+"51"+ZN	+ABNR	+<CR><LF>
Telegrammtyp 52:	"ZLV: "+UST+"52"+ZN	+ABNR	+<CR><LF>
Telegrammtyp 53:	"ZLV: "+UST+"53"+ZN	+ABNR	+<CR><LF>