

Simulator Exchanging Protokoll

Anforderungen an eine offene UDP Schnittstelle

Inhaltsverzeichnis

1.Hintergrund.....	2
2.Allgemeines.....	2
3.Systemschema.....	3
4.Anforderungen.....	4
4.1.Individuelle Beschreibung der Telegramme.....	4
4.2.Telegramme mit Prozesswerten.....	7
4.3.Telegramme mit Messagedaten.....	8
4.4.Datentypen.....	8
5.Verfügbare Prozesswerte.....	10
5.1.Digitale Prozesswerte.....	10
5.2.Analoge Prozesswerte.....	12
5.3.Messages.....	12
6.Beispiele.....	13
6.1.PZB-Leuchtmelder mit Sifa, Geschwindigkeit und Bremse.....	13
7.Glossar.....	14

Änderungsjournal

Datum	Bearbeiter	Kommentar	Version
21.07.18	Jens Eggert	Erster Entwurf	0.2
12.08.18	Jens Eggert	Modified order of requirements	0.3
28.08.18	Jens Eggert	Changed XML to JSON	0.4
13.08.18			0.5a

1. Hintergrund

Der Zeit gibt es verschiedene Ansätze zur Kommunikation von Simulatoren mit externen Programmen oder Hardware. Die Ansätze gehen über eine Kommunikation via TCP/IP, Einbindung eine DLL auf Simulatorseite, Einbindung eine DLL auf Zielanwendungsseite, Parsen von Prozessen und der Verwendung von OLE, das eigentlich für MS-Office-Anwendungen entwickelt wurde.

Die Ansätze haben dabei verschiedene Nachteile. Sie erzeugen viel Overhead, haben keine einheitlichen Telegramme, sind nicht ausreichend dokumentiert oder werden vom Simulatorhersteller nicht offiziell unterstützt. Dieses Dokument ist der Versuch eine Einheitliche Schnittstelle zu definieren, die von möglichst vielen Simulatorherstellern als ein einfach zu implementierender Standard angesehen wird und zu Anwendung kommt..

2. Allgemeines

Die Schnittstelle soll „Simulator Exchanging Protokoll“ -SEP- genannt werden. Sie soll einfach zu Interpretieren sein, keinen Verbindungsaufbau erfordern, möglichst wenig Overhead haben, ohne der Verwendung spezieller Bibliotheken zu implementieren sein und von PC-externen, wie internen Anwendungen verwendet werden können.

Das hier beschriebene Protokoll entspricht in Teilen denen, die in Eisenbahnfahrzeugen verwendeten verwendet werden. Unter anderem in den Bussystemen CAN, MVB oder Ethernet.

Konkrete Benennungen von Parametern in diesem Dokument beziehen sich auf den Bahnbereich. Benennungen für andere Anwendungen, wie z.B. Flug- oder Bussimulatoren können anderweitig gewählt werden.

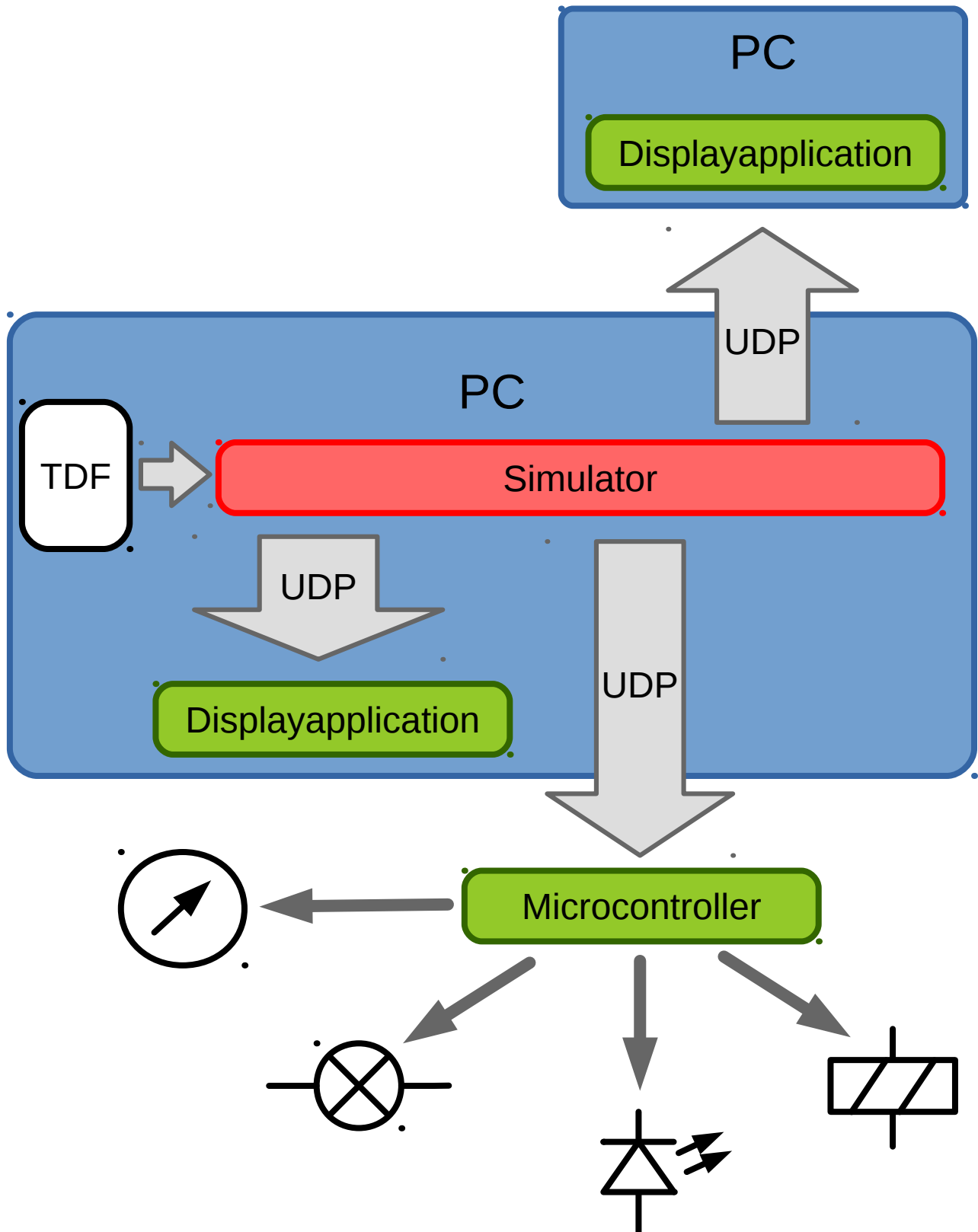
Im folgenden werden Anforderungen aufgelistet. Diese haben ID-Nummern und unterscheiden sich zwischen Pflichtanforderungen _p, optionalen Anforderungen _o und informativen Anforderungen _i.

Hersteller sind dazu angehalten, mindestens alle Pflichtanforderungen zu erfüllen, um Anwendungssoftware kompatibel zu halten.

Im Protokoll wird zwischen Prozesswerten und Messagedaten unterschieden. Prozesswerte bilden den Zustand von Leuchtmeldern, akustischen Signalgebern, Anzeigen von Analogwerten und Textmeldungen aus Textkonserven ab.

Messagedaten enthalten Texte, die für die Anzeige von Fahrplänen, Fahrdienstleiter-Befehlen oder Fahrtsuswertungen verwendet werden können.

3. Systemschema



4. Anforderungen

4.1. Individuelle Beschreibung der Telegramme

RequirementID_1010_p

Die Beschreibung eines Telegrammaufbaus erfolgt im JSON-Format.

Die JSON-Datei wird „Telegram describer file“ (TDF) bezeichnet und soll die Dateiendung *.tdf haben.

RequirementID_1020_p

Im TDF muss für jedes Telegramm ein UDP-Port definiert werden.

RequirementID_1030_p

Im TDF kann für jedes Telegramm eine IP-Adressen definiert werden.

Das fehlen der Angabe einer IP-Adresse impliziert die Verwendung des localhost.

RequirementID_1040_p

Das Format eines analogen Prozesswertes wird im TDF definiert.

Es muss die Länge des Wertes angegeben werden.

RequirementID_1050_p

Für analoge Prozesswerte muss ein Multiplikator im TDF definiert werden können.

RequirementID_1060_p

Die Anzahl aller verwendeten Bits eines Telegramms muss ein Vielfaches von acht sein.

RequirementID_1070_p

In der JSON-Struktur werden die Telegramme innerhalb von

```
"telegrams": [{ ... }, { ... }, { ... }]
```

als Array nacheinander notiert.

RequirementID_1080_p

In der JSON-Struktur wird eine IP-Adresse mit

```
"IP": ...
```

innerhalb eines Telegramms beschrieben.

RequirementID_1090_p

In der JSON-Struktur wird ein UDP-Port mit

```
"port": ...
```

innerhalb eines Telegramms beschrieben.

RequirementID_1100_p

In der JSON-Struktur das Format eines Telegramms innerhalb von

```
"format": [{ ... }, { ... }, { ... }]
```

innerhalb eines Telegramms beschrieben.

RequirementID_1110_p

In der JSON-Struktur wird ein digitales Element mit

```
"type": "digital"
```

innerhalb eines Formats beschrieben. Dabei ist der Name des Elements direkt zu verwenden.

RequirementID_1111_o

In der JSON-Struktur kann eine Zykluszeit in Millisekunden mit

```
"cycle": "250"
```

innerhalb eines Formats beschrieben werden.

RequirementID_1112_o

In der JSON-Struktur kann eine Hysterese mit

```
"hysteresis": "10"
```

innerhalb eines Formats angegeben werden.

RequirementID_1120_p

In der JSON-Struktur wird ein Indicator-Element mit

```
"type": "indicator"
```

innerhalb eines Formats beschrieben. Dabei ist der Name des Elements direkt zu verwenden.

RequirementID_1130_p

In der JSON-Struktur wird die Dimensionierung eines analogen Element mit der Wahl eines Datentyps

```
"type": "UInt16" oder z.B. "type": "digital"
```

innerhalb eines Formats beschrieben.

Siehe Tabelle 1 für weitere möglichen Datentypen.

RequirementID_1140_p

In der JSON-Struktur wird der Name eines analogen Elements mit

```
"name": " ... "
```

beschrieben und muss innerhalb des analogen Elements angegeben werden.

RequirementID_1150_p

In der JSON-Struktur wird ein Multiplikator eines analogen Elements mit

```
"factor": ...
```

beschrieben und muss innerhalb des analogen Elements angegeben werden.

RequirementID_1160_p

In der JSON-Struktur wird die Größe eines analogen Elements in Byte mit

```
"dimension": ...
```

beschrieben und muss innerhalb des analogen Elements angegeben werden.

RequirementID_1170_p

In der JSON-Struktur wird der Defaultwert eines analogen Elements mit

```
"default": ...
```

beschrieben und innerhalb des analogen Elements angegeben.

RequirementID_1180_p

In der JSON-Struktur wird ein Timer mit

```
"type": "timer"
```

innerhalb eines Formats beschrieben.

RequirementID_1190_p

In der JSON-Struktur wird die Größe eines Timers mit

```
"size": ...
```

innerhalb eines Formats eines Timers angegeben.

RequirementID_1200_p

In der JSON-Struktur wird die Zykluszeit eines Timers mit

```
"interval": ...
```

innerhalb eines Formats eines Timers angegeben.

RequirementID_1210_p

In der JSON-Struktur wird die Defaultwert eines Timers mit

```
"default": ...
```

innerhalb eines Formats eines Timers angegeben.

RequirementID_1220_o

Der Name eines Prozesswertes oder von Messagedaten sollte, sofern enthalten, der Namensliste dieses Dokuments entnommen werden. Siehe Tabellen.

RequirementID_1230_o

Die Namen von Prozesswerten und Messagedaten sollte, sofern nicht in diesem Dokument enthalten, auf Englisch gewählt werden.

RequirementID_1240_o

Für Prozesswerte können Defaultwerte im TDF definiert werden. Diese sollen nicht durch einen Multiplikator verändert werden.

RequirementID_1250_o

Für jedes Telegramm kann ein Zähler als Lebenszeichen definiert werden. Für diesen ist Größe in Byte und Zykluszeit in ms anzugeben.

RequirementID_1260_i

Im TDF können beliebig viele Telegramme definiert werden.

RequirementID_1270_i

Im TDF werden für jedes Telegramm entweder Prozesswerte oder Messagedaten definiert.

4.2. Telegramme mit Prozesswerten

RequirementID_2010_p

Prozesswerte müssen, der Reihenfolge aus dem TDF entsprechend, im Telegramm übertragen werden.

RequirementID_2020_p

Prozesswerte müssen unmittelbar aufeinander folgen.

RequirementID_2030_p

Für analoge Prozesswerte sind Fließkommazahlen nicht möglich.

Zur Darstellung von Nachkommastellen in einem Analogwert kann der Multiplikator aus dem TDF verwendet werden. Z.B. bewirkt die Multiplikation mit 100 die Darstellung von zwei Nachkommastellen.

RequirementID_2040_p

Prozesswerte des Typs Indicator bestehen aus 4 Bit.

Bit 0: Das Element ist aktiv.

RequirementID_2050_p

Digitale Prozesswerte bestehen aus einem Bit.

RequirementID_2060_p

Telegramme sollen ereignisbezogen übertragen werden, wenn im TDF weder eine Zähler, noch eine Zykluszeit definiert ist.

RequirementID_2070_p

Telegramme müssen zyklisch zu übertragen werden, wenn ein Zähler im TDF definiert ist.

RequirementID_2071_p

Telegramme müssen zyklisch zu übertragen werden, wenn eine Zykluszeit im TDF definiert ist.

RequirementID_2080_p

Werden Zähler nicht unterstützt, muss im Telegrammen für den Zähler der Defaultwert übertragen werden.

RequirementID_2090_p

Der Überlauf des Zählers führt zum Neubeginn bei Null.

RequirementID_2091_p

Ist im TDF für einen Element eine Hysterese definiert und ist keine Zykluszeit definiert, soll ein Telegramm dann übertragen werden, wenn sich der Betrag des Elements um mehr als den Betrag der Hysterese ändert.

RequirementID_2100_o

Für Prozesswerte des Typs Indicator kann, z.B. für Leuchtmelder, ein Blinken, schnelles Blinken und inverses Blinken definiert werden.

Bit 0: Das Element ist aktiv.

Bit 1: Das Element schaltet mit 1Hz.

Bit 2: Das Element schaltet mit 2Hz.

Bit 3: Das Element arbeitet invertiert.

Wird diese Anforderung unterstützt, muss Bit 0 auch bei einem Blinken immer gesetzt sein.

4.3. Telegramme mit Messagedaten

RequirementID_3010_p

Die enthaltene Information soll als Text im UTF-8-Format gestaltet werden.

RequirementID_3020_p

Tabellenspalten werden durch Kommas separiert.

RequirementID_3030_p

Das Ende von Messagedaten muss mit dem Nullterminator '\0' markiert werden.

RequirementID_3040_i

Die Länge von Messagedaten ist flexibel und richtet sich nach deren Inhalt.

4.4. Datentypen

Im folgenden werden Datentypen beschrieben, die dem Protokoll mindestens zu Verfügung stehen.

Name	Größe [Bit]	Anwendung
Digital	1	Einfache digitale Informationen.
Indicator	4	Leuchtmelder und Akkustische Geber.
BCD	4	Analoge Informationen mit 16 Werten.
UInt8	8	Analoge Informationen mit Werten von 0 bis 255
UInt16	16	Analoge Informationen mit Werten von 0 bis 65.535
UInt32	32	Analoge Informationen mit Werten von 0 bis 4.294.967.295
Int8	8	Analoge Informationen mit Werten von -128 bis 127
Int16	16	Analoge Informationen mit Werten von -32.768 bis 32.767
Int32	32	Analoge Informationen mit Werten von -2.147.483.648 bis 2.147.483.647
String		Zeichenkette mit jeweils festzulegender Länge

Tabelle 1

RequirementID_401_p

Der Datentyp Digital besteht aus einem Bit.

Wichtig: Er ist nicht wie in den meisten Programmiersprachen als ein Boolean zu betrachten, das aus 8 Bit besteht.

RequirementID_402_p

Der Datentyp Indicator besteht aus vier Bit. Er bildet eine komplexe Information über den Zustand eines Leuchtmelders oder Tongebers ab.

Bit 0: Das Element ist aktiv.

Bit 1: Das Element toggelt mit 1Hz.

Bit 2: Das Element toggelt mit 2Hz.

Bit 3: Das Element arbeitet invertiert.

RequirementID_403_p

Der Datentyp BCD besteht aus 4 Bit und hat kein Vorzeichen.

RequirementID_404_p

Der Datentyp UInt8 besteht aus 8 Bit und hat kein Vorzeichen.

RequirementID_405_p

Der Datentyp UInt16 besteht aus 16 Bit und hat kein Vorzeichen.

Das höchstwertige Byte steht am Ende. (*Big-Endian*)

RequirementID_406_p

Der Datentyp UInt32 besteht aus 32 Bit und hat kein Vorzeichen.

Das höchstwertige Byte steht am Ende. (*Big-Endian*)

RequirementID_407_p

Der Datentyp Int8 besteht aus 8 Bit und ist vorzeichenbehaftet.

RequirementID_408_p

Der Datentyp Int16 besteht aus 16 Bit und ist vorzeichenbehaftet.

Das höchstwertige Byte steht am Ende. (*Big-Endian*)

RequirementID_409_p

Der Datentyp Int32 besteht aus 32 Bit und ist vorzeichenbehaftet.

Das höchstwertige Byte steht am Ende. (*Big-Endian*)

RequirementID_4010_p

Der Datentyp String kann mit eine Länge von 2 bis 65.535 definiert werden.

Dabei entspräche die Länge von 2, einem einzelnen Zeichen.

65,507 ist die maximale Länge eines UDP-Packets. Diese Länge kann also nur erreicht werden, wenn kein anderer Prozesswert im Telegramm enthalten ist. Der Nullterminator ist in der Länge enthalten.

5. Verfügbare Prozesswerte

5.1. Digitale Prozesswerte

Die Liste der folgenden Prozesswerte hat noch keinen Anspruch auf Vollständigkeit.

LZB/PZB

Benennung	Beschreibung
IndLzbBef40	PZB-Leuchtmelder „Befehl 40“
IndLzb1000	PZB-Leuchtmelder „1000Hz“
IndLzb500	PZB-Leuchtmelder „500Hz“
IndLzbO	PZB-Leuchtmelder „85“
IndLzbM	PZB-Leuchtmelder „70“
IndLzbU	PZB-Leuchtmelder „55“
IndLzbH	LZB-Leuchtmelder „H“
IndLzbG	LZB-Leuchtmelder „G“
IndLzbE40	LZB-Leuchtmelder „E40“
IndLzbEl	LZB-Leuchtmelder „El“
IndLzbEnd	LZB-Leuchtmelder „Ende“
IndLzbV40	LZB-Leuchtmelder „V40“
IndLzbB	LZB-Leuchtmelder „B“
IndLzbS	LZB-Leuchtmelder „S“
IndLzbUe	LZB-Leuchtmelder „Ü“
IndLzbStoe	LZB-Leuchtmelder „Stör“
IndLzbHupe	Akustischer Signalgeber PZB-Hupe
IndLzbSchn	Akustischer Signalgeber LZB-Schnarre
IndLzbFb	LZB/PZB-Zwangsbremmung
IndLzbTl	LZB/PZB-Traktionssperre

Tabelle 2

Maschinentechnisch

Benennung	Beschreibung
IndMtMs	Leuchtmelder Hauptschalter
IndMtHD	Leuchtmelder hohe Abbremsung
IndMtEB	Leuchtmelder Elektrische Bremse
IndMtHvtl	Leuchtmelder Zugsammelschiene
IndMtEm	Leuchtmelder Notbremsung

Tabelle 3

Sifa

Benennung	Beschreibung
IndDsd	Leuchtmelder „Sifa“
IndDsdAco	Akustischer Signalgeber „Sifa“
IndDsdFb	Sifa Zwangsbremse
IndDsdTl	Sifa Traktionssperre

Tabelle 4

Türsystem

Benennung	Beschreibung
IndDorLOpen	Türen Links offen
IndDorROpen	Türen Rechts offen
IndDorSat	Leuchtmelder Türsystem SAT
IndDorTav	Leuchtmelder Türsystem TAV
IndDorTl	Türsystem Traktionssperre

Tabelle 5

Sonstiges

Benennung	Beschreibung
IndSimBreak	Der Simulator pausiert
IndSimFf	Der Simulator wird mit Zeitraffer ausgeführt
IndBlanc	Leeres Feld. Ein Element, das immer Null ist.

Tabelle 6

5.2. Analoge Prozesswerte

PZB/LZB

Benennung	Beschreibung	Einheit
LzbSpeedPermitted	LZB Soll-Geschwindigkeit	km/h
LzbSpeedTarget	LZB Ziel-Geschwindigkeit	km/h
LzbTargetDistance	LZB Zeil-Entfernung	m/10
LzbSpeedVue	LZB/PZB-Überwachungskurve	km/h

Tabelle 7

Maschinentechnisch

Benennung	Beschreibung	Einheit
speed	Ist-Geschwindigkeit in km/h	km/h
mainBrakePipe	Hauptluftleitung	bar
brakeCylinderPressure	Bremszylinderdruck	bar
mainAirReservoir	Hauptluftbehälter	bar
brakingForceAbs	Bremskraft absolut	kN
brakingForceRel	Bremskraft relativ	%
tractiveForceAbs	Zugkraft absolut	kN
tractiveForceRel	Zugkraft relativ	%

Tabelle 8

Sonstiges

Benennung	Beschreibung
telegramIndicator	In Verbindung mit Defaultwert zur Identifikation eines Telegramms

5.3. Messages

Benennung	Beschreibung
textTimeTable	Fahrplan (Generisch)
textEbuLa	EbuLa-Tabelle (Gesondert spezifiziert)
textMessageDmi	Textnachricht zur Anzeige im DMI nach ERA-Layout
textDispVerbal	Mündlicher Befehl vom Fahrdienstleiter
textDispWritten	Schriftlicher Befehl vom Fahrdienstleiter

Tabelle 9

6. Beispiele

6.1. PZB-Leuchtmelder mit Sifa, Geschwindigkeit und Bremse

TelegramDescriber.tdf

```
{
  "telegrams": [
    {
      "IP": "192.168.0.10"
      "port": "1001"
      "format": [
        {
          "type": "indicator"
          "name": "IndBlanc" // 4 leere Bits
        },
        {
          "type": "indicator"
          "name": "IndLzbBef40" // Lm Bef.40
        },
        {
          "type": "indicator"
          "name": "IndLzb1000" // Lm 1000Hz
        },
        {
          "type": "indicator"
          "name": "IndLzb500" // Lm 500Hz
        },
        {
          "type": "indicator"
          "name": "IndLzbO" // Lm 85
        },
        {
          "type": "indicator"
          "name": "IndLzbM" // Lm 70
        },
        {
          "type": "indicator"
          "name": "IndLzbU" // Lm 55
        },
        {
          "type": "indicator"
          "name": "IndDsd" // Lm Sifa
        }
      ]
    },
    {
      "IP": "192.168.0.10"
      "port": "1002"
      "format": [
        {
          "type": "UInt16"
          "name": "speed" // Geschwindigkeit
          "size": 2
          "factor": 10
        },
        {
          "type": "UInt16"
          "name": "mainBrakePipe" // Hauptluftleitung
          "size": 2
          "factor": 100
        },
        {
          "type": "UInt16"
          "name": "brakeCylinderPressure" // Bremszylinder
          "size": 2
          "factor": 100
        },
        {
          "type": "UInt16"
          "name": "mainAirReservoir" // Hauptluftbehälterleitung
          "size": 2
          "factor": 100
          "default": 10000
        }
      ]
    }
  ]
}
```

Telegramm 1 an 192.168.0.10 auf Port 1001 (4 Byte)

0000 0000	0001 0000	0011 0000	0000 0000
Bef. 40	1000Hz	500Hz	85 70 55 Sifa

1000Hz-Überwachung

0000 0000	0000 0001	0011 1011	0000 0000
Bef. 40	1000Hz	500Hz	85 70 55 Sifa

500Hz-Überwachung restriktiv

Telegramm 2 an 192.168.0.10 auf Port 1002 (8 Byte)

00000001	01001010	00000001	11001000	00000000	00010100	00000011	11101000
Geschwindigkeit		Hauptluftleitung		Bremszylinder		Hauptluftbehälter	

33km/h, 4,56bar HLL, 0,2bar Bremszylinder, 10bar Hauptluftbehälter

7. Glossar

Abkürzung	Bedeutung
ASCII	American Standard Code for Information Interchange Eine Codierung von Satzzeichen und Alphabet.
CAN	Controller Area Network. Ein Bussystem, das in Industrie Fahrzeugen eingesetzt wird.
DLL	Dynamic Link Library Eine Programmbibliothek mit einer Sammlung von Unterprogrammen.
EBuLa	Elektronischer Buchfahrplan und Langsamfahrstellen
Ethernet	Ein Bussystemen, dass zur Vernetzung von Computern verwendet wird.
IP	Internetprotokoll Ein Protokoll zur Übertragung durch Computernetze.
IP-Adresse	Die Adresse eines Teilnehmers eines IP-Computernetz.
JSON	JavaScript Object Notation Eine Sprache zur Darstellung und zum Austausch strukturierter Daten.
localhost	Die eigene Adresse eines Teilnehmers eines IP-Computernetz.
MVB	Multifunction Vehicle Bus Ein Bussystem, auf Schienenfahrzeugen eingesetzt wird.
OLE	Object Linking and Embedding. Ein Objektsystem zur Erstellung von Verbunddokumenten.
Port	Ordnet Daten innerhalb eines IP-Computernetz einer bestimmten Anwendung auf dem adressierten Computer zu.
RequirementID	Die Nummerierung der Anforderungen in diesem Dokument.
TCP	Transmission Control Protocol Ein auf IP basierendes, verbindungsbehaftetes Datenprotokoll.
UDP	Transmission Control Protocol Ein auf IP basierendes, verbindungsloses Datenprotokoll.

Tabelle 10