

Dokumentation



**Zugzielanzeiger Epoche VI von DB2
(V15NDB20015)**



<https://db-eep.de>

Inhalt

1	Modellübersicht	3
1.1	Zugzielanzeigen	3
1.2	Bahnhofsschilder	5
1.3	Schriften und sonstige Dateien	7
2	Vorwort	8
3	Aufbau	8
3.1	Zugzielanzeigen	8
3.2	Bahnhofsschilder	9
3.3	Gleisabschnitte/Gleisnummern.....	9
4	Beschriftung.....	9
4.1	Bahnhofsschilder	9
4.2	Zugzielanzeigen	9
4.2.1	Typen A und B.....	10
4.2.2	Typen C und D	11
4.2.3	Typen E und F	12
4.2.4	Typen G und H.....	12
5	Steuerung der Anzeigen	12
5.1	Grundsätzliche Einträge in der Lua-Datei.....	13
5.2	Anzeigetyp E und F	14
5.3	Anzeigetypen A und B	16
5.4	Anzeige Typ C und D.....	18
5.5	Anzeigen Typ G und H	24
6	Symbolschriftart Bahnsymbole Epoche VI	24
6.1	Schriftart „Bahnschrift“	24
6.2	Installation der Schriftarten	24
6.3	Symboltabelle.....	25
	Fragen, Anregungen oder Fehler gefunden?	26
	Rechtliche Hinweise	26
	Quellennachweise für die Modelle:	26
	Autor.....	26

1 Modellübersicht

Das Set Zugzielanzeigen Epoche VI besteht aus folgenden Modellen:

In normaler Schrift ist die Bezeichnung, wie sie in EEP zu finden ist.

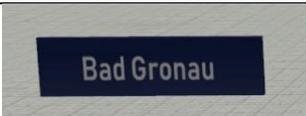
In kleiner Schrift ist der Name der Datei benannt.

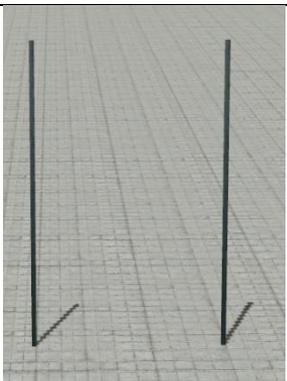
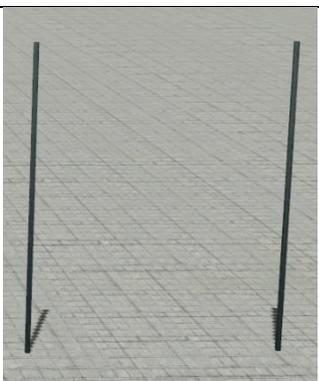
1.1 Zugzielanzeigen

	Immobilien:	In Immobilien / Verkehr / Bahnsteigzubehör
<p>Zugzielanzeigen mit 2 x 5 LCD-Displays</p> <p>Mit Beschriftungsfunktion und Laufschrift (abstellbar)</p>	<ul style="list-style-type: none"> • Wagenstandsanzeige • Maße (ohne Halter): 188 x 77 x 44 cm • Vorgegebene Einbauhöhe: 2,40 m • Halter links und rechts einzeln ausfahrbar • Länge der Halter: max. 74 cm 	
	<p>Zugzielanzeiger A, Epoche VI DB2</p> <p><i>ZZA6ADB2.3dm</i></p>	<p>Typ A:</p> <ul style="list-style-type: none"> • 2 Zeilen Zwischenziele • mit Vogelschutz
	<p>Zugzielanzeiger B, Epoche VI DB2</p> <p><i>ZZA6BDB2.3dm</i></p>	<p>Typ B:</p> <ul style="list-style-type: none"> • 2 Zeilen Zwischenziele • ohne Vogelschutz
	<p>Zugzielanzeiger C, Epoche VI DB2</p> <p><i>ZZA6CDB2.3dm</i></p>	<p>Typ C:</p> <ul style="list-style-type: none"> • Anzeige der Folgezüge • mit Vogelschutz
	<p>Zugzielanzeiger D, Epoche VI DB2</p> <p><i>ZZA6DDB2.3dm</i></p>	<p>Typ D:</p> <ul style="list-style-type: none"> • Anzeige der Folgezüge • ohne Vogelschutz

Zugzielanzeigen mit 1 x 5 LCD-Displays Mit Beschriftungsfunktion und Laufschrift (abstellbar)	Maße (ohne Halter): 188 x 48 x 44 cm Vorgegebene Einbauhöhe: 2,69 m Halter links und rechts einzeln ausfahrbar Länge der Halter: max. 46 cm	
	Zugzielanzeiger E, Epoche VI DB2 <i>ZZA6EDB2.3dm</i>	Typ E: <ul style="list-style-type: none"> • 1 Zeile Zwischenziele • Wagenstandsanzeige • mit Vogelschutz
	Zugzielanzeiger F, Epoche VI DB2 <i>ZZA6FDB2.3dm</i>	Typ F: <ul style="list-style-type: none"> • 1 Zeile Zwischenziele • Wagenstandsanzeige • ohne Vogelschutz
	Zugzielanzeiger G, Epoche VI DB2 <i>ZZA6GDB2.3dm</i>	Typ G: <ul style="list-style-type: none"> • 2 Zeilen Zugziele (typisch für S-Bahn) • Wagenanzeige • mit Vogelschutz
	Zugzielanzeiger H, Epoche VI DB2 <i>ZZA6HDB2.3dm</i>	Typ H: <ul style="list-style-type: none"> • 2 Zeilen Zugziele (typisch für S-Bahn) • Wagenanzeige • ohne Vogelschutz
Masten für Zugzielanzeiger		Höhe: 3,67 m (+ 50 cm unter Grund) Maststärke: 20 x 20 cm: 8 cm geeignet für Zugzielanzeiger mit 188 cm Breite
	Zugzielanz. Mast 1-seitig DB2 <i>ZZAT1DB2.3dm</i> Zugzielanz. Mast B 1-seitig DB2 <i>ZZAT3DB2.3dm</i>	Einseitiger Mast für Zugzielanzeiger Breite: 2,21 m, mit Vogelabwehr Ohne Vogelabwehr
	Zugzielanz. Mast 2-seitig DB2 <i>ZZAT2DB2.3dm</i> Zugzielanz. Mast B 2-seitig DB2 <i>ZZAT4DB2.3dm</i>	Zweiseitiger Mast für Zugzielanzeiger Breite: 4,16 m, mit Vogelabwehr Ohne Vogelabwehr

1.2 Bahnhofsschilder

Bahnhofsschilder (Stationstafeln) - frei beschriftbar	Vorgegebene Einbauhöhe: 2,40, Dicke: 8 cm (per Skalierung variabel) Epoche VI
	Bahnhofsschild 100x40, Ep VI DB2 <i>SA6_7DB2.3dm</i>
	Bahnhofsschild 150x40, Ep VI DB2 <i>SA6_6DB2.3dm</i>
	Bahnhofsschild 120x50, Ep VI DB2 <i>SA6_1DB2.3dm</i>
	Bahnhofsschild 150x50, Ep VI DB2 <i>SA6_2DB2.3dm</i>
	Bahnhofsschild 200x50, Ep VI DB2 <i>SA6_3DB2.3dm</i>
	Bahnhofsschild 250x50, Ep VI DB2 <i>SA6_4DB2.3dm</i>
	Bahnhofsschild 300x60, Ep VI DB2 <i>SA6_5DB2.3dm</i>
	Bahnhofsschild 40x40, Ep VI DB2 Für Gleisabschnitte oder Gleisnummern <i>SAATDB2.3dm</i>

Standträger für Bahnhofsschilder (und andere Schilder)	Länge 2,60 m plus 50 cm unter Boden, Stärke: 4 x 4 cm Verzinkter Stahl
	Bahnhofsschild Ständer 80cm Abstand DB2 <i>SA6T1DB2.3dm</i>
	Bahnhofsschild Ständer 120cm Abstand DB2 <i>SA6T2DB2.3dm</i>
	Bahnhofsschild Ständer 160cm Abstand DB2 <i>SA6T3DB2.3dm</i>
	Bahnhofsschild Ständer 210cm Abstand DB2 <i>SA6T4DB2.3dm</i>

2 Vorwort

Diese Anleitung geht möglichst ausführlich auf den Aufbau und die Funktionen der Modelle ein, um auch EEP-Anfängern die Möglichkeit zu geben, möglichst schnell die vielfältigen Möglichkeiten dieser neuen Zugzielanzeigen nutzen zu können. Erfahrene Nutzer werden sicher einige Absätze finden, die ihnen nichts Neues vermitteln. Sie mögen aber über die Gründlichkeit der Beschreibung hinwegsehen, damit Neulinge und ungeübte EEP´ler mehr Spaß an EEP bekommen.

Wenn dennoch etwas unklar ist, bin ich über die am Ende der Dokumentation stehenden Kanäle erreichbar und stehe gerne Rede und Antwort.

3 Aufbau

3.1 Zugzielanzeigen

Die Zugzielanzeigen können an den dafür bestimmten mitgelieferten Masten, als auch an Deckenkonstruktionen z.B. von Bahnsteigüberdachungen aufgehängt werden.

Die Höhe der Unterkante beträgt beim Einbau bei den zweizeiligen Anzeigen 2,40 m, bei den einzeiligen 2,69 m über dem Boden. Bei Nutzung der mitgelieferten Masten zur Aufhängung entspricht dies exakt der richtigen Höhe für die Aufhängung, wenn der Mast an derselben Stelle eingesetzt wird. Der seitliche Versatz der Anzeigen beträgt 2,08 m, auf der Längsachse ist derselbe Wert wie beim Masten zu wählen.

Diese Werte gelten natürlich nur bei rechtwinkligem Einbau. Um den Einbau auch in anderen Winkeln zu vereinfachen, sind zwei Blöcke bestehend aus Mast und Anzeige(n) enthalten. Nach Aufstellen des Blocks kann die Anzeige im 3D-Editor ausgetauscht werden. Dazu wählt man in der Modellliste die gewünschte Anzeige, klickt mit der rechten Maustaste auf die Anzeige, die ausgewechselt werden soll und wählt dann den Punkt „Tauschen gegen“. Die Anzeigen dieses Sets sind alle miteinander kompatibel, so dass auch die kleineren Anzeigen automatisch an der richtigen Position eingesetzt werden.

Im Inneren der Anzeigen befinden sich zwei Halter, die je nach Bedarf auf die gewünschte Länge ausgefahren werden können. Dazu wählt man in den Objekteigenschaften den linken und danach den rechten Halter aus und schiebt ihn mit dem Schieberegler in die gewünschte Höhe. Bei Aufhängung an den Masten platzieren sich diese Stangen genau in der Montageplatte, so dass der Eindruck entsteht, dass die Anzeigen am Mast angeschraubt sind.

3.2 Bahnhofsschilder

Die Bahnhofsschilder werden ebenfalls standardgemäß mit einer Unterkante von 2,40 m eingesetzt. Passende Ständer sind im Set enthalten und werden mit den gleichen Koordinaten wie die Schilder gesetzt.

Natürlich können die Schilder auch in der Größe angepasst werden. So ist im Demovideo ein Beispiel, wie ein Schild über einer Bahnhofsbank thront und das im Original der Bank vorhandene Schild komplett verdeckt. Dies wurde durch eine Verdreifachung der Stärke (Y-Achse) und eine leichte Skalierung in der Z-Achse (1,02) erreicht. Auch an eine Bahnhofswand kann das Schild angebracht werden.

3.3 Gleisabschnitte/Gleisnummern

Die Schilder 40 x 40 cm haben einen stabilen seitlichen Halter, der mit 2 Schrauben an Laternenmasten, Bahnsteigs-Dachstützen oder Wänden montiert werden kann. Über die Beschriftungsfunktion sind der gewünschte Buchstabe oder die gewünschte Nummer einzugeben.

4 Beschriftung

4.1 Bahnhofsschilder

Die Beschriftung der Schilder ist denkbar einfach: In den Objekteigenschaften ist seit EEP 15 die Schaltfläche „Aufschrift“. Wird diese angeklickt, öffnet sich ein weiteres Fenster, in dem im unteren Bereich der gewünschte Text eingegeben wird.

Darüber lassen sich bei Bedarf weitere Anpassungen vornehmen: Die Schrift lässt sich ändern, auch die Schriftgröße und die Darstellung (fett, leicht, kursiv). Über das Häkchen bei „zentriert“ wird die Ausrichtung des Textes innerhalb des vom Konstrukteur vorgegebenen Feldes geändert.

Zuletzt können Schrift- und Hintergrundfarbe geändert werden. Bitte beachten Sie, dass die Hintergrundfarbe sich nur auf die tatsächlich geschriebenen Buchstaben auswirkt. Wenn Sie einen blaün Text auf weißem Grund schreiben wollen, ist die Fläche nur etwas höher weiß, als der Buchstabe ist. Sofern rechts und/oder links der weiße Balken weitergehen soll (wie bei den Laufschrifttexten der Zugzielanzeigen), so können sie die gewünschte Breite durch eine entsprechende Menge an Leerzeichen auffüllen. Hierbei ist etwas Geduld gefragt, da man erst herausfinden muss, wie viele Leerzeichen optimal sind.

4.2 Zugzielanzeigen

Die Zugzielanzeigen verfügen über eine größere Menge an Textfeldern. Einige davon werden sinnvollerweise direkt beim Einsetzen ausgefüllt, andere per Lua-Programmierung (s.u.).

4.2.1 Typen A und B

Die Anzeigen enthalten folgende Textfelder:

Textnr.	Feldinhalt Vorderseite	Textnr.	Feldinhalt Rückseite
1	Zugziel	2	Zugziel
3	Zwischenziele 1	4	Zwischenziele 1
5	Zwischenziele 2	6	Zwischenziele 2
7	Zeit	8	Zeit
9	Zugnummer	10	Zugnummer
11	Wagenlauf	12	Wagenlauf
13	Bahnsteigabschnitte	14	Bahnsteigabschnitte
15	Gleisnummer	16	Gleisnummer
17	Infotext (Laufschrift)	Wird beidseitig angezeigt	

Aus technischen und optischen Gründen sind die Eintragungen beidseitig vorzunehmen. Die Felder 13 – 16 werden sinnvollerweise sofort statisch beschriftet. Hinweise zu den Bahnsteigabschnitten werden weiter unten bei Bahnsymbole Epoche VI beschrieben.

4.2.2 Typen C und D

Die Anzeigen enthalten mehr Textfelder, da auch die nächsten beiden Züge angekündigt werden:

Textnr.	Feldinhalt Vorderseite	Textnr.	Feldinhalt Rückseite
Eintragungen für den Hauptzug			
1	Zugziel	2	Zugziel
3	Zwischenziele	4	Zwischenziele
5	Zeit	6	Zeit
7	Zugnummer	9	Zugnummer
9	Wagenlauf	10	Wagenlauf
Eintragungen für den 1. Folgezug			
11	Zeit Folgezug 1	12	Zeit Folgezug 1
13	Zugnummer Folgezug 1	14	Zugnummer Folgezug 1
15	Ziel Folgezug 1	16	Ziel Folgezug 1
17	Information Folgezug 1 (hier ist ein kurzer Text blau auf weiß möglich)	18	Information Folgezug 1
Eintragungen für den 2. Folgezug			
19	Zeit Folgezug 2	20	Zeit Folgezug 2
21	Zugnummer Folgezug 2	22	Zugnummer Folgezug 2
23	Ziel Folgezug 2	24	Ziel Folgezug 2
25	Information Folgezug 2	26	Information Folgezug 2
Allgemeine statische Informationen			
27	Bahnsteigabschnitte	28	Bahnsteigabschnitte
29	Gleisnummer	30	Gleisnummer
31	Überschrift „Folgezüge“	32	Überschrift „Folgezüge“
Aus technischen Gründen steht die Laufschrift mit der allgemeinen Fahrgastinformation immer am Ende der Eintragungen.			
33	Infotext (Laufschrift)	Wird beidseitig angezeigt	

4.2.3 Typen E und F

Diese Anzeige ist funktionell wie die Typen A und B ausgestattet, hat aber nur eine LCD-Reihe und folglich weniger Felder.

Textnr.	Feldinhalt Vorderseite	Textnr.	Feldinhalt Rückseite
1	Zugziel	2	Zugziel
3	Zwischenziele 1	4	Zwischenziele 1
5	Zeit	6	Zeit
7	Zugnummer	8	Zugnummer
9	Wagenlauf	10	Wagenlauf
11	Bahnsteigabschnitte	12	Bahnsteigabschnitte
13	Gleisnummer	14	Gleisnummer
15	Infotext (Laufschrift)	Wird beidseitig angezeigt	

4.2.4 Typen G und H

Die Anzeige ist grundsätzlich anders aufgebaut als die vorherigen, weswegen hier die Anzeigen nicht vorder- und rückseitig separat beschriftet werden müssen.

Textnr.	Feldinhalt erste Zeile	Textnr.	Feldinhalt zweite Zeile
1	Linie 1. Zug	5	Linie 2. Zug
2	Wagenlauf 1. Zug	6	Wagenlauf 2. Zug
3	Ziel 1. Zug	7	Ziel 2. Zug
4	Abfahrt 1. Zug	8	Abfahrt 2. Zug
Statische Angaben			
9	Text "min" 1. Zug	10	Text "min" 2. Zug
Und wieder zum Schluss:			
11	Infotext (Laufschrift)		

5 Steuerung der Anzeigen

Jetzt kommt es zum eigentlich interessanten Teil. Denn das Besondere im Vergleich zu den bisherigen Zugzielanzeigen ist die beliebige Beschriftbarkeit – dank der neuen Beschriftungsfunktion in EEP 15.

Allerdings funktioniert dies sinnvollerweise nur mit dem Einsatz der in EEP enthaltenen Lua-Skriptsprache. Aber keine Angst, man muss kein Programmierer sein, nicht mal ein Mathe-As, um die Anzeigen zu steuern. Im einfachsten Fall reicht es einfach, den passenden unten stehenden Text (Programmcode), der so auch in der Demoanlage enthalten ist, zu kopieren und die eigenen Ziele

und Zeiten einzutragen. Dazu ist der Code extra so geschrieben, dass man nicht rätseln muss, wo etwas eingetragen wird, sondern die Einträge ganz offensichtlich sind.

Wer dagegen Programmierkenntnisse bzw. Erfahrung mit Lua hat, wird den Code sicher optimieren und um weitere Funktionen erweitern können. Aber auch weniger Erfahrene werden sich mit ein wenig Einarbeitung die ein oder andere Funktion zusätzlich implementieren können.

Die Anleitung folgt nicht der logischen Reihenfolge der Typen A, B, C ... H, sondern beginnt mit der einfachsten Anzeige.

5.1 Grundsätzliche Einträge in der Lua-Datei

Vorweg möchte ich jedem empfehlen, sich Folgendes kostenlos zu installieren:

1. Notepad++ - Editor

Der meines Erachtens sicherste und einfachste Download in deutscher Sprache ist bei Heise.de:

<https://www.heise.de/download/product/notepad-26659>

2. Die Lua-Befehlserweiterung speziell für EEP, Download mit Anleitung im Forum:

<https://www.eepforum.de/filebase/file/23-lua-xml-f%C3%BCr-notepad/>

Durch die farblichen Hervorhebungen und Einrückungen sowie dem größeren Texteditor wird der Code wesentlich übersichtlicher. Die Zeilennummern sind nützlich, wenn z.B. aufgrund eines Schreibfehlers ein Fehlermeldung ausgeworfen wird, in der EEP die Nummer der betreffenden Zeile angibt. Ohne Notepad++ kann man unter Umständen sehr lange suchen.

Die Programmzeilen beziehen sich auf die Demoanlage und können dort nachvollzogen werden.

Grün sind Kommentare. Sie beginnen mit zwei Bindestrichen:

```
-- Die folgende Codezeile wurde von Benjamin Hoge (BH2 - emaps-eep.de) erstellt und bewirkt,  
-- dass in den Kontaktpunkten beim Aufruf einer Lua-Funktion Parameter mit übergeben werden können.  
-- In diesem Beispiel wird das für die Steuerung der Anzeigen auf Gleis 3 genutzt.
```

Grau unterlegt ist der Programmcode – in Notepad++ ist der wiederum in verschiedene Farben unterteilt, was für das grundsätzliche Verstehen zwar hilfreich, aber nicht ganz so wichtig ist. Schließlich soll das kein Programmierkurs werden ;) Lediglich die Funktionsaufrufe sind in **dunkelblau**, die Übergabeparameter nicht fett gedruckt.

```
setmetatable(_ENV, {__index=function(s,k) local n=k:gsub("%.","");  
local p=load(n);if p then local f=function(z) local  
s=Zugname;Zugname=z;p();Zugname=s end;_ENV[k]=f;_ENV[n]=f;return f  
end;return nil end})
```

5.2 Anzeigetyp E und F

-- Der folgende Block steuert die Anzeigen auf Gleis 4 -----

Folgende Texte sollen automatisiert geändert werden:

- 1 Zugziel Vorderseite
- 2 Zugziel Rückseite
- 3 Zwischenziele Vorderseite
- 4 Zwischenziele Rückseite
- 5 Zeit Vorderseite
- 6 Zeit Rückseite
- 7 Zugnr Vorderseite
- 8 Zugnr Rückseite

Die weiteren Beschriftungen sind entweder über die Objekteigenschaften bereits gesetzt oder werden in diesem Beispiel nicht benötigt.

Zürst kommt eine Aufruffunktion. Diese legt für den jeweiligen Zug fest, welche Werte eingetragen werden sollen.

```
function IRE629_MD04() -- IRE 629 fährt in Magdeburg auf Gleis 4.
```

--Diese etwas umständliche Bezeichnung definiert eine eindeutige Funktion, die so kein zweites Mal auftritt (auftreten darf).

```
    ziel_ZZA_MD04 = "Nürnberg Hbf"           -- Das Ziel der Reise
    zwziel_ZZA_MD04 = "Halle (Saale) Erfurt" -- ausgewählte Zwischenziele
    zeit_ZZA_MD04 = "09:27"                 -- Abfahrtszeit
    zugnr_ZZA_MD04 = "IRE 629"              -- Bezeichnung des Zuges
```

-- Nun werden diese Werte an die eigentliche Funktion, die die Einträge vornimmt, übertragen:

```
    ZZA_MD04(ziel_ZZA_MD04, zwziel_ZZA_MD04, zeit_ZZA_MD04,
    zugnr_ZZA_MD04)
end -- jede Funktion muss mit einem "end" abgeschlossen werden.
```

-- (Die ZZA für weiteren Züge werden nach dem folgenden Block gesteuert)

-- Der "Umweg" wird schnell klar, wenn man sich die Länge der folgenden Eintragungsfunktion ansieht. Wenn die Eintragung direkt vorgenommen würde, müsste jedes Mal dieser lange Block geschrieben werden,

-- was den Programmcode unnötig aufbläst.

```
function ZZA_MD04(ziel_ZZA_MD04, zwziel_ZZA_MD04, zeit_ZZA_MD04, zugnr_ZZA_MD04)
```

-- Durch die in Klammern stehenden Variablen bekommt diese Funktion die einzutragenden Werte.

-- Nun wird die erste ZZA mit dem richtigen Text befüllt: Die Nummer nach der Raute steht in den Objekteigenschaften unter "Lua Name". Es wird nur die Zahl benötigt, der Rest ist nicht wichtig.

```
EEPStructureSetTextureText( "#345" , 1 , ziel_ZZA_MD04 )
EEPStructureSetTextureText( "#345" , 2 , ziel_ZZA_MD04 )
EEPStructureSetTextureText( "#345" , 3 , zwziel_ZZA_MD04 )
EEPStructureSetTextureText( "#345" , 4 , zwziel_ZZA_MD04 )
EEPStructureSetTextureText( "#345" , 5 , zeit_ZZA_MD04 )
EEPStructureSetTextureText( "#345" , 6 , zeit_ZZA_MD04 )
EEPStructureSetTextureText( "#345" , 7 , zugnr_ZZA_MD04 )
EEPStructureSetTextureText( "#345" , 8 , zugnr_ZZA_MD04 )
```

-- Die zweite ZZA hat die Nummer 342

```
EEPStructureSetTextureText( "#342" , 1 , ziel_ZZA_MD04 )
EEPStructureSetTextureText( "#342" , 2 , ziel_ZZA_MD04 )
EEPStructureSetTextureText( "#342" , 3 , zwziel_ZZA_MD04 )
EEPStructureSetTextureText( "#342" , 4 , zwziel_ZZA_MD04 )
EEPStructureSetTextureText( "#342" , 5 , zeit_ZZA_MD04 )
EEPStructureSetTextureText( "#342" , 6 , zeit_ZZA_MD04 )
EEPStructureSetTextureText( "#342" , 7 , zugnr_ZZA_MD04 )
EEPStructureSetTextureText( "#342" , 8 , zugnr_ZZA_MD04 )
```

-- So werden auch die übrigen Zielanzeigen befüllt. In diesem Fall sind das noch sechs Anzeigen. In der Lua-Datei der Anlage sind diese komplett nachzulesen.

-- Der geübte Lua-Kundige kann das mit einer Schleife noch etwas abkürzen.

end

--So wie oben der erste IRE werden auch die folgenden Züge angekündigt. Versuchen Sie mal zu verstehen, was da wichtig für Sie ist:

```
function IRE627_MD04()
```

```
ziel_ZZA_MD04 = "Dresden Hbf"
zwziel_ZZA_MD04 = "Dessau Leipzig"
zeit_ZZA_MD04 = "09:32"
```

```
zugnr_ZZA_MD04 = "IRE 627"  
ZZA_MD04(ziel_ZZA_MD04, zwziel_ZZA_MD04, zeit_ZZA_MD04,  
zugnr_ZZA_MD04)
```

end

-- Ich bin davon überzeugt, dass Sie nun wissen, dass sie nur die paar Daten bei ziel, zwziel, zeit und zugnr eintragen müssen. Der Rest kann einfach kopiert und mit einer der entsprechenden Gleis- und Zugnummer versehen werden.

--Die übrigen Züge finden Sie wieder in der Demoanlage

-- Ende Programmblock Anzeigen Gleis 4 -----

5.3 Anzeigetypen A und B

-- Der folgende Block steuert die Anzeigen auf Gleis 2 -----

-- Der Block ist ähnlich dem von Gleis 4. Er hat allerdings mehr Einträge. Die Werte der ZZA in Magdeburg Gleis 2 werden im folgenden Block eingetragen. Allerdings werden wieder nur Variablen als "Platzhalter" verwendet, die dann in den nachfolgenden Funktionen von den einzelnen Zügen eingetragen werden.

Im Falle des Gleis 2 zeige ich, wie der Eintrag vom Zug selbst gesetzt wird, wenn er aus dem Depot kommt und einen dort liegenden Kontaktpunkt überfährt.

Wenn ein Zug den Bahnhof verlässt, wird durch einen Kontaktpunkt nach dem Ausfahrtsignal die Funktion "keineAnzeige_MD02()" ausgelöst, die die Anzeigetafel leert.

Diese Art der Programmierung funktioniert so nur, wenn die Ausfahrt aus dem Depot, und damit das Überfahren des Kontaktpunktes zur Anzeigenerstellung, NACH dem Verlassen des Bahnhofs durch den vorherigen Zug erfolgt.

```
function ZZA_MD02(ziel_ZZA_MD02, zwziel1_ZZA_MD02, zwziel2_ZZA_MD02,  
zeit_ZZA_MD02, zugnr_ZZA_MD02, wagenlauf_ZZA_MD02)
```

```
EEPStructureSetTextureText( "#296" , 1 , ziel_ZZA_MD02 )  
EEPStructureSetTextureText( "#296" , 2 , ziel_ZZA_MD02 )  
EEPStructureSetTextureText( "#296" , 3 , zwziel1_ZZA_MD02 )  
EEPStructureSetTextureText( "#296" , 4 , zwziel1_ZZA_MD02 )  
EEPStructureSetTextureText( "#296" , 5 , zwziel2_ZZA_MD02 )  
EEPStructureSetTextureText( "#296" , 6 , zwziel2_ZZA_MD02 )  
EEPStructureSetTextureText( "#296" , 7 , zeit_ZZA_MD02 )
```

```
EEPStructureSetTextureText( "#296" , 8 , zeit_ZZA_MD02 )
EEPStructureSetTextureText( "#296" , 9 , zugnr_ZZA_MD02 )
EEPStructureSetTextureText( "#296" , 10 , zugnr_ZZA_MD02 )
EEPStructureSetTextureText( "#296" , 11 , wagenlauf_ZZA_MD02 )
EEPStructureSetTextureText( "#296" , 12 , wagenlauf_ZZA_MD02 )
```

```
EEPStructureSetTextureText( "#292" , 1 , ziel_ZZA_MD02 )
EEPStructureSetTextureText( "#292" , 2 , ziel_ZZA_MD02 )
EEPStructureSetTextureText( "#292" , 3 , zwziel1_ZZA_MD02 )
EEPStructureSetTextureText( "#292" , 4 , zwziel1_ZZA_MD02 )
EEPStructureSetTextureText( "#292" , 5 , zwziel2_ZZA_MD02 )
EEPStructureSetTextureText( "#292" , 6 , zwziel2_ZZA_MD02 )
EEPStructureSetTextureText( "#292" , 7 , zeit_ZZA_MD02 )
EEPStructureSetTextureText( "#292" , 8 , zeit_ZZA_MD02 )
EEPStructureSetTextureText( "#292" , 9 , zugnr_ZZA_MD02 )
EEPStructureSetTextureText( "#292" , 10 , zugnr_ZZA_MD02 )
EEPStructureSetTextureText( "#292" , 11 , wagenlauf_ZZA_MD02 )
EEPStructureSetTextureText( "#292" , 12 , wagenlauf_ZZA_MD02 )
```

-- entsprechend für die weiteren Anzeigetafeln auf Gleis 2.

end

-- Der ICE 152 kommt per Depotausfahrt gesteuert um 9:19 aus dem Depot und fährt über einen Fahrzeugkontakt, der diese Funktion auslöst:

function ICE152_MD02()

```
ziel_ZZA_MD02 = "Ostseebad Binz"
zwziel1_ZZA_MD02 = "Brandenburg Potsdam Berlin"
zwziel2_ZZA_MD02 = "Neubrandenburg Greifswald Stralsund"
zeit_ZZA_MD02 = "09:25"
zugnr_ZZA_MD02 = "ICE 152"
wagenlauf_ZZA_MD02 = "<222222s11>"
```

-- Die Liste der Symbole für den Wagenlauf finden Sie [weiter unten](#).

-- Die folgende Zeile übergibt die oben eingetragenen Werte an die oben stehende Funktion ZZA_MD002, die die Zugzielanzeiger beschriftet.

```
ZZA_MD02(ziel_ZZA_MD02, zwziel1_ZZA_MD02, zwziel2_ZZA_MD02,
zeit_ZZA_MD02, zugnr_ZZA_MD02, wagenlauf_ZZA_MD02)
```

end

-- So wird jetzt jeder Zug für Gleis 2 hier eingetragen

```
function TEE36_MD02()
```

```
    ziel_ZZA_MD02 = "München Hbf"  
    zwziel1_ZZA_MD02 = "Duisburg Köln Bonn"  
    zwziel2_ZZA_MD02 = "Frankfurt(M) Würzburg Nürnberg"  
    zeit_ZZA_MD02 = "09:34"  
    zugnr_ZZA_MD02 = "TEE 36"  
    wagenlauf_ZZA_MD02 = "<111111s11"  
    ZZA_MD02(ziel_ZZA_MD02, zwziel1_ZZA_MD02, zwziel2_ZZA_MD02,  
    zeit_ZZA_MD02, zugnr_ZZA_MD02, wagenlauf_ZZA_MD02)
```

```
end
```

```
-- usw. Bis am Ende die Funktion der leeren Anzeige kommt:
```

```
function keineAnzeige_MD02()
```

```
    ziel_ZZA_MD02 = ""  
    zwziel1_ZZA_MD02 = ""  
    zwziel2_ZZA_MD02 = ""  
    zeit_ZZA_MD02 = ""  
    zugnr_ZZA_MD02 = ""  
    wagenlauf_ZZA_MD02 = ""  
    ZZA_MD02(ziel_ZZA_MD02, zwziel1_ZZA_MD02, zwziel2_ZZA_MD02,  
    zeit_ZZA_MD02, zugnr_ZZA_MD02, wagenlauf_ZZA_MD02)
```

```
end
```

```
-- Ende Programmblock Anzeigen Gleis 2 -----  
-----
```

5.4 Anzeige Typ C und D

Nun wird es etwas komplizierter. Dass heißt, es sieht komplizierter aus, aber wer mit den bisherigen Anzeigen zurechtgekommen ist, wird auch diese Form schnell verstehen.

Bei diesen Anzeigen werden dem nächsten fahrenden Zug zwei Folgezüge hinzugefügt, als verbesserte Information für die Fahrgäste. Es wird daher eine „if .. else“ (entweder .. oder) – Schleife hinzugefügt, eine der einfachsten und wirkungsvollsten Programmroutinen überhaupt.

```
-----  
-- Der folgende Block steuert die Anzeigen auf Gleis 3 -----
```

```
-- Grundsätzlich unterscheidet sich der Aufruf dieser Programmroutine von den vorhergegangenen  
auch dadurch, dass er, außer beim ersten Zug, nach dem Verlassen des Bahnhofes durch diesen Zug  
ausgelöst wird.
```

```
Das heißt, dass damit sofort der nächste Zug - samt der beiden Folgezüge, angekündigt wird.
```

Zunächst wieder die, jetzt noch umfangreichere, Funktion zum Eintrag der Werte:

```
function ZZA_MD03(ziel0_ZZA_MD03, zwziel0_ZZA_MD03, zeit0_ZZA_MD03,  
zugnr0_ZZA_MD03, wagenlauf0_ZZA_MD03, zeitZug1_ZZA_MD_03,  
zugnrZug1_ZZA_MD_03, zielZug1_ZZA_MD_03, infoZug1_ZZA_MD_03,  
zeitZug2_ZZA_MD_03, zugnrZug2_ZZA_MD_03, zielZug2_ZZA_MD_03,  
infoZug2_ZZA_MD_03)
```

```
EEPStructureSetTextureText( "#347" , 1 , ziel0_ZZA_MD03 )  
EEPStructureSetTextureText( "#347" , 2 , ziel0_ZZA_MD03 )  
EEPStructureSetTextureText( "#347" , 3 , zwziel0_ZZA_MD03 )  
EEPStructureSetTextureText( "#347" , 4 , zwziel0_ZZA_MD03 )  
EEPStructureSetTextureText( "#347" , 5 , zeit0_ZZA_MD03 )  
EEPStructureSetTextureText( "#347" , 6 , zeit0_ZZA_MD03 )  
EEPStructureSetTextureText( "#347" , 7 , zugnr0_ZZA_MD03 )  
EEPStructureSetTextureText( "#347" , 8 , zugnr0_ZZA_MD03 )  
EEPStructureSetTextureText( "#347" , 9 , wagenlauf0_ZZA_MD03 )  
EEPStructureSetTextureText( "#347" , 10 , wagenlauf0_ZZA_MD03 )  
EEPStructureSetTextureText( "#347" , 11 , zeitZug1_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 12 , zeitZug1_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 13 , zugnrZug1_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 14 , zugnrZug1_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 15 , zielZug1_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 16 , zielZug1_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 17 , infoZug1_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 18 , infoZug1_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 19 , zeitZug2_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 20 , zeitZug2_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 21 , zugnrZug2_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 22 , zugnrZug2_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 23 , zielZug2_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 24 , zielZug2_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 25 , infoZug2_ZZA_MD_03 )  
EEPStructureSetTextureText( "#347" , 26 , infoZug2_ZZA_MD_03 )
```

-- Wie gehabt wird dieser Block kopiert und mit der jeweiligen Nummer der Anzeige versehen – oder per Schleife gesteuert.

end

-- Nun folgen die Funktionen für die Bestimmung der Einträge. Wenn diese Funktion über den Kontaktpunkt aufgerufen wird, bekommt sie den Wert 0 mit übergeben. Der Lua-Funktionsaufruf im Kontaktpunkt (nach dem Bahnhof) lautet also "ICE218_MD03(0)". In der Funktion nimmt z den übergebenen Wert an.

```
function ICE218_MD03(z)
```

-- Zunächst werden die Werte für diesen Zug übergeben:

```
ziel_ZZA_MD03 = "München Hbf"  
zziel_ZZA_MD03 = "Leipzig Nürnberg"  
zeit_ZZA_MD03 = "09:26"  
zugnr_ZZA_MD03 = "ICE 218"  
wagenlauf_ZZA_MD03 = "L22c1R"
```

```
if z == 0 then
```

-- die Bedingung `z == 0` ist wahr, also werden die folgenden Zeilen ausgeführt. Die Werte von oben werden in eine Variable mit dem Zusatz 0 geschrieben:

ÜBRIGENS: Das doppelte Gleichheitszeichen ist kein Fehler, sondern zeigt EEP, dass es den Wert, den `z` gerade hat, mit der 0 vergleichen soll. Möglicherweise würde das Programm sonst annehmen, dass wir der Variable `z` diesen Wert zuweisen wollen, was ja nicht der Fall ist.

```
ziel0_ZZA_MD03 = ziel_ZZA_MD03  
zziel0_ZZA_MD03 = zziel_ZZA_MD03  
zeit0_ZZA_MD03 = zeit_ZZA_MD03  
zugnr0_ZZA_MD03 = zugnr_ZZA_MD03  
wagenlauf0_ZZA_MD03 = wagenlauf_ZZA_MD03
```

-- Nun wird die Programmroutine des ersten Folgezuges aufgerufen. Dabei wird der Wert 1 übergeben.

```
TEE103_MD03(1)
```

-- Zunächst wird die Routine [TEE103_MD03](#) ausgeführt, schauen Sie also, was [dort](#) passiert.

-- Das Programm hat nun die Werte des Haupt- und ersten Folgezuges im Speicher. Fehlt also noch der zweite Folgezug. Aufruf, Eintrag und Rückgabe der Werte funktionieren genauso wie beim ersten Folgezug, nur dass dieses Mal der Wert 2 übergeben wird und damit das zweite `elseif` in der folgenden Funktion ausgewertet wird.

```
IRE815_MD03(2)
```

-- Also erst zur „[Subroutine](#)“ ...

-- Wir haben jetzt alle Daten, die wir für die Anzeige benötigen und übergeben diese an die Funktion zum Eintragen der Werte

```
ZZA_MD03(ziel0_ZZA_MD03, zwziel0_ZZA_MD03, zeit0_ZZA_MD03,  
zugnr0_ZZA_MD03, wagenlauf0_ZZA_MD03, zeitZug1_ZZA_MD_03,  
zugnrZug1_ZZA_MD_03, zielZug1_ZZA_MD_03, infoZug1_ZZA_MD_03,  
zeitZug2_ZZA_MD_03, zugnrZug2_ZZA_MD_03, zielZug2_ZZA_MD_03,  
infoZug2_ZZA_MD_03)
```

-- Die folgenden Zeilen werden von EEP jetzt nicht mehr gelesen

```
elseif z == 1 then  
zeitZug1_ZZA_MD_03 = zeit_ZZA_MD03  
zugnrZug1_ZZA_MD_03 = zugnr_ZZA_MD03  
zielZug1_ZZA_MD_03 = ziel_ZZA_MD03  
infoZug1_ZZA_MD_03 = ""  
return (zeitZug1_ZZA_MD_03), (zugnrZug1_ZZA_MD_03),  
(zielZug1_ZZA_MD_03), (infoZug1_ZZA_MD_03)
```

```
elseif z == 2 then  
zeitZug2_ZZA_MD_03 = zeit_ZZA_MD03  
zugnrZug2_ZZA_MD_03 = zugnr_ZZA_MD03  
zielZug2_ZZA_MD_03 = ziel_ZZA_MD03  
infoZug2_ZZA_MD_03 = ""  
return (zeitZug2_ZZA_MD_03), (zugnrZug2_ZZA_MD_03),  
(zielZug2_ZZA_MD_03), (infoZug2_ZZA_MD_03)
```

```
end
```

end

-- Am Ende werden die weiteren Züge genauso eingetragen, also einfach einen Bock kopieren, unten anfügen und die Daten und Aufrufe anpassen.

[→ zum nächsten Kapitel](#)

```
function TEE103_MD03(z)
```

-- Zunächst werden wieder die Werte für diesen Zug übergeben:

```
ziel_ZZA_MD03 = "Genf"  
zwziel_ZZA_MD03 = "Bonn Karlsruhe Basel"  
zeit_ZZA_MD03 = "09:31"  
zugnr_ZZA_MD03 = "TEE 103"  
wagenlauf_ZZA_MD03 = "1111s1>"
```

```
if z==0 then
```

-- Die Bedingung z == 0 ist falsch, weshalb die nächsten Zeilen nicht bearbeitet werden.

```
ziel0_ZZA_MD03 = ziel_ZZA_MD03
```

```

zwziel0_ZZA_MD03 = zwziel_ZZA_MD03
zeit0_ZZA_MD03 = zeit_ZZA_MD03
zugnr0_ZZA_MD03 = zugnr_ZZA_MD03
wagenlauf0_ZZA_MD03 = wagenlauf_ZZA_MD03
IRE815_MD03(1)
RE67_MD03(2)
ZZA_MD03(ziel0_ZZA_MD03, zwziel0_ZZA_MD03, zeit0_ZZA_MD03,
zugnr0_ZZA_MD03, wagenlauf0_ZZA_MD03, zeitZug1_ZZA_MD_03,
zugnrZug1_ZZA_MD_03, zielZug1_ZZA_MD_03, infoZug1_ZZA_MD_03,
zeitZug2_ZZA_MD_03, zugnrZug2_ZZA_MD_03, zielZug2_ZZA_MD_03,
infoZug2_ZZA_MD_03)

```

```
elseif z == 1 then
```

-- Nun wird geprüft, ob z = 1 ist. Da das beim ersten Aufruf der Fall ist, werden also die folgenden Zeilen abgearbeitet. Dieses Mal werden die notwendigen Werte in Variablen mit dem Zusatz 1 geschrieben...

```

zeitZug1_ZZA_MD_03 = zeit_ZZA_MD03
zugnrZug1_ZZA_MD_03 = zugnr_ZZA_MD03
zielZug1_ZZA_MD_03 = ziel_ZZA_MD03
infoZug1_ZZA_MD_03 = ""

```

-- ... und mittels `return` an die aufrufende Funktion (`ICE218_MD03(z)`) zurückgegeben.

```

return (zeitZug1_ZZA_MD_03), (zugnrZug1_ZZA_MD_03),
(zielZug1_ZZA_MD_03), (infoZug1_ZZA_MD_03)

```

-- Mit `return` wird diese Routine auch gleichzeitig beendet, die folgenden Zeilen liest EEP gar nicht mehr, sondern [springt an die Stelle zurück](#), von wo er aufgerufen wurde.

```

elseif z == 2 then
    zeitZug2_ZZA_MD_03 = zeit_ZZA_MD03
    zugnrZug2_ZZA_MD_03 = zugnr_ZZA_MD03
    zielZug2_ZZA_MD_03 = ziel_ZZA_MD03
    infoZug2_ZZA_MD_03 = ""
    return (zeitZug2_ZZA_MD_03),
(zugnrZug2_ZZA_MD_03),(zielZug2_ZZA_MD_03), (infoZug2_ZZA_MD_03)
end

```

-- Dieses `end` steht für den Abschluss der if-Anweisung, also das Ende der Schleife. Ansonsten würde das Programm nach weiteren Anweisungen suchen, die nicht vorhanden sind.

`end` -- Dieses `end` beendet die aktuelle Funktion.

```
function IRE815_MD03(z)
```

```
    ziel_ZZA_MD03 = "Würzburg Hbf"  
    zwziel_ZZA_MD03 = "Nordhausen-Erfurt-Suhl"  
    zeit_ZZA_MD03 = "09:35"  
    zugnr_ZZA_MD03 = "IRE 815"  
    wagenlauf_ZZA_MD03 = ""
```

```
    if z==0 then
```

```
-- Diese Bedingung ist falsch
```

```
        ziel0_ZZA_MD03 = ziel_ZZA_MD03  
        zwziel0_ZZA_MD03 = zwziel_ZZA_MD03  
        zeit0_ZZA_MD03 = zeit_ZZA_MD03  
        zugnr0_ZZA_MD03 = zugnr_ZZA_MD03  
        wagenlauf0_ZZA_MD03 = wagenlauf_ZZA_MD03  
        RE67_MD03(1)  
        TEE18_MD03(2)  
        ZZA_MD03(ziel0_ZZA_MD03, zwziel0_ZZA_MD03, zeit0_ZZA_MD03,  
        zugnr0_ZZA_MD03, wagenlauf0_ZZA_MD03, zeitZug1_ZZA_MD_03,  
        zugnrZug1_ZZA_MD_03, zielZug1_ZZA_MD_03, infoZug1_ZZA_MD_03,  
        zeitZug2_ZZA_MD_03, zugnrZug2_ZZA_MD_03, zielZug2_ZZA_MD_03,  
        infoZug2_ZZA_MD_03)
```

```
    elseif z == 1 then
```

```
-- Auch diese Bedingung ist falsch
```

```
        zeitZug1_ZZA_MD_03 = zeit_ZZA_MD03  
        zugnrZug1_ZZA_MD_03 = zugnr_ZZA_MD03  
        zielZug1_ZZA_MD_03 = ziel_ZZA_MD03  
        infoZug1_ZZA_MD_03 = ""  
        return (zeitZug1_ZZA_MD_03),  
        (zugnrZug1_ZZA_MD_03), (zielZug1_ZZA_MD_03),  
        (infoZug1_ZZA_MD_03)
```

```
    elseif z == 2 then
```

```
-- Endlich ein wahres Ergebnis ;)
```

```
        zeitZug2_ZZA_MD_03 = zeit_ZZA_MD03  
        zugnrZug2_ZZA_MD_03 = zugnr_ZZA_MD03  
        zielZug2_ZZA_MD_03 = ziel_ZZA_MD03  
        infoZug2_ZZA_MD_03 = ""  
        return (zeitZug2_ZZA_MD_03), (zugnrZug2_ZZA_MD_03),  
        (zielZug2_ZZA_MD_03), (infoZug2_ZZA_MD_03)
```

```
-- Und wieder zurück
```

```
    end
```

```
end
```

```
-- Ende Programmblock Anzeigen Gleis 3 -----
```

5.5 Anzeigen Typ G und H

Diese Anzeigen werden speziell für S- und U-Bahnen verwendet. Die Programmierung ist etwas aufwendiger. Wenn Sie mit den ersten drei Modellen zurechtkommen, können Sie mit Hilfe der Lua-Datei der Demoanlage ihre eigenen Anzeigen steuern. Dazu sind entsprechende Erklärungen in der Datei selbst enthalten.

6 Symbolschriftart Bahnsymbole Epoche VI

Speziell für den Einsatz mit den Zugzielanzeigen, aber auch zur Beschriftung von anderen Objekten mit Symbolen, habe ich die Symbolschrift „Bahnsymbole Epoche VI“ entwickelt. Mit ihr werden die Bahnsteigabschnitte und die Wagenstände angezeigt.

- Jeder Buchstabe (von A – I) entspricht genau einer Wagenlänge
- Ein Leerzeichen ist genau ein Viertel einer Wagenlänge
- Ausnahme: Die Wagenlänge des abgechrägten Front- bzw. Heckwagens beträgt ein Buchstabe plus ein Leerzeichen

Auf diese Art können die Bahnsteigabschnitte an die jeweiligen Gegebenheiten angepasst werden. In der Regel sind die Buchstaben auf den Bahnsteigen 53 Meter = 2 Wagenlängen voneinander entfernt, wobei sich die Wagenlänge auf die alten IC-Züge bezieht (26,40 m pro Wagen).

6.1 Schriftart „Bahnschrift“

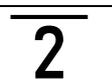
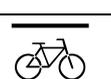
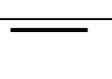
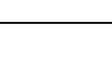
Die Schriftart „Bahnschrift“ ist eine Schriftart, die von Microsoft entwickelt wurde und in aktuellen Windows-10 Versionen enthalten ist.

Für ältere Windows-Versionen ist die Schriftart ebenfalls im Paket enthalten.

6.2 Installation der Schriftarten

Die Schriftarten sind nach der Installation des gesamten Sets mit dem EEP-Modellinstaller im EEP-Ordner unter Ressourcen/Fonts zu finden. Die Installation erfolgt mit Doppelklick auf die jeweilige Schriftart, anschließend einem Klick auf die Schaltfläche „Installieren“. Wenn die Schriftart bereits installiert sein sollte, fragt Windows, ob die bestehende Schriftart überschrieben werden soll. Diese Abfrage ist mit „Nein“ abzuberechnen.

6.3 Symboltabelle

Wagenstände				Gleisabschnitte			
L		R		A	A	B	B
l		r		C	C	D	D
<		>		E	E	F	F
1		2		G	G	H	H
c		s		I	I		
f		h					
w		0					
S		T					
Weitere Symbole							
3		4		5		6	
7		8		9			
!		§		\$		%	
&		/					

Die Schrift wird in den nächsten Wochen um etliche weitere Symbole ergänzt. Auf meiner Internetseite <https://db-eep.de/Schriften> wird es dazu die entsprechenden Updates geben.

Ich wünsche viel Spaß mit diesem Set !

Fragen, Anregungen oder Fehler gefunden?

Am besten ist, das Anliegen im offiziellen EEP-Forum einzustellen:

<https://www.eepforum.de/forum/index.php?board/359-db2-dieter-bauer/>

Rechtliche Hinweise

Die Demo-und Ausbauanlage kann und darf als Bestandteil dieses Sets nicht weitergegeben werden. Sie zu Demonstrations- und Vorführzwecken jeder Art, also auch öffentlich und kommerziell, zu nutzen, ist sowohl im Originalzustand wie auch weiter ausgebaut bzw. verändert ausdrücklich erlaubt.

Quellennachweise für die Modelle:

Eine oder mehrere Texturen dieser 3D-Modelle wurden mit Texturen von Textures.com erstellt. Diese Bilder dürfen nicht wiederverkauft werden. Für weitere Informationen besuche

www.textures.com

Originaltext:

One or more textures on this 3D-model have been created with images from Textures.com. These images may not be redistributed by default. Please visit www.textures.com for more information.

Einige oder mehrere Texturen dieser 3D-Modelle wurden mit Grafiken "[Designed by Freepik](#) " erstellt.

Autor

Dieter Bauer

Frauenstraße 13

89537 Giengen an der Brenz

Web: <https://db-eep.de>

Email: dieter.bauer@db-eep.de