

Dokumentowanie



Wskaźnik miejsca przeznaczenia pociągu epoka VI DB2
(V15NDB20015)



<https://db-eep.de>

Spis treści

1 Przegląd modelu	3
1.1 Wyświetlanie miejsca docelowego pociągu	3
1.2 Znaki stacji	5
1.3 Czcionki i inne pliki	7
2 Przedmowa	8
3 Montaż	8
3.1 Wyświetlacze celów podróży pociągu	8
3.2 Znak stacji	9
3.3 Odcinki torów/numer torów	9
4 Etykieta	9
4.1 Znak stacji	9
4.2 Wyświetlacze celów podróży pociągu	9
4.2.1 Typy A i B	10
4.2.2 Typy C i D	11
4.2.3 Typy E i F	12
4.2.4 Typy G i H	12
5 Sterowanie wyświetlaczem	12
5.1 Podstawowe wpisy w pliku Lua	13
5.2 Typy E i F	14
5.3 Typy A i B	16
5.4 Typy C i D	18
5.5 Typy G i H	24
6 Symbol czcionki Bahnsymbole Epoche VI	24
6.1 Font „Bahnschrift“	24
6.2 Instalacja czcionek	24
6.3 Symboltabelle	25
Znaleziono pytania, sugestie lub błędy?	26
Informacje prawne	26
Źródło dla modeli:	26
Oryginalny tekst:	26
Autor	26

1 Przegląd modelu

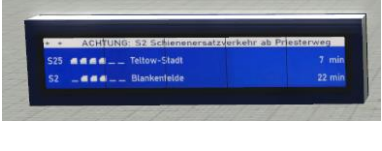
Wyświetlane są epokowe wskazania stacji docelowej pociągu VI, na które składają się następujące modele:

W zwykłej czcionce jest to nazwa taka, jaką znajduje się w EEP.

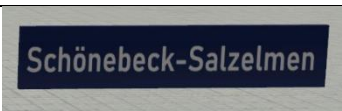
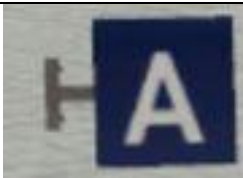
Nazwa pliku jest napisana małymi literami.

1.1 Wyświetlanie miejsca docelowego pociągu

	Nieruchomość:	W nieruchomościach / Ruch drogowy / Akcesoria do platform
Wyświetlacze celów podróży pociągu z 2 x 5 wyświetlaczami LCD Z funkcją etykietowania i kleszcza (można wyłączyć)	- Bieg rydwanów - Wymiary bez uchwytu: 188 x 77 x 44 cm - Domyslna wysokość montażu: 2,40 m - Uchwyt z lewej i prawej indywidualnie wysuwany - Długość uchwytu maks. 74 cm	
	Wskaznik celu pociągu A, Era VI DB2 ZZA6ADB2.3dm	Typ A: 2 linie punktów trasy - z ochrona ptaków
	Wskaznik celu pociągu B, Era VI DB2 ZZA6BDB2.3dm	Typ B: 2 linie punktów trasy - bez ochrona ptaków
	Wskaznik celu pociągu C, Era VI DB2 ZZA6CDB2.3dm	Typ C: - Wyświetlanie pociągów podążających w ślad za pociągiem - z ochrona ptaków
	Wskaznik celu pociągu D, Era VI DB2 ZZA6DDB2.3dm	Typ D: - Wyświetlanie pociągów podążających w ślad za pociągiem - bez ochrona ptaków

Wyświetlacze celów podróży pociągu z 1 x 5 wyświetlaczami LCD Z funkcją etykietowania i kleszcza (można wyłączyć)	• Wymiary bez uchwytu: 188 x 48 x 44 cm • Domyslna wysokość montażu: 2,68 m • Uchwyt z lewej i prawej indywidualnie wysuwany • Długość uchwytu maks. 46 cm	
	Wskaznik celu pociągu E, Era VI DB2 ZZA6EDB2.3dm	Typ E: • Posrednie miejsca docelowe • Bieg rydwanow • z ochrona ptakow
	Wskaznik celu pociągu F, Era VI DB2 ZZA6FDB2.3dm	Typ F: • Posrednie miejsca docelowe • Bieg rydwanow • bez ochrona ptakow
	Wskaznik celu pociągu G, Era VI DB2 ZZA6GDB2.3dm	Typ G: • dwa miejsca docelowe (S-Bahn) • Bieg rydwanow • z ochrona ptakow
	Wskaznik celu pociągu H, Era VI DB2 ZZA6HDB2.3dm	Typ H: • dwa miejsca docelowe (S-Bahn) • Bieg rydwanow • bez ochrona ptakow
Polacy dla wskaźników przeznaczenia pociągu		Wysokość: 3,67 m (+ 50 cm pod ziemią) Grubość masztu: 20 x 20 cm: 8 cm odpowiedni dla wskaźnika celu podróży pociągu o szerokości 188 cm
	Wskaznik celu maszt 1-stronny DB2 ZZAT1DB2.3dm Wskaznik celu maszt B 1-stronny DB2 ZZAT3DB2.3dm	Jednostronny maszt dla wskaźnika miejsca przeznaczenia pociągu szerokość: 2,21 m, z ochrona ptakow bez ochrona ptakow
	Wskaznik celu maszt B 1-stronny DB2 ZZAT2DB2.3dm Wskaznik celu maszt 1-stronny DB2 ZZAT4DB2.3dm	Dwustronny maszt dla wskaźnika celu podróży pociągu szerokość: 4,16 m, z ochrona ptakow bez ochrona ptakow

1.2 Znaki stacji

Znaki stacji (tablice stacji) - dowolnie opisywane	Domyślna wysokość montażu: 2.40, Grubość: 8 cm (zmienna przez skalowanie) Epoka VI
	Znak stacji 100x40, Ep VI DB2 <i>SA6_7DB2.3dm</i>
	Znak stacji 150x40, Ep VI DB2 <i>SA6_6DB2.3dm</i>
	Znak stacji 120x50, Ep VI DB2 <i>SA6_1DB2.3dm</i>
	Znak stacji 150x50, Ep VI DB2 <i>SA6_2DB2.3dm</i>
	Znak stacji 200x50, Ep VI DB2 <i>SA6_3DB2.3dm</i>
	Znak stacji 250x50, Ep VI DB2 <i>SA6_4DB2.3dm</i>
	Znak stacji 300x60, Ep VI DB2 <i>SA6_5DB2.3dm</i>
	Znak stacji 40x40, Ep VI DB2 Dla odcinków torów lub numerów torów <i>SAATDB2.3dm</i>

Belka podporowa dla Znak stacji (i inne znaki)	Długość 2,60 m plus 50 cm pod podłogą, grubość: 4 x 4 cm Stal ocynkowana
	Znak stacji podpórka 80cm rozstaw DB2 <i>SA6T1DB2.3dm</i>
	Znak stacji podpórka 120cm rozstaw DB2 <i>SA6T2DB2.3dm</i>
	Znak stacji podpórka 160cm rozstaw DB2 <i>SA6T3DB2.3dm</i>
	Znak stacji podpórka 210cm rozstaw DB2 <i>SA6T4DB2.3dm</i>

1.3 Czcionki i inne pliki

	Teczka	W folderze Ressourcen/Fonts
	BahnsymboleVI.ttf	Symbole kolejowe Ep. VI <i>Die Grafik zeigt nur einen Ausschnitt aller in der Schrift enthaltenen Symbole</i>
	Bahnschrift.ttf	Bahnschrift
	Bloki	W sektorze nieruchomości
	ZZA_A_Mast_eins.bl3	
	ZZA_A_Mast_zweis.bl3	
	Zakład demonstracyjny	W folderze Załączniki/ZZA Demo
	ZZA-Demo.anl3	
	Demovideo:	YouTube

2 Przedmowa

Niniejsze instrukcje opisują strukturę i funkcje modeli tak szczegółowo, jak to tylko możliwe, aby początkujący użytkownicy EEP mieli możliwość jak najszybszego wykorzystania różnych możliwości tych nowych wyświetlaczy miejsc docelowych pociągów. Doświadczeni użytkownicy z pewnością znajdą kilka paragrafów, które nie mówią im nic nowego. Mogą jednak przeoczyć dokładność opisu, tak aby nowicjusze i niedoświadczony EEP'ler czerpali więcej radości z EEP.

Jeśli coś jest nadal niejasne, można się z nami skontaktować poprzez kanały na końcu dokumentacji i chętnie odpowiem na wszelkie pytania.

3 Montaż

3.1 Wyświetlacze celów podróży pociągu

Wyświetlacze miejsca przeznaczenia pociągu mogą być podwieszane zarówno na dostarczonych w tym celu masztach, jak i na konstrukcjach stropowych, np. dachach peronów.

Wysokość dolnej krawędzi wynosi 2,40 m nad ziemią dla wyświetlaczy dwuwierszowych i 2,69 m nad ziemią dla wyświetlaczy jednowierszowych. W przypadku użycia dostarczonych masztów do podwieszenia, odpowiada to dokładnie właściwej wysokości podwieszenia, jeśli maszt jest używany w tym samym miejscu. Przesunięcie boczne wyświetlaczy wynosi 2,08 m, na osi wzdłużnej należy wybrać taką samą wartość jak dla masztu.

Oczywiście wartości te odnoszą się tylko do instalacji prostokątnej. Dwa bloki składające się z masztu i wskaźnika (wskaźników) są dołączone w celu uproszczenia instalacji w innych kątach. Po ustawieniu bloku, wyświetlacz może być wymieniany w edytorze 3D. W tym celu należy wybrać żądane wyświetlanie na liście modeli, kliknąć prawym przyciskiem myszy na wyświetlaczu, który ma zostać zastąpiony, a następnie wybrać element "Wymiana na". Wszystkie wyświetlacze w tym zestawie są kompatybilne ze sobą, dzięki czemu nawet mniejsze wyświetlacze są automatycznie umieszczane we właściwej pozycji.

Wewnątrz wyświetlaczy znajdują się dwa uchwyty, które mogą być przedłużane do żądanej długości w razie potrzeby. W tym celu należy wybrać lewy, a następnie prawy uchwyt we właściwościach obiektu i za pomocą suwaka przesunąć go na żądaną wysokość. Podwieszone do masztów pręty te są umieszczone dokładnie w płycie montażowej, co sprawia wrażenie, że wyświetlacze są przykręcone do masztu.

3.2 Znak stacji

Znaki stacji są standardowo wyposażone w dolną krawędź o długości 2,40 m. W zestawie znajdują się pasujące stoiska, które mają takie same współrzędne jak znaki.

Oczywiście, wielkość znaków może być również regulowana. Na przykład w filmie demonstracyjnym nad ławką na stacji znajduje się znak, a istniejący znak na oryginalnym brzegu jest całkowicie ukryty. Osiągnięto to poprzez potrojenie siły (oś Y) i lekkie skalowanie osi Z (1,02). Znak może być również przymocowany do ściany stacji.

3.3 Odcinki torów/numer torów

Znaki 40 x 40 cm posiadają stabilny uchwyt boczny, który można zamontować za pomocą 2 wkrętów na słupach latarni nawigacyjnych, wspornikach dachowych platform lub ścianach. Żądaną literę lub numer należy wprowadzić za pomocą funkcji etykietowania.

4 Etykietka

4.1 Znak stacji

Etykietowanie etykiet jest bardzo proste: od EEP 15, przycisk "Label" został włączony do właściwości obiektu. Jeśli zostanie to kliknięte, otworzy się kolejne okno, w którym w dolnej części zostanie wprowadzony żądany tekst.

W razie potrzeby można dokonać dalszych zmian: Czcionka może być zmieniona, również rozmiar czcionki i reprezentacji (pogrubiona, lekka, kursywa). Znacznik kontrolny dla "wyśrodkowany" zmienia ułożenie tekstu w polu określonym przez projektanta.

Wreszcie, czcionka i kolor tła mogą zostać zmienione. Należy pamiętać, że kolor tła wpływa tylko na litery, które są rzeczywiście napisane. Jeśli chcesz napisać niebieski tekst na białym tle, obszar jest tylko nieco wyższy od litery jest biały. Jeśli chcesz, aby biały pasek dalej w prawo i/lub w lewo (jak w zaznaczonych tekstach na wyświetlaczu stacji docelowej pociągu), możesz wypełnić żądaną szerokość odpowiednią ilością spacji. Wymaga to odrobiny cierpliwości, ponieważ najpierw należy sprawdzić, ile przestrzeni jest optymalnych.

4.2 Wyświetlacze celów podróży pociągu

Wyświetlacze miejsc docelowych pociągu mają większą liczbę pól tekstowych. Niektóre z nich są wypełniane bezpośrednio podczas wstawiania, inne przez programowanie Lua (patrz poniżej).

4.2.1 Typy A i B

Wyświetlacze zawierają następujące pola tekstowe:

Tekst nr.	Zawartość pola z przodu	Tekst nr.	Contenu du champ verso
1	Miejsce przeznaczenia	2	Miejsce przeznaczenia
3	Cele pośrednie 1	4	Cele pośrednie 1
5	Cele pośrednie 2	6	Cele pośrednie 2
7	Czas	8	Czas
9	Nr pociagu	10	Nr pociagu
11	Bieg rydwanów	12	Bieg rydwanów
13	Odcinki peronów	14	Odcinki peronów
15	Numer toru	16	Numer toru
17	Tekst informacyjny (kleszczyk)	Wyświetlane po obu stronach	

Z przyczyn technicznych i optycznych wpisy muszą być dokonywane po obu stronach. Pola 13-16 są od razu oznakowane statycznie. Informacje na temat odcinków peronów opisane są poniżej pod symbolem kolejowym Era VI.

4.2.2 Typy C i D

Wyświetlacze zawierają więcej pól tekstowych, ponieważ następne dwa ruchy również zostaną ogłoszone.:

Tekst nr.	Zawartość pola z przodu	Tekst nr.	Contenu du champ verso
Wpisy dotyczące głównego pociągu			
1	Miejsce przeznaczenia	2	Miejsce przeznaczenia
3	Cele pośrednie	4	Cele pośrednie
5	Czas	6	Czas
7	Nr pociągu	9	Nr pociągu
9	Bieg rydwanów	10	Bieg rydwanów
Wpisy dla pierwszego pociągu uzupełniającego			
11	Czas Kolejka następująca pociągu 1	12	Czas Kolejka następująca pociągu 1
13	Nr pociągu Kolejka następująca pociągu 1	14	Nr pociągu Kolejka następująca pociągu 1
15	Cel Kolejka następująca pociągu 1	16	Cel Kolejka następująca pociągu 1
17	Information Kolejka następująca pociągu 1 (tutaj możliwy jest krótki tekst w kolorze niebieskim na białym)	18	Information Kolejka następująca pociągu 1
2. Kolejka następująca pociągu			
19	Czas Kolejka następująca pociągu 2	20	Czas Kolejka następująca pociągu 2
21	Nr pociągu Kolejka następująca pociągu 2	22	Nr pociągu Kolejka następująca pociągu 2
23	Cel Kolejka następująca pociągu 2	24	Cel Kolejka następująca pociągu 2
25	Information Kolejka następująca pociągu 2	26	Information Kolejka następująca pociągu 2
Ogólne informacje statyczne			
27	Odcinki peronów	28	Odcinki peronów
29	Numer toru	30	Numer toru
31	Nagłówek "Kolejne pociągi".	32	Nagłówek "Kolejne pociągi".
Ze względów technicznych, na końcu wpisów zawsze pojawia się znacznik z ogólnymi informacjami dla pasażerów.			
33	Tekst informacyjny (kleszczyk)	Wyświetlane po obu stronach	

4.2.3 Typy E i F

Wyświetlacz ten jest tak samo funkcjonalny jak wyświetlacze typu A i B, ale posiada tylko jeden rząd LCD, a więc mniej pól.

Tekst nr.	Zawartość pola z przodu	Tekst nr.	Contenu du champ verso
1	Miejsce przeznaczenia	2	Miejsce przeznaczenia
3	Cele pośrednie 1	4	Cele pośrednie 1
5	Czas	6	Czas
7	Nr pociagu	8	Nr pociagu
9	Bieg rydwanów	10	Bieg rydwanów
11	Odcinki peronów	12	Odcinki peronów
13	Numer toru	14	Numer toru
15	Tekst informacyjny (kleszczyk)	Wyświetlane po obu stronach	

4.2.4 Typy G i H

Wyświetlacz ma zasadniczo inną strukturę niż poprzednie, dlatego wyświetlacze nie muszą być oddzielnie oznakowane na przedniej i tylnej stronie.

Tekst nr.	Feldinhalt erste Zeile	Tekst nr.	Feldinhalt zweite Zeile
1	Linia 1. pociagu	5	Linia 2. pociagu
2	Bieg rydwanów 1. pociagu	6	Bieg rydwanów 2. pociagu
3	Cel 1. pociagu	7	Cel 2. pociagu
4	Abfahrt 1. pociagu	8	Abfahrt 2. pociagu
Statische Angaben			
9	Tekst "min" 1. pociagu	10	Tekst "min" 2. pociagu
Und wieder zum Schluss:			
11	Tekst informacyjny (kleszczyk)		

5 Sterowanie wyświetlaczem

Teraz chodzi o rzeczywiście interesującą część. Cechą szczególną wyświetlaczy miejsca docelowego pociagu jest to, że można je dowolnie etykietować - dzięki nowej funkcji etykietowania w EEP 15.

Ma to jednak sens tylko wtedy, gdy używasz języka skryptowego Lua zawartego w EEP. Ale nie martw się, nie musisz być programistą, nawet matematykiem, aby kontrolować wyświetlacze. W najprostszym przypadku wystarczy skopiować odpowiedni tekst poniżej (kod programu), który jest również zawarty w systemie demo, i wprowadzić własne cele i czasy. Kod jest specjalnie napisany tak, że nie musisz się zastanawiać, gdzie coś jest wprowadzane, ale wpisy są dość oczywiste.

Jeśli masz wiedzę z zakresu programowania lub doświadczenie z Lua, będziesz w stanie zoptymalizować kod i rozszerzyć go o dodatkowe funkcje. Ale nawet mniej doświadczeni użytkownicy będą w stanie wdrożyć kilka dodatkowych funkcji z niewielkim przeszkoleniem.

Instrukcje nie są zgodne z logiczną kolejnością typów A, B, C H, ale zaczynają się od najprostszego wyświetlacza.

5.1 Podstawowe wpisy w pliku Lua

Przed wszystkim chciałbym polecić wszystkim, aby zainstalowali następującą bezpłatną instalację:

1. notepad++ - edytor

Moim zdaniem najbezpieczniejszym i najłatwiejszym do pobrania w języku niemieckim jest Heise.de:

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

2. rozszerzenie komendy Lua specjalnie dla EEP, pobierz z instrukcjami na forum:

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

Podświetlenia kolorystyczne i wcięcia, jak również większy edytor tekstu sprawiają, że kod jest znacznie bardziej czytelny. Numery wierszy są użyteczne, jeżeli na przykład komunikat o błędzie jest wyrzucany z powodu błędu zapisu, w którym EEP określa numer odpowiedniego wiersza. Bez Notepad+++ możesz być w stanie szukać przez długi czas.

Linie programu odnoszą się do systemu demo i mogą być tam śledzone.

Zielony to komentarze. Zaczynasz od dwóch myślników:

-- Następująca linia kodu została stworzona przez Benjamin Hoge (BH2 - emaps-eep.de),
-- że parametry te mogą być przekazywane w punktach kontaktowych podczas wywołania funkcji Lua.
-- W tym przykładzie jest on używany do sterowania wyświetlaczami na torze 3.

```
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})
```

Kod programu jest zaznaczony na szaro - w Notatniku+++ jest on podzielony na różne kolory, co jest pomocne dla podstawowego zrozumienia, ale nie tak ważne. Przecież nie ma to być przecież kurs programowania ;) Tylko wywołania funkcji są przechowywane w

granatowy, parametry transferu nie są drukowane pogrubioną czcionką.

5.2 Typi E i F

-- Poniższy blok steruje wyświetlaczami na torze 4 -----

Następujące teksty mają być zmieniane automatycznie:

- 1 Miejsce przeznaczenia frontowy
- 2 Miejsce przeznaczenia cofnąć się
- 3 Cele pośrednie frontowy
- 4 Cele pośrednie cofnąć się
- 5 Czas frontowy
- 6 Czas cofnąć się
- 7 Nr pociągu frontowy
- 8 Nr pociągu cofnąć się

Pozostałe etykiety są już ustawione we właściwościach obiektu lub nie są wymagane w tym przykładzie.

W pierwszej kolejności przychodzi funkcja wywołania. To decyduje o tym, jakie wartości należy wprowadzić dla każdego ruchu.

function IRE629_MD04() -- IRE 629 podróżuje w Magdeburgu z platformy 4

-- Ta nieco uciążliwa nazwa definiuje unikalną funkcję, która nie występuje (nie może wystąpić) po raz drugi.

<code>ziel_ZZA_MD04 = "Nürnberg Hbf"</code>	-- Cel podróży
<code>zwziel_ZZA_MD04 = "Halle (Saale) Erfurt"</code>	-- wybierany Cele pośrednie
<code>zeit_ZZA_MD04 = "09:27"</code>	-- czas odjazdu
<code>zugnr_ZZA_MD04 = "IRE 629"</code>	-- Nazwa pociągu

-- Teraz wartości te są przenoszone do aktualnej funkcji, która dokonuje wpisów:

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

-- każda funkcja musi być zakończona komunikatem "end".

-- (ZZA dla dalszych pociągów są kontrolowane zgodnie z następującym blokiem)

-- „Objazd” szybko staje się jasny, gdy spojrzymy na długość kolejnej funkcji wprowadzania danych. Jeśli wpis został dokonany bezpośrednio, ten długi blok musiałby być napisany za każdym razem, niepotrzebnie nadmuchując kod programu.

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

-- Zmienne w nawiasach nadają tej funkcji wartości do wprowadzenia.

-- Teraz pierwszy ZZA jest wypełniony prawidłowym tekstem: Liczba po skrócie znajduje się we właściwościach obiektu pod "Lua Name". Potrzebna jest tylko liczba, reszta nie jest ważna.

```
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 )
```

-- Drugi ZZA ma numer 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 )
```

-- Pozostałe wyświetlacze docelowe są wypełniane w ten sam sposób. W tym przypadku jest to jeszcze sześć ogłoszeń. Można je całkowicie odczytać w pliku Lua systemu.

-- Doświadczony programista Lua może to skrócić za pomocą pętli.

end

--A nad pierwszym IRE ogłaszane są następujące pociągi. Spróbuj zrozumieć, co jest dla Ciebie ważne:

```
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

-- Jestem przekonany, że teraz wiesz, że musisz wprowadzić tylko kilka danych dotyczących celu, czasu i obrotów. Reszta może być po prostu skopiowana i przypisany jeden z odpowiednich numerów torów i pociągów.

--Pozostałe pociągi można znaleźć w obiekcie demonstracyjnym

-- Koniec bloku programu Ścieżka wyświetlania 4 -----

5.3 Typi A i B

-- Poniższy blok steruje wyświetlaczami na torze 2 -----

- Blok jest podobny do ścieżki 4, ale ma więcej wpisów. Wartości ZZA w Magdeburskim Torze 2 wprowadza się w następnym bloku. Ponownie jednak jako "zmiennne zastępcze" używane są tylko zmienne, które są następnie wprowadzane w następujących funkcjach przez poszczególne ruchy.

W przypadku toru 2 pokazuję, w jaki sposób wjazd jest wyznaczany przez sam pociąg, gdy wychodzi on z zajezdni i przecina tam punkt kontaktowy.

Gdy pociąg opuszcza stację, punkt kontaktowy po sygnale wyjścia wyzwala funkcję "no display_MD02()", która opróżnia displayboard.

Ten typ programowania działa tylko wtedy, gdy odjazd z zajezdni, a tym samym przekroczenie punktu kontaktowego w celu wygenerowania wyświetlacza, ma miejsce PO odjeździe poprzedniego pociągu ze stacji.

**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 )
```

-- dla innych tablic wyników na torze 2..

end

-- ICE 152 wychodzi ze składu o godzinie 9:19, sterowany przez wyjście ze składu i jeździ przez styk pojazdu, który uruchamia tę funkcję:

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>"
```

-- Lista symboli dla jazdy samochodem znajduje się poniżej.

-- Następująca linia przenosi wartości wprowadzone powyżej do funkcji ZZA_MD002 powyżej, która oznacza wskaźniki celu pociągu

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

end

-- W ten sposób każdy pociąg na tor 2 zostanie wpisany tutaj

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

-- itp. Dopóki funkcja pustego wyświetlacza nie zostanie zakończona:

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

-- Koniec bloku programu Ścieżka wyświetlania 2 -----

5.4 Typi C i D

Teraz staje się to trochę bardziej skomplikowane. Oznacza to, że wygląda to bardziej skomplikowane, ale ci, którzy udało się z poprzednich reklam szybko zrozumieć tę formę.

W tych wyświetlaczach, dwa kolejne pociągi są dodawane do następnego poruszającego się pociągu w celu poprawy informacji dla pasażerów. Dlatego też dodawana jest pętla "if... else", jedna z najprostszych i najbardziej efektywnych procedur programu w historii.

Następujący blok steruje wyświetlaczami na torze 3 -----

-- Zasadniczo wywołanie tej procedury programu różni się od poprzednich tym, że jest wywoływane przez ten pociąg po opuszczeniu stacji, z wyjątkiem pierwszego pociągu.

Oznacza to, że następny pociąg - w tym dwa kolejne pociągi - zostanie ogłoszony natychmiast.

Po raz pierwszy po raz kolejny, teraz jeszcze bardziej rozbudowane, funkcje do wprowadzania wartości:

```
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 , zwziel0_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 )
```

-- Jak zwykle, blok ten jest kopiowany i dostarczany z odpowiednim numerem wyświetlacza - lub sterowany przez pętlę.

end

-- Teraz postępuj zgodnie z funkcjami określania wpisów. Jeśli funkcja ta jest wywoływana przez punkt kontaktowy, otrzymuje wartość 0. Wywołanie funkcji Lua w punkcie kontaktowym (za stacją) to zatem "ICE218_MD03(0)". W funkcji z przyjmuje przekazaną wartość.

```
function ICE218_MD03(z)
```

-- Po pierwsze, wartości dla tego ruchu są przekazywane:

```
ziel_ZZA_MD03 = "München Hbf"  
zwziel_ZZA_MD03 = "Leipzig Nürnberg"  
zeit_ZZA_MD03 = "09:26"  
zugnr_ZZA_MD03 = "ICE 218"  
wagenlauf_ZZA_MD03 = "L22c1R"
```

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

-- warunek z == 0 jest prawdziwy, więc wykonywane są następujące wiersze. Wartości z powyższego są zapisywane do zmiennej z dodatkiem 0.

OTHER: Podwójny znak równości nie jest błędem, ale pokazuje EEP, że powinien porównać wartość z obecnie ma wartość z z wartością 0. W przeciwnym razie program może założyć, że chcemy przypisać tę wartość do zmiennej z, co nie jest prawdą.

```
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
```

-- Teraz wywoływana jest rutynowa procedura programowa pierwszego kolejnego ruchu. Wartość 1 jest przekazywana.

TEE103_MD03(1)

-- Najpierw wykonywana jest procedura [TEE103 MD03](#), więc zobacz co się tam dzieje..

-- Program posiada teraz wartości głównego i pierwszego kolejnego ruchu w pamięci. W związku z tym nadal brakuje drugiej kontynuacji. Wywołanie, wprowadzanie i zwracanie wartości działa w taki sam sposób jak pierwszy kolejny ruch, z tym wyjątkiem, że tym razem wartość 2 jest przekazywana i tym samym druga inna, jeśli jest oceniana w następującej funkcji

IRE815_MD03(2)

-- Więc najpierw do „[Subroutine](#)” ...

-- Mamy teraz wszystkie dane potrzebne do wyświetlania i przekazujemy je do funkcji wprowadzania wartości

```
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)
```

-- Les lignes suivantes ne sont plus lues par EEP

```
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

-- Na końcu pozostałe ruchy są wprowadzane w ten sam sposób, tzn. po prostu skopiować blok, dodać go poniżej i dostosować dane i połączenia.

[→ do następnego rozdziału](#)

function TEE103_MD03(z)

-- Najpierw ponownie przekazywane są wartości dla tego ruchu.:

```
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
```

-- Warunek z == 0 jest false, dlatego następne wiersze nie są przetwarzane.

```
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
```

-- Teraz jest sprawdzane, czy z = 1. Ponieważ tak jest w przypadku pierwszego połączenia, przetwarzane są następujące linie. Tym razem niezbędne wartości są zapisywane w zmiennych z dodatkiem 1....

```

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 = ""

```

-- ... i powrócił do funkcji wywołania (**ICE218_MD03(z)**) za pomocą **return**.

```

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

```

-- Z **return** ta procedura jest zakończona w tym samym czasie, kolejne wiersze nie są już odczytywane przez EEP, [ale wracają do miejsca](#), z którego zostały wywołane..

```

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

```

-- Ten **end** oznacza koniec instrukcji if, tzn. koniec pętli. W przeciwnym razie program wyszukałby dalsze stwierdzenia, które nie istnieją.

end -- Ten **end** kończy bieżącą funkcję

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
```

-- Ten warunek jest fałszywy

```
    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
```

-- Ten warunek jest również błędny

```
    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
```

-- Wreszcie prawdziwy wynik ;)

```
    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)
```

-- [I z powrotem.](#)

```
end
```

end

-- Koniec bloku programu Ścieżka wyświetlania 3 -----

5.5 Typi G i H

Wyświetlacze te są używane w szczególności dla pociągów S-Bahn i U-Bahn. Programowanie jest nieco bardziej złożone. Jeśli jesteś w stanie poradzić sobie z pierwszymi trzema modelami, możesz użyć pliku Lua systemu demo do sterowania własnymi wyświetlaczami. Odpowiednie wyjaśnienia znajdują się w samym pliku..

6 Symbol czcionki Bahnsymbole Epoche VI

Stworzyłem symbol czcionki "Bahnsymbole Epoche VI" specjalnie do użytku z monitorami miejsc przeznaczenia pociągu, ale także do oznaczania innych obiektów za pomocą symboli. Wyświetla sekcje peronów i statusy wagonów.

- Każda litera (od A do I) odpowiada dokładnie jednej długości wagonu.
- Przestrzeń to dokładnie jedna czwarta długości samochodu.
- Wyjątek: Długość przedniego lub tylnego samochodu to jedna litera plus jedna spacja.

W ten sposób sekcje podestu mogą być dostosowane do odpowiednich warunków. Z reguły litery na peronach są oddalone od siebie o 53 metry = 2 długości wagonów, przy czym długość wagonów odnosi się do starych pociągów IC (26,40 m na wagon).

6.1 Font „Bahnschrift“

Czcionka "Bahnschrift" jest czcionką, która została opracowana przez Microsoft i jest zawarta w aktualnych wersjach Windows-10 w Niemczech.

Dla starszych i innych wersji Windows, czcionka jest również dołączona do pakietu.

6.2 Instalacja czcionek

Po zainstalowaniu kompletnego zestawu z instalatorem modelu EEP, czcionki można znaleźć w folderze EEP w Resources/Fonts. Instalacja odbywa się poprzez dwukrotne kliknięcie odpowiedniej czcionki, a następnie kliknięcie przycisku "Install". Jeśli czcionka jest już zainstalowana, Windows pyta, czy chcesz nadpisać istniejącą czcionkę. To zapytanie musi być przerwane z "Nie".

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			
!		§			\$		%	
&		/						

W następnych tygodniach czcionka zostanie uzupełniona o kilka kolejnych symboli. Na mojej stronie internetowej <https://db-eep.de/Schriften> znajdą Państwo odpowiednie aktualizacje.

Życzę Ci dużo zabawy z tym zestawem 😊

Znaleziono pytania, sugestie lub błędy?

Najlepiej ustawić żądanie na oficjalnym forum EEP:

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

Informacje prawne

System demonstracyjny i demontażowy nie może być przekazany w ramach tego zestawu. Można go używać do celów demonstracyjnych i pokazowych wszelkiego rodzaju, także publicznie i komercyjnie, w sposób wyraźny dozwolony w stanie pierwotnym, a także dalej rozwijanym i / lub zmodyfikowanym.

Źródło dla modeli:

Jedna lub więcej tekstur tych modeli 3D została utworzona przy użyciu tekstur Textures.com. Obrazy te nie mogą być sprzedawane. Aby uzyskać więcej informacji, odwiedź www.textures.com

Oryginalny tekst:

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.

Niektóre lub kilka tekstur tych modeli 3D zostały utworzone przy użyciu grafiki "Designed by Freepik".

Autor

Dieter Bauer

Frauenstraße 13

89537 Giengen an der Brenz

Niemcy

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

Email: dieter.bauer@db-eep.de