

Skripte für HL-Signale im Train Simulator (Version 6.11)



Im Rahmen der Entwicklung von Tageslichtsignalen als Ersatz für die veralteten Formsignale einigten sich die Eisenbahnverwaltungen der Ostblockstaaten auf ein weitgehend vereinheitlichtes Signalsystem, das durch die Zusammenfassung von Vor- und Hauptsignalfunktion in einem einzigen Signalschirm als Mehrabschnittssignale gekennzeichnet ist.

Die Signalisierung der erlaubten Höchstgeschwindigkeit erfolgt durch unterschiedliche Anordnung farbiger Lichter (unten Hauptsignalbegriff und oben Vorsignalbegriff).

Bei der Deutschen Reichsbahn wurden diese Signale als HL-Signale bezeichnet und 1958 in die Signalordnung aufgenommen.

Quelle: Wikipedia

Das HL-Signalsystem überrascht vor allem durch eine Vielfalt der Ausführungen der Signalschirme. Oft wurden Signaloptiken, die nicht benötigt wurden, aus Kostengründen demontiert oder gar nicht erst verbaut. So gab es zum Beispiel Vorsignale mit einer einzigen Signaloptik.

Mit diesen Skripten können Sie selbst gebaute HL-Signale verschiedener Bauformen auf ihren Freewarestrecken im Train Simulator verwenden.

Auf Grund der Analogie des EZMG-Signalsystems zu den HL-Signalen, können Sie auch die EZMG-Signale mit diesen Skripten betreiben.

Dieses Paket beinhaltet die Anleitung und Skripte, um selbst erstellte HL-Signale mit den Skript-Modulen aus dem Paket „**Freeware Skript-Module und Signal-Trigger**“ zu einem funktionierenden Signalsystem zu verbinden.

Wenn Sie lediglich fertige Signale nutzen wollen, die für die Schuster/Freeware - Skript-Module erstellt wurden, benötigen Sie dieses Paket nicht, sondern laden Sie sich von Rail-Sim.de das Paket „**Freeware Skriptmodule und Signal-Trigger**“ herunter und installieren dieses mit der Utilities.exe.

Bestandteile der Datei:

- | | |
|----------------------|----------------------------------|
| | Skriptpaket_HL-Signale_V6.11.zip |
| • Anleitung | Skriptpaket_HL-Signale_V6.11.pdf |
| • Excel-Tabelle | Code-Berechnung_HL-Signale.xlsx |
| • Skripte für Source | Skripte_HL-Signale_V6.0.zip |

Indem Sie dieses Paket nutzen, akzeptieren Sie auch die Lizenzbestimmungen.

Bei Problemen/Anregungen bitte über das Forum www.Rail-Sim.de anschreiben.

Viel Erfolg beim Einsatz der Skripte für die HL-Signale wünscht
Mathias Gundlach (Schuster at Rail-Sim)

Inhaltsverzeichnis

1. Einleitung	3
1.1. Einige Vorworte	3
1.2. Lizenzbestimmungen	3
1.3. Installation	3
1.4. Neuerungen gegenüber der vorherigen Version	3
2. Ordnerstruktur	4
3. Vorgaben für 3D-Modelle	4
3.1. Allgemeines	4
3.1.1. Regeln für die Benennung der Licht-Nodes	4
3.1.2. Anmerkungen zu den Child-Namen	5
3.1.3. HP_CODE und VR_CODE	5
3.1.4. Zusatzanzeiger	5
3.2. Vorsignale	6
3.2.1. Vorsignal	6
3.2.2. Vorsignal mit nur einer gelben Optik	6
3.2.3. Vorsignalwiederholer	6
3.3. Hauptsignale (einige Beispiele)	6
3.3.1. Hauptsignal	7
3.3.2. Hauptsignal als Vollausbau	7
3.3.3. Hauptsignal als Minimalvariante	8
3.4. Rangiersignale	8
3.4.1. Wartesignal / Rangiersignal / Ra12	8
3.5. Zusatzsignale und Zusatzanzeiger	8
3.5.1. Fahrtanzeiger	9
3.5.2. Abfahrtsignal	9
3.5.3. Richtungsanzeiger	9
3.5.4. Gleiswechselanzeiger	9
4. Skripte.....	10
5. BlueprintEditor	13
5.1. Grundsätzliches	13
5.2. Köpfe	13
5.3. Einstellungen für Signale und Zusatzanzeiger	13
5.4. Export des BluePrints.....	16
5.5. Einsatz der Signale in einer Train Simulator Strecke.....	16

1. Einleitung

1.1. Einige Vorworte

Wer sich mit dem Bau von HL-Signalen beschäftigt, und diese Skripte verwenden möchte, sollte sich grundsätzlich mit dem Signalkpaket Deutsche_HV-Signale vom SignalTeam auseinandersetzen, um die allgemeinen Sonderfunktionen der Skripte zu verstehen.

Bezüglich der Sonderfunktionen im Zusammenhang mit Triggern und Einstellungen im Signal-Flyout unterscheiden sich diese Skripte nicht von denen, die zum Beispiel für die HV-Signale unter „**Schuster/SignalTeam**“ veröffentlicht wurden.

Trigger sind nicht Bestandteil dieses Skriptpaketes. Diese werden später separat unter „**Schuster/Freeware**“ veröffentlicht und können dann für alle Signalsysteme dieser Veröffentlichungsreihe gemeinsam genutzt werden. Gleichfalls können auch alle bisher veröffentlichten Trigger aus anderen Signalkpaketen, soweit sie dem aktuellen Stand entsprechen, genutzt werden.

1.2. Lizenzbestimmungen

Das Paket wird als Freeware auf Rail-Sim (www.Rail-Sim.de) angeboten und darf nicht auf weiteren Plattformen ohne meine Erlaubnis angeboten werden.

Die Skripte und Module dürfen ausschließlich auf **Freeware-Strecken** verwendet werden. Es ist nicht gestattet, die Skripte oder Module für kommerzielle Strecken zu verwenden.

Die Module dürfen nicht geändert, angepasst oder in anderen Provider-/Produktordnern gespeichert und dort heraus geladen werden. Updates stelle ich ausschließlich selbst zur Verfügung.

Die Module dürfen nicht als Bestandteil von Strecken oder Signalkpaketen verteilt werden. Sie dürfen nur per Link aus ihrer ursprünglichen Downloadquelle unter Rail-Sim angeboten werden.

Die im Klartext gelieferten Skripte dürfen angepasst, kopiert, exportiert und im eigenen Source-Ordner verwendet werden. Sie dürfen dann auch im OUT-Format innerhalb der eigenen Streckenpakete verpackt und verteilt werden.

Sollte Bedarf bestehen, dass diese Module in **Payware-Projekten** verwendet werden, so bitte ich um Kontaktaufnahme per E-Mail an Railworks@mgundlach.de. Die Module werden dann durch mich an den entsprechenden Provider- und Produktordner angepasst und eine Lizenz zur Verwendung vergeben.

1.3. Installation

Das Skriptpaket ist eine ZIP-Datei und muss mit einem geeigneten Programm entpackt werden. Die Skripte werden dann in den Source-Ordner kopiert, in dem später auch die eigenen Signale abgelegt werden.

1.4. Neuerungen gegenüber der vorherigen Version

Version 6.11

- Hauptsignale ohne gelber und grüner Signaloptyk verbleiben nun standardmäßig auf Hp0
- Aktivierung des Ersatzrot mit Hp0-Trigger (ZE)
- Korrektur Lichtstreifenzuordnung
- Aktivierung des Gegengleis-Ersatzsignals Zs8
- Geschwindigkeitsabhängige Signalisierung bei fehlenden Signaloptyken verbessert
- Filtermöglichkeit von Debug-Nachrichten
- Überarbeitung dieser Anleitung
- Die notwendigen Skript-Module sind nicht mehr Bestandteil dieses Paketes und nur noch im Paket **Freeware Skript-Module und Signal-Trigger** enthalten.

Version 6.01

- Erste veröffentlichte Version des Skriptpaketes

2. Ordnerstruktur

- Hauptordner
 - **Assets\Schuster\Freeware\RailNetwork\Signals\German HL**

In diesem Ordner befindet sich lediglich die Optionsdatei. Diese muss dort unbedingt gespeichert bleiben und dient der Einstellung von einigen Optionen für das Signalsystem.

- Module

Der Ordner Module ist ein Unterordner von „German HL“ und enthält alle Skript-Module im OUT-Dateiformat, die aus den Skripten heraus nachgeladen werden. Diese Skriptmodule werden in keinem Blueprint direkt angegeben!

- Skripte

Im Bereich Source liegen in einem Unterordner der sinnvoller Weise „Skripte“ lauten sollte, unterhalb des eigenen Signalverzeichnisses alle Skripte mit der Dateiendung „Lua“. Diese Skripte werden dann im **BlueprintEditor** jeweils angegeben. In diesen Skripten befindet sich dann auch der Verweis („require....“) auf das entsprechende Skript-Modul.

3. Vorgaben für 3D-Modelle

Damit die Signale mit den Skripten funktionieren, müssen verschiedene Bedingungen erfüllt sein:

- Die Namen der Licht-Nodes im 3D-Modell müssen mit denen im Skript übereinstimmen
- Die Child-Namen der Signalschirme, die im BlueprintEditor angegeben werden, müssen mit denen im Skript übereinstimmen
- Für jede im BlueprintEditor zusammengebaute Kombination aus Signalschirmen und Zusatzanzeigern, muss ein passender Signalskript erstellt und im BlueprintEditor angegeben werden.

3.1. Allgemeines

Um für dieses und andere Signalsysteme möglichst wenig unterschiedliche 3D-Modelle erstellen zu müssen, sollten die Bezeichnungen der Licht-Nodes einem System folgen. Einmal vorgegeben, dürfen diese auch nicht mehr geändert werden.

Die in den Skripten stehenden Verweise auf die Module dürfen nicht geändert werden und passen für das entsprechende Signal oder Zusatzanzeiger bei Einhaltung aller Vorgaben.

3.1.1. Regeln für die Benennung der Licht-Nodes

Die Namen der Licht-Nodes werden vorgegeben, da diese Namen im Skript festgelegt sind.

Node-Name	Bezeichnung
Rot_1	Hauptrot
Rot_2	Ersatzrot
Gruen_1	Oben
Gelb_1	Oben
Gelb_2	Unten
Weiss_2	links Mitte/unten
Weiss_3	rechts Mitte
Weiss_42	Ra12 komplett (nur verwenden, wenn kein Ersatzsignal, oder Kennlicht verwendet werden soll)
Gruen_74	Grüner Lichtstreifen
Gelb_74	Gelber Lichtstreifen

In späteren Abschnitten folgt eine weitere Tabelle mit der Berechnung der notwendigen Code-Summe um dem Skript mitzuteilen, welche Licht-Nodes im Signal verbaut wurden.

3.1.2. Anmerkungen zu den Child-Namen

Für den 3D-Bau der Signalschirme werden außer den festgelegten Licht-Node-Namen keine weiteren Vorgaben benötigt. Erst die Verwendung des **BluePrintEditors** erfordert einige Einstellungen, von denen zum Beispiel die Child-Namen ganz wichtig für die Funktion der Signale sind. Diese Child-Namen sind entsprechend des Signaltyps unterschiedlich.

In den nachfolgenden Tabellen werden die jeweils zu verwendenden Child-Namen in der ersten Spalte aufgeführt. Bei diesen Child-Namen muss unbedingt auf Groß- und Kleinschreibung geachtet werden.

3.1.3. HP_CODE und VR_CODE

Die Vergangenheit hat gezeigt, dass es eine Unmenge an Sonderbauformen bei den Signalschirmen gibt. Gerade die Verwendung dieser Sonderbauformen macht eine Strecke so interessant. Leider war der Aufwand bei der Skriptgestaltung durch diese Sonderbauformen bisher sehr hoch.

Aus diesem Grunde habe ich für diese Skripte einen ganz neuen Lösungsansatz entwickelt:

- Es werden einheitliche Child-Namen auch bei unterschiedlichen Signalschirmen verwendet.
- Jede Signaloptik erhält einen eindeutigen Node-Namen, der auch bei unterschiedlichen Signalschirmen entsprechend der Funktion gleich lautet.
- Jedem dieser Node-Namen wird ein eindeutiger Zahlencode zugeordnet.
- Im Skript wird dann nur noch die Summe aller Zahlencodes je Signalschirm übermittelt und daraus alle gültigen Signaloptiken errechnet.
- Hierfür wurde im Skriptpaket eine Excel-Tabelle zur automatischen Berechnung der Code-Summe mitgeliefert. Auswahlen, die sich widersprechen erzeugen eine rote Einfärbung.

Wer kein Excel zur Verfügung hat, muss sich diese Code-Summe anhand der Tabelle manuell errechnen.

Berechnungstabelle für HL-Signale					
Node-Name	Bezeichnung	Vorsignale	Hauptsignale	Rangiersignale	Basis-Code
Rot_1	Hauptrot		x	x	2
Rot_2	Ersatzrot		x	x	4
Rot_3	Sperr-Rot (dual bei SH)		x	x	8
Gruen_1	Oben	x	x		16
Gelb_1	Oben	x	x		32
Gelb_2	Unten		x		128
Weiss_2	links Mitte/unten		x	x	512
Weiss_3	rechts Mitte		x	x	1024
Weiss_42	Ra12 komplett		x	x	2048
Gruen_74	Grüner Lichtstreifen		x		32768
Gelb_74	Gelber Lichtstreifen		x		65536
Code-Summe berechnet:					

Node-Namen und Basis-Codes, die in anderen Signalsystemen möglich sind, aber in dieser Tabelle nicht aufgeführt wurden, dürfen nicht verwendet werden!

3.1.4. Zusatzanzeiger

An einem Hauptsignal bzw Vorsignal können verschiedene Geschwindigkeitsanzeiger angefügt werden. Dies sind dann separate 3D-Modelle, die in der Signal-BIN über den Child-Namen angesprochen werden.

Alle möglichen Zusatzanzeiger sind im Skript als Kommentarzeile bereits aufgeführt und können durch entfernen der Kommentarstriche aktiviert werden. In den Skripten, in denen keine Zusatzanzeiger als aktivierbare Variable aufgeführt wurden, sind diese auch nicht möglich. In diesem Fall sind die Zusatzanzeiger als separates Signalobjekt zu erstellen.

3.2. Vorsignale

3.2.1. Vorsignal

Beschreibung	Child-Name
Vorsignal	VS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	VR_CODE
Gruen_1	oberes grünes Signallicht	HI1, HI4	16
Gelb_1	oberes gelbes Signallicht	HI10, HI7	32
Code-Summe:			48

3.2.2. Vorsignal mit nur einer gelben Optik

Beschreibung	Child-Name
Vorsignal mit einer gelben Optik	VS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	VR_CODE
Gelb_1	oberes gelbes Signallicht	HI10, HI7	32
Code-Summe:			32

3.2.3. Vorsignalwiederholer

Beschreibung	Child-Name
Vorsignal (ohne optionale PZB-Funktion)	VS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	VR_CODE
Gruen_1	oberes grünes Signallicht	HI1, HI4	16
Gelb_1	oberes gelbes Signallicht	HI10, HI7	32
Code-Summe:			48

3.3. Hauptsignale (einige Beispiele)

Auf Grund der Vielzahl an Möglichkeiten folgen nur einige Beispiele für die Hauptsignale. Es folgt nochmals der Hinweis, dass der Child-Name bei den Hauptsignalschirmen immer gleich ist und „HS“ lautet. Lediglich die Code-Summe wird für unterschiedliche Kombinationen an verwendeten Signaloptiken geändert.

In diesem Schema sind sämtliche sinnvolle Kombinationen von Signaloptiken realisierbar.

Obwohl bei den Hauptsignalen der HL-Signale der Hauptsignalschirm auch Vorsignalbegriffe signalisieren kann, gibt es nur einen Child-Namen („HS“). Die Information, ob auch Vorsignalbegriffe signalisiert werden können, wird durch das Vorhandensein der oberen gelben Optik automatisch erkannt.

Nur wenn keine obere gelbe Signaloptik (Gelb_1) vorhanden ist und trotzdem Hl4 o.ä. signalisiert werden soll, muss auch die Variable VR_SIGNAL_HEAD_NAME aktiviert werden.

Es wird lediglich ein HP_CODE im Hauptsignal übergeben. Der VR_CODE entfällt grundsätzlich bei allen Hauptsignalen. Egal, ob Vorsignalbegriffe signalisiert werden können oder nicht.

Da die Signalbegriffe immer aus einer Kombination von Haupt- und Vorsignalteil des Signalschirmes benannt werden, wurde in den Tabellen der Hauptsignale ggf. auf eine genaue Bezeichnung verzichtet.

3.3.1. Hauptsignal

Beschreibung	Child-Name
Hauptsignal komplett, aber ohne Lichtstreifen	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	Hl13, Zs1, Ra12	2
Rot_2	Ersatzrot	Keine Verwendung im regulären Betrieb	4
Gruen_1	oberes grünes Signallicht	Höchstgeschwindigkeit, 100 km/h erwarten	16
Gelb_1	oberes gelbes Signallicht	Halt erwarten, 40 km/h erwarten	32
Gelb_2	unteres gelbes Signallicht	40 km/h	128
Weiss_2	links Mitte/unten	Ra12, Zs1, Kennlicht	512
Weiss_3	rechts Mitte	Ra12	1024
Code-Summe:			1718

3.3.2. Hauptsignal als Vollausbau

Beschreibung	Child-Name
Hauptsignal als „Vollausbau“	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	Hl13, Zs1, Ra12	2
Rot_2	Ersatzrot	Keine Verwendung im regulären Betrieb, nur bei Signalstörung aktiv	4
Gruen_1	oberes grünes Signallicht	Höchstgeschwindigkeit, 100 km/h erwarten	16
Gelb_1	oberes gelbes Signallicht	Halt erwarten, 40 km/h erwarten	32
Gelb_2	unteres gelbes Signallicht	40 km/h	128
Weiss_2	links Mitte/unten	Ra12, Zs1, Kennlicht	512
Weiss_3	rechts Mitte	Ra12	1024
Gruen_74	Grüner Lichtstreifen	100 km/h	32768
Gelb_74	Gelber Lichtstreifen	40 km/h bzw. 60 km/h	65536
Code-Summe:			100022

3.3.3. Hauptsignal als Minimalvariante

Beschreibung	Child-Name
Hauptsignal mit Hauptrot und grüner Signaloptik	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	HI13	2
Gruen_1	oberes grünes Signallicht	HI1	16
Code-Summe:			18

Beschreibung	Child-Name
Hauptsignal mit Hauptrot, grüner Signaloptik und Ersatzsignal	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	HI13	2
Gruen_1	oberes grünes Signallicht	Höchstgeschwindigkeit, 100 km/h erwarten	16
Weiss_2	links Mitte/unten	Zs1, Kennlicht	512
Code-Summe:			530

Dies sollen nur einige Beispiele der verschiedenen Ausführungen darstellen. Dazwischen sind eigentlich alle sinnvollen Kombinationen möglich.

3.4. Rangiersignale

Bei der Deutschen Reichsbahn wurde im Rangierbereich überwiegend das Wartesignal mit Ra12 eingesetzt. Dieses Wartesignal besitzt keine roten Signaloptiken. Lediglich 2 weiße Signaloptiken wurden als Vorrückesignal genutzt.

3.4.1. Wartesignal / Rangiersignal / Ra12

Beschreibung	Child-Name
Sperrsignal	SH

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Weiss_42	zwei weiße Signallichter	Ra12	2048
Code-Summe:			2048

3.5. Zusatzsignale und Zusatzanzeiger

Sämtliche Zusatzsignale und Zusatzanzeiger schalten ausschließlich Texturen. Deshalb gibt es hier keine Licht-Nodes, sondern es muss lediglich auf die richtige Verwendung des Child-Namens geachtet werden.

Alle folgenden Zusatzanzeiger sind separate Signalobjekte als eigenes BluePrint und mit eigenen Links, die später in vorgesehener Weise in der Nähe der Hauptsignale platziert werden können.

3.5.1. Fahrtanzeiger

Child-Name	Beschreibung
FAZ	Fahrtanzeiger

Der Fahrtanzeiger besitzt in der einseitigen Ausführung nur einen Link und spricht über den Buchstaben „V“ die Textur an. Werden beim Fahrtanzeiger in der Signal-BIN 2 Links festgelegt, so wird für die Gegenrichtung die Textur über dem Buchstaben „R“ aktiviert.

Die Buchstaben sind im Skript festgelegt und müssen verwendet werden.



3.5.2. Abfahrtsignal

Child-Name	Beschreibung
ZP9 Signal Head	Fahrtanzeiger für DR

Die Textur für den Abfahrauftrag wird aus dem Signalskript in der Variable SIGNAL_TYP mit übergeben und kann frei gewählt werden.



3.5.3. Richtungsanzeiger

Child-Name	Beschreibung
ZS2v Signal Head	Licht - Richtungsvoranzeiger Zs2v
ZS2 Signal Head	Licht - Richtungsanzeiger Zs2

Die Texturen werden über die Eintragungen im Bereich PrimaryNamedTextureSet geladen. Die Buchstaben für die Texturen können frei gewählt werden.

Bei dem Zs2 muss grundsätzlich zwischen der Varianten mit einem Link und mehreren Links unterschieden werden:

- Besitzt das Zs2 nur einen Link, so wird der Skript „DEs HL Zs2.lua“ verwendet. Hierbei werden die Buchstaben bei den Links im Hauptsignal eingetragen. Und der Link 0 des Zs2 wird direkt vor den Link 0 des Hauptsignallink 0 gesetzt.
- Besitzt das Zs2 mehrere Links, so werden die Links 1+ als Ziellinks verwendet. In diesem Falle muss der Skript „DEs HL Zs2 T.lua“ verwendet werden. Die Buchstaben für die Richtungen werden nun direkt im Zs2 T eingetragen.

3.5.4. Gleiswechselanzeiger

Child-Name	Beschreibung
ZS6 Signal Head	Gleiswechselanzeiger Zs6

Die Texturen werden über die Eintragungen im Bereich PrimaryNamedTextureSet geladen. Die Buchstaben für die Texturen können frei gewählt werden.

Das Zs6 besitzt grundsätzlich mindestens 2 Links. Es können auch Varianten mit mehr als 2 Links erstellt werden. Die Links 1+ werden immer als Ziellinks verwendet. Der Skript „DEs HL Zs6.lua“ wird verwendet. Die Buchstaben für die Richtungen sind nicht vorgegeben und werden später direkt im Zs6 eingetragen.

4. Skripte

Für jede Kombination aus Signalschirmen und Zusatzanzeigern muss genau ein Skript im Verzeichnis Skripte in Ihrem Source-Verzeichnis existieren, der dann im BluePrintEditor angegeben wird.

Der Inhalt der Skripte folgt den speziellen Vorgaben, die weiter oben genannt wurden. So darf auch das unter „require“ zugeordnete Modul inklusive der Pfadangabe keinesfalls geändert werden.

Da lediglich **Beispielkombinationen** mitgeliefert wurden, müssen Sie Skripte für nicht vorhandene Kombinationen selbst erstellen. Es liegt nun an Ihnen, **nach persönlichem Geschmack Skriptnamen festzulegen**. Durch die Verwendung des BluePrint-Editors wird beim Export ein gleichnamiger Skript mit der Dateiendung „out“ im entsprechenden Ordner unter Assets erstellt. Dieser Skript benötigt dann die Module und die Optionsdatei im Unterordner bei Schuster/Freeware, damit das Signal funktionieren kann.

Ich empfehle im Dateinamen auch den HP_CODE zu verwenden. Auch wenn dieser kryptisch erscheint, so gibt er doch eindeutig Hinweis auf den Typ des verwendeten Signalschirms. Im Umkehrschluss haben Sie jedoch die freie Wahl bei der Benennung der Dateinamen für die Skripte in Ihrem Provider bzw. Produktordner.

Im Skript wird je nach Signalschirm der errechnete HP_CODE und / oder der VR_CODE hinter der entsprechenden Variablen angegeben.

Auf den nächsten Seiten folgt eine Tabelle von den im Skriptpaket mitgelieferten Skripten.

Bemerkungen:

Kasten	= Es wird ein 3D-Objekt ohne Licht-Nodes in Form eines Gehäuses benötigt
Anbau	= Spezielles Signal zur Realisierung eines Hauptsignals an einer DKW

Signaltyp	Skript-Name	Child-Name				Bemerkung
		Haupt-signal	Vor-signal	Zusatzanzeiger HS	Zusatzanzeiger VS	
Vorsignale						
Vorsignal (nur gelbe Optik)	DEs HL VS 32.lua		VS			
Vorsignal (nur gelbe Optik) als Wiederholer	DEs HL VS 32 Wh.lua		VS			
Vorsignal	DEs HL VS 48.lua		VS			
Vorsignal als Wiederholer	DEs HL VS 48 Wh.lua		VS			
Hauptsignale						
Hauptsignal mit Zs1, ohne Gelb 2 ohne Vorsignalfunktion , ohne Lichtstreifen	DEs HL HS 530.lua	HS				
Hauptsignal mit Ra12, ohne Gelb 2 ohne Vorsignalfunktion , ohne Lichtstreifen	DEs HL HS 1558.lua	HS				
Hauptsignal mit Ra12, ohne Vorsignalfunktion , ohne Lichtstreifen	DEs HL HS 1686.lua	HS				
Hauptsignal mit Vorsignalfunktion und Ra12, ohne Lichtstreifen	DEs HL MS 1718.lua	HS				
Hauptsignal mit Vorsignalfunktion und Ra12, grünem und gelbem Lichtstreifen	DEs HL MS 100022.lua	HS				
Rangiersignale						
Wartesignal (HP_Code je nach Licht-Nodes noch setzen!)	DEs HL SH.lua	SH				

Signaltyp	Skript-Name	Child-Name				Bemerkung
		Haupt-signal	Vor-signal	Zusatzanzeiger HS	Zusatzanzeiger VS	
Vorsignale						
Sonderbauformen						
Hauptsignal – Dummy	DEs HL HS Dummy.lua	HS				Kasten
Wartesignal als +Anbau für Form- und HL-Signale (HP_Code je nach Licht-Nodes noch setzen!)	DEs HL SH MainCo.lua	SH				
Vorsignal (nur gelbe Optik) als +Anbau für Formsignale	DEs HL VS 32 MainCo.lua		VS			
Vorsignal als +Anbau für Formsignale	DEs HL VS 48 MainCo.lua		VS			
Fahrtanzeiger (einfach oder doppelt)	DEs HL FAZ.lua			FAZ Signal Head		
Abfahrtsignal - Deutsche Reichsbahn	DEs HL Zp9 DR.lua			ZP9 Signal Head		
Richtungsanzeiger für Zielgleise (ab 1T = 2 Links)	DEs HL Zs2 T.lua			ZS2 Signal Head		
Richtungsanzeiger	DEs HL Zs2.lua			ZS2 Signal Head		
Richtungsvoranzeiger	DEs HL Zs2v.lua				ZS2v Signal Head	
Gleiswechselanzeiger (ab 1T = 2 Links)	DEs HL Zs6.lua			ZS6 Signal Head		

5. BluePrintEditor

Um das 3D-Modell mit dem Skript zu verbinden, ist der BluePrintEditor erforderlich. Hier werden einige Einstellungen für das Signal vorgenommen. Aus den erstellten Signalschirmen und Zusatzanzeigern kann nun unter Auswahl des richtigen Signalskriptes ein funktionierendes Signal erstellt werden.

Dieses Thema kann nur angerissen und nicht erschöpfend beschrieben werden. Deshalb gibt es hier nur einige Hinweise, die bei speziellen Bauarten von Signalschirmen auch noch abweichen können.

5.1. Grundsätzliches

Kenntnisse zur Benutzung des BluePrintEditors setze ich voraus. Ich werde nur auf die relevanten Einstellungen für die Funktion des 3D-Modells mit dem Skript eingehen.

5.2. Köpfe

Für alle Signalschirme und Zusatzanzeiger muss vorab ein AnimSceneryBluePrint erstellt werden.

Später werden diese Signalschirme und Zusatzanzeiger in einem SignalBluePrint zusammengefügt.

Die 3D-Modelle der Signalschirme bzw. Zusatzanzeiger-Objekte werden zuerst als ein separater „Kopf“ im BluePrintEditor definiert. Hierfür wird in der Regel das AnimSceneryBluePrint verwendet.

Im BluePrint wird im Bereich BrowseInformation unbedingt die **Category „eExcludeFromBrowserList“** angegeben, damit der Kopf später im Editor nicht auswählbar ist, sondern verborgen bleibt.

Ein Name muss bei allen BluePrints angegeben werden. Jedoch DisplayName und Beschreibung können entfallen.

Ein weiterer wichtiger Eintrag ist die **Geometrie** im Bereich **RenderComponent**, in der der Signalschirm bzw. der Zusatzanzeiger angegeben wird.

Lediglich bei den Form-Geschwindigkeitsanzeigern gibt es keinen separaten Kopf, damit zusätzlich Zahlen (im ID-Feld eingetragen) direkt im Schild angezeigt werden können.

5.3. Einstellungen für Signale und Zusatzanzeiger

Zuerst wird ein neues BluePrint geöffnet. Wir wählen ein „**Signal BluePrint**“ und nehmen sorgfältig alle notwendigen Eintragungen vor:

- ① **Name**
 - **Name**

Ein eindeutiger Name muss bei allen BluePrints angegeben werden. Dieser wird in der Tracks.bin vermerkt und erscheint auch bei Fehlermeldungen.

- ① **BrowseInformation**
 - **DisplayName**
 - **Description**

In dem Bereich DisplayName ist grundsätzlich die Sprache English und German mit einer passenden Bezeichnung auszufüllen. Die restlichen Sprachen werden wohl auf English verwiesen, wenn nichts ausgefüllt ist. Man sollte sich vorher ein gutes System überlegen, um die Signale später eindeutig identifizieren zu können.

Keinesfalls darf sich die Bezeichnung mit anderen Editorobjekten überschneiden, da bei gleichzeitiger Freischaltung ungewollte Ergebnisse entstehen.

Auch ein Eintrag im Feld Description kann später bei der Auswahl im Welteditor hilfreich sein, da diese Information als Tool-Tipp unterhalb des Mauszeigers angezeigt wird.

① **BrowseInformation**

- **Category**

Der Eintrag Category legt fest, in welcher Gruppe später im Welteditor das Signal gelistet werden soll. Wir verwenden für alle Signale die Auswahl: eTrackInfrastruktur

① **_object**

- **PrimaryNamedTextureSet**
- **SecondaryTextureSet**

Im Bereich PrimaryNamedTextureSet und SecondaryTextureSet werden die Textursets für die erste und zweite Zeile der Mastnummer, soweit vorhanden, angegeben. Ist keine Mastnummer vorhanden, können diese Angaben entfallen. Auch wenn diese Bereiche frei bleiben, können später die im ID-Feld eingetragenen Bezeichnungen zum Debug verwendet werden.

Bei allen **Zusatzanzeigern** wird im PrimaryNamedTextureSet das zu verwendende Texturset für die darzustellenden Zeichen angegeben.

① **_object**

- **GeometryID**

Im Eintrag „GeometryID“ wird die Mastgeometrie angegeben. Hierbei handelt es sich **nicht** um die Signalschirme. Diese werden später im Bereich _container als Children definiert.

① **_signal**

- **NumberOfTrackLinks**

Im Bereich **_signal** werden nun weitere Signaleigenschaften definiert. Dort findet sich als erster Eintrag NumberOfTrackLinks, welcher die Anzahl der Ziellinks zuzüglich des Link 0 beinhaltet. Für ein Blocksignal (OT), alle Vorsignale und bei allen Zusatzanzeigern bis auf die „T“-Varianten für Zs2 und Zs6 wird hier der Wert 1 eingetragen. Bei allen Hauptsignalen, die Fahrstraßen über Weichen haben, zählen wir für jedes Ziel den Wert 1 hinzu. Liegt hinter dem Hauptsignal zum Beispiel eine Weiche, dann wird der Wert 3 in das Feld NumberOfTrackLinks eingetragen.

① **_signal**

- **Stopping**

In das Kästchen bei Stopping wird bei allen Hauptsignalen und Rangiersignalen ein Häkchen gesetzt. Hierdurch wird später ein neuer Blockabschnitt, beginnend am Link 0, gebildet.

Dies ist die einzige Eigenschaft, die KI-Züge vom Signal kennen und darauf auch reagieren.

① **_signal**

- **JunctionSignalType**

Bei der Auswahlliste JunctionSignalType sollte bei allen Signalen der Wert **eJunctionTypSpeed** ausgewählt werden, um später im Signal-Flyout die Eingabemöglichkeit der Richtungsangabe und der Geschwindigkeit zu haben.

① **_signal**

- **ControlMode**

Die Auswahl im Bereich ControlMode ist für die normale Funktion weniger von Bedeutung. Wird jedoch beabsichtigt, das Signalsystem gemeinsam mit der von mir erstellen 2DMapPro zu nutzen, dann sollte folgende Einstellung vorgenommen werden:

Hauptsignale ohne Vorsignal:	eControlModeControlled
Hauptsignale mit Vorsignal:	eControlModeControlledCallOn
Alle anderen Signale:	eControlModeAutomatic

Nur mit diesen Einstellungen werden dann die passenden Mastsymbole in der 2DMapPro angezeigt.

① **_script**

Nun kommt im Bereich **_script** die wichtige Entscheidung, welcher Skript zu verwenden ist. Wie schon weiter oben geschrieben, gibt es in der Regel für eine bestimmte Kombination aus Signalschirmen immer nur einen einzigen Skript, der zu verwenden ist. Schauen Sie also genau in der unter Punkt 4. aufgeführten Tabelle nach, welcher Skript zu ihrem Signal passt. Ist dieser nicht vorhanden, dann kopieren Sie am besten einen vorhandenen Skript und passen dessen Einstellungen durch Aktivierung von Variablen an und tragen den jeweils gültigen CODE für den oder die Signalschirme ein.

Der dort einzutragende Skript wird mit dem Verzeichnisnamen oberhalb von Provider- Produktnamen und ohne Dateierweiterung angegeben.

Ändern Sie keinesfalls den Verweis auf das dazugehörige Modul!

Beispiel für ein einfaches Hauptsignal: RailNetwork\Signals\German HL\Skripte\DEs HL HS

Der Skript muss als Datei im Verzeichnis Source existieren, sonst schlägt der Export fehl. Im oben genannten Beispiel also genau diese Datei:

Source\[Provider]\[Produkt]\RailNetwork\Signals\German HL\Skripte\DEs HL HS.lua

Die Werte für [Provider] und [Produkt] entsprechen Ihren Verzeichnisnamen.

① **_container**

○ **Children**

Erst jetzt, im Bereich **_container** -> Children werden die Einstellungen, die die einzelnen Signalschirme und Zusatzanzeiger betreffen, festgelegt.

Für jeden Signalschirm bzw. Zusatzanzeiger muss ein Child hinzugefügt werden (Add Element...).

Der Übersicht halber beginnen wir immer mit dem Hauptsignal. Als 2. Child folgen nun die Zusatzanzeiger (wenn vorhanden).

Im Feld ChildName wird der ChildName aus der Tabelle (Punkt 3.3) für den entsprechenden Signalschirm eingetragen. Hierbei ist auf die korrekte Schreibweise zu achten.

Es kann immer nur ein Zs3 und/oder Zs3v Zusatzanzeiger hinzugefügt werden. Die gleichzeitige Verwendung von **Licht- und Form-** Geschwindigkeits- bzw. voranzeigern ist nicht möglich.

① **_container**

○ **Children**

▪ **BlueprintID**

An dieser Stelle wird die XML-Datei vom Signalschirm eingetragen.

Die Einstellungen hierzu sind bei den Signalschirmen sehr unterschiedlich und werden deshalb nicht näher aufgeführt. Anzumerken ist jedoch, dass bei den Signalschirmen die Category: **eExcludeFromBrowserList** auszuwählen ist und in diesem Blueprint kein Skript angegeben wird.

① **_container**

○ **Children**

▪ **Matrix**

Die Matrix ermöglicht es, den Signalschirm am Mast entsprechend zu positionieren.

5.4. Export des BluePrints

Im unteren Bereich sollte man sich unbedingt den Bereich Output aufklappen [+], damit man die Meldungen verfolgen kann.

Wenn alles richtig eingegeben wurde, wird das BluePrint mit der Funktionstaste F7 nach Assets exportiert. Nur wenn am Ende des Protokolls „Export Succeded“ steht, ist der Export erfolgreich abgeschlossen. Anderen falls muss man die Einstellungen korrigieren.

Anschließend sollte das Signal im Welteditor des Train Simulators zur Verfügung stehen.

5.5. Einsatz der Signale in einer Train Simulator Strecke

Wenn Sie das Signal in eine Strecke einbauen, muss das Signal nach dem Neustart der Strecke das richtige Signalbild, zum Beispiel Hp0 oder Vr0 anzeigen. Leuchten auf dem Signalschirm dann noch alle Signallichter, so gibt es ein Problem, welches an falschen Einstellungen liegen kann.

Prüfen Sie dann folgende Dinge:

- Starten Sie LogMate mit dem TS und kontrollieren Sie, ob dort Fehler angezeigt werden
- Passt der Skript auch zu den verwendeten Signalschirmen, die im BluePrint verwendet wurden?
- Sind alle Child-Namen richtig geschrieben?
- Wurde die richtige Code-Summe berechnet und im Skript eingetragen?
- Prüfen Sie das 3D-Modell, ob die Licht-Node-Namen stimmen

Ich hoffe, dass ich das Thema hinreichend ausführlich und richtig beschrieben habe.
Sollte es Probleme geben, bitte ich darum, im Forum bei einem geeigneten Thema nachzufragen.
So können Fragen schnell geklärt werden und die Lösungen stehen Vielen zur Verfügung.
Danke.

Ich wünsche viel Erfolg beim Einsatz der Skripte
Mathias Gundlach
