

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



Basierend auf den Nachtzeichen der Formsignale führte die Deutsche Bundesbahn nach dem Zweiten Weltkrieg die H/V-Lichtsignale (Haupt-/ Vorsignal-System) ein.

Vom Signal „Zughalt und Rangierverbot“ (Hp00) abgesehen unterschied sich die Zahl der Signalbilder nicht von denen der Formsignale.

Quelle: Wikipedia

Die HV-Signale haben eine mehrfache äußerliche Entwicklung hinter sich. So gibt es Bauformen von 1948, 1951, 1969 und Kompaktsignale.

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

Dieses Paket beinhaltet die Anleitung und Skripte, um selbst erstellte HV-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_HV-Signale_V6.11.zip

- Anleitung Skriptpaket_HV-Signale_V6.11.pdf
- Excel-Tabelle Code-Berechnung_HV-Signale.xlsx
- Skripte für Source Skripte_HV-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 HV-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 ohne Zusatzlicht	6
3.2.2. Vorsignal im verkürzten Bremswegabstand	6
3.2.3. Vorsignalwiederholer	6
3.3. Hauptsignale (einige Beispiele)	7
3.3.1. Hauptsignal	7
3.3.2. Hauptsignal mit Kennlicht	8
3.3.3. Hauptsignal (Bauform 1948)	8
3.3.4. Hauptsperrsignal	9
3.3.5. Hauptsperrsignal mit Kennlicht und Vorsichtssignal	9
3.3.6. Hauptsperrsignal als Halbregelabstandssignal	10
3.3.7. Blocksignal	10
3.3.8. Blocksignal mit Kennlicht im Halbregelabstand	11
3.4. Sperrsignale	11
3.4.1. Sperrsignal ohne Deckungssignalfunktion	11
3.4.2. Sperrsignal und Deckungssignal	11
3.5. Zusatzsignale und Zusatzanzeiger	12
3.5.1. Fahrtanzeiger	12
3.5.2. Abfahrtsignal	12
3.5.3. Richtungsanzeiger	12
3.5.4. Gleiswechselanzeiger	13
3.5.5. Geschwindigkeitsanzeiger	13
4. Skripte	13
5. BluePrintEditor	16
5.1. Grundsätzliches	16
5.2. Köpfe	16
5.3. Einstellungen für Signale und Zusatzanzeiger	16
5.4. Export des BluePrints	19
5.5. Einsatz der Signale in einer Train Simulator Strecke	19

1. Einleitung

1.1. Einige Vorworte

Wer sich mit dem Bau von HV-Signalen beschäftigt, und diese Skripte verwenden möchte, sollte sich grundsätzlich mit dem Signalpaket Deutsche_HV-Signale vom SignalTeam auseinandersetzen, um die Sonderfunktionen des Signalsystems zu verstehen.

Bezüglich der Sonderfunktionen im Zusammenhang mit Triggern und Einstellungen im Signal-Flyout unterscheiden sich diese Skripte nicht von denen, die unter „**Schuster/SignalTeam**“ veröffentlicht wurden. Hierbei ist jedoch der jeweilige Versionsstand zu beachten.

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 Signalpaketen, 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 Signalpaketen 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 Signaloptik verbleiben nun standardmäßig auf Hp0
- Aktivierung des Ersatzrot mit Hp0-Trigger (ZE)
- 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

- Korrekturen in der Anleitung
- Automatische Debugmeldungen in LogMate, bei nicht korrekter Code-Summe

Version 6.0

- Erste veröffentlichte Version des Skriptpaketes

2. Ordnerstruktur

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

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 HV“ 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 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
Rot_3	Sperr-Rot (dual bei SH)
Gruen_1	Oben
Gelb_1	Oben
Gruen_2	Mitte oder Unten
Gelb_2	Unten
Weiss_2	links Mitte/unten
Weiss_3	rechts Mitte
Weiss_42	Sh1 komplett
Weiss_9	Kennlicht / Zusatzlicht
Weiss_63	Ersatzsignal (A)
Gelb_63	Vorsichtssignal (V)

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. Auswählen, 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 HV-Signale					
Node-Name	Bezeichnung	Vorsignale	Hauptsignale	Sperrsignale	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			32
Gruen_2	Mitte oder Unten	x	x		64
Gelb_2	Unten	x	x		128
Weiss_2	links Mitte/unten		x	x	512
Weiss_3	rechts Mitte		x	x	1024
Weiss_42	Sh1 komplett		x	x	2048
Weiss_9	Kennlicht / Zusatzlicht	x	x		4096
Weiss_63	Ersatzsignal (A)		x		8192
Gelb_63	Vorsichtssignal (V)		x		16384
	Halbregelabstand		x		262144
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 ohne Zusatzlicht

Beschreibung	Child-Name
Vorsignal ohne Zusatzlicht	VS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	VR_CODE
Gruen_1	oberes grünes Signallicht	Vr1, Vr2	16
Gelb_1	oberes gelbes Signallicht	Vr0	32
Gruen_2	unteres grünes Signallicht	Vr1	64
Gelb_2	unteres gelbes Signallicht	Vr0, Vr2	128
Code-Summe:			240

3.2.2. Vorsignal im verkürzten Bremswegabstand

Beschreibung	Child-Name
Vorsignal mit Zusatzlicht	VS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	VR_CODE
Gruen_1	oberes grünes Signallicht	Vr1, Vr2	16
Gelb_1	oberes gelbes Signallicht	Vr0	32
Gruen_2	unteres grünes Signallicht	Vr1	64
Gelb_2	unteres gelbes Signallicht	Vr0, Vr2	128
Weiss_9	weißes Zusatzlicht	Vr0, Vr1,Vr2, deaktiviert	4096
Code-Summe:			4336

3.2.3. Vorsignalwiederholer

Beschreibung	Child-Name
Vorsignal mit Zusatzlicht	VS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	VR_CODE
Gruen_1	oberes grünes Signallicht	Vr1, Vr2	16
Gelb_1	oberes gelbes Signallicht	Vr0	32
Gruen_2	unteres grünes Signallicht	Vr1	64
Gelb_2	unteres gelbes Signallicht	Vr0, Vr2	128
Weiss_9	weißes Zusatzlicht	Vr0, Vr1,Vr2, deaktiviert	4096
Code-Summe:			4336

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.

Durch die Verwendung unterschiedlicher Signaloptiken in Kombination sind selbstverständlich auch Blocksignale und Hauptsperrsignale in verschiedensten Ausführungen möglich.

Weiterhin ist es auch möglich, gleichzeitig ein Ersatzsignal und ein Vorsichtssignal zu verbauen.

Alle Hauptsignale können mit den zuvor genannten Vorsignalen an einem Mast kombiniert werden. Im Skript wird dann zusätzlich zum HP_CODE auch der VR_CODE angegeben. Natürlich muss außerdem der VR_SIGNAL_HEAD_NAME aktiviert werden.

3.3.1. Hauptsignal

Beschreibung	Child-Name
Hauptsignal mit Ersatzsignal ohne Kennlicht, ohne Sh1	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	Hp0, Zs1, Zs8	2
Rot_2	Ersatzrot	Keine Verwendung im regulären Betrieb	4
Gruen_1	oberes grünes Signallicht	Hp1, Hp2	16
Gelb_2	unteres gelbes Signallicht	Hp2	128
Weiss_63	3 weiße Signallichter in A-Form	Ersatzsignal, Zs8	8192
Code-Summe:			8342

Beschreibung	Child-Name
Hauptsignal mit Ersatzsignal ohne Kennlicht, ohne Sh1, ohne Ersatzrot	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	Hp0, Zs1, Zs8	2
Gruen_1	oberes grünes Signallicht	Hp1, Hp2	16
Gelb_2	unteres gelbes Signallicht	Hp2	128
Weiss_63	3 weiße Signallichter in A-Form	Ersatzsignal, Zs8	8192
Code-Summe:			8338

3.3.2. Hauptsignal mit Kennlicht

Beschreibung	Child-Name
Hauptsignal mit Ersatzsignal mit Kennlicht, ohne Sh1	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	Hp0, Zs1, Zs8	2
Rot_2	Ersatzrot	Keine Verwendung im regulären Betrieb	4
Gruen_1	oberes grünes Signallicht	Vr1, Vr2	16
Gelb_2	unteres gelbes Signallicht	Vr0, Vr2	128
Weiss_9	weißes Kennlicht	Leuchtet wenn Signal deaktiviert oder bei Verwendung als Halbregelabstandssignal	4096
Weiss_63	3 weiße Signallichter in A-Form	Ersatzsignal, Zs8	8192
Code-Summe:			12438

3.3.3. Hauptsignal (Bauform 1948)

Beschreibung	Child-Name
Hauptsignal mit Ersatzsignal ohne Kennlicht, ohne Sh1, mit zweiter grüner Optik	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	Hp0, Zs1, Zs8	2
Rot_2	Ersatzrot	Keine Verwendung im regulären Betrieb	4
Gruen_1	oberes grünes Signallicht	Hp2	16
Gruen_2	mittleres grünes Signallicht	Hp1	64
Gelb_2	unteres gelbes Signallicht	Hp2	128
Weiss_63	3 weiße Signallichter in A-Form	Ersatzsignal, Zs8	8192
Code-Summe:			8406

3.3.4. Hauptsperrsignal

Beschreibung	Child-Name
Hauptsperrsignal mit Ersatzsignal ohne Kennlicht, mit Sh1	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	Hp0, Zs1, Zs8	2
Rot_3	Sperrrot	Hp0, Bei Sh1 dunkel	8
Gruen_1	oberes grünes Signallicht	Hp1, Hp2	16
Gelb_2	unteres gelbes Signallicht	Hp2	128
Weiss_42	2 weiße Signallichter für Sh1	Sh1	2048
Weiss_63	3 weiße Signallichter in A-Form	Ersatzsignal, Zs8	8192
Code-Summe:			10394

Rein technisch gesehen ist es auch möglich, anstelle des **Weiss_42** zwei einzeln zu schaltende weiße Licht-Nodes **Weiss_2** und **Weiss_3** einzubauen.
Die Code-Summe muss dann entsprechend angepasst werden.

3.3.5. Hauptsperrsignal mit Kennlicht und Vorsichtssignal

Beschreibung	Child-Name
Hauptsperrsignal mit Sh1, Kennlicht und Vorsichtssignal	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	Hp0, Zs1, Zs8	2
Rot_3	Sperrrot	Hp0, Bei Sh1 dunkel	8
Gruen_1	oberes grünes Signallicht	Hp1, Hp2	16
Gelb_2	unteres gelbes Signallicht	Hp2	128
Weiss_42	2 weiße Signallichter für Sh1	Sh1	2048
Weiss_9	weißes Kennlicht	Leuchtet wenn Signal deaktiviert oder bei Verwendung als HalbregeLabstandssignal	4096
Gelb_63	3 gelbe Signallichter in V-Form	Vorsichtssignal	16384
Code-Summe:			22682

3.3.6. Hauptsperrsignal als HalbregeLabstandsignal

Beschreibung	Child-Name
Hauptsperrsignal mit Ersatzsignal mit Kennlicht, mit Sh1 im HalbregeLabstand	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	Hp0, Zs1, Zs8	2
Rot_3	Sperrrot	Hp0, Bei Sh1 dunkel	8
Gruen_1	oberes grünes Signallicht	Hp1, Hp2	16
Gelb_2	unteres gelbes Signallicht	Hp2	128
Weiss_42	2 weiße Signallichter für Sh1	Sh1	2048
Weiss_9	weißes Kennlicht	Leuchtet wenn Signal deaktiviert oder bei Verwendung als HalbregeLabstandssignal	4096
Weiss_63	3 weiße Signallichter in A-Form	Ersatzsignal, Zs8	8192
	HalbregeLabstand		262144
Code-Summe:			276634

3.3.7. Blocksignal

Beschreibung	Child-Name
Block mit Ersatzsignal ohne Kennlicht, ohne Sh1	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	Hp0, Zs1, Zs8	2
Gruen_2	unteres grünes Signallicht	Hp1	64
Weiss_63	3 weiße Signallichter in A-Form	Ersatzsignal, Zs8	8192
Code-Summe:			8258

Ein Blocksignal kann auch mit der Licht-Node Gruen_1 gebaut werden. Dann ist jedoch die Code-Summe entsprechend anzupassen.

3.3.8. Blocksignal mit Kennlicht im HalbregeLabstand

Beschreibung	Child-Name
Block mit Ersatzsignal ohne Kennlicht, ohne Sh1 im HalbregeLabstand	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	Hauptrot	Hp0, Zs1, Zs8	2
Gruen_2	unteres grünes Signallicht	Hp1	64
Weiss_9	weißes Kennlicht	Leuchtet wenn Signal deaktiviert oder bei Verwendung als HalbregeLabstandssignal	4096
Weiss_63	3 weiße Signallichter in A-Form	Ersatzsignal, Zs8	8192
	HalbregeLabstand		262144
Code-Summe:			274498

3.4. Sperrsignale

Bei den Sperrsignalen können wahlweise Rot_1 und / oder Rot_3 verwendet werden. Ebenso kann bei Bedarf das Ersatzrot Rot_2 mit eingebaut werden, welches aber im regulären Betrieb nie leuchtet, sondern nur per Hp0-Trigger bei einer manuellen Störung angesteuert werden kann.

3.4.1. Sperrsignal ohne Deckungssignalfunktion

Beschreibung	Child-Name
Sperrsignal	SH

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_3	1 oder 2 rote Signallichter	Hp0	8
Weiss_42	2 weiße Signallichter	Sh1	2048
Code-Summe:			2056

3.4.2. Sperrsignal und Deckungssignal

Beschreibung	Child-Name
Sperrsignal, auch als Deckungssignal einsetzbar	SH

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_3	1 oder 2 rote Signallichter	Hp0	8
Weiss_2	linkes weißes Signallicht	Sh1	512
Weiss_3	rechtes weißes Signallicht	Sh1, Leuchtet bei Funktion als Deckungssignal	1024
Code-Summe:			1544

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 und DB

Die Textur für den Abfahrtauftrag 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 HV 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 HV 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.

3.5.5. Geschwindigkeitsanzeiger

Child-Name	Beschreibung
Zs3v Signal Head	Licht - Geschwindigkeitsvoranzeiger Zs3v
Zs3 Signal Head	Licht - Geschwindigkeitsanzeiger Zs3
Zs3v Form Head	Form - Geschwindigkeitsvoranzeiger Zs3v
Zs3 Form Head	Form - Geschwindigkeitsanzeiger Zs3

Die Texturen werden über die Eintragungen im Bereich PrimaryNamedTextureSet geladen.

Geschwindigkeitsanzeiger und Geschwindigkeitsvoranzeiger können in ein Signal-Blueprint von einem Hauptsignal integriert werden. Hierzu muss im Blueprint das Child mit eingetragen und im Skript die Variable des Child-Namens aktiviert werden.

Form – Geschwindigkeitsanzeiger bzw. Form – Geschwindigkeitsvoranzeiger besitzen keinen separaten Kopf. Die Texturen für die Zahlen werden dennoch im Bereich PrimaryNamedTextureSet vereinbart.

4. Skripte

Für jede Kombination aus Signalschirmen und Zusatzanzeigern muss genau ein Skript im Verzeichnis Skripte unter Ihrem Source-Verzeichnis existieren.

Es wurden ja nur Beispielskombinationen mitgeliefert. Nun liegt es an Ihnen, nach persönlichem Geschmack Skriptnamen festzulegen. Durch die Verwendung des Blueprint-Editors wird beim Export ein gleichnamiger Skript mit der Dateierweiterung „.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.

Es können also auch die mitgelieferten Skriptnamen geändert werden. Lediglich 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.

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

Auf den nächsten Seiten folgen Tabellen mit den im Skriptpaket vorhandenen Skripten.

Signaltyp	Skript-Name	Child-Name				Bemerkung
		Haupt-signal	Vor-signal	Zusatzanzeiger HS	Zusatzanzeiger VS	
Vorsignale						
Vorsignal	DEs HV VS.lua		VS			
Vorsignal im reduzierten Abstand	DEs HV VS rd.lua		VS			
Vorsignal als Wiederholer	DEs HV VS Wh.lua		VS			
Hauptsignale						
Hauptsignal	DEs HV HS.lua	HS				
Hauptsignal mit Vorsignal	DEs HV HS VS.lua	HS	VS			
Hauptsignal mit Vorsignal im reduzierten Abstand	DEs HV HS VS rd.lua	HS	VS			
Sperrsignale						
Sperrsignal	DEs HV SH.lua	SH				
Sperrsignal als Deckungssignal	DEs HV SP.lua	SP				
Sonderbauformen						
Hauptsignal – Dummy	DEs HV HS Dummy.lua					Kasten
Hauptsignal als Anbauvariante (speziell für DKW-Bereiche)	DEs HV Anbau HS.lua	HS				Anbau
Sperrsignal als +Anbau für Form- und HV-Signale	DEs HV SH MainCo.lua	SH				
Vorsignal als +Anbau für Formsignale	DEs HV VS MainCo.lua		VS			
Vorsignal im reduzierten Abstand als +Anbau für Formsignale	DEs HV VS rd MainCo.lua		VS			

Signaltyp	Skript-Name	Child-Name				Bemerkung
		Haupt-signal	Vor-signal	Zusatzanzeiger HS	Zusatzanzeiger VS	
Zusatzsignale						
Fahrtanzeiger (einfach oder doppelt)	DEs HV FAZ.lua			FAZ Signal Head		
Abfahrtsignal - Deutsche Bundesbahn	DEs HV Zp9 DB.lua			ZP9 Signal Head		
Abfahrtsignal - Deutsche Reichsbahn	DEs HV Zp9 DR.lua			ZP9 Signal Head		
Richtungsanzeiger für Zielgleise (ab 1T = 2 Links)	DEs HV Zs2 T.lua			ZS2 Signal Head		
Richtungsanzeiger	DEs HV Zs2.lua			ZS2 Signal Head		
Richtungsvoranzeiger	DEs HV Zs2v.lua				ZS2v Signal Head	
Kurze Fahrt Trigger	DEs HV Zs3 KF Tr.lua					Kasten
Geschwindigkeitsanzeiger für Kurze Fahrt	DEs HV Zs3 KF.lua			ZS3 Signal Head		
Geschwindigkeitsanzeiger	DEs HV Zs3.lua			ZS3 Signal Head		
Geschwindigkeitsanzeiger fest	DEs HV Zs3Fo.lua			ZS3 Form Head		
Geschwindigkeitsvoranzeiger	DEs HV Zs3v.lua				ZS3v Signal Head	
Geschwindigkeitsvoranzeiger fest	DEs HV Zs3vFo.lua				ZS3v Form Head	
Gleiswechselanzeiger (ab 1T = 2 Links)	DEs HV Zs6.lua			ZS6 Signal Head		

Bemerkungen:

Kasten = Es wird ein 3D-Objekt ohne Licht-Nodes in Form eines Gehäuse benötigt

Anbau = Spezielles Signal zur Realisierung eines Hauptsignals an einer DKW

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 Sperrsignalen 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 Dateiendung angegeben.

Ändern Sie keinesfalls den Verweis auf das dazugehörige Modul!

Beispiel für ein einfaches Hauptsignal: `RailNetwork\Signals\German HV\Skripte\DEs HV 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 HV\Skripte\DEs HV 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 dann das Vorseignal (wenn vorhanden) und zum Schluss 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 Succeeded“ 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
