



Mit der Veröffentlichung der Plugin's Eins und Zwei zu EEP 10.2 wurden dem Anwender zwei Tools an die Hand gegeben, die ungeahnte Möglichkeiten der Steuerung und Beeinflussung in EEP eröffnen. In Verbindung mit den verfügbaren Eigenschaften aus den Vorgängerversionen ergibt sich ein weites Betätigungsfeld für die Automatisierung und Animation von EEP-Szenarien.

Die genannten Veränderungen waren ausschlaggebend für die Realisierung dieses Projekts. Es liefert eine Basis, mit der die Erstellung eines Drucktasten-Stellwerks möglich wird. Wichtig ist, darauf hinzuweisen, dass es sich nicht um eine "Zweiklick-Lösung" handelt. Das Pult wird aus einzelnen Elementen erstellt und über Dialoge und Verknüpfungen "zum Leben erweckt". Prinzipiell kann man die Vorgehensweise mit einer realen Modellbahn vergleichen, an der ein Gleisbildstellpult nachgerüstet wird. Dementsprechend ist eine gewisse Sorgfalt beim Aufbau Voraussetzung.

Um in bestimmten Fällen eine sinnvolle Funktionalität sicherzustellen, war die Verwendung von LUA-Skripten und das Plugin 1 unabdingbar. *Dieser Umstand setzt für die effektive Nutzung dieses Baukastens mindestens die EEP-Version 10.2 inkl. der Plugin's 1 und 2 voraus.*

### Die Basis

Dieses Basisset ist der Einstieg in den Bau eines Drucktasten-Stellwerks. Es enthält alle wichtigen Tischfelder, Taster und Signalpiktogramme. Ein weiterer Bestandteil sind die benötigten LUA-Dateien und eine Demoanlage, auf der die Installation eines Pultes erkundet werden kann.

Der Funktionsumfang des Basissets ermöglicht die Anzeige der Besetztmeldung bei Gleismeldern, die Lageanzeige und das Stellen von Weichen und Signalen sowie eine Signalverkettung von Einfahrt-, Ausfahrt- und erstem Blocksignal mit der entsprechenden Haupt-Vorsignalanzeige auf dem Stellpult. Bei Bedarf ist es möglich, eine "echte" Besetztmeldung aufzubauen, die Gleisabschnitte auch auf einzelne, stehengelassene Fahrzeuge überprüft.

Da der Umgang mit der Skriptsprache LUA nicht jedem "in die Wiege gelegt" wurde, beschränkt sich dieses Projekt auf die notwendigsten Eingaben durch den User. Diese können auch durch nicht so versierte Anwender in Sachen LUA getätigt werden. Darüber hinaus wird an den entsprechenden Stellen dieses Handbuchs detailliert auf die nötigen Verfahren eingegangen.

Aufgrund EEP-spezifischer Umstände mussten bei einigen Funktionen gegenüber dem Vorbild Kompromisse eingegangen werden. Ist dies der Fall, wird an den entsprechenden Stellen der Beschreibung darauf hingewiesen.

### Aussichten

Für die noch folgenden Ergänzungen ist unter anderem eine Fahrstraßen-Modul geplant. Die Gleis- und Weichenmelder des Basissets sind schon dafür vorbereitet, so dass bei Erscheinen der Erweiterung lediglich die entsprechenden LUA-Dateien nachgeladen oder vervollständigt und die zusätzlich benötigten Tischelemente bereitgestellt werden.

### Credits

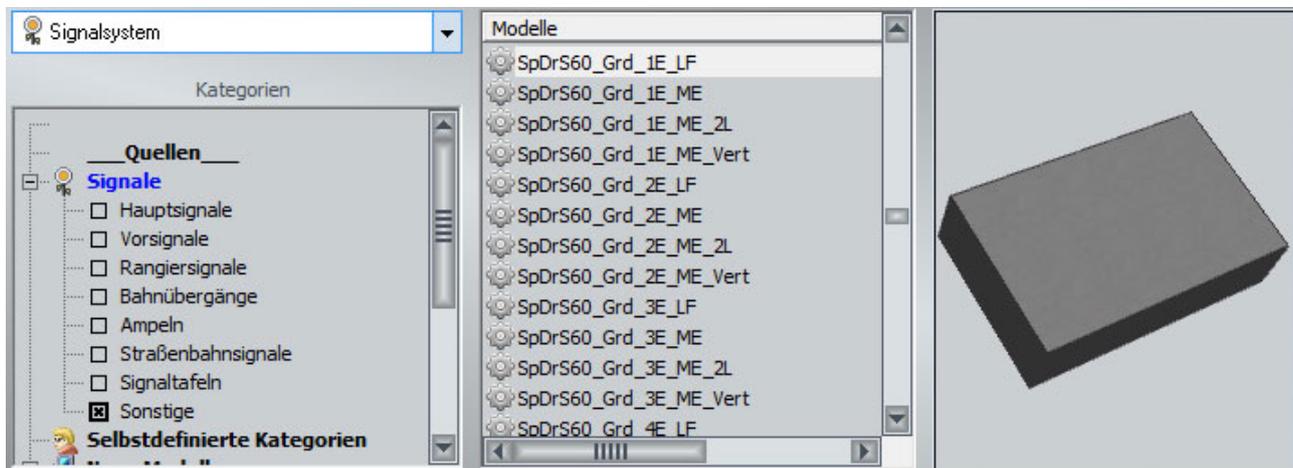
An dieser Stelle möchte ich allen, die mich bei diesem Projekt unterstützt haben, recht herzlich danken.

Mein besonderer Dank gilt Andreas Hempel (AH1) für die Tipps und die Bereitstellung von einigen Lichtsignalen und Tom Schulz (DR\_Bahner) für die Erstellung der "Spielanlage", die als Demo im Set enthalten ist.

## Erste Schritte

### Die Dateinamen

Nach der Installation des Sets sind alle Tischelemente unter "Signale - Sonstige" zu finden. Die Bezeichnung "SpDrS60" am Anfang der Dateinamen sorgt dafür, dass alle Elemente in einem zusammenhängenden Bereich zu finden sind.



Der nächste Bezeichner im Dateinamen gliedert die Objekte in Kategorien (siehe Abb. unter "Modelle").

Folgende Kategorien sind aktuell vorhanden:

SpDrS60_Grd_xxx	Tischfelder gerade, Länge 1 - 8 Einheiten
SpDrS60_Sig_xxx	Signalsymbole
SpDrS60_Strk_xxx	Tischfelder Strecke
SpDrS60_Tstr_xxx	Taster alle Typen
SpDrS60_Wche_xxx	Tischfelder Weichen, DKWs, EKWs, Gleissperren

Eine detaillierte Übersicht aller Objekte finden Sie unter "Das Inventar".

### Das Raster

Um unsere Tischfelder in EEP verwenden zu können, benötigen wir natürlich etwas, auf das wir sie aufsetzen können. Wir wählen dazu das "unsichtbare Gleis". Damit die Elemente nahtlos nebeneinander liegen, muss das Gleis in einem *Parallelabstand von 2.5 m* verlegt werden. Die Höhe belassen wir auf Standardwert.



Es empfiehlt sich, die verlegten Gleise in ihrer ursprünglichen Länge zu belassen und die erforderliche Anzahl durch das Andocken weiterer Gleisstücke herzustellen. Über den Dialog "Gleise vervielfältigen" lässt sich das komfortabel erledigen. Dabei kann die erforderliche Länge auch um ein oder zwei Gleislängen überschritten werden. Diese zusätzliche Länge kann beim Bau des Pultes als "Einfädel- und Ausrichtezone" genutzt und nach Fertigstellung wieder entfernt werden.

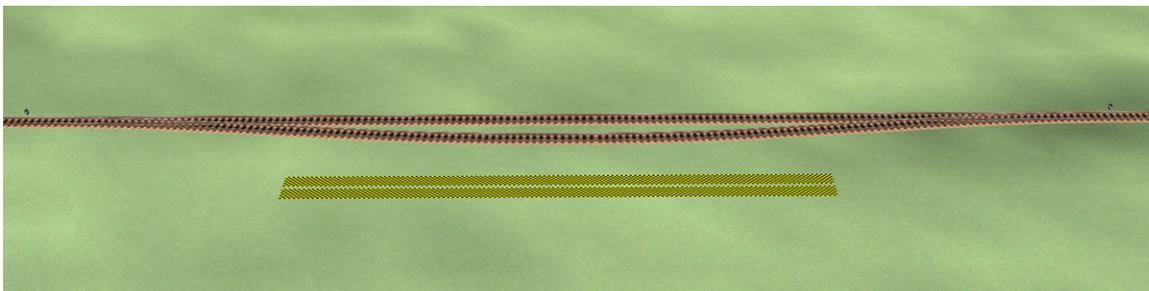
*Der Bau eines Stellpultes sollte so weit wie möglich im "3D-Objekt Editiermodus" erfolgen! Das gewährleistet die max. Kontrolle beim Zusammenstellen und dem Positionieren der Objekte .*

## Erste Schritte

---

### Ein Beispiel

Um uns mit der grundsätzlichen Vorgehensweise vertraut zu machen, werden wir einen kleinen Pult für eine Ausweichstelle mit zwei Weichen aufbauen (s. Bild). Für diese recht einfache Aufgabe benötigen wir zwei Reihen Tischfelder, pro Gleis jeweils eine. Dazu setzen wir uns im "2D-Planeditor" zwei unsichtbare Gleise mit dem erforderlichen Gleisabstand ein und wechseln in den "3D-Objekteditormodus".

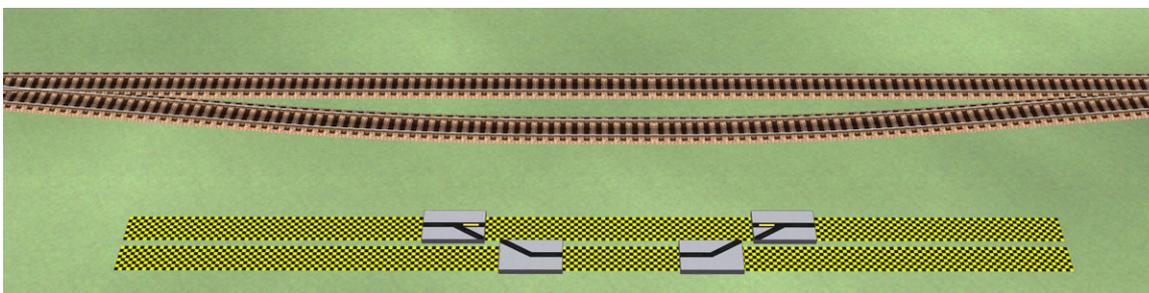


Jetzt legen wir fest, was an Symbolen benötigt wird. Wir haben eine rechte und eine linke Weiche und zwei Ausgleichsbögen. Weiterhin brauchen wir 4 gerade Gleismelder. Einen rechts vor der Weiche, einen links vor der Weiche und zwei für die beiden Gleise der Ausweichstelle.

Grundsätzlich sollte man beim Bau eines Pultes immer mit den Weichenstrassen beginnen und die dann verbleibenden Zwischenräume mit geraden Tischfeldern auffüllen. Wir beginnen den Aufbau in unserem Beispiel also mit den Weichen und den Ausgleichsbögen.

Wir suchen uns nun die Elemente zusammen und setzen sie auf die unsichtbaren Gleise. Bei der Auswahl sind wir zu folgendem Ergebnis gekommen:

Für die Weichen: Weiche links ist "SpDrS60\_Wche\_Li" ; Weiche rechts ist "SpDrS60\_Wche\_Re"  
Für die Ausgleichsbögen: Bogen links ist "SpDrS60\_Krve\_Re" ; Bogen rechts ist "SpDrS60\_Krve\_Li"

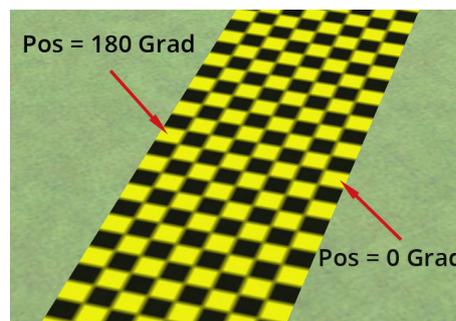


Das Ergebnis sollte dann in etwa so aussehen.

#### Hinweis:

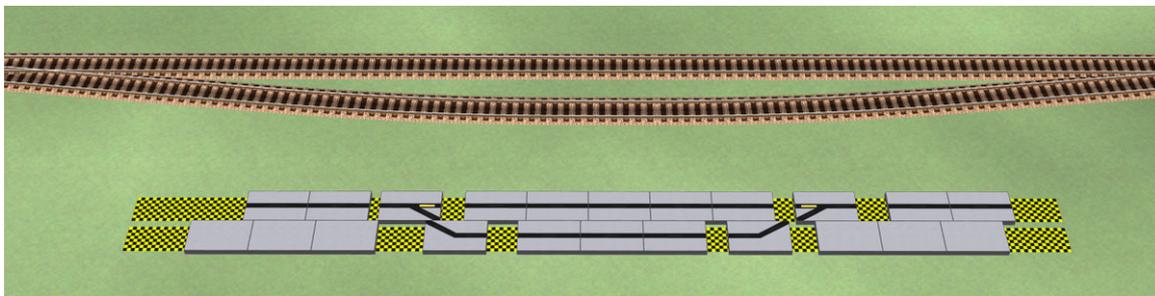
Hier möchte ich noch eine Zwischenbemerkung zum Einsetzen der richtungsabhängigen Symbole anbringen. Wenn die Elemente in die falsche Richtung zeigen, können sie einfach wieder entfernt und neu positioniert werden. Dabei ist darauf zu achten, dass die Ränder der Gleismarkierungen "Klick-sensitiv" sind und die Ausrichtung des eingesetzten Objekts bestimmen.

In der Abbildung sehen wir die Gleismarkierung als gelb-schwarzes Band. Nehmen wir nun an, dass ein Klick auf den rechten Rand der Markierung das Objekt auf 0 Grad einsetzt, so wird ein Klick auf den linken Rand das Objekt um 180 Grad gedreht einsetzen. Wir müssen also den "3D Modus" nicht verlassen, um ein Objekt in seine richtige Lage zu bringen.



## Erste Schritte

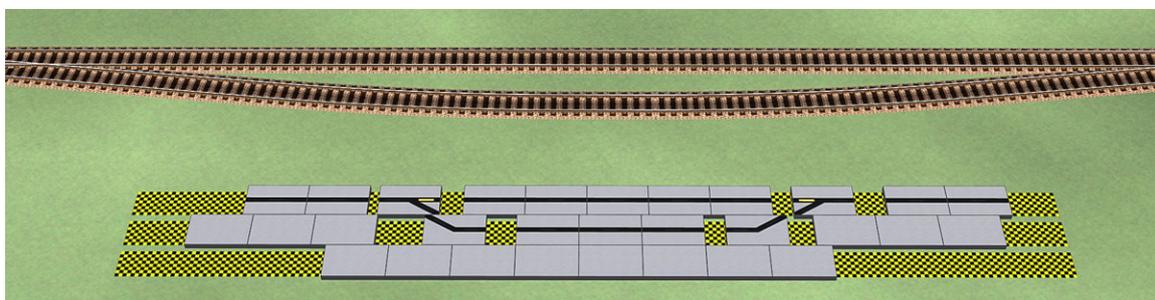
Nun erweitern wir unseren Aufbau um die Tischfelder mit den geraden Gleismeldern. Um die Gesamtausdehnung klein zu halten, setzen wir zuerst einen Melder für das Abzweiggleis mit 3 Einheiten Länge. Um das durchgehende Gleis darzustellen, benötigen wir jetzt einen Melder mit 5 Einheiten Länge. Damit gleichen wir die beiden Bögen und den Gleismelder des Abzweiggleises aus ( $3 + 1 + 1 = 5$ ). Für die Streckendarstellung vor den Weichen wählen wir uns jeweils einen Melder mit 2 Längeneinheiten. Um einen sauberen Abschluss der Tischfläche zu erreichen, ergänzen wir die Elemente um zwei Leerfelder mit 3 Einheiten Länge rechts und links neben dem Abzweiggleis. Wir gleichen damit die Weiche und den Streckenmelder aus ( $1 + 2 = 3$ ).



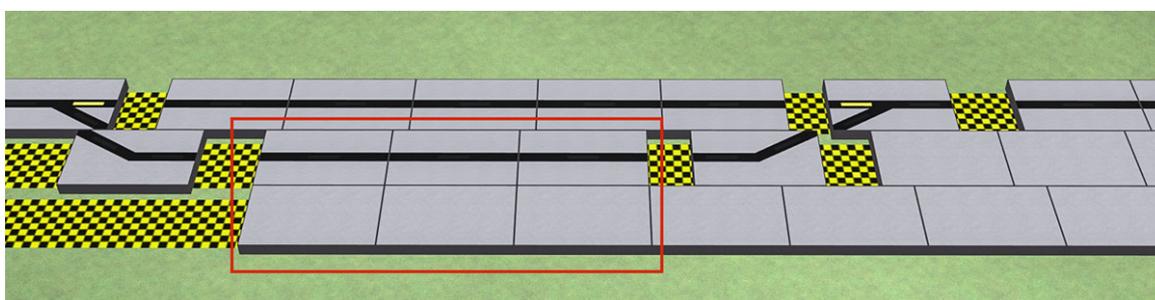
Die benötigten Elemente finden wir in der Inventarliste unter "Gerades Gleis". Dort sind die Dateinamen verzeichnet. Hat alles geklappt, sollte unser Beispiel wie auf der Abbildung aussehen.

### Das Ausrichten

Als nächstes müssen wir unsere Tischelemente ausrichten. Eine Rastfunktion in Richtung der Gleisachse ist konstruktiv nicht möglich, da es sich bei allen Objekten um Signale handelt. In EEP sind Signalobjekte lediglich stufenlos auf der Gleisachse verschiebbar. Für diesen Schritt werden wir uns mit einem zusätzlichen Gleis und einem Leerfeld von 8 Einheiten Länge behelfen. Durch das Raster auf den Tischelementen stellt das Ausrichten kein Problem dar.



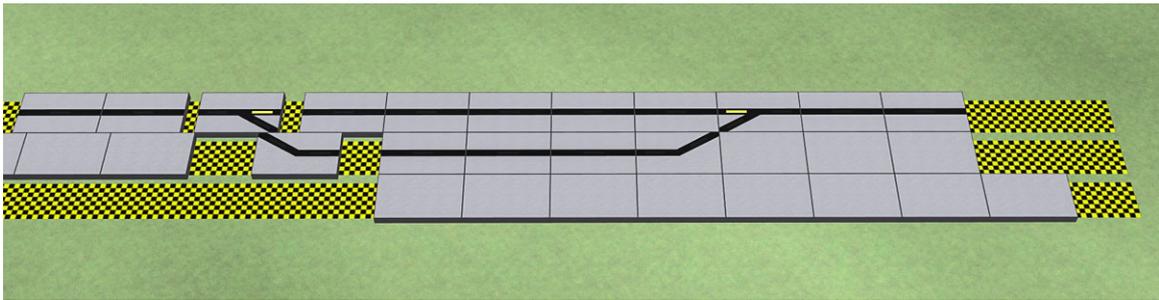
Im Vordergrund der Abb. kann man nun das Hilfsgleis mit dem 8er Leerfeld sehen. Wir positionieren als erstes die Symbole der rechten Seite.



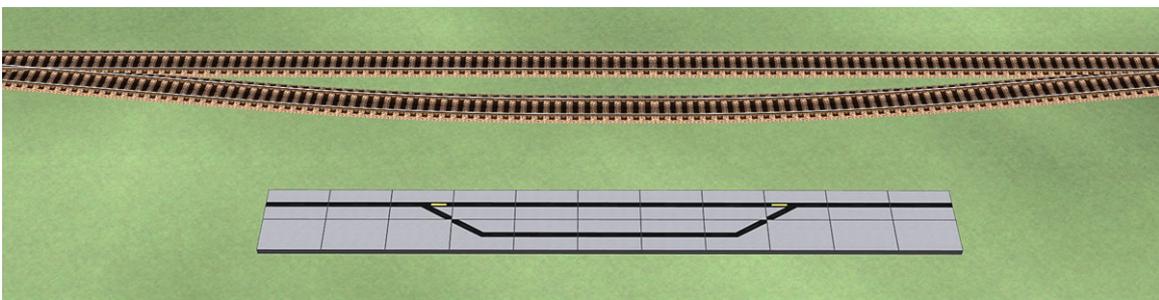
Dazu bringen wir die 3 linken Felder des 8er-Leerfelds mit den "Nuten" des 3er-Gleismelders (Abb. roter Rahmen) zur Deckung. Wir beginnen jetzt die Elemente unseres Gleisplans in der entsprechenden Zoomstufe am Nutenraster auszurichten. Dabei gilt die Regel "Je höher die Zoomstufe, um so feiner die Bewegung des Objekts". Mit ein wenig Übung lassen sich schnell gute Ergebnisse erzielen.

## Erste Schritte

---

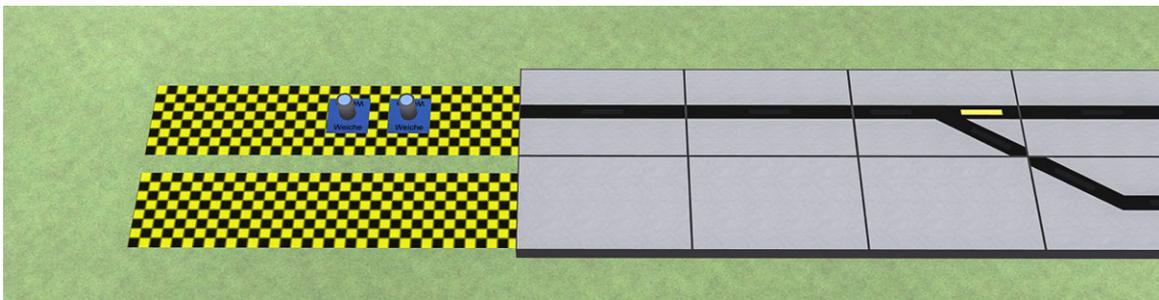


Nach diesem Schritt nimmt unser Stellpult schon Gestalt an. Das Ergebnis sollte dann etwa der Abbildung entsprechen. Das gleiche Verfahren wenden wir jetzt auf der rechten Seite an. Sind alle Elemente ausgerichtet, können wir das Leerfeld und das Hilfsgleis wieder löschen. Unser kleines Pult ist jetzt im "Rohbau" fertig.

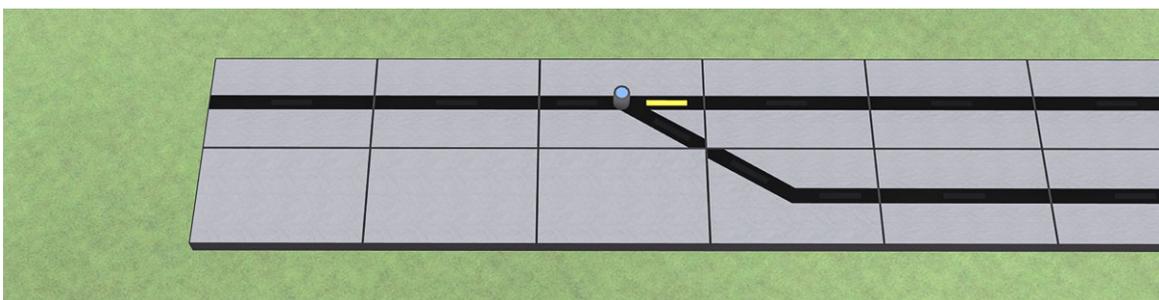


### Die Taster

Wir haben nun unser Gleisbild erstellt. Es besteht allerdings noch keinerlei Verbindung zu unserem kleinen Bahnhof. Widmen wir uns nun den Weichen. Um sie schalten zu können, benötigen wir einen Taster pro Weiche. Dazu schauen wir wieder in die Inventarliste und finden unter "Taster" den Weichentaster "SpDrS60\_Tstr\_Wche". Den werden wir zweimal in unseren Pult einbauen.



Nach dem Einsetzen der Taster auf das freie Gleis am Ende der Hauptgleis-Symbole verschieben wir die Taster auf ihre Position in die Mitte der Weichenfelder.



Als nächstes verknüpfen wir die Taster und Melder mit den Weichen. Dazu benötigen wir die ID-Nummern der Objekte. Die Id's der Weichen finden wir in der 2D-Planansicht, die der Weichenmelder im "Objekteigenschaften-Dialog". Im folgenden Abschnitt wird die Verbindung ausführlich beschrieben.

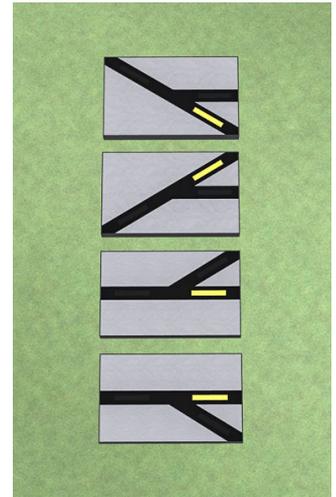
## Die Weiche

Zur Schaltung und optischen Anzeige der Weichenstellung stehen Ihnen ein Weichen-taster und vier Tischfelder zur Verfügung. Sie haben die Wahl zwischen der horizontalen und der diagonalen Ausführung, jeweils in rechter und linker Darstellung.

Der Taster und die Meldersymbole (Abb. 1) tragen folgende Bezeichnungen:

- Weichenmelder diagonal Links - SpDrS60\_Wche\_diag\_Li
- Weichenmelder diagonal Rechts - SpDrS60\_Wche\_diag\_Re
- Weichenmelder horizontal Links - SpDrS60\_Wche\_Li
- Weichenmelder horizontal Rechts - SpDrS60\_Wche\_Re
- Weichentaster (Blau) - SpDrS60\_Tstr\_Wche

Die einzelnen Funktionen der Komponenten können Sie der Inventarliste entnehmen.



### Taster mit Weiche verbinden

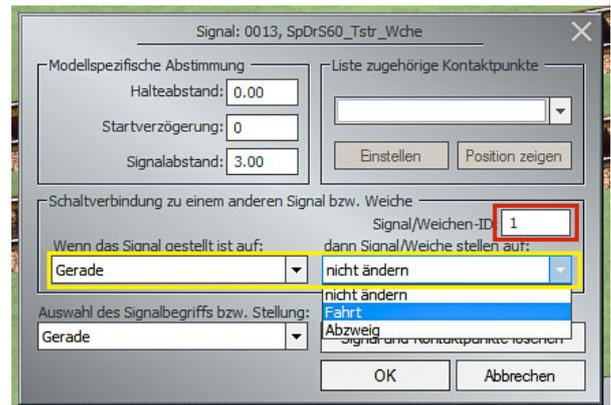
Die Weiche in diesem Beispiel hat durch EEP die ID1 erhalten. Wir öffnen durch einen "Rechtsklick" auf den zugehörigen Taster das Dialogfeld und rufen die Objekteigenschaften auf. Nun tragen wir im Feld "Signal/Weichen-Id" die Id-Nummer unserer Weiche ein (roter Rahmen). Erst dann ist das Menü "Schaltverbindung" aktiv.

In den Pull-down-Menüs "Wenn..dann" (gelber Rahmen) stellen wir die Verknüpfungen zwischen Taster und Weiche mit den folgenden Bedingungen ein:

- Wenn "Gerade" dann "Fahrt"
- Wenn "Abzweig" dann "Abzweig"

Durch schließen mit dem "OK-Button" übernehmen wir die Einstellungen.

Zum Testen verlassen wir den Edit-Modus und klicken einige Male auf den Weichentaster. Unsere Weiche sollte nun bei jedem Klick ihre Stellung ändern, zu erkennen an der sich drehenden Weichenlaterne.



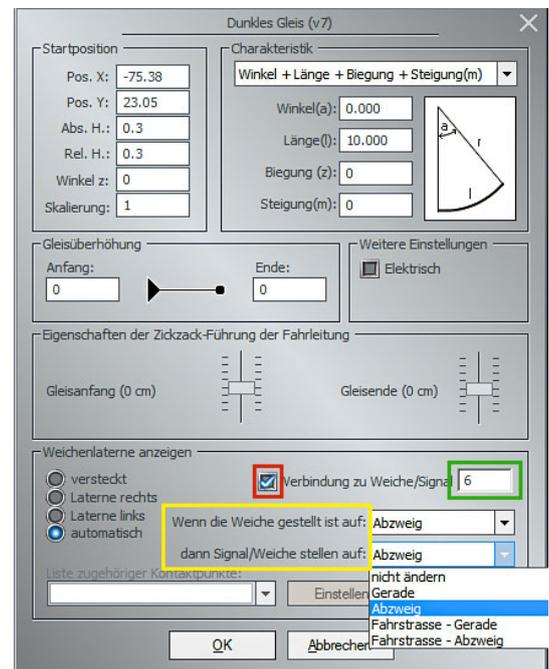
### Weiche mit Melder verbinden

Als nächstes stellen wir die Verknüpfung zwischen der Weiche und der Meldefunktion des Tischfeldes her. Dazu wechseln wir wieder in den 3D-Editiermodus und aktivieren mit einem "Rechtsklick" auf die Weichenlaterne das Dialogfeld für die Objekteigenschaften der Weiche. Wir setzen das Häkchen (roter Rahmen) und tragen in das Eingabefeld (grüner Rahmen) die ID unseres Tischfeldes ein. Auch hier sind die Menüs für die Verknüpfungen erst nach Setzen des Häkchens nutzbar, andernfalls sind sie ausgegraut.

Die Verknüpfungen (gelber Rahmen) nehmen wir mit folgenden Bedingungen vor:

- Wenn "Fahrt" dann "Gerade"
- Wenn "Abzweig" dann "Abzweig"

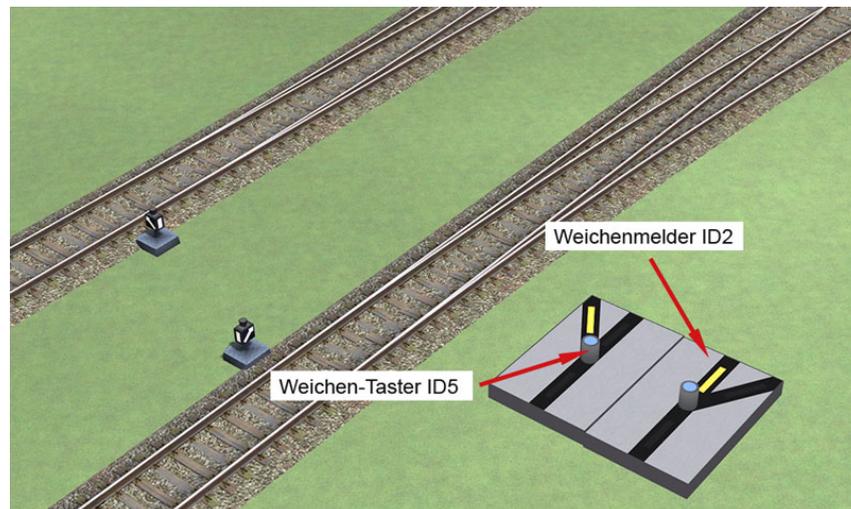
Durch schließen mit dem "OK-Button" übernehmen wir die Einstellungen. Zum Testen verlassen wir den 3D-Editiermodus und klicken einige Male auf den Weichentaster. Der Tischmelder sollte jetzt bei jedem Klick die Anzeige ändern und die Stellung der Weiche korrekt wiedergeben. Damit ist die Verknüpfung der Weiche vollständig.



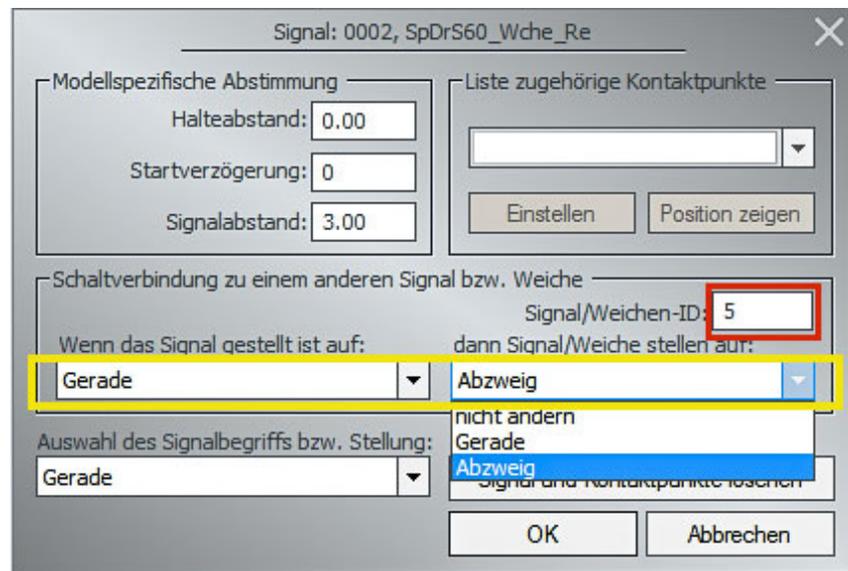
## Die Weiche

### Abhängigkeit zwischen zwei Weichen ( z.B. Flankenschutz )

Durch die Verknüpfung der Weiche mit dem Tischmelder ist keine weitere Verbindung mit anderen Objekten möglich, da der Objekteigenschaften-Dialog hierfür nur eine mögliche Eingabe vorsieht. Um trotzdem eine Abhängigkeit zwischen zwei Weichen herzustellen, z.B. einen Flankenschutz, muss ein alternativer Weg beschritten werden. Wir bedienen uns dazu der Stellpultelemente.



Bei dem Aufbau im Bild nutzen wir dazu den Weichenmelder ID2 der Rechtsweiche im Bildvordergrund. Durch Verknüpfung des Weichenmelders mit dem Weichentaster ID5 für die Linksweiche im Bildhintergrund stellen wir eine lageabhängige Verbindung zwischen den beiden Weichen her.



Die Wirkrichtung dieser Verknüpfung ist aus dem Dialogfeld ersichtlich. In diesem Fall sendet der Weichenmelder ID2, dessen Objekteigenschaften wir sehen (s. Fenstertitelzeile), dem Weichentaster ID5 (roter Rahmen) einen entgegengesetzten Stellbefehl (Gerade dann Abzweig, gelber Rahmen), den man auch an den Melderstellungen der Weichen erkennen kann.

Es ist also möglich, trotz der genannten Einschränkungen von nur einer Wenn..Dann Verknüpfung pro Objekt, die Verbindung zwischen zwei Weichen herzustellen.

## Die Signale

---

Das Basisset enthält die wichtigsten Signale und Taster, wobei die Symbole als frei positionierbare Piktogramme ausgeführt sind. Das ermöglicht die flexible Verwendung auf beliebig langen Gleismeldern (1 - 8 LE). Dabei sind einzelne Signale als auch Signalkombinationen berücksichtigt. In diesem Abschnitt werden die Objekte beschrieben.

### Die Hauptsignale

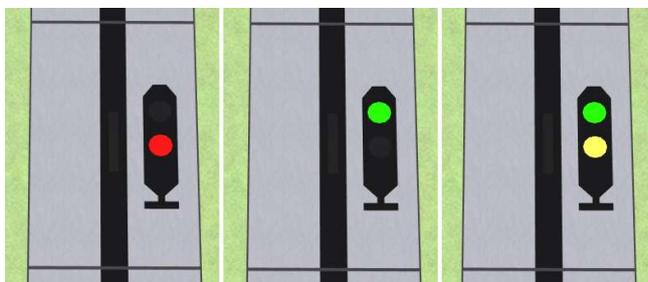
Im Set sind fünf Hp-Signalsymbole zu finden. Drei stellen einzeln stehende Signale dar und zwei können in Kombination mit anderen Signalmeldern genutzt werden. Die zwei Ergänzungssignale dienen zur Darstellung von Signalen "an einem Mast". Für den Begriff Hp2 wird entgegen der Vorbildfunktion bei allen Hp-Meldern zusätzlich die gelbe Leuchte gezeigt. Dieser Kompromiss wurde zur besseren Unterscheidung der Stellungen Hp1 und Hp2 eingegangen.

Alle Hauptsignalmelder können drei Signalbegriffe darstellen.

- Hp0 = Halt
- Hp1 = Fahrt
- Hp2 = Langsamfahrt

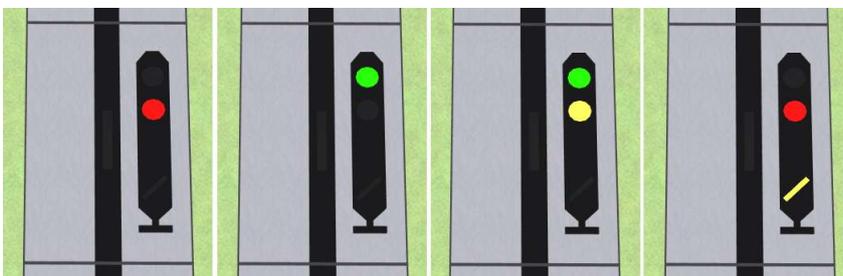
#### Hp-Signal "SpDrS60\_Sig\_HP"

Ausführung: einzeln stehend  
Begriffe: Hp0 - Hp1 - Hp2



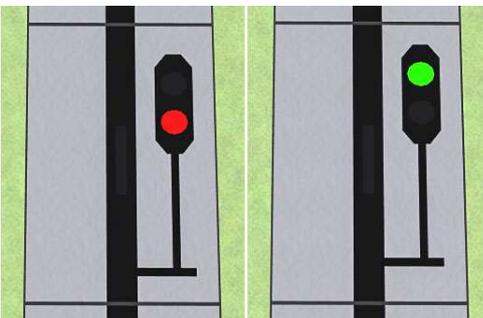
#### Hp-Signal "SpDrS60\_Sig\_HP\_Sh2"

Ausführung: einzeln stehend  
Begriffe: Hp0 - Hp1 - Hp2 - Sh1



#### Hp-Blocksignal "SpDrS60\_Sig\_HP\_BlK"

Ausführung: einzeln stehend  
Begriffe: Hp0 - Hp1



## Die Signale

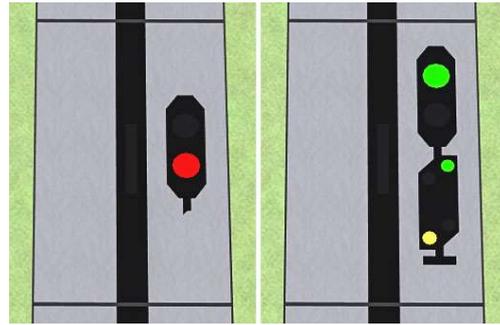
---

### Die Hauptsignalergänzungen

#### Hp-Signal "SpDrS60\_Sig\_A\_E\_Ergz"

Ausführung: Ergänzung  
Begriffe: Hp0 - Hp1 - Hp2

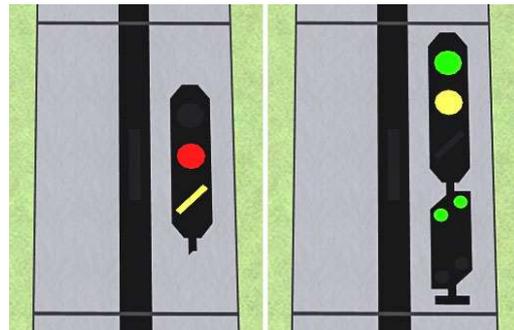
Die rechte Abbildung zeigt das Signal in Verbindung mit einem Vr-Signal in Stellung Hp1-Vr2.



#### Hp-Signal "SpDrS60\_Sig\_Hp\_Sh\_Ergz"

Ausführung: Ergänzung  
Begriffe: Hp0 - Hp1 - Hp2 - Sh1

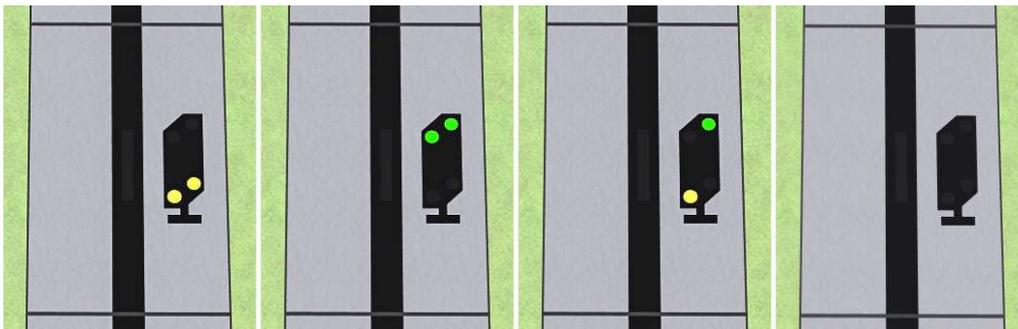
Die rechte Abbildung zeigt das Signal in Verbindung mit einem Vr-Signal in Stellung Hp2-Vr1.



### Das Vorsignal

Im Set ist ein Vorsignalmelder "SpDrS60\_Sig\_VR" enthalten. Er kann als einzeln stehendes Signal als auch in Kombination mit den Ergänzungen der Hauptsignale verwendet werden. Es können drei Signalbegriffe dargestellt werden:

- Vr0 = Halt erwarten
- Vr1 = Fahrt erwarten
- Vr2 = Langsamfahrt erwarten



Als vierte Stellung lässt sich der Melder dunkel schalten. Diese Position dient unter anderem zur Darstellung des Hp0 Signalbegriffs bei Signalkombinationen an einem Mast. Zeigt der Hp-Melder den Begriff Hp0, erlischt der Vr-Melder.

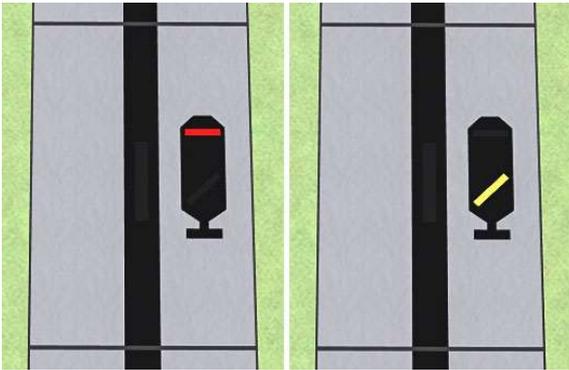
## Die Signale

---

### Das Rangiersignal

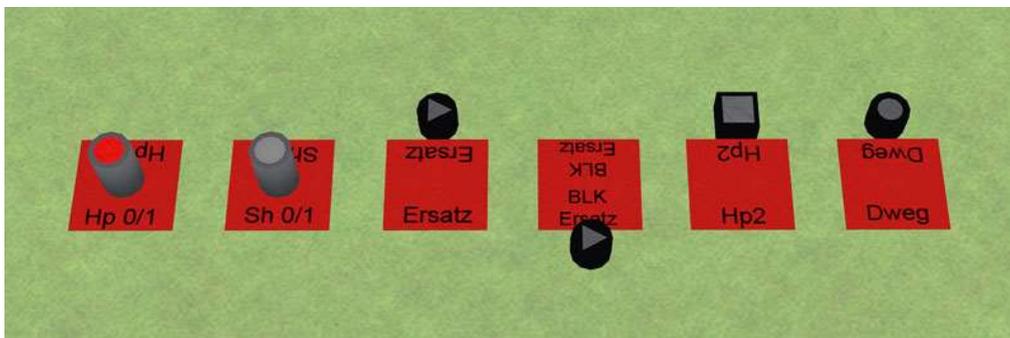
Zur Darstellung von LS- und Form-Gleissperrsignalen ist im Set ein Rangiersignalmelder "SpDrS60\_Sig\_Ls\_Sh" enthalten. Er ist als einzeln stehendes Signal ausgeführt. Es können zwei Signalbegriffe dargestellt werden:

- Sh0 = Halt
- Sh1 = Rangierfahrt



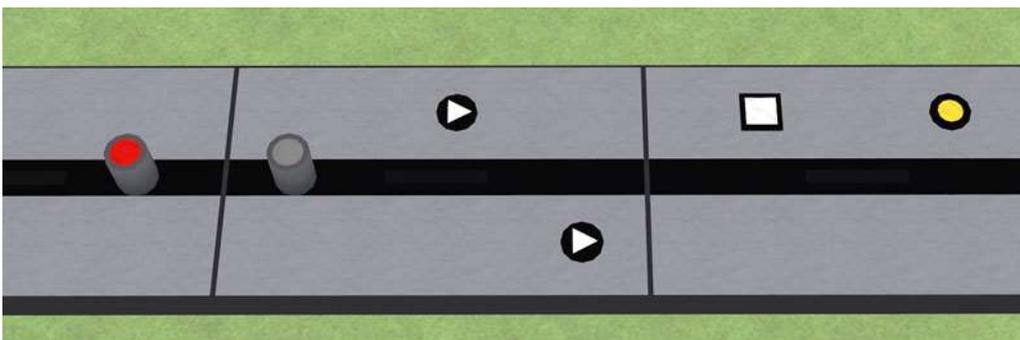
### Die Signaltaster

Um die Signale betätigen zu können, sind in der Grundausstattung sechs verschiedene Taster enthalten. Ihre Funktion lässt sich an der Sockelbeschriftung erkennen. Wir finden sie in der Inventarliste unter "Taster".



Regulär ist der Hp2-Taster ein Festlegemelder, wird im Set aber abweichend dazu genutzt, die Hp2-Stellung von mehrbegriffigen Signalen direkt zu schalten. Darum trägt er die entsprechende Sockelbezeichnung.

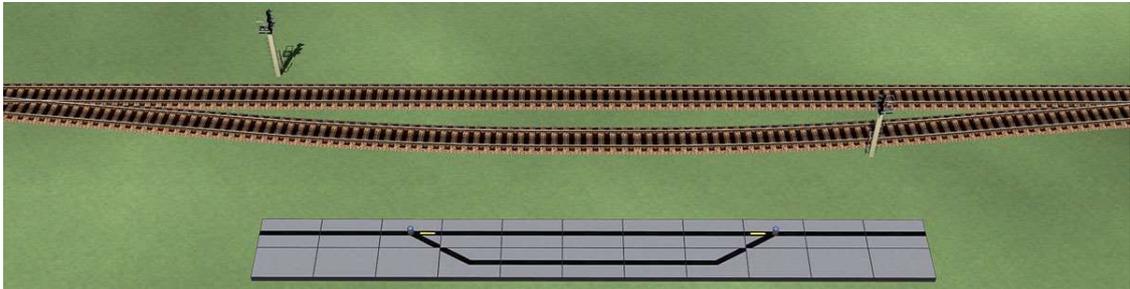
Das gleiche gilt für den D-Weg Melder, der zum Aufruf der Funktion "Vorbeifahrt am Halt zeigenden Signal" verwendet werden kann. Das folgende Bild zeigt die Taster, zum Teil aktiv, in einem Gleismelder.



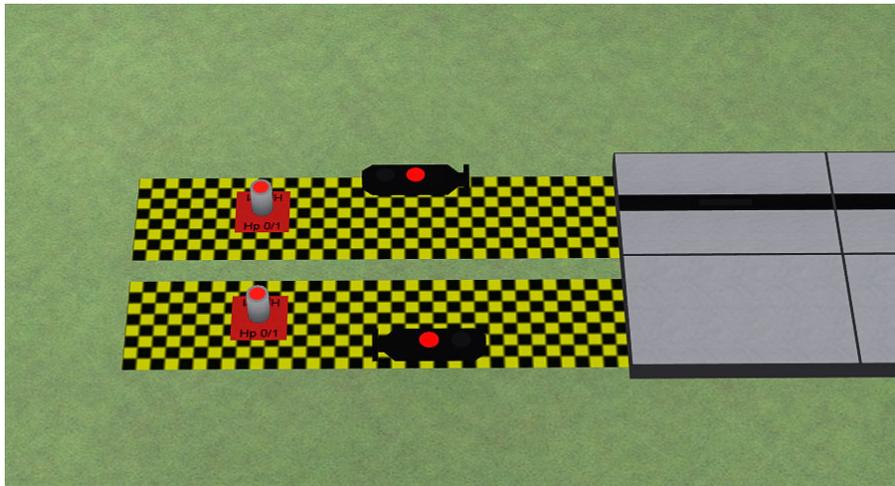
## Die Signale

---

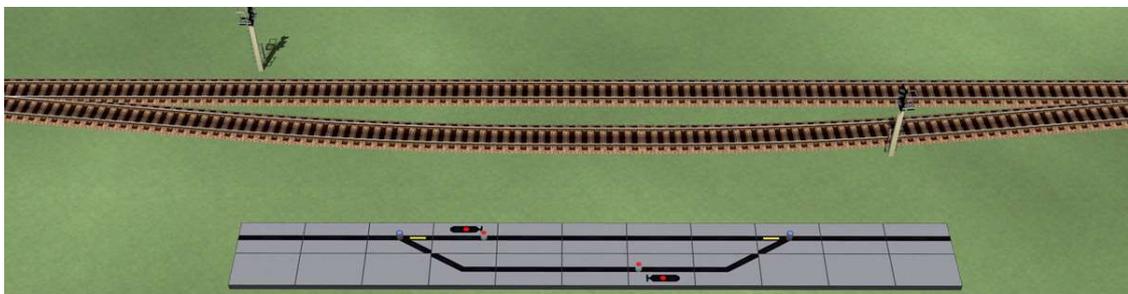
Wir wenden uns nun wieder unserem Beispielaufbau zu und werden den kleinen Bahnhof mit Signalen ausstatten. Wir verwenden dazu einfache Signale mit einem Halt- und Fahrtbegriff.



Als erstes benötigen wir die passenden Signalmelder und Taster. Wir nutzen dazu einen Hp-Melder "SpDrS60\_Sig\_HP" und als Taster den "SpDrS60\_Tstr\_Sig\_Hp". Wir setzen sie auf dem freien Gleisabschnitt neben unseren Feldelementen ein und beachten dabei die Einsetzrichtung der Hp-Melder.



Dann schieben wir die Objekte an ihre Position in den Gleismelder. Ist das geschehen, sollte unser Beispiel in etwa wie in dem folgenden Bild aussehen.

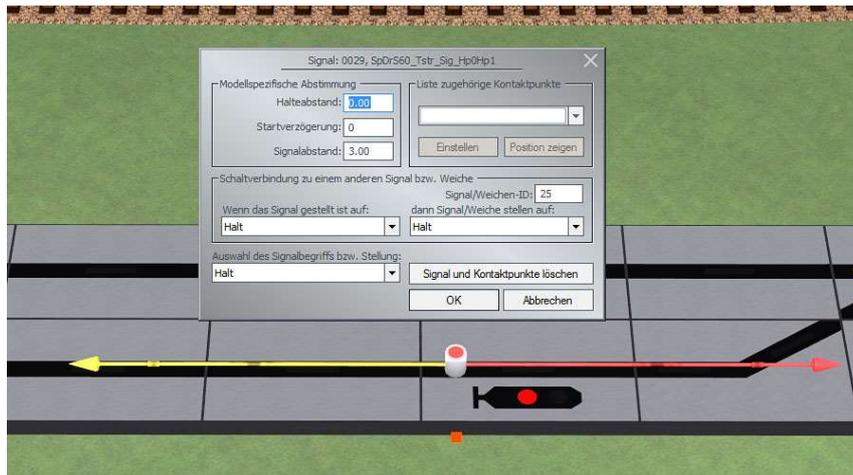


## Die Signale

### Taster mit Signal verbinden

Wir stellen nun die Verknüpfung zwischen dem Taster und dem Signal her. Dazu öffnen wir die Objekteigenschaften des Tasters und tragen die Id des Signals ein. In den Schaltverbindungen nehmen wir folgende Verknüpfungen vor:

- Wenn "Fahrt" dann "Fahrt"
- Wenn "Halt" dann "Halt"

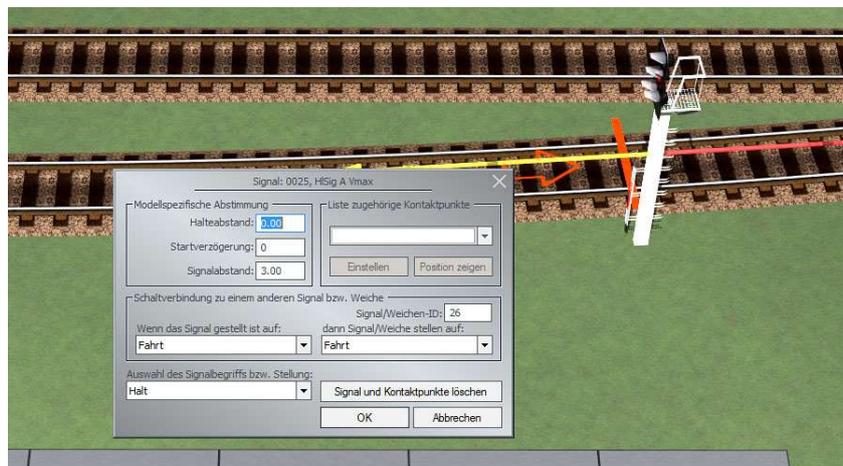


Durch schließen mit dem "OK-Button" übernehmen wir die Einstellungen. Wir können unsere Verbindung jetzt schon ausprobieren. Dazu klicken wir einige Male auf den Taster und beobachten den Wechsel am Signal.

### Signal mit Melder verbinden

Jetzt erfolgt der letzte Schritt unserer Signalverbindung. Dazu rufen wir die Objekteigenschaften des Signals auf. Nachdem wir die ID des Hp-Melders eingetragen haben, stellen wir die Schaltverbindungen mit folgenden Optionen ein:

- Wenn "Fahrt" dann "Fahrt"
- Wenn "Halt" dann "Halt"



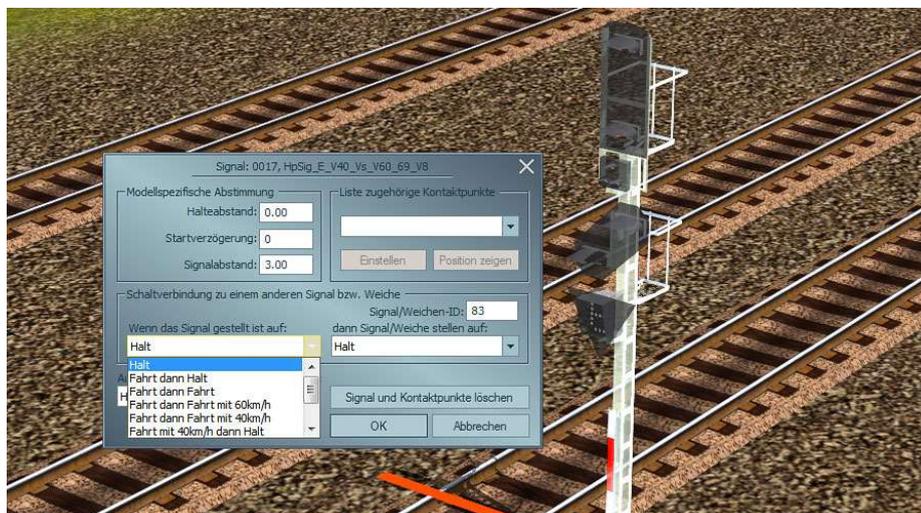
Durch schließen mit dem "OK-Button" übernehmen wir auch diese Einstellungen. Zur Überprüfung verlassen wir den Editermodus und klicken wieder einige Male auf den Taster. Der Wechsel der Signalbegriffe sollte jetzt am Melder angezeigt werden. Nach dem selben Schema verfahren wir für das zweite Signal.

## Die Signale

### Die redundante Signal-Melder Verknüpfung

In unserem Beispielbahnhof handelte es sich um ein recht einfaches Signal mit lediglich zwei Signalbegriffen, Halt und Fahrt. Mittlerweile existieren in EEP aber Signale mit bis zu 11 und mehr verschiedenen Positionen.

Welche sollen nun mit dem Melder verknüpft werden? Die Antwort ist einfach: ALLE!



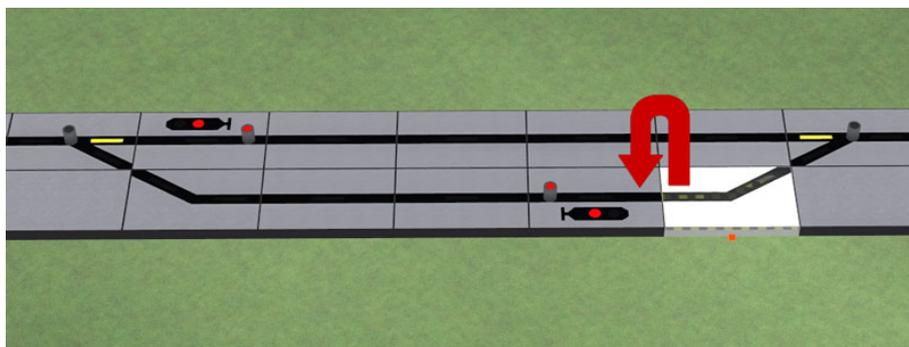
Das Bild zeigt den Objekteigenschaften-Dialog eines solchen Signals. Wie man erkennen kann, besitzt es unter dem Halt-Begriff einige Positionen, die mit dem Ausdruck "Fahrt" beginnen. Alle diese Optionen verknüpfen wir mit der Melderposition "Fahrt". Es ist aber darauf zu achten, ob sich hinter dem Begriff "Fahrt" vielleicht eine Geschwindigkeitsbegrenzung verbirgt. In dem Dialog ist z.B. der letzte Eintrag "Fahrt mit 40 km/h dann Halt" ein Hp2-Befehl, also eine "Langsamfahrt". Die muss dann mit der Melderstellung "Langsamfahrt" verknüpft werden. Dieses Verfahren erzeugt also eine sog. Redundanz, zu unserem Bezug frei übersetzt ein "Überangebot von Bedingungen".

Der Sinn dieser Vorgehensweise erschließt sich aber recht schnell. Klicke ich mit gedrückter Shift-Taste mehrmals auf ein Signal, so werden die Signalpositionen Schritt für Schritt durchlaufen. Ist das Signal redundant mit dem Hp-Melder verbunden, wird er immer die korrekte Anzeige auf dem Pult liefern. Wir benötigen das beschriebene Verfahren später für die Signalverkettung, die in einem eigenen Thema behandelt wird.

## Die einfache Gleisbesetzungsmeldung

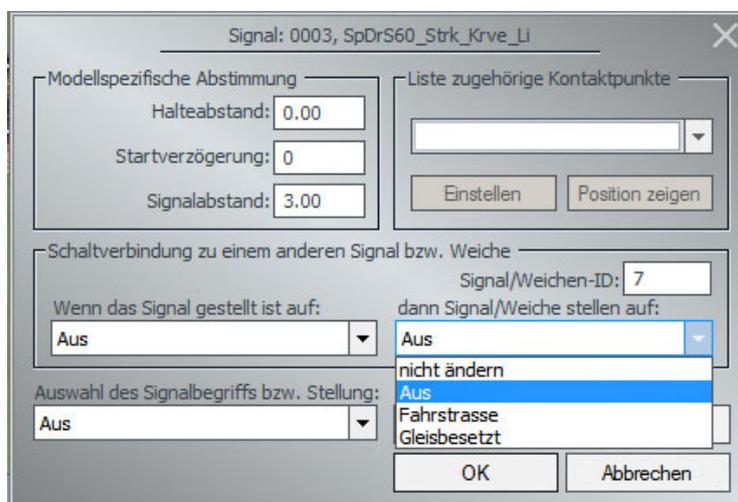
In diesem Abschnitt werden wir in unserem Beispiel eine einfache Gleisbesetzungsmeldung einrichten. Wir beginnen dabei mit dem Ausweichgleis.

Die Darstellung des Gleises auf dem Pult besteht aus drei einzelnen Elementen, zwei Kurven und dem Gleismelder 3E. Dieser Umstand wäre für das spätere Handling allerdings ungünstig, da wir für ein Gleis drei Objekte ansprechen müssten. Also werden wir im ersten Schritt diese drei Elemente zu einem "verketteten". Dazu legen wir uns eine Seite der Elementgruppe als Ursprung fest. In diesem Fall wählen wir das rechte Kurventischfeld. Dieses Feld soll seinen Zustand an den Gleismelder 3E als Stellbefehl senden.



Wir öffnen dazu im "3D-Edit Modus" das Dialogfeld "Objekteigenschaften" des Kurventischfeldes und geben die ID des Gleismelder 3E im Abschnitt "Schaltverbindung" ein. Dann stellen wir die "Wenn...Dann" Verknüpfung nach folgendem Schema ein:

- Wenn "Aus" dann "Aus"
- Wenn "Fahrstraße" dann "Fahrstraße"
- Wenn "Gleisbesetzt" dann "Gleisbesetzt"



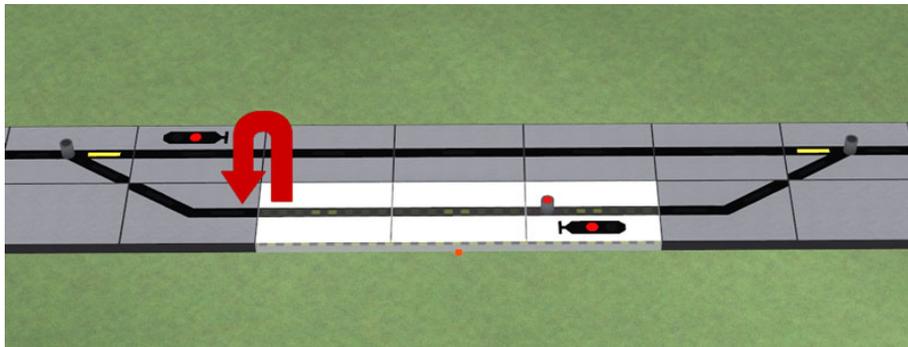
Mit einem Klick auf "OK" schließen wir den Dialog. Der Gleismelder 3E folgt nun jedem Stellbefehl, den wir an das rechte Kurvenelement senden.

*Hier möchte ich noch eine Zwischenbemerkung anbringen:*

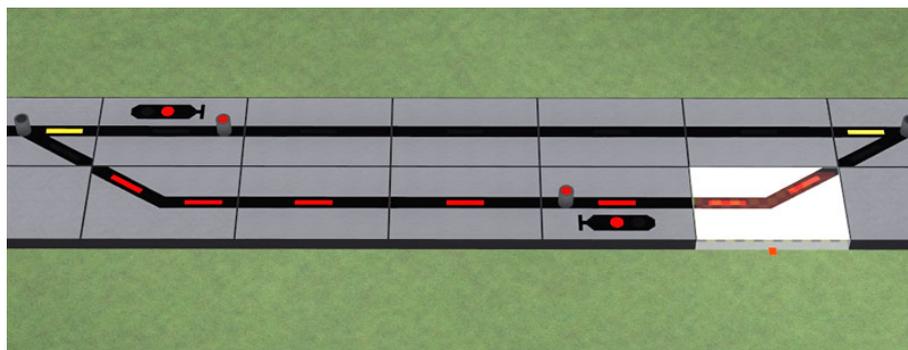
Wir haben im Dialog die Bedingung - Wenn "Fahrstraße" dann "Fahrstraße" - eingesetzt. Im Basisset wird diese Option allerdings noch nicht unterstützt. Dazu wird eine Ergänzung erscheinen, die diese Möglichkeit beinhaltet. Wir stellen diese Bedingung jedoch jetzt schon ein. Das erspart uns bei einem Upgrade eine Nachbearbeitung der Verknüpfungen.

## Die einfache Gleisbesetztmeldung

Nach dem gleichen Schema wie oben beschrieben verfahren wir nun bei der Verknüpfung des Gleismelders 3E und dem linken Kurvenelement.



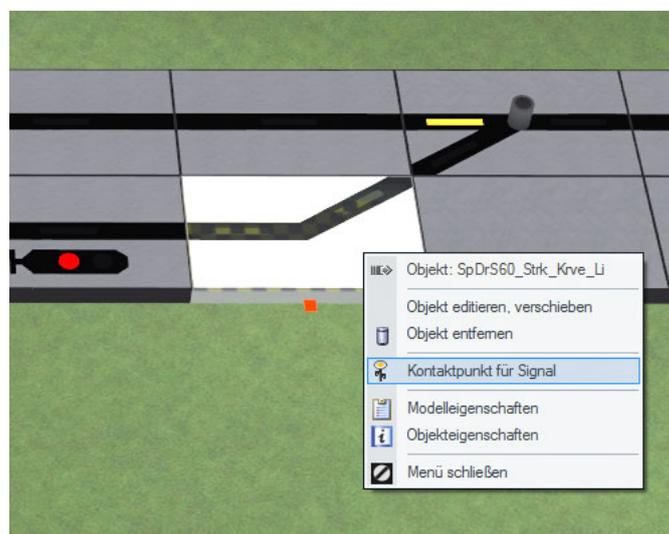
Wir können nun eine erste Funktionsprüfung ausführen. Dazu öffnen wir die Objekteigenschaften des rechten Kurventischfeldes, also dem Ursprung unserer Verknüpfung, und wählen im Abschnitt "Auswahl des Signalbegriffs" die Option "Gleisbesetzt". Wenn wir den Dialog mit "OK" beenden, sollten alle Melder für das Ausweichgleis rot ausgeleuchtet sein.



Für den nächsten Schritt haben wir jetzt statt drei nur noch einen "Ansprechpartner".

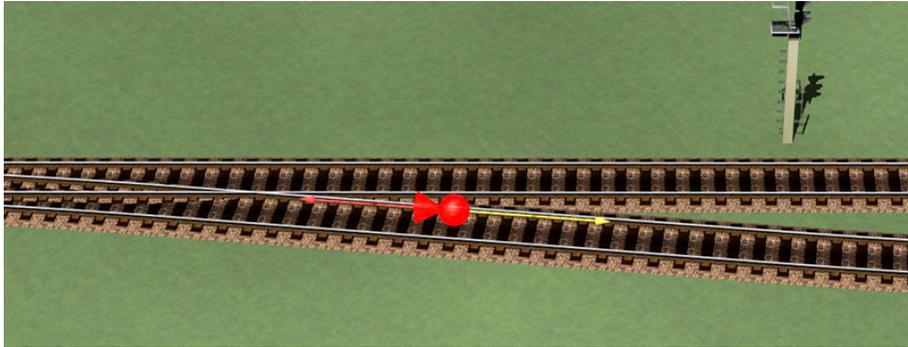
### Die Signalkontakte

Damit uns der Gleisbesetzungsstatus angezeigt werden kann, müssen wir eine Verbindung zu unserem Ausweichgleis auf der Anlage herstellen. Dazu verwenden wir Signalkontakte. In unserem Beispiel haben wir auf den Gleisen einen "Rechtsverkehr", d.h. wir befahren die Gleise immer nur aus einer Richtung. Aus diesem Grund genügen uns zwei Kontakte pro "Bahnhofsgleis". Durch einen Rechtsklick auf unser Ursprungselement öffnen wir das "Kontextmenü". Dort wählen wir die Option "Kontaktpunkt für Signal".

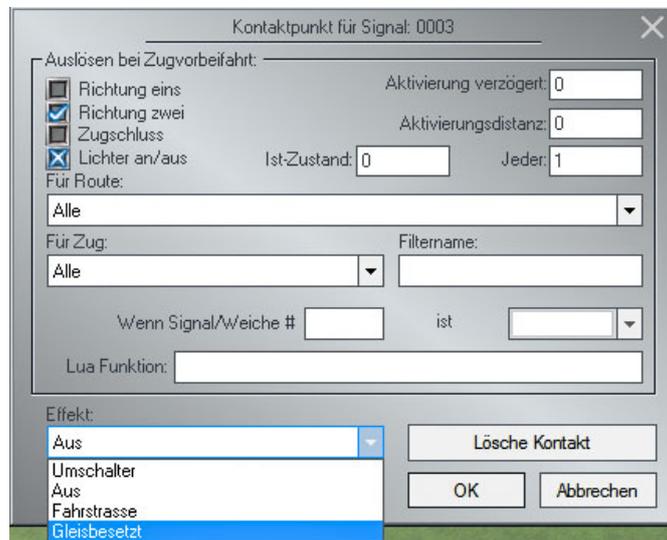


## Die einfache Gleisbesetzmeldung

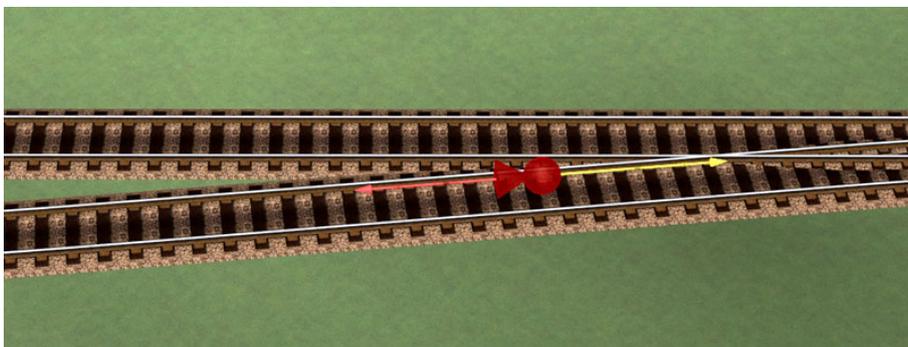
Der erste Kontaktpunkt wird kurz hinter der linken Weiche platziert, in diesem Fall die Einfahrweiche für das Ausweichgleis.



Mit einem Rechtsklick auf den Kontaktpunkt öffnen wir sein Kontextmenü und wählen die "Objekteigenschaften" aus.

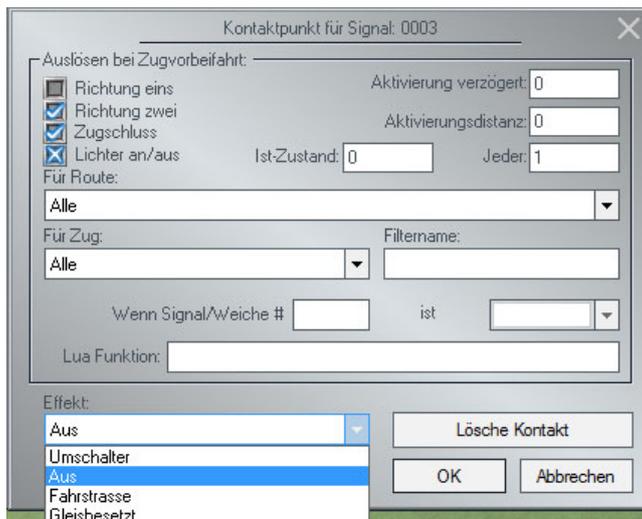


Hier stellen wir die Wirkrichtung des Kontakts und den Effekt "Gleisbesetzt" ein. Mit einem Klick auf "OK" schließen wir den Dialog. Überfährt nun ein Zug unseren Kontakt, wird er die Gleisbesetzmelder einschalten. Damit ein ausfahrender Zug die Melder wieder ausschaltet, benötigen wir natürlich auch einen Kontakt, der diese Aufgabe bewerkstelligt. Um uns ein wenig Arbeit zu sparen, klicken wir wieder mit Rechts auf unseren Einfahrkontakt und wählen aus dem Kontextmenü die Option "Kontaktpunkt kopieren". Dann bewegen wir den Mauszeiger, der jetzt ein Kontaktsymbol an seiner Spitze führt, auf das Gleis kurz vor der Ausfahrweiche und platzieren ihn mit einem Linksklick an dieser Position.



## Die einfache Gleisbesetzungsmeldung

Nun öffnen wir die "Objekteigenschaften" und nehmen die Einstellungen für diesen Kontaktpunkt vor.



Hier ist darauf zu achten, dass der Haken bei "Zugschluss" gesetzt ist. Das stellt sicher, dass die Gleisbesetzungsmeldung erst nach verlassen des letzten Rollmaterials aufgehoben wird. Als Effekt wählen wir die Option "Aus". Wir schließen den Dialog mit "OK". Unsere erste einfache Gleisbesetzungsanzeige ist jetzt fertiggestellt.

Bei der Einrichtung der Besetzungsmeldung für das andere Gleis verfahren wir nach dem gleichen Schema. Da wir dort nur ein Gleisbesetzungsmelder mit 5 Längeneinheiten haben, entfällt die "Verkettung". Das zeigt uns auch, dass wir beim Verlegen der Gleismelder die minimale Anzahl anstreben sollten. Dazu stehen uns Gleismelder mit bis zu 8 Längeneinheiten zu Verfügung.

Bei den "Streckengleisen" haben wir einen Zweirichtungsbetrieb. Dort müssen wir dann selbstverständlich noch ein Kontaktpaar für die Gegenrichtung vorsehen. Aber auch da hilft uns die Kopierfunktion für Kontakte den Aufwand zu minimieren.

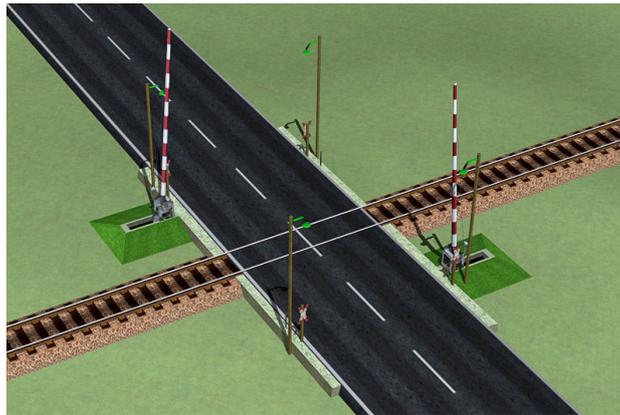
### *Nachüberlegung*

Die Stärke der hier besprochenen einfachen Gleisbesetzungsmeldung liegt in der Überwachung von Blockabschnitten und Gleisen, die von Zügen ohne Zerlegung durchfahren werden.

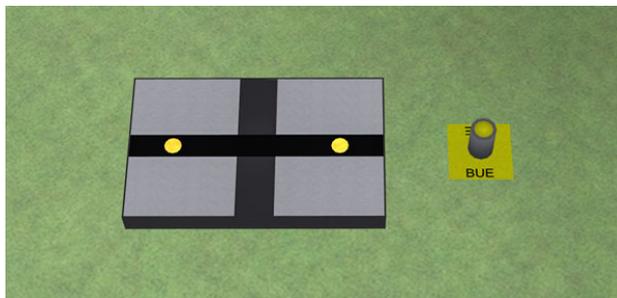
Sie eignet sich nicht für Bahnhofsgleise, auf denen Züge zerlegt werden oder für den Einsatz in Kopfgleisen. In diesen Situationen greift man besser auf die abschnittsdeckende Gleisbesetzungsmeldung zurück, die in einem eigenen Thema behandelt wird.

## Der Bahnübergang

---



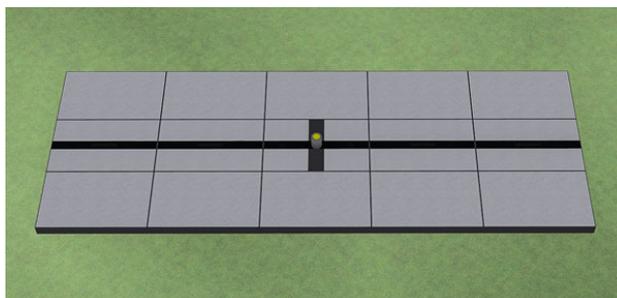
Um Bahnübergänge in unserem Pult mit einzubeziehen, steht uns ein BÜ-Tischfeld und ein passender Taster zur Verfügung. Wir finden beiden Objekte in der Inventarliste unter "Streckenfelder" und "Taster". Das Tischfeld ist als Basiselement ausgelegt, das zur Bedienung von Fü\*- und Hp\*-Bahnübergängen verwendet wird. In der im Set vorliegenden Ausführung werden lediglich die Bahnübergangsanschaltmelder angezeigt, die bei geschlossener Schranke leuchten.



Bei den beiden Vollschranken handelt es sich um zwei Signale, die jeweils in eine Fahrtrichtung der Straße wirken. Um die beiden Schranken ( Signale ) mit einem Taster anzusprechen, müssen wir sie verknüpfen. Zur Verdeutlichung hilft diese Skizze der gesamten Verknüpfungsreihenfolge:

*BÜ-Taster* → *Schranke 1* → *Schranke 2* → *Schranke..n* → *BÜ-Tischfeld*

Der Schritt "*Schranke..n*" soll dabei anzeigen, dass wir in dieser Kette weitere Schranken aufnehmen können, so z.B. zwei Halbschranken für die Straße plus zwei Halbschranken für den Fuß- oder Radweg.

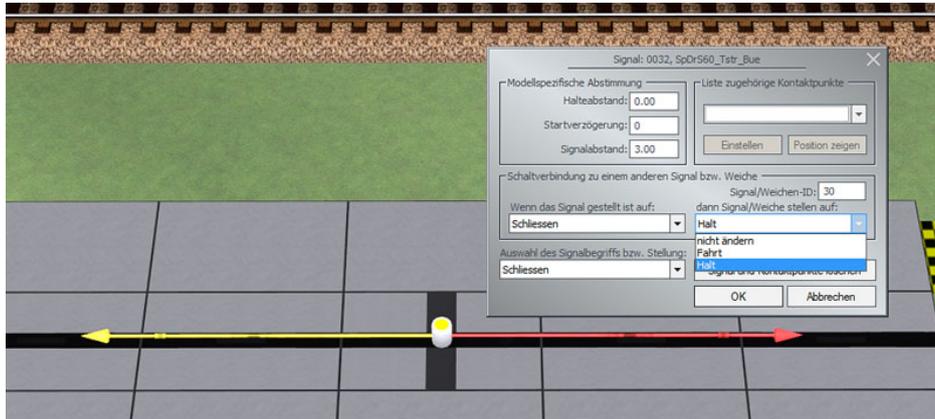


Zuerst wählen wir eine Schranke aus, mit der wir die Verknüpfung beginnen. Die in diesem Abschnitt gezeigten Identnummern der Objekte sind ausschließlich auf dieses Beispiel bezogen und können in eigenen Anlagen abweichen.

## Der Bahnübergang

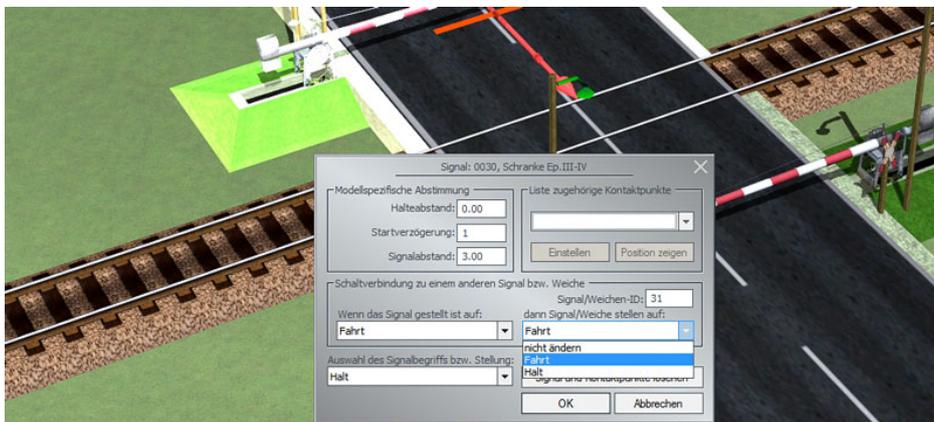
Im ersten Schritt verbinden wir den Taster mit einer der beiden Schranken, in diesem Fall die hinter dem Gleis. Nachdem die Objekteigenschaften des Tasters geöffnet sind, geben wir in der Schaltverbindung die ID der Schranke ein und setzen die Bedingungen auf folgende Optionen:

- Wenn "Schliessen" dann "Halt"
- Wenn "Oeffnen" dann "Fahrt"



Nachdem wir den Dialog mit "OK" geschlossen haben, wenden wir uns der ersten Schranke zu und öffnen dort die Objekteigenschaften. Jetzt geben wir in der Schaltverbindung die ID der *zweiten Schranke* ein und setzen die Bedingungen auf folgende Optionen:

- Wenn "Halt" dann "Halt"
- Wenn "Fahrt" dann "Fahrt"

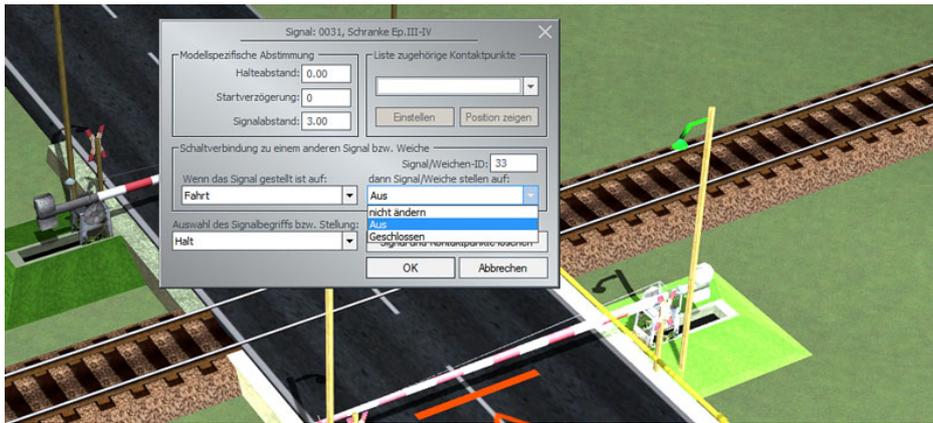


Im letzten Schritt verbinden wir die *zweite Schranke* mit dem BÜ-Tischfeld. Wir öffnen also die Objekteigenschaften der Schranke, geben in der Schaltverbindung die ID des Tischfelds ein und setzen die Bedingungen auf folgende Optionen:

- Wenn "Halt" dann "Geschlossen"
- Wenn "Fahrt" dann "Aus"

## Der Bahnübergang

---



Wir beenden den Dialog mit "OK", verlassen den 3D-Editmodus und überprüfen unser Ergebnis mit einem Funktionstest. Anzumerken wäre noch, dass dieser Aufbau mit dem Bahnübergangs-Wizzard kompatibel ist.

## Das SpDrS60 und LUA

---

Seit EEP10.2 und den Plugin's 1 + 2 steht uns unter anderem die Skriptsprache LUA zur Verfügung. LUA ermöglicht uns die Automatisierung von EEP-Anlagen durch die Erstellung von Skriptdateien. Dieses Verfahren ersetzt die in den Vorgängerversionen benötigten "Schaltkreise". Um einige der Möglichkeiten des Pultes sinnvoll zu nutzen, bedienen wir uns dieser Schnittstelle. Dazu sind dem Basisset einige LUA-Scripte beigegeben, die nach der Installation im EEP LUA-Ordner zu finden sind.

- SpDrS60\_Lib\_RL2.lua - die elementare Bibliothek des SpDrS60
- SpDrS60\_SigPara\_RL2.lua - eine Katalogdatei mit Signalparametern (wird fortlaufend erweitert)
- SpDrS60\_AnL3\_MainScript.lua - das Schnittstellenskript für die Usereingaben

Das für den User wichtigste Script des Sets finden wir unter dem Namen "SpDrS60\_AnL3\_MainScript.lua". Dabei handelt es sich um eine "Vorlage", deren Inhalt mithilfe eines externen Editors (z.B. Notepad++) via "Copy and Paste" in den LUA-Editor von EEP übertragen werden.

### Das MainScript

In dieser Beschreibung beschränken wir uns auf die coderelevanten Zeilen und vernachlässigen diejenigen mit den Kommentaren.

- Path = ("..\LUA\")
- SpDrS60\_Lib\_RL2 = require("SpDrS60\_Lib\_RL2");
- SpDrS60\_SigPara\_RL2 = require("SpDrS60\_SigPara\_RL2");

In den ersten drei Zeilen weisen wir der Variablen "Path" den Dateipfad zum EEP Lua-Ordner zu und binden die Bibliothek und eine Signal-Katalogdatei ein. **Der Inhalt dieser Einträge darf nicht verändert werden!**

### Die Parametertabellen

Im nächsten Abschnitt finden wir vier Tabellen. Sie dienen zur Aufnahme von Objektparametern. Die eingesetzten Daten ( *Objekt-IDs* ) ermöglichen die Kommunikation verschiedener Objekte mit den entsprechenden Symbolen und Tastern auf dem Pult. Diese werden durch Bibliotheksfunktionen verarbeitet.

Werden Tabellen im Skript nicht benötigt, sollten sie nicht gelöscht werden. In diesem Fall bleiben die geschweiften Klammern einfach leer.

### Die Tabellen usedEKWs\_SpDr - usedDKWs\_SpDr

- usedEKWs\_SpDr = {}
- usedDKWs\_SpDr = {}

Die Tabellen "usedEKWs\_SpDr" und "usedDKWs\_SpDr" stellen Parameter für die Verbindung von EKW- und DKW-Gleisobjekten mit dem Pult zur Verfügung. Die Tabellen werden mit Datensätzen gefüllt, die wiederum Tabellen sind. Sie werden auf folgende Weise eingegeben ( hier im Beispiel DKW-Gleisobjekte ):

```
usedDKWs_SpDr = { {7,25,32} , {9,27,35} }
```

Ein Datensatz innerhalb der Tabellen *usedEKWs\_SpDr* und *usedDKWs\_SpDr* besteht immer aus **drei** Parametern, eingerahmt von einer öffnenden und einer schließenden, geschweiften Klammer und untereinander mit einem Komma getrennt ( rot dargestellt ). Mehrere Datensätze werden ebenfalls durch ein Komma getrennt.

Die Parameter sind die Objekt-IDs und müssen in folgender Reihenfolge eingetragen werden:

- Pos. 1 - {X,X,X} = die ID des EKW bzw. DKW-Gleisobjekts
- Pos. 2 - {X,X,X} = die ID des zugehörigen EKW bzw. DKW-Tischfelds
- Pos. 3 - {X,X,X} = die ID des zugehörigen EKW bzw. DKW-Tasters

Für unser Beispiel {7,25,32} würde das also bedeuten, dass DKW-Gleisobjekt ID7 und DKW-Tischfeld ID25 sowie DKW-Taster ID32 schaltungstechnisch zusammenwirken.

## Das SpDrS60 und LUA

---

### Die Tabelle - *usedGLBSigs\_SpDr*

Die Tabelle "*usedGLBSigs\_SpDr*" stellt Objekt-IDs für die Verbindung von Fahrstraßen-Startsignalen und Gleisbesetzmeldern auf dem Pult bereit. Dabei wird durch die verarbeitenden Funktionen eine abschnittsdeckende Besetztmeldung gewährleistet. Die abschnittsdeckende Besetztmeldung erkennt auf dem Gleis verbliebene Zugteile, z.B. einen vom ausfahrenden Zug abgekuppelten Waggon.

- `usedGLBSigs_SpDr = {}`

Die Tabelle wird mit Datensätzen gefüllt, die wiederum Tabellen sind. Sie werden auf folgende Weise eingegeben:

```
usedGLBSigs_SpDr = { {16,48}, {27,35} }
```

Ein Datensatz innerhalb der Tabelle *usedGLBSigs\_SpDr* besteht immer aus zwei Parametern. Die Objekte-IDs müssen in folgender Reihenfolge eingetragen werden:

- Pos. 1 - {X,X} = die ID des Fahrstraßen-Startsignals
- Pos. 2 - {X,X} = die ID des zugehörigen Gleisbesetzmelders

Im Beispiel {16,48} würde das bedeuten, dass Fahrstraßen-Startsignal ID16 und der Gleisbesetzmelder ID48 zusammenwirken.

### Die Tabelle - *SigDtcTbl\_SpDrS60*

Die Tabelle "*SigDtcTbl\_SpDrS60*" stellt Objekt-IDs und Verweise für eine Signalverkettung bereit. Dabei werden Einfahrt-, Ausfahrt- und erstes Block-Signal(e) miteinander "verkettet". Diese Verbindung von Signalen ermöglicht die korrekte Darstellung von HpVr-Begriffen in Abhängigkeit von nachgeschalteten Signalen sowie dessen korrekte Anzeige auf dem Pult.

- `SigDtcTbl_SpDrS60 = { }`

Von allen Tabellen im MainScript ist *SigDtcTbl\_SpDrS60* die komplexeste. Da wir uns im Abschnitt "Signalverknüpfung" noch intensiver mit dieser Thematik auseinandersetzen werden, möchte ich hier lediglich einen Auszug zu deren Verwendung zeigen. Er stammt aus der Beispielanlage "SpDrS60\_DemoAnlage\_RL2.anl3", die im Set enthalten ist. In dieser Anlage kann die Funktionsweise des Features leicht nachvollzogen werden.

```
SigDtcTbl_SpDrS60 = {  
{ SigId = 17, SigType = 0, SigChar = HpSig_E_V40_Vs_V60_69_V8, HpDtclId = 83, VrDtclId = 87, HpButtonId = 96, Hp2ButtonId = 98, ShButtonId = 0, CnctSigId = {8, 10, 11}, },  
{ SigId = 8, SigType = 1, SigChar = HpSig_A_V40_Vs_69_V7, HpDtclId = 79, VrDtclId = 70, HpButtonId = 97, Hp2ButtonId = 0, ShButtonId = 0, CnctSigId = {14}, },  
{ SigId = 10, SigType = 1, SigChar = HpSig_A_V40_Vs_69_V7, HpDtclId = 71, VrDtclId = 80, HpButtonId = 91, Hp2ButtonId = 0, ShButtonId = 106, CnctSigId = {14}, },  
{ SigId = 11, SigType = 1, SigChar = HpSig_A_V40_Vs_69_V7, HpDtclId = 72, VrDtclId = 81, HpButtonId = 93, Hp2ButtonId = 0, ShButtonId = 104, CnctSigId = {14}, },  
{ SigId = 14, SigType = 2, SigChar = HP01_Form_oVsig, HpDtclId = 84, VrDtclId = 0, HpButtonId = 0, Hp2ButtonId = 0, ShButtonId = 0, CnctSigId = {0}, },  
}
```

### Die Stringvariable - *SubFuncFileName\_SpDr*

In dieser Zeile wird der Stringvariablen "*SubFuncFileName\_SpDr*" ein Default-Dateiname zugewiesen.

- `SubFuncFileName_SpDr = "SpDrS60_TmpFunctionFile.lua"`

Unter diesem Namen wird beim Öffnen einer Anlage oder beim anklicken des LUA Editor-Buttons "Script neu laden" eine Datei generiert, die alle für die aktuelle Anlagen-Stellpultkonfiguration benötigten Funktionen enthält. Zusätzlich wird eine weitere Datei mit der gleichen Bezeichnung erzeugt, jedoch ergänzt um die Dateierweiterung ".txt".

Beispiel: "SpDrS60\_TmpFunctionFile.lua" und "SpDrS60\_TmpFunctionFile.lua.txt"

Beide Dateien finden wir im EEP Lua-Ordner. Mit dem Zweck der TXT-Datei werden wir uns an entsprechender Stelle genauer befassen.

Bei jedem Öffnen einer SpDr-Anlage werden die beiden Dateien "*SpDrS60\_TmpFunctionFile*" neu geschrieben. Das bedeutet, dass die existierenden Dateien durch die der neu geladenen SpDr-Anlage überschrieben werden. Das stellt jedoch kein Problem dar. In EEP kann immer nur eine Anlage aktiv sein und so greift die geladene Anlage immer auf die korrekte Datei zu.

## Das SpDrS60 und LUA

---

### Die Funktion - *setCntrObjFnc\_SpDr()*

Alle in den Tabellen gesammelten Daten und der Default-Dateiname werden hier zentral an die Funktion übergeben. In ihr werden die Daten nun aufbereitet und in einer Funktionsdatei (SpDrS60\_TmpFunctionFile.lua) mit allen relevanten Einträgen im EEP LUA-Ordner hinterlegt.

- `setCntrObjFnc_SpDr(usedEKWs_SpDr, usedDKWs_SpDr, usedGLBSigs_SpDr, SigDtcTbl_SpDrS60, SubFuncFileName_SpDr)`

Das bedeutet, das an diesem Funktionsaufruf keine Änderung vorgenommen werden darf. Das gleiche gilt für alle Tabellen- und Variablennamen in diesem Skript, die im Bezug zum SpDrS60 stehen.

Nachdem *setCntrObjFnc\_SpDr* ausgeführt wurde, binden wir die nun vorhandene Funktionsdatei mit "dofile" ein.

- `dofile(Path..SubFuncFileName_SpDr)`

Der *dofile*-Aufruf muss in der Reihenfolge immer unter *setCntrObjFnc\_SpDr* stehen, da die Funktionsdatei vorher noch nicht existiert. Also die Reihenfolge nicht verändern.

### Die Funktion - *SpDr\_DtcRun()*

Einige Objekte des SpDrS60 benötigen für eine einwandfreie Funktion zyklische Abfragen. Diese Abfragen werden in der Funktionsdatei gebündelt und sind für den User nicht sichtbar. Um diese Abfragen anzustoßen, benötigen wir die Funktion

- `SpDr_DtcRun()`

Diese kopieren wir in den Rumpf der Funktion *EEPMain()*. Die Funktion *EEPMain()* wird standardmäßig 5 mal pro Sekunde durchlaufen. Somit wird auch *SpDr\_DtcRun()* in diesem Zyklus aufgerufen.

Damit haben wir die für das Basisset erforderlichen Codezeilen behandelt. In den Demoanlagen finden wir schon aufbereiteten Code, den wir uns jederzeit kopieren und nach entsprechenden Änderungen in eigenen Anlagen verwenden können. Bleibt nur noch das Einfügen des Scriptcodes in EEP.

### Einfügen des Scriptcodes in den EEP LUA-Editor

#### In eine neue Anlage

Das ist der einfachste Weg. Wir markieren den Inhalt des Skripts im externen Editor mit der Tastenkombination "Strg+A" und kopieren dann mit "Strg+C" alles in die "Zwischenablage".

Jetzt wechseln wir zum EEP LUA-Editor und markieren dort alles mit "Strg+A". Wenn wir jetzt die Tastenkombination "Strg+V" drücken, wird der markierte Inhalt durch den Inhalt der "Zwischenablage" ersetzt. Unser Skriptcode befindet sich jetzt im EEP Lua-Editor. Nun brauchen wir nur noch den Button "Skript neu laden" anzuklicken und danach die Anlage zu speichern.

#### In eine bestehende Anlage

Wenn die Anlage schon LUA-Code enthält, der über das EPP-Standardskript hinaus geht, können wir nur Teile des MainScript kopieren.

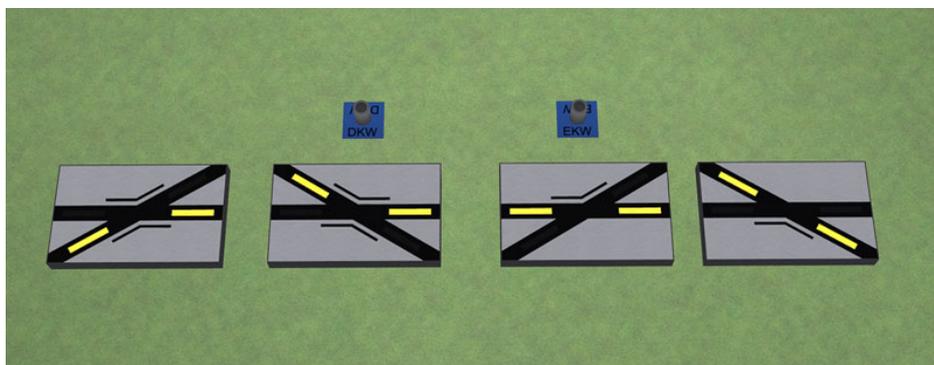
Dazu markieren wir den Bereich zwischen den Zeilen "-- BEGINN BLOCK <<<" und "-- ENDE BLOCK <<<" und kopieren ihn in unseren EEP LUA-Editor oberhalb der Funktion "EEPmain()".

Dann kopieren wir noch die Zeile mit dem Eintrag "SpDr\_DtcRun()" aus dem MainScript in die Funktion "EEPmain()".

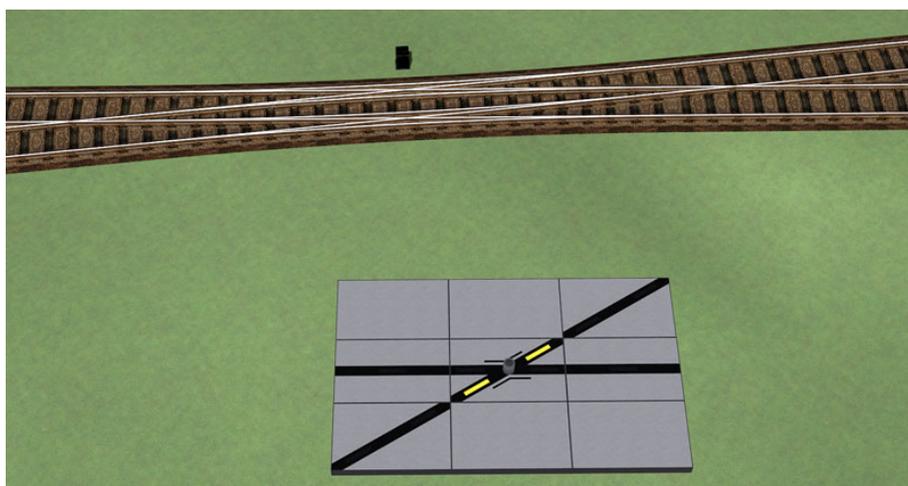
Jetzt den Button "Skript neu laden" anklicken und danach die Anlage speichern.

## Die DKW und EKW Gleisobjekte

Für den Einbau von DKWs und EKW finden wir in den Gleisobjekten einige Typen, die mit einem Klick eingesetzt werden können. Der Vorteil ist dabei, dass die Kreuzungsweiche aus einem Stück besteht und nicht aus vier oder zwei Weichen zusammgebaut werden muss. In dieser Beschreibung werden wir also Gleisobjekte verwenden.

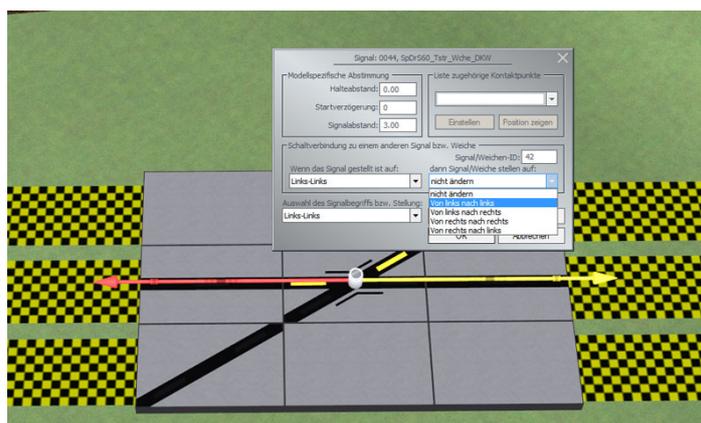


Das Set enthält zur Steuerung von EKWs/DKW jeweils zwei Tischfelder in rechter und linker Ausführung und passende Taster dazu. Die Taster lassen sich an der Sockelbeschriftung unterscheiden. Wir werden jetzt eine DKW mit dem Pult verbinden.



Dazu nutzen wir einen Beispielaufbau, wie er im Bild zu sehen ist. Er besteht aus einer DKW in linker Ausführung und einem kleinen Stellpultausschnitt. Im ersten Schritt öffnen wir die Objekteigenschaften des Tasters. Wir geben unter Schaltverbindungen die Objekt-ID der DKW ein und verknüpfen die "Wenn..Dann" Bedingungen mit folgenden Optionen:

- Wenn "Links-Links" dann "Von links nach links"
- Wenn "Links-Rechts" dann "Von links nach rechts"
- Wenn "Rechts-Rechts" dann "Von rechts nach rechts"
- Wenn "Rechts-Links" dann "Von rechts nach links"



Für den nächsten Schritt, die Lua-Anbindung, benötigen wir die Objekt-IDs der DKW, des Tischfeldes und des Tasters.

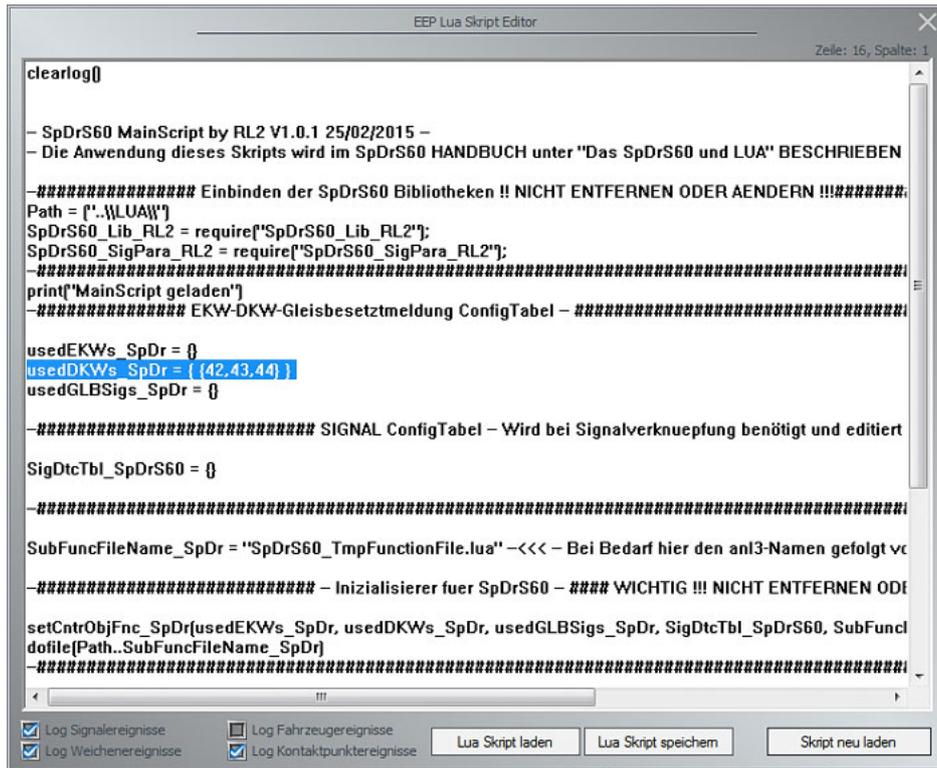
## Die DKW und EKW Gleisobjekte

Im Beispiel ergeben sich folgende Ids: die DKW hat die ID 42, das Tischfeld hat die ID 43 und der Taster hat die ID 44

### Die Lua-Anbindung

Da Gleisobjekte in EEP keinen Dialog für Schaltverbindungen besitzen, ist es nicht möglich, eine direkte Rückmeldung an das SpDrS60 zu senden. Hier kommt nun das Lua-Plugin zum Einsatz. Wir haben im Abschnitt "Das SpDrS60 und LUA" den Inhalt des MainScripts behandelt. Damit werden wir uns jetzt die Möglichkeit der Rückmeldung schaffen.

Wir öffnen den EEP LUA-Editor und suchen den Eintrag "usedDKWs\_SpDr = {}" auf. In die geschweiften Klammern tragen wir die Id's unserer Objekte ein, hier bezogen auf das Beispiel: usedDKWs\_SpDr = { {42,43,44} }



```
clearlog()

-- SpDrS60 MainScript by RL2 V1.0.1 25/02/2015 --
-- Die Anwendung dieses Skripts wird im SpDrS60 HANDBUCH unter "Das SpDrS60 und LUA" BESCHRIEBEN

----- Einbinden der SpDrS60 Bibliotheken !! NICHT ENTFERNEN ODER AENDERN !!-----
Path = [".\LUA\"]
SpDrS60_Lib_RL2 = require["SpDrS60_Lib_RL2"];
SpDrS60_SigPara_RL2 = require["SpDrS60_SigPara_RL2"];
-----
print["MainScript geladen"]
----- EKW-DKW-Gleisbesetztmeldung ConfigTabel -----

usedEKWs_SpDr = {}
usedDKWs_SpDr = { {42,43,44} }
usedGLBSigs_SpDr = {}

----- SIGNAL ConfigTabel -- Wird bei Signalverknuepfung benötigt und editiert

SigDtcTbl_SpDrS60 = {}

-----

SubFuncFileName_SpDr = "SpDrS60_TmpFunctionFile.lua" --<<< -- Bei Bedarf hier den anI3-Namen gefolgt v

----- Initalisierer fuer SpDrS60 -- ### WICHTIG !!! NICHT ENTFERNEN ODI

setCntrObjFunc_SpDr(usedEKWs_SpDr, usedDKWs_SpDr, usedGLBSigs_SpDr, SigDtcTbl_SpDrS60, SubFunc
dofile[Path..SubFuncFileName_SpDr]
-----

Log Signalereignisse
Log Fahrzeugereignisse
Log Weichenergebnisse
Log Kontaktpunktereignisse

Lua Skript laden
Lua Skript speichern
Skript neu laden
```

Mit einem Klick auf den Button "Skript neu laden" schließen wir den Editor und speichern die Anlage. Um die Funktion zu prüfen, klicken wir einige Male auf den DKW-Taster. Das Tischfeld sollte jetzt die wechselnde Lage der DKW anzeigen. Für den Test der Rückmeldung klicken wir auf die Weichenlaterne der DKW. Das Tischfeld sollte ebenfalls die wechselnde Lage wiedergeben.

Für die Verbindung einer EKW wäre die Vorgehensweise die gleiche, allerdings dann mit "usedEKWs\_SpDr = {}".

Wie weiter oben im Abschnitt "Das SpDrS60 und LUA" schon angemerkt werden zwei temporäre Dateien im EEP Lua-Ordner erzeugt (*SpDrS60\_TmpFunctionFile*). Eine besitzt die Endung ".txt", die wir durch einen Doppelklick öffnen können. Dort finden wir verschiedene Einträge, die jeweils durch eine horizontale Linie begrenzt werden. Hier ein Beispielausschnitt:

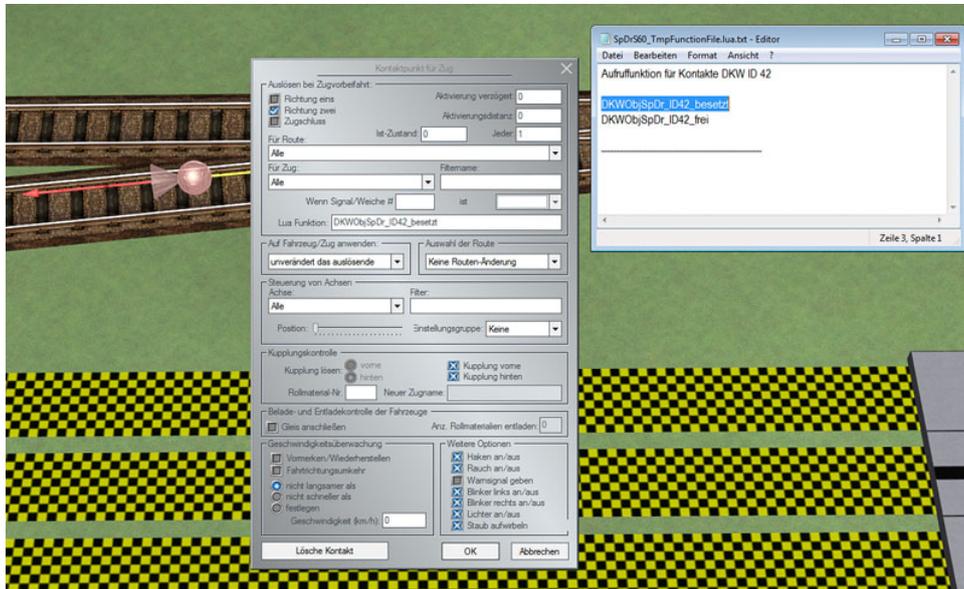
Aufruffunktion für Kontakte DKW ID 42

```
DKWObjSpDr_ID42_besetzt
DKWObjSpDr_ID42_frei
```

Jeder Abschnitt enthält Aufruffunktionen zu denen in "usedDKWs\_SpDr" oder "usedEKWs\_SpDr" definierten Kreuzungsweichen. Damit ist es möglich, bei Bedarf eine lokale Gleisbesetztmeldung der Kreuzungen zu installieren. Um diese Möglichkeit zu nutzen, positionieren wir in jedem Gleiszweig einen Fahrzeug-Kontaktpunkt in Einfahr- und in Ausfahrtrichtung.

## Die DKW und EKW Gleisobjekte

Wir beginnen mit der Einfahrtrichtung. Nachdem wir die Objekteigenschaften des Kontaktpunktes geöffnet haben, stellen wir die Wirkrichtung ein. Dann holen wir uns das Editorfenster mit der geöffneten Txt-Datei nach vorne und kopieren aus dem passenden Abschnitt für unsere Kreuzung den Eintrag mit der Erweiterung "\_besetzt". Dabei ist darauf zu achten, dass wirklich nur der Aufruf der Funktion markiert ist. Mitkopierte Leerzeichen verursachen später Probleme beim Funktionsaufruf.

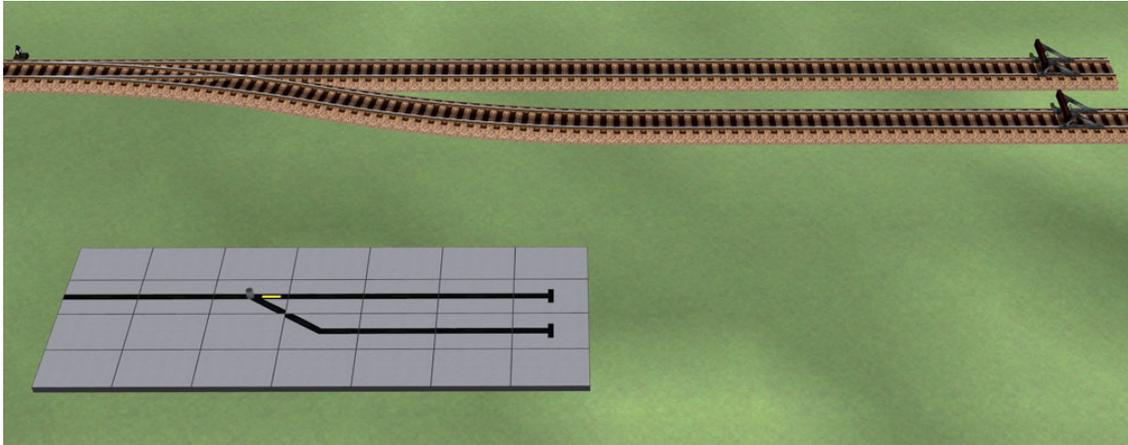


Nun klicken wir im Dialogfeld des Kontaktpunktes in die Zeile "Lua-Funktion". Mit "Strg+V" wird der kopierte Aufruf in die Eingabe eingefügt. Wir schließen das Dialogfeld mit "OK". Um uns die Arbeit zu erleichtern, nutzen wir jetzt die Kontakt-Kopierfunktion. Damit statten wir die verbliebenen drei Gleiszweige aus und passen lediglich die Wirkrichtung an.

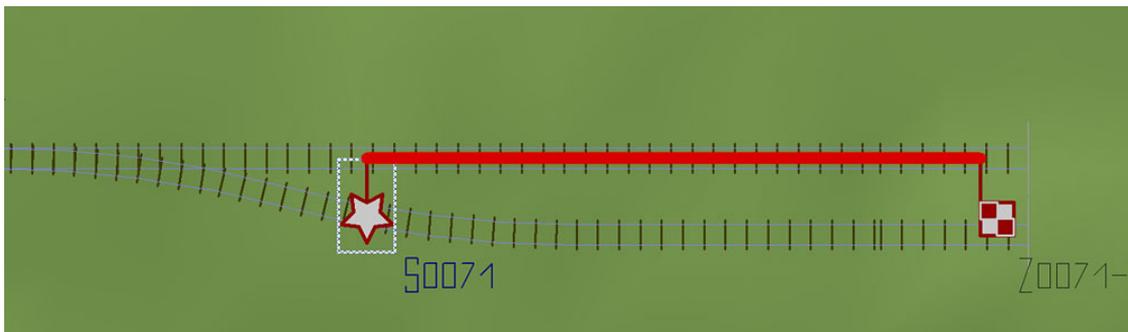
Bei den Ausfahrkontakten verfahren wir nach dem gleichen Schema, verwenden jetzt aber den Eintrag mit der Erweiterung "\_frei" und setzen im Dialogfeld der Kontakte zusätzlich die Option "Zugschluss". Damit ist unsere lokale Besetztmeldung komplett.

## Die Abschnitts-Gleisbesetzungsmeldung

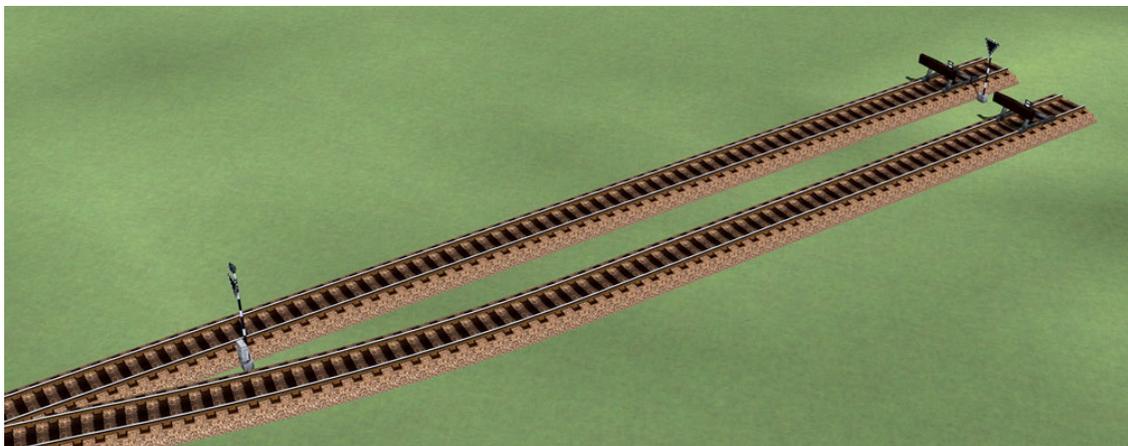
Im Thema "*Die einfache Gleisbesetzungsmeldung*" wurde bereits darauf hingewiesen, dass diese Art der Besetzungsmeldung vorwiegend für Blockabschnitte und Gleise, die von Zügen ohne Zerlegung durchfahren werden, geeignet ist. Bei Stumpf- oder Kopfgleisen und Gleisen in Rangierbereichen ist sie jedoch lückenhaft und somit unbrauchbar. Zur Lösung dieses Problems werden wir eine Kombination aus LUA und dem Fahrstraßen-Plugin einsetzen. Das Bild zeigt die Beispielsituation.



Für den ersten Schritt begeben wir uns in das 2D-Planfenster und wechseln in den Fahrstraßeneditor. Wir legen für das erste, in unserem Beispiel das gerade Gleis, ein Start- und ein Zielsignal an.

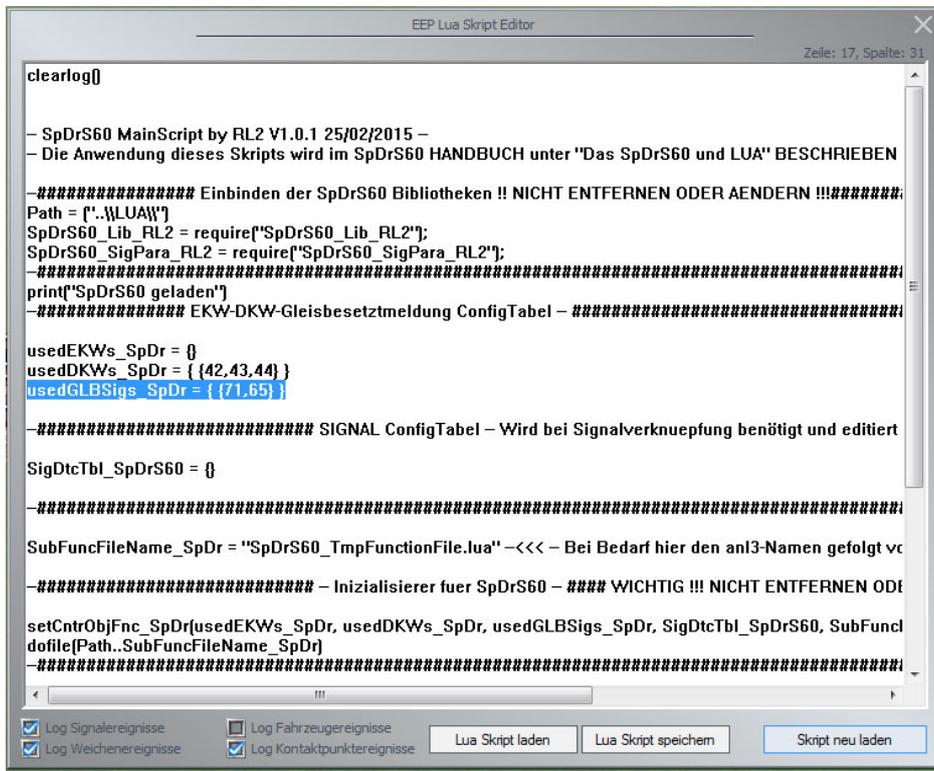


Nachdem wir die Fahrstraße eingerichtet ( durch einen Klick auf Start- und Zielsignal ) und uns die Startsignal-Id gemerkt haben, wechseln wir wieder in den 3D-Modus. Das folgende Bild zeigt das Ergebnis.



## Die Abschnitts-Gleisbesetzungsmeldung

Im folgenden Schritt werden wir unser Lua-Skript anpassen. Dazu öffnen wir den EEP Lua-Editor und suchen den Eintrag "usedGLBSigs\_SpDr" auf. Dort tragen wir die Startsignal-Id sowie die zugehörige Gleismelder-Id ein, in diesem Beispiel Id 71 und Id 65. Im Bild ist es der markierte Eintrag.



```
clearlog()

-- SpDrS60 MainScript by RL2 V1.0.1 25/02/2015 --
-- Die Anwendung dieses Skripts wird im SpDrS60 HANDBUCH unter "Das SpDrS60 und LUA" BESCHRIEBEN

##### Einbinden der SpDrS60 Bibliotheken !! NICHT ENTFERNEN ODER AENDERN !!!#####
Path = {".\\LUA\\"}
SpDrS60_Lib_RL2 = require("SpDrS60_Lib_RL2");
SpDrS60_SigPara_RL2 = require("SpDrS60_SigPara_RL2");
#####
print("SpDrS60 geladen")
##### EKW-DKW-Gleisbesetzungsmeldung ConfigTabel - #####

usedEKWs_SpDr = {}
usedDKWs_SpDr = { {42,43,44} }
usedGLBSigs_SpDr = { {71,65} }

##### SIGNAL ConfigTabel - Wird bei Signalverknuepfung benötigt und editiert

SigDtcTbl_SpDrS60 = {}

#####

SubFuncFileName_SpDr = "SpDrS60_TmpFunctionFile.lua" --<<< - Bei Bedarf hier den anI3-Namen gefolgt vc

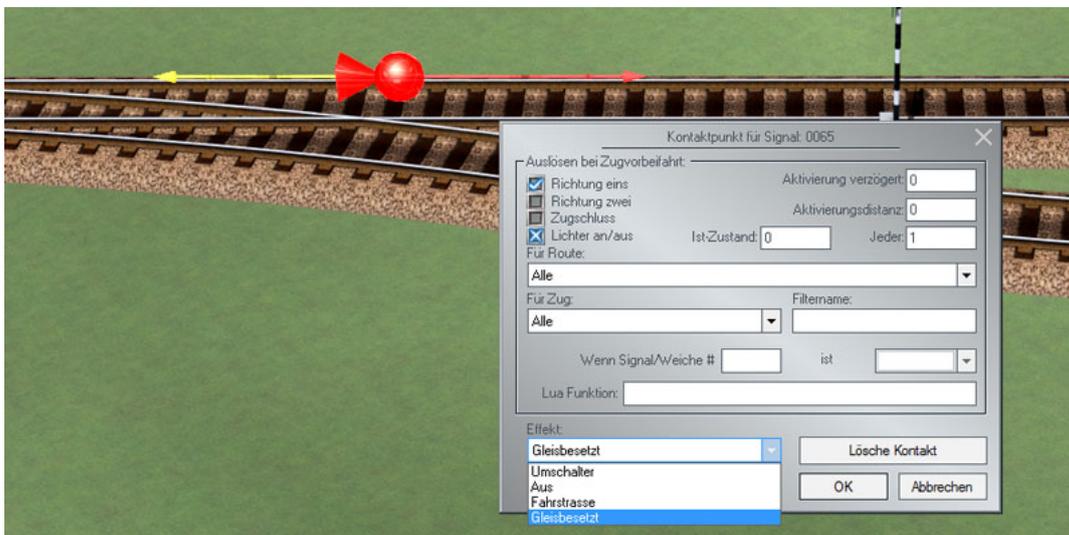
##### - Initalisierer fuer SpDrS60 - ##### WICHTIG !!! NICHT ENTFERNEN ODI

setCntrObjFnc_SpDr(usedEKWs_SpDr, usedDKWs_SpDr, usedGLBSigs_SpDr, SigDtcTbl_SpDrS60, SubFuncFile
dofile(Path..SubFuncFileName_SpDr)
#####
```

Durch schließen des Editors mit Klick auf "Skript neu laden" werden unsere Funktionsdateien im Hintergrund neu geschrieben ("SpDrS60\_TmpFunctionFile").

### Die Kontakte

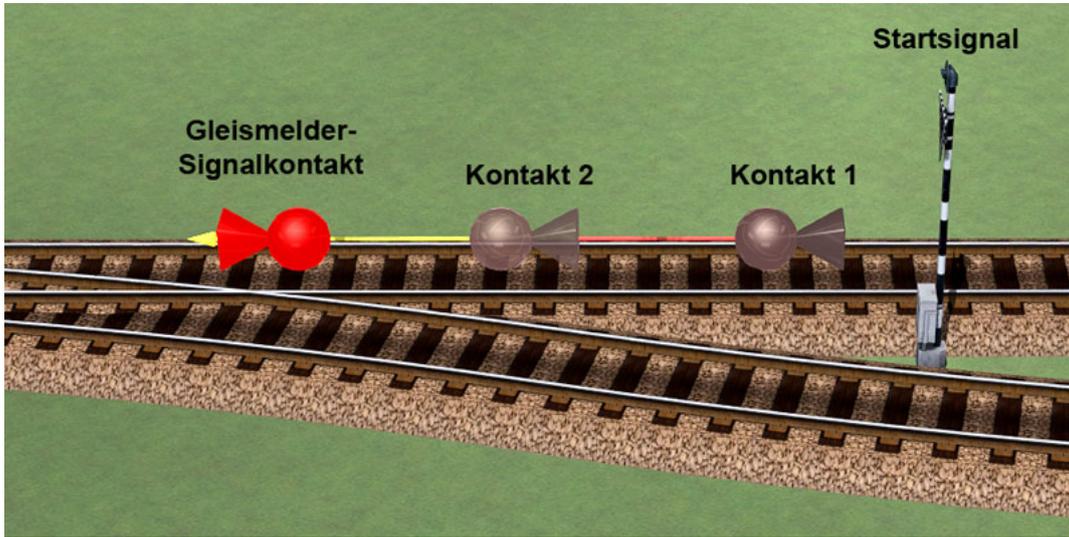
Nun wenden wir uns dem Setzen und Konfigurieren der Kontakte zu. Dazu rufen wir das Kontextmenü des Gleismelders auf und wählen die Option "Kontaktpunkt für Signal". Der Kontaktpunkt wird kurz hinter der Weiche im Stumpfgleis platziert (siehe Bild).



Die Wirkrichtung zeigt zum Stumpfgleis und als Effekt setzen wir "Gleisbesetzt" ein. Durch einen Klick auf "OK" beenden wir den Dialog.

# Die Abschnitts-Gleisbesetztmeldung

Jetzt folgen zwei Fahrzeug-Kontaktpunkte. Das Bild zeigt die Anordnung der Kontakte, ihre Wirkrichtung und ihre Reihenfolge im Bezug zum Fahrstraßen-Startsignal.

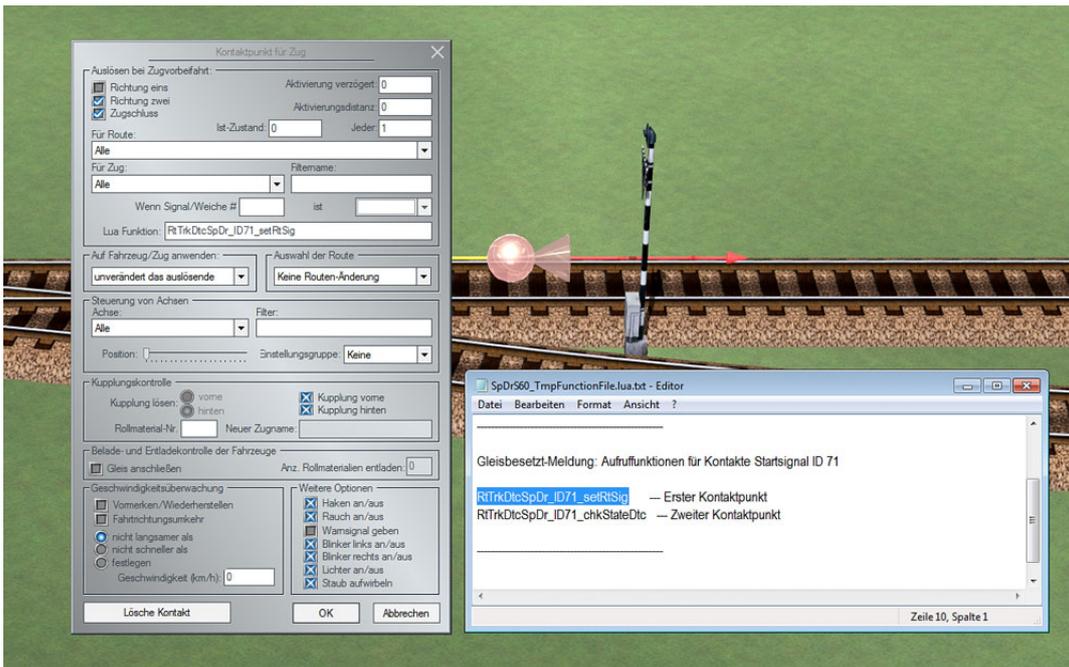


Für den nächsten Schritt müssen wir die im EEP Lua-Ordner erzeugte Txt-Datei (*SpDrS60\_TmpFunctionFile*) durch einen Doppelklick öffnen und den Eintrag für unser Startsignal aufsuchen. Hier der Auszug für das Beispiel:

Gleisbesetzt-Meldung: Aufruffunktionen für Kontakte Startsignal ID 71

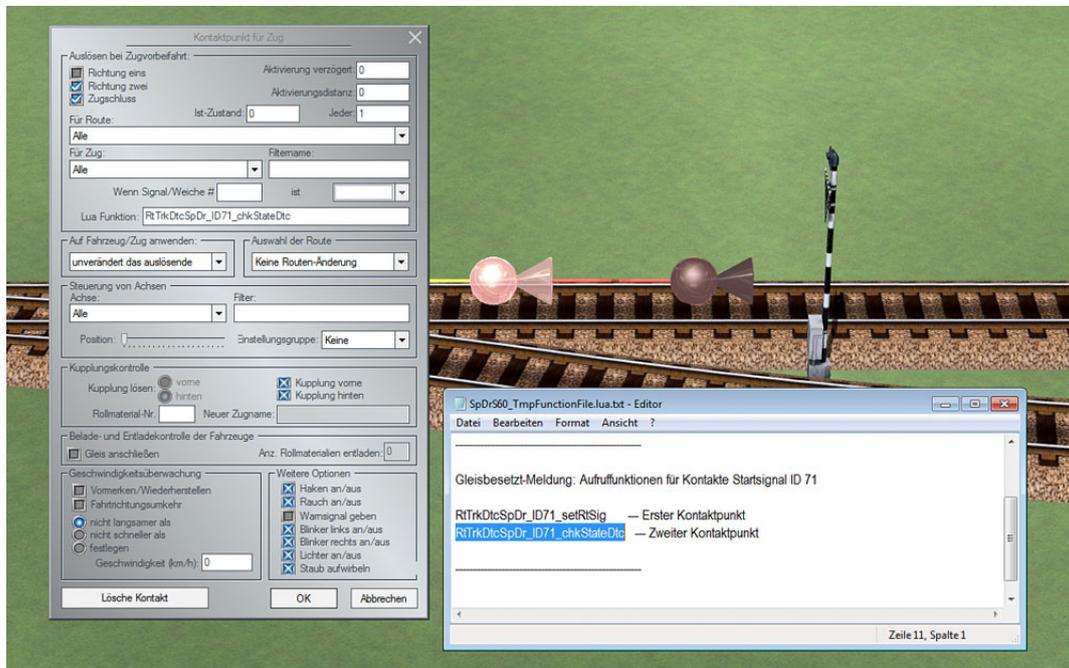
```
RtTrkDtcSpDr_ID71_setRtSig    --- Erster Kontaktpunkt  
RtTrkDtcSpDr_ID71_chkStateDtc --- Zweiter Kontaktpunkt
```

Wir öffnen die Objekteigenschaften von Kontakt 1 und kopieren den Eintrag für den ersten Kontaktpunkt aus der Txt-Datei in die Zeile "Lua Funktion" und bestimmen die Wirkrichtung. **Wichtig ist der Haken bei "Zugschluss"!**



## Die Abschnitts-Gleisbesetzmeldung

Mit dem Kontakt 2 verfahren wir auf die gleiche Weise, jetzt kopieren wir allerdings den Eintrag für den zweiten Kontaktpunkt aus der Txt-Datei in die Zeile "Lua Funktion" und bestimmen die Wirkrichtung. **Wichtig ist der Haken bei "Zugschluss"!**



Unsere abschnittsdeckende Gleisbesetzmeldung für das Stumpfgleis ist nun komplett. Für eine Installation in einem durchgehenden Gleis müssen wir die gleiche Kontaktkonfiguration auf der anderen Seite zwischen Zielsignal und Weiche anordnen. Dazu leistet uns die Kontakt-Kopierfunktion gute Dienste und minimiert den Aufwand. Wir müssen dann lediglich die Wirkrichtung anpassen. Beispiele beider Varianten sind in den Demoanlagen zu finden.

## Das Inventar

Diese Inventarliste gibt Ihnen eine Übersicht der im Basisset enthaltenen Tischfelder, Taster und Piktogramme. Soweit vorhanden, sind die Funktionen in der Reihenfolge ihrer Position in den Objekten beschrieben.

Objekt	Beschreibung	Funktionen
<b>Gerades Gleis</b>		
SpDrS60_Grd_1E_LF	Leerfeld - 1 Einheit	---
SpDrS60_Grd_1E_ME_2L	Gleismelder - 1 Einheit - 1 Licht/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_1E_ME	Gleismelder - 1 Einheit - 2 Lichter/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_1E_ME_Vert	Gleismelder - 1 Einheit - Vertikal	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_2E_LF	Leerfeld - 2 Einheiten	---
SpDrS60_Grd_2E_ME_2L	Gleismelder - 2 Einheiten - 1 Licht/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_2E_ME	Gleismelder - 2 Einheiten - 2 Lichter/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_2E_ME_Vert	Gleismelder - 2 Einheiten - Vertikal	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_3E_LF	Leerfeld - 3 Einheiten	---
SpDrS60_Grd_3E_ME_2L	Gleismelder - 3 Einheiten - 1 Licht/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_3E_ME	Gleismelder - 3 Einheiten - 2 Lichter/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_3E_ME_Vert	Gleismelder - 3 Einheiten - Vertikal	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_4E_LF	Leerfeld 4 Einheiten	---
SpDrS60_Grd_4E_ME_2L	Gleismelder - 4 Einheiten - 1 Licht/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_4E_ME	Gleismelder - 4 Einheiten - 2 Lichter/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_5E_LF	Leerfeld - 5 Einheiten	---
SpDrS60_Grd_5E_ME_2L	Gleismelder - 5 Einheiten - 1 Licht/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_5E_ME	Gleismelder - 5 Einheiten - 2 Lichter/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_6E_LF	Leerfeld - 6 Einheiten	---
SpDrS60_Grd_6E_ME_2L	Gleismelder - 6 Einheiten - 1 Licht/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_6E_ME	Gleismelder - 6 Einheiten - 2 Lichter/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_7E_LF	Leerfeld - 7 Einheiten	Aus - Ein
SpDrS60_Grd_7E_ME_2L	Gleismelder - 7 Einheiten - 1 Licht/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_7E_ME	Gleismelder - 7 Einheiten - 2 Lichter/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_8E_LF	Leerfeld - 8 Einheiten	---
SpDrS60_Grd_8E_ME_2L	Gleismelder - 8 Einheiten - 1 Licht/Feld	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Grd_8E_ME	Gleismelder - 8 Einheiten - 2 Lichter/Feld	Aus - Fahrstraße - Gleisbesetzt
<b>Signalsymbole</b>		
SpDrS60_Sig_A_E_Ergz	Ausfahrt- oder Einfahrtsignal-Ergänzung	Halt - Fahrt - Langsamfahrt
SpDrS60_Sig_HP_BlK	Blocksignal	Halt - Fahrt
SpDrS60_Sig_HP	Hauptsignal	Halt - Fahrt - Langsamfahrt
SpDrS60_Sig_HP_Sh2	Haupt + Rangiersignal Kombi	Halt - Fahrt - Langsamfahrt - Rangiersignal
SpDrS60_Sig_Hp_Sh_Ergz	Haupt + Rangiersignal Kombi-Ergänzung	Halt - Fahrt - Langsamfahrt - Rangiersignal

## Das Inventar

Objekt	Beschreibung	Funktionen
SpDrS60_Sig_Ls_Sh	Lichtsperrsignal	Halt - Langsamfahrt
SpDrS60_Sig_Merker_RL2	Merkersignal - Leerfeld mit Status-LED	Aus - Ein
SpDrS60_Sig_Relaiss_RL2	Relaissignal - Fahrstraßen-Hilfssignal	Aus - Ein
SpDrS60_Sig_VR	Vorsignal	Halt erw. - Fahrt erw. - Langsamfahrt erw.
<b>Streckenfelder</b>		
SpDrS60_Strk_Bue	Strecke Bahnübergang	Aus - Geschlossen
SpDrS60_Strk_diag_Li	Strecke diagonal Links	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Strk_diag_Re	Strecke diagonal Rechts	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Strk_Krve_Li_90	Strecke Kurve Links - 90 Grad	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Strk_Krve_Li	Strecke Kurve Links - 45 Grad	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Strk_Krve_Re_90	Strecke Kurve Rechts - 90 Grad	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Strk_Krve_Re	Strecke Kurve Rechts - 45 Grad	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Strk_Krzng_diag	Strecke Kreuzung Diagonal	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Strk_Krzng_Li	Strecke Kreuzung Links Gerade-Diagonal	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Strk_Krzng_Re	Strecke Kreuzung Rechts Gerade-Diagonal	Aus - Fahrstraße - Gleisbesetzt
SpDrS60_Strk_Prell	Strecke Prellbock	-----
<b>Taster</b>		
SpDrS60_Tstr_Bue	Taster Bahnübergang	Schliessen - Oeffnen
SpDrS60_Tstr_Dweg	Taster D-Weg	Halt - Vorbeifahrt
SpDrS60_Tstr_Sig_BlK_Erstz	Taster Block-Ersatzsignal	Halt - Ersatzsignal
SpDrS60_Tstr_Sig_Erstz	Taster Ersatzsignal	Halt - Ersatzsignal
SpDrS60_Tstr_Sig_Hp2	Taster Hp2 ( Flankenschutzmelder )	Halt - Langsamfahrt
SpDrS60_Tstr_Sig_Hp	Taster Hauptsignal	Halt - Fahrt
SpDrS60_Tstr_Sig_Sh	Taster Rangiersignal	Halt - Langsamfahrt
SpDrS60_Tstr_Wche_DKW	Taster Weiche DKW	Li-Re - Re-Re - Re-Li
SpDrS60_Tstr_Wche_EKW	Taster Weiche EKW	Li-Li - Li-Re - Re-Re - Re-Li
SpDrS60_Tstr_Wche_Glspr	Taster Gleissperre	Geöffnet - Geschlossen
SpDrS60_Tstr_Wche	Taster Weiche	Gerade - Abzweig
<b>Weichenfelder</b>		
SpDrS60_Wche_diag_Li	Weiche diagonal Links	Gerade - Abzweig / Fahrstraße / Besetzt
SpDrS60_Wche_diag_Re	Weiche diagonal Rechts	Gerade - Abzweig / Fahrstraße / Besetzt
SpDrS60_Wche_DKW_Li	Weiche DKW Links	Alle Richtungen / Fahrstraße / Besetzt
SpDrS60_Wche_DKW_Re	Weiche DKW Rechts	Alle Richtungen / Fahrstraße / Besetzt
SpDrS60_Wche_EKW_Li	Weiche EKW Links	Alle Richtungen / Fahrstraße / Besetzt
SpDrS60_Wche_EKW_Re	Weiche EKW Rechts	Alle Richtungen / Fahrstraße / Besetzt
SpDrS60_Wche_GLSpr_Li	Weiche Gleissperre Links	Geschlossen-Geöffnet / Fahrstraße / Besetzt
SpDrS60_Wche_GLSpr_Re	Weiche Gleissperre Rechts	Geschlossen-Geöffnet / Fahrstraße / Besetzt
SpDrS60_Wche_Li	Weiche Links	Gerade - Abzweig / Fahrstraße / Besetzt
SpDrS60_Wche_Re	Weiche Rechts	Gerade - Abzweig / Fahrstraße / Besetzt

