

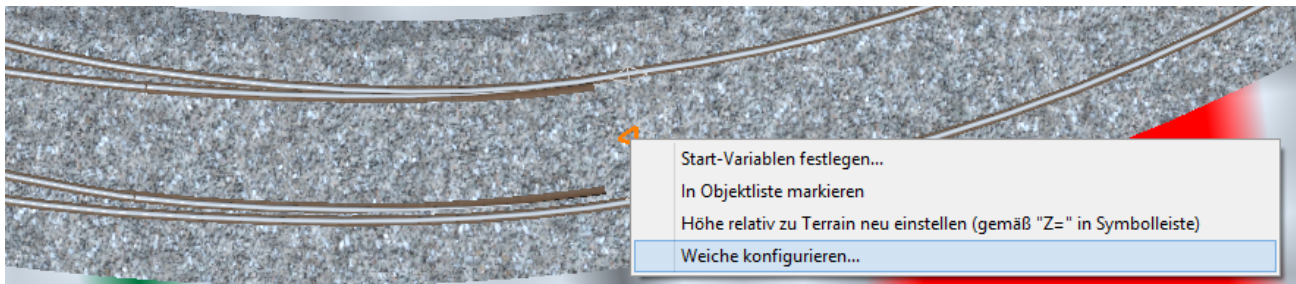
Fahrstraßen, Signale und elektrische Weichen

Table Of Contents

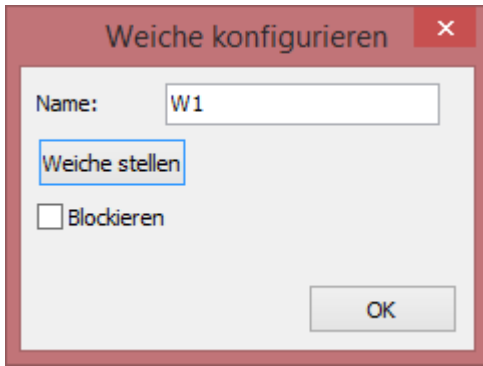
- [1 Weichen vorbereiten](#)
- [2 Bezirk anlegen](#)
- [3 Fahrstraße anlegen](#)
 - [3.1 Weichen und Pfade hinzufügen](#)
 - [3.2 Weitere Eigenschaften der Fahrstraßen](#)
- [4 Trigger/Baken konfigurieren](#)
 - [4.1 Trigger platzieren](#)
- [5 Einfache Signale anschließen](#)
- [6 Komplexere Signale](#)
 - [6.1 Signalobjekt konfigurieren](#)
 - [6.2 Signal im MapEditor konfigurieren](#)
- [7 Fahrzeug ausrüsten](#)
- [8 Script-Implementierung](#)
 - [8.1 Fahrzeug](#)
 - [8.2 Einfache Signale](#)
 - [8.3 Komplexe Signale](#)

1 Weichen vorbereiten

Es ist nicht *nötig*, aber *empfehlenswert*, den Weichen (zumindest denen, die in Fahrstraßen verwendet werden sollen) Namen zu geben. Hierzu im "Gleis"-Modus mit der rechten Maustaste auf diejenige Markierung klicken, die in Richtung Weichenspitze zeigt, und im Kontextmenü auf "Weiche konfigurieren..." klicken.



Im erscheinenden Dialogfeld kann sofort der Weichenname eingegeben werden.

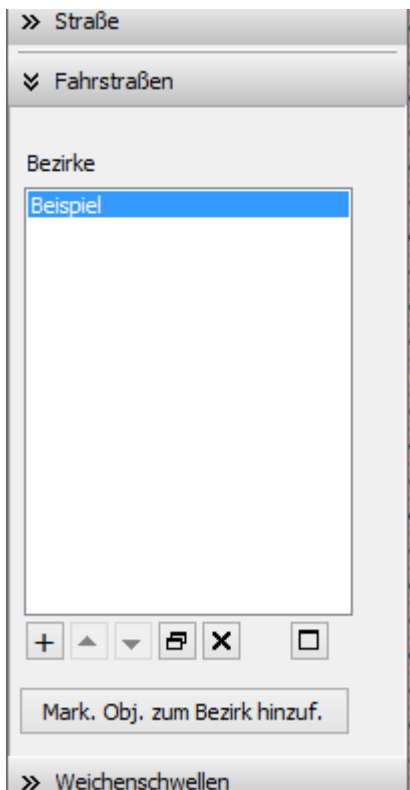


2 Bezirk anlegen

Der Übersichtlichkeit aber auch der Performance halber werden Fahrstraßen zu Bezirken gruppiert, die jeweils unabhängig voneinander arbeiten. Man kann sich diese wie einzelne Stellwerke vorstellen.

Für die Performance relevant ist die Gruppierung insbesondere deshalb, weil jeder Bezirk eine Ausschluss-Matrix enthält, die die Größe "Anzahl der Fahrstraßen zum Quadrat" hat, und über die bei der Anforderung einer Fahrstraße nachgeschaut wird, ob bereits eine "feindliche" Fahrstraße eingelegt ist. Angenommen, man hätte ein großes Netz mit 20 Stellwerken à 10 Fahrstraßen. Würde man nun keine Bezirke definieren, dann würde man eine 200x200-Matrix benötigen, die dementsprechend 40.000 Einträge benötigt. Würden die Fahrstraßen zu den jeweiligen Stellwerken gruppiert werden, dann wären es 20 Matrizen mit nur noch $10 \times 10 = 100$ Feldern, also nur insgesamt 1.000 Einträgen (gegenüber 40.000!).

Die Bezirke werden über die in vielen anderen Bereichen bekannte Listen-Box verwaltet, die sich links im Abschnitt "Fahrstraßen" befindet.

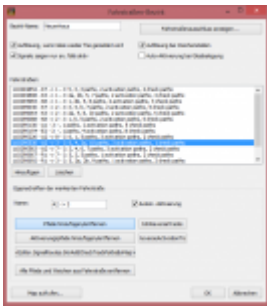


3 Fahrstraße anlegen

Im Eigenschaften-Dialogfeld (erreichbar per Doppelklick oder die Schaltfläche mit dem Rechteck unterhalb der Liste) kann zunächst dem Bezirk ein Name gegeben werden und es können folgende Grundeinstellungen für den Bezirk angegeben werden:

- **Auflösung, wenn Gleis wieder frei gemeldet wird:** Aktive Fahrstraßen werden automatisch aufgelöst, wenn die zugehörigen Gleise wieder als frei gemeldet werden, nachdem zuvor ein Zug darauf gestanden hat. Option ist standardmäßig eingeschaltet.
- **Signale zeigen nur an, falls aktiv:** Ist diese Option aus, dann erscheinen die Weichensignale auch dann, wenn eine zugehörige Fahrstraße "weichenmäßig" aktiviert werden könnte, d.h. sie geben praktisch ständig den Stand der Weiche an. Leuchtet dennoch kein Weichensignal, auch wenn die Option abgeschaltet ist, dann sind die aktuellen Weichenlagen so, dass es keine sinnvolle Fahrstraße gibt. Option ist standardmäßig eingeschaltet.
- **Auflösung bei Weichenstellen:** Wird eine Weiche manuell gestellt, die eigentlich durch eine aktive Fahrstraße geschützt wird, dann soll die Fahrstraße aufgelöst werden. Option ist standardmäßig eingeschaltet.
- **Auto-Aktivierung bei Gleisbelegung:** Falls eine Fahrstraße befahren wird, die nicht eingelegt ist, die Weichenstellungen aber passen und auch keine "feindliche" Fahrstraße eingelegt ist, dann soll sie automatisch aktiviert werden.

Darunter können mit "Hinzufügen" und "Löschen" die einzelnen Fahrstraßen erstellt oder entfernt werden. Wird eine davon angeklickt, können direkt unterhalb der Liste deren Eigenschaften bearbeitet werden:



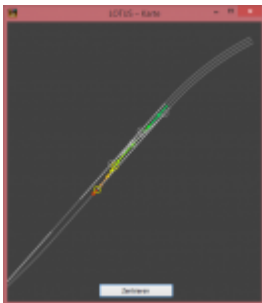
Dies umfasst aktuell den Namen der einzelnen Fahrstraße und die Weichen, die gestellt werden sollen, wenn die Fahrstraße aktiviert wird.

3.1 Weichen und Pfade hinzufügen

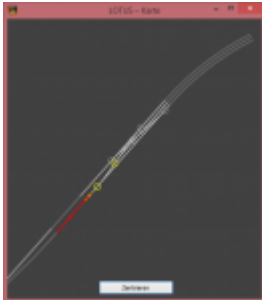
Zusammen mit dem Eigenschaften-Dialogfeld öffnet sich auch eine 2D-Karte. In ihr werden Pfade zu den Fahrstraßen hinzugefügt oder entfernt. Generell werden nur Pfade hinzugefügt oder entfernt, welche Weichen zur Fahrstraße gehören, ermittelt der MapEditor daraus automatisch.

Pfade können einer der folgenden drei Kategorien hinzugefügt werden:

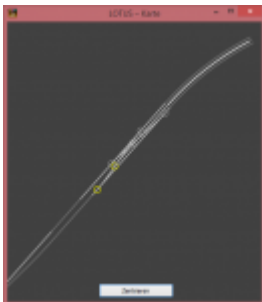
- Die normalen Pfade, die die eigentlich Fahrstraße ausmachen. Dieser Bereich beginnt hinter dem Signal und endet hinter der letzten Weiche. Damit die Fahrstraße eingelegt werden kann, müssen diese frei von Fahrzeugen sein und die zugehörigen Weichen gestellt werden. Dementsprechend wird über sie auch der Fahrstraßenausschluss ermittelt, d.h. welche anderen Fahrstraßen gleichzeitig eingestellt werden können. Sobald das Fahrzeug diese befährt, gilt die Fahrstraße als besetzt und das zugehörige Signal fällt auf "Halt", sofern dieses entsprechend konfiguriert wurde. Verlässt das Fahrzeug diesen Bereich wieder vollständig, wird die Fahrstraße aufgelöst.



- Aktivierungspfade: Diese kommen nur zur Anwendung, wenn der Stellwerksbezirk ein "echtes" Stellwerk ist, d.h. nicht, wenn es sich nur um geschützte Weichen handelt, die vor Ort durch Stellbefehle der Fahrzeuge legen. Die Aktivierungspfade liegen vor dem zugehörigen Signal. Sobald diese befahren werden, wird die Stellwerksautomatik bzw. der virtuelle Fahrdienstleiter aufgefordert, die entsprechende Fahrstraße einzulegen, sofern sie zur Fahrplan-Fahrt passt und der Zug weniger als 2 Minuten Verfrühung hat oder die Option "Autom. Aktivierung" der Fahrstraße angehakt wurde – und natürlich nur dann, wenn sie auch eingelegt werden darf/kann.



- Prüfpfade: Der Bereich zwischen letzter Weiche und End-Signal: Befindet sich der Zug bereits hier, dann kann die Fahrstraße aufgelöst werden, da alle Weichen geräumt wurden. Trotzdem sollen aber keine Fahrstraßen eingelegt werden, wenn sich hier ein Zug befindet oder das Gleis zu einer bereits eingelegten Fahrstraße gehört. Deshalb wird dieser Bereich dennoch vor Einlegen der Fahrstraße geprüft.



Je nachdem, zu welcher Kategorie Pfade hinzugefügt oder von ihr entfernt werden sollen, muss eine der drei zugehörigen Schaltflächen durch Anklicken aktiviert werden. Danach können die Pfade in der 2D-Karte durch Anklicken hinzugefügt oder entfernt werden.

Außer bei Prüfpfaden muss auf die korrekte Ausrichtung der Pfade geachtet werden. Der MapEditor sorgt selbstständig dafür, dass die Pfade, wenn sie korrekt miteinander verbunden sind, einheitlich ausgerichtet sind. Soll diese Ausrichtung umgekehrt werden, muss die zugehörige Schaltfläche "Ausrichtung umkehren" angeklickt werden.

Der Eintrag in der Fahrstraßenliste wird jeweils aktualisiert, sodass man dort u.a. stets ablesen kann, ob die entsprechenden Weichen wie gewünscht hinzugefügt wurden.

Achtung: Die Option "Auto-Aktivierung" sollte nie in "echten" Stellwerksbezirken genutzt werden, in denen insbesondere Weichen aus verschiedenen Richtungen befahren werden! Sie sollte nur in typischen

einfachen Situationen im Straßenbahn-Bereich genutzt werden, bei denen die Fahrstraßen nur aus höchstens zwei Weichen bestehen, die nur aus einer Richtung befahren werden.

3.2 Weitere Eigenschaften der Fahrstraßen

- Autom. Aktivierung: Hierbei wird eigentlich geschummelt, weshalb es eine entsprechende Realismus-Option geben wird, mit der diese Eigenschaft unwirksam geschaltet werden kann: In der Realität muss im Allgemeinen jeder Zug in einem Stellwerksbezirk auch eine Zugnummer haben oder – im Falle von Rangierfahrstraßen – zumindest mit dem Fahrdienstleiter kommunizieren. Dass man einfach "so" mit einem Zug daherkommt und sich dann trotzdem Fahrstraßen automatisch einlegen, gibt es ansonsten nur bei Selbstblocksignalen, niemals aber bei Fahrstraßen, die Weichen decken. Um es dem Erkundenden etwas netter zu gestalten, kann man Fahrstraßen mit dieser Option versehen. Hierdurch wird – sofern der Zug nicht fahrplanmäßig unterwegs ist – die Fahrstraße automatisch eingelegt, sobald der Zug die Aktivierungspfade befährt.
- Alle Pfade und Weichen aus Fahrstraße entfernen: Hier wird genau das getan – die Fahrstraße ist danach frei von jedweden Pfaden.

4 Trigger/Baken konfigurieren

Nun, nachdem die Fahrstraßen definiert wurden, geht es daran, mittels Triggern deren automatisches Aktivieren und Deaktivieren zu ermöglichen. "Trigger" sind spezielle (in der Simulation unsichtbare) Szenerieobjekte. Trigger werden vielfältige Aufgaben übernehmen können, momentan dienen sie aber noch nur dem Einlegen und Auflösen von Fahrstraßen.

4.1 Trigger platzieren

Man findet das Trigger-Objekt rechts in der Kategorie "Helfer". Es muss einfach nur "ungefähr" auf dem Gleis platziert werden; die interne "Verknüpfung" mit dem Gleis wird dennoch gewährleistet. Eine exakte Ausrichtung/Drehung ist auch nicht nötig, aber die grobe Richtung (vorwärts oder rückwärts) ist sehr wichtig, weil die Sensoren/Trigger jeweils immer nur auf die Sensoren/Trigger "ihrer" Richtung reagieren.

Mittels Rechtsklick auf den Trigger und Auswahl von "Trigger-Eigenschaften..." wird das Eigenschafts-Dialogfeld aufgerufen:

Trigger-Eigenschaften

Bezirk: Test1

Fahrstraße setzen, wenn "Weiche links" gewählt wird: 56117254 - L - J: W1

Fahrstraße setzen, wenn "Weiche geradeaus" gewählt wird: 56117237 - M - J: W2, W1

Fahrstraße setzen, wenn "Weiche rechts" gewählt wird: 56117233 - R - J: W1, W2

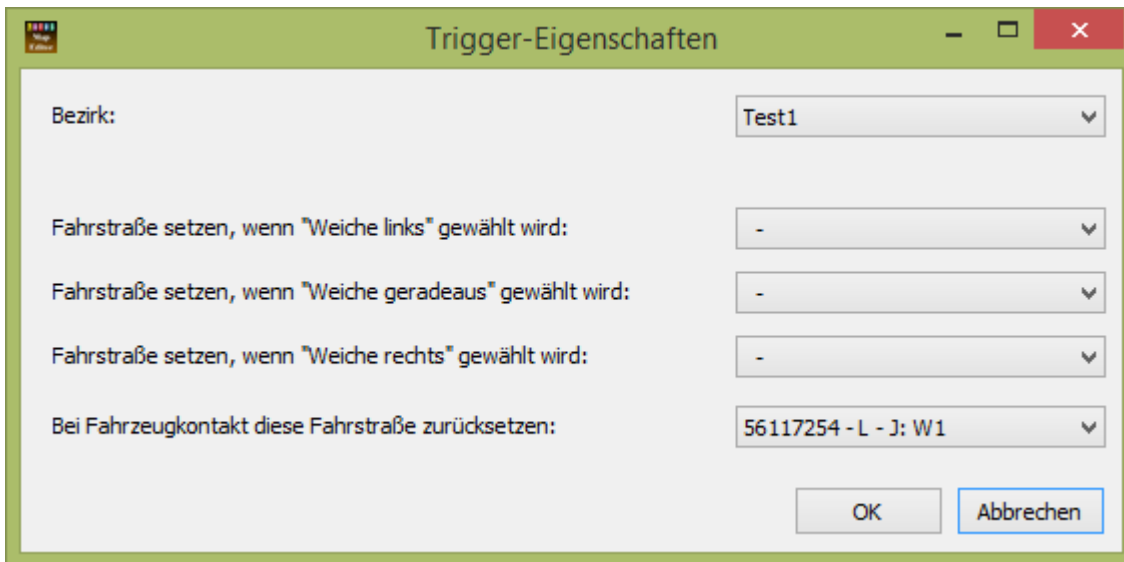
Bei Fahrzeugkontakt diese Fahrstraße zurücksetzen: -

OK Abbrechen

Zunächst wird hier ausgewählt, zu welchem Fahrstraßen-Bezirk der Trigger gehören soll.

Sendet das Fahrzeug einen Stellbefehl an den Trigger, dann soll dieser - je nach Stellbefehl - eine bestimmte Fahrstraße einlegen. Dieses Verhalten kann in den folgenden drei Dropdown-Menüs eingestellt werden. Soll der Trigger bestimmte oder alle Stellbefehle ignorieren, dann ist jeweils " - " auszuwählen.

Verlässt das Fahrzeug den Bereich der Fahrstraße, dann soll diese wieder aufgelöst (deaktiviert) werden. Sofern dies nicht über die Gleisfreimeldung erfolgt/erfolgen soll, kann für diesen Zweck einem Trigger ebenfalls eine Fahrstraße zugewiesen werden. Hierbei "wartet" der Trigger aber nicht auf eine bestimmte Information/Aufforderung seitens des Fahrzeuges, sondern die Aufhebung wird sofort bei "Berührung" des Triggers ausgelöst.



5 Einfache Signale anschließen

Für einfache Signale (z.B. Einheits-Weichensignale nach BOStrab wie W1, W2, W3 usw.), bei denen ein Szenarioobjekt nur anzeigt, ob eine bestimmte Fahrstraße eingelegt ist, oder nicht, kommt ein vereinfachtes System zur Anwendung:

- Platzieren und Auswählen des [Signals](#)
- Auswählen des Bezirks links
- Auf "Mark. Obj. zum Bezirk hinzuf." unterhalb der Bezirks-Liste klicken
- Die Fahrstraße auswählen, die dieses Signal anzeigen soll

6 Komplexere Signale

6.1 Signalobjekt konfigurieren

Komplexe Signale müssen zuerst vorbereitet werden: Im Object & Vehicle Tool gibt es im Bereich "Allgemeine Einstellungen" die Schaltfläche "Signalbegriffe". Hierüber kann ein Texteingabe-Feld geöffnet werden, in welches (vereinfacht gesagt) diejenigen Signalbegriffe eingetragen werden müssen, die das Signal zeigen kann. Hierbei ist aber zu beachten, dass

- das "Nichtvorhandensein" von Fahrstraßen usw., also ein Hp0 oder ein dunkles Weichensignal usw., nicht ein eigener Signalbegriff sind
- keine "Kombinationen" eingetragen werden
- maximal sind 32 Zeilen und somit Begriffe erlaubt

Kann das Signal beispielsweise sowohl Vor- als auch Hauptsignal-Begriffe anzeigen, dann erhält es bspw. die Einträge:

- Hp1
- Hp2
- Vr1
- Vr2

FALSCH wäre also sowas wie:

- Hp0+Vr0
- Hp1+Vr0

- Hp2+Vr0
- Hp1+Vr1
- usw.

Warum das so ist, wird vermutlich klarer, wenn wir uns mit der Konfiguration im MapEditor beschäftigt haben! 😊

6.2 Signal im MapEditor konfigurieren

Sobald dem Signal-Objekt auf die beschriebene Weise die möglichen Signalbegriffe mitgeteilt wurden, erscheint im MapEditor, wenn man auf das Signal mit rechts klickt, im Kontextmenü der Eintrag "Signal-Eigenschaften". Wird dieser angeklickt, öffnet sich das hierfür vorgesehene Dialogfeld.

Von der Struktur her gibt es für jeden Signalbegriff einen Listen-Abschnitt, zu dem jeweils beliebig viele Fahrstraßen hinzugefügt werden können. Die Anzahl der Signalbegriffe ergibt sich durch die Eingaben beim Signal-Objekt im Object & Vehicle Tool. Für jeden Fahrstraßeneintrag kann jeweils individuell der Bezirk ausgewählt werden, damit es auch möglich ist, Signalbegriffe bezirksübergreifend zusammen zu stellen. Außerdem wird jeweils ausgewählt, ob geprüft werden soll, ob

- die Fahrstraße aktiv ist,
- deren Gleise belegt sind,
- sie aktiv ist und gleichzeitig ihre Gleise nicht belegt sind oder
- sie "möglich" ist, d.h. die zugehörigen Weichen bereits passend liegen

Das ganze System arbeitet nun wie folgt: Für jeden Signalbegriff wird stets geprüft, ob für zumindest eine der Fahrstraßen die ausgewählte Bedingung erfüllt ist. Falls dies der Fall ist, "gilt" dieser Signalbegriff. Schließlich werden die so ermittelten "Gesamtzustände" jeder Fahrstraße kodiert und per Script an das Signal übertragen.

7 Fahrzeug ausrüsten

Damit ein Fahrzeug mit den Baken interagieren kann, muss es mindestens einen Sensor bekommen, Zweirichtungsfahrzeuge benötigen mindestens einen jeweils an jedem Ende. Eingerichtet wird dieser in den Objekteinstellungen im Abschnitt "Sensoren":

The screenshot shows a dialog box titled "Sensoren". On the left, there is a list of sensors with a count of "0" and a minus sign. To the right of the list, there are two input fields: "Y-Position" with the value "5.50000" and "Gegen Fahrtrichtung" with an unchecked checkbox. At the bottom right of the dialog is a button labeled "Hinzufügen".

Die Parameter bedeuten:

- Y-Position: Die Position des Sensors am Fahrzeug in Längsrichtung
- Gegen Fahrtrichtung: Die Sensoren interagieren immer nur mit jenen Triggern, die dieselbe Ausrichtung haben, weshalb hier entsprechend angegeben werden muss, in welche Richtung dieser Sensor eingebaut werden soll.

8 Script-Implementierung

8.1 Fahrzeug

Scriptseitig gibt es bereits folgende Möglichkeiten der Interaktion zwischen Fahrzeug-Sensor und Strecken-Trigger:

- Senden von Meldungen an alle in Reichweite befindliche Trigger mit `SendMessageToTrigger(Self; ID: string; value: string; sensor_index: integer)`. Bisher unterstützt werden die IDs
 - "SWITCH" mit den Werten (jeweils als String!) "0" (links), "1" (rechts) und "2" (geradeaus) zum Übertragen eines Stellbefehls oder "toggle" zum Umschalten zwischen zweier der drei aufgeführten Fahrstraßen.
 - "LINEROUTE", bei dem die Zahlen der Linie und der Route und der Name des ITCS-Servers übertragen werden, als String getrennt mit einem "/". So wird aus Linie 123 mit Route 45 der String "123/45/{Name}". Die Übertragung der Linie und Route sorgt dafür, dass LOTUS anhand der vorliegenden, gültigen und zum ITCS-Server-Namen passenden Fahrpläne herausucht, welche Route hierfür hinterlegt ist, und anhand dieser den Stellbefehl passend gibt. Der Name des ITCS-Servers wird zuvor üblicherweise mit der Funktion `PIS_GetITCSServer(self: integer)` geholt.
- Wird in das Script eine Prozedur eingebunden mit der Deklaration `procedure OnEnterLeaveTrigger(triggerid: string; entering: boolean; sensorindex: integer)`, dann wird diese stets aufgerufen, wenn das Fahrzeug einen Trigger erreicht oder verlässt. "Triggerid" wird aktuell nicht verwendet, später wird es z.B. zur Identifikation für das IBIS usw. verwendet werden.

Im GT6N werden zum Beispiel beide Funktionen zusammen verwendet: Wenn der Fahrer einen Stellbefehl aktiviert, dann wird dieser Wert (für eine bestimmte maximale Wegstrecke) "vorgemerkt". Wird dann `OnEnterLeaveTrigger` ausgelöst, dann benutzt das Script sofort `SendMessageToTrigger` und sendet damit den Stellbefehl an den Trigger. Denkbar sind aber auch andere Varianten: Wenn der Trigger einen größeren Radius hat (z.B. für Kontakt-Stellbefehle oder längere Kontaktschleifen), dann könnte `SendMessageToTrigger` auch unmittelbar ausgelöst werden, wenn der Fahrer den Stellbefehl betätigt.

8.2 Einfache Signale

So ähnlich wie bei den Ampeln muss lediglich die "PUBLIC"-Variable `signalroute_active: boolean` eingebunden werden. Wenn alles korrekt verknüpft ist und die Fahrstraße eingelegt wird, wird diese Variable "true", sonst ist sie "false". Hiermit lassen sich aber dementsprechend nur einfache Signale realisieren. Komplexe Signale kommen später! 😊

8.3 Komplexe Signale

Auch bei komplexen Signalen werden die Daten über eine "PUBLIC"-Variable, nämlich `signalstate: integer` übertragen. Achtung: Im Gegensatz zu einfachen Signalen handelt es sich hierbei um eine Integer-Variable!

In diese werden nun im Betrieb die Signalbegriffe bitweise eingetragen – siehe hierzu auch [Bit-Flags erklärt](#). Verfügt das Signal z.B. über die Begriffe Hp1, Hp2, Hp3 und Hp4, dann wird der Wert der Variable `signalstate`

die folgenden Werte annehmen:

- 1, falls *nur* die Hp1-Bedingungen gelten
- 2, falls *nur* die Hp2-Bedingungen gelten
- 4, falls *nur* die Hp3-Bedingungen gelten oder
- 8, falls *nur* die Hp4-Bedingungen gelten.

Gelten mehrere Bedingungen gleichzeitig, dann werden diese Werte einfach zusammengerechnet.

Auseinander nehmen kann man die Bedingungen dann wie folgt:

- wenn `(signalstate and 1) <> 0`, dann ist die Hp1-Bedingung erfüllt
- wenn `(signalstate and 2) <> 0`, dann ist die Hp2-Bedingung erfüllt
- wenn `(signalstate and 4) <> 0`, dann ist die Hp3-Bedingung erfüllt
- wenn `(signalstate and 8) <> 0`, dann ist die Hp4-Bedingung erfüllt

Komplexer, aber trotzdem problemlos ist es, wenn Zusatz-Anzeiger zur Anwendung kommen.

Beispiel: Ein nur Hp4-fähiges U-Bahnsignal mit Gleis-Nummern-Anzeige. Die Signalbegriffe wären dann:

- Hp4
- Gleis 1
- Gleis 2
- Gleis 3
- Gleis 4
- Gleis 5
- Gleis 6
- Gleis 7
- Gleis 8
- Gleis 9

Angenommen, es gibt drei Fahrstraßen A, B und C, die auf Gleis 3, 4 und 7 führen.

Dann werden im MapEditor in der Signal-Konfiguration dem Signalbegriff *Hp4* alle drei Fahrstraßen zugeordnet, den Signalbegriffen *Gleis 3*, *Gleis 4* und *Gleis 7* jeweils nur die Fahrstraße A, B und C, und den restlichen Signalbegriffen werden gar keine Fahrstraßen zugeordnet.

Nach obiger Formel wird die Variable `signalstate` dann folgende Werte annehmen:

- Fahrstraße A: Hp4+Gleis 3, d.h. $1 + 8 = 9$
- Fahrstraße B: Hp4+Gleis 4, d.h. $1 + 16 = 17$
- Fahrstraße C: Hp4+Gleis 7, d.h. $1 + 128 = 129$

Im Signalscript können dann die Signalbegriffe wieder mit den obigen "and-Formeln" auseinander genommen werden:

Falls `(signalstate and 1) <> 0`, dann soll das Signal auf Hp4 gehen, sonst auf Hp0. Nur, wenn es Hp4 zeigt, dann soll er die restlichen Werte in `signalstate` prüfen und dann entsprechend daraus ableiten, wie er die jeweiligen Gleisnummern anzeigen soll.