42.2 LUA-Skript – Von der Problemstellung zur Realisierung Teil 2

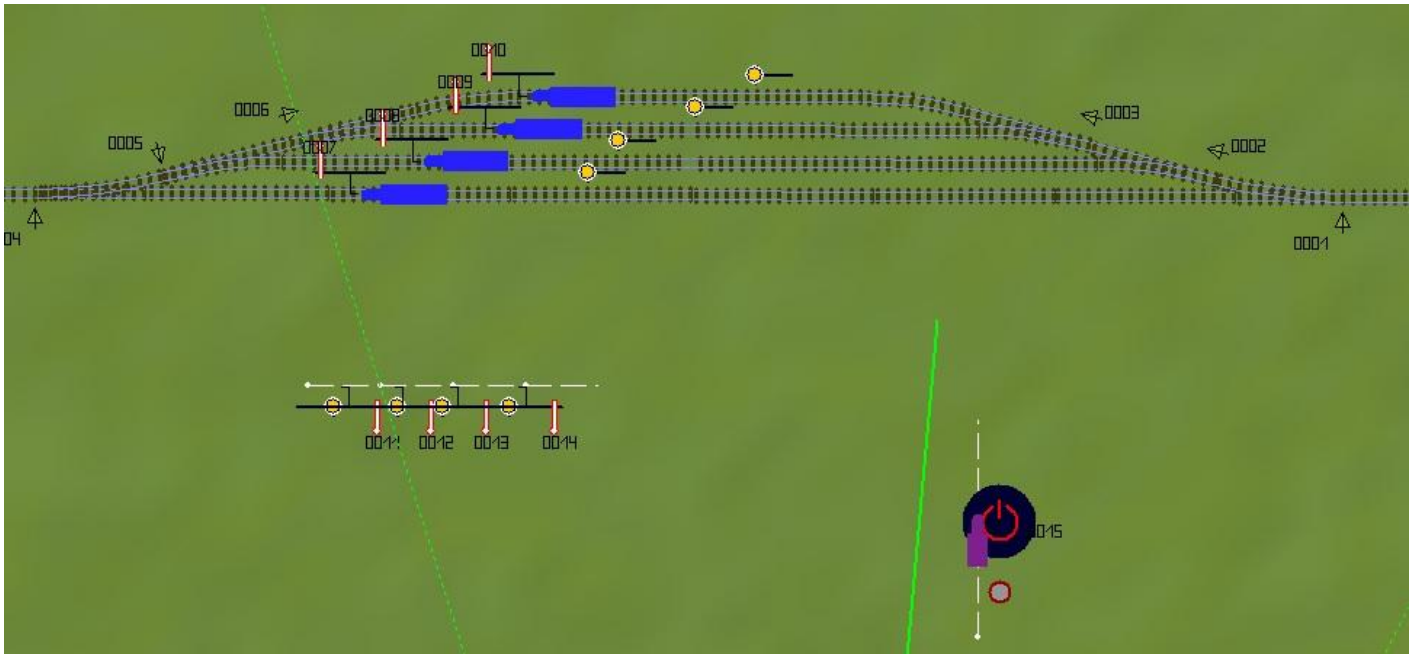
Autor: Mover
Quelle: EEP-Freunde-Forum

Download:
Anlage aktueller Stand

Kommen wir nun zum zweiten Teil. Dazu habe ich die in Teil 1 verwendete Anlage etwas umgebaut.

Bevor wir starten, möchte ich euch eine Seite ans Herz legen: LUA für Anfänger

Nun zuerst die Ausgangssituation.

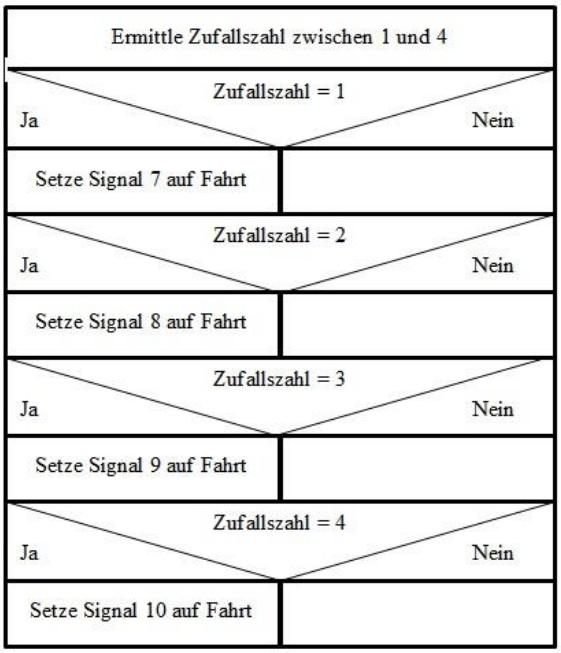


Auf den Gleisen 1 bis 4 stehen jeweils 1 Zug (in dem Fall nur die Loks). Das Startsignal soll zufällig einen der Züge auf die Reise schicken. Bei der Einfahrt soll geprüft werden welches Gleis frei ist und die entsprechende Fahrstraße schalten. Dabei binden wir die in Teil 1 erstellten Funktionen im LUA-Skript ein (sind in der Ausgangsanlage bereits drin). Da nur so viel Züge wie Gleise auf der Anlage sind, müssen wir uns hier keine Gedanken machen, was zu tun ist wenn alle Gleise besetzt sind. Bei der Einfahrt wird dann wieder ein neuer Zug losfahren.

Die Signale 11 bis 14 zeigen an, welches Gleis frei (Fahrt) oder besetzt (Halt) ist. Im Moment sind alle besetzt, da ja noch kein Zug unterwegs ist.

Als erstes müssen wir etwas erstellen, was einen Zug zufällig losfahren lässt.

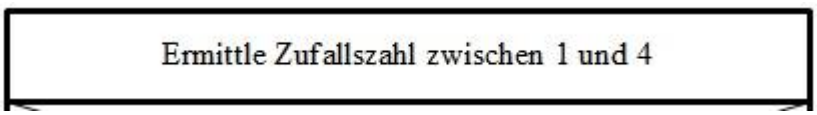
Struktogramm Funktion ZugLos()



Schauen wir uns mal das Struktogramm an. Als erstes muss die Zufallszahl, die dann der Gleisnummer entspricht, ermittelt werden. Das geht nur an dieser Stelle, da wir das Ergebnis anschließend brauchen. Es würde also nichts bringen, diese Anweisung ganz an das Ende zu bringen.
Als nächste Anweisung haben wir eine Verzweigung (hier eine einseitige) wo wir nun unsere Zufallszahl prüfen, ob der Wert 1 ist. Ist dies der Fall wird das Signal 7 auf Fahrt gestellt. Wenn nicht passiert hier nichts.
Die nächste Anweisung ist wieder eine Verzweigung. Dieses mal wird geprüft ob die Zufallszahl den Wert 2 hat. Nun könnte man dies auch in den Nein-Zweig der 1 Prüfung machen, aber am Ergebnis würde sich am Ende nichts ändern und für Anfänger ist es so einfacher, da am Ende dann eine Verschachtelung bis zur 4. Ebene vorhanden wäre.
Die restlichen 2 Abfragen könnt ihr euch ja nun selbst denken warum die sind.

Kommen wir nun zur Umsetzung in LUA.

Unsere erste Anweisung im Struktogramm:

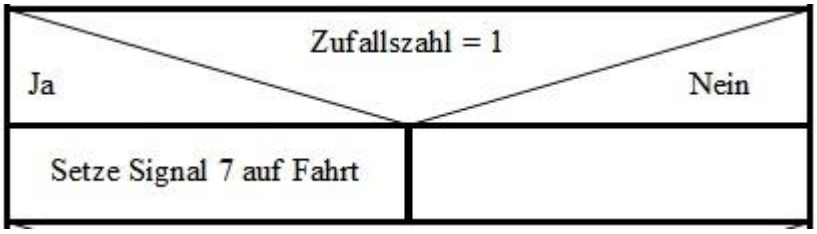


Wir brauchen in dem Fall die Variable Zufallszahl. Diese speichert das Ergebnis der LUA-eigenen Funktion math.random. Dies ist also keine EEPX spezifische Funktion. Dieser Funktion müssen wir 2 Parameter übergeben: Parameter 1 = Untergrenze, Parameter 2 = Obergrenze.

Unser LUA Befehl sieht dann so aus:
Zufallszahl = math.random(1,4)

Durch das Gleichheitszeichen wird das Ergebnis der Zufallsfunktion der Variablen Zufallszahl zugewiesen.

Weiter geht es mit der nächsten Anweisung im Struktogramm.



Für Verzweigungen gibt es in LUA die if-Anweisung. Diese sieht allgemein so aus:

```
if Bedingung then
   -- Hier kommt der Ja-Zweig rein
else
   -- Hier kommt der Nein-Zweig rein
end     -- Hier endet der if-Block
```

Ist kein Nein-Zweig nötig kann else weggelassen werden.

Übertragen wir dies jetzt auf unser konkretes Vorhaben:

```
if Zufallszahl == 1 then
   EEPSetSignal(7,1)
end
```

Nun noch etwas zu den Gleichheitszeichen:

= weist einen Wert zu wie dies hier: Zufallszahl = math.random(1,4)
== vergleicht auf Gleichheit. also wie bei: Zufallszahl == 1

Da die restlichen Abfragen im Struktogramm nichts neues aufweisen, komme ich nun zum fertigen LUA-Skript unserer Funktion.

```
function ZugLos()
    Zufallszahl = math.random(1,4)
    if Zufallszahl == 1 then
        EEPSetSignal(7,1)
    end
    if Zufallszahl == 2 then
        EEPSetSignal(8,1)
    end
    if Zufallszahl == 3 then
        EEPSetSignal(9,1)
    end
    if Zufallszahl == 4 then
        EEPSetSignal(10,1)
    end
end
```

Rein theoretisch könnte jetzt, wenn wir schon einen Kontaktpunkt hätten der die Funktion ZugLos aufruft, die Funktion testen. Aber damit wir die Anlage danach nicht neu laden müssen, fehlt noch ein bisschen.

Was passiert wenn ein Zug jetzt losfahren würde? Er würde seine Runde drehen und wenn es passen würde durch reinen Zufall wieder auf das gleiche Gleis fahren. Wenn es nicht passt kracht es. Außerdem würde er durchrauschen, da das Signal nicht wieder auf Halt gestellt würde. Weiterhin müssten wir das Gleis als frei kennzeichnen. Ohne LUA bräuchten wir dazu 2 Signalkontaktpunkte hinter dem jeweiligen Signal eines Gleises. Heißt also 8 Kontaktpunkte. Mit LUA könnten wir dies auf 4 reduzieren.

Die Funktion dafür nenne ich AusfahrtG1. Hier das Struktogramm für die Ausfahrt Gleis 1:

Struktogramm AusfahrtG1()

Setze Signal 7 auf Halt
Setze Signal 11 auf Fahrt

Für die anderen Gleise müssten wir nur die Signal-IDs ändern, die Logik wäre die gleiche. Deshalb werde ich für die Gleise 2 bis 4 kein weiteres Struktogramm anfertigen und zeigen.

Wir brauchen hier also noch 3 weitere Funktionen und zwar für jedes Gleis eine eigene. An dieser Stelle wünschte ich mir, dass man bei den Kontaktpunkten eine Funktion mit Parametern aufrufen könnte. Dann könnte man das auf eine Funktion beschränken.

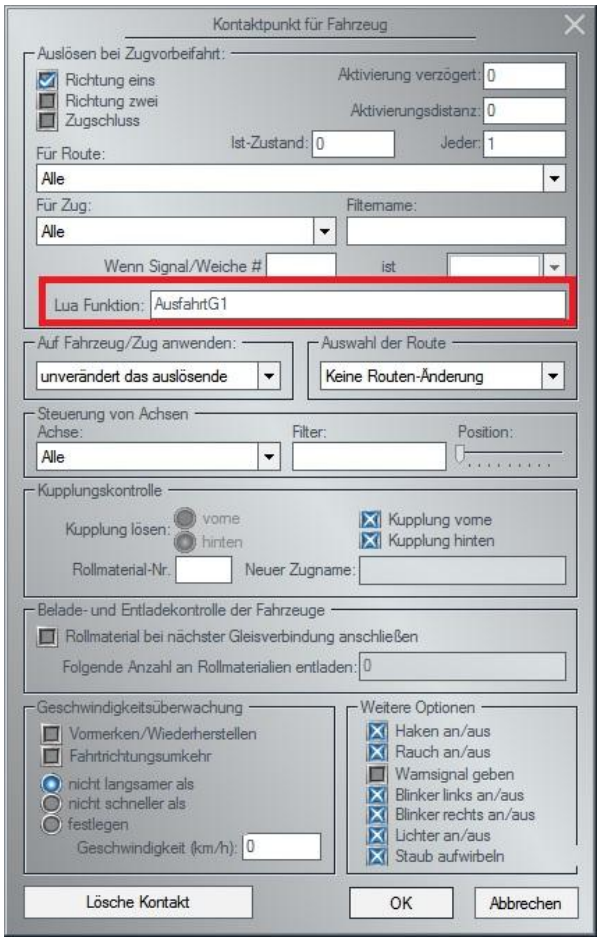
Es ginge auch so. Nur möchte ich jetzt keinen Wirrwar produzieren, da mir das eben auch erst eingefallen ist. Dazu mache ich dann noch ein eigenes Thema.

Nun aber das LUA-Skript für die Funktionen zum Ausfahren der Züge:

```
function AusfahrtG1()
    EEPSetSignal(7,2)
    EEPSetSignal(11,1)
end

function AusfahrtG2()
    EEPSetSignal(8,2)
    EEPSetSignal(12,1)
end

function AusfahrtG3()
    EEPSetSignal(9,2)
    EEPSetSignal(13,1)
end

function AusfahrtG4()
    EEPSetSignal(10,2)
    EEPSetSignal(14,1)
end
```

Setzen wir jetzt unsere Fahrzeugkontaktpunkte so wie im Bild.

In den Kontaktpunkt von Gleis 1 (unten) tragen wir folgendes ein.



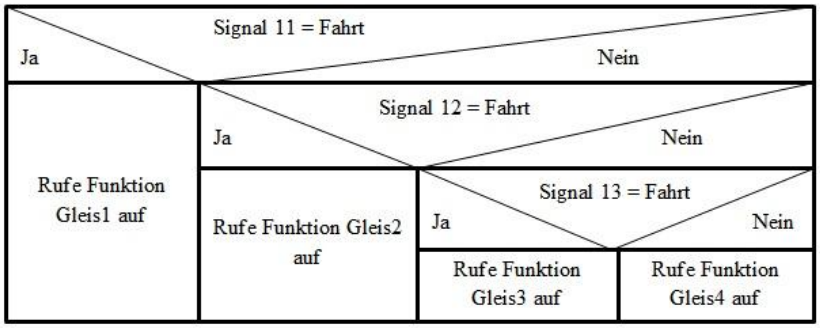
Das wiederholen wir nun für die anderen Gleise. Nur müssen wir da die entsprechende Funktion nehmen AusfahrtG2 bis 4. Gleis 4 wäre das oberste.

Jetzt haben wir dafür gesorgt, dass ein Zug wieder in einem Gleis hält und dass man weiß welches Gleis frei ist. Was jetzt noch fehlt ist eine Funktion, die dafür sorgt in das freie Gleis einzufahren.

In diesem Beispiel verwende ich nun mal eine verschachtelte Verzweigung. Im Prinzip wird dadurch folgendes gemacht: Finde das erst beste freie Gleis und lasse den Zug dahin fahren. Als erstes Prüfen wir das Signal 11 ob es auf Fahrt ist. Ist dies der Fall fährt der Zug auf Gleis 1, wenn nicht prüfen wir Signal 12 ebenso. Dies geht dann so weiter.

Das fertige Struktogramm sieht am Ende so aus:

Struktogramm Funktion Einfahrt



Woher kommen jetzt die Funktionen Gles1 bis Gleis4. Nun die haben wir schon in Teil 1 erstellt. Da sie genau das erledigen, was wir wollen, verwenden wir sie gleich wieder. Aus diesem Grund habe ich diese im Skript gelassen. Man muss ja das Fahrrad nicht 2 mal erfinden.

Signal 14 habe ich nicht weiter abgefragt, da dies in unserem konkreten Fall so ist, dass Gleis 4 frei ist, wenn alle anderen besetzt sind.

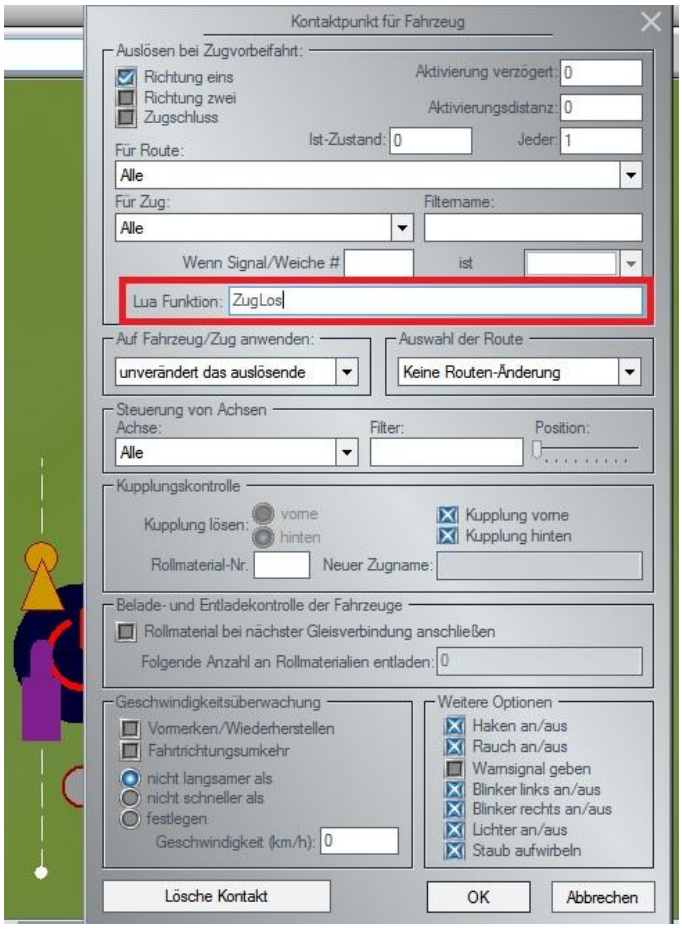
Nun das LUA-Skript für unsere Funktion Einfahrt.

```
function Einfahrt()
    if EEPGetSignal(11) == 1 then
        Gleis1()
    else
        if EEPGetSignal(12) == 1 then
            Gleis2()
        else
            if EEPGetSignal(13) == 1 then
                Gleis3()
            else
                Gleis4()
            end
        end
    end
end
```

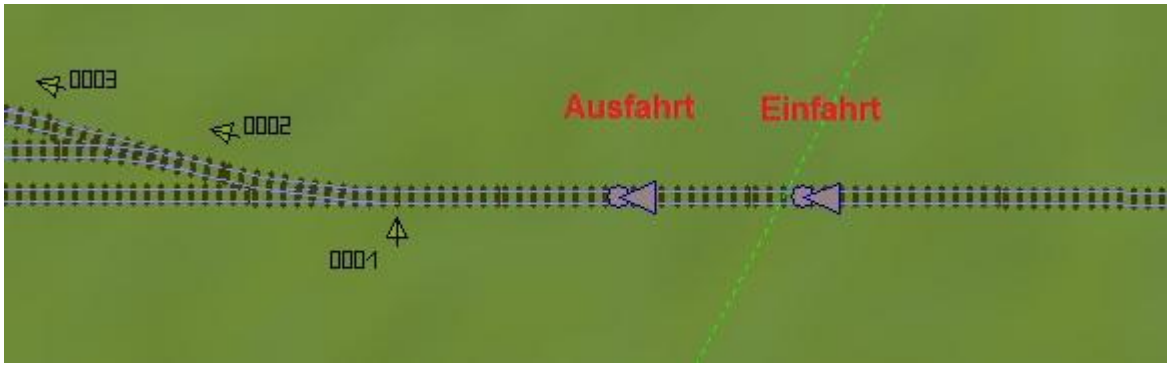
Das mit dem if-Block hatte ich ja schon erklärt. Fehlt im Prinzip die im Skript verwendete Funktion EEPGetSignal. Diese Funktion ermittelt den Schaltzustand eines Signals. Dazu muss man dieser Funktion in den Klammern die ID des Signals übergeben. Das Ergebnis dieser Funktion kann man in einer Variablen speichern (wenn man den Wert später in der Funktion noch mal braucht) oder wie hier im Skript direkt in einer Abfrage verwenden.

Damit hätten wir alle Funktionen zusammen. Fehlen nur noch die Kontaktpunkte für das Abfahren bzw. Einfahren.

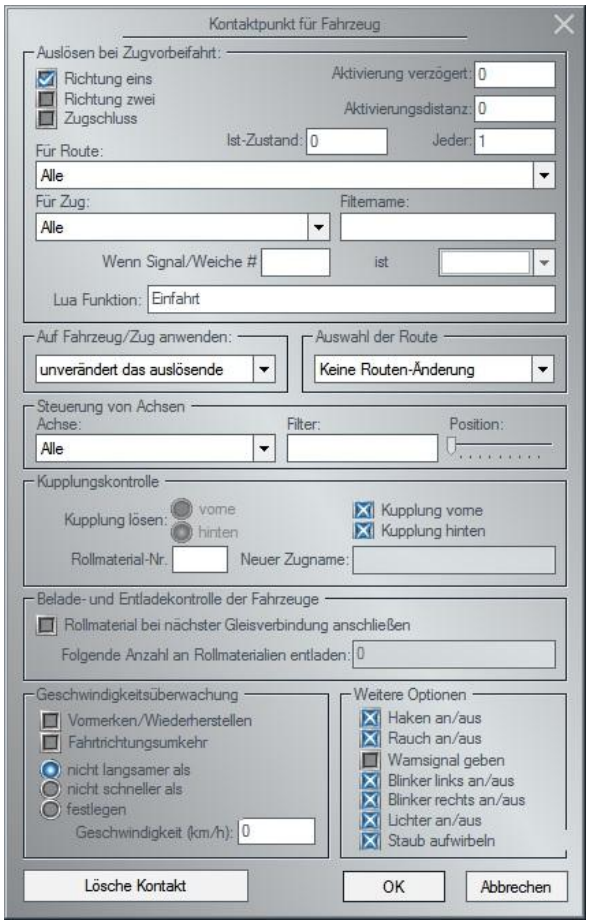
Fangen wir mit dem Anlagenstart an in dem wir wie folgt einen Fahrzeugkontakt setzen.



Jetzt noch zwei weitere wie im nächsten Bild.



Zum Abschluss noch das Eigenschaftenfenster vom Kontaktpunkt für die Einfahrt.



Jetzt können wir das ganze testen. Wer das jetzt macht, wird feststellen, dass bei der Einfahrt irgendwann etwas schief läuft. Beobachtet mal die Signale, welche anzeigen ob ein Gleis frei ist oder nicht. Beim zweiten Durchlauf zeigen 2 Signale an, dass ein Gleis frei ist. Ich habe schlicht vergessen bei der Einfahrt das jeweilige Signal wieder auf besetzt zu stellen, also auf Halt.

Aber das überlasse ich nun euch, dies zu korrigieren.