

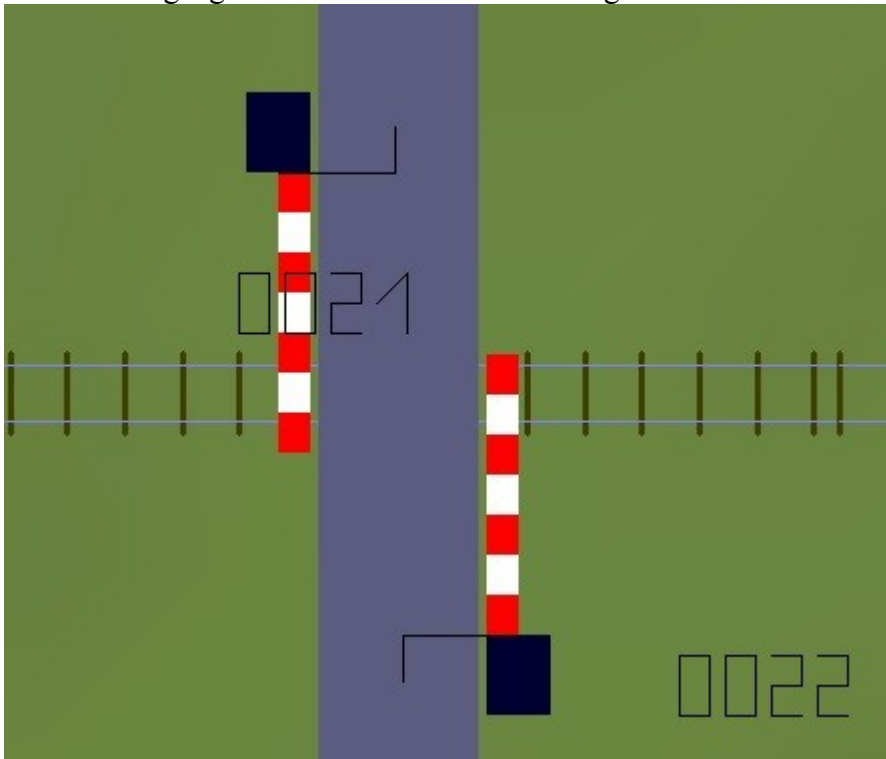
40. LUA-Skript - Schranken steuern

Hallo EEP-Freunde,

meinen ersten und vor allem sofort erfolgreichen Versuch mit der neuen Skriptsprache möchte ich euch nicht vorenthalten.

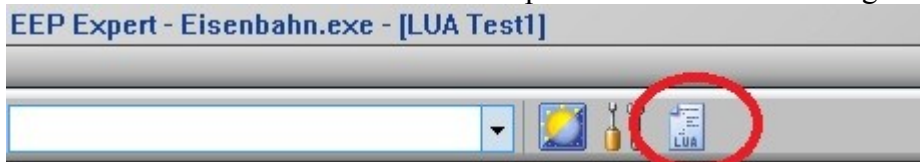
Hier geht es um das Steuern von Schranken über die in EEPX ab Plugin 2 integrierte Skriptsprache LUA. Ich werde in Zukunft in loser Folge meine Erkenntnisse zur Skriptsprache kund tun und versuchen sie verständlich darzulegen.

Nun zur Ausgangssituation. In meiner Testanlage zum Thema habe ich einen Bahnübergang gebaut.



Wie zu sehen ist, haben die Schranken die ID's 21 und 22. Das müssen wir uns im Vorfeld merken, da ich die Erfahrung gemacht habe, dass sich das Fenster vom LUA-Editor nicht verschieben lässt (lag natürlich direkt über den Schranken).

Das Fenster zum editieren des LUA-Skriptes öffnen wir mit der folgender Schaltfläche:



Als nächstes den Quelltext des Skripts im Ganzen. Den erläutere ich dann Zeile für Zeile.

```

clearlog()

print("Hey let's start, EEP Version is: ", EEPVer)
function EEPMain()
  return 1
end

-- Schliesst die Schranken
function CloseSchranke()
  EEPSetSignal(21,2)  -- 21=Signal-ID, 2 = schliessen
  EEPSetSignal(22,2)  -- 22=Signal-ID, 2 = schliessen
end

--oeffnet die Schranken
function OpenSchranke()
  EEPSetSignal(21,1)  -- 21=Signal-ID, 1 = oeffnen
  EEPSetSignal(22,1)  -- 22=Signal-ID, 1 = oeffnen
end

```

Bevor ich die Zeilen erkläre, möchte ich erst einige Dinge zu Funktionen sagen, die hier benötigt werden. Der grundsätzliche Aufbau einer Funktion sieht wie folgt aus:

```

function Funktionsname()
end

```

Eine Funktion beginnt immer mit dem Schlüsselwort **function**. Anschließend kommt nach einem Leerzeichen der Funktionsname, mit dem wir später die Funktion aufrufen. Dieser Name darf nur aus Buchstaben (groß und klein), Ziffern und dem Unterstrich bestehen. Alles andere ist nicht erlaubt und führt zu Fehlern. Dann kommen noch die Klammern, die in der einfachsten Form so wie gezeigt ohne etwas dazwischen sein müssen.

Auch ist wichtig, dass Gross- und Kleinschreibung beim Aufruf der Funktion beachtet werden muss. **Open** ist in dem Fall nicht gleich **open**.

Jede Funktion endet mit dem Schlüsselwort **end**.

Nun zurück zur Problematik. Ein Zug soll vor den Schranken diese schließen und nach der Durchfahrt diese wieder öffnen. Wir brauchen also 2 Funktionen. Eine zum Schließen und eine zum Öffnen der Schranken.

Im Skript haben wir dafür (in der Reihenfolge zuerst schließen und dann öffnen):

```

function CloseSchranke() und function OpenSchranke().

```

Die Funktionsrümpfe, die übrigens ohne Fehler laufen würden sehen so aus.

```

function CloseSchranke()
end

```

```

function OpenSchranke()
end

```

Wie gesagt, würden die ohne Fehler laufen. Nur passieren würde da noch nichts.

Jetzt müssen wir noch sagen was die Funktionen machen sollen. Dazu bedienen wir uns der Funktion `EEPSetSignal`.

Diese sieht allgemein so aus: `EEPSetSignal(Signal-ID,Signalstellung)`

Im Handbuch werden die zu übergebenden Werte für die Signalstellung nicht direkt beschrieben. Das sieht man erst wenn man sich das Beispiel anschaut.

Für die Schranken gelten folgende Werte:

1 = öffnen Schranke
2 = schließen Schranke

Der Befehl zum Schließen der Schranke mit der ID 21 muss demzufolge so lauten:

```
EEPSetSignal(21,2)
```

Die komplette Funktion mit den Schranken ID's 21 und 22 dann so:

```
function CloseSchranke()  
    EEPSetSignal(21,2)  
    EEPSetSignal(22,2)  
end
```

Die komplette Funktion zum Öffnen:

```
function OpenSchranke()  
    EEPSetSignal(21,1)  
    EEPSetSignal(22,1)  
end
```

Die restlichen Zeilen des Skripts.

`clearlog()` Löscht jeden Text im EEP Ereignisfenster, könnte man drauf verzichten, wenn man das Fenster nicht zum Testen braucht

`print(" ... ",EEPVer)` Gibt den Text zwischen " und " plus der EEP-Version (EEPVer) im EEP-Ereignisfenster aus (kann man also auch drauf verzichten)

```
function EEPMain()  
    return 1  
end
```

Bitte immer drin lassen. Dies ist die Einsprungsfunktion. Wer schon mal in C programmiert hat, weiß das. Diese Grundversion der Funktion reicht, wenn man nicht besondere Dinge machen will. Hier könnte man z.B. Variablen mit Startwerten versehen (initialisieren), die im gesamten Skript verwendet werden (globale Variablen). Mehr möchte ich jetzt nicht drauf eingehen. Das wäre etwas für ein anderes Thema.

Was jetzt noch benötigt wird ist ein Fahrzeugkontaktpunkt zum Schließen und einer zum Öffnen der Schranken.

Kontaktpunkt zum Schließen:

Kontaktpunkt für Fahrzeug ✕

Auslösen bei Zugvorbeifahrt:

Richtung eins Aktivierung verzögert:
 Richtung zwei Aktivierungsdistanz:
 Zugschluss

Für Route: Ist-Zustand: Jeder:

Für Zug: Filtername:

Wenn Signal/Weiche # ist

Lua Funktion: CloseSchranke

Auf Fahrzeug/Zug anwenden:

Auswahl der Route:

Steuerung von Achsen

Achse: Filter: Position:

Kupplungskontrolle

Kupplung lösen: vorne Kupplung vorne
 hinten Kupplung hinten

Rollmaterial-Nr. Neuer Zugname:

Belade- und Entladekontrolle der Fahrzeuge

Rollmaterial bei nächster Gleisverbindung anschließen
 Folgende Anzahl an Rollmaterialien entladen:

Geschwindigkeitsüberwachung

Vormerken/Wiederherstellen
 Fahrtrichtungsumkehr
 nicht langsamer als
 nicht schneller als
 festlegen
 Geschwindigkeit (km/h):

Weitere Optionen

Haken an/aus
 Rauch an/aus
 Wamsignal geben
 Blinker links an/aus
 Blinker rechts an/aus
 Lichter an/aus
 Staub aufwirbeln

Der Aufruf erfolgt mit dem Funktionsnamen CloseSchranke ohne Klammern.

Kontaktpunkt zum Öffnen:

Kontaktpunkt für Fahrzeug ✕

Auslösen bei Zugvorbeifahrt:

Richtung eins Aktivierung verzögert:
 Richtung zwei Aktivierungsdistanz:
 Zugschluss

Für Route: Ist-Zustand: Jeder:

Für Zug: Filtername:

Wenn Signal/Weiche # ist

Lua Funktion:

Auf Fahrzeug/Zug anwenden:

Auswahl der Route:

Steuerung von Achsen

Achse: Filter: Position:

Kupplungskontrolle

Kupplung lösen: vorne Kupplung vorne
 hinten Kupplung hinten

Rollmaterial-Nr. Neuer Zugname:

Belade- und Entladekontrolle der Fahrzeuge

Rollmaterial bei nächster Gleisverbindung anschließen

Folgende Anzahl an Rollmaterialien entladen:

Geschwindigkeitsüberwachung

Vormerken/Wiederherstellen
 Fahrtrichtungsumkehr
 nicht langsamer als
 nicht schneller als
 festlegen
 Geschwindigkeit (km/h):

Weitere Optionen

Haken an/aus
 Rauch an/aus
 Wamsignal geben
 Blinker links an/aus
 Blinker rechts an/aus
 Lichter an/aus
 Staub aufwirbeln

Das war es auch schon. Viel Spaß beim Ausprobieren.