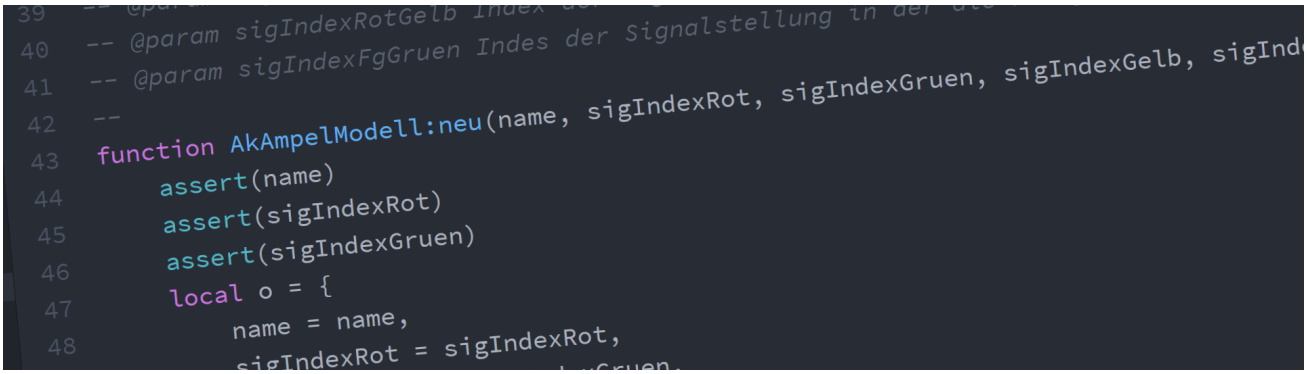
Ampelkreuzung automatisch steuern



Inhalt

Was Du lernst

- Diese Anleitung zeigt Dir, wie Du in EEP eine mit Ampeln versehene Kreuzung mit dem Skript verdrahten kannst.

Was Du brauchst

- **EEP 14** - das Programm - *Download*
- **Ak-Lua-Skripte-für-EEP** - Eine Sammlung verschiedener Lua Skripte für EEP
  So kannst Du die Skripte installieren - Installation von "Ak-Lua-Skripte-für-EEP"
- **Einen Editor für Lua-Skripte** - ein Editor Deiner Wahl, z.B. Notepad++
- **Zettel und Stift** - *Download Kreuzungsaufbau.pdf*
- **Eine Anlage mit fertigen Kreuzungen und vorinstallierten Ampeln**
  Extra für diese Anleitung wurde eine Anlage erstellt: _Andreas_Kreuz_Anleitung_Ampel_. Wenn Du diese Anlage verwenden möchtest, brauchst Du folgende Modelle:
  - 1Spur-Großstadtstraßen-System-Grundset (V10NAS30002) - *Download*
  - 1Spur-Ergänzungsset - *Download*
  - Ampel-Baukasten für mehrspurige Straßenkreuzungen (V80NJS20039) - *Download*
  - Straßenbahnsignale als Immobilien (V80MA1F010 und V10MA1F011) - *Download*

Checkliste

- [ ] Skriptsammlung "Ak-Lua-Skripte-für-EEP" ist installiert
- [ ] Du hast eine Anlage mit einer Ampelkreuzung oder "Andreas_Kreuz_Anleitung_Ampel" geöffnet

Los geht's

- Öffne die Anlage in EEP
- Öffne Deinen Editor

Das Lua-Hauptskript anlegen

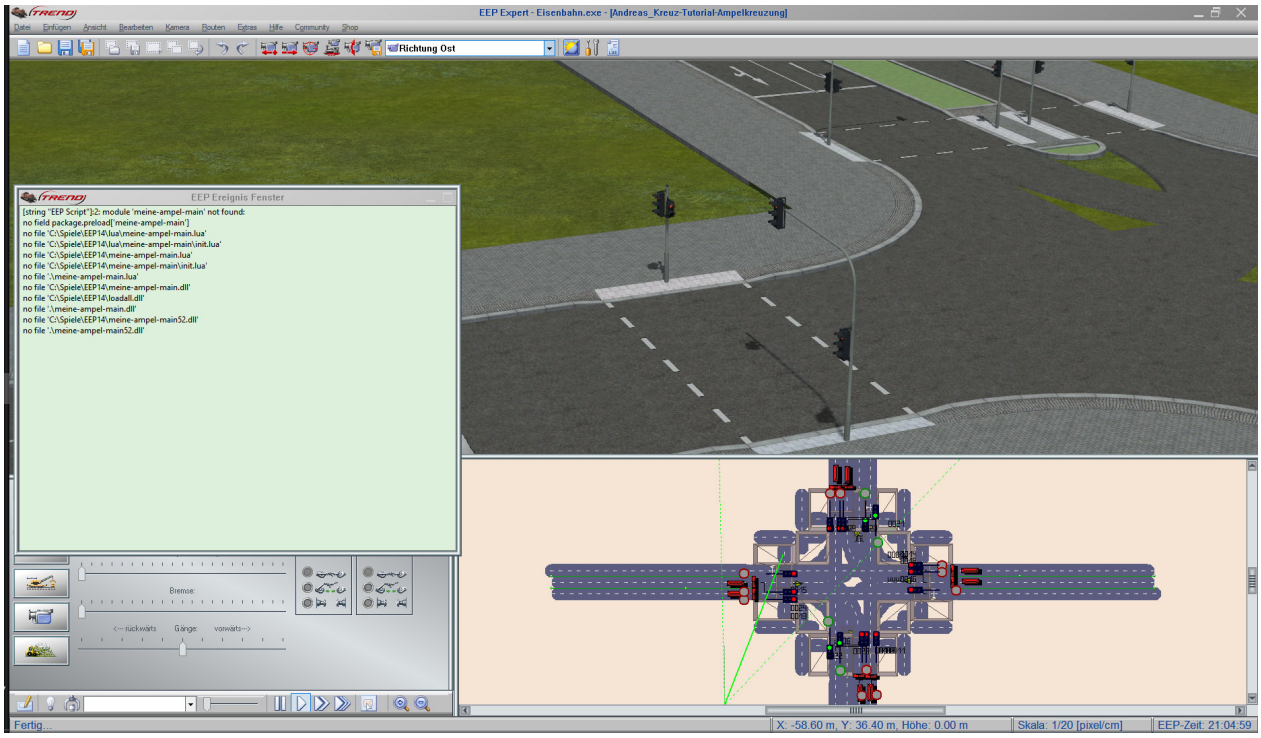
Tipp: *Aktiviere in EEP unter Programmeinstellungen das EEP Ereignisfenster, damit Du die Lua Meldungen lesen kannst.*

Tipp: *Diese Anleitung geht davon aus, dass in der geöffneten Anlage noch nichts mit LUA gemacht wurde. Verwendest Du Dein eigenes Anlagen-Skript, dann lösche es nicht, sondern ergänze es um die weiter unten aufgeführten Befehle.*

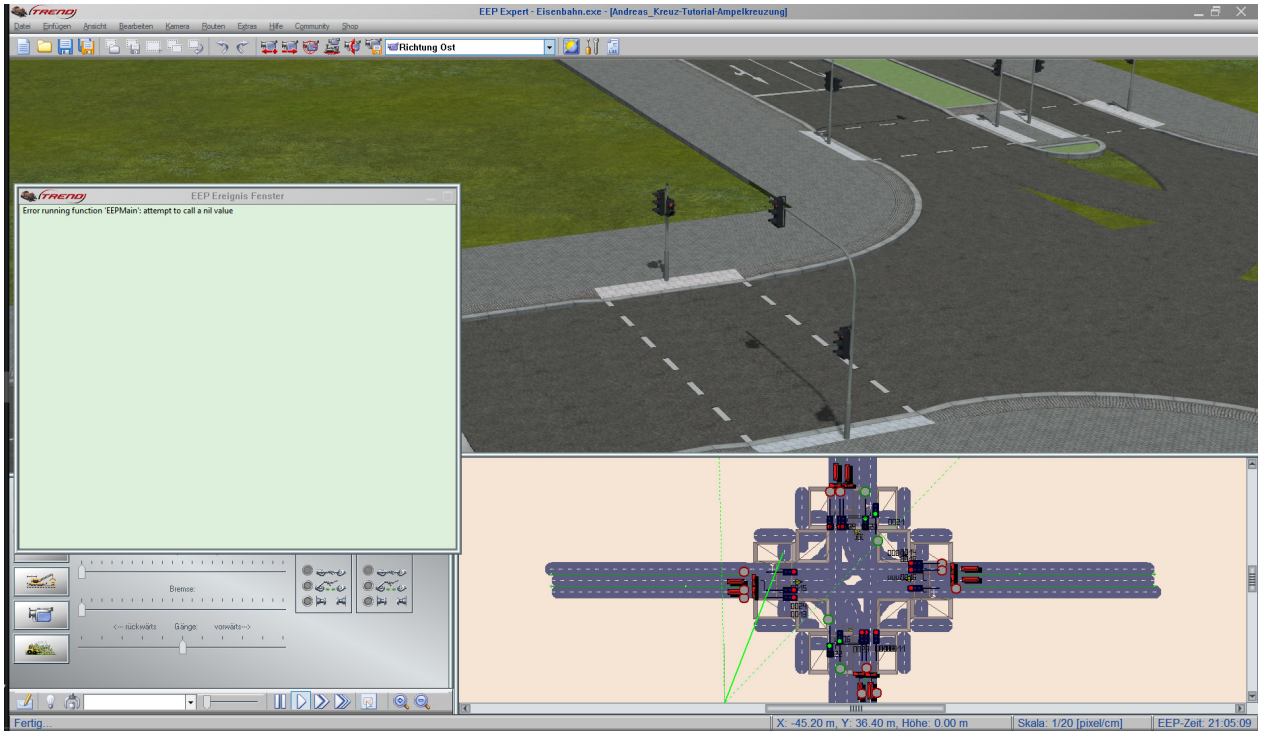
- Das Editieren der Skripte findet im LUA-Verzeichnis von EEP statt, z.B. in `C:\Trend\EEP14\LUA`
- Öffne den LUA-Editor in EEP, wähle alles mit `<Strg> + <A>` aus und ersetze es durch

```
clearlog()
require("meine-ampel-main")
```

- Klicke danach in EEP auf Skript neu laden und wechsle in den 3D-Modus.
  Wenn Du alles richtig gemacht hast, erscheint im Log eine Fehlermeldung, dass `meine-ampel-main.lua` nicht gefunden werden kann.



- Lege nun das Haupt-Skript an `C:\Trend\EEP14\LUA\meine-ampel-main.lua` im Verzeichnis `LUA` an
  Dies wird das Skript werden, welches in der Anlage verwendet wird. Egal, wie Deine Anlage heißt.
- Klicke danach in EEP auf Skript neu laden und wechsle in den 3D-Modus. **Wenn Du alles richtig gemacht hast**, erscheint eine Fehlermeldung, dass `meine-ampel-main.lua` nicht gefunden werden kann.

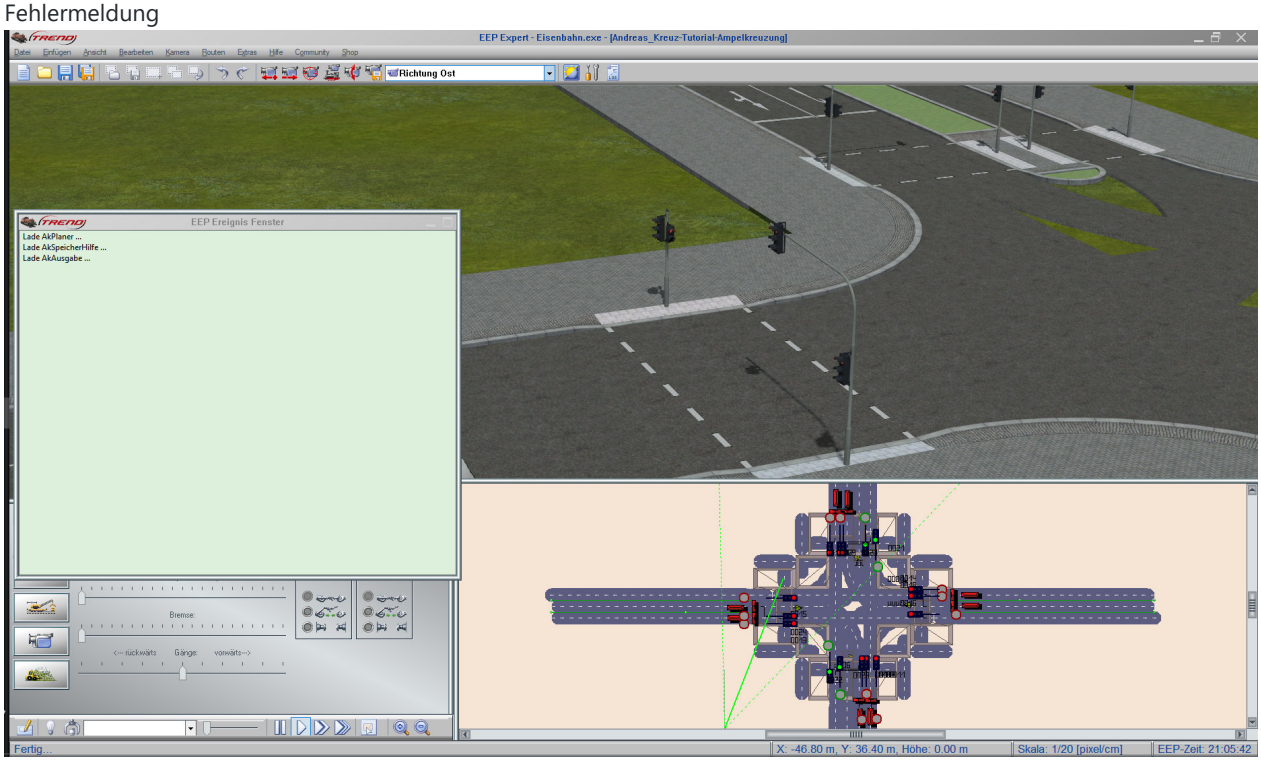


Notwendige Befehle in das Lua-Skript aufnehmen

- Ergänze das Lua-Hauptskript um die folgenden Zeilen.

```
require("ak.strasse.AkStrasse")

-- Hier kommt der Code

function EEPMain()
    AkKreuzung:planeSchaltungenEin()
    AkPlaner:fuehreGeplanteAktionenAus()
    return 1
end
```

- Klicke danach auf Skript neu laden und wechsle in den 3D-Modus. **Wenn Du alles richtig gemacht hast**, verschwindet die Fehlermeldung



Was ist grade passiert?

- Die Zeile `require("ak.strasse.AkStrasse")` sorgt dafür, daß die Datei `ak/strasse/AkStrasse.lua` einmal eingelesen wird. Nach diesem Aufruf stehen Dir alle Funktionen dieser Datei zur Verfügung.
- Die Zeile `AkKreuzung:planeSchaltungenEin()` ist für das Einplanen aller Schaltvorgänge der Kreuzungen notwendig. Diese Vorgänge werden als Aktionen im Planer hinterlegt.
- Die Zeile `AkPlaner:fuehreGeplanteAktionenAus()` ist für das Ausführen aller geplanten Aktionen notwendig.
- Wichtig ist auch, dass die Funktion EEPMain mit `return 1` beendet wird, damit sie alle 200 ms aufgerufen wird.

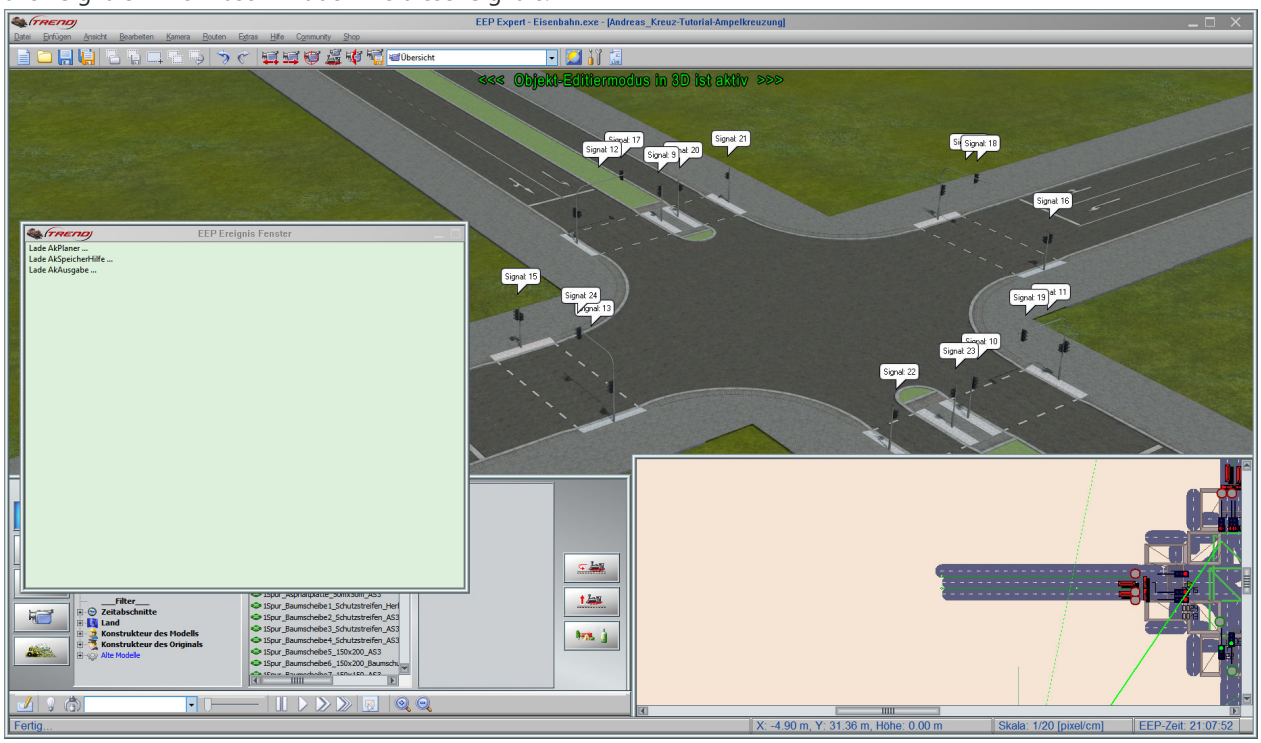
Alle Signale mit Info-Blasen markieren

Tipp: *Verwende diesen Code nicht, wenn Du Deine Anlagen manuelle Info-Blasen mit* `EEPShowSignalInfo(...)` *an Deinen Signalen anzeigst. Denn all diese Info-Blasen werden gelöscht.*

- Um die Signale (in dem Fall Ampeln) der Kreuzung zu bearbeiten ist es am einfachsten, wenn Du die Signal-IDs aller Signale in Info-Blasen anzeigst. - Füge die Zeile `AkKreuzung.zeigeSignalIdsAllerSignale = true` und danach `AkKreuzung.zeigeSchaltungAlsInfo = true` vor der EEPMain()-Methode hinzu:

```
-- Hier kommt der Code
AkKreuzung.zeigeSignalIdsAllerSignale = true
AkKreuzung.zeigeSchaltungAlsInfo = true
```

- Klicke danach auf Skript neu laden und wechsle in den 3D-Modus. **Wenn Du alles richtig gemacht hast**, siehst Du an allen Signalen Info-Blasen mit den IDs dieser Signale.



Was ist grade passiert?

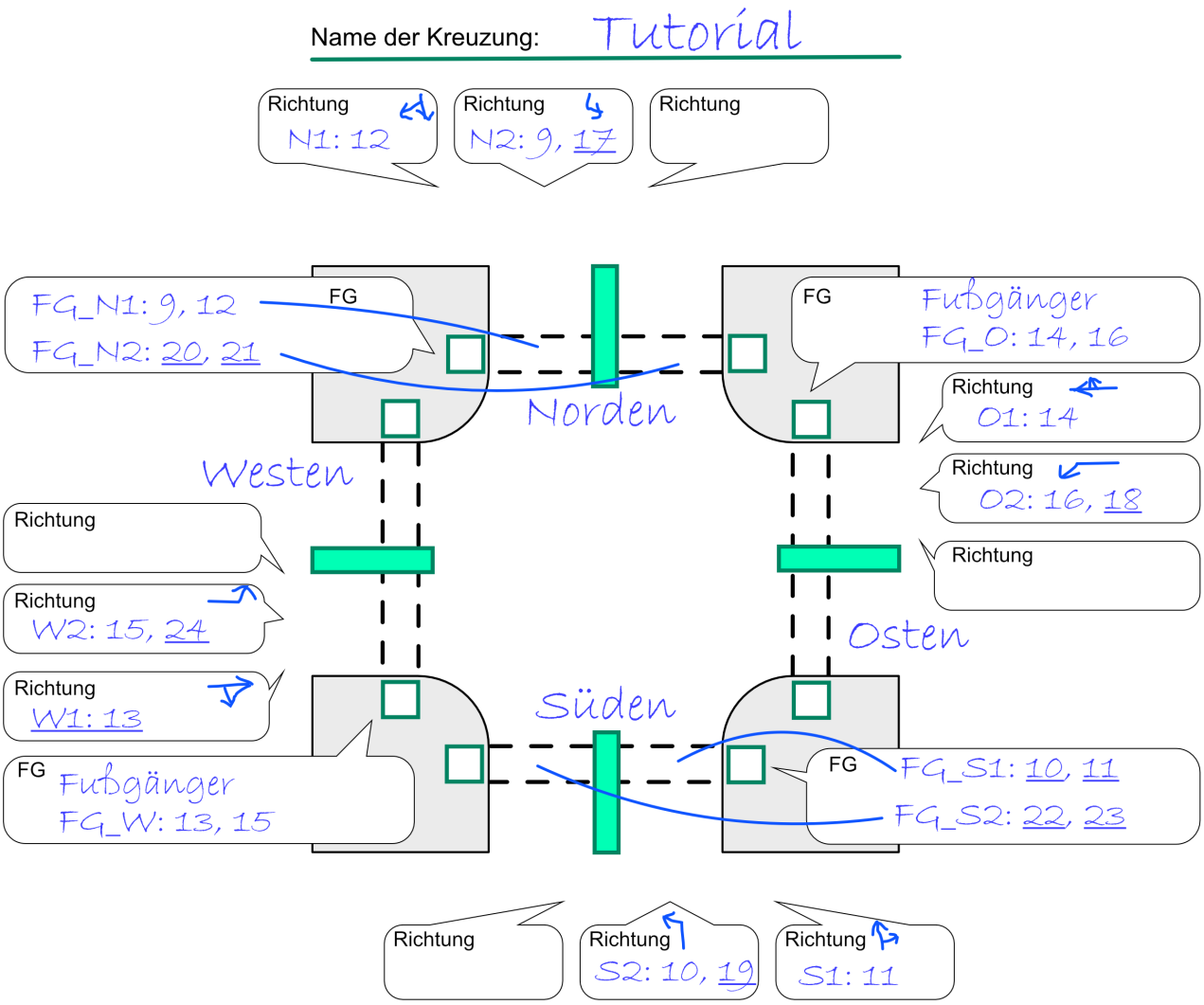
- Das neu Laden der Anlage hat dafür gesorgt, dass die Lua-Skripte anhand der Variable alle Signale von 1 bis 1000 mit einer Info-Blase versehen haben.

Die Richtungen und Signal-IDs der Kreuzung notieren

Tipp: *Das PDF-Dokument hilft Dir deine Kreuzung zu notieren.*

Notiere Dir, welche *Richtungen* es gibt und wie die IDs der zu schaltenden Ampeln heißen - merke Dir dabei, welche unterschiedlichen Ampelmodelle eingesetzt werden. In der Beispielanlage sind es:

- Kombinierte Fußgänger- und Strassenverkehrsampeln
- Reine Fußgängerampeln *(die sind in der Skizze bei "FG" unterstrichen)*
- Strassenverkehrsampeln *(die sind in der Skizze bei "Richtung" unterstrichen)*



Was ist eine *Richtung*: In diesem Abschnitt wird viel von *Richtungen* geredet. So eine *Richtung* fasst mehrere Fahrspuren, die den selben Weg nehmen zusammen und schaltet alle Ampeln, die diesen Weg freigeben.

- **Fasse mehrere Fahrspuren in die den selben Weg nehmen immer zu einer *Richtung* zusammen** (also mehrere Linksabbieger-, Geradeaus- **oder** Rechtsabbieger-Spuren), denn Spuren in die selbe *Richtung* werden immer gemeinsam geschaltet.
- **Erstelle immer eigene Fahrspuren und *Richtungen* für Linksabbieger, denn diese achten nicht auf den Gegenverkehr**. Darum solltest Du den Linksabbieger-Fahrspuren immer eine eigene *Richtung* geben und diese nur dann auf grün schalten, wenn der Gegenverkehr den Fahrweg der Linksabbieger nicht kreuzen kann.
  - **Alternative:** Du kannst auch mit unsichtbaren Ampeln arbeiten, die bei Gegenverkehr auf rot schalten. Dies musst Du jedoch selbst machen.
- **Du kannst getrennte Rechtsabbieger-Fahrspuren und Geradeaus-Fahrspuren zu einer *Richtung* zusammenfassen.**
- Spendiere Deinen Fußgängerampeln eigene *Richtungen*, denn diese werden von der Automatik eher auf rot geschaltet.

Erst im nächsten Schritt werden mehrere dieser *Richtungen* zu Schaltungen zusammengefasst.

Schreibe die Richtungen in das Haupt-Skript

Tipp: *Für jede Ampel musst Du den* `AkAmpelModell` *kennen, da sich die Signalstellungen in EEP unterscheiden. Weitere Informationen findest Du unter: Unterstütze weitere Ampeln in AkAmpelModell*

Schreibe nun die einzelnen Richtungen in das Haupt-Skript. Jede Richtung muss dabei eine noch nicht verwendete Speicher-ID zwischen 1 und 1000 bekommen.

```
--------------------------------------------------------------------------------
-- Definiere die Richtungen fuer die Kreuzung
--------------------------------------------------------------------------------

--   +------------------------------------------- Neue Richtung
```

```
--   |             +------------------------------ Name der Richtung
--   |             |     +------------------------ Speicher ID - um die Anzahl der Fahrzeuge
--   |             |     |                                       und die Wartezeit zu speichern
--   |             |     |      +----------------- neue Ampel für diese Richtung
--   |             |     |      |           +------ Signal-ID dieser Ampel
--   |             |     |      |           |   +-- Modell kann rot, gelb, gruen und FG schalten
n1 = AkRichtung:neu("N1", 100, { AkAmpel:neu(12, AkAmpelModell.JS2_3er_mit_FG) })

-- Die Richtung N2 hat zwei Ampeln fuer's Linksabbiegen, 9 mit Fussgaengerampel und 17 ohne
n2 = AkRichtung:neu("N2", 101, {
    AkAmpel:neu(9, AkAmpelModell.JS2_3er_mit_FG),
    AkAmpel:neu(17, AkAmpelModell.JS2_3er_ohne_FG)
})

-- Die Richtungen für Fussgaenger haben auch je zwei Ampeln
fg_n1 = AkRichtung:neu("FG_N1", 102, {
    AkAmpel:neu(9, AkAmpelModell.JS2_3er_mit_FG), -- Wird geteilt mit N2
    AkAmpel:neu(12, AkAmpelModell.JS2_3er_mit_FG) -- Wird geteilt mit N1
})
fg_n2 = AkRichtung:neu("FG_N2", 103, {
    AkAmpel:neu(20, AkAmpelModell.JS2_2er_nur_FG),
    AkAmpel:neu(21, AkAmpelModell.JS2_2er_nur_FG),
})

-- Richtungen im Osten
o1 = AkRichtung:neu("O1", 100, { AkAmpel:neu(14, AkAmpelModell.JS2_3er_mit_FG) })
o2 = AkRichtung:neu("O2", 100, {
    AkAmpel:neu(16, AkAmpelModell.JS2_3er_mit_FG),
    AkAmpel:neu(18, AkAmpelModell.JS2_3er_ohne_FG)
})
fg_o = AkRichtung:neu("FG_O", 102, {
    AkAmpel:neu(14, AkAmpelModell.JS2_3er_mit_FG), -- Wird geteilt mit O1
    AkAmpel:neu(18, AkAmpelModell.JS2_3er_mit_FG) -- Wird geteilt mit O2
})

-- Richtungen im Sueden
s1 = AkRichtung:neu("S1", 100, { AkAmpel:neu(11, AkAmpelModell.JS2_3er_mit_FG) })
s2 = AkRichtung:neu("S2", 101, {
    AkAmpel:neu(10, AkAmpelModell.JS2_3er_mit_FG),
    AkAmpel:neu(19, AkAmpelModell.JS2_3er_ohne_FG)
})
fg_s1 = AkRichtung:neu("FG_S1", 102, {
    AkAmpel:neu(10, AkAmpelModell.JS2_3er_mit_FG), -- Wird geteilt mit S2
    AkAmpel:neu(11, AkAmpelModell.JS2_3er_mit_FG) -- Wird geteilt mit S1
})
fg_s2 = AkRichtung:neu("FG_S2", 103, {
    AkAmpel:neu(22, AkAmpelModell.JS2_2er_nur_FG),
    AkAmpel:neu(23, AkAmpelModell.JS2_2er_nur_FG),
})

-- Richtungen im Westen
w1 = AkRichtung:neu("W1", 100, { AkAmpel:neu(13, AkAmpelModell.JS2_3er_mit_FG) })
w2 = AkRichtung:neu("W2", 100, {
    AkAmpel:neu(15, AkAmpelModell.JS2_3er_mit_FG),
    AkAmpel:neu(24, AkAmpelModell.JS2_3er_ohne_FG)
})
fg_w = AkRichtung:neu("FG_W", 102, {
    AkAmpel:neu(13, AkAmpelModell.JS2_3er_mit_FG), -- Wird geteilt mit O1
    AkAmpel:neu(15, AkAmpelModell.JS2_3er_mit_FG) -- Wird geteilt mit O2
})
```

- Klicke danach im LUA-Editor von EEP auf "Skript neu laden" und wechsle in den 3D-Modus. **Wenn Du alles richtig gemacht hast**, siehst Du weiterhin an allen Signalen Info-Blasen mit den IDs dieser Signale und keine Fehlermeldung im Log.

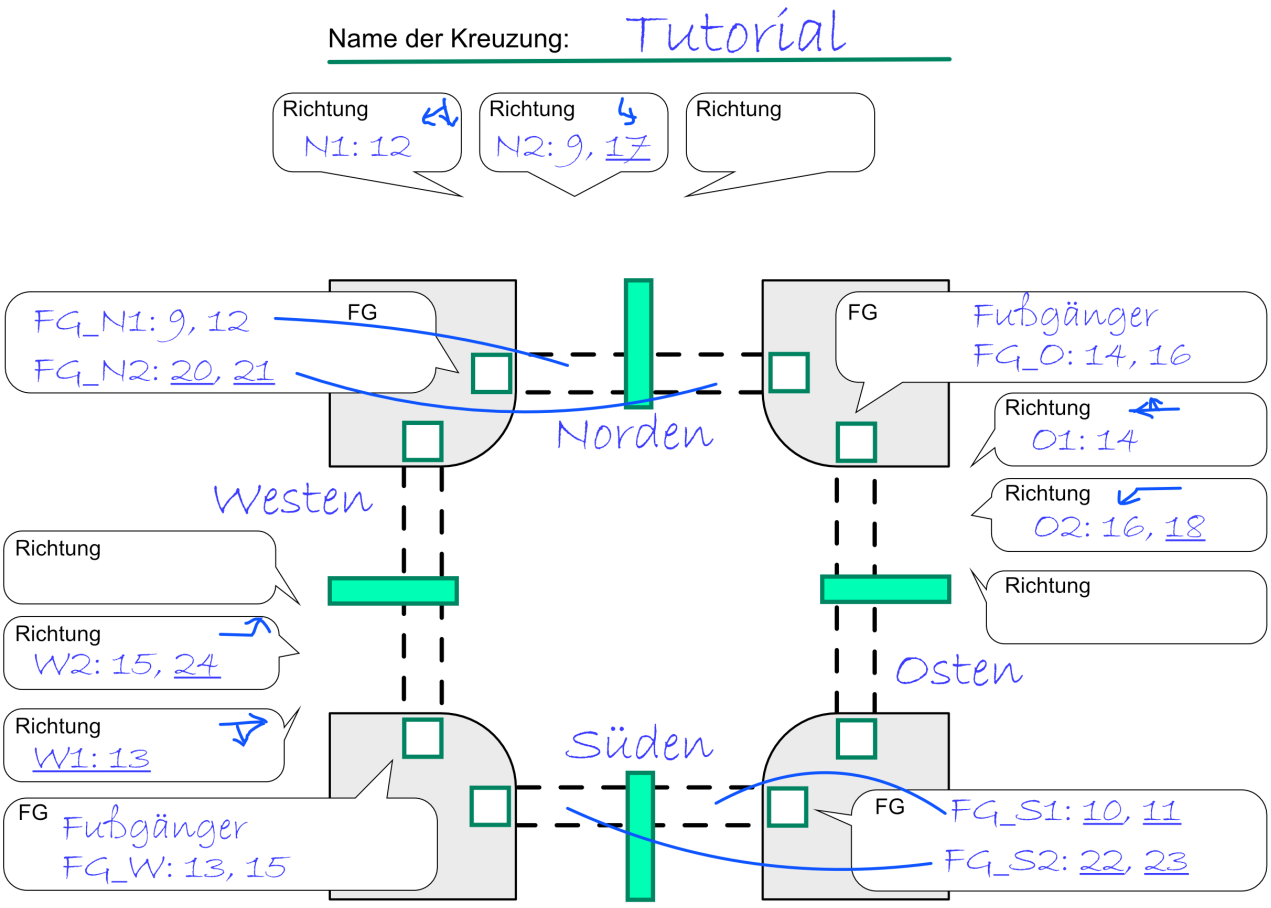
Was ist grade passiert?

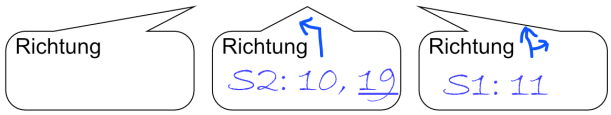
- Du hast soeben die Richtungen der Kreuzung festgelegt. Jede kann für sich allein geschaltet werden oder zusammen mit anderen Richtungen. Dazu dient `AkKreuzungsSchaltung`, welches im nächsten Schritt zum Einsatz kommt.

Fasse die Richtungen nun zu Schaltungen zusammen

Notiere Dir, welche der *Richtungen* zu *Schaltungen* zusammengefasst werden sollen.

Beachte: Eine *Schaltung* darf nur *Richtungen* enthalten, die sich nicht gegenseitig überlappen.





Richtungen	FG_W N1	N2	FG_N1 FG_N2 O1	FG_S1 O2	FG_O S1	S2	FG_S1 FG_S2 W1	FGN1 W2
Schaltung 1	X				X			
Schaltung 2		X				X		
Schaltung 3			X				X	
Schaltung 4				X				X
Schaltung 5	X	X						
Schaltung 6			X	X				
Schaltung 7					X	X		
Schaltung 8							X	X
Schaltung								
Schaltung								
Schaltung								

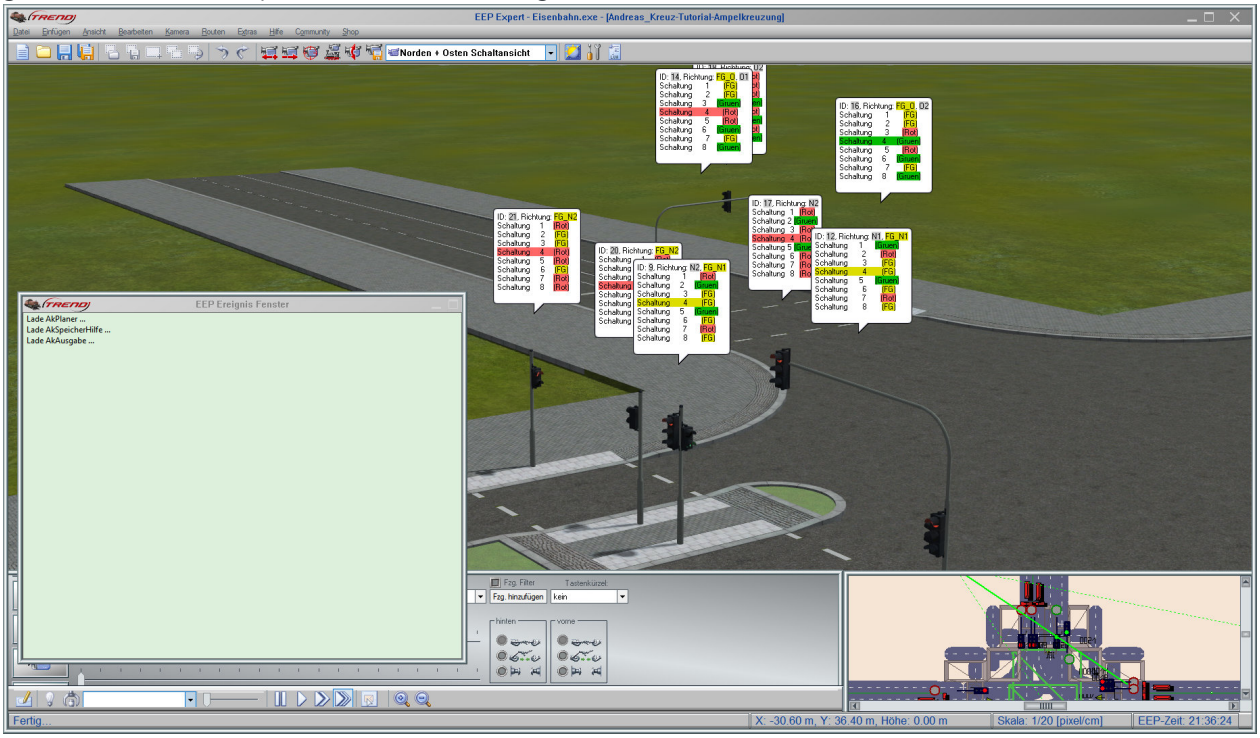
Im Beispiel siehst Du, dass Richtungen in mehreren Schaltungen enthalten sein können.

Schreibe die Schaltungen in das Haupt-Skript

```
-----------------------------------------------------------------
-- Definiere die Schaltungen und die Kreuzung
-----------------------------------------------------------------
-- Eine Schaltung bestimmt, welche Richtungen gleichzeitig auf
-- grün geschaltet werden dürfen, alle anderen sind rot

--- Tutorial 1: Schaltung 1
local sch1 = AkKreuzungsSchaltung:neu("Schaltung 1")
sch1:fuegeRichtungHinzu(n1)
sch1:fuegeRichtungHinzu(s1)
sch1:fuegeRichtungFuerFussgaengerHinzu(fg_o)
sch1:fuegeRichtungFuerFussgaengerHinzu(fg_w)

--- Tutorial 1: Schaltung 2
local sch2 = AkKreuzungsSchaltung:neu("Schaltung 2")
sch2:fuegeRichtungHinzu(n2)
sch2:fuegeRichtungHinzu(s2)
sch2:fuegeRichtungFuerFussgaengerHinzu(fg_n2)
sch2:fuegeRichtungFuerFussgaengerHinzu(fg_o)
sch2:fuegeRichtungFuerFussgaengerHinzu(fg_w)
sch2:fuegeRichtungFuerFussgaengerHinzu(fg_s2)

--- Tutorial 1: Schaltung 3
local sch3 = AkKreuzungsSchaltung:neu("Schaltung 3")
sch3:fuegeRichtungHinzu(o1)
sch3:fuegeRichtungHinzu(w1)
sch3:fuegeRichtungFuerFussgaengerHinzu(fg_n1)
sch3:fuegeRichtungFuerFussgaengerHinzu(fg_n2)
sch3:fuegeRichtungFuerFussgaengerHinzu(fg_s1)
sch3:fuegeRichtungFuerFussgaengerHinzu(fg_s2)

--- Tutorial 1: Schaltung 4
local sch4 = AkKreuzungsSchaltung:neu("Schaltung 4")
sch4:fuegeRichtungHinzu(o2)
sch4:fuegeRichtungHinzu(w2)
sch4:fuegeRichtungFuerFussgaengerHinzu(fg_n1)
sch4:fuegeRichtungFuerFussgaengerHinzu(fg_s1)

--- Tutorial 1: Schaltung 5
local sch5 = AkKreuzungsSchaltung:neu("Schaltung 5")
sch5:fuegeRichtungHinzu(n1)
sch5:fuegeRichtungHinzu(n2)
sch5:fuegeRichtungFuerFussgaengerHinzu(fg_w)

--- Tutorial 1: Schaltung 6
local sch6 = AkKreuzungsSchaltung:neu("Schaltung 6")
sch6:fuegeRichtungHinzu(o1)
sch6:fuegeRichtungHinzu(o2)
sch6:fuegeRichtungFuerFussgaengerHinzu(fg_n1)
sch6:fuegeRichtungFuerFussgaengerHinzu(fg_n2)
sch6:fuegeRichtungFuerFussgaengerHinzu(fg_s1)

--- Tutorial 1: Schaltung 7
local sch7 = AkKreuzungsSchaltung:neu("Schaltung 7")
sch7:fuegeRichtungHinzu(s1)
sch7:fuegeRichtungHinzu(s2)
sch7:fuegeRichtungFuerFussgaengerHinzu(fg_o)

--- Tutorial 1: Schaltung 6
local sch8 = AkKreuzungsSchaltung:neu("Schaltung 8")
sch8:fuegeRichtungHinzu(o1)
sch8:fuegeRichtungHinzu(o2)
sch8:fuegeRichtungFuerFussgaengerHinzu(fg_n1)
sch8:fuegeRichtungFuerFussgaengerHinzu(fg_s1)
sch8:fuegeRichtungFuerFussgaengerHinzu(fg_s2)


k1 = AkKreuzung:neu("Tutorial 1")
k1:fuegeSchaltungHinzu(sch1)
k1:fuegeSchaltungHinzu(sch2)
k1:fuegeSchaltungHinzu(sch3)
k1:fuegeSchaltungHinzu(sch4)
k1:fuegeSchaltungHinzu(sch5)
k1:fuegeSchaltungHinzu(sch6)
```

```
k1:fuegeSchaltungHinzu(sch7)
k1:fuegeSchaltungHinzu(sch8)
```

- Klicke danach im LUA-Editor von EEP auf "Skript neu laden" und wechsle in den 3D-Modus. **Wenn Du alles richtig gemacht hast**, siehst Du plötzlich, dass die Schaltungen zum Leben erwachen.



Was ist grade passiert?

- Du hast soeben die Richtungen zu Schaltungen zusammengefasst und diese einer Kreuzung zugewiesen. Durch die beiden am Anfang hinzugefügten Aufrufe in EEPMain() plant die Kreuzung automatisch ihre Schaltungen der Planer führt sie aus.

Schalte die Hilfsfunktionen wieder aus

Erinnerst Du Dich den Code, der die Info-Blasen zu den Signalen hinzugefügt hat?

- Wenn Du möchtest, kannst Du die Info-Blasen wieder abschalten. Entferne nicht die Zeilen, sondern setze die Werte von `true` auf `false`.

```
-- Hier kommt der Code
AkKreuzung.zeigeSignalIdsAllerSignale = false
AkKreuzung.zeigeSchaltungAlsInfo = false
```

- Klicke danach auf Skript neu laden und wechsle in den 3D-Modus. **Wenn Du alles richtig gemacht hast**, verschwinden die Info-Blasen von den Signale.

Tipp: Setze die Werte wieder auf `true`, wenn Du denkst, dass Du die Signale falsch gesetzt hast.

Herzlichen Glückwunsch!

Du hast diese Anleitung abgeschlossen.

So kannst Du weitermachen:

- Füge noch fehlende Richtungen zu Schaltungen hinzu:
  - Wenn `n2` geschaltet ist, kann immer auf `fg_n2` geschaltet werden.
  - Wenn `s2` geschaltet ist, kann immer auf `fg_s2` geschaltet werden.

Tipps

- Ampeln aufstellen

Themen für Fortgeschrittene

- Füge Kontaktpunkte und Zähler hinzu
- Füge Richtungen hinzu, die nur auf Anforderung geschaltet werden