

Beispiel: Zweigleisiger Abzweig

mit Signalen, Motorantrieben und virtuellen Fahrstraßen

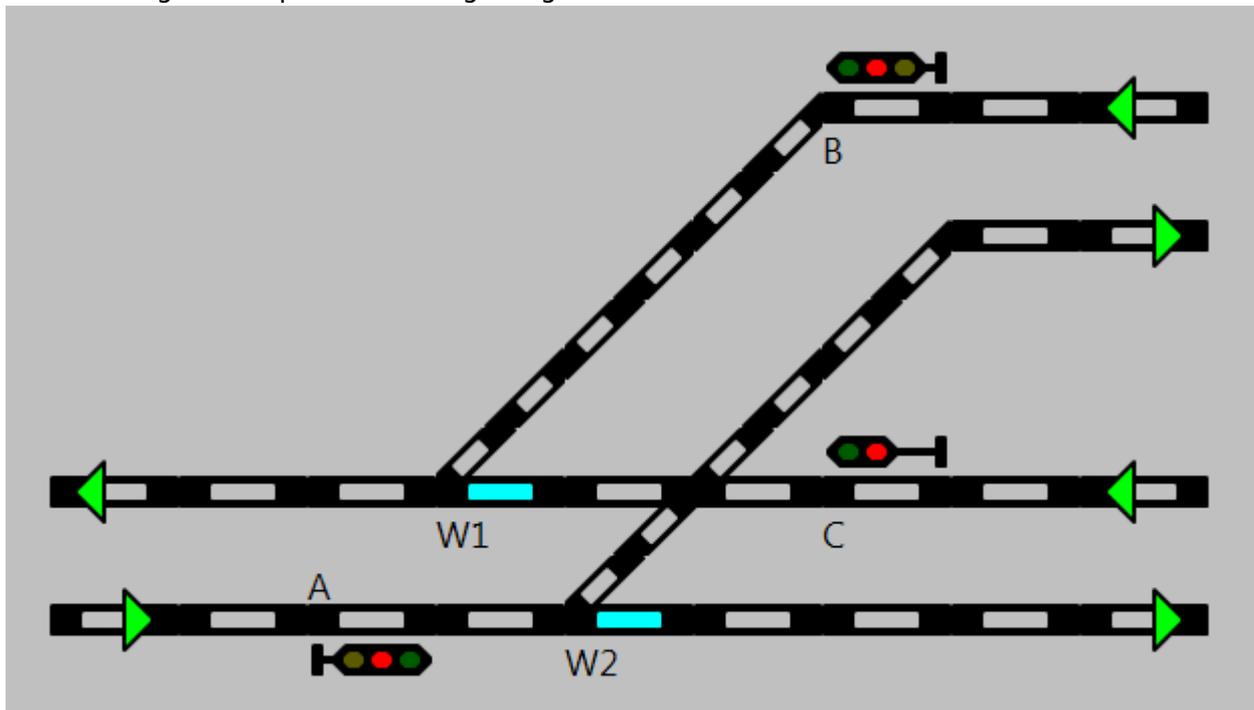
Stand: 02/2018

Anhand eines zweigleisigen Abzweigs werden in diesem Beispiel folgende Themen gezeigt:

- Ansteuerung des motorischen Weichenantrieb MP1 von MTB
- Verwendung von Flags als Zustandsspeicher
- Steuerung von virtuellen Fahrstraßen
- Sequentieller Abläufe mit Makros
- Aktionen ohne PC über Schalter auslösen

Der Abzweig wird vollständig von nur einer einzigen LightControl gesteuert, was diese Lösung auch für Modulbahner interessant macht.

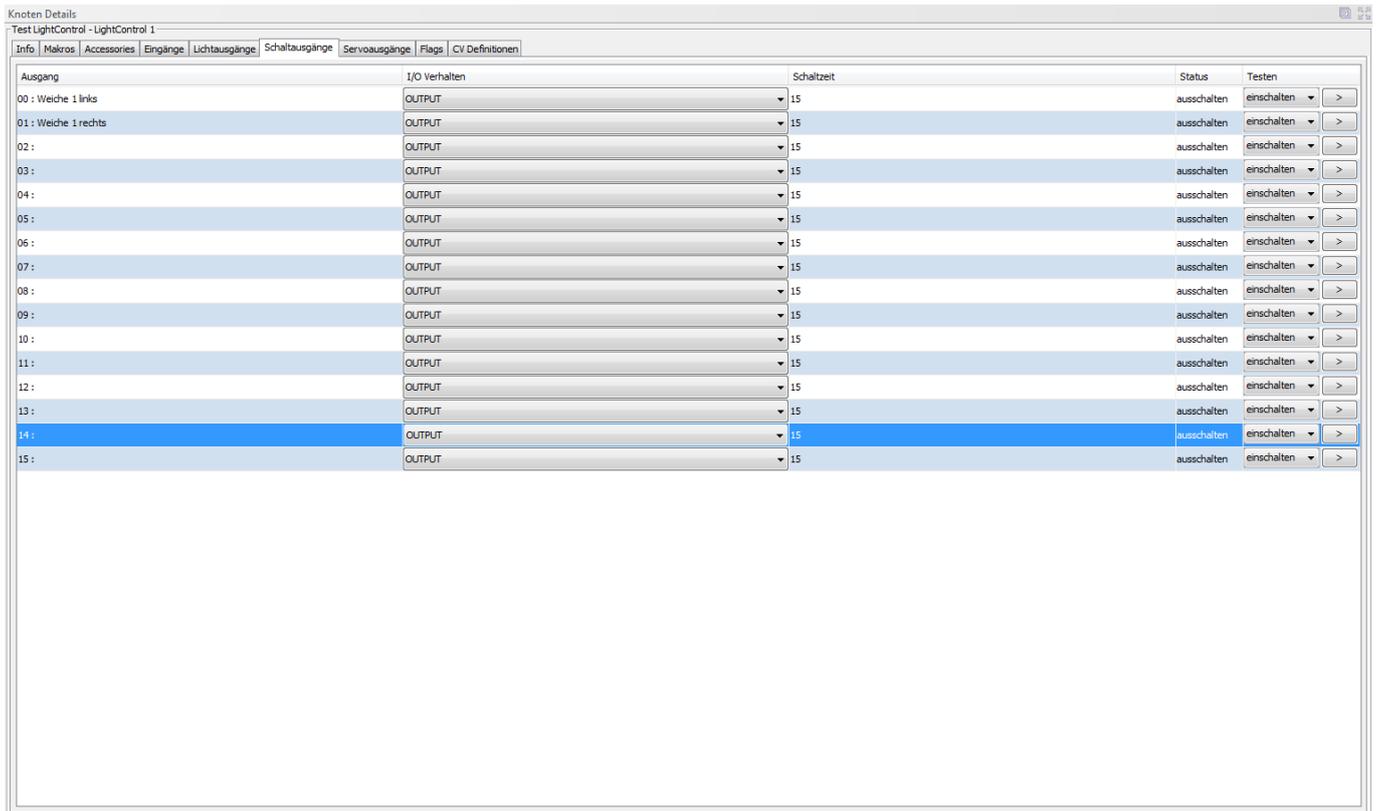
Der Abzweig im Beispiel ist wie folgt aufgebaut:



Grundkonfiguration

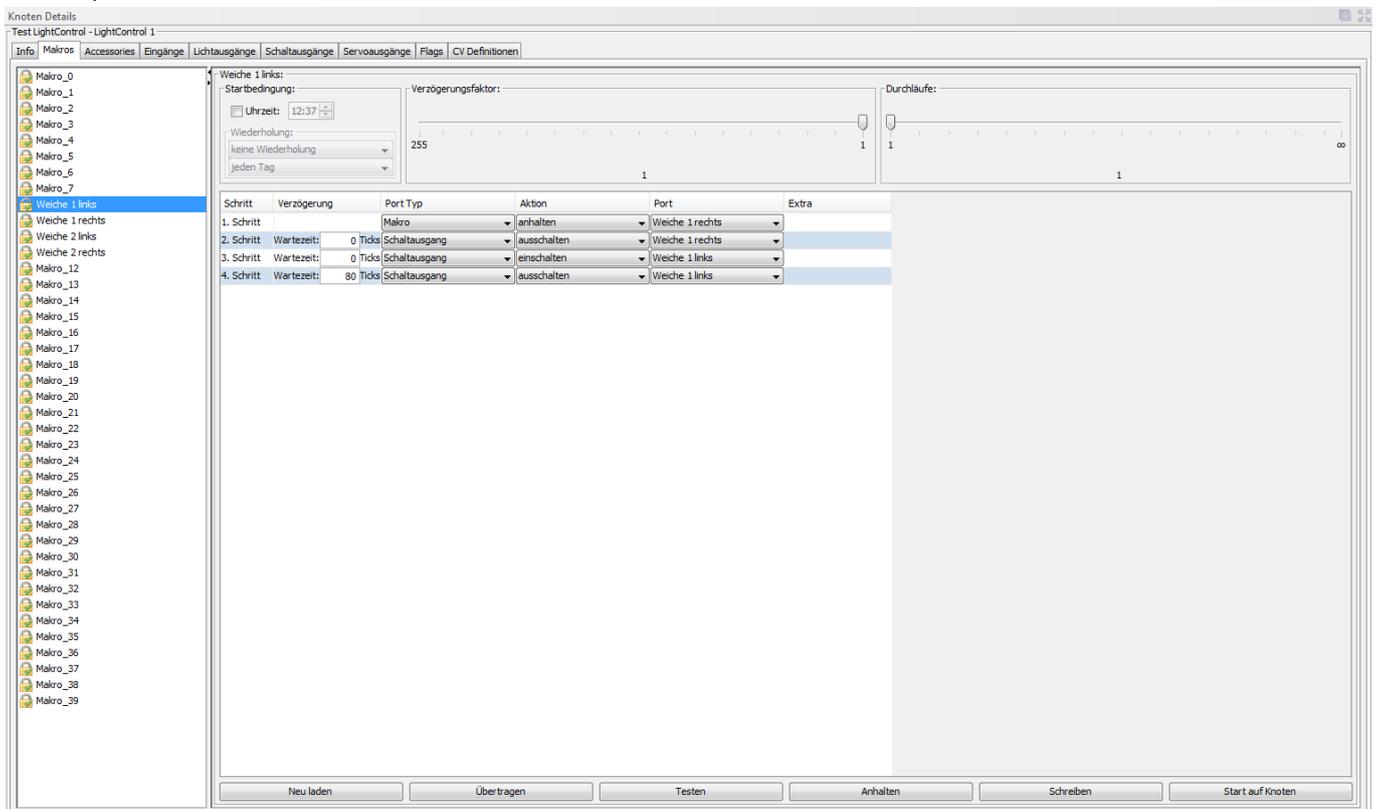
Weichen

Im ersten Schritt werden von Hand die Schaltausgänge für die Motorantriebe konfiguriert:



(Alle Bilder des Wizards zeigen die simulierte LightControl und sehen deshalb teilweise etwas anders aus)

Die zugehörigen Makros werden ab Makronummer 8 angelegt (warum nicht ab Nummer 0 wird später erklärt)

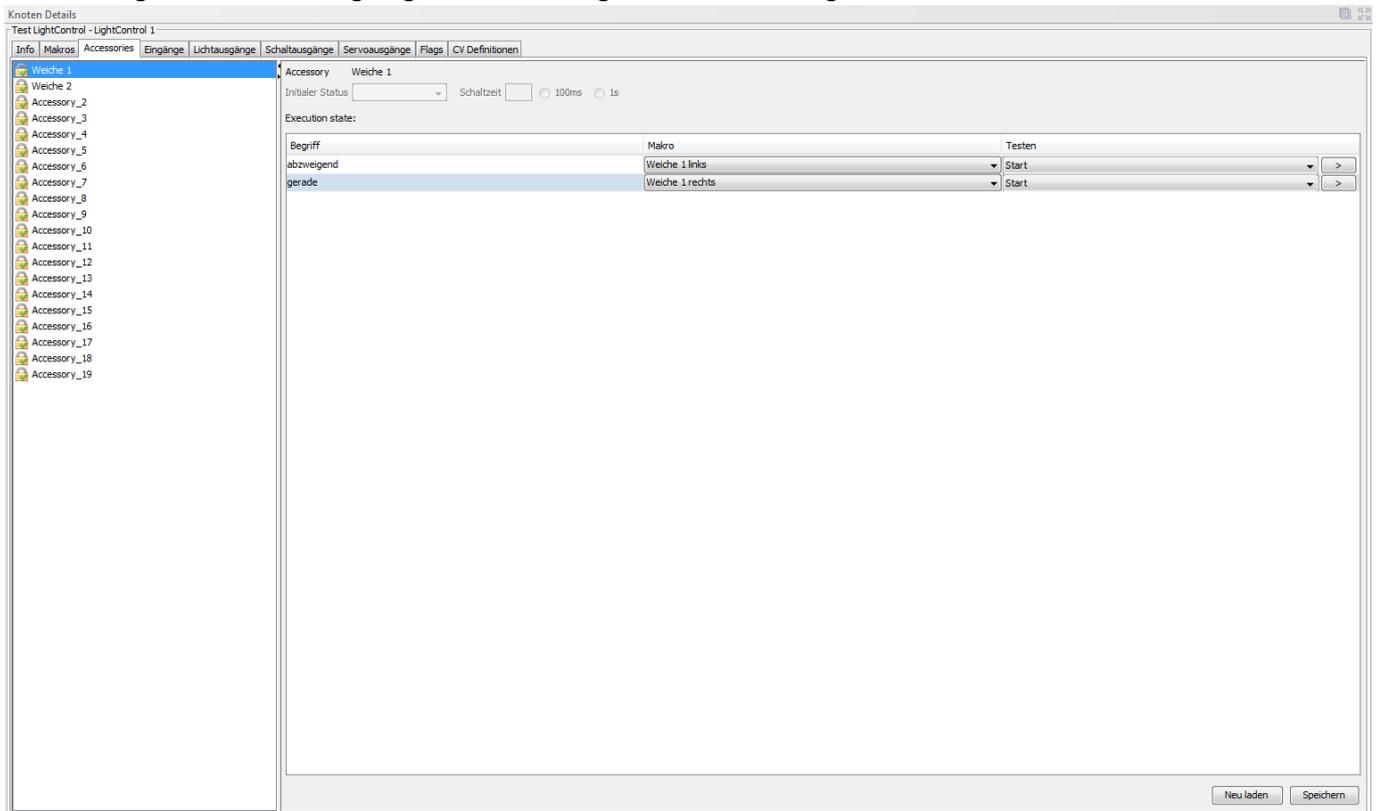


(Das Bild zeigt beispielhaft das Makro für Weiche 1 links. Die restlichen drei Makros werden nach dem gleichen Prinzip erstellt.)

Verwendung von Servos: Wenn statt Schaltausgängen Servo zum stellen der Weichen verwendet

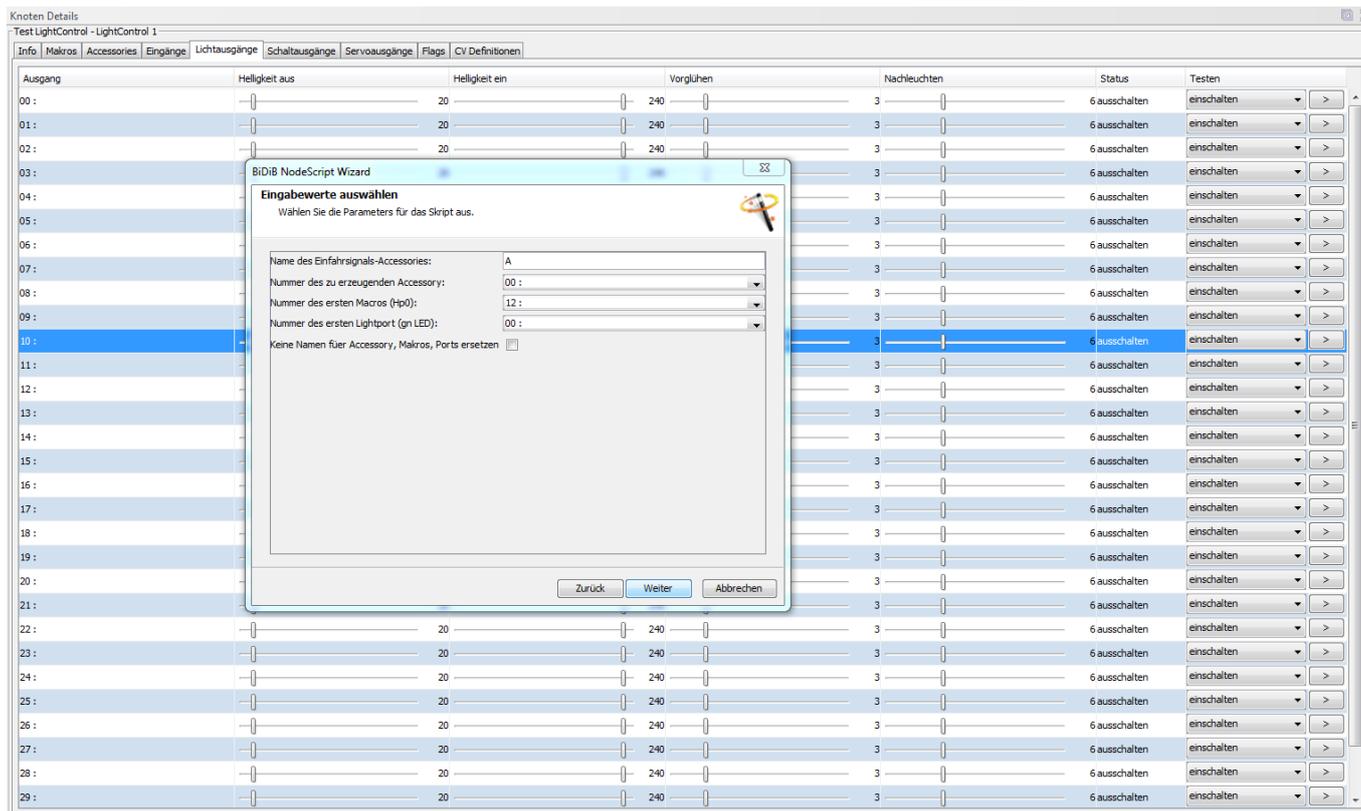
werden sollen, sind die Schritte 2 bis 4 zum Schalten der Ausgänge im obigen Makro durch Anfahren der Servo-Position und Abwarten der Servo-Bewegung zu ersetzen. Der Rest der Makros bleibt gleich.

Zum Schluss werden noch die Accessories ab Nummer 0 angelegt. Hier erfolgt dann auch die logische Zuordnung der Weichenlage (gerade/abzweigend) zur Stellung (links/rechts)



Signale

Danach werden mit Hilfe des NodeScripting Wizzard (Zauberstab-Symbol in der Symbolleiste) die Signale konfiguriert.

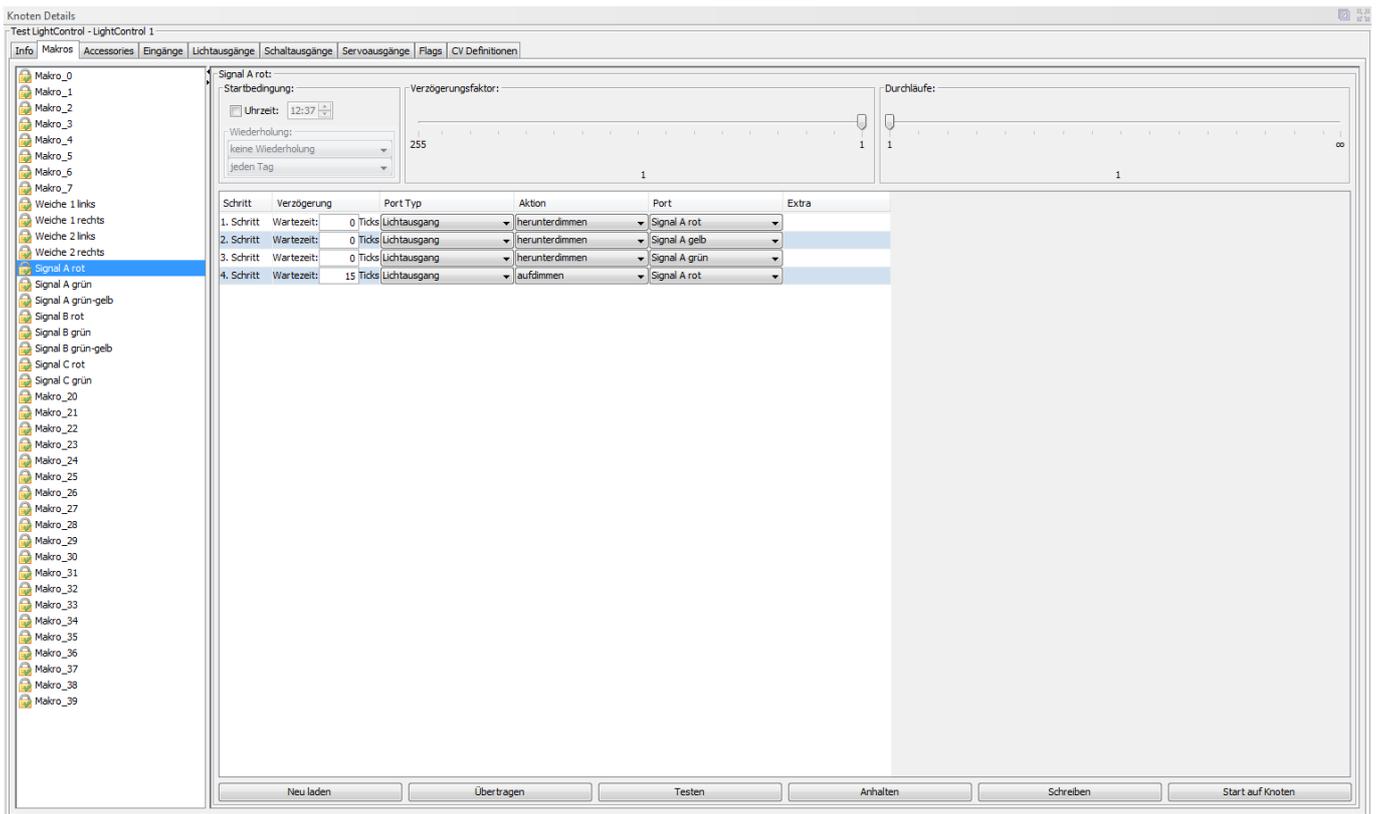
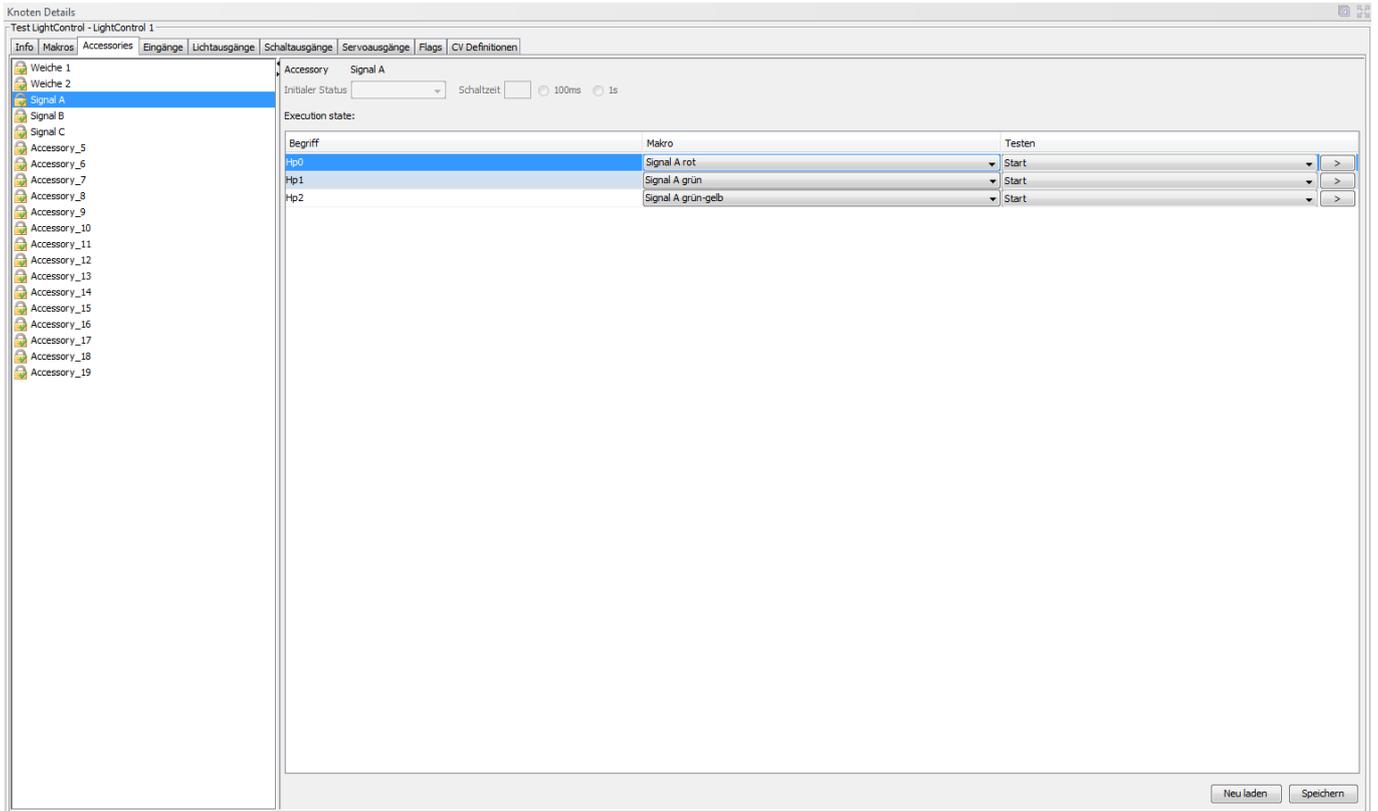


(Das Bild zeigt beispielhaft die Einstellungen für Signal A. Die restlichen Signale werden mit entsprechend angepassten Werten erstellt.)

Dabei werden für die Signale A und B Einfahrsignale verwendet und für Signal C ein Blocksignal. Die Accessories werden ab Nummer 2 verwendet, die Makros ab Nummer 12 und die LED-Ausgänge ab Nummer 0.

(Ich habe anschließend die Ausgänge für rt und gn bei den Einfahrsignalen vertauscht und die Makros entsprechend angepasst, da ich lieber beide Signaltypen nach dem gleichen Schema „rt vor gn“ habe. Dies ist für die richtige Funktion des Abzweigs aber nicht notwendig)

Um die Lesbarkeit der Makros usw. zu erhöhen, werden anschließend die Accessories, Makros und Lichtausgänge wie folgt umbenannt



Ausgang	Helligkeit aus	Helligkeit ein	Vorglühen	Nachleuchten	Status	Testen
00: Signal A rot	0	200	10	10	ausschalten	anschalten
01: Signal A grün	0	200	10	10	ausschalten	anschalten
02: Signal A gelb	0	200	10	10	ausschalten	anschalten
03: Signal B rot	0	200	10	10	ausschalten	anschalten
04: Signal B grün	0	200	10	10	ausschalten	anschalten
05: Signal B gelb	0	200	10	10	ausschalten	anschalten
06: Signal C rot	0	200	10	10	ausschalten	anschalten
07: Signal C grün	0	200	10	10	ausschalten	anschalten
08:	20	240	3	6	ausschalten	anschalten
09:	20	240	3	6	ausschalten	anschalten
10:	20	240	3	6	ausschalten	anschalten
11:	20	240	3	6	ausschalten	anschalten
12:	20	240	3	6	ausschalten	anschalten
13:	20	240	3	6	ausschalten	anschalten
14:	20	240	3	6	ausschalten	anschalten
15:	20	240	3	6	ausschalten	anschalten
16:	20	240	3	6	ausschalten	anschalten
17:	20	240	3	6	ausschalten	anschalten
18:	20	240	3	6	ausschalten	anschalten
19:	20	240	3	6	ausschalten	anschalten
20:	20	240	3	6	ausschalten	anschalten
21:	20	240	3	6	ausschalten	anschalten
22:	20	240	3	6	ausschalten	anschalten
23:	20	240	3	6	ausschalten	anschalten
24:	20	240	3	6	ausschalten	anschalten
25:	20	240	3	6	ausschalten	anschalten
26:	20	240	3	6	ausschalten	anschalten
27:	20	240	3	6	ausschalten	anschalten
28:	20	240	3	6	ausschalten	anschalten
29:	20	240	3	6	ausschalten	anschalten

Damit ist die Grundkonfiguration der LightControl beendet. An dieser Stelle sollten nun alle Ausgänge verkabelt werden und alle Ausgänge, Makros und Accessories mit Hilfe des Wizard getestet werden.

Die hier vorgestellte Grundkonfiguration genügt, um den Abzweig über ein PC-Programm (z.B. Rocrail) zu steuern. Wie der Abzweig ohne PC gesteuert werden kann, ohne das die Signale und Weichen alle von Hand geschaltet werden müssen, wird im restlichen Teil des Artikels erklärt

Zustandsspeicher (Flags)

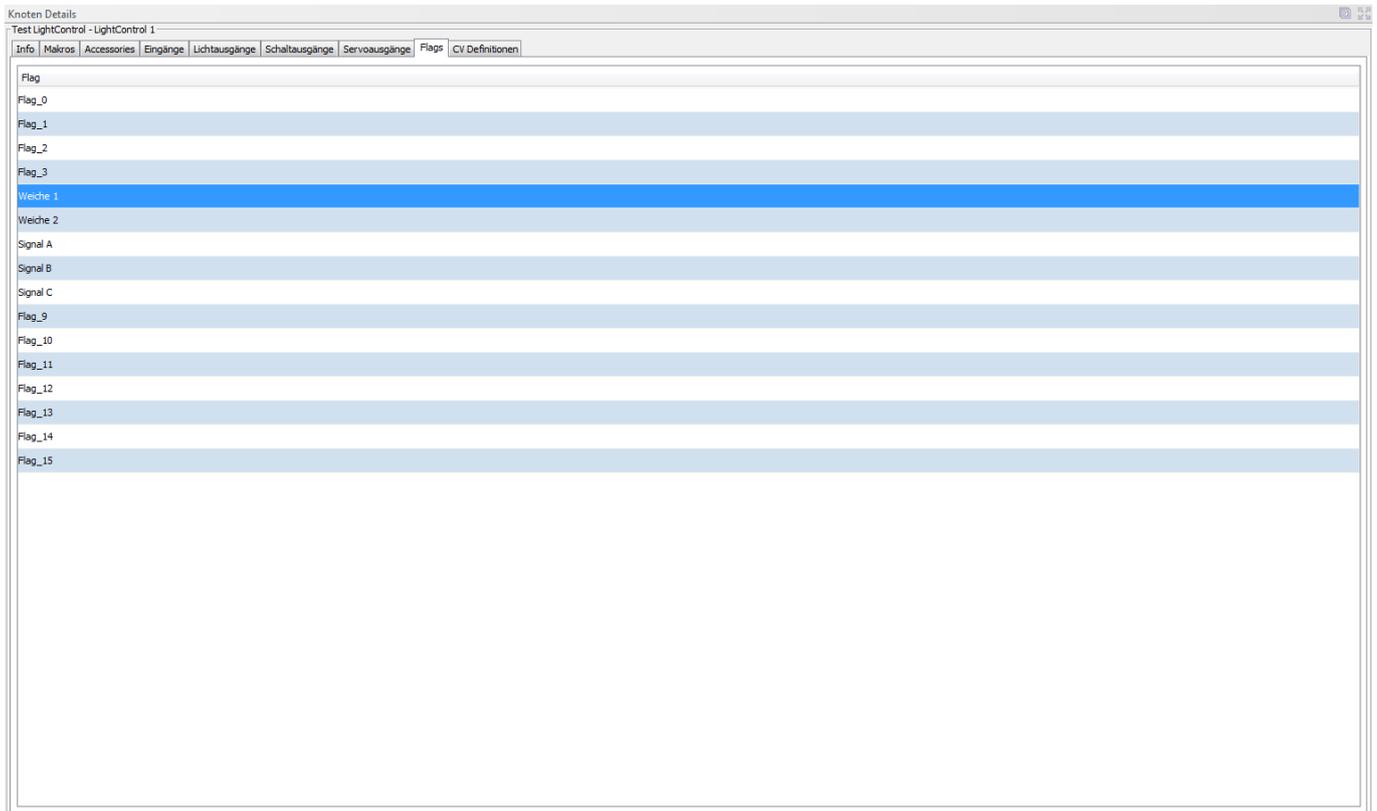
Wenn man den Abzweig mithilfe von Abläufen steuern möchte, dann ist es hilfreich den aktuellen Zustand der Signale, Weichen und Fahrstraßen zu wissen.

Da man allerdings in den Makros die Zustände der Ausgänge nicht abfragen kann, müssen die Zustände zusätzlich in den sogenannten Flags gespeichert werden.

Dabei handelt es sich um interne Zwischenspeicher der Baugruppe (hier: der LightControl). Diese Zwischenspeicher können wie ein Ausgang gesetzt und gelöscht werden, allerdings auch zusätzlich noch auf 0 oder 1 abgefragt werden. Sie bleiben auch nach dem Ende eines Makros erhalten und können somit z. B. verwendet werden um Zustände zu speichern.

Achtung: Das im Folgenden gezeigte Umbenennen der Flags ist erst ab Wizard Version > 1.9.2 möglich (genauer gesagt ab Wizard 1.9 Snapshot 3032) möglich

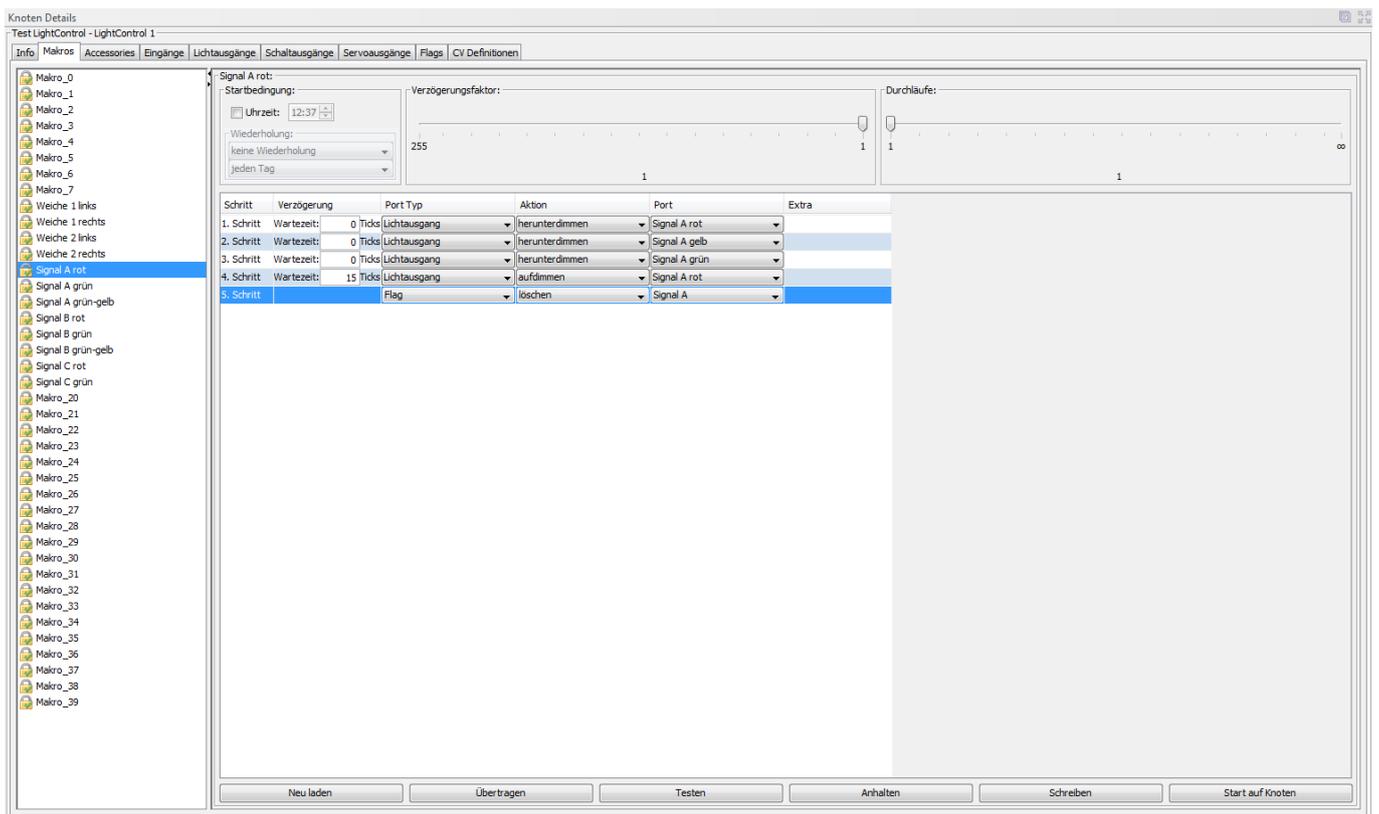
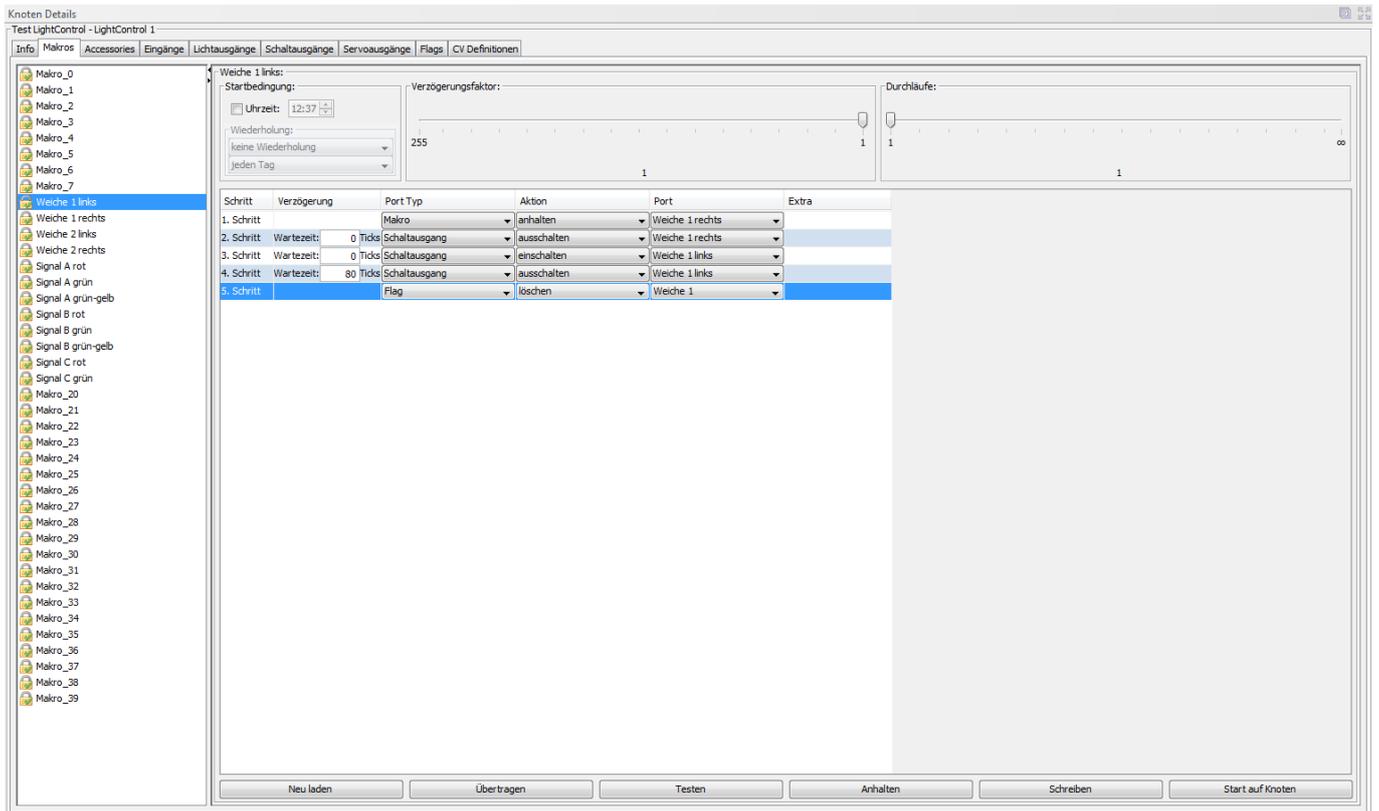
Für die Speicherung der Weichen- und Signalzustände werden die Flags ab Nummer 4 wie folgt benannt:



Anschließend werden die Flags in den jeweiligen Makros für Signale und Weichen am Ende über einen zusätzlichen Schritt gesetzt bzw. gelöscht.

Dabei gilt die Zuordnung:

- Weiche 1 ist links \Rightarrow Flag Weiche 1 löschen
- Weiche 1 ist rechts \Rightarrow Flag Weiche 1 setzen
- Signal A ist rot \Rightarrow Flag „Signal A“ löschen
- Signal A ist grün oder grün-gelb \Rightarrow Flag „Signal A“ setzen
- usw.



Virtuelle Fahrstraßen

Mit Hilfe von Makros können virtuelle Fahrstraßen realisiert werden. Bei diesen Fahrstraßen wird über den Ablauf festgelegt, welche Weichen und Signale bei einer Fahrstraße zu schalten sind.

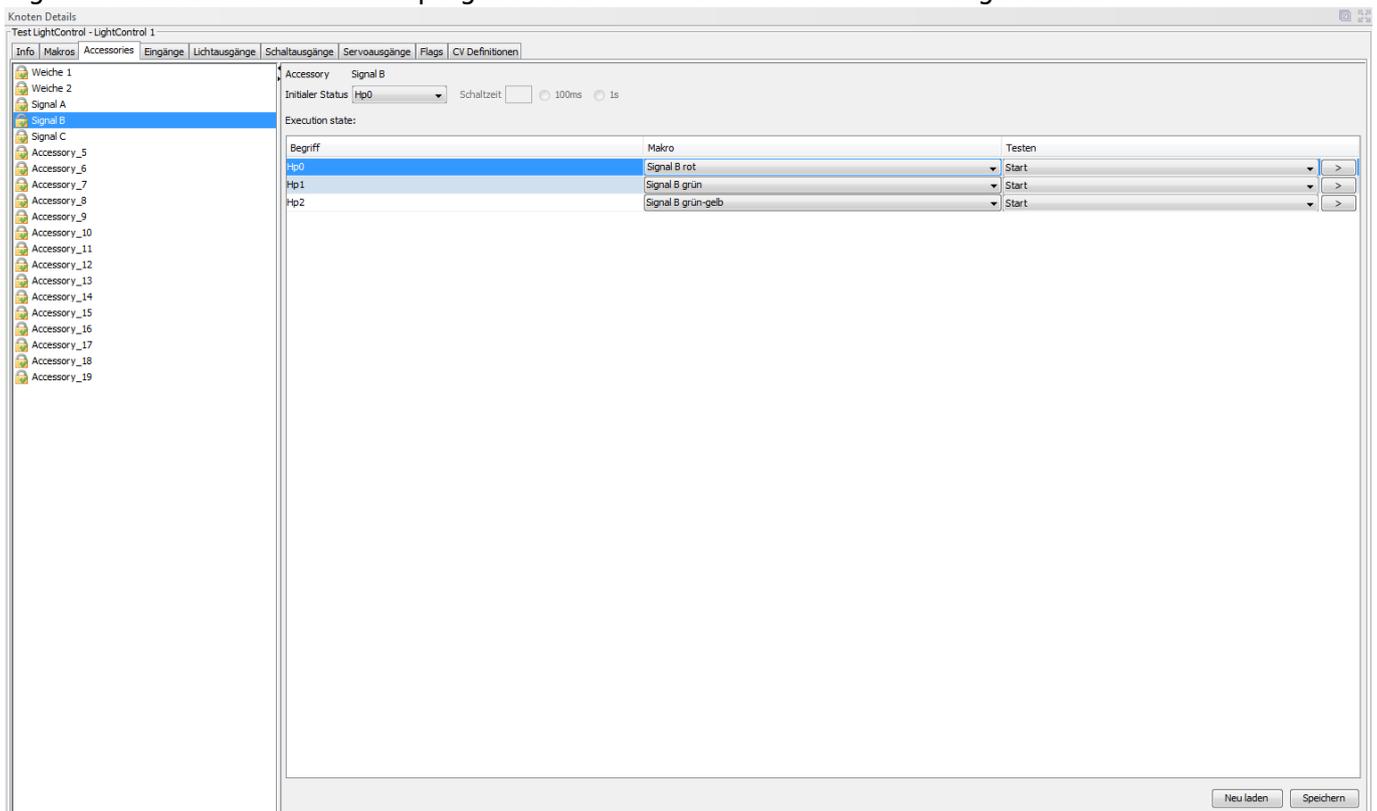
Eine Verriegelung der Fahrstraßen oder eine Auflösung durch den Zug erfolgen nicht.

Es geht also um manuell ausgelöste Fahrstraßen, welche die einfache Bedienung des Abzweigs von Hand ermöglichen, ohne jedes Signal und jede Weiche einzeln steuern zu müssen.

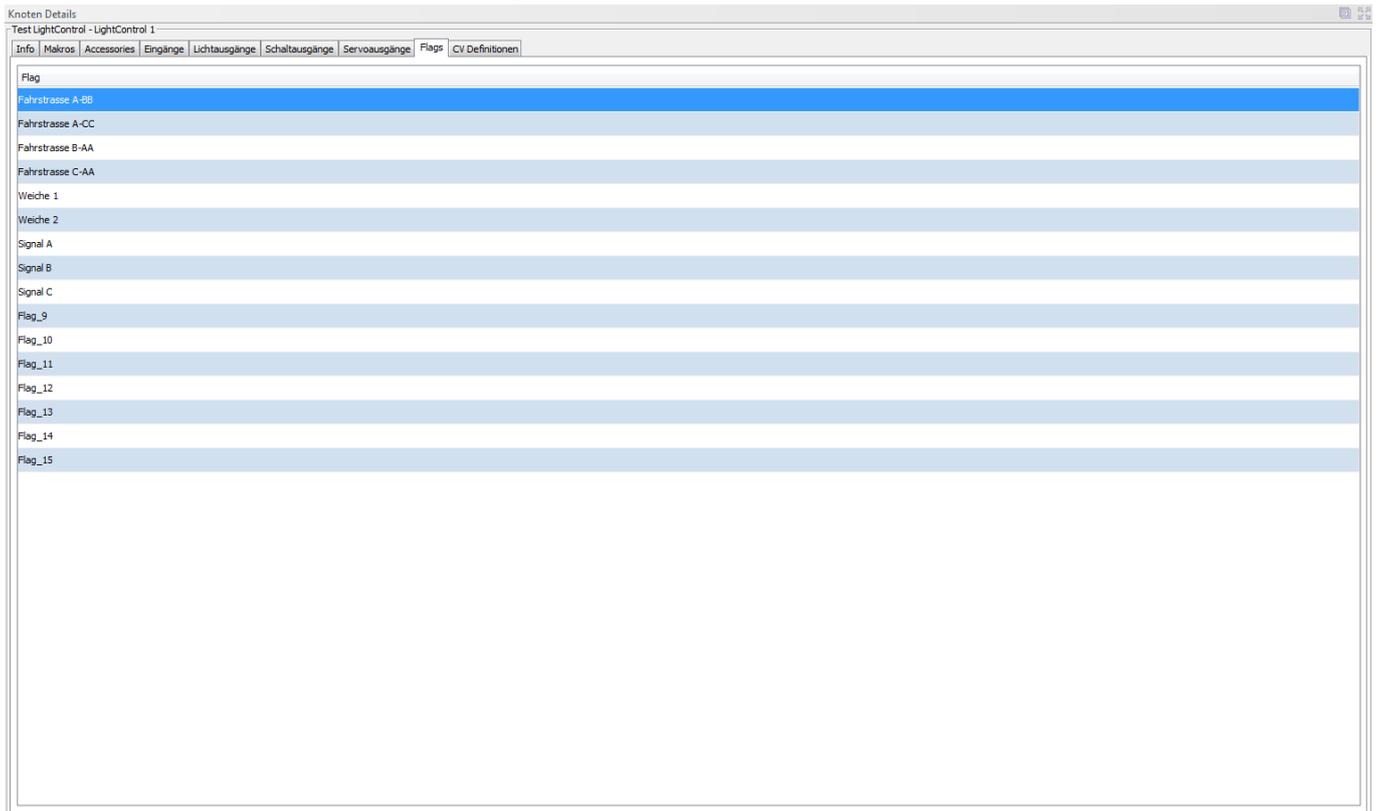
Bei Betrachtung des Abzweigs ergeben sich insgesamt vier Fahrstraßen:

- Fahrstraße A-BB: Von Signal A abzweigend zum Gegengleis von Signal B
- Fahrstraße A-CC: Von Signal A geradeaus zum Gegengleis von Signal C
- Fahrstraße B-AA: Von Signal B abzweigend zum Gegengleis von Signal A
- Fahrstraße C-AA: Von Signal C geradeaus zum Gegengleis von Signal A

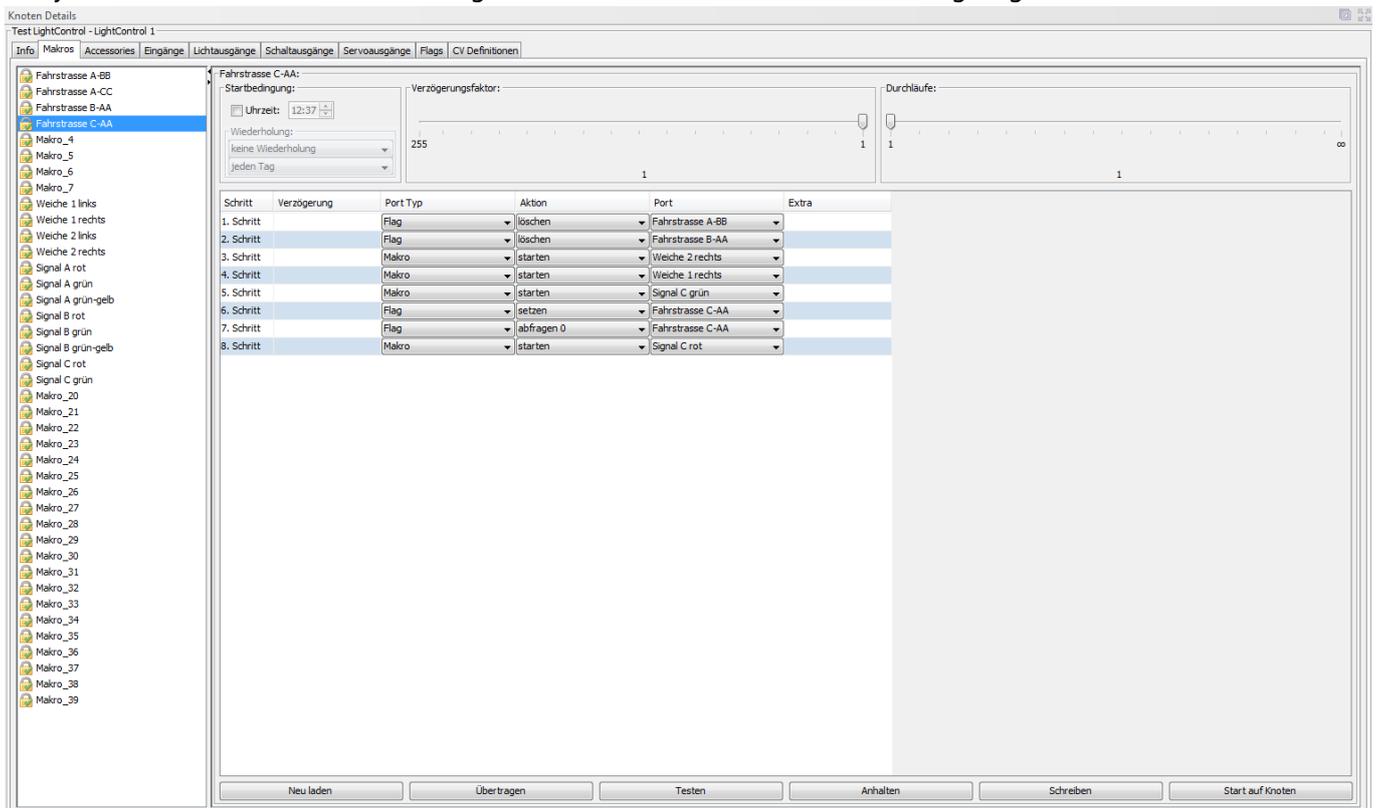
Damit der Abzweig nach dem Einschalten einen sicheren Zustand hat, werden die Accessories aller Signale beim Einschalten auf Hp0 gestellt. Das funktioniert auch ohne angeschlossenen PC!



Um die Zustände der Fahrstraßen abfragen zu können werden die Flags 0-3 verwendet. Der Zustand 1 bedeutet bei diesen Flags, dass die jeweilige Fahrstraße aktiv ist.



Für jede Fahrstraße wird nun ein eigenes Makro ab Makronummer 0 angelegt.



(Das Bild zeigt das Makro für die Fahrstraße C-AA. Die restlichen Makros werden nach dem gleichen Muster programmiert)

Das Makro hat dabei folgenden Ablauf:

- Löschen der Flags von allen Fahrstraßen, die aufgelöst werden müssen
- Stellen der Weichen

- Stellen des Signals auf Grün
- Setzen des eigenen Flags
- *Warten* bis das eigene Flag von einer anderen Fahrstraße gelöscht wird
- Stellen des Signals auf Rot

Die restlichen Fahrstraßen werden nach diesem Prinzip als Makro 1 bis 3 erstellt.

Sequentielle Abläufe

Die bisher realisierten Makros haben leider Schönheitsfehler: Die Stellung der Weichen und Signale erfolgt quasi gleichzeitig, was nicht sehr vorbildgerecht ist. Besser wäre es, wenn alle Elemente nacheinander schalten würden.

Um das zu erreichen müssen die Makros aufgerufen werden und auf deren Ende gewartet werden. Dazu kann man einfach die bereits angelegten Zustandsmerker (Flags) verwenden.

Die Makros der Fahrstraßen müssen hierzu folgendermaßen angepasst werden:

Schritt	Verzögerung	Port Typ	Aktion	Port	Extra
1. Schritt		Flag	löschen	Fahrstrasse A-BB	
2. Schritt		Flag	löschen	Fahrstrasse B-AA	
3. Schritt		Flag	abfragen 0	Signal A	
4. Schritt		Flag	abfragen 0	Signal B	
5. Schritt		Makro	starten	Weiche 2 rechts	
6. Schritt		Flag	abfragen 1	Weiche 2	
7. Schritt		Makro	starten	Weiche 1 rechts	
8. Schritt		Flag	abfragen 1	Weiche 1	
9. Schritt		Makro	starten	Signal C grün	
10. Schritt		Flag	abfragen 1	Signal C	
11. Schritt		Flag	setzen	Fahrstrasse C-AA	
12. Schritt		Flag	abfragen 0	Fahrstrasse C-AA	
13. Schritt		Makro	starten	Signal C rot	
14. Schritt		Flag	abfragen 0	Signal C	

Beim Auflösen der Fahrstraßen können mehrere Fahrstraßen gleichzeitig aufgelöst werden. Deshalb erfolgt die Abfrage erst nach dem Auflösen aller Fahrstraßen. Abgefragt wird hier einfach der Zustand der von den aufgelösten Fahrstraßen zurückgestellten Signale. Danach werden die Weichen und das jeweilige Signal sequentiell (nacheinander) geschaltet und jeweils gewartet bis das aufgerufene Makro beendet ist.

Ein kleiner Nachteil dieser Lösung ist, dass aufgrund der nur 16 Flags auch nur eine begrenzte Anzahl an Makros auf diese Weise aufgerufen werden kann. Für den Abzweig reicht es jedoch und es bleiben sogar noch Reserven.

Aktionen über Schalter auslösen

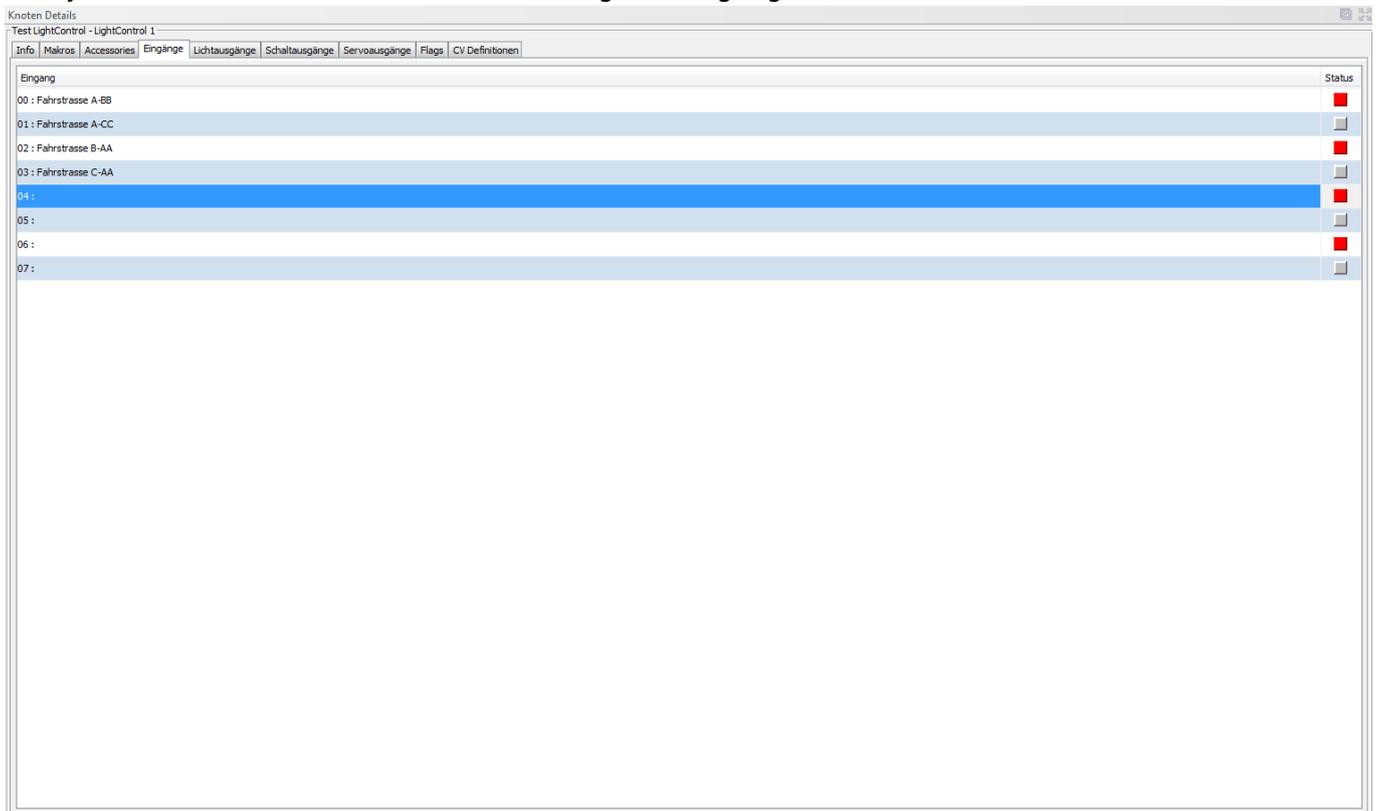
Nachdem die Fahrstraßen nun angelegt sind, sollen diese natürlich auch vom Bediener ausgelöst werden können. Damit hierzu kein PC benötigt wird, werden die Eingänge der LightControl verwendet.

Hierzu muss im Wizard lediglich im Menü unter **Knoten** die Option **Makros über externen Eingang starten** aktiviert werden.

Jetzt wird auch deutlich, warum die Makros für die Weichen erst bei Makronummer 8 beginnen sollen. Durch Aktivierung der Option werden durch die Eingänge 0-7 an der LightControl automatisch die Makros 0-7 aufgerufen:

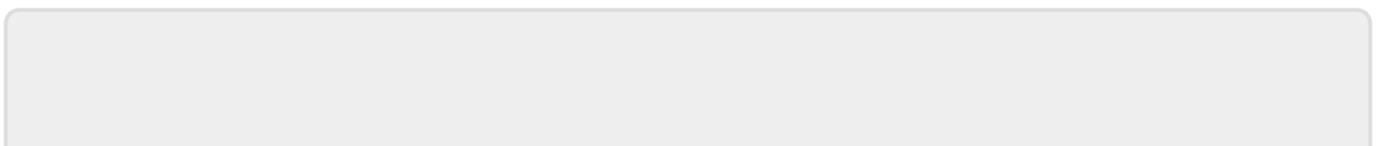
- Eingang 0 aktiviert Makro 0
- Eingang 1 aktiviert Makro 1
- usw.

Was jetzt nur noch fehlt ist die Umbenennung der Eingänge



Dateien

Node-Datei für den Import in eine LightControl



From:

<https://forum.opendcc.de/wiki/> - **BiDiB Wiki**

Permanent link:

<https://forum.opendcc.de/wiki/doku.php?id=beispielkonfig:lightcontrol&rev=1553322495>

Last update: **2019/03/23 07:28**

