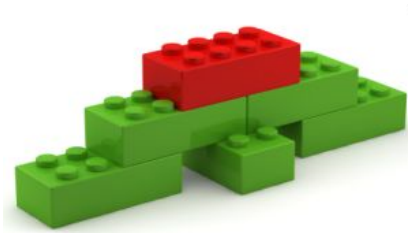


Was sind Makros und Accessories?

Das Prinzip könnte man in die Legowelt transportieren und sich damit vorstellen:



Die **Port-Aktionen** (Eingänge, Schaltausgänge, GPIOs, LED-Ausgänge oder auch Servo-Ports) sind die einfachen Legobausteine welche auf einen einzelnen Port wirken:

- Die Port-Aktion eines Servo-Port lässt den Servo an eine bestimmte Position fahren.
- Die Port-Aktion eines Schaltausgang lässt den Ausgang ein- oder ausschalten.

Die **Makros** sind quasi die Grundplatte, auf die man die Legobausteine draufbauen kann um eine oder mehrere Port-Aktionen auszuführen:

Zuerst einen Ausgang schalten, dann warten, dann Servo bewegen, dann anderen Ausgang einschalten, ...

Ein Makro ist also eine Zusammenfassung des sequentiellen Ablauf von Aktionen.

Normalerweise steckt man hier alles rein, was auch zusammengehört und das ist dann in der Legodenke schon ein Art „Funktionsbaustein“. Im realen Betrieb ist es oft sinnvoll den Ablauf in einzelne Makros aufzuteilen und diese von einem übergeordneten Makro auszuführen. Damit verschafft man sich einen Überblick und eine ordentliche Struktur.

Die Makros sind toll für Dinge „neben“ der Bahn, also Häuser beleuchten, Spielplatz bewegen usw. die *einfach laufen* sollen.

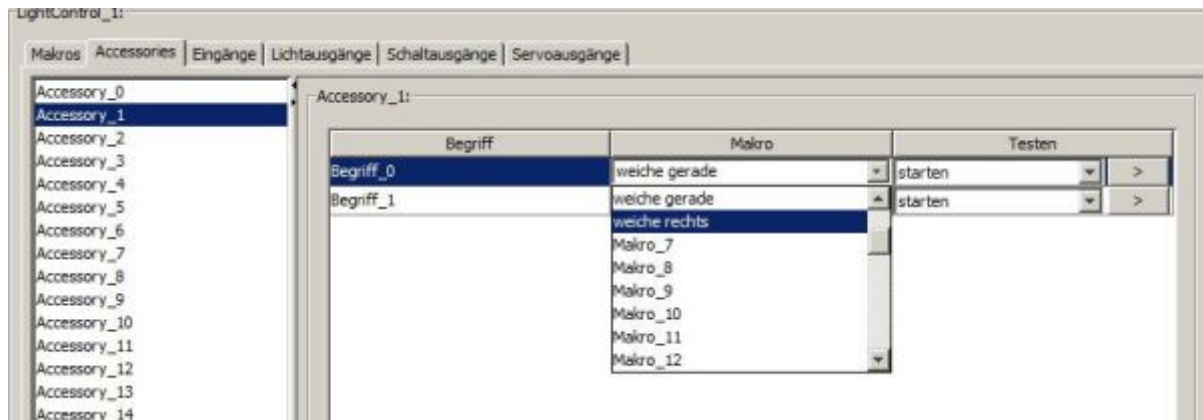
Die Dinge „auf der Bahn“ haben aber noch weitergehende Ansprüche:

Ein Signal zeigt einen Begriff oder eine Weiche steht in einer bestimmten Stellung.

Manchmal ist das direkt ein Schaltausgang (dann ist der ganze Klapperatismus eigentlich überflüssig), aber oft sind das kombinierte Abläufe „Signal umstellen“:

- Rotes Licht abblenden,
- bisschen warten,
- grünes und gelbes Licht aufblenden.

Genau für diesen Ablauf verwendet man die Makros, da kann man diese Sachen schön unterbringen.



Bei Bahnbetriebsabläufen ist die korrekte und überwachte Ausführung von Makros gefordert. Dazu wird eine weitere Ebene der **Accessory** eingeführt.

Dazu werden Makros einem **Accessory** zugeordnet, welches einen oder mehrere Zustände (Aspekte) haben kann. Bei einem Accessory besteht zudem die Möglichkeit, dass der letzte Zustand gespeichert und bei einem Einschalten der Anlage wieder ausgeführt wird.

Des weiteren erfolgt beim Ausführen des Accessory eine Rückmeldung an das PC-System, z.B. „Servo hat Position erreicht“ oder „Fehler aufgetreten“!

Mein erstes Makro

Um einen Weichenantrieb schalten zu können, muss ein BiDiB-Knoten mit Makro und Accessories konfiguriert werden.

- Makro sind Abfolgen von Befehlen (Schaltbefehlen, Wartezeiten, Eingang abfragen ...) die auf der OneControl gespeichert werden können.
- Accessories sind eine übergeordnete Struktur die sogenannte Aspekte haben, welche pro Aspekt ein Makros schalten. Das Accessory wird später vom Steuerungsprogramm (Rocrail, iTrain, WDP, o.ä.) angesteuert.

Der nachfolgende Abschnitt zeigt wie man eine Weiche mit Spulenantrieb mit Makros und Accessory konfigurieren kann, um die Weiche zu schalten.

Weiche mit Spulenantrieb schalten

Eine Weiche hat im einfachsten Fall 2 Schaltzustände:

- Geradeaus
- Abzweigend

Um diese Weiche zu schalten braucht man 2 Makro und 1 Accessory mit 2 Aspekten (*geradeaus* und *abzweigend*). Dazu erstellt man ein Makro Weiche *geradeaus* mit folgenden Schritten:

- Makro „Weiche abzweigend“ stoppen
- Ausgang X einschalten
- Wartezeit 20 Ticks

Die Liste mit den Schritten ist am Anfang leer. Über das Kontextmenü (rechte Maustaste) kann man mit *Danach einfügen* einen leeren Schritt einfügen und konfigurieren.

Anschliessend muss das Makro mit Klick auf den Schreiben-Button auf die OneControl übertragen werden. Das Icon in der Makroliste wechselt dadurch zu einem Schloss mit grünem Haken.

Schritt	Verzögerung	Port Typ	Aktion	Port
1. Schritt		Makro	anhalten	Weiche_X_abzweig
2. Schritt	Wartezeit: 0 Ticks	Schaltausgang	ausschalten	Ausgang_9
3. Schritt	Wartezeit: 5 Ticks	Schaltausgang	einschalten	Ausgang_8
4. Schritt	Wartezeit: 20 Ticks	Verzögerung		
5. Schritt	Wartezeit: 0 Ticks	Schaltausgang	ausschalten	Ausgang_8



Die Wartezeit von 20 Ticks am Ende ist bei der OneControl wichtig weil die Endabschaltung standardmässig auf 15 Ticks eingestellt ist! D.h. die OneControl überwacht nach 15 Ticks ob die Spule gestellt ist und schaltet dann den Antrieb ab. Ein zu schnelles Hin- und Herschalten würde sonst zu sporadischen Fehlermeldungen führen.

Danach wird ein Makro *Weiche abzweigend* erstellt mit folgenden Schritten:

- Makro „Weiche geradeaus“ stoppen
- Ausgang Y einschalten
- Wartezeit 20 Ticks

Anschliessend muss auch dieses Makro mit Klick auf den *Schreiben*-Button auf die OneControl übertragen werden.

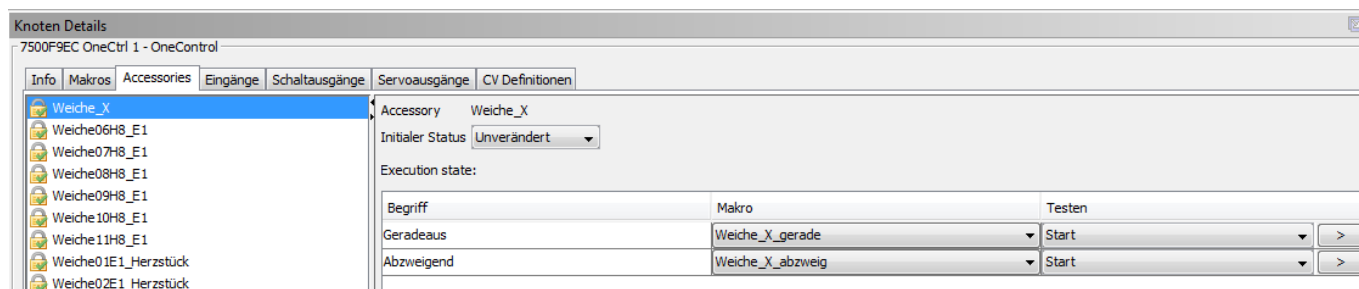
Schritt	Verzögerung	Port Typ	Aktion	Port
1. Schritt		Makro	anhalten	Weiche_X_gerade
2. Schritt	Wartezeit: 0 Ticks	Schaltausgang	ausschalten	Ausgang_8
3. Schritt	Wartezeit: 5 Ticks	Schaltausgang	einschalten	Ausgang_9
4. Schritt	Wartezeit: 20 Ticks	Verzögerung		
5. Schritt	Wartezeit: 0 Ticks	Schaltausgang	ausschalten	Ausgang_9

Jetzt hat man 2 Makro um die Weiche hin- und herstellen zu können, wenn das jeweilige Makro mit Klick auf den *Testen*-Button gestartet wird.

Anschliessend muss das Accessory Weiche X erstellt werden mit den 2 Aspekten:

- Geradeaus -> Schaltet Makro „Weiche geradeaus“
- Abzweigend -> Schaltet Makro „Weiche abzweigend“

Nachdem die beiden Aspekte angelegt sind, muss das Accessory mit Klick auf den *Speichern*-Button auf die OneControl übertragen werden. Danach wird das Icon in der Liste der Accessory auch wieder zum Schloss mit dem grünen Haken.



Jetzt kann man im Wizard mit den Pfeil-Buttons bei den Aspekten die Weiche hin- und herschalten.

Anschliessend im Wizard die Verbindung zum Master trennen, weil die Schnittstelle jetzt vom Steuerungsprogramm (Rocrail, iTrain, WDP o.ä.) benötigt wird.

Wo gibt es fertige Makros?

Oben haben wir gesehen, wie das im Prinzip funktioniert. Mag ja ganz toll sein, aber muß man das jetzt immer wieder machen? Nein, für viele, viele Standardfälle gibt es einen [Baukasten mit vorgefertigten Makros!](#)

Und warum ist das so gebaut?

Ein Steuerungsprogramm will ein Signal (wir nennen das Accessory) auf ein bestimmtes Signalbild (=Aspect) stellen. Dabei ist es dem Steuerungsprogramm egal, wie dieses Signalbild dargestellt wird: das kann ein Flügelsignal sein, ein Ks-Signal oder irgendetwas anderes. Und genau auf dieser Ebene spricht das Steuerprogramm auch die Baugruppe an: Accessory 'abc' soll Aspect '123' zeigen. Wie das die Baugruppe dann konkret macht, ist alleinige Sache der Baugruppe.

Es gibt BiDiB-Baugruppen, die manchen z.B. nur eine Weiche. Aspect 1 ist eine Spule, Aspect 2 ist die andere Spule. Fertig. Nichts umzustellen, keine Makros o.ä.

Oft reicht die einfache Ansteuerung, aber gerade bei Signalen ist es umfangreicher und recht stark von Bahnverwaltung / Epoche und Ausbauzustand des Vorbildes abhängig. Manchmal blinkt es auch. Wie bringt man das hin?

Bei OpenDCC-Baugruppen (und davon abgeleiteten, lizenzierten Baugruppen) gibt es eine flexible Konfiguration: Die Idee: bei jedem Begriff eines Accessory statt eines festen Musters eine Art Schrittkettensteuerung auszuführen. Diese Schrittketten nennen wir Makro.

Von oben rein ist der Weg also:

PC stellt Aspect beim Accessory ein.
Aspect startet eine Schrittkette (Makro)
Im Makro sind dann die An- Ausknipser fürs Licht.

Damit ist man voll flexibel und kann auch besondere Signalbilder darstellen - bis hin zum Lauflicht eines Baustellenblitzers. Und für Standardfälle (z.B. Blocksignal) gibt es den [Zauberstab](#)!

From:

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

Permanent link:

https://forum.opendcc.de/wiki/doku.php?id=grundlage_makro&rev=1596222608

Last update: **2020/07/31 21:10**

