



Im Grunde funktioniert eine Blocksteuerung nach einem sehr einfachen Prinzip:

- der Streckenverlauf wird in Abschnitte (=Blöcke) aufgeteilt.
- jeder Zug in einem Block kann diesen nur verlassen, wenn der nachfolgende Abschnitt (=Block) frei ist.

So ist dies auch bei dem "Test-Stellwerk" mit dem Traincontroller (siehe Beschreibung auf meinen Internetseiten).

Grundlegendes ist auf meinen Seiten [http://www.hjb-electronics.de/Steuerungen TPPs](http://www.hjb-electronics.de/Steuerungen_TPPs), dort nachzulesen:

BLOCK-Steuerung, konventionell, analog (die gute alte Relaisstechnik)

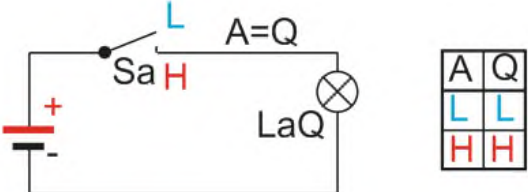
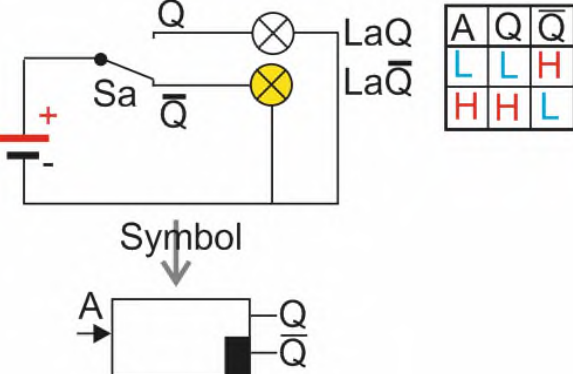
und

BLOCK-Steuerung, digital, Simulation. Zur Simulation steht ein [Download "Blocksteuerung Abzeig.zip"](#) zur Verfügung, sodass jeder den logischen Ablauf bei einer Blocksteuerung selber nachvollziehen kann. Zur Demonstration ist auch hier wieder unser vertrauter Gleisverlauf mit einer Ausweichstelle dargestellt. Wie aber funktioniert diese Simulation, wie ist sie "programmiert"? Dies soll nachfolgend näher betrachtet werden.

Vorweg sind dazu jedoch einige Grundkenntnisse zur Digitaltechnik erforderlich, die zuvor erläutert werden sollen:

Einige Grundlagen zur Digitaltechnik

Ob man nun mit Relais (konventionell, wie früher), digital diskret mit ICs (Hardware-Schaltungen), oder per digitaler Software arbeitet, stets ist das Prinzip gleich. Wenn man sich der Mühe unterzieht einmal etwas in die Tiefe der Materie einzusteigen, so wird man die Logik (welche dahinter steckt) besser begreifen und sich dann auch leichter bei der Programmierung von Steuerungssoftwares tun! Kommen wir also zu den "kleinsten" Grundbausteinen der Digitaltechnik:

 Abb.1	Ein einfacher Ein-/Aus-Schalter: L steht für "Low" (englisch "niedrig", Spannung), andere übliche Bezeichnungen: aus, 0, f (für "falsch"). H steht für "High" (englisch "hoch", Spannung), andere übliche Bezeichnungen: an, 1, w (für "wahr"). Weitere Informationen siehe auch unter ARDUINO => Grundlagen => Hilfreich zu wissen auf meiner Homepage
 Abb.2	Ein einfacher Wechselschalter: im Grunde wie obiger Schalter, nur dass er nun zwei Ausgänge hat, im eingeschalteten Zustand ist der Ausgang Q an, im ausgeschalteten Zustand ist der Ausgang Q quer an (die Bezeichnung "Q quer" wird in der Schaltungstechnik und bei ICs, beispielsweise bei FlipFlops, gerne für invertierenden Ausgänge verwendet, der invertierende Ausgang wird dabei durch ein schwarzes Kästchen gekennzeichnet; siehe Symbol in der linken Abbildung). Mit diesen Schaltern lassen sich alle möglichen Kombinationen für logische Schaltungen realisieren. Dazu hat man zur besseren und einfacheren Übersicht Symbole eingeführt, welche nachfolgend dargestellt werden.



Schalter oder Relais	Gatter-Bezeichnung	veraltete Gatter-Bezeichnung	In dieser Abbildung werden übliche "Gatter" dargestellt. Ganz links veranschaulichen Schalter die jeweilige Funktion (hier kann es sich auch um Relais handeln).															
	<table border="1"><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>L</td><td>L</td><td>L</td></tr><tr><td>L</td><td>H</td><td>L</td></tr><tr><td>H</td><td>L</td><td>L</td></tr><tr><td>H</td><td>H</td><td>H</td></tr></table> $A \wedge B = Q$ AND	A	B	Q	L	L	L	L	H	L	H	L	L	H	H	H	 UND	Dann folgt eine kleine Tabelle, in der die möglichen logischen Eingangs-Zustände (A, B) aufgeführt sind und der Zustand des resultierenden Ausgangs (Q). Das umgedreht "v" bedeutet "und" (soll an ein "A" erinnern, aus dem Englischen für "and" = "und"), das "v" steht für "oder" (aus dem Lateinischen "velum" = "oder"). Ein "Querbalken" über einen Buchstaben oder über einen ganzen Ausdruck bedeutet "nicht" (Englisch "not" = "nicht"). <u>Anmerkung:</u> man kann logische Funktionen auch rechnen (mit der booleschen Algebra). Nehmen wir beispielsweise den Ausdruck A UND B = Q und negieren diesen: NICHT (A UND B) = NICHT Q = NICHT A ODER NICHT B. Dazu ein reales Beispiel: A=Spannung ein; B=Schalter geschlossen; Q=Lampe an; Spannung ein UND Schalter geschlossen = Lampe an. NICHT(Spannung ein UND Schalter geschlossen) = Lampe NICHT an = Spannung NICHT ein ODER Schalter NICHT geschlossen = Lampe NICHT an. So kann man ganze logische Abläufe (Schaltungen) "ausrechnen"; aber das wäre ein separater Grundkurs, welcher hier den Rahmen sprengen würde. Rechts davon sind die heute genormten Symbole der Gatter mit ihren Bezeichnungen aufgeführt. Ein kleine runder Kreis steht hier für "nicht" (äquivalent zum Querbalken bei Formeln) Ganz rechts findet man entsprechend die alten Symbole. Ich finde die alten Symbole besser, da die unterschiedlichen Funktionen besser mit einem Blick erkennbar sind. Bei den neuen Symbolen muss man erst in ihr kleines Kästchen schauen, um zu sehen was gemeint ist.
A	B	Q																
L	L	L																
L	H	L																
H	L	L																
H	H	H																
	<table border="1"><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>L</td><td>L</td><td>H</td></tr><tr><td>L</td><td>H</td><td>H</td></tr><tr><td>H</td><td>L</td><td>H</td></tr><tr><td>H</td><td>H</td><td>L</td></tr></table> $A \wedge B = Q$ NAND	A	B	Q	L	L	H	L	H	H	H	L	H	H	H	L	 Nicht UND	
A	B	Q																
L	L	H																
L	H	H																
H	L	H																
H	H	L																
	<table border="1"><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>L</td><td>L</td><td>L</td></tr><tr><td>L</td><td>H</td><td>H</td></tr><tr><td>H</td><td>L</td><td>H</td></tr><tr><td>H</td><td>H</td><td>H</td></tr></table> $A \vee B = Q$ OR	A	B	Q	L	L	L	L	H	H	H	L	H	H	H	H	 ODER	
A	B	Q																
L	L	L																
L	H	H																
H	L	H																
H	H	H																
	<table border="1"><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>L</td><td>L</td><td>H</td></tr><tr><td>L</td><td>H</td><td>L</td></tr><tr><td>H</td><td>L</td><td>L</td></tr><tr><td>H</td><td>H</td><td>L</td></tr></table> $A \vee B = Q$ NOR	A	B	Q	L	L	H	L	H	L	H	L	L	H	H	L	 Nicht ODER	
A	B	Q																
L	L	H																
L	H	L																
H	L	L																
H	H	L																
	<table border="1"><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>L</td><td>L</td><td>L</td></tr><tr><td>L</td><td>H</td><td>L</td></tr><tr><td>H</td><td>L</td><td>L</td></tr><tr><td>H</td><td>H</td><td>H</td></tr></table> $(A \wedge \bar{B}) \vee (\bar{A} \wedge B) = Q$ EXOR	A	B	Q	L	L	L	L	H	L	H	L	L	H	H	H	 Exklusiv ODER	
A	B	Q																
L	L	L																
L	H	L																
H	L	L																
H	H	H																
	<table border="1"><tr><th>A</th><th>Q</th></tr><tr><td>L</td><td>H</td></tr><tr><td>H</td><td>L</td></tr></table> $\bar{A} = Q$ NOT	A	Q	L	H	H	L	 Nicht										
A	Q																	
L	H																	
H	L																	

Abb.3

Die hier dargestellten Gatter haben jeweils 2 Eingänge (bis auf NOT), sie können aber auch mehr Eingänge haben, wobei sich dann die jeweilige Tabelle erweitert.

Ganz rechts findet man entsprechend die alten Symbole.

Ich finde die alten Symbole besser, da die unterschiedlichen Funktionen besser mit einem Blick erkennbar sind. Bei den neuen Symbolen muss man erst in ihr kleines Kästchen schauen, um zu sehen was gemeint ist.

Abb.3

Die hier dargestellten Gatter haben jeweils 2 Eingänge (bis auf NOT), sie können aber auch mehr Eingänge haben, wobei sich dann die jeweilige Tabelle erweitert.



Steuerungen => Grundlagen Blocksteuerung, wie funktioniert das? Mit ergänzenden Informationen

Aus Kombinationen der Schaltelemente der obigen Abbildung lassen sich alle denkbaren logische Funktionen (und damit Abläufe) realisieren. Man könnte dies also alles auch nur mittels einfacher Schalter realisieren (per Hand schalten), oder diese durch Relais ersetzen, um Abläufe elektrisch zu schalten (MÄRKLIN Fernschalter; oder früher beim Vorbild Drucktastenstellwerke).

Wie eine solche Kombination aussehen kann, soll an einem Beispiel von 4 Blockstellen mit einem Ausweichgleis im Folgenden dargestellt werden.

Unter http://www.hjb-electronics.de/Steuerungen_TPPs sehen Sie unter **BLOCK-Steuerung, digital, Simulation** folgende Abbildung:

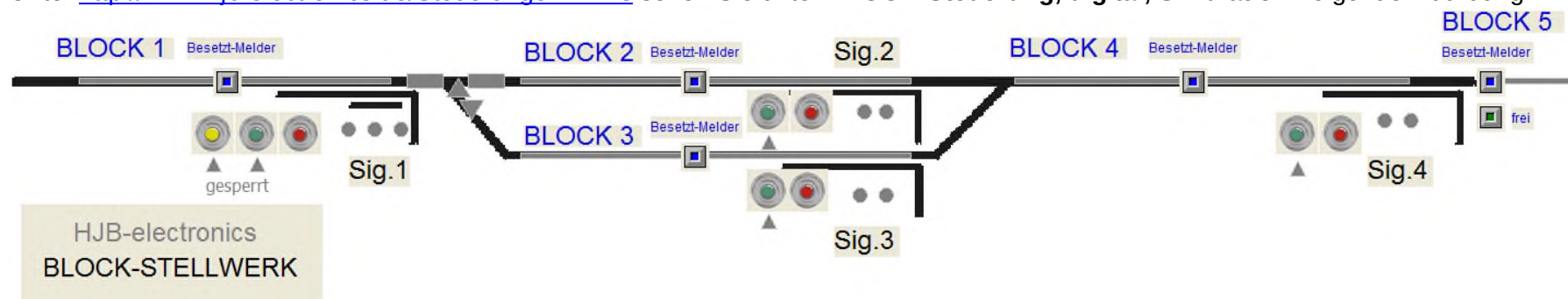


Abb.4

Diese Simulation einer Blocksteuerung mit einem Ausweichgleis wurde mit einem Programm "ProfiLab-Expert" von der Firma "ABACOM" (LINK: <http://www.abacom-online.de/>) erstellt. Damit erstellte Programme oder Simulationen können in ausführbare EXE-Dateien gewandelt und gespeichert werden, welche dann lizenzfrei und ausführbar weitergegeben werden können.

Wie hiermit eine Blocksteuerung programmiert wurde, soll nun kurz beschrieben werden:

Wir betrachten 4 Blöcke 1, 2, 3, und 4:

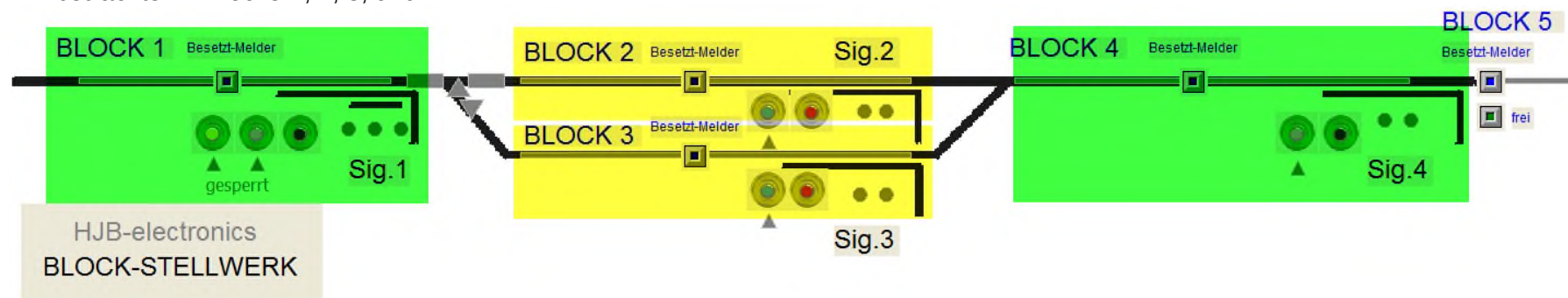


Abb.5

Es gibt hier einen kleinen Unterschied zu beachten, grün und gelb markierte Blöcke. Dazu später mehr.



Trotz gewisser kleiner Unterschiede überlegen wir uns eine Logik, welche allen Blöcken gemeinsam ist und erstellen so zu sagen ein kleines "Unterprogramm", hier "Makro" genannt. Dies erleichtert dann später das komplette Programm enorm.

Das "Makro":

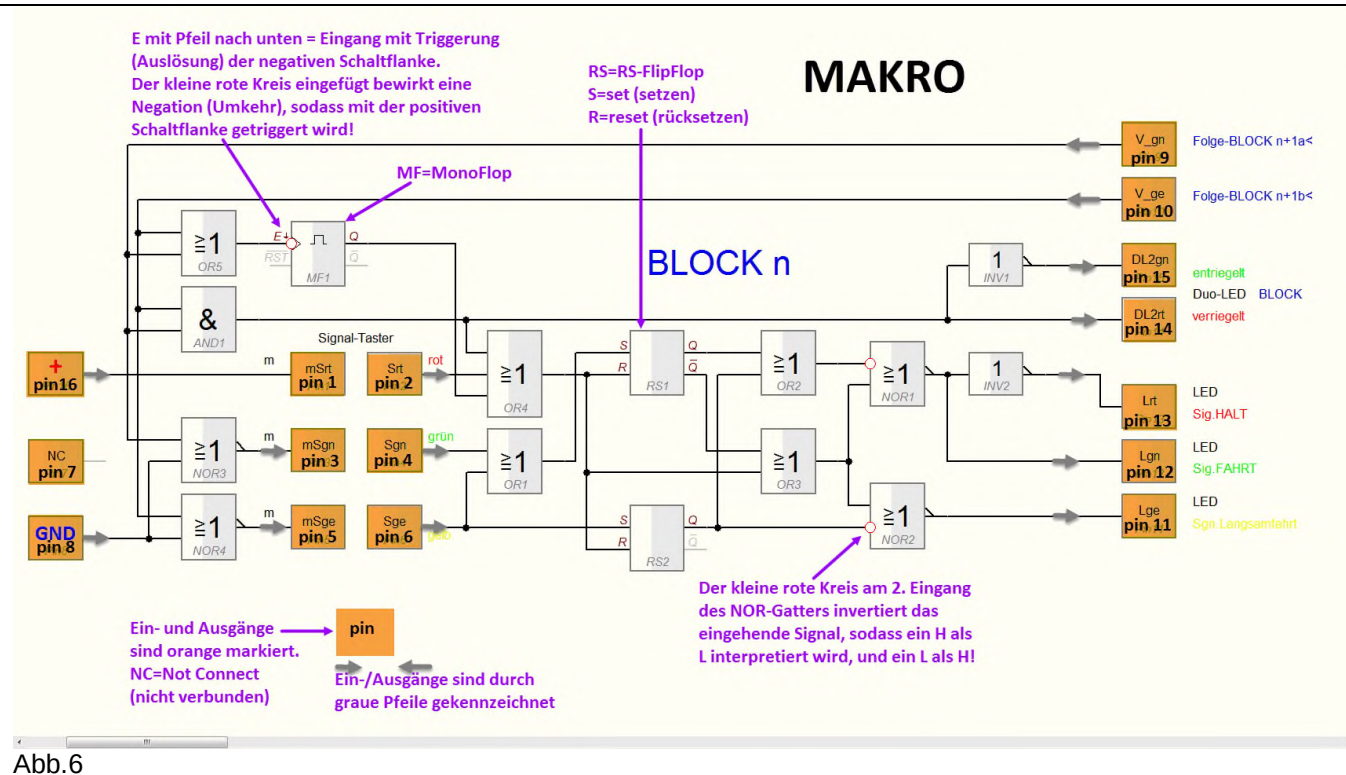


Abb.6

Die Gatter kennen wir bereits aus der weiter oben stehenden Abbildung. Kurz noch zu zwei neuen Bauteilen:

MF=MonoFlop: ein Rechtecksignal triggert den Baustein, sodass sein Ausgang Q eine gewisse Zeit (einstellbar) auf H geht, um dann selbsttätig wieder auf L zu gehen.

RS=RS-FlipFlop: ein kleiner Speicher. Ein kurzes H-Signal an Eingang S (set=setzen) schaltet den Ausgang Q auf H und damit Q quer auf L. Zum Rücksetzen dient der R-Eingang (reset=rücksetzen); ein kurzer H-Impuls an R schaltet also Q auf L und Q quer auf H.

An **pin9** wird gemeldet, ob der nachfolgende Block (hier für den geradeaus folgenden Block (deshalb "V_gn", gn = grün) besetzt ist.

An **pin10** wird gemeldet, ob der nachfolgende Block (hier für den abzweigenden folgenden Block (deshalb "V_ge", ge = gelb) besetzt ist.

Existiert nur ein Folgeblock (kein Abzweig), so bleibt pin10 einfach frei.

Existieren mehr als 2 Abzweige (Folgeblöcke), so werden die Eingänge der nachfolgenden Gatter (OR5, AND1, NOR3, NOR4) einfach um weitere Eingänge erweitert.

Anmerkung: allgemein betrachten wir einen Block "n"; ein Block davor bezeichnen wir dann mit "n-1", einen Block danach mit "n+1", wobei "n" für 1, 2, 3, usw. steht.



Das Makro als IC dargestellt:

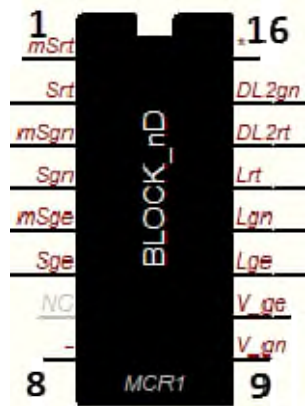


Abb.7

pin15 führt zum Anschluss einer Duo-LED grün (DL2gn), welche signalisiert, dass das Ausfahrtsignal des Blocks auf Grün geschaltet werden kann.
pin16 führt zum Anschluss der selben Duo-LED rot (DL2rt), welche signalisiert, dass das Ausfahrtsignal des Blocks nicht auf Grün geschaltet werden kann (Taster gesperrt).
pin13 führt zum Anschluss LED rot (Lrt): Signal steht auf Halt
pin12 führt zum Anschluss LED grün (Lgn): Signal steht auf Fahrt (Weiche gerade)
pin11 führt zum Anschluss LED gelb (Lge): Signal steht auf Langsamfahrt (Weiche Abzweig)
pin1 Mittelanschluss des Tasters für Signal rot (mSrt)
pin2 Ausgangsanschluss des Tasters für Signal rot (Srt)
pin3 Mittelanschluss des Tasters für Signal grün (mSgn)
pin4 Ausgangsanschluss des Tasters für Signal grün (Sgn)
pin5 Mittelanschluss des Tasters für Signal gelb (mSge)
pin6 Ausgangsanschluss des Tasters für Signal gelb (Sge)
pin16 Spannungseingang **+** (im Simulationsprogramm **H**, in einer Schaltung z.B. **+5V**)
pin 8 Spannungseingang **-** (im Simulationsprogramm **L**, in einer Schaltung **Minus**; **GND** steht für "**Ground**"="**Masse**")
pin7 ist nicht belegt (NC=Not Connect, englisch: "nicht belegt")

Mittels Abbildung 6 sollte man logische Abläufe durchspielen, in dem man (bei Zuhilfenahme der Gatterdarstellungen aus Abbildung 3) Ein- und Ausgänge entsprechend mit **H**- und **L**-Pegel versieht.

Als Anfangshilfe hierzu zeigen die folgenden drei Seiten Makros der Abb.6 mit eingetragenen **H**- und **L**-Pegeln für die Zustände

1. Signal HALT
2. Signal auf Grün stellen
3. Signal Grün

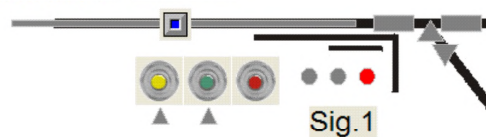
an.

Versuchen Sie auf diese Weise weitere Schaltungs-Stellungen selber einzutragen (beispielsweise "Signal auf Abzweig"), so üben Sie Schaltung und Gatter besser zu verstehen.



1. Makro Signal HALT

BLOCK 1 Besetzt-Melder



Ausgangszustand:
BLOCK 1 nicht besetzt
Signal HALT
Folgeblöcke nicht besetzt

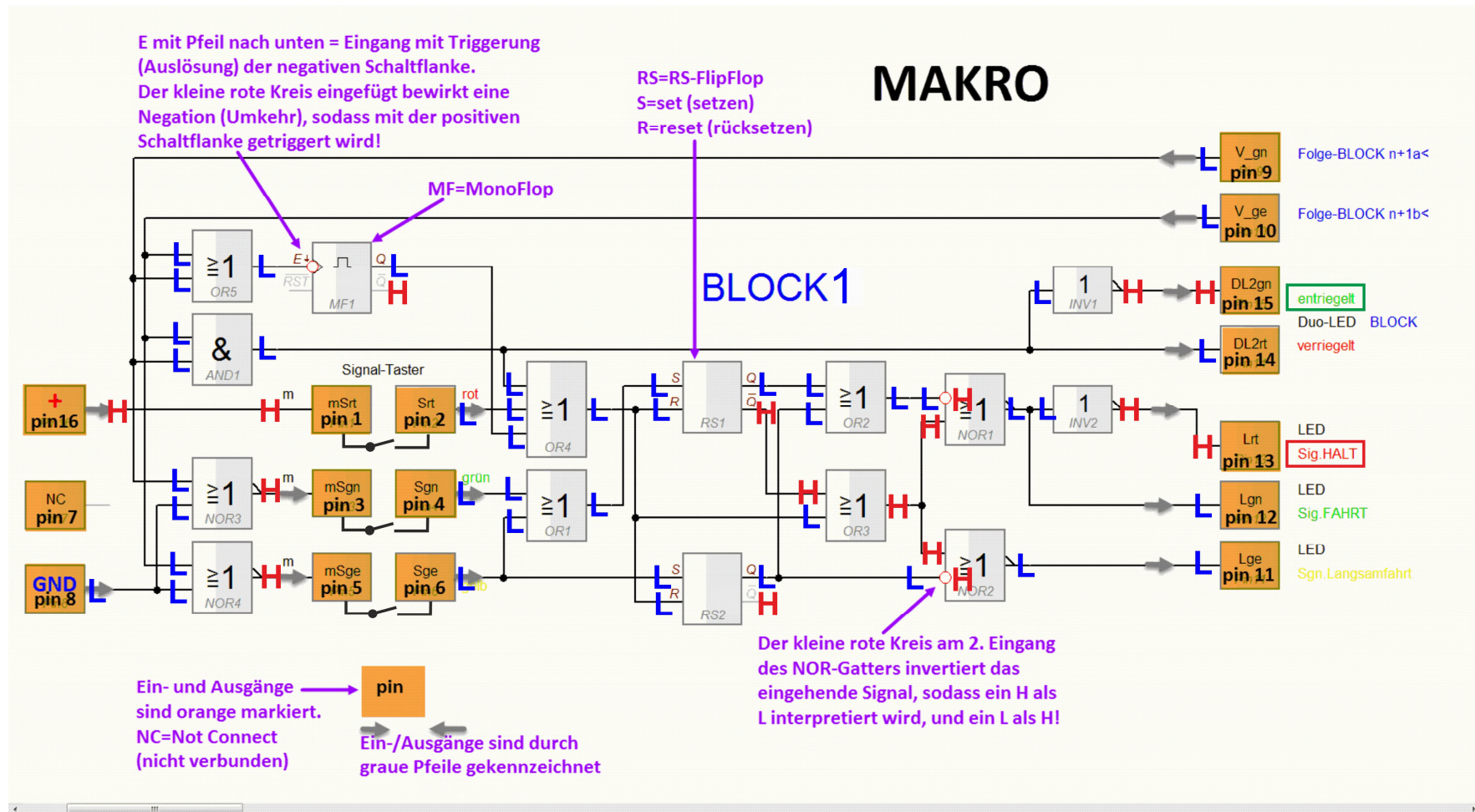
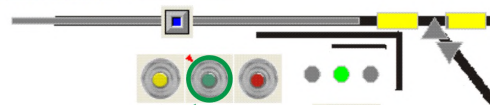


Abb.8



2. Makro Signal Grün stellen

BLOCK 1 Besetzt-Melder



Sig. 1

Taste gedrückt

Signal (Sig.1) auf Fahrt (Grün) stellen:
BLOCK 1 nicht besetzt
Signal HALT=> Fahrt Stellvorgang !
Folgeblöcke nicht besetzt

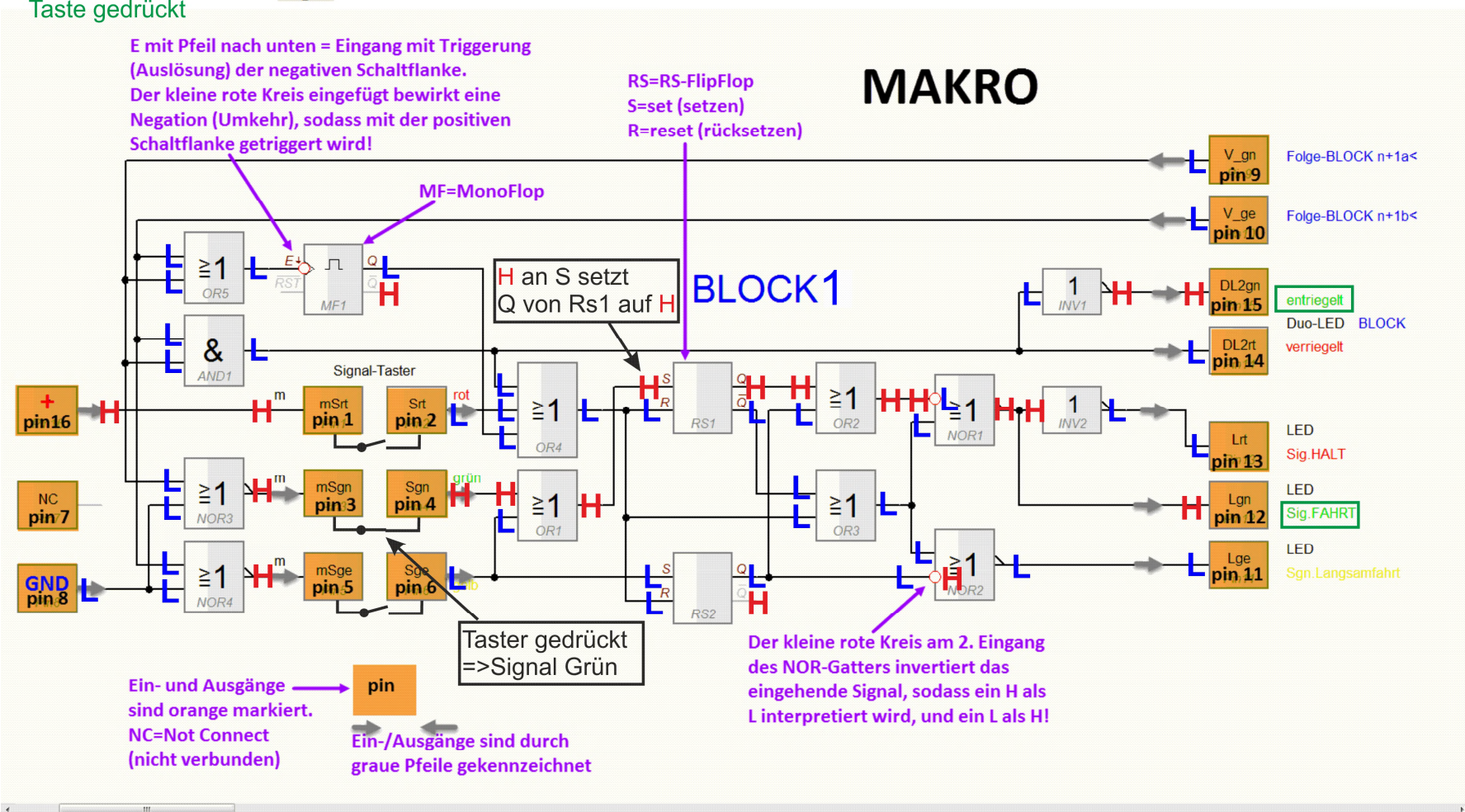
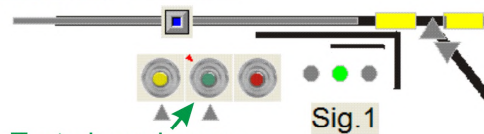


Abb.9



3. Makro Signal Grün

BLOCK 1 Besetzt-Melder



Signal (Sig.1) auf Fahrt (Grün) gestellt:
BLOCK 1 nicht besetzt
Signal Fahrt (Grün)
Folgeblöcke nicht besetzt



Taste losgelassen

E mit Pfeil nach unten = Eingang mit Triggerung (Auslösung) der negativen Schaltflanke.
Der kleine rote Kreis eingefügt bewirkt eine Negation (Umkehr), sodass mit der positiven Schaltflanke getriggert wird!

RS=RS-FlipFlop
S=set (setzen)
R=reset (rücksetzen)

MAKRO

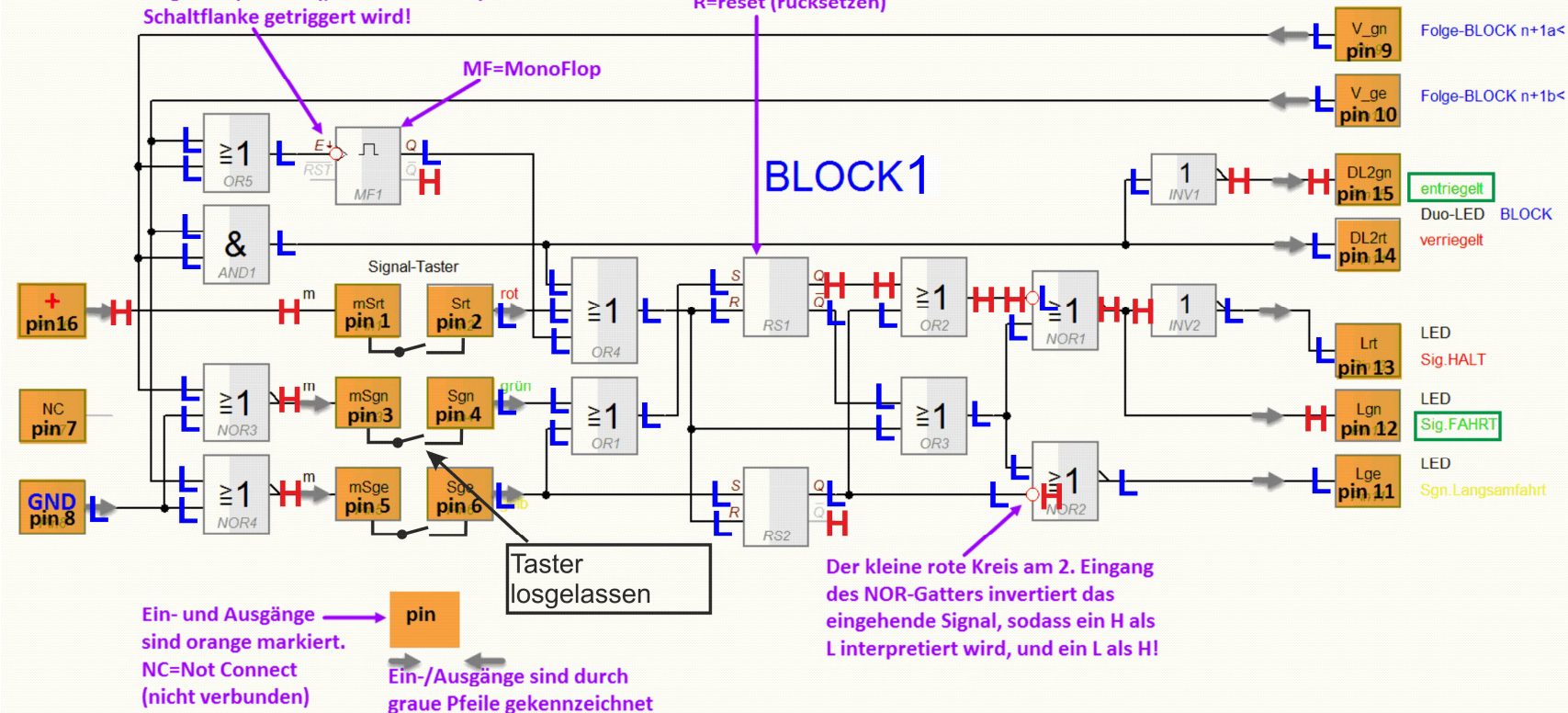


Abb.10



Steuerungen => Grundlagen Blocksteuerung, wie funktioniert das? Mit ergänzenden Informationen

Haben wir dieses Makro in seiner Funktion verstanden, können wir uns der Gesamtschaltung des Blocksteuerungsbeispiels zuwenden:

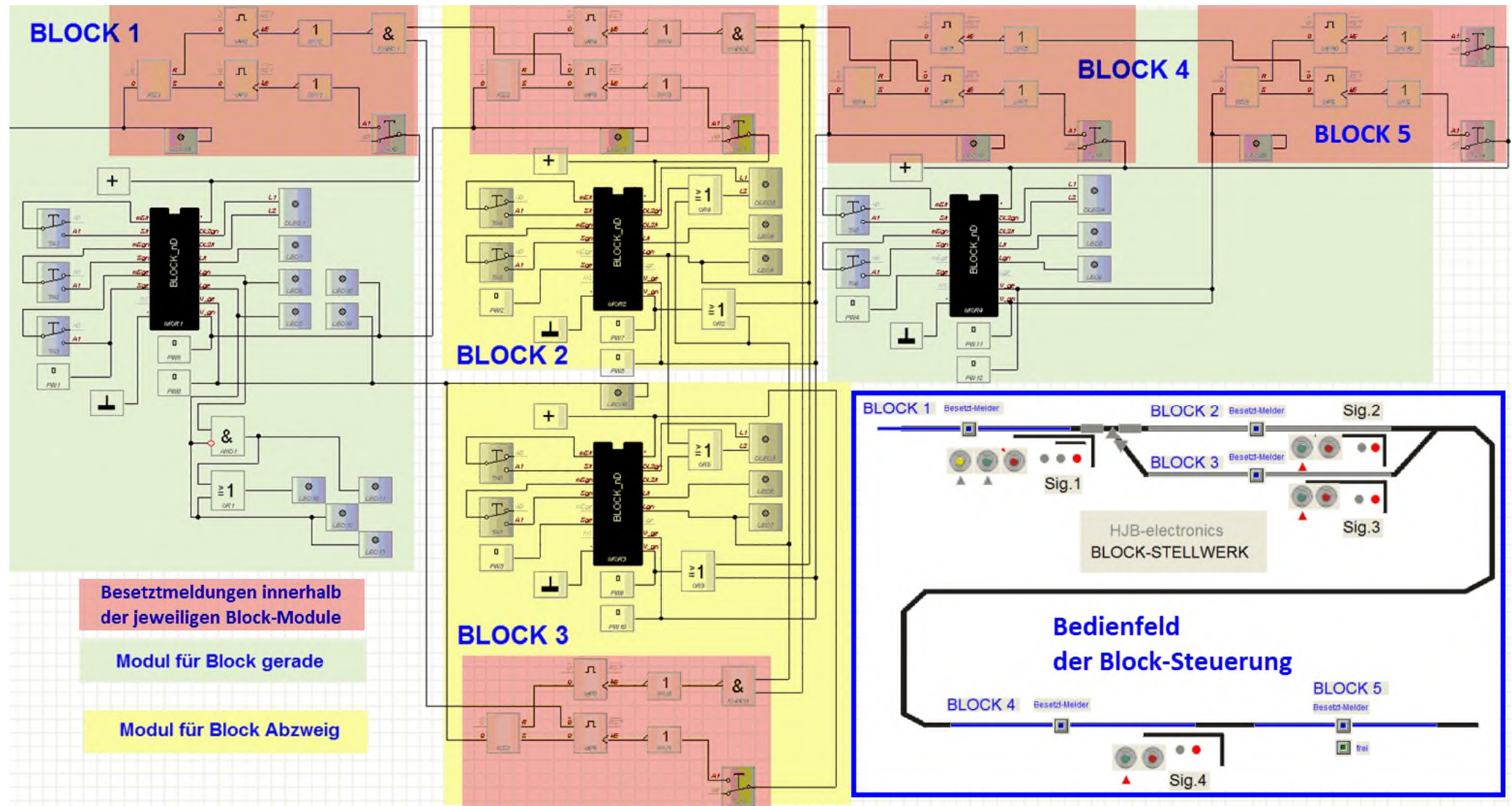


Abb.11



Steuerungen => Grundlagen Blocksteuerung, wie funktioniert das? Mit ergänzenden Informationen

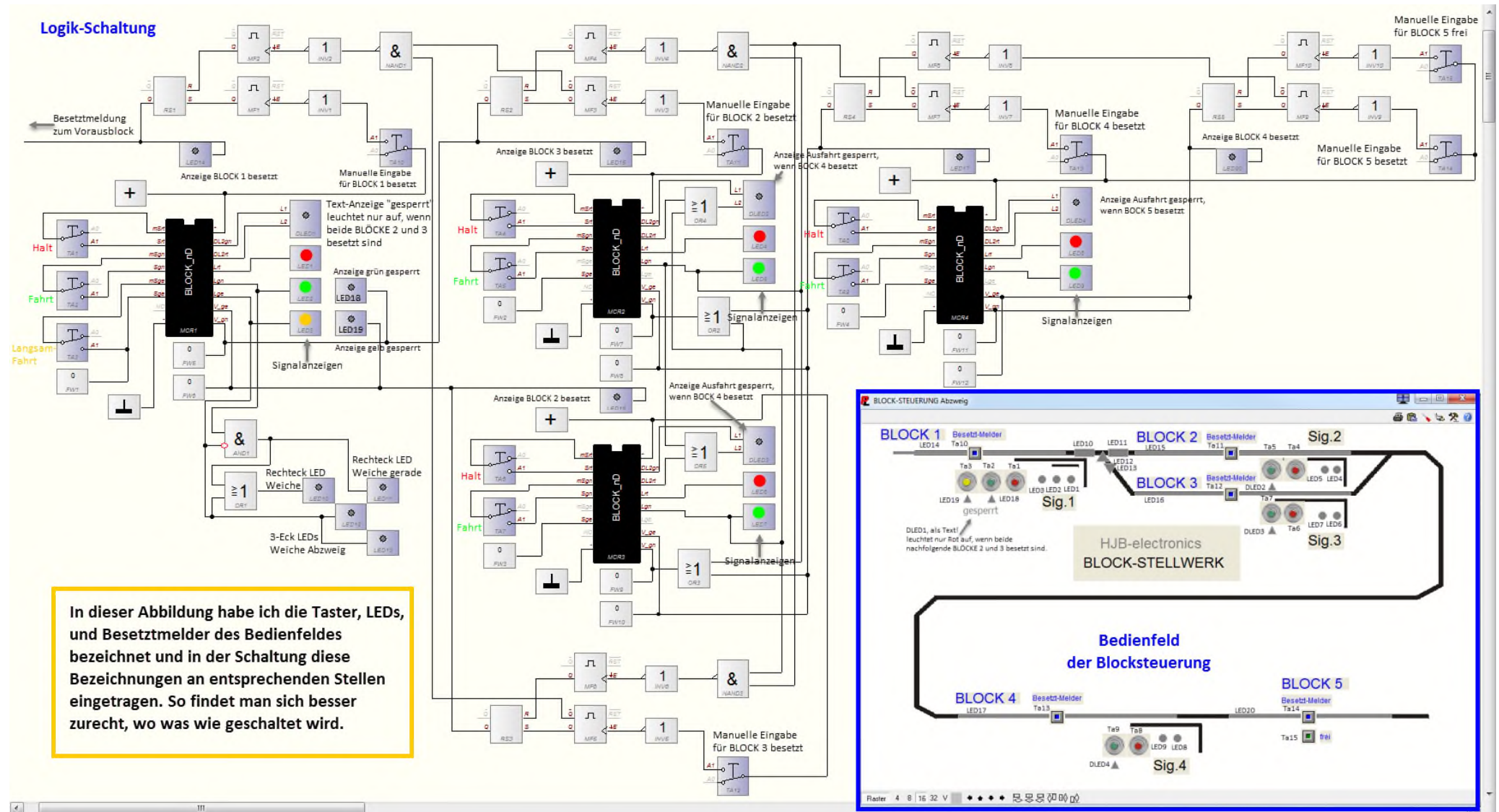


Abb.12



<http://www.hjb-modellbahn.de>

Steuerungen => Grundlagen Blocksteuerung, wie funktioniert das? Mit ergänzenden Informationen

Sehen wir uns im Folgenden noch die **3 Schaltfolgen**:

1. Signal HALT
2. Signal auf Grün stellen
3. Signal Grün

aus den obigen Makros (Abbildungen 8, 9, und 19) **in der Gesamtschaltung** an.

Darin stellt das Programm ProfiLab Expert Leitungen mit H-Signal in Rot dar und Leitungen mit L-Signal Schwarz.



Steuerungen => Grundlagen Blocksteuerung, wie funktioniert das? Mit ergänzenden Informationen

1. Schaltung Signal HALT

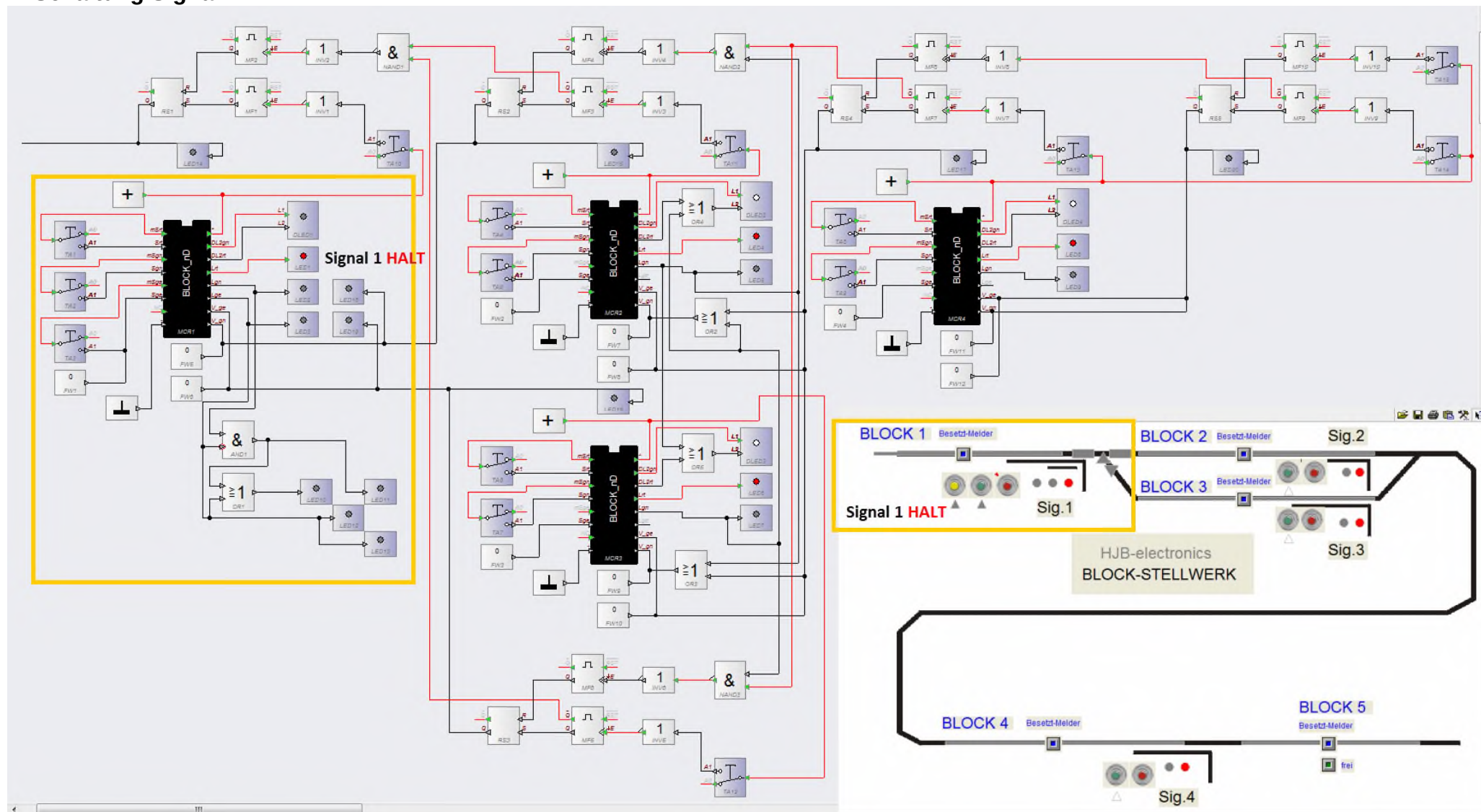


Abb.13



Steuerungen => Grundlagen Blocksteuerung, wie funktioniert das? Mit ergänzenden Informationen

2. Makro Signal Grün stellen

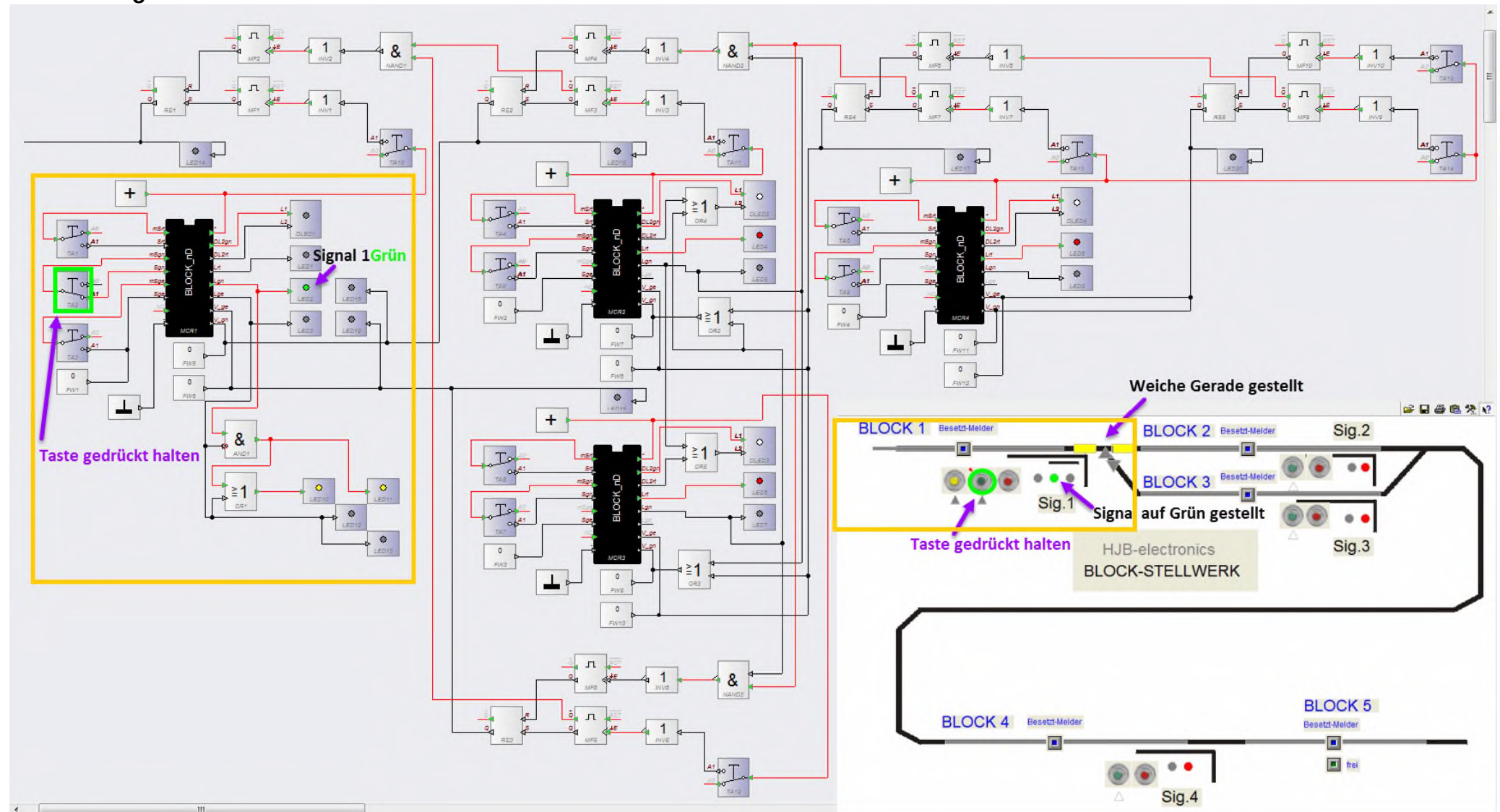


Abb.14



Steuerungen => Grundlagen Blocksteuerung, wie funktioniert das? Mit ergänzenden Informationen

3. Makro Signal Grün

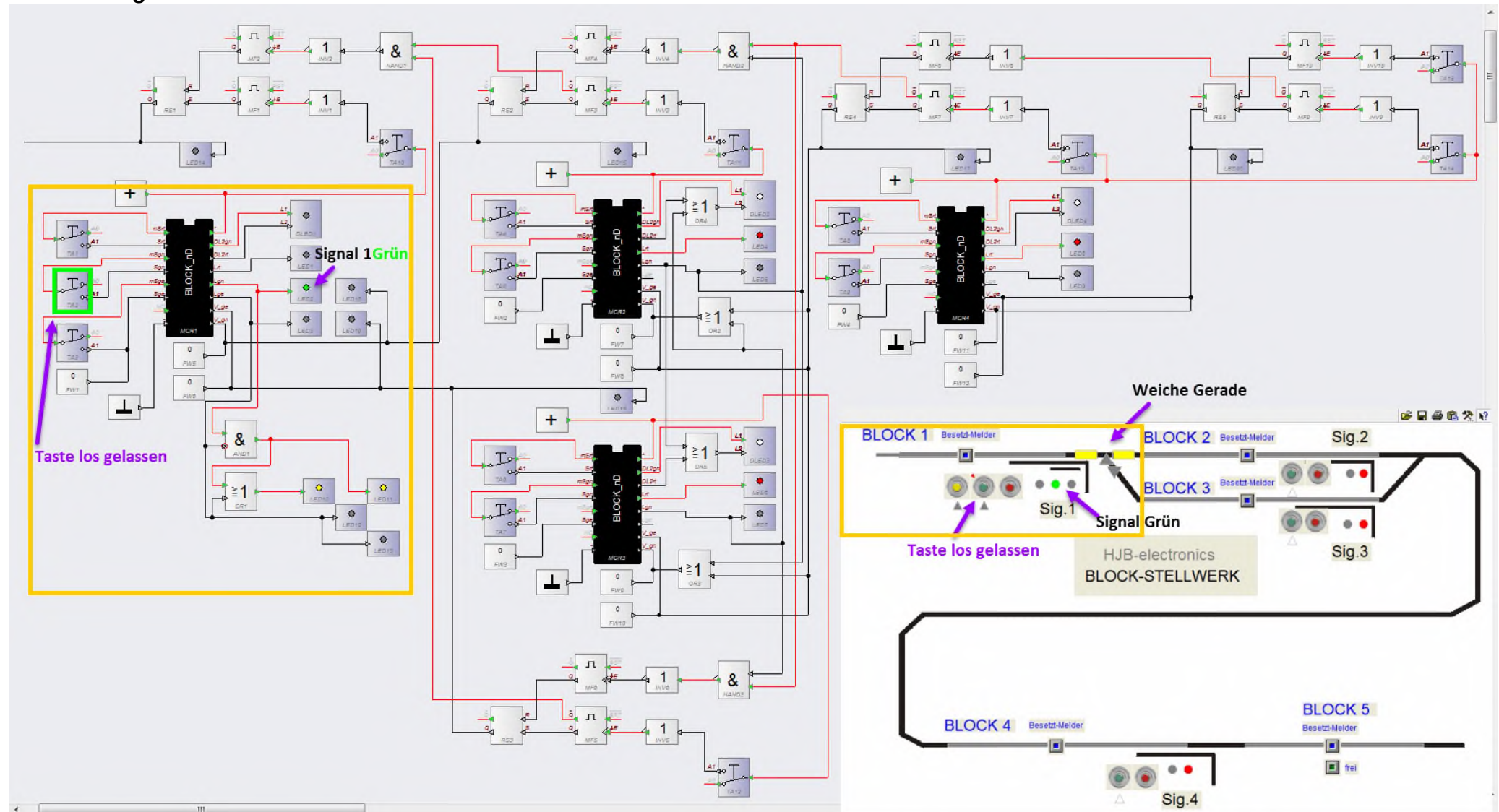


Abb.15



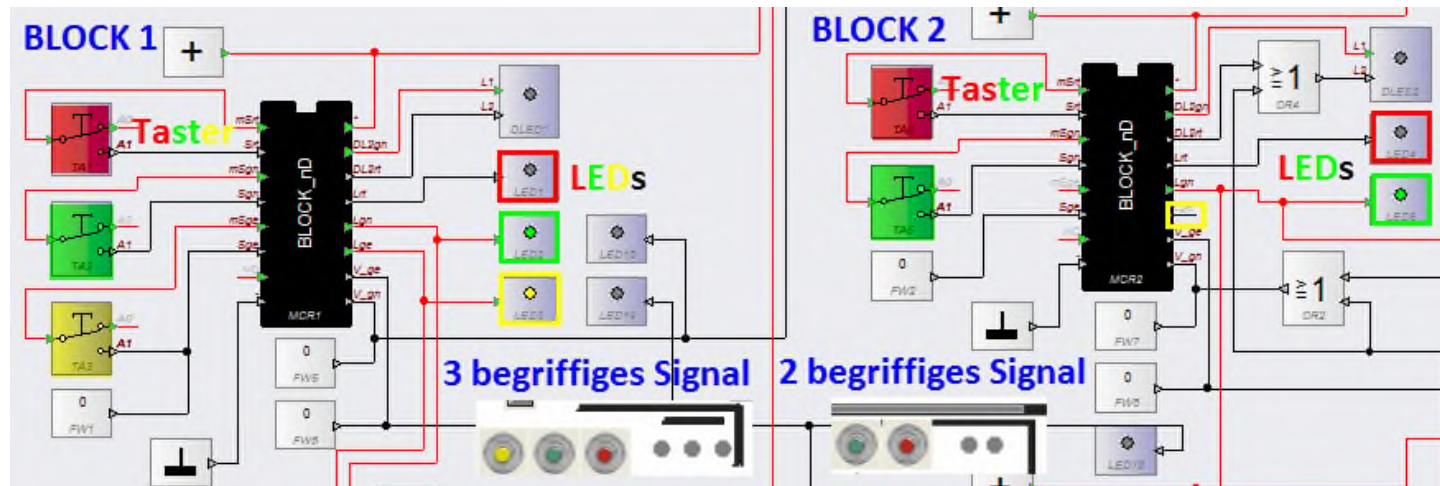
Steuerungen => Grundlagen Blocksteuerung, wie funktioniert das? Mit ergänzenden Informationen

Noch ein Hinweis, wie universal (und damit modular!) eine Schaltung realisiert werden kann:

BLOCK 1 besitzt ein 3-begriffiges Signal (Abzweig), wohingegen BLOCK 2 ein 2-begriffiges Signal (Ausfahrt) besitzt.

In der Schaltung wird am MAKRO (oder IC) für BLOCK 2 einfach kein Taster Gelb (für Langsamfahrt) angeschlossen, und der entsprechende Ausgang für die gelbe LED bleibt einfach frei. So lässt sich die Schaltung modular und lässt sich leicht beliebig erweitern.

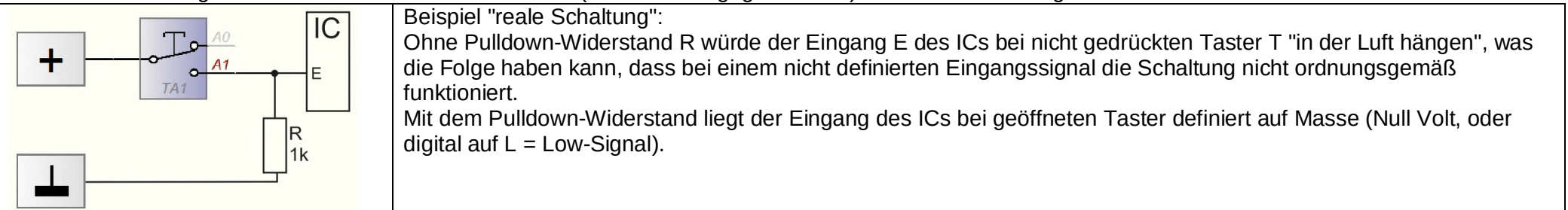
Die folgende Abbildung zeigt dies in einem Ausschnitt der Schaltung:



Anmerkung:

Die Schaltungsbauteile "0 FW1", "0 FW 1", usw. stellen in ProfiLab **Festwerte** da, hier mit dem Wert 0 (also Null Volt analog, oder L digital).

Die Funktion ist vergleichbar mit Pull-Down-Widerständen (Widerstände gegen Masse) bei realen Schaltungen:





<http://www.hjb-modellbahn.de>

Steuerungen => Grundlagen Blocksteuerung, wie funktioniert das? Mit ergänzenden Informationen

Eine noch bessere Darstellung und Möglichkeiten erhält man, wenn man sich von der ABACOM-Seite das Programm Profilab-Expert herunter lädt (funktioniert als Demo), installiert es, und lädt darin die Datei "BLOCK-Steuerung Abzweig Taster VGA.prj", welche auf meiner Internetseite zum Download verfügbar ist. Auch kann man dann noch einiges mehr selber ausprobieren.

Ebenso steht auf meiner Internetseite eine ausführbare Exe-Datei als Demo für das Beispiel 4 Blockstellen zum Download bereit.

diese einführende Beschreibungen mögen fürs Erste genügen. Natürlich ließe sich darüber noch viel mehr schreiben, doch "selbst ist der Mann"! Wenn man sich weiter mit dieser Thematik beschäftigt - sei es auch nur per Simulation - so wird man immer besser verstehen angebotene Modellbahnsteuerungsprogramme für seine eigenen Ziele zu "programmieren". **Logik Verstehen ist Alles!** Wenn Ihnen dazu mein kleiner Beitrag geholfen haben sollte, würde es mich freuen.

Ausblick:

Mit dem Programm ProfiLab kann nicht nur auf dem Rechner simuliert werden. Nein, es kann auch Hardware angeschlossen werden! So können Teile in der ProfiLab-Simulation "ausgelagert" werden und durch Hardware ersetzt werden. Die Firma ABACOM bietet dazu einiges an; sehen Sie auf der Internetseite nach. Ich nenne hier als Beispiel "USB-uPIO / GPIO18": PC-Interface mit 18 digitalen I/O-Kanälen. Damit können dann bereits real Rückmelder (auch Anzeigen, Signal- und Weichen-Stellen) einer Modellbahn per Rechner mit diesem Programm realisiert werden: kreieren Sie Ihr eigenes BLOCK-Stellwerk!