

Digibahn

**Automatisierung mit
dem Makro Manager**



Einleitung

Der Makro Manager

Die Modellbahnsteuerung Digibahn besteht aus den drei Hauptkomponenten Loksteuerung, Stellpult und Makromanager. Die beiden Programme Loksteuerung und Stellpult ermöglichen die Steuerung der Anlage, was mit diesen Programmen von Hand geschieht. Der Makro Manager hingegen ist für die Automatisierung der Anlage zuständig

Für diese Aufgaben kann der Makromanager vielseitig eingesetzt werden. Ein Automatikbetrieb kann von dem einfachen Schalten einer Fahrstraße bis zum vollautomatischen Betrieb reichen. Besonders interessant ist die gemeinsame Verwendung mit den Steuerungsprogrammen, was einen vielseitigen Anlagenbetrieb ermöglicht.

Dieses Handbuch widmet sich dem Makro Manager und vermittelt die Grundlagen die für eine erfolgreiche Makroprogrammierung erforderlich sind.

Was sind Makros?

Makros sind kleine (oder auch umfangreiche) Programme mit denen die Anlage gesteuert wird.

Aber keine Angst, es müssen dazu keine Programmierkenntnisse vorhanden sein. Der Makro Manager erlaubt es Steuerungsbefehle aus den Programmen Stellpult und Loksteuerung aufzuzeichnen und später beliebig oft wiederzugeben. So kann man sich z.B. eine Fahrstraße durch anklicken der benötigten Weichen aufzeichnen. Durch Anklicken des so erstellten Makros kann später die Fahrstraße beliebig oft wieder geschaltet werden, ohne dass man die Weichen einzeln anklickt.

Auf diese Art und Weise können Makros erstellt werden, ohne dass man einen einzigen Befehl der Makrosprache beherrscht. Wirklich viele Möglichkeiten ergeben sich, wenn man sich mit der Makrosprache beschäftigt und durch die Eingabe von Makrobefehlen die Makros entsprechend gestaltet. Da die Makrosprache aus einfachen deutschsprachigen Sätzen besteht, ist sie einfach zu verstehen.

Etwas kniffliger wird es bei der Verwendung von Variablen. Durch die Verwendung von Variablen hat man fast unbegrenzte Möglichkeiten. Aus diesem Grund widme ich ihnen ein eigenes Kapitel.

Der Inhalt dieses Handbuches wurde mit großer Sorgfalt erarbeitet. Dennoch können Fehler nicht ausgeschlossen werden. Ich weise deshalb darauf hin, dass ich weder eine Gewähr noch irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen kann. Für die Mitteilung etwaiger Fehler bin ich jederzeit dankbar.

© 2011 bei Joachim Baumann, Katharinenweg 10, 72135 Dettenhausen, Deutschland
www.digibahn.de

Inhalt

Arbeiten mit dem Makro-Manager	4
Der Makro-Manager	4
Das Makro Gruppenfenster	4
Das Makro Editorfenster	5
Makrosteuerung durch Flagge	5
Makroaufzeichnung mit Stellpult und Loksteuerung	6
Makro aufzeichnen.....	6
Kommunikation mit dem Makro Manager.....	7
Die Makroaufzeichnung mit Schablone	7
Aktionstasten im Stellpult	8
Automatischer Makrostart.....	8
Die Makrosprache	9
Aufbau der Makrosprache	9
Weichen, Signale und Ausgänge stellen	10
Fahrbefehle für Lokomotiven	11
Warten auf Bedingung	13
Bedingte Ausführung	14
Sprünge	16
Klänge und Geräusche	18
Makrosteuerung	19
Variablen und Ausdrücke.....	21
Bedienereingaben.....	23
Lok-Position auswerten.....	25
Belegtmelder abfragen	26
Zufällige Makroabläufe.....	28
Flexible Makros mit Variablen	29
Verwendung von Variablen	29
Einsatzbereiche von Variablen.....	31
Tipps zur Fehlersuche	32
Syntaxfehler	32
Logische Fehler.....	33
Beispielmakros.....	36
Einfache Beispiele.....	36
Beispielanlage	43
Anhang	49
Befehlsübersicht.....	49

Arbeiten mit dem Makro-Manager

Der Makro-Manager

Aufgaben von Makros

Makros ermöglichen die Automatisierung von Anlagenabläufen. Mit den Programmen wie Stellpult können nur einzelne Funktionen ausgeführt werden, z.B. einzelne Weichen nacheinander zu stellen. Der Makro-Manager ermöglicht es die Aktivitäten von den anderen Programmen aufzuzeichnen. Diese Aktivitäten werden in einem Makro gespeichert und können dann beliebig oft wieder ausgeführt werden.

Makros können auch von Hand bearbeitet werden. Dadurch sind Schleifen und vieles mehr möglich.

Makros können für verschiedene Aufgaben erstellt werden:

- Fahrstraßen stellen
- Automatisches Fahren eines Zuges
- komplexe Steuerung mehrerer Züge und Fahrstrecken
- Überwachung von Anlagenabläufen

Makros organisieren

Die Makros werden in Gruppen organisiert. Alle gleichartigen, zusammengehörenden Makros können in je einer Gruppe zusammengefasst werden.

Sie können die Makros in einer Gruppe aus dem Gruppenfenster starten.

Makro aufzeichnen

Aus jedem der Steuerungsprogramme kann eine Makroaufnahme gestartet werden. Andere Programme haben die Möglichkeit sich an der Makroaufnahme zu beteiligen. Die Programme schreiben die vorgenommenen Aktionen in das Makro.

Die beteiligten Programme können zwischendurch die Aufzeichnung verlassen und danach nochmals teilnehmen. Sobald das Programm, das die Aufzeichnung gestartet hat, beendet wird, wird auch die Aufnahme des Makros beendet.

Makros können von Hand editiert werden. Sie können z.B. Schleifen und Bedingungen einfügen. Ein Makro kann natürlich auch komplett von Hand geschrieben werden.

Makro wiedergeben

Ein einmal aufgezeichnetes Makro kann beliebig oft wiedergegeben werden. Klicken Sie einfach mit der Maus doppelt auf das Makro im Gruppenfenster

Das Makro Gruppenfenster



Ein Gruppenfenster stellt alle Makros dar, die in einer Gruppe enthalten sind.

Vor jedem Makro zeigt ein Symbol den Status des Makros an:



Makro wird nicht bearbeitet.



Makro wird ausgeführt.



Makro wird ausgeführt, Flagge ist gesetzt.



Makro wird als Text bearbeitet.



Makro wird aufgenommen.



Makro ist fehlerhaft. Ausführung wurde abgebrochen.

Sie öffnen ein Gruppenfenster im Menü „Makro/Gruppe öffnen“

Verwendung des Gruppenfensters

Makro ausführen

Maus: Klicken Sie doppelt auf das Makro.

Tastatur: Markieren Sie das Makro und drücken die Eingabetaste.

Flagge setzen/löschen

Maus: Klicken Sie doppelt auf das Makro.

Tastatur: Markieren Sie das Makro und drücken die Eingabetaste.

Die Flagge kann nur bei einem Makro gesetzt werden, das gerade ausgeführt wird.

Makro verschieben

Maus: Klicken Sie das Makro an und ziehen es an eine andere Stelle in der Liste oder in eine andere Gruppe.

Makro oder Gruppe löschen

Tastatur: Markieren Sie das Makro und drücken die Entf-Taste. Wenn Sie das letzte Makro löschen wird auch die Gruppe gelöscht.

Fehlerhaftes Makro bearbeiten

Wenn Sie versuchen ein fehlerhaftes Makro zu starten wird der Makroeditor geöffnet.

Hinweis:

Klicken Sie mit der rechten Maustaste auf ein Makro. Sie erhalten dann ein kleines Menü, in dem Sie alle z. Z. möglichen Aktionen auswählen können.

Das Makro Editorfenster

Ein Editorfenster stellt den Inhalt eines Makros als Text dar.



Im Editorfenster können Makros bearbeitet werden.

Es können beliebig viele Editorfenster geöffnet werden. Ein Makro, das als Text editiert wird kann nicht gleichzeitig ausgeführt werden.

Sie öffnen ein Editorfenster im Menü „Bearbeiten/Als Text“

Hinweis:

So erhalten Sie schnelle Hilfe über einen Befehl: Stellen Sie den Cursor auf den entsprechenden Befehl und drücken die Tasten Strg+F1.

Makrosteuerung durch Flagge

Über die Flagge kann der Bediener die Makroausführung steuern.

Durch doppelklicken mit der Maus auf ein laufendes Makro im Gruppenfenster wird die Flagge gesetzt. Durch ein weiteres doppelklicken wird sie wieder zurückgesetzt. Die Flagge kann auch durch die Enter-Taste gesetzt bzw. gelöscht werden.

Im Makro kann die Flagge abgefragt werden und der Bediener hat dadurch die Möglichkeit den Ablauf des Makros zu verändern. Die Abfrage der Flagge ist mit den Befehlen Wenn und Warte möglich.

Die Flagge kann nur gesetzt werden, wenn das Makro ausgeführt wird. Beim Makrostart ist die Flagge immer gelöscht.

Anwendungsmöglichkeiten

Makro definiert beenden

Viele Makros laufen in einer Endlosschleife (z.B. Zug fährt immer im Kreis). Ein solches Makro kann sonst nur über das Menü „Spezial/Makro abrechnen“ beendet werden. Dadurch fährt der Zug in der aktuellen Geschwindigkeit weiter und muss von Hand gestoppt werden.

Durch setzen der Flagge kann ein gezieltes Anhalten des Zuges im Bahnhof eingeleitet werden. Das Makro fragt vor dem Bahnhof die Flagge ab und verzweigt gegebenenfalls in einen anderen Makroteil (Sprung) und bremst den Zug im Bahnhof ab.

Makroausführung beeinflussen

Über Die Flagge kann die Fortsetzung des Makros beeinflusst werden, so dass ein Zug bei gesetzter Flagge im Kreis 1 fährt und bei gelöschter Flagge in Kreis 2.

Makroaufzeichnung mit Stellpult und Loksteuerung

Aufzeichnungsmodis

Schablone ausfüllen

Während der Makroaufnahme werden keine Weichen und Signale auf der Anlage angesteuert. Der Gleisplan wird als leere Schablone dargestellt. Markieren Sie die Weichen und Signale so, wie sie im Makro gespeichert werden sollen. Wenn Sie den Befehl „Makro/Ablegen“ im Menü wählen werden die markieren Weichen- und Signalzustände im Makro abgelegt.

Aktivitäten aufzeichnen

Während der Makroaufnahme werden weiterhin Weichen und Signale auf der Anlage angesteuert. Der Gleisplan wird als Schablone dargestellt und sie können Weichen und Signale markieren wie bei „Schablone ausfüllen“.

Live aufzeichnen

Weichen und Signale werden normal angesteuert. Zusätzlich werden alle Stellvorgänge (sofort) im Makro abgelegt.

Makro aufzeichnen

Mit dem Menü „Makros“ des Stellpults und der Loksteuerung können Makros aufgezeichnet werden.



Aufnehmen

Startet die Makroaufnahme.



Speichern

Beendet die Makroaufnahme. Markierte Gleise werden in das Makro geschrieben.



Abbrechen

Beendet die Makroaufnahme, ohne die markierten Gleise in das Makro zu schreiben.



Aufnahme ablegen

Schreibt alle markierten Gleise in das Makro.



Aufnahme löschen

Löscht alle Markierungen, ohne ins Makro zu schreiben.

Kommunikation mit dem Makro Manager

Alle Makros werden vom Makro Manager verwaltet. Das Stellpult und die Loksteuerung können Makrobefehle an den Makro Manager weitergeben, die dieser in einem Makro speichert.

Der Makromanager tauscht mit den anderen Programmen Informationen aus. Dadurch können sich mehrere Programme an einer Makroaufnahme beteiligen. Sie können gleichzeitig Befehle in ein Makro schreiben.

Ein Programm startet eine Makroaufnahme. Es bleibt den anderen Programmen überlassen, ob sich diese an der Makroaufnahme beteiligen. Sie können sich jederzeit manuell über das Menü Makro an eine bestehende Aufnahme einklinken oder diese verlassen. Das Programm, das die Makroaufnahme gestartet hat beendet die Aufnahme.

Die Makroaufnahme mit Schablone

Was ist die Schablone?

Im normalen Betrieb werden im Gleisplan die Zustände von Weichen, Signalen, Ausgängen und Eingängen angezeigt.

Wenn Sie ein Makro aufzeichnen ist der gesamte Gleisplan leer. Der aktuelle Status der verschiedenen Gleise ist während der Makroaufnahme unwichtig.

Der Gleisplan ist eine Schablone, mit der Sie Makrostellungen in den Plan zeichnen können:

Klicken Sie alle Gleise an, die in das Makro eingefügt werden sollen, um sie zu markieren. Alle Gleise, die Sie auf diese Weise markiert haben, werden nun mit dem gewünschten Zustand angezeigt.

Die Lokregler werden ebenfalls als Schablone angezeigt und ändert die Farbe des Schiebereglers.

Im Makro ablegen (aus dem Stellpult)

Wenn Sie die gewünschten Stellungen hergestellt haben, werden diese im Makro abgelegt.

Die markierten Gleise werden immer in dieser Reihenfolge im Makro abgelegt:

1. Wartebedingung Schaltkontakt oder Eingang
2. Signale, die auf Halt gestellt werden
3. Weichen und Ausgänge
4. Signale, die auf Fahrt gestellt werden

Sie können beliebig oft Gleisstellungen im Makro ablegen.

Wartebedingungen

Mit Schaltkontakten und Eingängen können Sie Wartebedingungen herstellen. Es kann immer nur ein Schaltkontakt oder ein Eingang als Wartebedingung ausgewählt werden.

Schaltkontakte

Schaltkontakte werden nur während der Makroaufnahme angezeigt.

Das Makro wird später die gewünschten Stellungen ausführen, sobald eine Lok den gewählten Schaltkontakt betätigt.

Eingänge

Die Eingänge zeigen auch während der Makroaufzeichnungen ihren aktuellen Zustand an.

Sobald Sie einen Eingang anwählen, wird dieser als Wartebedingung aktiviert. Es kann auf den Zustand 'Ein' oder 'Aus' gewartet werden (die Vorzugsstellung ist die gerade nicht aktive).

Hinweis:

Ein aktivierter Eingang wird in den Makrofarben eines Ausgangs dargestellt. Diese Farben sollten deshalb unterschiedlich sein.

Aktionstasten im Stellpult

Eine Aktionstaste ist eine Schaltfläche, auf der ein kleines Makro hinterlegt werden kann. Wird die Taste später angeklickt, so wird die programmierte Aktion ausgeführt. Somit ist es möglich, Makrofunktionen direkt vom Stellpult aus ausführen zu können.

Aktionstaste in Bahnplan erstellen

Mit einer Aktionstaste können einfache Makros ausgeführt werden. Es können alle Befehle des Makromanagers verwendet werden.

Es ist dadurch möglich, Makros aus dem Stellpult heraus auszuführen

Einfachere Aktionen, wie z.B. das Schalten einer Fahrstraße können direkt programmiert werden. Bei aufwändigeren Aktionen werden diese in einem Makro abgelegt und dieses mit dem Befehl Start Makro aufgerufen.

Wichtig:

Es können keine Wartebedingungen in einer Aktion programmiert werden! Starten Sie in diesem Fall ein Makro, dieses wird normal abgearbeitet. Um fehlerhafte Aktionen zu stoppen bricht der Makromanager eine Aktion nach 10 Sekunden ab.

Beschriftung

Geben Sie eine Beschriftung für die Taste ein.

Aktion

Geben Sie hier die Befehle ein, die ausgeführt werden sollen.

Hinweis:

Wenn Sie mehrere Befehle eintippen möchten, verwenden Sie die Tastenkombination Strg+Enter um eine neue Zeile zu erzeugen.

Automatischer Makrostart

Es ist möglich, ein Makro automatisch beim Start des Makromanagers ausführen zu lassen. Ebenso kann beim Beenden des Makromanagers ein Makro ausgeführt werden.

So ist es z.B. möglich, eine Grundstellung auf der Anlage einzustellen oder andere Initialisierungsmaßnahmen durchzuführen.

Hinweis:

Wenn Sie beim Programmieren ein Makro ausführen, verzögert dies das Schließen des Makro Managers, da das Makro zuerst vollständig abgearbeitet werden muss.

Die Makrosprache

Aufbau der Makrosprache

Die Makrosprache besteht aus einfachen deutschsprachigen Anweisungen. Jeder Makrobefehl ist ein kurzer Satz, der mit einem Punkt endet.

Die einzelnen Worte in einem Makrobefehl werden durch Leerzeichen getrennt. Es können aber auch Tabulatoren und Zeilenumbrüche verwendet werden, da sie wie Leerzeichen behandelt werden. Die Anzahl der Leerzeichen, Tabulatoren und Zeilenumbrüche ist beliebig.

Groß-/Kleinschreibung wird nicht unterschieden.

Normalerweise steht in einer Zeile jeweils ein Befehl. Dies ist aber keineswegs zwingend, sondern dient lediglich der Übersichtlichkeit.

In ein Makro können Kommentare eingefügt werden. Alles was in geschweiften Klammern eingeschlossen ist wird als Kommentar behandelt.

In einem Makro werden alle Befehle hintereinander angereiht. Das Makro wird Befehl für Befehl abgearbeitet. Wenn der letzte Befehl ausgeführt wurde wird das Makro beendet.

Tipp: Kommentare

{Kommentar} Texte in geschweiften Klammern werden ignoriert.

Machen Sie von der Möglichkeit Gebrauch, ihr Makro mit passenden Hinweisen zu kommentieren. So können Sie auch nach längerer Zeit die Arbeitsweise des Makro noch gut verstehen.

Es ist auch möglich bestimmte Makrobefehle auszukommentieren. Setzen Sie Makrobefehle in geschweifte Klammern, die Sie momentan nicht benötigen. Dadurch ist es oft einfacher ein Makro zu testen, wenn bestimmte Passagen ignoriert werden sollen.

Befehlsbeschreibung

Auf den folgenden Seiten finden Sie eine Befehlsreferenz. Um die Befehle besser verstehen zu können, sind sie anhand von Beispielen erläutert.

In der Befehlsbeschreibung werden folgende Markierungen benutzt:

<Nummer> Angaben in spitzen Klammern geben an, was für eine Art von Angaben erwartet wird. Sie müssen hier eine Nummer z.B. 3 eingeben.

[Option] Angaben in eckigen Klammern sind optional. Sie können diese Angaben auch weglassen.

Weichen, Signale und Ausgänge stellen

Stelle Weiche <Nummer> gerade .

Stelle Weiche <Nummer> rund .

Stellt die Weiche, deren Nummer angegeben ist, gerade oder rund (schräg).

Mit diesem Befehl können normale Weichen und Kreuzungsweichen gestellt werden.

Stelle Dreiwegweiche <Nummer> gerade .

Stelle Dreiwegweiche <Nummer> links .

Stelle Dreiwegweiche <Nummer> rechts .

Stellt eine Dreiwegweiche in die gewünschte Fahrtrichtung.

Dreiwegweichen benötigen einen anderen Befehl als andere Weichen, da zwei Antriebe gesteuert werden müssen.

Stelle Signal <Nummer> Halt .

Stelle Signal <Nummer> Fahrt .

Stellt das Signal, dessen Nummer angegeben ist, auf Halt oder auf Fahrt.

Stelle Ausgang <Nummer> ein .

Stelle Ausgang <Nummer> aus .

Schaltet den angegebenen Ausgang ein oder aus.

Ausgänge können für die verschiedensten Zwecke eingesetzt werden. Es können Beleuchtungen, Motoren, Entkupplungsgleise usw. als Ausgang angesprochen werden.

Beispiele:

Stelle Weiche 7 rund.

Stelle Dreiwegweiche 15 links.

Stelle Signal 12 Fahrt.

Stelle Ausgang 3 ein.

Fahrbefehle für Lokomotiven

Geschwindigkeit steuern

Fahre Lok <Nummer> mit <Geschwindigkeit> .

Dies ist der Grundbefehl zur Ansteuerung vom Lokomotiven.

Es wird die Lok-Nummer und die gewünschte Geschwindigkeit angegeben.

Hinweis:

Der Makromanager unterscheidet nicht zwischen Lok- und Blockansteuerung. Mehrere Blöcke werden gleichzeitig angesteuert indem die Nummern, durch Komma getrennt, aufgelistet werden. Auf diese Weise kann auch eine Doppeltraktion realisiert werden.

Fahre Lok <Nummer> Stop .

Hält die Lok sofort an. Es wird keine Bremsverzögerung vorgenommen.

Fahre Lok <Nummer> andere Richtung .

Dieser Befehl gibt einen Umschaltimpuls an die Lokomotive aus. Die Lok fährt bei den folgenden Fahrbefehlen in die andere Richtung.

Dieser Befehl kann nur in Wechselspannungsanlagen verwendet werden.

Fahre Lok alle Stop .

Alle Lokomotiven werden sofort angehalten.

Wertebereich für Geschwindigkeit

Die Geschwindigkeit wird als Wert zwischen 0 und 1000 angegeben. Bei 0 steht die Lok, bei 1000 fährt sie mit Höchstgeschwindigkeit.

Gleichspannungsanlage

Bei positiven Werten fährt die Lok vorwärts, bei negativen Werten rückwärts.

Wechselspannungsanlage

Es sind nur positive Werte erlaubt. Um die Fahrrichtung zu ändern muss obiger Befehl verwendet werden.

Beispiele für Gleichspannungsanlage:

Fahre Lok 1 mit 1000.

Fahre Lok 3 mit -500.

Fahre Lok 6 mit 0.

Fahre Lok 7 Stop.

Fahre Lok 4,5 mit 800.

Beispiele für Wechselspannungsanlage:

Fahre Lok 1 mit 1000.

Fahre Lok 3 andere Richtung.

Fahre Lok 3 mit 500.

Fahre Lok 6 mit 0.

Fahre Lok alle Stop.

Lokfunktionen schalten

Fahre Lok <Lok Nr> Funktion <Funktion Nr> ein .

Fahre Lok <Lok Nr> Funktion <Funktion Nr> aus .

Mit diesem Befehl wird eine Lokfunktion ein- oder ausgeschaltet.

Beispiel:

Fahre Lok 3 Funktion 1 ein.

Fahre Lok 2 Funktion 4 aus.

Warten auf Bedingung

Ein Makro kann mit einem der folgenden Befehle auf eine Bedingung warten. Das Makro wird so lange nicht fortgesetzt, bis die Bedingung eingetroffen ist.

Warte Schaltkontakt <Nummer> .

Wartet bis der Schaltkontakt mit der angegebenen Nummer betätigt wird.

Warte Weiche <Nummer> gerade .

Warte Weiche <Nummer> rund .

Wartet, bis die Weiche mit der angegebenen Nummer gerade oder schräg gestellt ist.

Warte Signal <Nummer> Halt .

Warte Signal <Nummer> Fahrt .

Wartet, bis das Signal mit der angegebenen Nummer auf Halt oder auf Fahrt gestellt ist.

Warte Eingang <Nummer> ein .

Warte Eingang <Nummer> aus .

Warte Ausgang <Nummer> ein .

Warte Ausgang <Nummer> aus .

Wartet, bis der Eingang oder Ausgang mit der angegebenen Nummer ein- oder ausgeschaltet ist.

Warte Zeit <Sekunden> .

Das Makro wird für die angegebene Zeit in Sekunden nicht fortgesetzt.

Warte Flagge .

Warte nicht Flagge .

Das Makro wartet, bis die Flagge gesetzt oder gelöscht wird. Dadurch kann der Bediener in den Makroableuf eingreifen.

Beispiele:

Warte Schaltkontakt 85.

Warte Weiche 7 gerade.

Warte Zeit 20.

Bedingte Ausführung

Abhängig von diversen Zuständen können Sie bestimmte Makrobefehle ausführen oder nicht.

**Wenn <Bedingung>
<Befehl> .**

oder in negierter Form

**Wenn nicht <Bedingung>
<Befehl> .**

Zusätzlich zum bedingten Befehl kann noch ein alternativer Befehl angegeben werden:

**Wenn <Bedingung>
<Befehl>
Sonst
<Befehl> .**

Folgende Bedingungen können verwendet werden:

Weiche <Nummer> gerade

Weiche <Nummer> rund

Signal <Nummer> Fahrt

Signal <Nummer> Halt

Ausgang <Nummer> ein

Ausgang <Nummer> aus

Eingang <Nummer> ein

Eingang <Nummer> aus

Flagge

<Ausdruck>

Die Bedingung eines Ausdrucks ist erfüllt, wenn der Wert des Ausdrucks ungleich 0 (Null) ist.

Als Befehl können Sie jeden beliebigen Makrobefehl verwenden. Es darf aber immer nur ein Befehl sein (Punkt nicht vergessen). Wenn Sie mehrere Befehle abhängig von einer Bedingung ausführen möchten müssen Sie einen Sprung verwenden.

Über die Flagge kann der Bediener die Makroausführung beeinflussen.

Mehrere Bedingungen können logisch verknüpft werden:

ab V3.0

**Wenn <Bedingung> und <Bedingung>
<Befehl> .**

**Wenn <Bedingung> oder <Bedingung>
<Befehl> .**

Es können beliebig viele Bedingungen verknüpft werden. UND und ODER können beliebig gemischt werden, dabei werden die Bedingungen von links nach rechts ausgewertet.

Beispiele:

Wenn Weiche 1 gerade
 Stelle Weiche 2 gerade.
Sonst
 Stelle Weiche 2 rund.

Wenn nicht Signal 7 Fahrt
 GeheZu Sprung.
Stelle Weiche 5 gerade.
Stelle Weiche 8 gerade.
:Sprung

Wenn Flagge Makro Ende.

@schleife = 3.

:Runde

Warte Schaltkontakt 7.

{3 Runden fahren}

@schleife = @schleife - 1.

Wenn @schleife GeheZu Runde.

{entspricht: Wenn @schleife <> 0
GeheZu Runde}

Wenn Eingang 17 ein und Weiche 3 rund
 GeheZu Sprung.

Wenn Eingang 17 ein und Eingang 18 ein oder Flagge
 GeheZu Sprung.

Sprünge

Mit dem Befehl

GeheZu <Marke> .

wird das Makro an einer anderen Stelle fortgesetzt.

Dazu muss eine Marke vor den Befehl gesetzt werden, zu dem gesprungen werden soll.

:<Marke>

Eine Marke wird durch einen Doppelpunkt gekennzeichnet.

Der Doppelpunkt muss das erste Zeichen der Marke sein. Die Marke darf keine Leerzeichen enthalten.

Beispiel:

:Schleife

Stelle Weiche 8 gerade.

Warte Schaltkontakt 3.

Stelle Weiche 8 rund.

Warte Schaltkontakt 17.

GeheZu Schleife.

Dieses Makro stellt die Weiche 8 immer dann um, wenn der zugehörige Schaltkontakt überfahren wurde, so dass der Zug immer abwechselnd zwei unterschiedliche Strecken durchfährt.

Sprungverteiler

ab V3.0

Ein Sprung kann auch auf unterschiedliche Ziele verteilt werden. Dazu wird der Sprungmarke eine Nummer angehängt.

GeheZu <Marke> # <Ausdruck> .

Die Marke wird durch den Zahlenwert des Ausdrucks ergänzt. Dadurch ergeben sich variable Sprungziele.

Existiert keine passende Marke, dann wird kein Sprung ausgeführt.

:<Marke><Nummer>

An den verschiedenen Sprungzielen werden entsprechende Marken verwendet, die je nach Ausdruck angesprungen werden.

Beispiel:

GeheZu Gleis # @GleisNummer.

:Gleis8

Stelle Weiche 15 gerade.

GeheZu Ende.

:Gleis9

Stelle Weiche 15 rund.
Stelle Weiche 16 gerade.
GeheZu Ende.

:Gleis10

Stelle Weiche 15 rund.
Stelle Weiche 16 rund.

:Ende

Dieses Makro schaltet die Weichen passend für das Bahnhofsgleis, das in der Variable **@GleisNummer** vorgegeben wird.

Ist die Variable ungleich 8, 9 oder 10 so existiert kein passendes Sprungziel und es wird kein Sprung ausgeführt. In diesem Fall werden die Weichen für Gleis 8 gestellt, da dieser Anweisungsblock direkt auf den Sprungverteiler folgt.

Klänge und Geräusche

Diese Befehle ermöglichen das Abspielen von Sounddateien.

Spiele Klang "<Klangdatei>".

Spiele Geraeusch "<Klangdatei>".

Eine Klangdatei wird auf der Soundkarte wiedergegeben.

Die beiden Befehle spielen jeweils die angegebene Datei ab. Bei einem Klang wird die Datei nur einmal abgespielt. Bei einem Geräusch wird die Klanguausgabe solange wiederholt, bis das Geräusch gestoppt wird.

Spiele Klang Stop .

Spiele Geraeusch Stop .

Beendet die Klang- bzw. Geräuschausgabe.

Sobald ein neuer Klang oder Geräusch abgespielt wird, wird der Alte automatisch gestoppt. Es kann immer nur ein Klang oder Geräusch gespielt werden.

Klangdateien

Es können alle Dateien, die im WAV-Format vorliegen wiedergegeben werden. Wenn sich die Datei im Sound-Verzeichnis von Digibahn befindet, reicht es wenn nur der Dateiname angegeben wird (auch ohne .WAV). Es kann aber auch ein Pfad mit angegeben werden, dadurch können auch andere Dateien gespielt werden.

Vorhandene Klänge und Geräusche

Pfeife1, Pfeife2	Lokpfeifen
Horn1	Lok Horn
SchBus1	Schienenbus
Schrank1	Bahnschranken
Pfiff1	Schaffner Abfahrtspfiff
Dampf1	Dampflokomotivengeräusch

Hinweis:

Wenn Sie eigene Klangdateien in das Sound-Verzeichnis kopieren, können diese Klänge am einfachsten abgespielt werden.

Beispiele:

Spiele Klang "Pfeife1".

Spiele Klang "d:\eigene\dateien\sounds\klingel.wav".

Das Geräusch der Bahnschranke soll zwischen zwei Schaltkontakten ausgegeben werden:

Warte Schaltkontakt 1.

Spiele Geraeusch "Schrank1".

Warte Schaltkontakt 2.

Spiele Geraeusch Stop.

Makrosteuerung

Ein anderes Makro aufrufen

Es wird ein Makro aufgerufen, das eine Teilaufgabe ausführt.

Makro Aufruf "<Makroname>" .

Das Makro Makroname wird als Unterprogramm aufgerufen. Der nächste Befehl wird erst ausgeführt, nachdem das aufgerufene Makro beendet (komplett abgearbeitet) wurde.

Hinweis:

Mehrere Makros können gleichzeitig dasselbe Makro aufrufen. Ein Makro sollte sich jedoch nicht selbst aufrufen.

Beispiel:

Makro Aufruf "Fahren!Kreis 1 stellen".

Das Makro „Kreis 1 stellen“ aus der Gruppe „Fahren“ wird aufgerufen

Ein zusätzliches Makro starten

Es wird ein weiteres Makro ausgeführt.

Makro Start "<Makroname>" .

Das Makro Makroname wird gestartet. Das gestartete Makro läuft unabhängig vom aktuellen Makro ab. Die beiden Makro laufen parallel, also gleichzeitig ab.

Hinweis:

Ein Makro, das bereits läuft kann nicht nochmals gestartet werden.

Wenn Sie Ein Makro nochmals von Anfang an starten möchten und das Makro eventuell noch läuft sollten Sie es zuvor abbrechen (siehe unten).

Beispiel:

Makro Start "Automatik!zweiter Zug fahren".

Das Makro zweiter Zug fahren aus der Gruppe Automatik wird gestartet.

Ein Makro beenden

Makro Ende .

Das Makro wird an dieser Stelle beendet.

Ein Makro abbrechen

Sie können in einem Makro andere Makros abbrechen.

Makro Ende "<Makroname>" .

Das Makro Makroname wird abgebrochen.

Beispiel:**Makro Ende "Steuerung!Automatik Betrieb".**

Das Makro „Automatik Betrieb“ aus der Gruppe „Steuerung“ wird abgebrochen.

Information zur Angabe von Makronamen

Wenn sich das Makro in der gleichen Gruppe wie das aktuelle Makro befindet muss nur der Makrotitel angegeben werden.

Wenn sich das Makro in einer anderen Gruppe befindet muss der Gruppenname, durch ein Ausrufezeichen getrennt, vor dem Makrotitel angegeben werden.

Der Makroname muss in Anführungszeichen eingeschlossen werden. Im Makronamen dürfen keine überflüssigen Leerzeichen enthalten sein.

Beispiel: "Gruppe!MakroTitel"

Variablen und Ausdrücke

Variablen verwenden

Eine Variable ist ein veränderbarer Wert, der anstelle einer Zahlenangabe verwendet werden kann.

Eine Variable hat einen Namen, zur Kennzeichnung wird dem Namen eine Kennzeichnung vorangestellt. Über diese Kennung werden vier verschiedene Variablentypen unterschieden:

Kennung	Variablentyp
@	lokale Variable, nur in diesem Makro gültig.
~	globale Variable, in allen Makros gültig bis der Makromanager beendet wird.
\$	globale Variable, ist auch beim Neustart des Makromanagers noch vorhanden.
&	Referenz auf eine andere Variable. Wird für indirekte Zugriffe verwendet. (ab V3.0)

Für Schleifen im Makro werden lokale Variablen verwendet, so kann der gleiche Variablenname in mehreren Makros unabhängig voneinander verwendet werden.

Globale Variablen werden verwendet, wenn deren Wert in mehreren Makros benötigt wird.

Wert einer Variablen zuweisen

Solange einer Variablen kein Wert zugewiesen wurde ist deren Wert 0 (Null).

Um den Wert zu ändern muss ein Wert zugewiesen werden:

<Variable> = <Ausdruck> .

Der Variable wird ein Wert zugewiesen. Dieser Wert kann eine feste Wertangabe (Literal) oder ein Ausdruck (Berechnung) sein. Es können auch Variablen verwendet werden.

Beispiele:

\$LokInKreis1 = 7.

@Schleife = @Schleife + 1.

Über die Zuweisung können auch Benutzerabfragen durchgeführt und Lok-Positionen oder Zufallszahlen ermittelt werden.

Ausdrücke verwenden

Der Makromanager kann Berechnungen durchführen. Überall, wo eine Wertangabe erforderlich ist, kann ein Ausdruck berechnet werden.

Ein Ausdruck besteht aus mehreren Wertangaben oder Variablen, die durch Rechenoperatoren verknüpft sind.

Für die Berechnungen dürfen ganze Zahlen zwischen -32768 und +32767 verwendet werden.

Rechenoperatoren

Folgenden Rechenoperatoren können verwendet werden. Sie werden in der angegebenen Reihenfolge ausgewertet (Klammern werden immer zuerst berechnet, Punkt- vor Strichrechnung wird beachtet):

()	Klammern
- +	Vorzeichen
* /	Multiplikation, Division
+ -	Addition, Subtraktion
= <> > < >= <=	Vergleiche

Hinweis:

Ein Vergleich liefert auch einen Wert zurück, das Ergebnis ist 1, wenn der Vergleich zutrifft und 0, wenn der Vergleich nicht zutrifft. Das Vergleichsergebnis kann deshalb bei Bedarf in einer Variablen gespeichert werden.

Referenzen verwenden**ab V3.0**

Mit einer Referenz kann indirekt auf eine Variable zugegriffen werden. Der Referenz muss eine Variable zugewiesen werden:

<Variable> = "<Variable>" .

Der Referenz wird der Name einer Variablen zugewiesen. Der Variablenname steht in Anführungszeichen. Es können alle vier Variablentypen verwendet werden.

Die Referenz kann wie jede andere Variable verwendet werden. Dabei wird nicht der Wert der Referenz verwendet, sondern der der zugewiesenen Variable.

Beispiel:

\$LokInKreis1 = 7.

&Indirekt = "\$LokInKreis1".

Fahre Lok &Indirekt mit 500.

&Indirekt = 10.

{Fährt mit Lok 7}

{\$LokInKreis1 wird 10 zugewiesen}

Bedienereingaben

Über die Zuweisung von Variablen kann ein Makro Eingaben vom Benutzer abfragen.

Zahlenwert eingeben

<Variable> = Eingabe "<Hinweistext>" [<Vorgabewert> [< Abbruchwert>]] .

Der Variablen wird ein Wert zugewiesen, den der Benutzer eingeben kann.

Der Hinweistext wird dem Benutzer zur Information angezeigt, damit er weiß, was er eingeben soll.

Es kann zusätzlich ein Vorgabewert vorgegeben werden. Wird kein Vorgabewert angegeben, so wird 1 als Standardwert eingetragen.

Wenn ein zusätzlicher Abbruchwert angegeben wird, kann der Benutzer die Eingabe durch eine Abbruch-Taste abbrechen. In diesem Fall wird der Abbruchwert zurückgegeben.

Beispiel:

@Runden = Eingabe "Wie viele Runden soll der Zug fahren?" 1 0.

Wenn @Runden = 0 Makro Ende. {0 = Abbruchwert: Makro beenden}

:Schleife

{... Befehle für eine Runde fahren hier einfügen}

@Runden = @Runden - 1.

Wenn @Runden > 0 GeheZu Schleife.

Benutzerabfrage

<Variable> = Frage "<Hinweistext>" [<Vorgabetaste>] .

Dem Benutzer kann mit den Tasten **OK** und **Abbrechen** oder **Ja** und **Nein** auf eine Aufforderung des Makros antworten. Wenn der Benutzer mit OK oder Ja antwortet, dann ist das Ergebnis 1, sonst 0.

Der Hinweistext wird dem Benutzer zur Information angezeigt, Damit er weiß, auf was er reagieren soll.

Es kann zusätzlich eine Vorgabetaste angegeben werden. Wird OK angegeben, so werden die Tasten OK und Abbrechen angezeigt (Standard). Wird Ja angegeben, so werden die Tasten Ja und Nein angezeigt.

Beispiel:

@Antwort = Frage "Soll der Zug losfahren?" Ja.

Wenn nicht @Antwort Makro Ende. {Benutzer hat mit Nein geantwortet}

{Weitere Befehle...}

Lokomotive auswählen

<Variable> = LokWahl ["<Hinweistext>"] [<Vorgabelok> [<Abbruchwert>]] .

Dem Benutzer wird eine Liste mit allen Lokomotiven angezeigt. Er kann eine Lok auswählen.

Es kann ein Hinweistext für den Benutzer zur Information angezeigt werden.

Es kann eine Vorgabelok (Loknummer) angegeben werden.

Wenn ein zusätzlicher Abbruchwert angegeben wird, kann der Benutzer die Eingabe durch eine Abbruch-Taste abbrechen. In diesem Fall wird der Abbruchwert zurückgegeben.

Beispiel:

\$LokInGleis1 = LokWahl "Welche Lok steht in Gleis 1?" \$LokInGleis1 .

Lok-Position auswerten

Über diese Funktion kann das Makro in Abhängigkeit von der Lok-Position, verschiedene Aktionen ausführen.

<Variable> = Position <Schaltkontaktliste> [Zeit <Sekunden>] .

Es wird der als nächstes überfahrene Schaltkontakt aus der Schaltkontaktliste ermittelt. Die Nummer des Schaltkontaktes wird der Variablen zugewiesen.

Die Schaltkontaktliste ist eine Aufzählung von Schaltkontaktnummern, die durch Komma getrennt sind.

Zusätzlich kann eine Zeit in Sekunden angegeben werden. Wird innerhalb dieser Zeit keiner der angegebenen Schaltkontakte betätigt, so wird 0 (Null) zurückgegeben.

Beispiele:

In Abhängigkeit von den Weichenstellungen fährt der Zug in ein anderes Gleis. Abhängig vom befahrenen Gleis soll das richtige Signal auf Fahrt geschaltet werden.

@gleis = Position 1, 2, 3.

Wenn @gleis = 1 Stelle Signal 1 Fahrt.

Wenn @gleis = 2 Stelle Signal 2 Fahrt.

Wenn @gleis = 3 Stelle Signal 3 Fahrt.

Es soll die Fahrtrichtung eines Zuges ermittelt werden. Die Lok steht zwischen den Schaltkontakten 2 und 5. Wenn Schaltkontakt 2 betätigt wird, soll die Richtung gewechselt werden. Wird Schaltkontakt 5 betätigt, darf die Lok weiterfahren. Wird innerhalb von 10 Sekunden keiner der beiden Kontakte betätigt, so soll die Lok stehen bleiben.

Fahre Lok 1 mit 500.

@kontakt = Position 2, 5 Zeit 10.

Wenn @kontakt <> 2 GeheZu RichtungOK.

Fahre Lok 1 andere Richtung.

Fahre Lok 1 mit 500.

:RichtungOK

Wenn @kontakt = 0 Fahre Lok 1 Stop.

Belegtmelder abfragen

ab V3.0

Über diese Funktion kann das Makro einfach ermitteln, ob einer von mehreren Eingängen belegt ist.

<Variable> = Melder <Eingangsliste> [Zeit <Sekunden> [ms]] .

Es wird ermittelt, ob einer der Eingänge aus der Eingangsliste ein ist. Die Nummer des ersten Eingangs, der ein ist, wird als Ergebnis geliefert. Ist kein Eingang ein, so ist das Ergebnis 0 (Null). Das Ergebnis wird der Variablen zugewiesen.

Die Eingangsliste ist eine Aufzählung von Eingangsnummern, die durch Komma getrennt sind.

Zusätzlich kann eine Wartezeit in Sekunden oder Millisekunden angegeben werden. Diese Zeit wird abgewartet wenn kein Eingang ein ist. Ein Belegtmelder, der durch schlechten Radkontakt zwischendurch kurz frei wird, wird dadurch korrekt erfasst.

Beispiel:

Ein Gleisabschnitt besteht aus insgesamt drei Belegtmeldern. Es soll ermittelt werden, ob sich dort ein Zug befindet. In diesem Fall soll das Signal (hier Ausgang 2) auf Halt gestellt werden.

@belegt = Melder 10, 11, 12.

Wenn @belegt

Stelle Ausgang 2 aus.

Lok-Geschwindigkeit ermitteln

Diese Funktion ermittelt die aktuelle Geschwindigkeit einer Lokomotive.

<Variable> = Tempo <Loknummer> .

Die aktuelle Geschwindigkeit der Lokomotive wird der Variablen zugewiesen.

Beispiele:

Die Lok soll angehalten werden, solange die Flagge gesetzt ist und danach mit der alten Geschwindigkeit weiterfahren.

Warte Flagge.

@lok = Tempo 15.

Fahre Lok 15 mit 0.

Warte nicht Flagge.

Fahre Lok 15 mit @lok

Wenn die Flagge gesetzt ist soll die Lok langsam abbremesen. Die Geschwindigkeit kann zusätzlich mit einem Lokregler beeinflusst werden.

:schleife

Warte Flagge.

@lok = Tempo 7.

Wenn @lok > 10

Fahre Lok 7 mit @Lok - 10.

sonst

Fahre Lok 7 mit 0.

Warte Zeit 2.

GeheZu schleife.

Das folgende Makro beschleunigt und bremst eine Lok abwechselnd (Die eigentliche Beschleunigung wird durch das Lokprofil vorgenommen).

:schleife Fahre Lok 1 mit 1000.

:warte1 @lok = Tempo 1.

wenn @lok < 1000 GeheZu warte1.

{bis Endgeschwindigkeit erreicht ist}

Fahre Lok 1 mit 0.

:warte2 @lok = Tempo 1.

wenn @lok > 0 GeheZu warte2.

GeheZu schleife.

Zufällige Makroabläufe

Über die Zufalls-Funktion kann die Makroausführung zufällig beeinflusst werden.

<Variable> = Zufall [<Minimalwert>] <Maximalwert> .

Es wird eine Zufallszahl ermittelt, die zwischen dem Minimalwert und dem Maximalwert (einschließlich) liegt. Die Zufallszahl wird der Variablen zugewiesen.

Der Minimalwert muss nicht mit angegeben werden, der Standardwert ist 1.

Beispiele:

Es soll eine von 4 Loks losfahren (das entsprechende Signal wird auf Fahrt gestellt).

@zufall = Zufall 4.

Wenn @zufall = 1 Stelle Signal 8 Fahrt.

Wenn @zufall = 2 Stelle Signal 15 Fahrt.

Wenn @zufall = 3 Stelle Signal 47 Fahrt.

Wenn @zufall = 4 Stelle Signal 11 Fahrt.

Das Makro soll eine zufällige Zeit zwischen 10 und 30 Sekunden warten.

@zeit = Zufall 10 30.

Warte Zeit @zeit.

Flexible Makros mit Variablen

Verwendung von Variablen

Wozu Variablen?

Mit dem Makro Manager ist es möglich aus dem Stellpult und der Loksteuerung heraus, durch einfaches klicken mit der Maus, ein Makro aufzuzeichnen. Dies ist die einfachste Art ein Makro zu erstellen. So ein Makro hat dann einen einfachen fest strukturierten Ablauf.

Um ein solch einfaches Makro zu erweitern kann man den Inhalt als Text bearbeiten und somit die Befehle beliebig erweitern.

Auch ein Makro, das direkt als Text eingegeben wurde, ist noch recht starr. Dies lässt sich mit Variablen ändern.

Nehmen wir an, wir haben ein Makro geschrieben, das eine Lok, die im Gleis 1 im Bahnhof steht, eine Runde fahren lässt und die Lok dann wieder im Bahnhof anhält.

Im Makro werden Fahrbefehle verwendet, die folgendermaßen aussehen:

Fahre Lok 18 mit 700.

Das Makro verwendet immer die Lok mit der Adresse 18. Das ist recht unflexibel, da ja im Gleis 1 auch andere Loks stehen können, die ebenfalls diese Strecke fahren sollen. Für jede Lok ein neues Makro zu schreiben ist sicher die aufwändigste Lösung für dieses Problem. Wir brauchen also eine Möglichkeit das Makro flexibler zu machen, damit es mit allen Loks funktioniert.

Anstelle der fest vorgegebenen Lokadresse 18 benötigen wir einen Platzhalter, dem wir je nach Bedarf eine andere Lokadresse zuweisen können.

Was sind Variablen?

An dieser Stelle kommen die Variablen ins Spiel. Eine Variable ist ein Platzhalter, der einem Zahlenwert entspricht.

Eine Variable hat einen Namen, dem ein Zahlenwert zugeordnet wird. Der Zahlenwert einer Variable kann beliebig oft verändert werden. Anstelle eine Zahl direkt in das Makro zu schreiben, kann man nun den Namen der Variablen verwenden. Diese ersetzt dadurch den festen Wert.

Wo kann ich Variablen verwenden?

Eine Variable kann überall verwendet werden, wo im Makro Zahlen stehen. Man kann so z.B. die Lokadresse durch eine Variable ersetzen. Es kann aber genauso gut die Geschwindigkeit variabel gemacht werden. Natürlich gibt es noch viele andere Stellen an denen eine Zahl durch eine Variable ersetzt werden kann.

Wie kann ich eine Variable verwenden?

Eine Variable ist durch eines der Zeichen @, ~, \$ oder & gekennzeichnet. An dieses Zeichen wird der Name der Variablen angehängt. Man kann sich einen beliebigen Name ausdenken. Der Name darf aus beliebigen Buchstaben und auch Ziffern bestehen. (Bitte keine Umlaute verwenden.)

Das Beispiel zeigt wie die feste Adresse 18 mit einer Variablen ersetzt wird. Die Variable hat den Namen @LokAdresse.

Fahre Lok @LokAdresse mit 700.

Welchen Wert hat die Variable?

Wir haben nun die Adresse der Lok durch eine Variable ersetzt. Aber welche Lok fährt nun?

Das Makro wird so noch nicht funktionieren, da wir der Variablen noch nicht gezielt gesagt haben, welchen Wert sie hat. Das Makro weiß also noch nicht welche Lok fahren soll.

Grundsätzlich können beliebige Variablenamen verwendet werden. Solange wir einer Variable, keinen Wert zuweisen ist deren Wert immer 0.

Wie ändere ich den Wert einer Variablen?

Der Wert einer Variablen wird geändert indem man ihr einen Wert zuweist.

@Lokadresse = 18.

Eine solche Zuweisung kann beliebig oft erfolgen. Man kann also den Wert beliebig ändern.

Da in der Zuweisung ein Zahlenwert (im Beispiel 18) steht, kann man diesen natürlich auch wieder durch eine Variable ersetzen.

\$Gleis1 = 40. {Im Gleis 1 steht die Lok mit Adr. 40}

@Lokadresse = \$Gleis1.

Eine andere Möglichkeit ist es Berechnungen durchzuführen:

@Geschwindigkeit = 700 + 100.

~Rundenzahl = ~Rundenzahl + 1.

Der Makromanager bietet auch die Möglichkeit dass der Benutzer die Werte während das Makro läuft eingeben kann:

@Runden = Eingabe "Wie viele Runden soll der Zug fahren?".

Was bedeuten die Zeichen vor dem Variablennamen?

Im vorigen Beispiel stehen vor dem eigentlichen Variablennamen die Sonderzeichen @, ~, \$ und &. Diese Zeichen legen unterschiedliche Variablentypen fest. Der Typ der Variable sagt aus, wie der Makro Manager die Variable verwendet.

Lokale Variablen @

Variablen die mit dem Klammeraffen (@) beginnen werden im gerade ausgeführten Makro selbst gespeichert. Das bedeutet, dass nur dieses Makro selbst auf die Variable zugreifen kann. Die Variable ist lokal auf das Makro beschränkt. Sobald das Makro beendet wird werden alle zugehörigen lokalen Variablen gelöscht.

Lokale Variablen sind immer dann sinnvoll, wenn die darin gespeicherten Werte nur innerhalb dieses einen Makros benötigt werden. Dies ist z.B. bei der Lokadresse sinnvoll. Verwenden mehrere Makros die Variable @Lokadresse, so arbeitet jedes Makro mit einer anderen Variable, die jeweils lokal im Makro verfügbar ist.

Globale Variablen ~ und \$

Globale Variablen werden Makroübergreifend gespeichert. Daher kann jedes Makro auf die selbe globale Variable zugreifen. Da diese Variablen nicht an ein Makro gebunden sind bleiben sie auch nach der Beendigung des Makros erhalten und können von anderen Makros weiter verwendet werden.

Es werden zwei Varianten von globalen Makros unterschieden:

- Wird eine Variable mit der Tilde (~) gekennzeichnet, handelt es sich um eine Variable die nur so lange zur Verfügung steht, wie der Makro Manager ausgeführt wird. Wird der Makro Manager beendet und neu gestartet sind diese Variablen alle gelöscht.
- Im Gegensatz dazu werden Variablen mit dem Dollarzeichen (\$) dauerhaft gespeichert. Um diese Variablen wieder loszuwerden muss deren Wert auf 0 gesetzt werden.

Globale Variablen sind immer dann sinnvoll, wenn die darin gespeicherten Werte für mehrere Makros interessant sind.

Soll z.B. die Adresse der Lok in Gleis 1 gespeichert werden macht es Sinn die Variable \$Gleis1 zu verwenden. Diese Information wird sicherlich von mehreren Makros benötigt. Außerdem steht die Lok, die bei Programmende in Gleis 1 steht, beim nächsten Programmstart sicherlich auch noch dort.

Sollen die Runden gezählt werden, die eine Lok fährt, ist sicherlich eine nicht speichernde Variable ~Rundenzahl ausreichend.

Referenzen &

Referenzen sind spezielle Variablen, mit denen indirekt auf andere Variablen zugegriffen werden kann. Sie sind Variablen, die mit anderen Variablen arbeiten. Referenzen gehören zu den lokalen Variablen.

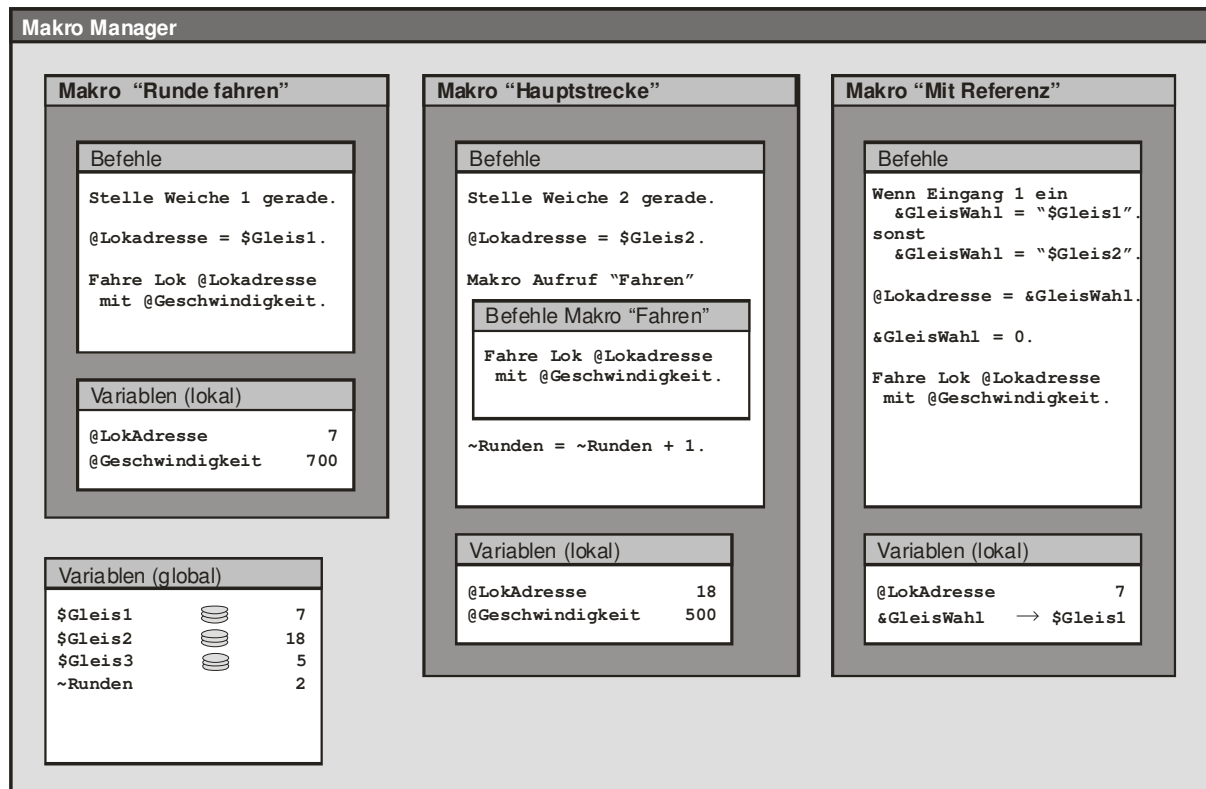
Da Referenzen nur für spezielle Zwecke verwendet werden, werde ich hier nicht näher darauf eingehen.

Welche Variable soll ich verwenden?

Wenn man eine Variable im Makro verwendet, sollte man sich zuerst immer Gedanken darüber machen, wie die Variable verwendet wird und wo sie benötigt wird.

Die Verwendung ist also immer von der Aufgabe des gespeicherten Wertes abhängig. Die nachfolgende Zeichnung verdeutlicht nochmals diese Zusammenhänge.

Einsatzbereiche von Variablen



Diese Zeichnung zeigt wie drei unterschiedliche Makros mit Variablen arbeiten. Die Makros selbst wurden lediglich verkürzt dargestellt um die Funktion der Variablen zu verdeutlichen.

Das Makro „Runde fahren“ holt sich die Information der Lokadresse aus der globalen Variable `$Gleis1` und speichert sie in `@Lokadresse`.

Ebenso geht auch das Makro „Hauptstrecke“ vor. Da `@Lokadresse` eine lokale Variable ist, ist zwar der Name der selbe, jedoch besitzt jedes Makro seine eigenen Variable `@Lokadresse`, die unterschiedliche Werte enthalten.

Das Makro „Hauptstrecke“ ruft das Untermakro „Fahren“ auf. Im Beispiel ist zu erkennen, dass dieses Untermakro keine eigenen lokalen Variablen enthält, sondern mit den lokalen Variablen des Makros „Hauptstrecke“ arbeitet. Dadurch ist es einfach möglich Werte in Untermakros zu übergeben und von dort Ergebnisse weiter zu verwenden.

Das Makro „Mit Referenz“ verwendet eine Referenz. Im Beispiel wird in Abhängigkeit von Eingang 1 eine andere Variable ausgewählt. Der Wert der ausgewählten Variable wird als `@Lokadresse` übernommen und anschließend die Lokadresse aus der ausgewählten Variable gelöscht.

Tipps zur Fehlersuche

Sie werden bei der Programmierung von Makros vermutlich immer wieder feststellen, dass das Makro nicht das tut, was Sie von ihm erwarten. Auf den ersten Blick ist das natürlich frustrierend, in vielen Fällen handelt es sich aber um Kleinigkeiten, die sich leicht beheben lassen.

Es gibt viele unterschiedliche Fehlerursachen. Auf den folgenden Seiten möchte ich zeigen, wie der Makro Manager auf Fehler reagiert und mit welchen Strategien Sie Fehler erkennen und beseitigen können.


Syntaxfehler

Vorraussetzung für den korrekten Ablauf eines Makros ist die korrekte Schreibweise (Syntax) der Makrobefehle. Bei der Aufzeichnung von Makros mit Hilfe des Stellpults oder der Loksteuerung können keine Syntaxfehler auftreten, da diese Programme die korrekten Befehle in das Makro schreiben. Sobald Sie jedoch ein Makro von Hand schreiben können sich leicht Tippfehler einschleichen.

Hilfe des Makro Managers

Der Makro Manager bietet im Hilfesystem zu jeden Befehl eine Beschreibung an. Diese lässt sich leicht direkt aus dem Mako Editorfenster aufrufen: Stellen Sie die Einfügemarke (Cursor) auf den Befehl zu dem Sie Hilfe benötigen und drücken die Tasten Strg + F1. Der Makro Manager öffnet dann eine passende Hilfeseite.

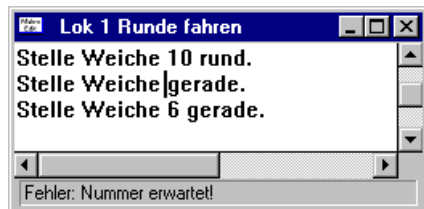
Makro wird mit Fehler abgebrochen

Kann der Makro Manager einen Befehl, während der Ausführung eines Makros, nicht interpretieren wird das Makro abgebrochen. Es wird eine Fehlermeldung angezeigt und des Makro wird im Gruppenfenster mit dem Symbol  angezeigt.

Doppelklicken Sie auf das Makro. Es wird in diesem Fall nicht gestartet, sondern es wird das Editorfenster geöffnet. Der Cursor wird an der Stelle im Makro positioniert, an der der Fehler aufgetreten ist. Zusätzlich wird im Editor eine Statuszeile angezeigt die eine genaue Fehlerursache liefert.

Beispiel:

In einem Makro wurde ein Befehl falsch eingegeben.



Beim Öffnen des Editorfensters steht der Cursor nach dem Wort "Weiche". Es wird folgender Fehler angezeigt: „Nummer erwartet!“. Wir erkennen aus der Cursorposition dass der Makro Manager den Befehl in der zweiten Zeile bis „Stelle Weiche“ verstanden hat. Die Fehlermeldung sagt aus, dass er als nächstes eine Nummer erwartet. In diesem Fall ist es die Nummer der Weiche.

Stelle Weiche **8** gerade.

Wir ergänzen den Makrobefehl und das Makro läuft beim nächsten Aufruf korrekt ab.

Tipp:

Erstellen Sie ein eigenes Makro und probieren Sie mit diesem die verschiedenen Fehlermöglichkeiten aus, indem Sie absichtlich Fehler einbauen. So sehen Sie am besten, wie der Makro Manager auf die verschiedenen Fehler reagiert.

Punkt am Befehlsende

Ein häufiger Fehler ist der Punkt am Ende eines Befehls. Der Punkt wird gerne einmal vergessen, es gibt aber auch Situationen an denen kein Punkt hinter einem Befehl stehen darf. Da der Punkt ein sehr kleines Zeichen ist wird ein derartiger Fehler auch leicht übersehen.

Solange der Makro Manager das Makro wegen eines Fehlers abbricht, wird man schnell auf den Fehler aufmerksam gemacht. Es können sich daraus aber auch Fehler ergeben bei denen die Schreibweise erlaubt ist und das Makro dann einfach nicht korrekt funktioniert.

Beispiel:

Wenn Eingang 1 ein
Stelle Weiche 3 rund .
richtig

Wenn Eingang 1 ein .
Stelle Weiche 3 rund .
falsch

Bei Bedingungen wird meist der bedingt ausgeführte Befehl in die nächste Zeile geschrieben und leicht eingerückt. Dadurch entsteht leicht der Eindruck, dass in der ersten Zeile der Punkt fehlt. In diesem Fall hat ein fehlerhafter Punkt aber ungeahnte Auswirkungen:

Laut Befehlsbeschreibung folgt nach der Bedingung „Wenn Eingang 1 ein“ ein Befehl, der nur ausgeführt wird, wenn die Bedingung zutrifft. Da ein Befehl immer mit einem Punkt endet, ist mit dem nachfolgenden Punkt der Befehl sofort beendet. Für das Makro ist das dann ein leerer Befehl. Dieser leere Befehl wird nun bedingt ausgeführt. Der Befehl „Stelle Weiche 3 rund“ ist nun schon der nächste Befehl nach dem Leerbefehl, der immer ausgeführt wird. Die Weiche 3 wird also in diesem Beispiel immer geschaltet.

Logische Fehler

Während Syntaxfehler meist vom Makro Manager erkannt werden, erhalten Sie bei logischen Fehlern keine Fehlermeldung. Bei logischen Fehlern ist die Schreibweise korrekt, so dass der Makro Manager alle Befehle abarbeiten kann. Die Abarbeitung der Befehle führt aber nicht zu dem gewünschten Ergebnis.

Logische Fehler entstehen meist aus Denkfehlern bei denen man sich bestimmte Abläufe anders vorgestellt hat, als diese eigentlich sind. Bei umfangreichen Makros schleichen sich häufig solche Fehler ein. Dies passiert auch erfahrenen Programmierern und ist kein Beinbruch. Mit etwas Geduld und den richtigen Strategien lassen sich solche Fehler aufspüren und beseitigen.

Hinweis:

Man sollte nicht vernachlässigen, dass es auch zu Ablaufproblemen durch fehlerhafte Bauteile auf der Anlage kommen kann. Eine Weiche die nicht schaltet, oder ein Belegtmelder der nicht reagiert, führen zu Problemen bei denen das Makro selbst nicht die Ursache ist.

Befehle auskommentieren

Wenn ein Makro unkontrolliert abläuft hat dies oft zur Folge, dass auch die Loks unkontrolliert weiterfahren. Es ist daher sinnvoll immer die passenden Lokregler in der Loksteuerung parat zu haben, um die Loks zu stoppen. Dies geht aber bei längerer Fehlersuche gehörig auf die Nerven, manchmal ist es auch schwierig daraus resultierende Zusammenstöße rechtzeitig zu verhindern.

Um das Makro dennoch laufen zu lassen kann man die Fahrbefehle auskommentieren.

Beispiel:

Stelle Weiche 8 gerade.
{ Fahre Lok 3 mit 900. }
Warte Eingang 7 ein. {Belegtmelder}
Stelle Weiche 8 rund.

Die Lok wird nun nicht vom Makro gesteuert, sondern kann von Hand mit dem Lokregler auf den Belegtmelder gefahren werden. Ob der Belegtmelder reagiert kann man sehen, wenn die Weiche rund geschaltet wird.

Makro abbrechen.

Wenn sich ein Makro verrannt hat (Sie nicht mehr wissen, an welcher Stelle das Makro momentan arbeitet), oder es sich in einer Endlosschleife befindet, kann das Makro abgebrochen werden.

Wenn Sie nach einem Abbruch das Editorfenster öffnen, befindet sich der Cursor an der Position, an der das Makro abgebrochen wurde. So wissen Sie, was das Makro zum Zeitpunkt des Abbruchs gerade gemacht hat.

Absichtlich Fehler einbauen.

Es kann durchaus Sinn machen absichtlich einen Fehler in ein Programm einzubauen. Dies ist besonders dann interessant, wenn in einem Makro oft hin und her gesprungen wird. Dann kann es durchaus sein, dass Sie wissen

wollen, ob Befehle an einer bestimmten Stelle überhaupt ausgeführt werden. Sobald der fehlerhafte Befehl erreicht wird bricht das Makro ab und sie wissen, dass er jetzt ausgeführt wurde.

Beispiel:

```
:sprung7  
@LokAdresse = $Gleis1. X  
GeheZu Sprung3.
```

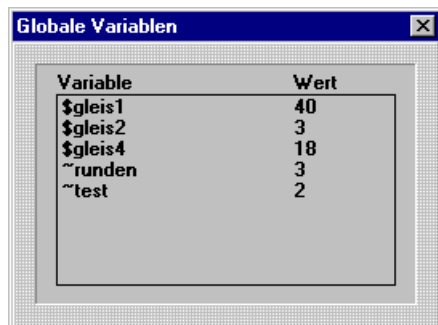
Sie möchten wissen, ob bzw. wann die LokAdresse zugewiesen wird. Das X hinter dem Befehl sorgt dafür dass das Makro an dieser Stelle fehlerhaft ist und abgebrochen wird.

Zugegeben, diese Methode ist recht brachial. In vielen Fällen kann man aber damit schnell der Fehlerursache auf den Grund gehen. Die folgenden Methoden sind eleganter, aber auch aufwendiger.

Globale Variablen anschauen und ändern

Mit dem Makro Manager haben Sie die Möglichkeit den Wert aller globalen Variablen anzuzeigen.

Wählen Sie im Menü „Spezial/Variablen“ um das Variablenfenster zu öffnen. In diesem Fenster werden alle globalen Variablen mit aktuellem Wert angezeigt.



Variable	Wert
\$gleis1	40
\$gleis2	3
\$gleis4	18
~runden	3
~test	2

Hinweis:

Variablen die den Wert 0 haben werden gelöscht und deshalb nicht angezeigt.

Der Wert der Variablen kann geändert werden indem man doppelt auf den Variablennamen klickt.

Variablenwerte anzeigen

Mit den Bediennereingabe-Befehlen „Eingabe“ und „Lokwahl“ kann man nicht nur Werte vom Bediener abfragen, man kann sich auch aktuelle Werte anzeigen lassen. Dies ist im normalen Betrieb möglich und kann natürlich auch zur Fehlersuche eingesetzt werden:

```
@dummy = Eingabe "Lokadresse:" @Lokadresse.
```

Die gewünschte Variable wird als Vorgabewert verwendet. Da der eingegebene Wert in diesem Fall uninteressant ist wird er einer Pseudovariablen zugewiesen.

In diesem Fall kann man auch den Befehl Lokwahl verwenden. Statt der Lokadresse wird dann der Name der Lok angezeigt.

Makro Arbeitsweise verfolgen

Bei umfangreicheren Makros kann man schon mal den Überblick verlieren und weiß nicht welche Makroteile abgearbeitet werden und welche nicht. In so einem Fall kann man sich mit einer Hilfsvariablen anzeigen lassen, in welchem Teil das Makro gerade arbeitet. Man verwendet dazu eine globale Variable, die man sich im Variablenfenster anzeigen lassen kann.

Beispiel 1:

```
{Teil 1}  
~Test = 1.
```

...

...

```
{Teil 2}  
~Test = 2.  
...
```

```
...  
{Teil 3}  
~Test = 3.  
...
```

In allen wichtigen Makroteilen (die für die Fehlersuche interessant sind) wird die Test-Variable auf einen eindeutigen Wert gesetzt. Anhand des Wertes der Variablen im Variablenfenster ist sofort klar welcher Makroteil gerade bearbeitet wird.

Beispiel 2:

```
:schleife  
~Test = ~Test + 1.  
...  
GeheZu schleife.
```

Läuft das Makro in einer Schleife, die schnell abgearbeitet wird, kann man eine Variable hochzählen um die Schleifendurchläufe zu ermitteln:

Beispielmakros

Einfache Beispiele

Diese Beispiele demonstrieren verschiedene einfache Aufgaben, die mit Makros realisiert werden können.

Lok 1 Runde fahren

Dieses Makro zeigt, wie ein Zug eine Runde in Gleis 3 fährt und dann automatisch im Bahnhof anhält.

(Die Funktion kann im Testtreiber simuliert werden indem die Kontakte 5 und 6 simuliert werden.)

Ein genaues Anhalten im Bahnhof wird am besten mit zwei Schaltkontakten (oder Gleisbesetzmeldern) erreicht:

In jedem Bahnhofsgleis ist am Anfang und am Ende ein solcher Kontakt, der die Zugposition meldet. Mit dem ersten Kontakt (hier K5) wird der Zug auf "Schleichfahrt" abgebremst. Am zweiten Kontakt (K6) wird er gestoppt.

Ist ein passendes Profil in der Loksteuerung eingestellt dann hält der Zug vorbildgetreu an. Die Geschwindigkeit "Schleichfahrt" sollte so gewählt werden, dass der Zug noch ganz langsam fährt.

{Weichen stellen}

Stelle Weiche 7 gerade.

Stelle Weiche 8 rund.

Stelle Weiche 10 gerade.

Stelle Weiche 6 gerade.

{Signal auf grün}

Stelle Ausgang 53 ein.

{Lok losfahren}

Fahre Lok 1 mit 750.

{Etwas warten und dann Signal wieder auf rot stellen}

Warte Zeit 2.

Stelle Ausgang 53 aus.

{Warten bis lok wieder in den Bahnhof kommt}

Warte Schaltkontakt 5.

{Lok mit Schleichgeschw. fahren}

Fahre Lok 1 mit 130.

{Bis sie am Gleisende ankommt}

Warte Schaltkontakt 6.

{Dann sofort anhalten}

Fahre Lok 1 Stop.

Auf Bedingung warten

Dieses Makro zeigt, wie ein Zug in Gleis 3 fährt und abhängig von einem Signal automatisch im Bahnhof anhält.

Der Zustand des Signals (Ausgang 53) kann im Makro abgefragt werden. Das Makro kann darauf reagieren und dadurch den Zug anhalten.

Stelle Weiche 7 gerade. {Weichen stellen}

Stelle Weiche 8 rund.

Stelle Weiche 10 gerade.

Stelle Weiche 6 gerade.

Stelle Ausgang 53 ein. {Signal auf grün}

Fahre Lok 1 mit 750. {Lok losfahren}

{Warten bis das Signal auf rot gestellt wurde}
Warte Ausgang 53 aus.

Warte Schaltkontakt 5. {Warten bis Lok wieder in den Bahnhof kommt}
Fahre Lok 1 mit 130. {Lok mit Schleichgeschw. fahren}
Warte Schaltkontakt 6. {Bis sie am Gleisende ankommt}
Fahre Lok 1 Stop. {Dann sofort anhalten}

Makrobeeinflussung mit Flagge

Dieses Makro zeigt, wie ein Zug in Gleis 3 fährt und auf Wunsch automatisch im Bahnhof anhält.

Ein laufendes Makro kann vom Benutzer beeinflusst werden indem er doppelt auf das laufende Makro klickt. Dann wird auf dem Makrosymbol eine Flagge angezeigt. Durch nochmaliges doppelklicken verschwindet die Flagge wieder.

Der Zustand dieser Flagge kann im Makro abgefragt werden. Das Makro kann darauf reagieren und dadurch unterschiedliche Reaktionen auslösen.}

Stelle Weiche 7 gerade. {Weichen stellen}
Stelle Weiche 8 rund.
Stelle Weiche 10 gerade.
Stelle Weiche 6 gerade.
Stelle Ausgang 53 ein. {Signal auf grün}
Fahre Lok 1 mit 750. {Lok losfahren}

{Solange die Flagge nicht gesetzt ist passiert nichts:
Der Zug dreht seine Runden, da das Makro wartet.
Wird die Flagge gesetzt, läuft das Makro weiter und hält den Zug an.}
Warte Flagge.

Stelle Ausgang 53 aus. {Signal auf rot stellen}

Warte Schaltkontakt 5. {Warten bis Lok wieder in den Bahnhof kommt}
Fahre Lok 1 mit 130. {Lok mit Schleichgeschw. fahren}
Warte Schaltkontakt 6. {Bis sie am Gleisende ankommt}
Fahre Lok 1 Stop. {Dann sofort anhalten}

Bedingte Ausführung 1

Dieses Makro zeigt, wie abhängig von der Einfahrweiche für die Gleise 3 und 4 die Ausfahrweiche gestellt wird

Dieses Makro kann gleichzeitig mit anderen Makros ausgeführt werden (z.B. dem Makro "Lok 1 Runde fahren")

Das Makro wird in einer Endlosschleife ausgeführt. Um die Ausführung zu beenden muss es abgebrochen werden. Dazu mit der rechten Maustaste auf das Makro klicken und dann "Abbrechen" wählen.

{Sprungmarke für die Schleife}
:schleife

{Stellung von Weiche 8 abfragen abhängig davon wird Weiche 7 geschaltet}
Wenn Weiche 8 gerade
Stelle weiche 7 rund.
sonst
Stelle Weiche 7 gerade.

{Kurz warten}
Warte Zeit 1.

{Sprung zum Anfang}
GeheZu schleife.

Bedingte Ausführung 2

Dieses Makro zeigt, wie abhängig von der Einfahrweiche für die Gleise 3 und 4 die Ausfahrweiche gestellt wird

Das Makro "Bedingte Ausführung 1" ist für diese Zwecke nur bedingt geeignet, da die Weiche 7 jede Sekunde neu gestellt wird. Um dies zu umgehen wird zusätzlich die Stellung der Weiche 7 abgefragt. Nur wenn die Weiche falsch steht wird sie gestellt.

Das Makro wird in einer Endlosschleife ausgeführt. Um die Ausführung zu beenden muß es abgebrochen werden. Dazu mit der rechten Maustaste auf das Makro klicken und dann "Abbrechen" wählen.}

:schleife {Sprungmarke für die Schleife}

{Stellung von Weiche 8 abfragen
wenn sie gerade ist wird ein Sprung ausgeführt}
Wenn Weiche 8 gerade
GeheZu w8gerade.

{Dieser Teil wird ausgeführt, wenn Weiche 8 rund ist
Es wird die Stellung von Weiche 7 geprüft
Hier ist zusätzlich eine Abfrage auf NICHT eingebaut}
Wenn nicht Weiche 7 gerade
Stelle Weiche 7 gerade.

{Die alternative Bearbeitung überspringen}
GeheZu W8ende.

{Sprungziel für alternativen Bearbeitung}
:W8gerade

{Stellung von Weiche 7 prüfen}
Wenn nicht Weiche 7 rund
Stelle Weiche 7 rund.

{Sprungziel Ende der Bearbeitung}
:W8ende

{Kurz warten}
Warte Zeit 1.

{Sprung zum Anfang}
GeheZu schleife.

Runden zählen mit Variable

Dieses Makro läßt den Zug in Gleis 3 fünf Runden lang fahren und ihn dann im Bahnhof anhalten. Die Runden werden in einer Variablen gezählt.

Normalerweise werden für solche Aufgaben lokale Variablen (@) verwendet. Damit die Variable unter "Spezial/Variablen" beobachtet werden kann, wird in diesem Beispiel mit einer globalen Variable (~) gearbeitet.

Die Runden werden mit Schaltkontakt 5 gezählt.

Stelle Weiche 7 gerade. {Weichen stellen}
Stelle Weiche 8 rund.
Stelle Weiche 10 gerade.

Stelle Weiche 6 gerade.
 Stelle Ausgang 53 ein. {Signal auf grün}
 Fahre Lok 1 mit 750. {Lok losfahren}

{Variable initialisieren, dies wäre bei einer lokalen Variable nicht
 notwendig, da alle nicht vorhandenen Variablen den Wert 0 haben}
 ~runden = 0.

{Schleife zur Rundenzählung}
 :schleife

{Warten bis Lok wieder in den Bahnhof kommt}
 Warte Schaltkontakt 5.

{Kurz warten, damit die Lok den Kontakt nicht mehr berührt,
 sonst kann es zu fehlerhaftem zählen kommen
 (Zur Simulation ist die Zeit hier nur eine Sekunde)}
 Warte Zeit 1.

{Rundenzahl erhöhen}
 ~runden = ~runden + 1.

{Wenn noch keine 4 Runden zurückgelegt sind wird die Schleife
 wiederholt}
 Wenn ~runden < 4
 GeheZu schleife.

{Signal auf rot stellen}
 Stelle Ausgang 53 aus.

{Warten bis Lok das fünfte mal in den Bahnhof kommt
 um sie dann anzuhalten}
 Warte Schaltkontakt 5.

Fahre Lok 1 mit 130. {Lok mit Schleichgeschw. fahren}
 Warte Schaltkontakt 6. {Bis sie am Gleisende ankommt}
 Fahre Lok 1 Stop. {Dann sofort anhalten}

Makro aufrufen

Dieses Makro entspricht in der Funktion dem Makro "Lok 1 Runde fahren".

Statt die Weichen direkt im Makro zu stellen wird ein Fahrstraßenmakro aufgerufen.

{Es wird das Makro "Gleis3" aus der Gruppe "Fahrstraßen" aufgerufen.}
 Makro Aufruf "Fahrstraßen!Gleis 3".

Stelle Ausgang 53 ein. {Signal auf grün}
 Fahre Lok 1 mit 750. {Lok losfahren}

Warte Zeit 2. {Etwas warten und dann Signal wieder auf rot stellen}
 Stelle Ausgang 53 aus.

Warte Schaltkontakt 5. {Warten bis Lok wieder in den Bahnhof kommt}
 Fahre Lok 1 mit 130. {Lok mit Schleichgeschw. fahren}
 Warte Schaltkontakt 6. {Bis sie am Gleisende ankommt}
 Fahre Lok 1 Stop. {Dann sofort anhalten}

Rundenzahl abfragen

Dieses Makro läßt den Zug in Gleis 3 eine beliebige Anzahl von Runden fahren und ihn dann im Bahnhof anhalten. Die Anzahl der Runden wird vom Bediener abgefragt.

Normalerweise werden für solche Aufgaben lokale Variablen (@) verwendet. Damit die Variable unter "Spezial/Variablen" beobachtet werden kann, wird in diesem Beispiel mit einer globalen Variable (~) gearbeitet.

Die Runden werden mit Schaltkontakt 5 gezählt. Mit dem Befehl "Eingabe" kann ein Zahlenwert vom Bediener abgefragt werden. Es erscheint ein Fenster auf dem Bildschirm, um einen Wert einzugeben. Im Fenster wird ein Hinweistext angezeigt, der Informationen über die gewünschte Eingabe gibt. Die "3" ist der Wert, der dem Bediener vorgeschlagen wird.

~runden = Eingabe "Wie viele Runden soll der Zug fahren?" 3.

Stelle Weiche 7 gerade. {Weichen stellen}
 Stelle Weiche 8 rund.
 Stelle Weiche 10 gerade.
 Stelle Weiche 6 gerade.
 Stelle Ausgang 53 ein. {Signal auf grün}
 Fahre Lok 1 mit 750. {Lok losfahren}

{Schleife zur Rundenzählung}
 :schleife

{Warten bis Lok wieder in den Bahnhof kommt}
 Warte Schaltkontakt 5.

{Rest-Rundenzahl verringern}
 ~runden = ~runden - 1.

Warte Zeit 1. {Kurz warten}

{Wenn noch nicht genug Runden zurückgelegt sind wird die Schleife wiederholt}
 Wenn ~runden > 1
 GeheZu schleife.

{Signal auf rot stellen}
 Stelle Ausgang 53 aus.

{Warten bis Lok das letzte mal in den Bahnhof kommt um sie dann anzuhalten}
 Warte Schaltkontakt 5.

Fahre Lok 1 mit 130. {Lok mit Schleichgeschw. fahren}
 Warte Schaltkontakt 6. {Bis sie am Gleisende ankommt}
 Fahre Lok 1 Stop. {Dann sofort anhalten}

Kontakte oder Belegmelder

Es gibt zwei grundsätzliche Möglichkeiten zur Positionsbestimmung von Lokomotiven bzw. Zügen:

1. Schaltkontakte

Diese sind normalerweise mit Schaltgleisen oder Reedkontakten realisiert. Dieser wird durch die Lok ausgelöst, wenn sie darüberfährt. Dadurch entsteht ein kurzer Schaltimpuls. Dieser kann im Makro mit dem Befehl

Warte Schaltkontakt 5.

abgefragt werden. Da der Impuls nur kurz ansteht, kann nicht ermittelt werden ob der Zug noch dort steht.

2. Belegtmelder

Belegtmelder ermitteln über die Räder oder über Stromfühler ob eine Lok bzw. Zug auf dem Gleis steht. Belegtmelder können wie Schaltkontakte mit dem "Warte" Befehl abgefragt werden:

Warte Eingang 5 ein.

Zusätzlich kann der Status des Belegtmelders abgefragt werden um zu ermitteln, ob der Zug dort steht:

Wenn Eingang 5 ein
Stelle Ausgang 53 aus.

Fazit:

Es ist vom Anlagenaufbau abhängig, welche Möglichkeit eingesetzt wird. Belegtmelder sind vielseitiger als Schaltkontakte.

Im diesen Beispielmakros sind Schaltkontakte verwendet, da diese besser über den Testtreiber simuliert werden können.

Lok auswählen

Dieses Makro zeigt, wie ein Zug ausgewählt werden kann.

Die Lok wird vom Benutzer abgefragt und in einer Variable gespeichert. Diese Variable wird als Loknummer in allen Fahrbefehlen verwendet.

Die Loknummer wird in der Variable \$LokGleis3 gespeichert. Somit weiß Digibahn welche Lok im Gleis 3 steht.

Beim nächsten Makroaufruf steht die Lok wahrscheinlich auch noch dort. Daher kann der Variablenwert als Vorgabe für die Lokwahl verwendet werden.

Diese Vorgehensweise kann auch in den anderen Gleisen verwendet werden.

\$LokGleis3 = LokWahl "Welche Lok steht im Gleis 3?" \$LokGleis3.

Stelle Weiche 7 gerade. {Weichen stellen}

Stelle Weiche 8 rund.

Stelle Weiche 10 gerade.

Stelle Weiche 6 gerade.

Stelle Ausgang 53 ein. {Signal auf grün}

{Ausgewähle Lok fahren}

Fahre Lok \$LokGleis3 mit 750.

Stelle Ausgang 53 aus. {Signal auf rot stellen}

Warte Schaltkontakt 5. {Warten bis Lok wieder in den Bahnhof kommt}

{Ausgewähle Lok fahren}

Fahre Lok \$LokGleis3 mit 130.

Warte Schaltkontakt 6. {Bis sie am Gleisende ankommt}

{Ausgewähle Lok fahren}

Fahre Lok \$LokGleis3 Stop.

Lokgeschwindigkeit ermitteln

Dieses Makro ermittelt die Geschwindigkeit der Lok 1. Wenn eine Geschwindigkeit von über 900 (90%) erreicht ist wird die Lok wieder bis zum stehen abgebremst.

{Lok losfahren - wird bis 1000 beschleunigt}

Fahre Lok 1 mit 1000.

:schleife

{Geschwindigkeit ermitteln und in Variable @speed speichern}

@speed = Tempo 1.

{Solange die Geschwindigkeit kleiner 900 läuft die Schleife}

Wenn @speed < 900

GeheZu schleife.

{Lok wieder abbremesen}

Fahre Lok 1 mit 0.

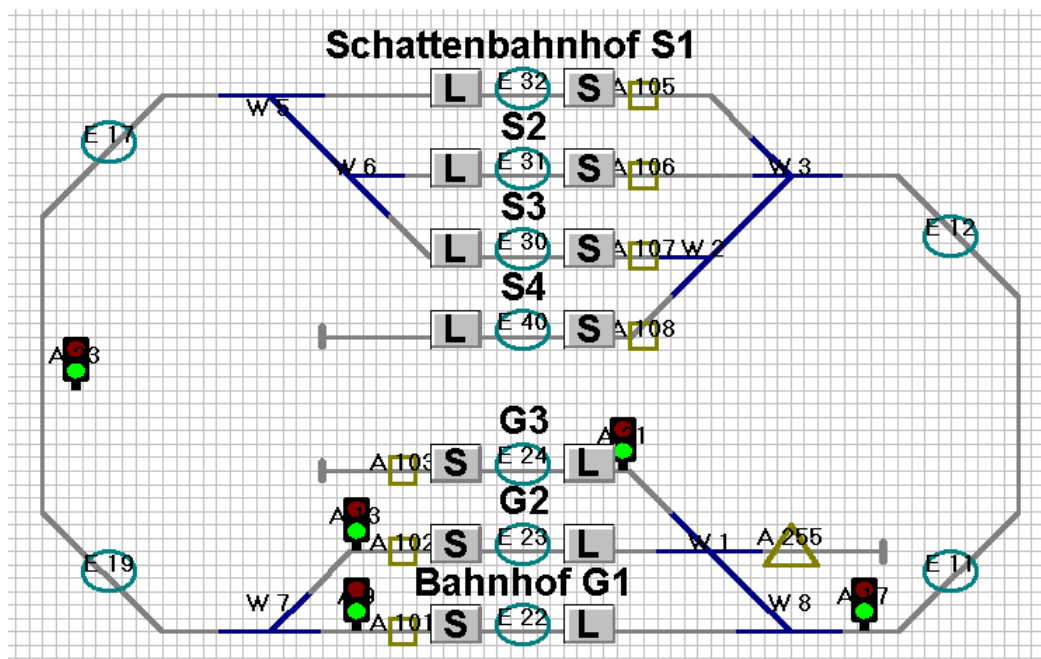
Beispielanlage

Dieses Beispiel zeigt die Makroprogrammierung anhand der kleinen Ausstellungsanlage des AMMS Stuttgart. Auf dieser Anlage wurde ein vollständiger Automatikbetrieb realisiert.

Aufbau der Anlage



Auf dieser technischen Vorführanlage sind verschiedenste Komponenten in Einsatz.



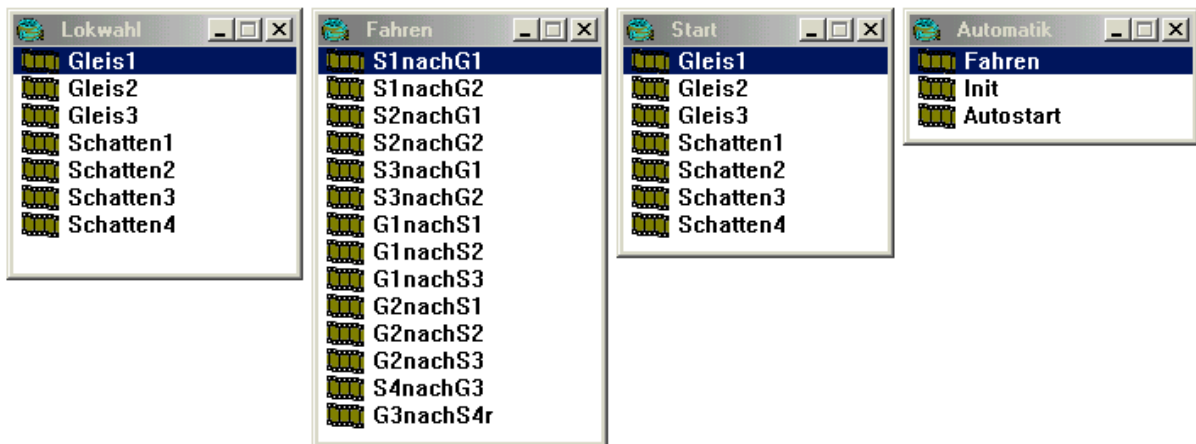
Gleisplan der Anlage

Dieses Bild zeigt den Gleisplan der Anlage. Es sind die Adressen der Weichen und Belegtmelder zu erkennen.

Deutlich sind auch die Aktionstasten mit der Aufschrift „L“ und „S“ zu erkennen.

Die Signale sind im Plan ebenfalls eingezeichnet, jedoch werden sie im Automatikbetrieb noch nicht berücksichtigt.

Die Makrogruppen



Die vier Makrogruppen

Die Makros dieser Anlage sind in mehrere Gruppen aufgeteilt. Dadurch kann die Anlage sowohl manuell, halbautomatisch sowie in vollautomatisch betrieben werden.

Die Gruppe „Lokwahl“

Zu Betriebsbeginn muss Digibahn mitgeteilt werden, welche Gleise mit welchen Loks belegt sind. Für jedes Gleis ist ein entsprechendes Makro vorhanden.

Das Makro „Lokwahl!Gleis1“

\$Gleis1 = Lokwahl "Lok in Gleis 1" \$Gleis1 0.

Das Makro fragt mit dem Befehl „Lokwahl“ ab, welche Lok auf dem Gleis steht. Die Adresse wird in der dem Gleis zugehörigen Variable „\$Gleis1“ gespeichert.

Diese Variable wird auch als Vorgabewert übergeben. Dadurch wird die bisher dort stehende Lok vorgeschlagen. Durch Angabe des Abbruchwertes „0“ hat man die Möglichkeit die Lokwahl mit der Taste „Abbruch“ abzubrechen. In diesem Fall wird 0 in „\$Gleis1“ gespeichert, was bedeutet, dass sich dort keine Lok befindet.

Die im Gleisplan eingetragenen Aktionstasten mit der Beschriftung „L“ rufen diese Lokwahlmakros auf. So kann direkt über den Gleisplan die Lok gewählt werden.

Die Gruppe „Fahren“

Mit dieser Gruppe wird die erste Stufe der Automatik bereitgestellt.

Die Anlage ermöglicht es vom Bahnhof (von den Gleisen G1 bis G3) in den Schattenbahnhof (Gleise S1 bis S4) zu fahren. Ebenso kann vom Schattenbahnhof in den Bahnhof gefahren werden.

Für jede mögliche Zugfahrt wurde ein Makro erstellt, das diese Fahrt ausführt. Dabei wird überwacht ob das Abfahrgleis belegt, das Zielgleis und die Strecke frei sind. Nur wenn die Fahrt möglich ist, wird sie auch ausgeführt.

Alle Makros sind identisch aufgebaut. Es werden lediglich andere Weichen gestellt, andere Variablen verwendet und die passenden Belegtmelder abgefragt.

Das Makro „Fahren!S1nachG1“

Wenn nicht \$Schatten1 {keine Lok im Abfahrgleis, dann Ende}
Makro Ende.

Wenn \$Gleis1 {Zielgleis belegt über Variable}
Makro Ende.

Wenn Eingang 22 ein Makro Ende.	{Zielgleis belegt vom Melder}
~SnachG = 1.	{Diese Variable sperrt diese Strecke}
Stelle Dreiwegweiche 3 rechts.	{Bahnhof Ausfahrt Weichen Stellen}
Warte Eingang 12 aus. Warte Eingang 11 aus.	{Warten bis die Stecke frei ist}
Stelle Weiche 8 gerade.	{Weichen im Zielbahnhof stellen}
@Lok = \$Schatten1. \$Schatten1 = 0.	{Adresse der zu fahrenden Lok} {Abfahrtsgleis ist jetzt frei}
Fahre Lok @Lok mit 800.	{Jetzt wird gefahren}
Warte Eingang 22 ein. Fahre Lok @Lok mit 0.	{Zielbahnhof erreicht} {Lok anhalten}
\$Gleis1 = @Lok.	{Die Adresse der Lok im Bahnhofsgleis merken}
~SnachG = 0.	{Die Strecke wieder freigeben}

Hinweis:

Auf dieser Anlage ist in jedem Bahnhofsgleis nur ein Belegtmelder vorhanden. Der Bremsweg der Lokomotiven wird mit einem Profil der Loksteuerung eingestellt.

Da zudem die Bahnhofsgleise sehr kurz sind ist ein genaues Anhalten im Bahnhof mit unregelmäßigen Decodern fast unmöglich. Es muss daher immer wieder von Hand eingegriffen werden, wenn die Lok zu weit oder zu wenig fährt.

Ein zielgenaues Anhalten ist mit mehreren Belegtmeldern (2 oder 3) einfach zu realisieren. Das Makro muss dann entsprechend angepasst werden.

Die Gruppe „Start“

Um eine einfache Halbautomatik zu realisieren wurde die Makrogruppe „Start“ erstellt. Im Unterschied zur Gruppe „Fahren“ muss man hier nur die Lok wählen, die losfahren soll. Das Makro sucht ein freies Gleis im Zielbahnhof und startet dann das passende Makro aus der Gruppe „Fahren“.

Diese Makros können über die Aktionstasten „S“ (wie Start) im Gleisbild gestartet werden.

Als zusätzliche Steuerungsmöglichkeit befindet sich in jedem Bahnhofsgleis ein Ausgang mit dem die Ausfahrt des Zuges gesperrt werden kann. Anhand dieses Ausgangs kann auch beobachtet werden, welcher Zug gerade fährt.

Das Makro „Start!Gleis1“

Wenn Ausgang 101 ein Makro Ende.	{Gleis ist gesperrt, nicht fahren}
Stelle Ausgang 101 ein.	{Mit diesem Ausgang anzeigen, dass eine Zugfahrt aktiv ist}
Wenn Eingang 32 aus GeheZu Schatten1.	{Schattenbahnhof 1 ist frei}
Wenn Eingang 31 aus GeheZu Schatten2.	{Schattenbahnhof 2 ist frei}
Wenn Eingang 30 aus GeheZu Schatten3.	{Schattenbahnhof 1 ist frei}
GeheZu Ende.	{Kein Gleis frei, fahren nicht möglich}

:Schatten1 Makro Aufruf "Fahren!G1nachS1". GeheZu Ende.	{nach S1 fahren} {Makro aufrufen}
:Schatten2 Makro Aufruf "Fahren!G1nachS2". GeheZu Ende.	{nach S2 fahren} {Makro aufrufen}
:Schatten3 Wenn \$Gleis1 = 11 GeheZu Ende.	{nach S3 fahren} {Besonderheit: Der Zug mit Adresse} {11 ist zu lang für S3 und darf nicht losfahren}
Makro Aufruf "Fahren!G1nachS3". GeheZu Ende.	{Makro aufrufen}
:Ende Stelle Ausgang 101 aus.	{Zugfahrt ist beendet}

Die Gruppe „Automatik“

Diese Gruppe enthält die Makros, die für einen vollständig automatischen Betrieb zuständig sind. Der Automatikbetrieb selbst steckt im Makro „Fahren“. Die restlichen Makros vereinfachen den Betrieb der Anlage.

Das Makro „Automatik!Init“

Mit diesem Makrobetrieb wird die Anlage auf den Automatikbetrieb vorbereitet.

Dazu werden die Steckensperren aufgehoben und für jedes Bahnhofsgleis die Lok abgefragt.

~SnachG = 0.
~GnachS = 0.

Makro Aufruf "Lokwahl!Gleis1".
Makro Aufruf "Lokwahl!Gleis2".
Makro Aufruf "Lokwahl!Gleis3".
Makro Aufruf "Lokwahl!Schatten1".
Makro Aufruf "Lokwahl!Schatten2".
Makro Aufruf "Lokwahl!Schatten3".
Makro Aufruf "Lokwahl!Schatten4".

Das Makro „Automatik!Autostart“

In diesem Makro werden Aufgaben erledigt, die beim Start des Makromanagers nötig sind. Für das Beispiel sind hier keine interessanten Aufgaben vorhanden.

Das Makro „Automatik!Fahren“

Dieses Makro steuert automatisch den Fahrbetrieb. Dazu werden die Makros der Gruppe „Start“ gestartet.

Durch setzen der Flagge kann der Automatikbetrieb jederzeit beendet werden.

Das Makro prüft nur bedingt, ob die entsprechende Zugfahrt möglich ist. Die eigentliche Prüfung findet in den entsprechenden Untermakros statt.

Das Makro steuert zusätzlich einen Pendelbetrieb auf der Nebenbahn. Diese Fahrt wird durch einen Zähler der Hauptstrecke gesteuert.

Grob unterteilt besteht diese Makro aus zwei identischen Teilen. Der eine steuert die Fahrt vom Schattenbahnhof in den Bahnhof, der andere die Gegenstrecke.

:SnachG	{Teil1 Schattenbahnhof nach Bahnhof}
Wenn Flagge	{Betrieb beenden, wenn Flagge ein}
Makro Ende.	
Warte Zeit 2.	{Schleifendurchlauf etwas verzögern}
Wenn ~SnachG	{Zug in dieser Richtung ist unterwegs}
GeheZu GnachS.	
Wenn ~Zaehler < 10	{Nebenbahn fahren wenn Zähler hoch}
GeheZu keinNB.	{genug ist}
Wenn \$Gleis2	{Gleis 2 sperrt Nebenbahn}
GeheZu keinNB.	
Wenn Eingang 8 ein	{Nebenbahn zurückfahren}
GeheZu NBruock.	
Makro Start "Start!Schatten4".	{Nebenbahn vorwärts fahren}
~Zaehler = 0.	{Zähler rücksetzen}
GeheZu GnachS.	
:NBruock	
Makro Start "Start!Gleis3".	{Nebenbahn rückwärts fahren}
~Zaehler = 0.	{Zähler rücksetzen}
GeheZu GnachS.	
:keinNB .	{Hier geht normaler Betrieb weiter}
:Bfrei	{Ein Bahnhofsgleis ist frei}
@SStart = Zufall 1 3.	{Zufällig ein Gleis auswählen}
Wenn @SStart <> 1	{Nicht dieses Gleis - Schatten 1}
GeheZu SStart2.	
Wenn Eingang 32 aus	{Gleis ist nicht belegt}
GeheZu SStart2.	
Makro Start "Start!Schatten1".	{Zugfahrt starten}
~Zaehler = ~Zaehler + 1.	
GeheZu GnachS.	
:SStart2	
Wenn @SStart <> 2	{Nicht dieses Gleis - Schatten 2}
GeheZu SStart3.	
Wenn Eingang 31 aus	{Gleis ist nicht belegt}
GeheZu SStart3.	
Makro Start "Start!Schatten2".	{Zugfahrt starten}
~Zaehler = ~Zaehler + 1.	
GeheZu GnachS.	
:SStart3	
Wenn @SStart <> 3	{Nicht dieses Gleis - Schatten 3}
GeheZu SStart4.	
Wenn Eingang 30 aus	{Gleis ist nicht belegt}
GeheZu SStart4.	
Makro Start "Start!Schatten3".	{Zugfahrt starten}
~Zaehler = ~Zaehler + 1.	
GeheZu GnachS.	
:SStart4	
GeheZu GnachS.	

```
:GnachS                                {Teil2 Bahnhof nach Schattenbahnhof}
Wenn Flagge
  Makro Ende.
Warte Zeit 2.

Wenn ~GnachS                             {Zug unterwegs}
  GeheZu SnachG.

:Sfrei                                    {ein Schattenbahnhofsgleis ist frei}
@GStart = Zufall 1 2.

Wenn @GStart <> 1                         {Gleis 1}
  GeheZu GStart2.
Wenn Eingang 22 aus
  GeheZu GStart2.
Makro Start "Start!Gleis1".
GeheZu SnachG.

:GStart2.
Wenn @GStart <> 2                         {Gleis 2}
  GeheZu GStart3.
Wenn Eingang 23 aus
  GeheZu GStart3.
Makro Start "Start!Gleis2".

:GStart3
  GeheZu SnachG.
```

Anhang

Befehlsübersicht

<i>Befehl</i>	<i>Typ</i>	<i>Parameter</i>	<i>Status</i>	<i>Beschreibung</i>
Weichen und Signale stellen				
Stelle	Weiche	<Nummer>	gerade.	Stellt eine Weiche oder Kreuzungsweiche in die angegebene Richtung.
			rund.	
	Dreiwegweiche		gerade.	Stellt die beiden Antriebe einer Dreiwegweiche in die angegebene Richtung.
			links.	
			rechts.	
	Signal		Halt.	Stellt das Signal auf Halt bzw. Fahrt.
			Fahrt.	
	Ausgang		ein.	Schaltet einen Ausgang ein bzw. aus.
aus.				
Lok Fahrbefehle				
Fahre	Lok	<Nummern>	mit <Gesch>.	Gibt Geschwindigkeit für eine oder mehrere Lokomotiven vor.
			Stop.	Hält die Lokomotive sofort an.
			andere Richtung.	Schaltet sie Fahrtrichtung der Lokomotive um.
			alle	Hält alle Lokomotiven sofort an.
	Lok <Nummer>	Funktion <Nummer>	ein .	Schaltet eine Lokfunktion ein bzw. aus.
			aus .	
Bedingungen				
Wenn [Nicht]	<Bedingung> <Befehl> . [Sonst <Befehl>.]		Abhängig von der Bedingung wird der Befehl ausgeführt. Es mit Sonst kann ein alternativer Befehl ausgeführt werden.	
	[<Bedingung> UND] <Bedingung> [ODER <Bedingung>] <...> .		Mehrere Bedingungen können logisch mit UND oder ODER verknüpft werden.	

Befehl	Typ	Parameter	Status	Beschreibung
Warten auf Bedingung				
Warte	Schaltkontakt	<Nummer>	.	Wartet bis der Schaltkontakt betätigt wird.
	Weiche		gerade.	Wartet bis die Weiche oder Kreuzungsweiche in den gewünschten Zustand gestellt wird.
			rund.	
	Signal		Halt.	Wartet bis das Signal in den gewünschten Zustand gestellt wird.
			Fahrt.	
	Ausgang		ein.	Wartet bis der Ausgang in den gewünschten Zustand gestellt wird.
			aus.	
	Eingang		ein.	Wartet bis der Eingang den gewünschten Zustand annimmt.
			aus.	
	Zeit	<Sekunden> [ms].		Wartet die angegebene Zeit in Sekunden.
		[nicht] Flagge.		Wartet bis die Flagge gesetzt oder zurückgesetzt ist.
Sprünge				
GeheZu		<Marke> .		Führt einen Sprung zur angegebenen Marke durch.
		<Marke> # <Ausdruck> .		Der Marke wird der Wert des Ausdrucks angehängt. Führt Sprung zu dieser Marke aus, wenn sie existiert.
		:<Marke>		Definiert eine Marke.
Makrosteuerung				
Makro	Aufruf	"<Makroname>" .		Ruft das angegebene Makro als Unterprogramm auf.
	Start			Startet das angegebene Makro. Es wird parallel ausgeführt.
	Ende .			Beendet das Makro.
	Ende	"<Makroname>" .		Beendet das angegebene Makro.
Variablen				
<Variable> =		<Ausdruck>.		Weist einer Variablen einen Wert zu. Es gibt vier Arten von Variablen: @ lokale Variablen sind nur innerhalb eines Makros während dessen Laufzeit gültig. ~ globale Variablen sind von allen Makros erreichbar und während der Laufzeit des Makro Managers gültig. \$ global speichernde Variablen sind auch beim nächsten Programmstart des Makro Managers noch vorhanden. & eine Referenz (lokal) verweist auf den Inhalt einer Variablen. So lassen sich Variablen indirekt ansprechen.
<Referenz> =		"<Variablenname>" .		Weist einer Referenz einen Variablennamen zu.

Befehl	Typ	Parameter	Status	Beschreibung
Klänge und Geräusche				
Spiele	Klang	"<Klangdatei>" .		Spielt eine Klangdatei.
		Stop.		Bricht die Klanguausgabe ab.
	Geräusch	"<Klangdatei>" . Stop.		Ein Geräusch ist identisch mit einem Klang. Ein Geräusch wird jedoch ständig wiederholt.
Bedienereingaben				
<Variable> =	Eingabe	"<Hinweistext>" [<Vorgabewert> [<Abbruchwert>]].		Der Benutzer kann einen Wert eingeben, der einer Variablen zugewiesen wird.
	Frage	"<Hinweistext>" [<Vorgabetaste>].		Der Benutzer kann eine Frage mit Ja/Nein oder OK/Abbruch beantworten.
	LokWahl	["<Hinweistext>"] [<Vorgabelok> [<Abbruchwert>]].		Der Benutzer kann eine Lok auswählen, deren Nummer wird einer Variablen zugewiesen.
Lokdaten auswerten				
<Variable> =	Position	<Schaltkontaktliste> [Zeit <Sekunden> [ms]].		Ermittelt ob ein Eingang aus der Schaltkontaktliste betätigt ist. Die Nummer des Schaltkontaktes wird der Variablen zugewiesen.
	Melder	<Eingangliste> [Zeit <Sekunden> [ms]].		Ermittelt den nächstes überfahrenen Schaltkontakt aus der Schaltkontaktliste. Ist ein Eingang belegt, so wird dessen Nummer der Variablen zugewiesen. Mit der Wartezeit kann eine Fehlmeldung durch flackern der Eingänge unterdrückt werden.
	Tempo	<Loknummer>.		Ermittelt die Geschwindigkeit der angegebenen Lokomotive.
Zufällige Abläufe				
<Variable> =	Zufall	[<Minimalwert>] <Maximalwert>.		Es wird eine Zufallszahl erzeugt, die im angegebenen Wertebereich liegt.
Rechenoperatoren in Ausdrücken (die Operatoren sind in der Bearbeitungsreihenfolge aufgelistet)				
	()			Klammern
	- +			Vorzeichen
	* /			Multiplikation, Division
	+ -			Addition, Subtraktion
	= <>			Vergleiche
	> < >= <=			
Sonstiges				
	{<Kommentar>}			Ein Text in geschweiften Klammern wird ignoriert.