================== ERKLÄRUNG MAKROKOMMANDOS ================

Update vom 16.05.2014

Sie finden unten eine Liste mit möglichen Makrokommandos und ihre Bedeutung. Kommandos und Parameter werden durch Klammern “(“ und “)” getrennt, Parameter werden durch “,” getrennt, evtl mit Leerzeichen.

Beispiel: lok(3, 10) lässt die Lok mit Folgenummer 3 mit Fahrstufe 10 fahren.

Die Parameter sind alle 16 bit, die Gültigkeit hängt jedoch vom Parametertyp ab. Die Makros werden in Klartext in einer Datei gespeichert, und können von einen beliebigen Editor geändert werden.

switch(folgenummer, stellung)
Stellt eine Weiche oder ein Signal, Stellung hat den Wert 1 bis 99
switch(folgenummer, stellung, lock)
Stellt eine Weiche oder ein Signal, Stellung hat den Wert 1 bis 99, Weiche wird verriegelt
switch(folgenummer, stellung, unlock)
Verriegelung einer Weiche wird weggenommen (Weiche wird nicht gestellt)

Beispiele:
switch(5, 2) Stellt Weiche 5 abbiegend.
switch(12, 3) Stellt Signal 12 langsam (Hp2).
switch(3, 1, lock) Stellt Weiche 3, und verriegele sie nachher.
switch(3, 1, unlock) Die Verriegelung von Weiche 3 freigeben.

lok(folgenummer, fahrstufe)
Lässt eine Lokomotive mit bestimmter Geschwindigkeit fahren.

<folgenummer>: interne Lokadresse, die von ModellStw vergeben wurde
<fahrstufe>: Kommando für den Interface: 0 - 127: geschwindigkeit. Wenn die Lok eine Masse hat wird die Geschwindigkeit langsam erreicht.

Beispiel:
lok(3,8) lässt Lok 3 fahren mit geschwindigkeit 8

dir (folgenummer)
Ändert die Fahrtrichtung einer Lokomotive.

function(folgenummer, funktionen)
Setzt die Funktionen einer Lokomotive. <funktionen> ist die Summe der nachstehenden Werten:
1 = funktion 0 (licht) einschalten
2 = funktion 1 einschalten
4 = funktion 2 einschalten
8 = funktion 3 einschalten
16 = funktion 4 einschalten
.... usw
32768 = funktion 15 einschalten

lokspeedslow(folgenummer)

Lässt eine Lok weiterfahren mit ihre 'langsame' (30kmh) geschwindigkeit.

lokspeedsfast(folgenummer)
Lässt eine Lok weiterfahren mit ihre 'schnelle' (120kmh) geschwindigkeit.

route(folgenummer, on/off)
Stellt die Weichen einer Fahrstrasse, und legt diese fest ('on') oder gibt eine Fahrstrasse frei ('off').

Beispiele:
route(5, on) stellt die Weichen der Fahrstrasse 5 und legt die Fahrstrasse fest
route(5, off) gibt die Fahrstrasse wieder frei

contact(kontaktnummer, on)
Unterbricht die Ausführung des Makros, bis der Meldekontakt <kontaktnummer> eingeschaltet wird. Danach wird das Makro weiter ausgeführt.

Beispiel:
Das Beispiel zeigt ein Makro, das die Lok 2 langsam fahren lässt und nach vorbeifahren am Meldekontakt 5 die Lok 2 halten lässt:
01 lok(2, 4)
02 contact(5, on)
03 lok(2, 0)

contact(kontaktnummer, off)
Unterbricht die Ausführung des Makros bis der Meldekontakt ausgeschaltet wird. Danach läuft das Makro weiter.

pause(sekunden)
Unterbricht die Ausführung des Makros bis die angegebene Zeit <sekunden> abgelaufen ist. Danach läuft das Makro weiter.

Beispiel:
Dieses Makro lässt die Lok 2 anhalten, zum Beispiel am Bahnhof und nach 15 Sekunden weiterfahren:
01 lok(2, 0)
02 pause(15)
03 lok(2, 8)

jump(kontakt, makrozeile)
Lässt das Makro zu einer anderen Zeile springen, wenn der Meldekontakt <kontakt> eingeschaltet ist.

Beispiel:
jump(5, 7) lässt das Makro weiterlaufen mit Zeile 7 wenn Meldekontakt 5 eingeschaltet ist, sonst läuft es weiter mit Zeile 2.

goto(makrozeile)
Lässt der Makro zu einer anderen Zeile springen.

Beispiel:
goto(5) lässt der Makro witerlaufen mit Zeile 5.

bel(zeit)
Lässt während der aufgegebene Zeit ein 'Beep' aus den Lautsprechern des PC hören.

Das Argument <zeit> ist in 1/20-stel sekunden.

Beispiel:
bel(20) lässt ein “Beep” von einer Sekunde hören.

stop
Nothalt

ifswitch(folgenummer, stellung, makrozeile)
Testet die Stellung einer Weiche. Wenn die Stellung dem zweiten Parameter entspricht, springt das Makro zur Zeile <makrozeile>, <stellung> kann die Werte 1..99 haben.

Beispiel:
In diesem Beispiel wird Signal 3 auf Halt gestellt, wenn Weiche 2 abbiegend ist, sonst wird Signal 4 auf Halt gestellt:
01 ifswitch(2, 2, 4)
02 switch(4, 1)
03 end
04 switch(3, 1)

ifroute(folgenummer fahrstrasse, stellung, zeile)
Testet die Stellung einer Fahrstrasse. Wenn die Stellung dem zweiten Parameter entspricht, springt das Makro zur Zeile <zeile>, <stellung> kann die Werte “on” (ein) oder “off” (aus) haben.

Beispiel:
In diesem Beispiel wird Signal 3 auf Halt gestellt, wenn die Fahrstrasse 5 eingestellt ist, sonst wird Signal 4 auf Halt gestellt
01 ifroute(5, on, 4)
02 switch(4, r)
03 end
04 switch(3, r)

ifblock(blockabschnitt, makrozeile)
Lässt das Makro weiterlaufen mit Zeile <makrozeile>, wenn der Blockabschnitt besetzt ist.

iflok(blockabschnitt, loknummer, makrozeile)
Lässt das Makro weiterlaufen mit Zeile <zeile>, wenn der Blockabschnitt besetzt ist mit Lokomotive <loknummer>.

start(makronummer)
Startet das Makro mit Nummer <makronummer>. Die Nummer muss >= 1 und <= 100 sein.

block(folgenummer blockabschnitt, wert)
Meldet einen Blockabschnitt besetzt oder frei. Der Blockabschnitt wird im Gleisbild eingefärbt. Bei Freimeldung folgt keine Zugverfolgung, die Lok 'verschwindet' vom Gleisplan. <wert> = “on” (ein) oder “off” (aus)

end
Beendet das Makro

Variablen
=======

Das Makromodul des PctWin kennt 26 Programmvariablen. Diese Variablen können einen Wert von 0 bis 255 enthalten. Eine bestimmte Variable steht jedem Makro zu Verfügung, wenn in Makro 1 eine Variable einen Wert bekommt, kann diesen z.B. in Makro 5 wieder gelesen werden. Jede Variabel wird angedeutet mit einer Buchstabe A bis Z.
Die Kommandos für die Variablen finden Sie hier unten.

let(var, wert)

Gibt eine von den 26 Variablen einen wert. Als <var> wird hier eine Buchstabe gegeben, A bis Z. Der Wert soll 0 bis 255 sein.

Beispiel:
01 let(b, 10) gibt Variablen B den Wert 10

inc(var)

Erhöht den Wert der Variablen um 1.

Beispiel:
01 let(q,10) Q hat den Wert 10
02 inc(q) Q hat jetzt den Wert 11

dec(var)

Zieht 1 von dem Wert der Variablen ab.

Beispiel:
01 let(q, 10) Q hat den Wert 10
02 dec(q) Q hat jetzt den Wert 9

ifvar(var, wert, zeile)

Testet den Wert der Variablen, wenn dies den 3. Parameter entspricht, geht das Makro weiter mit Zeile <zeile>.

Beispiel:
Lässt Lok 2 10 Runden fahren und hält dann die Lok an.
01 inc(a)
02 ifvar(a,10,4)
03 end
04 lok(2,0)

ifvarg(var, wert, zeile)

Testet den Wert der Variablen, wenn dieser grösser ist als der 3. Parameter wird das Makro auf Zeile <zeile> fortgesetzt.

Kommandos für die Zugverfolgung
==========

Die nachfolgenden Kommandos können nur verwendet werden wenn Zugverfolgung aktiviert

ist.

blockspeed(blockabschnitt, fahrstufe)
Gibt der Lok in dem <blockabschnitt> die angegebene Geschwindigkeit. Die jetzige Geschwindigkeit wird behalten und kann mit dem Kommando <blockspeedprev> aufgerufen werden, um eine Lok mit der vorherigen Geschwindigkeit weiter fahren zu lassen.

blockspeedprev(blockabschnitt)
Lässt die Lok die im Blockabschnitt angehalten worden war, mit der der vorherigen Geschwindigkeit weiterfahren.
Alternativ kann für die Eiterfahrt auch die Komandos <lokspeed>, <lokspeedslow> oder <lokspeedfast> verwendet werden.

Beispiel:
Dieses Beispiel lässt jede Lok im Blockabschnitt 4 anhalten, 5 Sekunden warten und wieder weiterfahren.
lokspeed(4,0) lässt die lok in block 4 anhalten
pause(5) wartet 5 sekunden
lokspeedprev(4) lässt die Lok in Block 4 mit vorheriger Fahrstufe weiterfahren

blockspeedslow (blockabschnitt)
Lässt die Lok in dem Blockabschnitt weiter fahren mit der Geschwindigkeit die bei der Lok unter 30kmh aufgegeben ist.

blockspeedfast(blockabschnitt)
Lässt die Lok in dem Blockabschnitt weiter fahren mit der Geschwindigkeit die bei der Lok unter 120kmh aufgegeben ist.

blockmove(von, nach)
Hilft die Zugverfolgung durch eine Loknummer von Blockabschnitt <von> nach Blockabschnitt <nach> weiter zu schieben. Unter spezielle Anforderungen kann dieses Kommando behilflich sein, die Loknummer schneller in einem Blockabschnitt zu verschieben.

blockmoveauto(von, richtung)
Hilft die Zugverfolgung durch eine Loknummer von Blockabschnitt <von> nach dem nächsten Blockabschnitt in <richtung> weiter zu schieben.

========== ENDE ==========