

Projekt: DINAMO plus iTrain5-Testanlage für Spur Z

Beginn: 06.02.22

Wie alles begann:

Für meine noch in Planung befindliche Anlage (204x64cm), bin/war ich auf der Suche nach einer Möglichkeit, die 50% analogen Loks parallel zu den digitalen Loks betreiben zu können. Ganz simpel wäre: ganze Anlage per Umschalter von digital auf analog schalten und die Halteabschnitte vorm Signal per Relais als Stoppgleis auszulegen. Das gefällt mir allerdings nicht und so begab ich mich auf die Suche. Aktuell bin ich bei dem DINAMO-System. Hier kann man (angeblich auch in Z) analog und digital parallel betreiben.

Ich habe gerade heute mit einem N-Bahner gesprochen, der seine gesamte Anlage mit DINAMO betreibt. Analog und Digital parallel. Per Videoanruf hatte er mir so einiges gezeigt. Sah recht gut aus. Problem bei ihm: neuere Loks mit großen Pufferkondensatoren wollen nicht ganz so toll anfahren.

Er hat vorgeschlagen, ein paar Z-Gleise und analoge Loks mitzubringen. Die Verkabelung seiner Einmessstrecke werden wir dann benutzen, um meine analogen Loks zu testen. Werde wohl mal 10 Stück mitnehmen aus allen Generationen, besonders die neueren mit Glocki und Pufferelkos. Zum Glück wohnt er nur 40 Minuten von mir entfernt.

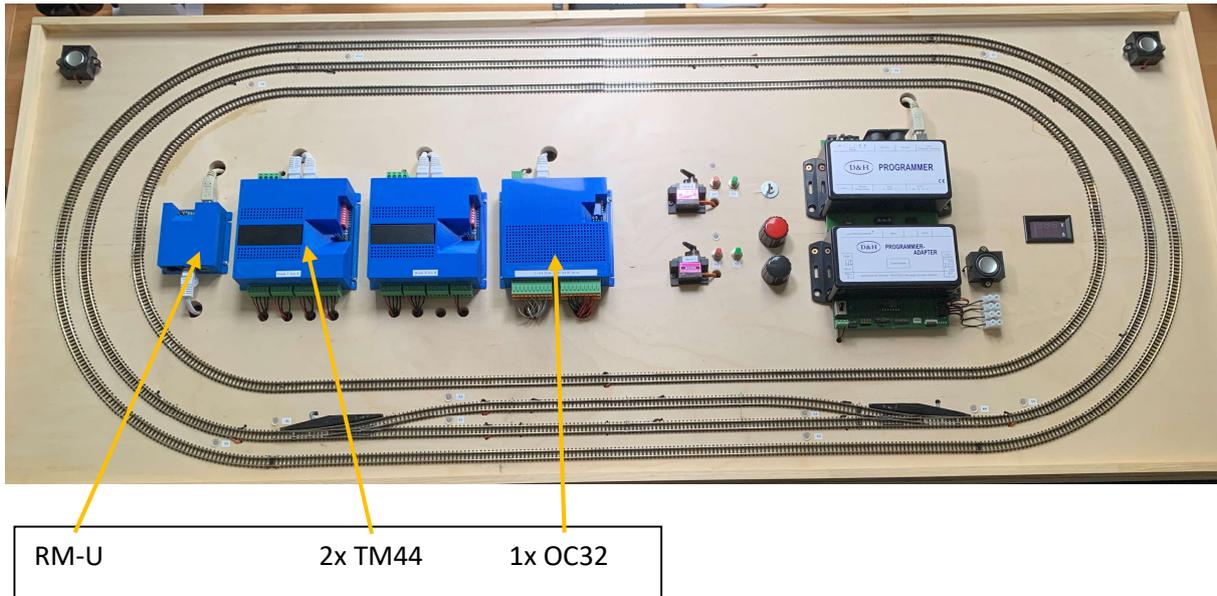
Da es eine N-Anlage ist, wird diese mit 16 Volt betrieben. Das hat fatale Auswirkungen, denn bei analogen Z-Loks liegt dann eine viel zu hohe Spannung am Gleis an. Dadurch fahren diese nicht mehr nur 250 km/h, sondern 400 km/h. Somit konnte ich dort nur testen, ob Z-Loks von Dinamo gesteuert werden können, aber keine Langsamfahrt.

Allerdings war ich begeistert von dem, was mir Markus Schneider (bei dem ich getestet hatte) auf seiner Anlage gezeigt hat. Somit stand mein Entschluss fest, eine Dinamo-Testanlage zu bauen.

Auf 1,2m x 0,5m entstand dann diese Anlage. Der Innenkreis ist ein Programmiergleis, dass mit Dinamo, D&H-Programmer, oder analog betrieben werden kann. An diese Strecke ist auch der „Bärenzähler“ (Spannungs- und Strommessgerät für analoge und digitale Gleisspannung) angeschlossen.

Der Außenkreis dient dem sehr wichtigen Einmessen von Loks und Zügen. Dabei ist die untere 660mm Gerade die Messtrecke.

Der Innenkreis mit „Bahnhof“ ist in 5 Blöcke unterteilt. Bahnhof Außen, Bahnhof Innen, Links, Oben, Rechts. Jeder Block ist mit einer unterschiedlichen Anzahl Meldeabschnitten versehen. Dazu später mehr.

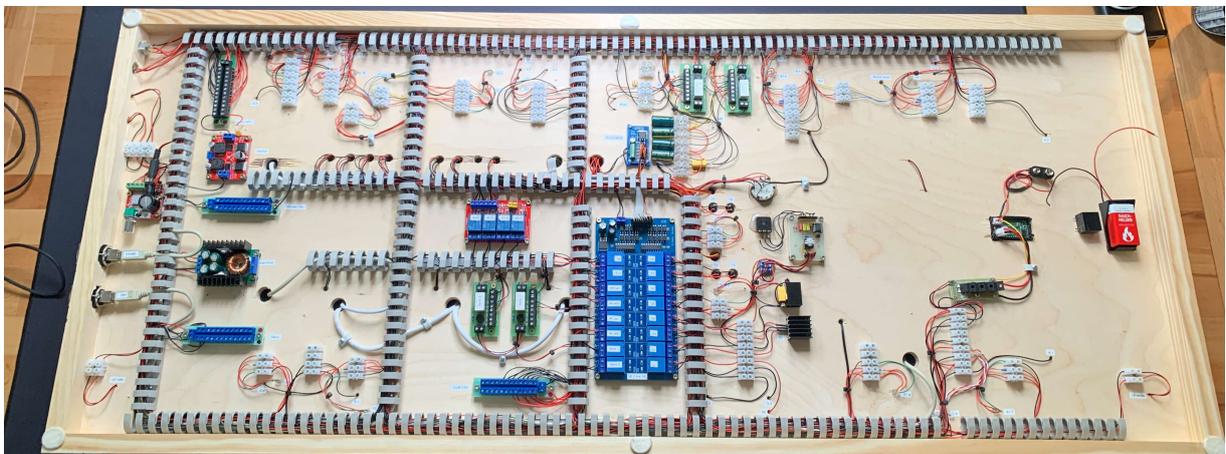


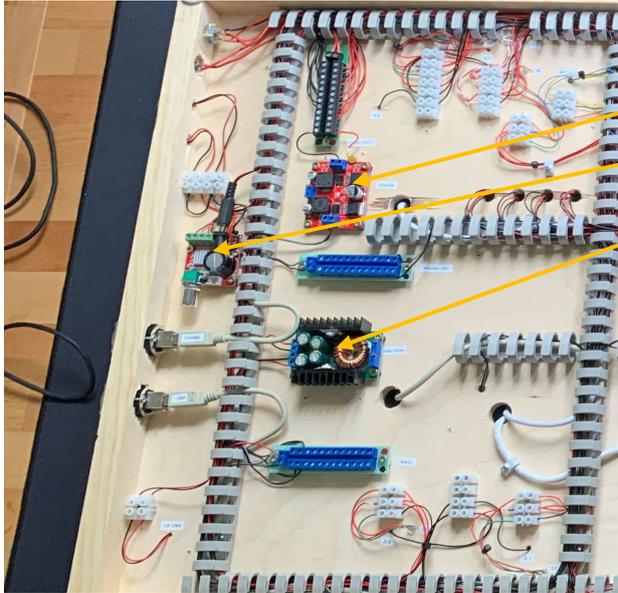
RM-U: Diese wird per USB an den PC angeschlossen und über Patchkabel mit den anderen Komponenten verbunden (typisch Ethernet)

TM44: Die Blocksteuerung mit 4 Meldeabschnitten pro Block

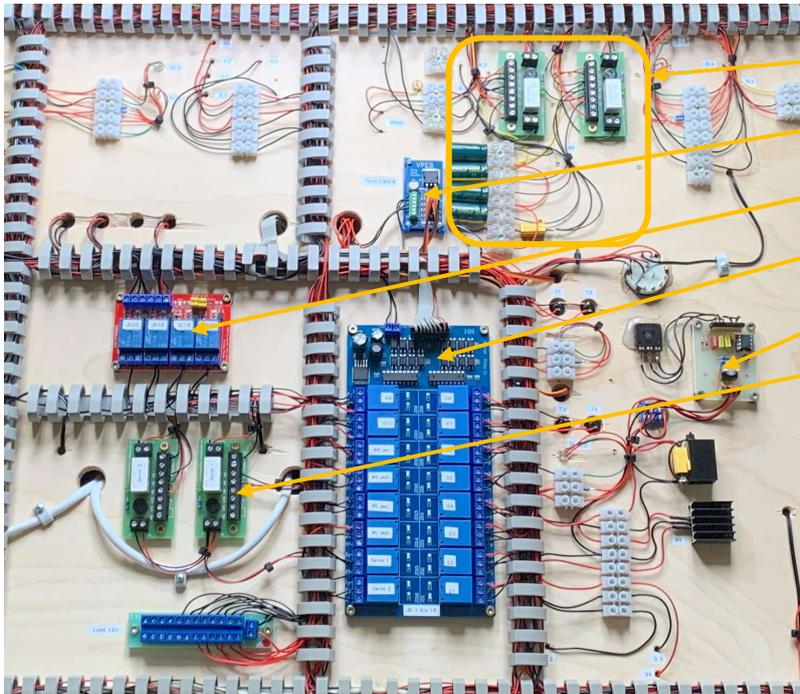
OC32: Das 32-fach Ein-Ausgabegerät für Schalter, Taster, Servo, Signale, Weichen, Licht, usw.

Der Verkabelungsaufwand, den ich hier betrieben habe, ist etwas übertrieben. Ich wollte aber auf jeden Fall viel testen können und evtl. auch noch erweitern können. Man wird ja nie fertig ;-)





Step-Up (12V auf 16V) für Weichen
Audio-Verstärker
Step-Down (ca. 10,5V) für Gleis



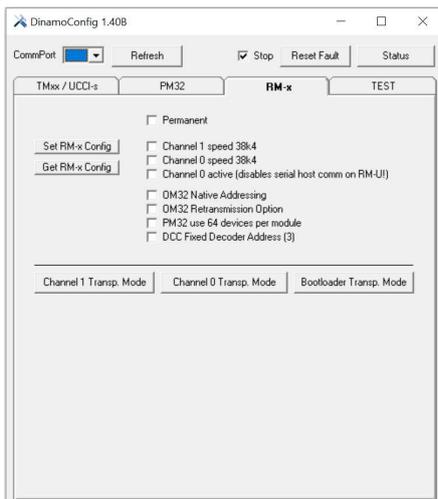
Weichenschutz mit LED-Anzeige
Servo-Ansteuerung
4-fach Relais-Karte
16-fach Relais-Karte
Analoger Regler
Test Herzstückpolarisierung



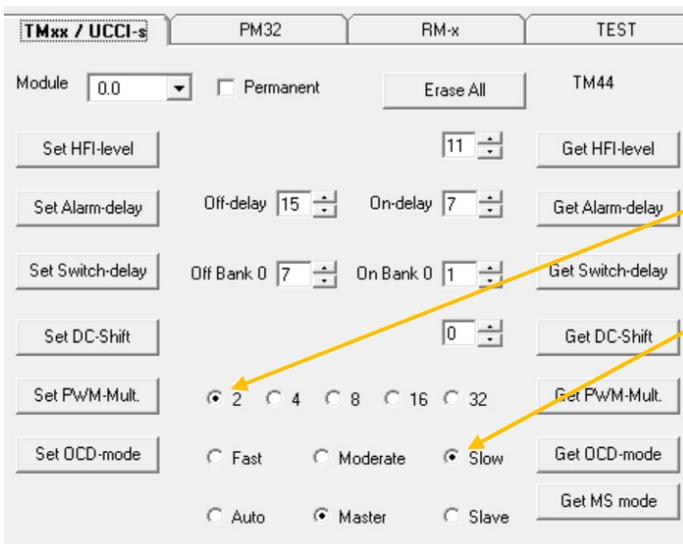
Auf der linken Seite sind die Anschlüsse. Dynamo wird, wie D&H, über ein USB-Kabel mit dem PC verbunden. Die Chinch-Buchsen sind für den Ton vom PC kommend (2 Lautsprecher sind auf der Anlage verbaut mit kleinem Verstärker-Modul).

Damit war dann die Testanlage fertig aufgebaut.

Ich möchte hier nicht das Handbuch für die Dinamo-Komponenten abtippen, aber einen Überblick geben.



Wenn alle Komponenten richtig angeschlossen sind und die DIP-Schalter richtig gesetzt sind, beginnt man nun mit der Basis-Konfiguration mit dieser Software. Wenn die Verbindung steht, dann müssen hierüber die RM-U und die TM44 konfiguriert werden.

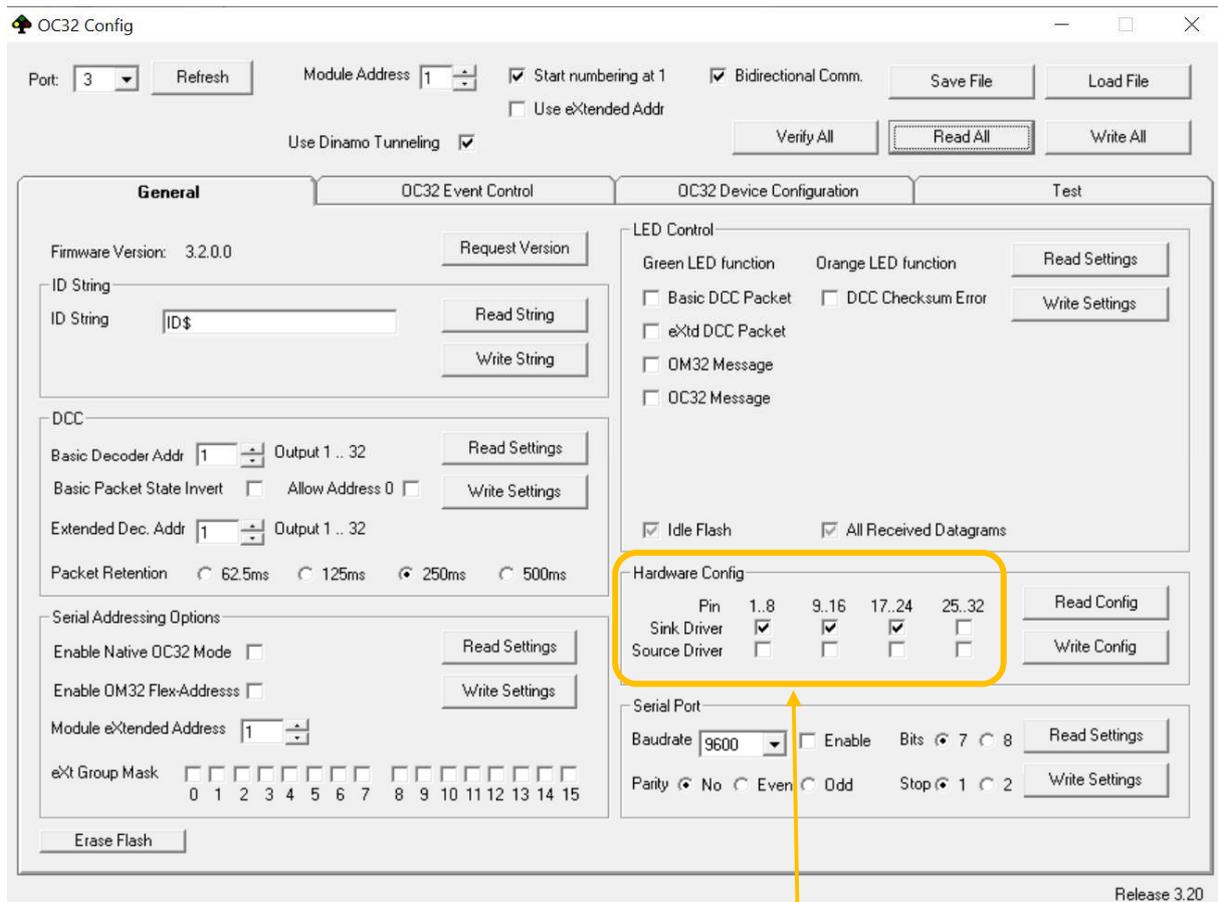


Sehr wichtig sind einige Settings bei den beiden TM44, wenn man Z fährt.

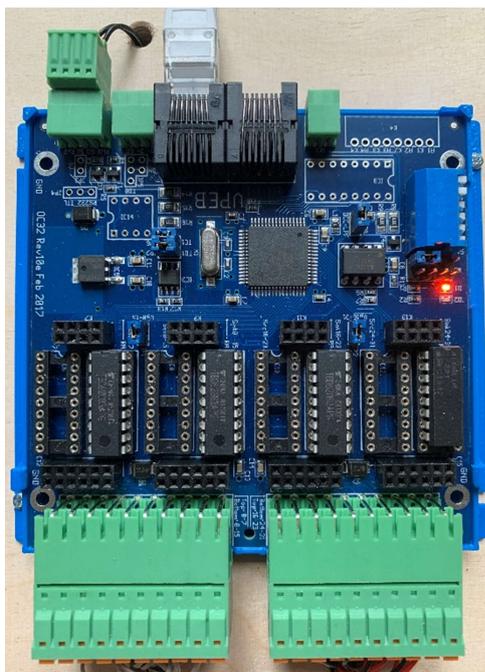
Setzt die PWM-Frequenz für analoge Loks auf ca. 60 Hz

Kurzschlussempfindlichkeit „slow“, sonst werden Loks mit Pufferkondensatoren als Kurzschluss gemeldet

Schwieriger wird es mit der OC32. Hierfür gibt es das Programm „OC32 Config“.



Nachdem per Refresh und Auswahl des richtigen Port die Verbindung steht, habe ich meine „General“-Settings mal ausgelesen.



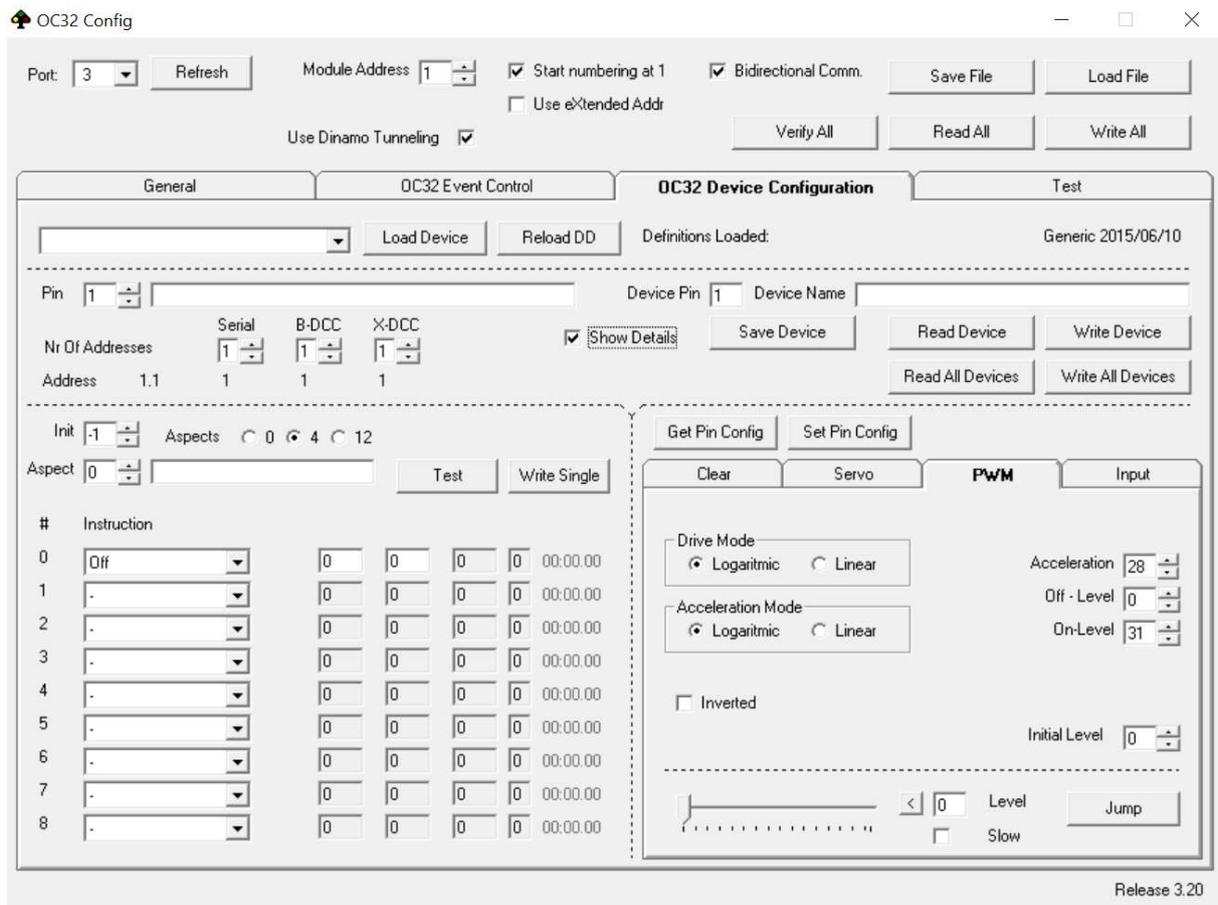
Die OC32 ist wirklich sehr komplex und sehr flexibel. Naja, ihr wisst, was das heißt: Es dauert etwas, bis man die Abläufe verstanden hat, wie man dem Biest beibringt, was es tun soll und (wichtiger) was nicht.

Hier die Innenansicht. Warum ich das zeige ist, man muss (kann) je 8er-Block (also 4 Bänke a 8 Ein-/Ausgänge) evtl. die IC-Sockel anders bestücken. Standard sind „Sink-Driver“, die gegen Masse schalten. Es können aber auch „Source-Driver“ eingesetzt werden, die dann gegen Plus schalten. Oder beide einsetzen, um eine Matrix anzuschließen, oder Widerstandsbänke für Eingänge (Schalter / Taster gegen Masse) und Servo-Ansteuerungen, oder Power-Extensions einsetzen, die man für mehr Schaltleistung benötigt (dann geht aber der Deckel nicht mehr drauf).

Alles klar? Ja, so ging es mir auch.

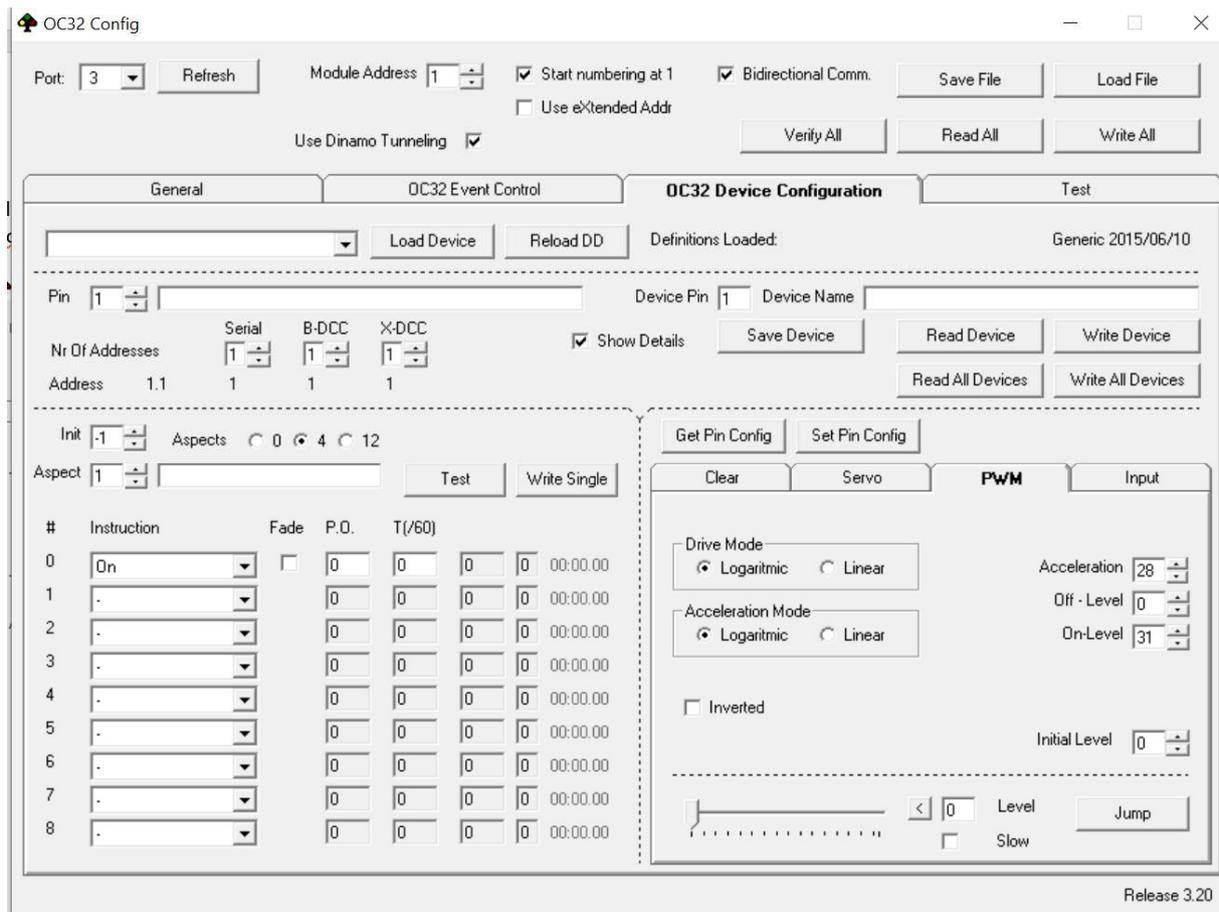
Besonders gern habe ich falsch geklickt bei „Save File“, „Load File“, „Read All“ und „Write All“. Immer sehr gut, um den Blutdruck zu erhöhen bei falschem

Klick. Unter „Event Control“ gibt es nicht so viel zu tun. Viel interessanter ist es bei „OC32 Device Configuration“.



Hier wird jeder einzelne der 32 Pins konfiguriert. Hier abgebildet ist Pin 1. Man muss erst einmal darauf kommen, dass man (rechts unten) „PWM“ auswählen muss, um ein einfaches Relais zu schalten. Mit dieser Option kann man auch Motoren sehr komfortabel steuern. Super Sache, wenn man kapiert hat, wie das geht.

Hier der Zustand „Aspect 0“ am Pin 1. Da soll der Pin 1 ausgeschaltet und somit der PWM-Level auf Null gesetzt werden.



Hier der „Aspect 1“ des Pin 1, wo der Ausgang eingeschaltet wird und der PWM-Level sehr schnell auf das Level 31 gesetzt wird. Damit hat der Ausgang dann gegen Masse geschaltet und das Relais zieht an. In diesem Fall wird ein Signal auf „Grün“ geschaltet.

Das war zu easy? OK! Kein Problem. Dann zu folgender Aufgabe:

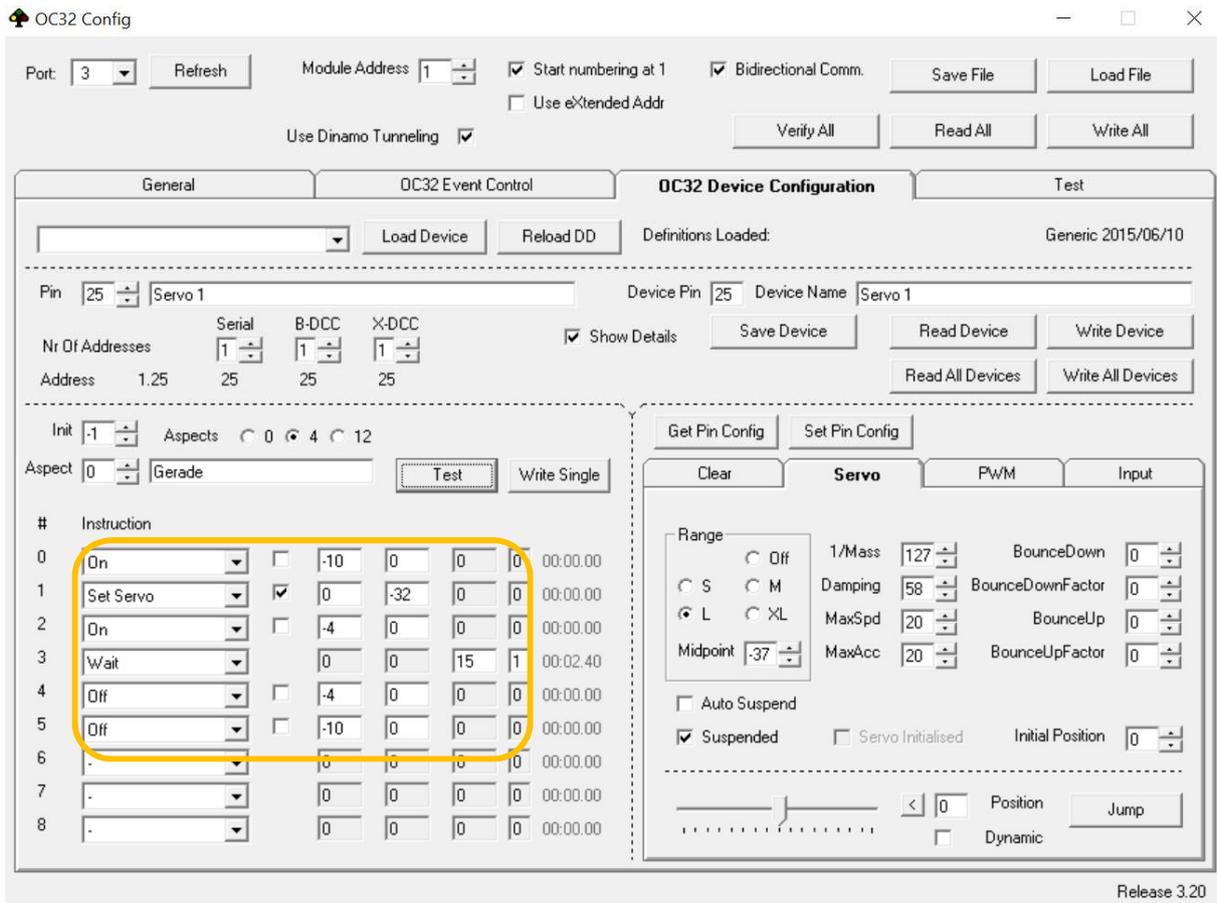
Wenn man Weichen von Weichen-Walter (oder gleich aufgebaute) betreiben möchte, dann muss man die Weiche über ein Servo umstellen UND das Weichenherz umpolen. Mal angenommen die Weichenzunge liegt, vom Servo gezogen, satt an der Schiene an. Wenn man nun, direkt beim Einschalten des Servo auch das Weichenherz umschaltet, könnte es zu einem Kurzschluss kommen. So dachte ich mir folgenden Ablauf aus:

Ein monostabiles Relais hinter diesem, schaltet die Spannung, die zum Weichenherz geht, an und aus.

Ein bistabiles Relais schaltet die Spannung für das Weichenherz von Seite A zu B um.

Gleichzeitig startet der Servo.

Wenn dieser in Endlage ist, schaltet das monostabile Relais die Spannung zum Herzstück wieder an. Auf dem zweiten Umschalter der bistabilen Relais werden LEDs angeschlossen zum Anzeigen, welche Seite nun am Herzstück aktiv geschaltet ist. Das nächste Bild zeigt den Pin25, der als Servoausgang konfiguriert wurde im Zustand „Aspect 0“. Also Servo schalten auf Seite 0 (bei mir – Servo 1 nach links und LED grün). Rechts unten seht ihr die vielen Einstellmöglichkeiten für den Betrieb des Servo. Beindruckend, verwirrend, genial (wenn man es verstanden hat ;-).



Generell lassen sich die Zeilen in „# Instruction“ lesen, wie eine Batch-Datei. Es wird einfach von oben nach unten abgearbeitet. Schlimmer sind dann verschachtelte Instruktionen über „Set Aspect 3“ und in diesem befinden sich weitere Batch-Befehle mit Rücksprung. So etwas wie BASIC programmieren vor 35 Jahren ;-)

Extrem übel zu verstehen ist, wie im gelben Kasten die richtigen Werte einzutragen sind.

Bei Zeile 0 wird der Ausgang 15 eingeschaltet (15...die magische 15....da kommen die -10 her, denn der aktuelle Pin ist 25 und wenn ich 15 schalten will, dann halt „-10“). Das ist das Relais, das an „Normal geschlossen“ die Spannung zum Weichenherz durchleitet. Wenn „On“ dann ist die Spannung abgeschaltet (Grund dafür ist die Grundstellung der Anlage). Bis ich das mit den -10 kapiert hatte, waren einige Flüche, Verzweiflungsschreie und Scotch vergangen.

In Zeile 1 wird der Servo auf die gewünschte Position gefahren.

In Zeile 2 wird das bistabile Relais (Pin 21) umgeschaltet.

In Zeile 3 wird gewartet (durch versuchen festgestellt, was die 15 und 1 bedeutet)

In Zeile 4 wird die Spule des bistabilen Relais stromlos geschaltet

In Zeile 5 wird das Herzstück wieder mit Spannung versorgt.

Eigentlich, auch easy. Oder? Wenn ja, bewerbe Dich bei der NASA oder NSA oder was weiß ich.

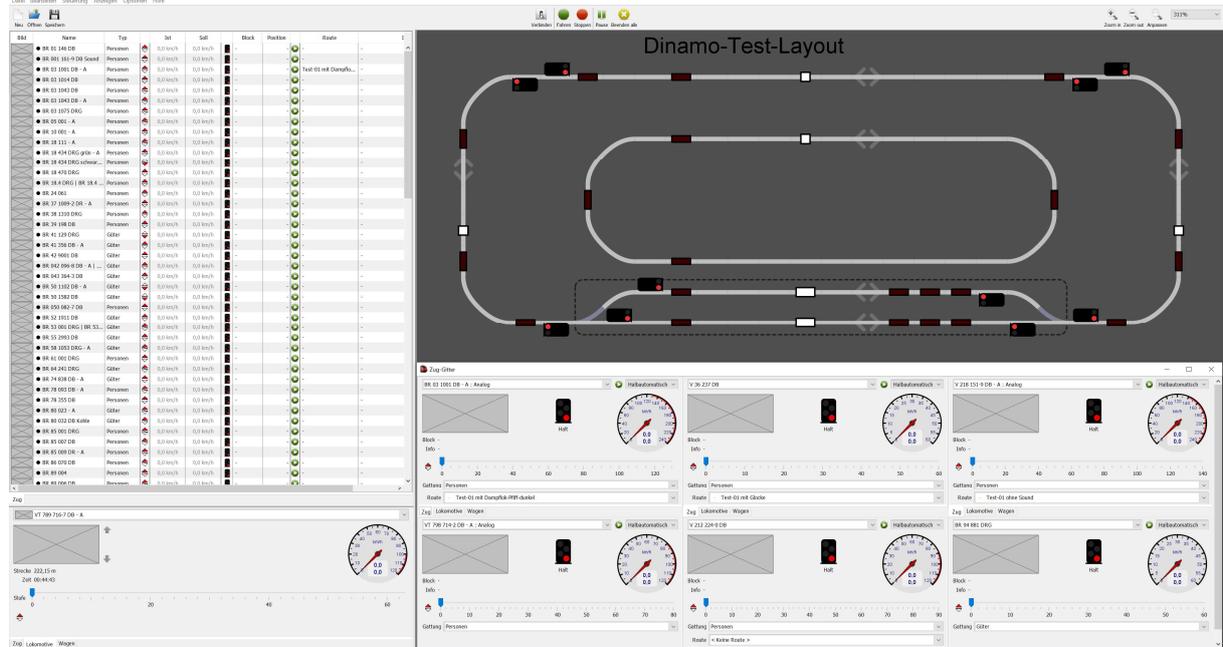
Ganz ehrlich, bis jeder Ein- und Ausgang das und wie getan hat, was ich wollte, war die Flasche Feuerwasser nur noch halb voll und ein paar Monde vergangen. Hugh, ich habe geschlafen.

Da jetzt die Konfiguration abgeschlossen war und ich die zweite Hälfte (Feuerwasser) für iTrain noch hatte, ging es dann an dieser Baustelle weiter, denn ich habe Null Nix Komma X Erfahrung mit iTrain. Ein neues, lustiges Kapitel, das noch nicht abgeschlossen ist (mehr Feuerwasser!!!!).

Generelle Frage, warum iTrain? Einfach zu beantworten: Der Hersteller von iTrain und der Hersteller von Dinamo (beide Holländer) sind sehr gute Freunde. Somit hat iTrain die Dinamo-Schnittstelle (unter vielen anderen) integriert.

Ich habe die Plus-Version gekauft, werde aber auf die Professional upgraden. Grund: Wie im ZFI beschrieben, kann man mit den Professional die App „Spacedesk“ benutzen, die mir sehr gut gefällt.

Zurück zu iTrain:



Der innere Kreis entspricht dem realen äußeren Kreis zum Einmessen. Easy kann ja jeder ;-)
Der (innere) Programmierkreis ist (noch) nicht abgebildet. Nicht sooo wichtig, da ich wegen der Benutzung des Programmers und nicht einer (leider nicht verfügbaren) FCC die Programmierung der Decoder nicht über iTrain machen kann. Mit einer FCC und einem zweiten USB-Kabel kann über iTrain die D&H-Decoder programmieren. Naja, über den Programmer und der D&H-Software geht das auch. Eeeeeasyyyy.

Mein Freund der iTrain. Da muss man auch erst einmal die interne Logik verstehen. Ich möchte keine Lobeshymne ausrufen, aber mit etwas Hilfe vom iTrain-Forum habe auch ich verstanden, wie man z.B. analoge Dampfloks zum Pfiff bekommt vor der Abfahrt am Bahnhof.

Kommen wir zum Wichtigem, der Konfiguration einer analogen Lok:

Hier ein sehr alter 3-pol Schienenbus mit zwei (Motorwagen in der Mitte), elektrisch verbundenen, Anhängern. Sehr wichtig sind die Länge und der Rückmeldungs-Offset. Trägheitssimulation und Reaktionsverzögerung variieren stark, je nach Lok. Unter Geschwindigkeit sollte man die realen km/h + 20% eingeben.

Eigenschaften der Lokomotive

Name: VT 789 716-7 DB - A

Beschreibung: Analog - 3er Gespann

Loktyp: D Diesel Länge: 185 mm

Decoder

Typ: Analog

Pseudo-Adresse: 1052 Kick-Start: 25

Schnittstelle: S Dinamo

Bild Geschwindigkeit Funktionen Konfiguration Optionen Erlaubnis Kommentar

Gleis

Zahnrad Spurweite: Z Polarität: Invertiert

Führerstand

Sichtseite: Vorderseite Symmetrisch

Trägheitsimulation

Beschleunigen Stufenverzögerung: 500 ms Stufenschrittweite: 1

Bremsen Stufenverzögerung: 500 ms Stufenschrittweite: 1

Rückmeldungs-Offset

Belegt Vorderseite: 20 mm Rückseite: 20 mm

Reedkontakt

Lichtschranke

Reaktionsverzögerung

Vorwärts: 400,0 ms Rückwärts: 400,0 ms

Zeitraum

Wartung: 40,0 h Brennstoff: 5,0 h

OK Abbrechen

So, kommen wir zum Einmessen einer Lok.

iTrain hat zwei Möglichkeiten die Position einer Lok zu bestimmen:

1. Belegtmelder
2. Weg-Zeit-Berechnung

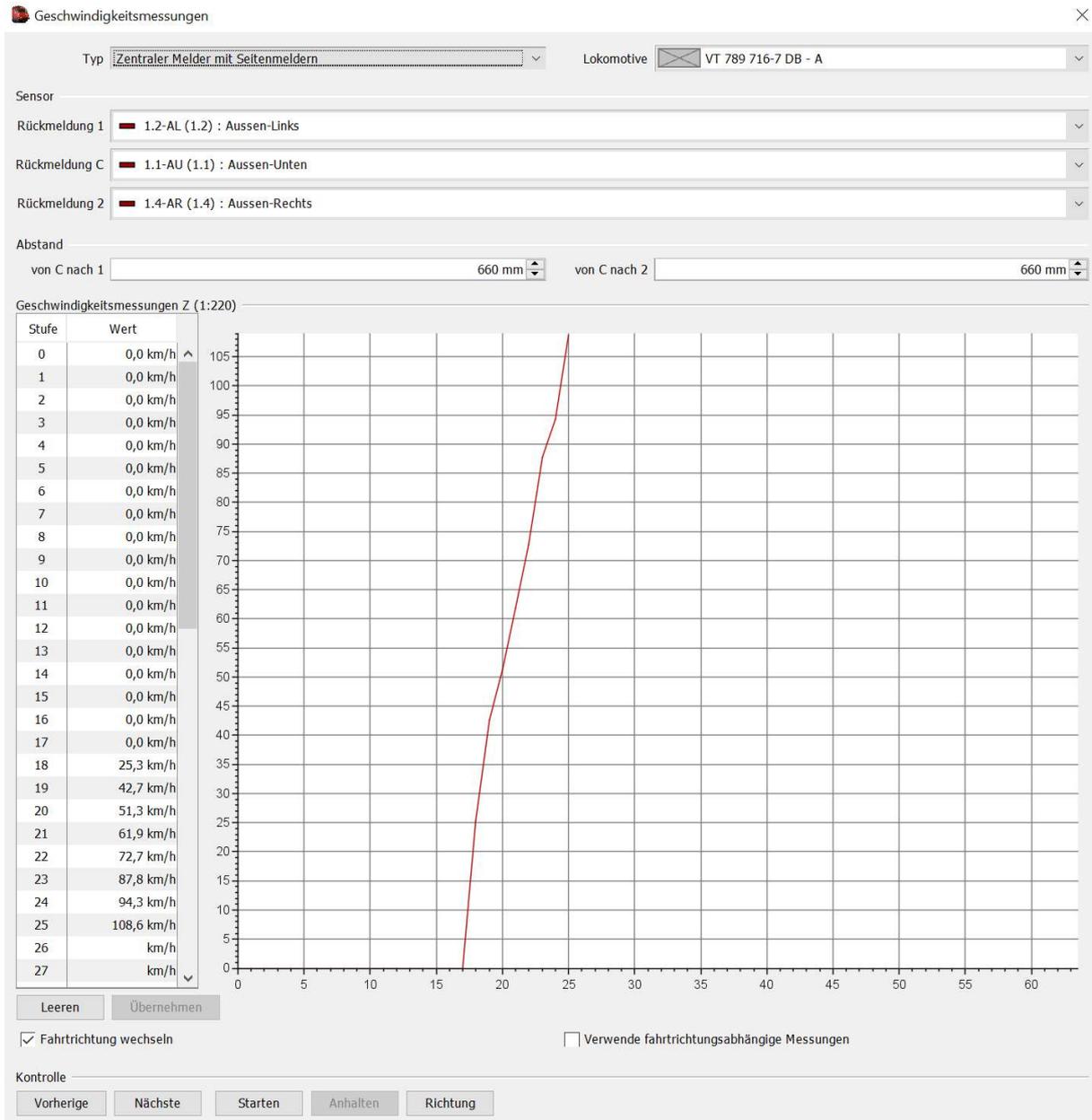
Vorab: Meine Loks fahren mit festgelegten Zügen.

Um die Weg-Zeit-Berechnung so optimal wie möglich zu erreichen, messe ich die Loks mit Wagen ein. Denn, bei analogen Loks unter Last, sinkt die Geschwindigkeit und das teilweise rapide (wenn keine Haftreifen). iTrain korrigiert die Position beim Erreichen eines Meldeabschnittes, aber besser ist natürlich, die wirkliche Geschwindigkeit zu hinterlegen.

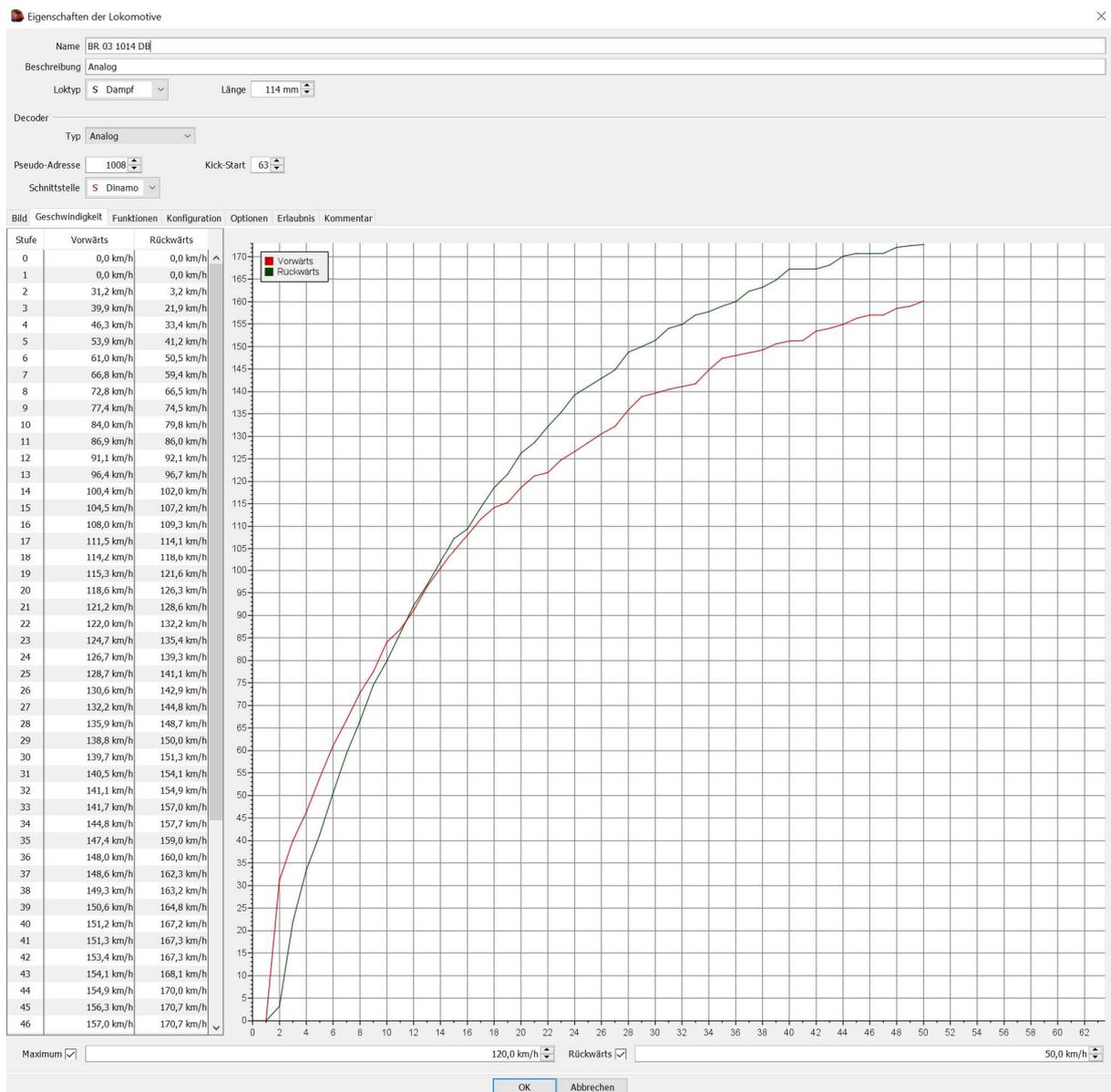
Was mir beim Einmessen aufgefallen ist:

Digitale Loks mit Sound wollen eine Eingangsspannung von 12 Volt an den TM44. Auch manche andere meiner digitalen Loks wollen bei den angestrebten 10,5 Volt nicht, oder unsauber fahren. Ich werde also als Nächstes die Eingangsspannung langsam am Step-Down erhöhen, bis alle digitalen Loks einwandfrei fahren. Das bedeutet auch, dass analoge Loks nochmals eingemessen werden müssen.

Wenn ich mir das Verhalten von analogen Loks betrachte, wäre eine logarithmische Kurve des PWM-Ausgangs für analoge Loks wesentlich besser. Ich habe das bereits beim Hersteller gemeldet, denn mit der aktuellen Auslegung ist, speziell im unteren Geschwindigkeitsbereich, nur eine sehr grobe Abstufung möglich. Das bedeutet, dass die Loks zu stark abbremsen und zu schnell anfahren vom Signal.



Hier die Kurve für die neue 03 1001 mit Glocki:



Oben herum sehr fein abgestuft (es gibt die Fahrstufen 0 bis 63 bei analog). Unten herum zu grob. Je höher die Gleisspannung, desto schlimmer wird das.

Jajajaja, sei nicht so pingelig, sei nicht so ein Nietenzähler. Aber: an meinem analogen Regler habe ich ein logarithmisches Poti und damit kann ich die langsamen Geschwindigkeiten sehr gut abbilden. Da bei den PWM-Einstellung für die OC32 das ja bereits vorgesehen ist, sollte das kein sooo großes Problem sein. Der Hersteller meinte, die Weg-Zeit-Berechnung wird komplizierter. Wir schießen Raketen mit Rovern auf den Mars, aber können so einfache Berechnungen nicht durchführen? Laberrhababer! Ich bleib' dran. Manchmal hilft abnerven ;-)

Das sind, so weit, die Z-spezifischen Eigenheiten. Alles andere ist unabhängig der verwendeten Spur.

Diese Datei wird weitergeführt und am Leben erhalten.

Schlusswort:

Ich bin zufrieden und hoffe auf Updates von iTrain und Dinamo, die sich mehr um unsere Spur Z kümmern. Denn, mal so zwischen uns, bis die wenigen Digitalisierer meine restlichen analogen Loks umgebaut haben, kann ich mit dieser Alternative sehr sehr gut leben.