BiDiB-Wizard

Der BiDiB-Wizard ist ein Java-Programm zur Darstellung der angebundenen BiDiB-Bus-Struktur und der angeschlossenen Knoten.

System Anforderungen

BiDiB-Wizard mit Version bis 1.4.1

- Java 1.6
- getestet mit Linux, Win7, MacOS

BiDiB-Wizard mit Version ab 1.5

- Java 1.7
- getestet mit Linux, Win7, MacOS

BiDiB-Wizard mit Version ab 1.6

- Java 1.7
- getestet mit Linux (x86 und ARM), Win7, MacOS

BiDiB-Wizard mit Version ab 1.9

- Java 1.8
- getestet mit Linux (x86 und ARM), Win7, MacOS

Download

Aktuelle Release-Version:

Die aktuelle Version ist der **Wizard-1.12.4** vom 21.12.2019.

Der BiDiB-Wizard kann hier heruntergeladen werden:
http://www.fichtelbahn.de/files/wizard-fw/bidibwizard-installer-1.12.4.jar

> Der BiDiB-Wizard bringt Unterstützung für mehrere *Serielle Port Treiber* mit. Je nach Betriebssytem kann der eine oder andere Treiber Vorteile haben. Diese *Seriellen Port Treiber* benötigen alle einen korrekt installierten nativen Treiber (z.B. FTDI).
>
> Alle Treiber die zur Auswahl stehen werden komplett mitgeliefert, d.h. es muss keine separate Installation erfolgen.

Unter Windows 10 64-bit muss eine 64-bit JVM installiert und verwendet werden.

Siehe auch USB hotplug.

Der Wizard 1.12 verwendet standardmässig ein anderes Format für den Knoten-Export und Import. Man kann das in den Einstellungen > Experimentelle Eigenschaften > Java-Serialisierung für Knotenexport verwenden auf das alte Format umschalten. Anschliessend den Wizard neu starten.

Bitte neue Knoten-Export nur noch mit dem neuen Format (*.nodex) erstellen! Die Unterstützung für das alte Format wird in einem nächsten Release komplett entfernt.

Im Wizard-1.9 wurde das **Datenformat der Bezeichnungen für Ports, Makros und Accessories geändert**! Aus diesem Grund werden die alten Bezeichnungen beim 1. Zugriff mit einer 1.9-er Version automatisch in das neue Format migriert.

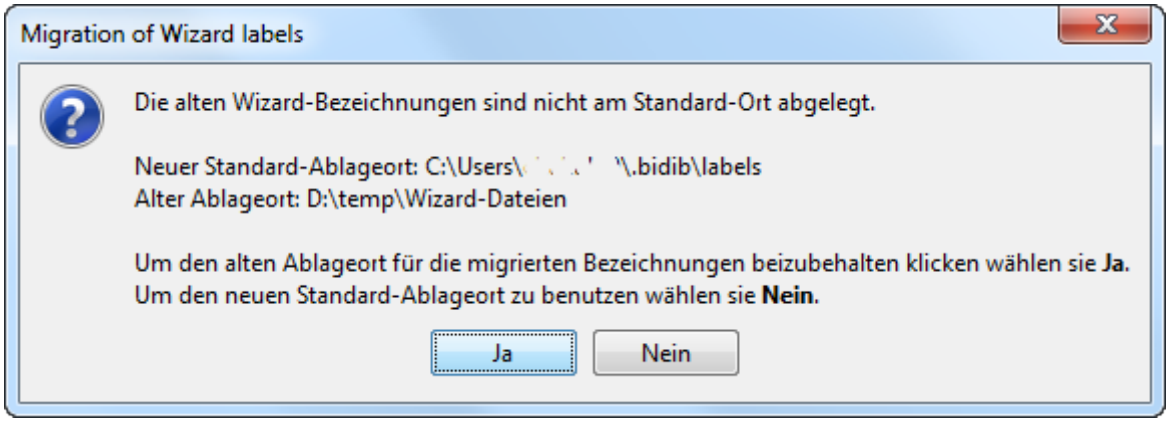
Die alten Dateien mit den Bezeichnungen werden nicht gelöscht und sind immer noch vorhanden für Versionen < 1.9. Anpassungen der Bezeichnungen in der 1.9-er Version sind aber beim Wizard-1.8 nicht sichtbar, ebenso sind Anpassungen der Bezeichnungen in der 1.8-er anschließend in der 1.9-Version nicht sichtbar wenn die Migration schon durchgelaufen ist!

Um die **Migration erneut ausführen** zu lassen muss die Datei mit den entsprechenden Bezeichnungen gelöscht werden (Wizard beenden!), z.B. **LightPortLabels.labels** für Lichtports. Die Dateien liegen im Home-Verzeichnis des Benutzer unter dem Verzeichnis *.BiDiBWizard*, z.B. C:\Users\<username>\.BiDiBWizard

Wichtig ist die Datei mit der Endung **.labels** zu löschen und nicht die gleich lautende Datei mit dem Namen LightPortLabels aber ohne Extension. In diesem Fall sind alle bisher geänderten Bezeichnungen mit dem Wizard-1.9 verloren! Falls Bedarf besteht beide Bezeichnungen weiterhin zu behalten, bitte bei mir (andreas.kuhtz@gmail.com) melden bevor die Dateien gelöscht werden.

Ab Build 2320 wurde das Format für das lokale Speichern der Bezeichnungen (Labels) angepasst. Beim ersten Start erfolgt automatisch eine Migration in das neue Format. Die Dateien des neuen Format liegen unter *.bidib/labels* und haben das Namensformat *<Knoten-UID>.xml*.

Während der Migration erscheint der folgende Dialog, falls die Bezeichnungen nicht im Standard-Verzeichnis gefunden wurden.



Entwicklungsstand

Download Entwicklungsstand

Der aktuelle Entwicklungsstand (Update: 12.01.2019, Build 3877, master) vom *BiDiB-Wizard-1.12* kann hier heruntergeladen werden
http://www.fichtelbahn.de/files/wizard-fw/bidibwizard-installer-1.12-20200112.083720-207.jar

- Ab Build 3180: Lauffähig unter Java 9 und Java 10.
- Ab Build 3016: Die zusätzliche Komponente BiDiB-Wizard-Spy unter Windows muss nicht installiert werden!

Installation

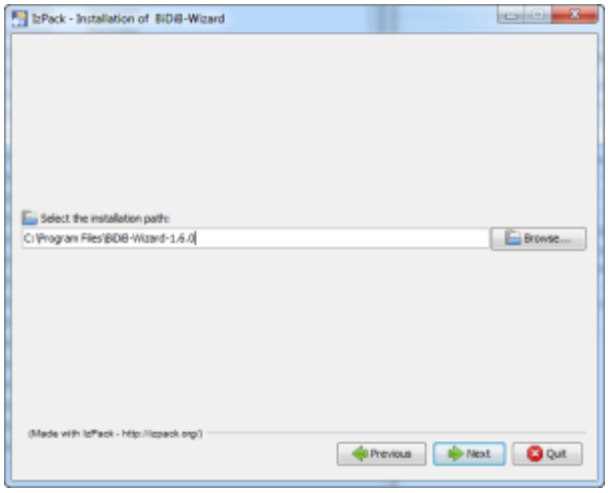
Die Installation des BiDiB-Wizard erfolgt über ein „executable-Jar“ welches im Normalfall unter Windows per Doppelklick gestartet werden kann, wenn eine *Java 8-Runtime* (oder neuer) installiert ist.

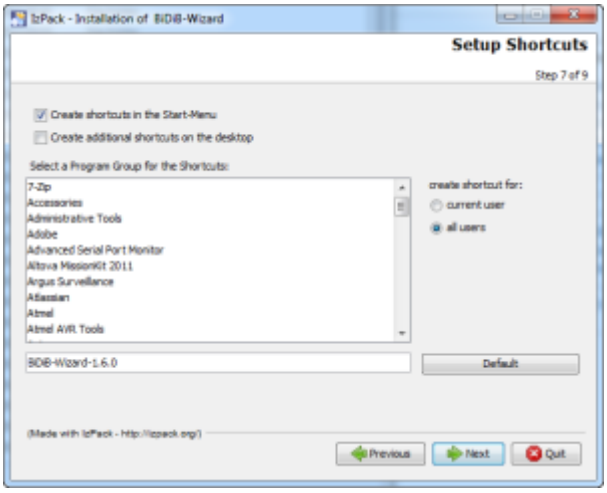
Sollte die Verknüpfung für Jar-Files durch ein anderes Tool verändert worden sein (z.B. 7-zip), hilft folgendes Kommando auf dem Command-Prompt:

```
java -jar bidibwizard-installer-1.x.jar
```

Hinweis: Der Name *bidibwizard-installer-1.x.jar* muss dem tatsächlichen Namen des Installer-jar angepasst werden.

Falls mehrere Versionen des BiDiB-Wizard parallel installiert werden sollen, kann während der Installation der Pfad und der Menü-Name angepasst werden. In den nachfolgenden Abbildungen wurde der Standardwert *BiDiB-Wizard* jeweils in *BiDiB-Wizard-1.6.0* geändert.



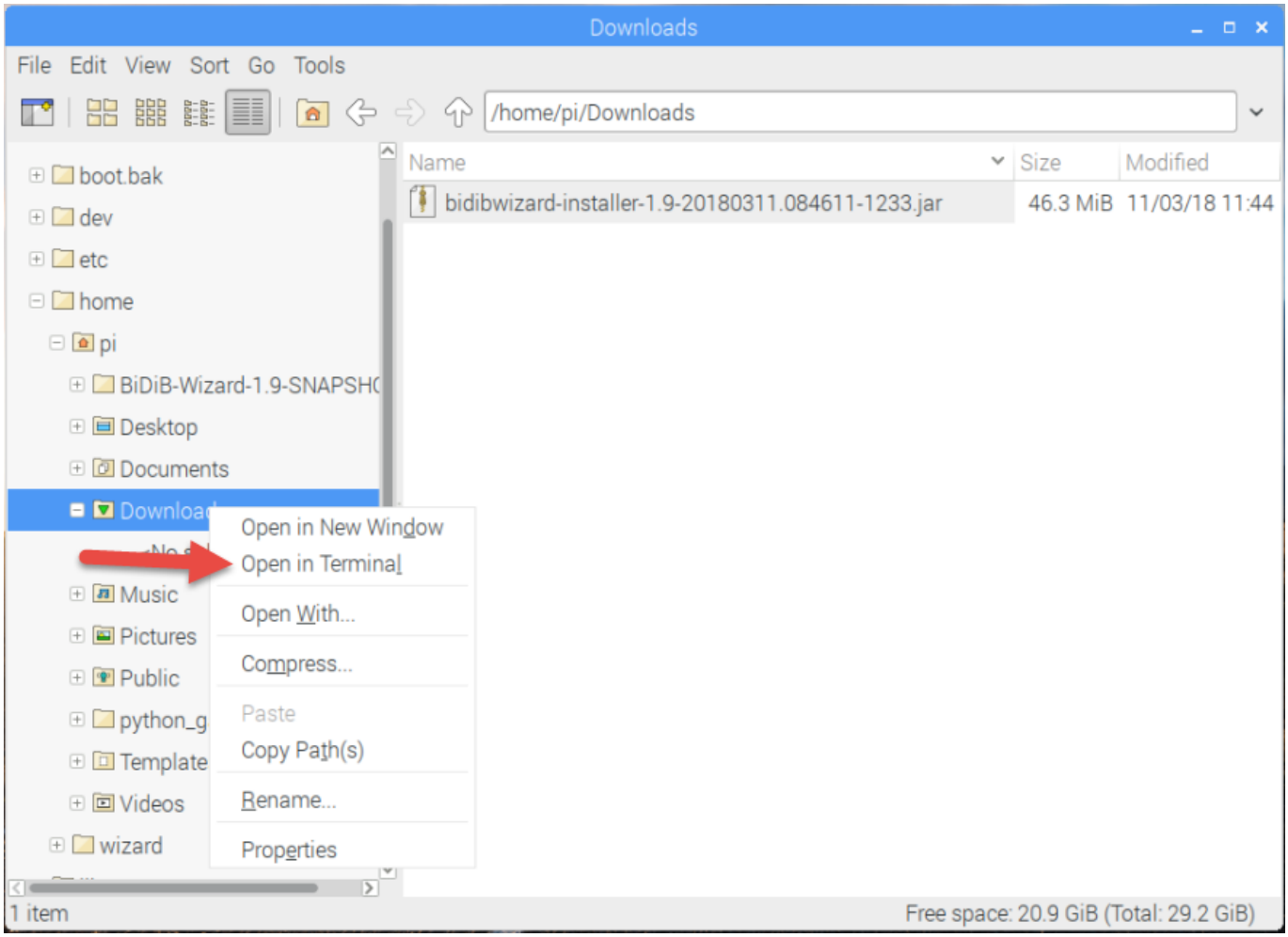


Nach der Installation kann der BiDiB-Wizard über das Menü gestartet werden.

Bevor erstmalig eine Verbindung zum Master erstellt werden kann, muss der verwendete Port über **Bearbeiten > Einstellungen** ausgewählt werden (USB-Verbindung zum GBMboost muss eingesteckt sein). Anschließend kann die Verbindung über **Datei > Verbinden** oder über den Connect-Button hergestellt werden. Sobald die Verbindung zum Master steht, werden alle Knoten gelesen und in einer Baumstruktur dargestellt.

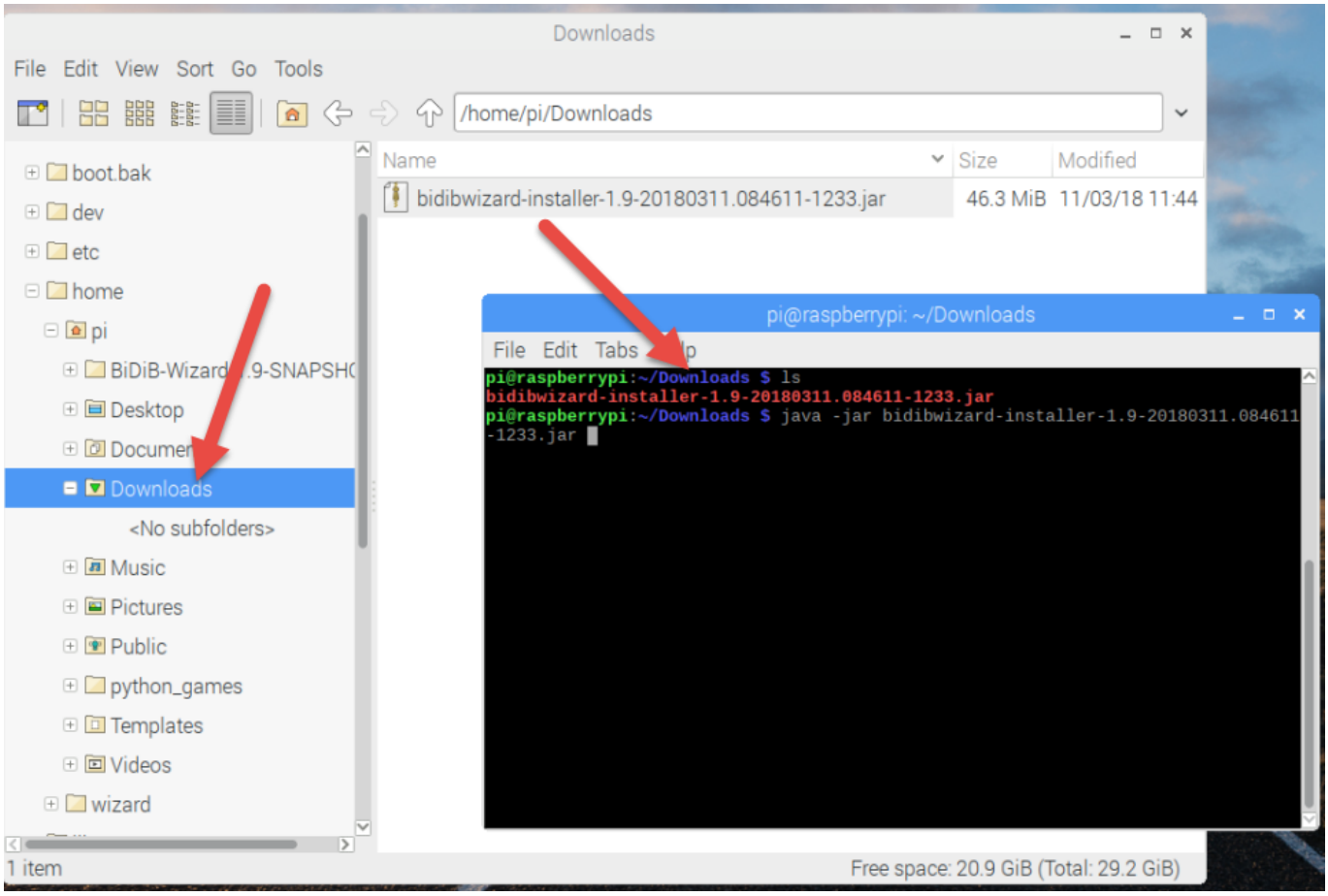
Installation auf Raspi 3

Auf dem Raspberry Pi funktioniert die Installation über Doppelklick nicht. Um den Wizard zu installieren muss ein Terminalfenster in dem Verzeichnis geöffnet werden wo das Installer-Jar liegt.

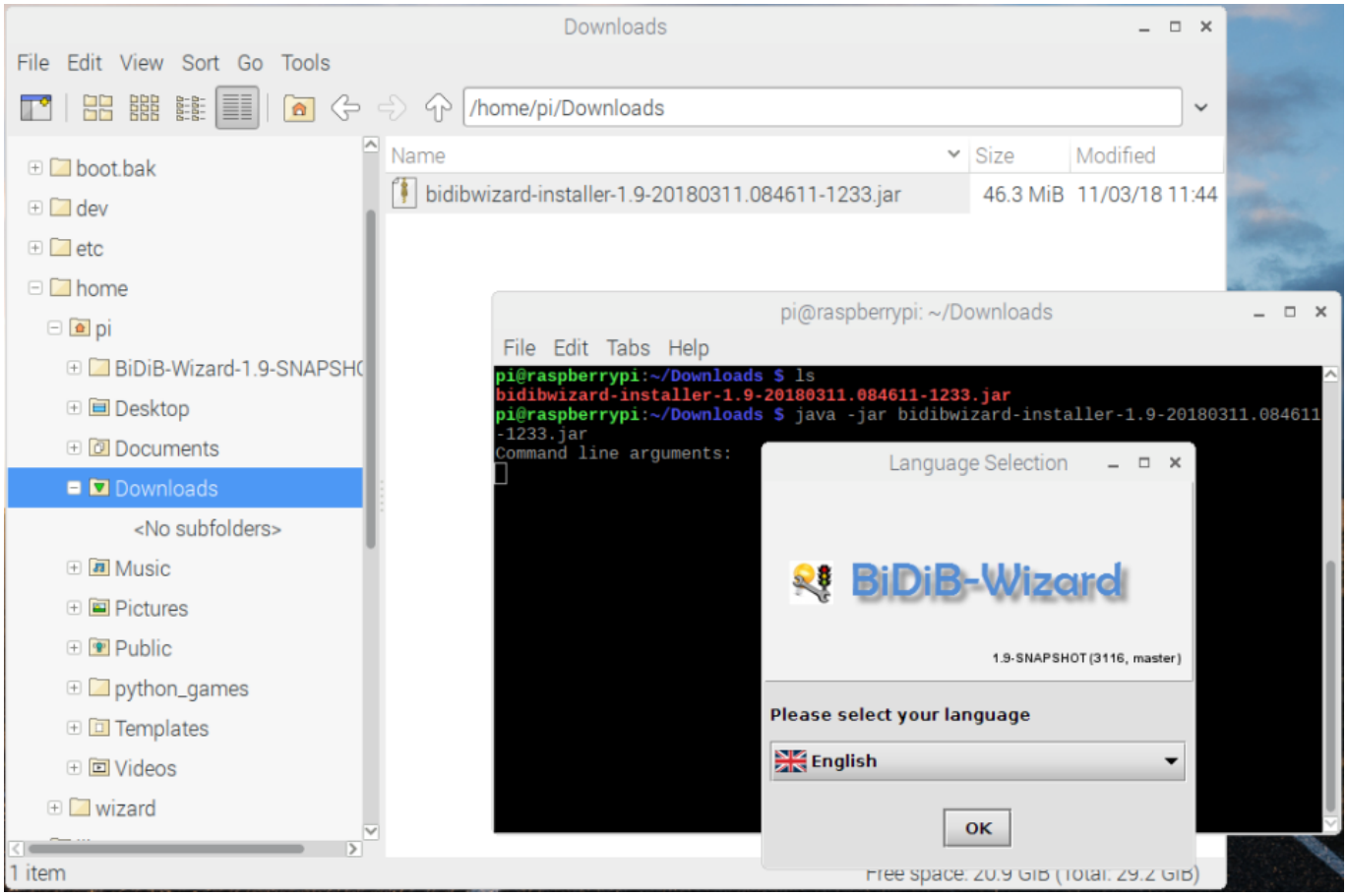


Mit dem Befehl `ls` kann man die Dateien im Verzeichnis anzeigen lassen. Um die installierte Java-Version anzuzeigen, kann der Befehl `java -version` verwendet werden. Aktuell wird eine Java 8 Version benötigt.

Ob das Terminal-Fenster im richtigen Verzeichnis geöffnet wurde, lässt sich mit dem Befehl pwd ermitteln, welcher das aktuelle Verzeichnis ausgibt. Um in ein anderes Verzeichnis zu wechseln kann der Befehl `cd <Verzeichnis>` verwendet werden, also z.B. `cd Downloads` um ins Unterverzeichnis *Downloads* zu wechseln.



Mit dem Befehl `java -jar bidibwizard-installer-1.9-20180311.084611-1233.jar` kann der Installer für den Wizard gestartet werden. Der Name der Datei `bidibwizard-installer-1.9-….jar` im Beispiel muss an den richtigen angepasst werden.



Update Java auf Raspi

Grundsätzlich sollte der Wizard mit jeder Java 8 Installation laufen. Falls ein Update auf eine aktuelle Java Version notwendig ist, kann ich diesen Blog empfehlen. Dort wird der Update der Java-Version auf dem Raspi beschrieben. Ich habe den Abschnitt „Installing Oracle Java“ verwendet um meinen Raspi zu aktualisieren.

DMX-Modeler

Zur Anleitung für den DMX-Modeler für die OneDMX gehts hier: DMX Modeler

---

Versionsübersicht

Version 1.12

Aktuelle Version

- Unterstützung für ReadyLine-Baugruppen
- Neues Dateiformat für Import/Export der Knoten

Version 1.11

- Unterstützung für OneControl 3.x

Version 1.10

Version 1.9

- Logfiles automatisch als Zip-Datei komprimieren
- Benutzerdefinierbare Namen für Rückmelder und Aspekte.
- Auswahl des Java-Treiber für den Seriellen Port (RXTX und SCM)
- USB hot-plug: Eingesteckter GBMboost wird mit COM-Port angezeigt
- NodeScript Wizard
- Unterstützung für Protokoll-Version 0.7

- Installation als Applikation unter OSX (1.9.2), Support für Copy/Paste etc. unter OSX
- Unterstützung Raspi 3 (ARMv7) (1.9.2)
- Positionen von OpenCar via BiDiB RF-Basis
- Konfiguration StepControl
- Auswertung Feature FEATURE_RELEVANT_PID_BITS (1.9.3)

Version 1.8

- Initialisierung Lok-Decoder mit FS0 nach dem Verbinden (Adresse in Channel 2).
- SUSI-CV auslesen per POM für D&H-Dekoder
- Anzeige von DYN-Status-Werten
- Knotenliste alphabetisch sortiert
- Unterstützung für Protokoll-Version 0.6
- Unterstützung Debug-Interface (wie hterm)
- Unterstützung für Tams Multi-Decoder
- DMX-Modeler
- CV-XML-Datei in Abhängigkeit von der Firmwareversion des BiDiB-Knoten

Version 1.7

- Unterstützung für DCC Accessory Decoder (Schaltdecoder).
- Unterstützung für POM (Lokomotivdecoder).
- Unterstützung für neuen Makrobefehl „Warten auf Ende der Servobewegung“.
- Unterstützung für neuen Makrobefehl „Flag abfragen 0“.
- Booster Tabelle und RX/TX-Log-Anzeige.

Version 1.6

- Warnhinweis wenn GBMboost im DEBUG Mode gestartet ist.
- CV-Verwaltung für die BiDiB-Knoten: Konfiguration Kehrschleife.
- Lesen und schreiben des benutzerdefinierten Node-Namen.
- Überarbeitung Layout auf Docking Frames.

- Toolbar für Connect/Disconnect.
- Erweiterungen im Simulations-Modus.
- Unterstützung für GPIO-Ports der OneControl.
- Lauffähig auf ARM-Prozessoren (z.B. Raspberry Pi).
- Unterstützung UDP-Protokoll für BiDiB (experimental).
- Unterstützung für Programmiergleis (PT).

Version 1.5

- Alle Knoten am Bus werden in einer Baumstruktur angezeigt.
- Unterstützung für hierarchische Strukturen mit OneHub.
- CV-Verwaltung für die BiDiB-Knoten (nicht für Lok-Dekoder).

Version 1.4.1

- Alle Knoten am Bus werden in einer Listenstruktur angezeigt.
- Auflistung von Firmware-Stand und Features aller Knoten.
- Belegt-Meldungen werden angezeigt. Wenn unterstützt auch Adressen, Geschwindigkeit und Aufgleisrichtung.
- Features eines Knoten können geändert werden.
- Jeder Knoten kann mit einer kurzen Beschreibung versehen werden.
- Modellbahnuhr mit einstellbarer Geschwindigkeit / Anfangs- u. Endzeit / Wiederholung.
- Firmware-Update über den BiDiB-Bus.
- Booster ein-/ausschalten, Anzeige Stromverbrauch, Temperatur und Spannung.
- Fahrpult für Vor-/Rückwärts, Halt, Nothalt und bis zu 28 Funktionstasten.
- Programmierung von Makros und Accessories.

Probleme und Fehler

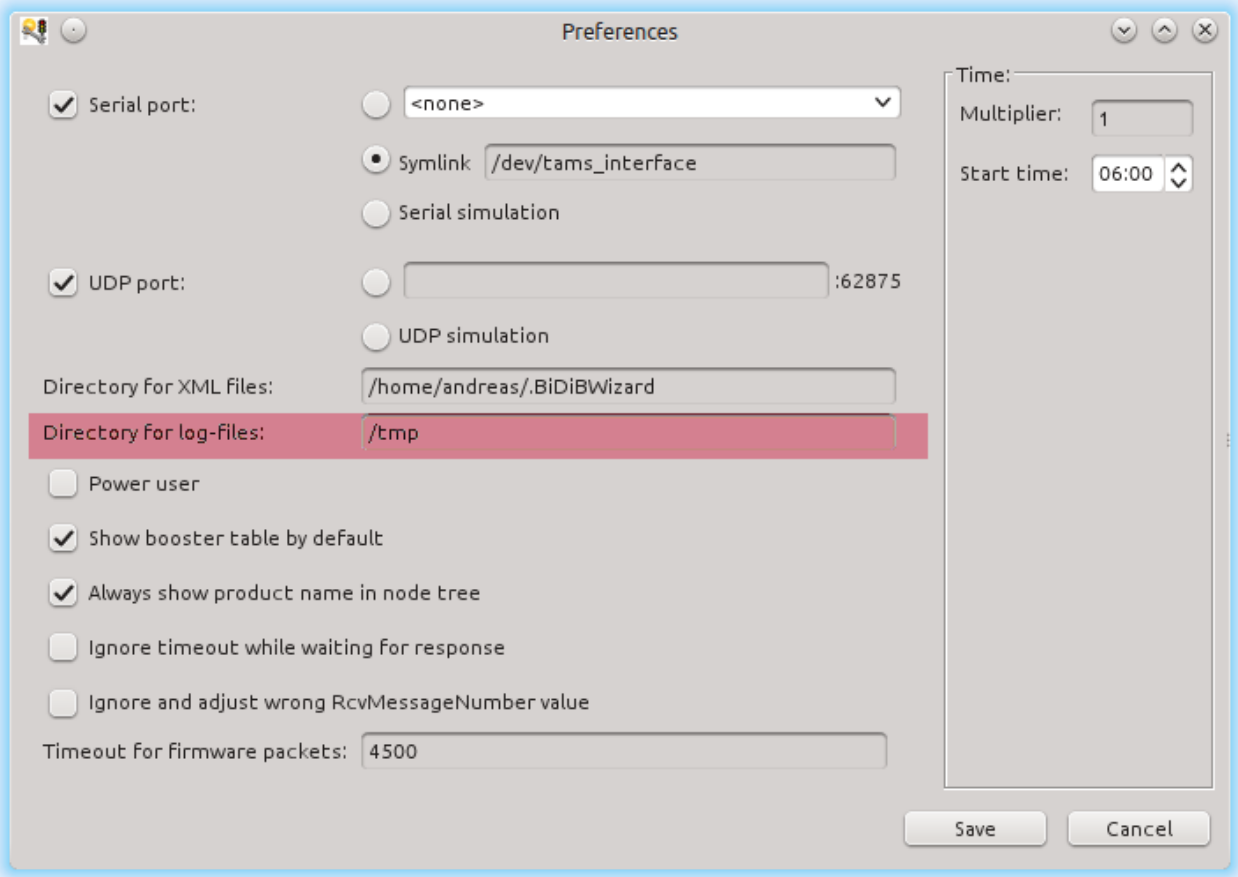
Um Probleme und Fehler analysieren zu können werden meistens Logfiles benötigt, in welchen der BiDiB-Wizard während der Ausführung den Ablauf und Fehlermeldungen mit mehr Informationen protokolliert als auf der Benutzeroberfläche ausgegeben wird.

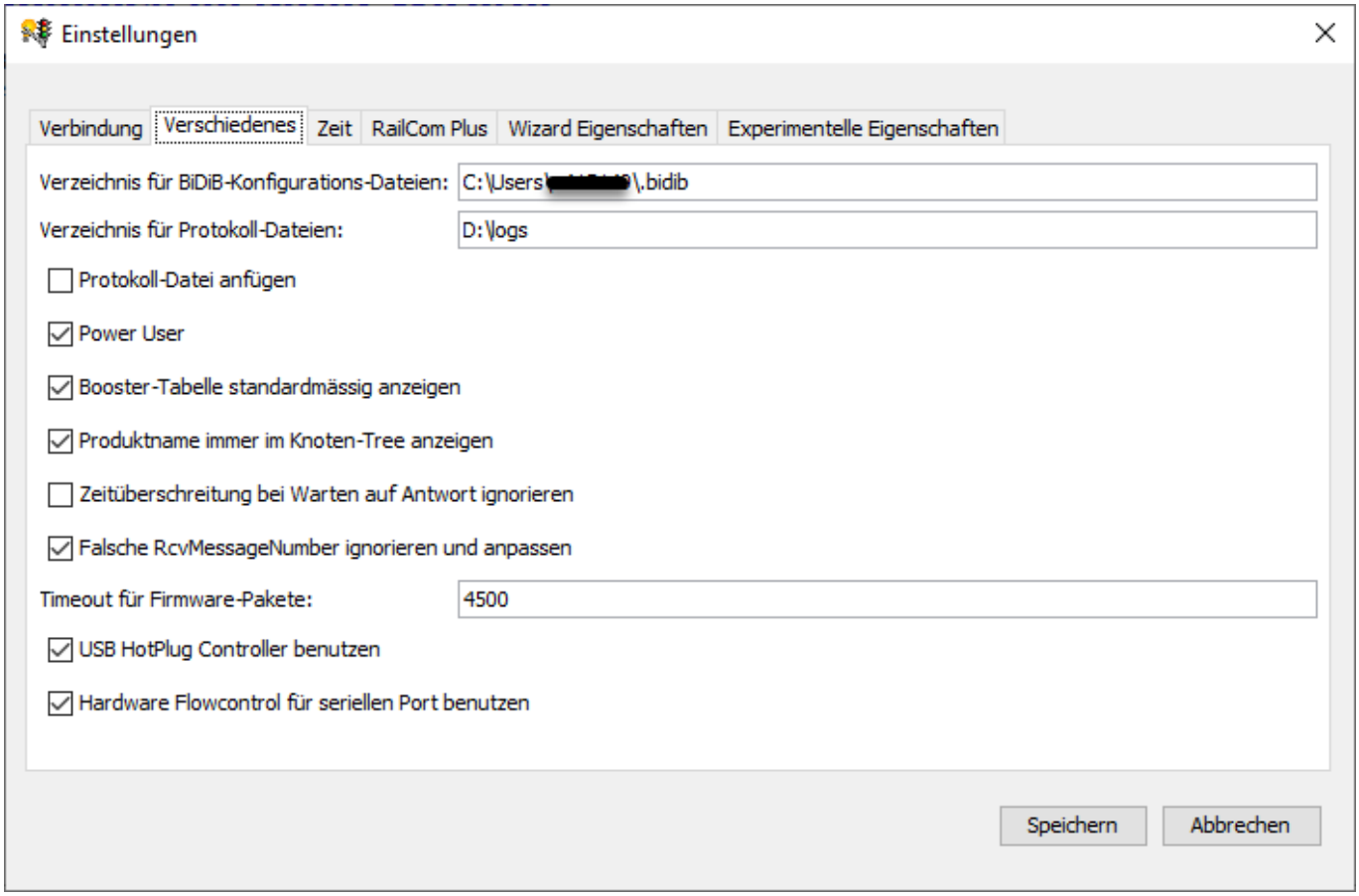
Diese Dateien werden bei jedem Start überschrieben, weshalb es im Fehlerfall wichtig ist die Dateien zu kopieren bevor der BiDiB-Wizard erneut gestartet wird.

Logdateien

Die Logfiles BiDiBWizard.log und BiDiBWizard-RXTX.log liegen standardmäßig unter *C:\Users\<username>\AppData\Local\Temp* (Siehe auch unter Menü *Bearbeiten > Einstellungen … > Verzeichnis für Protokoll-Dateien*). Ab BiDiB-Wizard-1.7 können die 2 Logfiles auch über das Menü *Hilfe > Logdateien sammeln* in ein Verzeichnis freier Wahl kopiert werden.

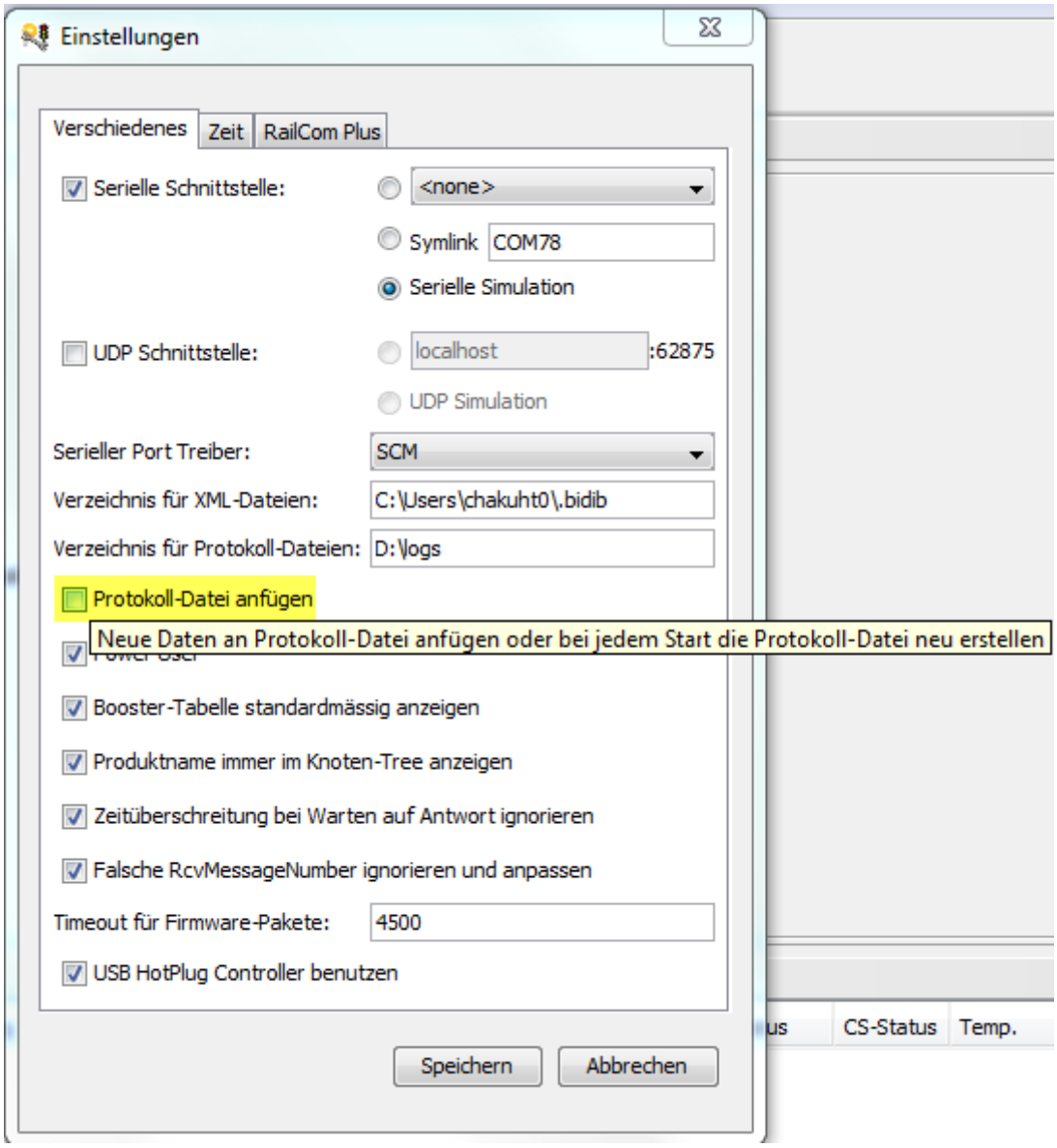
In den Einstellungen steht der Pfad zu den Protokolldateien (log-files). Unter Linux liegen diese standarmäßig unter */tmp*.





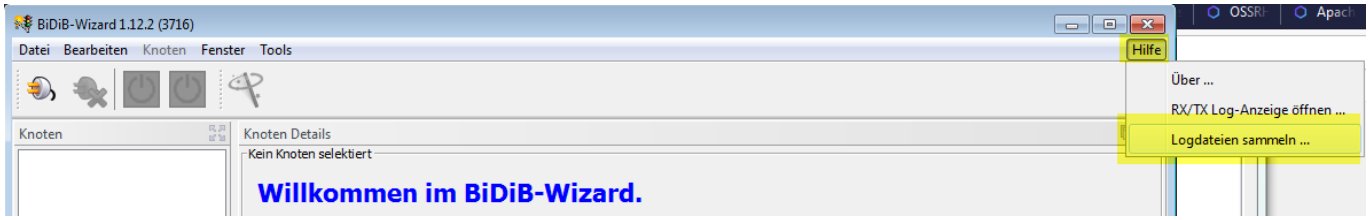
Ab dem Wizard Build 2841 gibt es zudem die Möglichkeit, das Logfile nicht mehr beim Start überschreiben zu lassen. Dadurch kann man im Fehlerfall die Daten nachträglich noch sammeln

lassen.

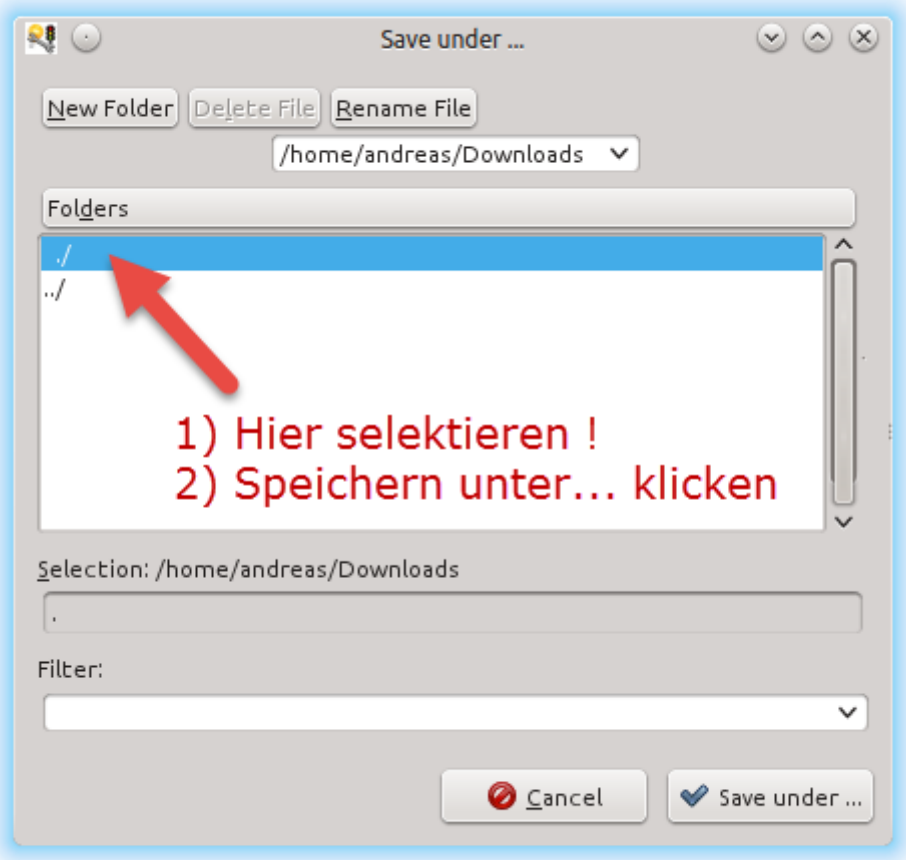


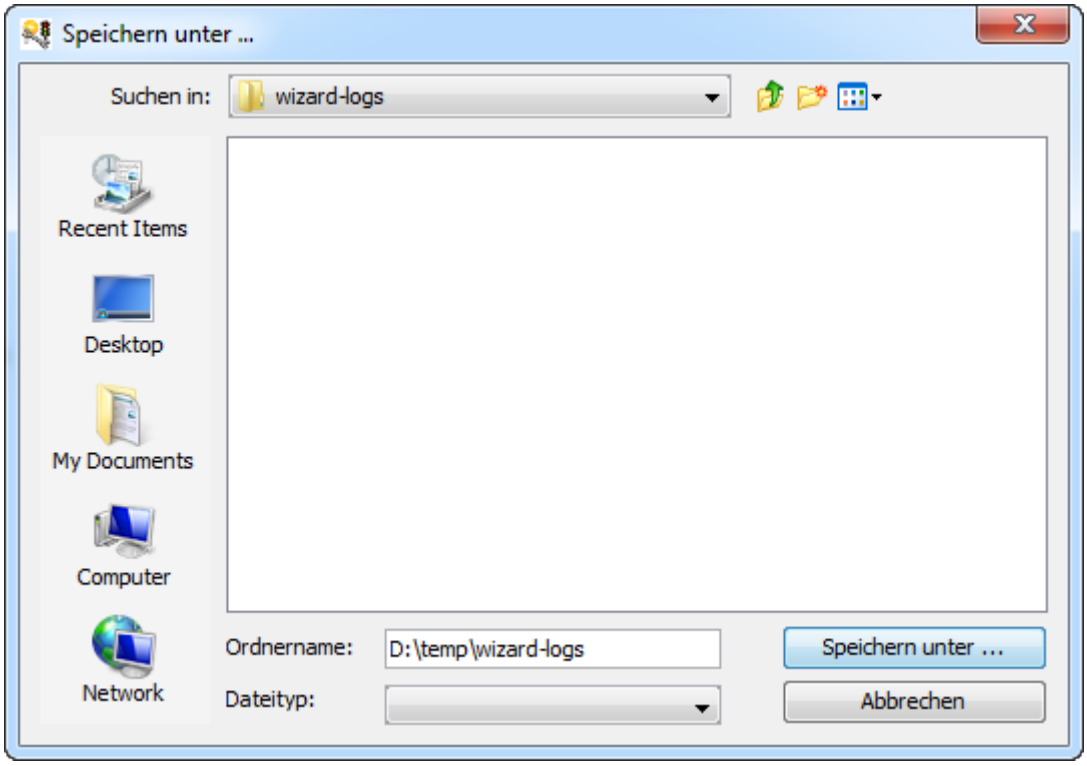
Logdateien sammeln

Der Dialog zum Sammeln der Logdateien wird über das Menü ***Hilfe > Logdateien sammeln*** geöffnet.



Mit dem nachfolgenden Dialog kann das Ziel festgelegt werden, an welchen die Logdateien (BiDiBWizard.log und BiDiBWizard-RXTX.log) kopiert werden:



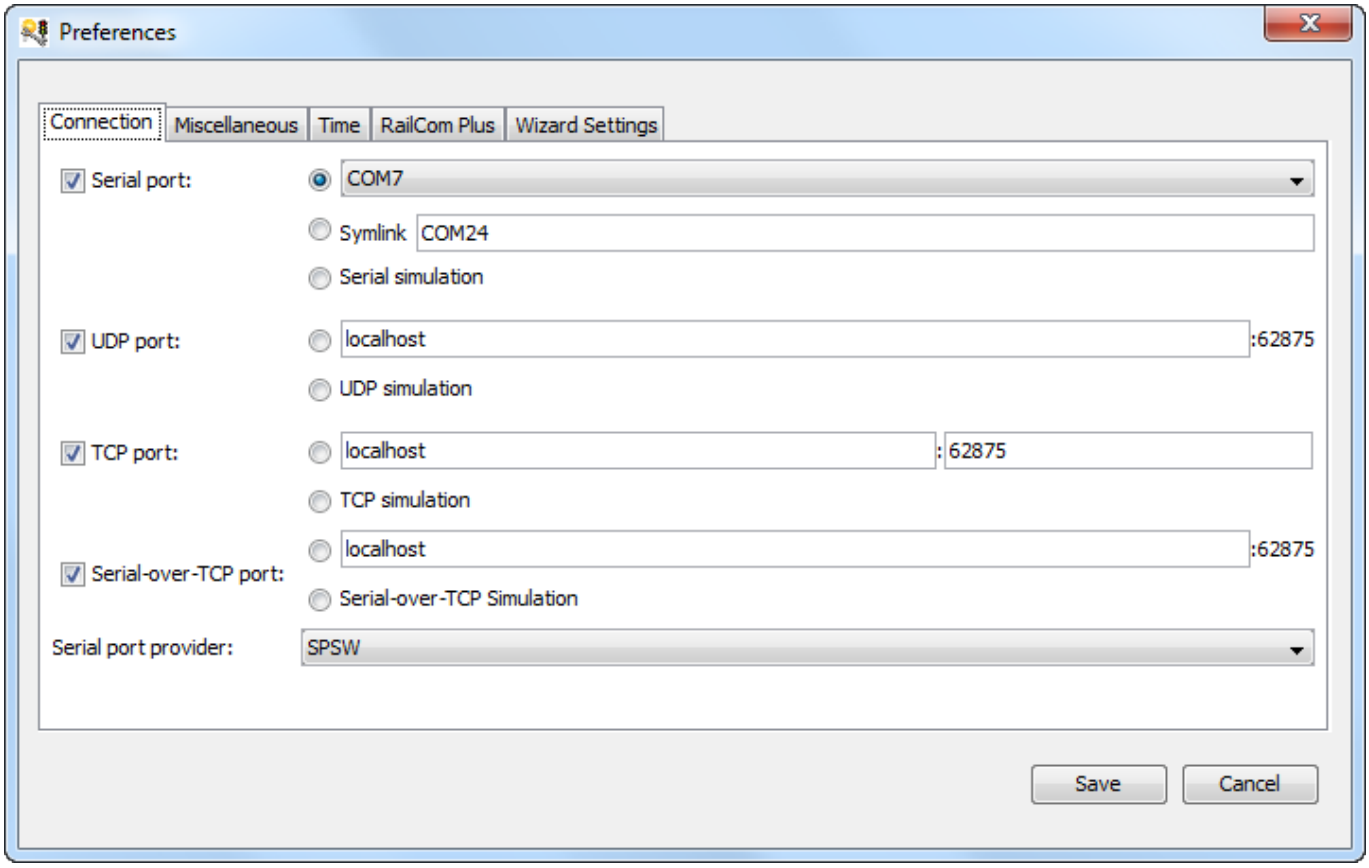


Anschließend befinden sich die Logdateien im ausgewählten Verzeichnis und können z.B. im Forum angehängt werden.

Hinweis: Um Platz zu sparen bitte die Logdateien nach Möglichkeit in eine Zip-Datei verpacken bevor sie ins Forum gestellt werden.

Verbindung einrichten

Um eine Verbindung auf den GBMBoost herzustellen, muss die serielle Schnittstelle (COM) ausgewählt werden. Unter Windows werden die verfügbaren Schnittstellen in der obersten Combobox angezeigt. Unter Linux muss der Symlink (z.B. /dev/ttyUSB0) im Eingabefeld Symlink eingegeben werden.



Neben der Auswahl der Schnittstelle kann der verwendete Treiber für den Seriellen Port ausgewählt werden. Hier stehen „RXTX“, „SCM“ und „SPSW“ zur Verfügung. Standardmässig wird RXTX verwendet.

SCM

Der „SCM“-Treiber sollte für folgende OS funktionieren:

- Windows (32- und 64-bit)
- Linux (32- und 64-bit, embedded)
- MacOSX (32- und 64-bit)

Unter Linux sind bisher folgende Probleme bekannt:

- SCM kann nicht geladen werden weil *libudev.so.0* nicht vorhanden ist:

Dieses Problem kann durch Einrichten eines Link (Adminrechte notwendig) behoben werden: `ln -s`

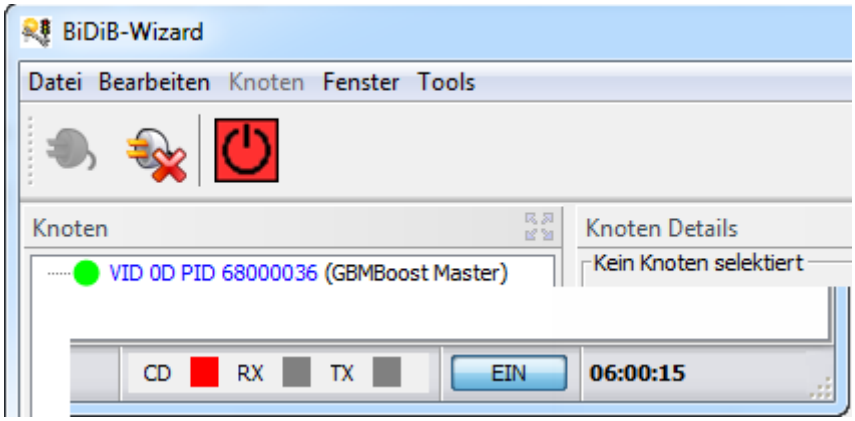
```
/lib/x86_64-linux-gnu/libudev.so.1 /lib/x86_64-linux-gnu/libudev.so.0
```

Quelle:
http://askubuntu.com/questions/288821/how-do-i-resolve-a-cannot-open-shared-object-file-libudev-so-0-error

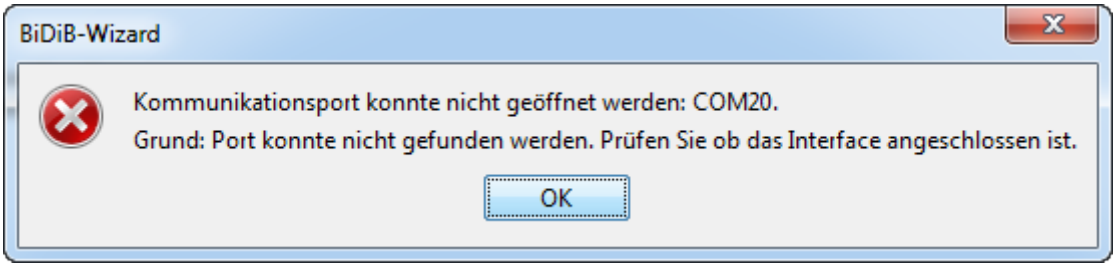
Falls ein Anwender ein Problem mit SCM findet bitte im Forum melden. Danke!

Verbindung herstellen

Nach dem Speichern der Konfiguration kann die Verbindung zum GBMboost über den Toolbar-Button mit dem Stecker oder das Menü aufgebaut werden. Bei erfolgreichem Verbindungsaufbau wird der Master (hier GBMBoost) angezeigt und das CD-Icon (connected) in der Statuszeile wird rot dargestellt.



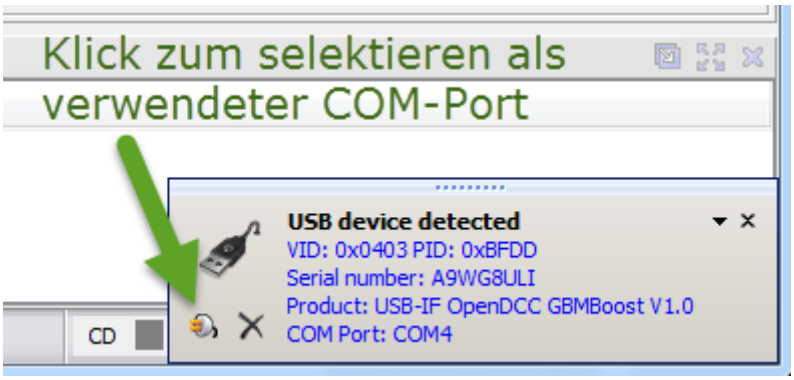
Falls der COM-Port nicht verfügbar ist, wird folgende Fehlermeldung angezeigt:



Falls der serielle Bootloader am GBMboost aktiviert ist (Jumper J3 geschlossen und Button gedrückt beim Start), wird folgende Fehlermeldung angezeigt:

USB Hot-plug

Ab Version 1.9 wird der USB-Controller überwacht. Sobald ein USB-gerät mit COM-Port eingesteckt wird, prüft der Wizard ob eine COM-Schnittstelle zugeordnet ist. Falls dies der Fall ist erscheint ein Hinweisdialog. Durch Klick auf das (kleine) Icon wird der Port als selektierter Port in den Settings eingetragen.



Ab Build 2507 wird anschliessend automatisch die Verbindung zum Interface aufgebaut.

Falls das Fenster nicht erscheint müssen vermutlich die „Visual C++ Redistributable für Visual Studio“ (Google nach: „vcredist_x64.exe 2013“ oder https://support.microsoft.com/de-de/help/4032938/update-for-visual-c-2013-redistributable-package, Achtung: Für Windows passende 64-bit oder 32-bit Version downloaden) installiert werden.

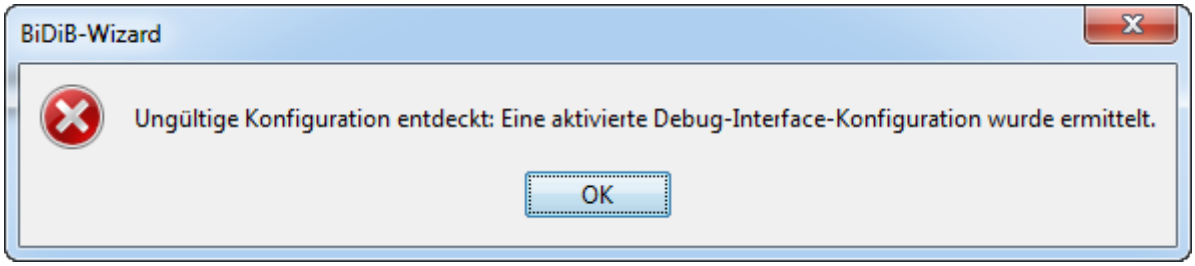
Command Line Argumente

Ab Build 2860 können folgende Command Line Argumente mitgegeben werden:

- `-autoconnect=true | false` : Automatisches Verbinden zum Interface mit den Werten aus den Preferences
- `-startupmode=normal | iconified` : *iconified* → Die Wizard-Applikation wird minimiert gestartet (Windows: nur Button in Taskbar sichtbar)

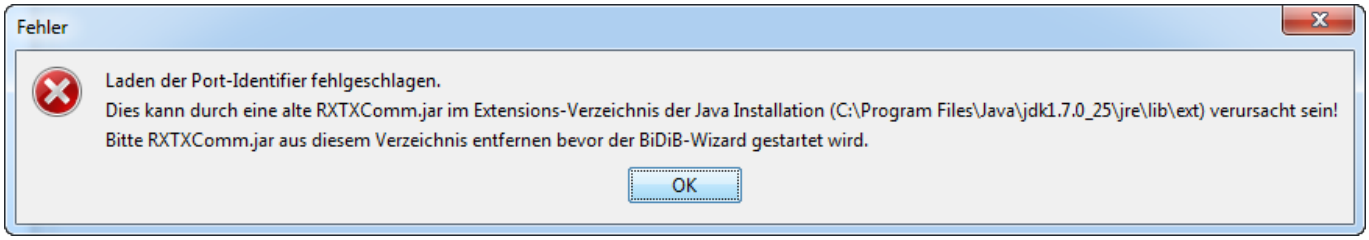
DEBUG-Mode

Wenn der GBMboost im DEBUG Mode gestartet wird, weil der Jumper0 oder das CV1 gsetzt ist, wird nach dem Programmstart folgende Meldung angezeigt.



RXTXComm.jar in Java Extensions-Verzeichnis

Falls sich ein RXTXComm.jar im Java Extensions-Verzeichnis befindet, so führt dies u.U. zu Problemen beim Betrieb des Wizard. Wenn ein solches Problem erkannt wird folgender Fehlerdialog angezeigt:

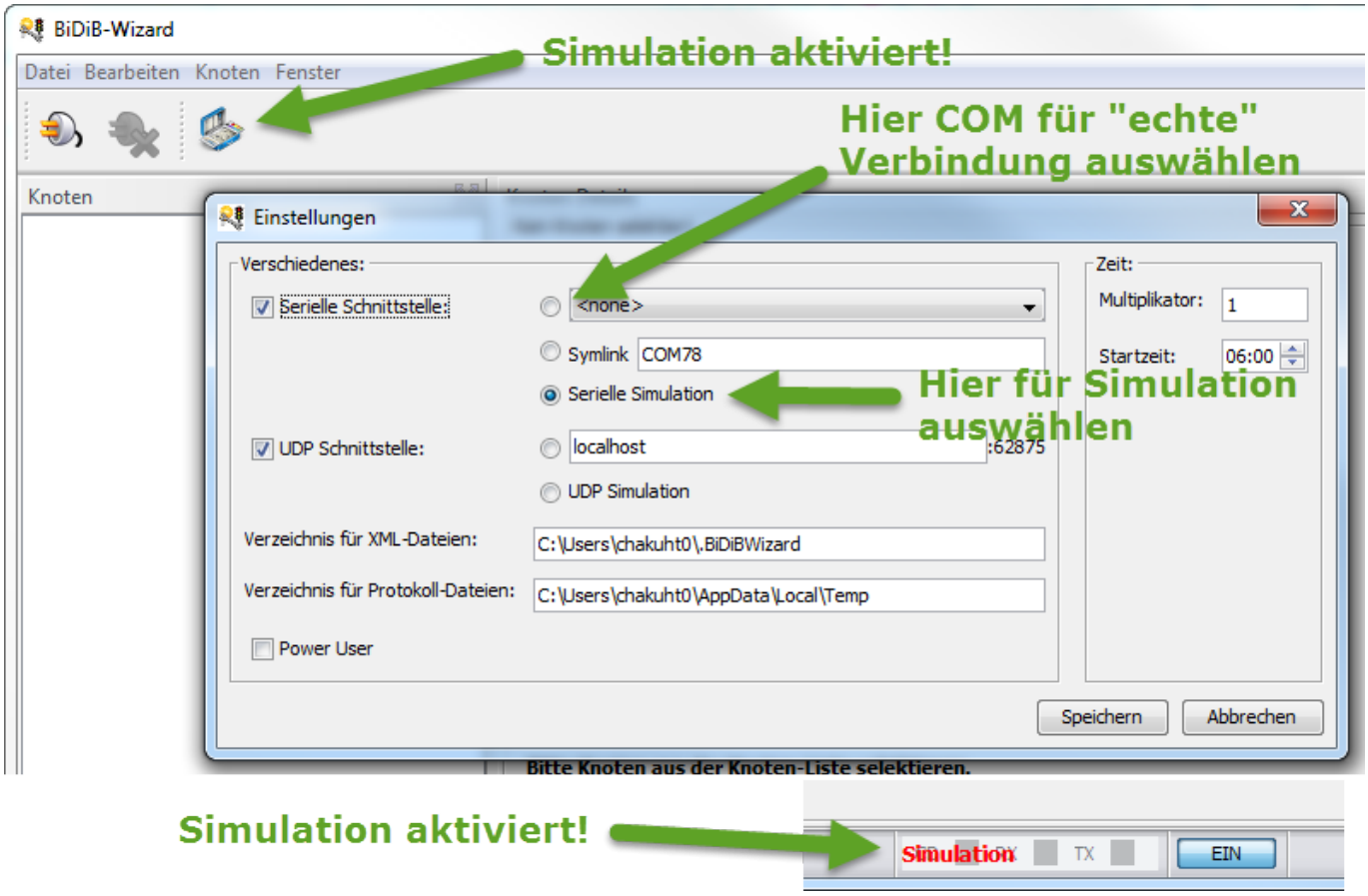


In diesem Fall muss die RXTXComm.jar aus dem angezeigten Verzeichnis entfernt werden damit der Wizard korrekt arbeitet.

Simulations-Modus

Der BiDiB-Wizard kann für Offline-Test in einem Simulations-Modus betrieben werden. Bei dieser Betriebsart werden Knoten simuliert und es erfolgt kein Zugriff auf das Interface des GBMboost.

Wenn der Simulations-Modus aktiv ist wird in der Toolbar ein „Werkzeugkasten“ angezeigt und die Verbindungsanzeige in der Statusbar ist ausgegraut und mit dem Hinweis „Simulation“ versehen.



Knoten-Baum

Im Knoten-Baum werden die Knoten am BiDiB-Bus angezeigt. Jeder Knoten wird normalerweise mit einem grünen Kreis dargestellt. Knoten die Subknoten haben können werden mit einem zusätzlichen +-Zeichen dargestellt.

Falls sich ein Knoten im Identifizierungs-Modus befindet, wird die Farbe des Kreis auf rot geändert. Der Identifizierungs-Modus kann entweder über das Kontext-Menü *Identifizieren* oder durch Drücken des Identifizierungs-Taster auf dem Knoten aktiviert/deaktiviert werden.

Das Knoten-Symbol mit einem Ausrufezeichen auf rotem Hintergrund wird angezeigt, falls ein Fehler vom Knoten geliefert wird. Wenn der Mauszeiger über das Symbol gefahren wird erscheint die Fehlermeldung als Tooltip. Über das Kontextmenü kann der Fehler quittiert werden.

Falls durch eine Konfigurationsanpassung ein Knoten-Neustart erforderlich wird, erscheint das Knoten-Symbol wie nachfolgend dargestellt.

Über das Kontext-Menü kann der Neustart ausgelöst werden.

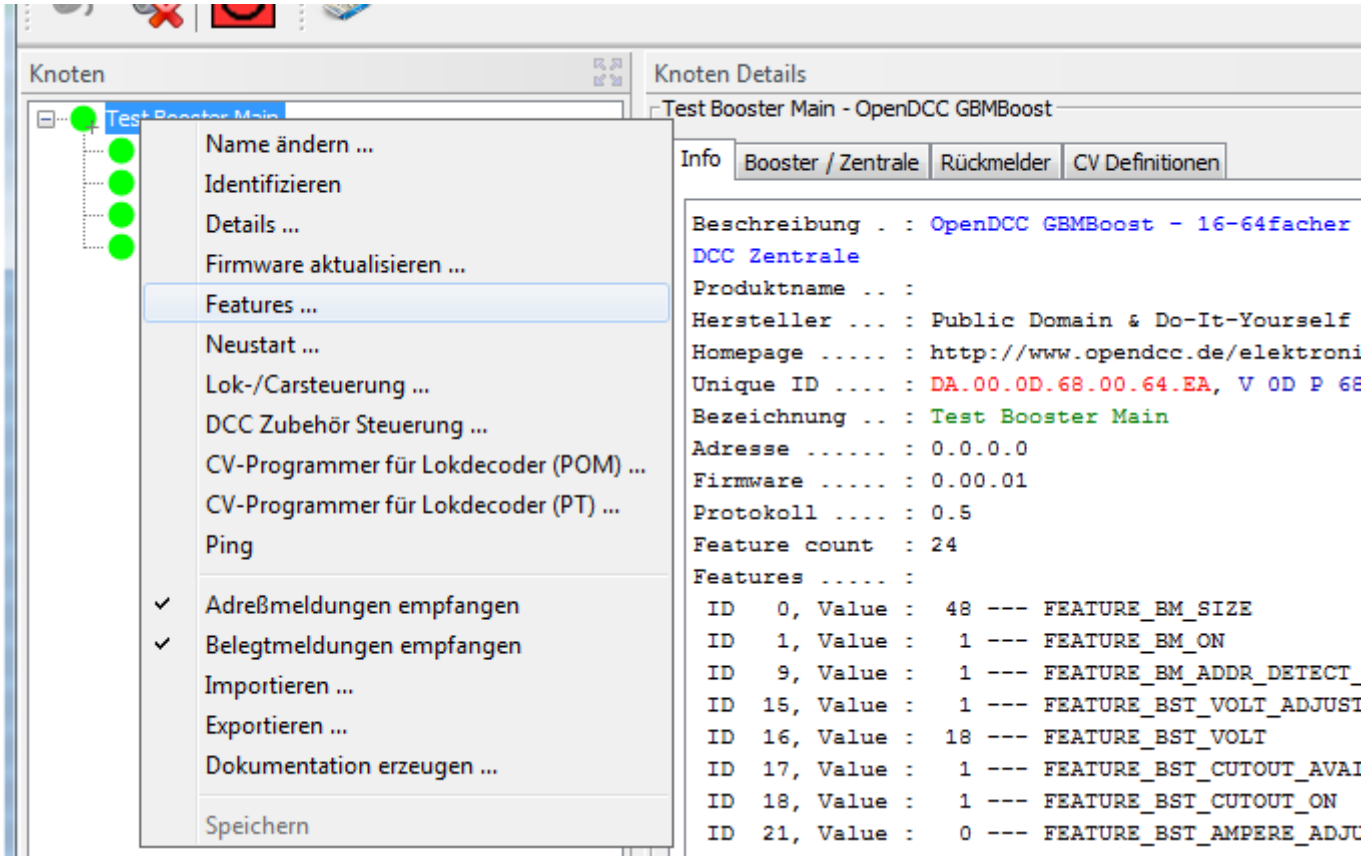
Anzeige Knotenname

Wenn man dem Knoten einen Namen gibt, wird diese Information lokal gespeichert (<userhome>\.bidib\labels) sowie auf den Knoten übertragen. Der Speicherplatz auf dem Knoten ist limitiert (i.A. 16-24 Zeichen) während lokal keine Begrenzung besteht. Wird eine Name eingegeben der länger ist als das was auf dem Knoten gespeichert werden kann, dann wird der Teil der zu lang ist im Wizard Knotenbaum blau dargestellt statt schwarz.

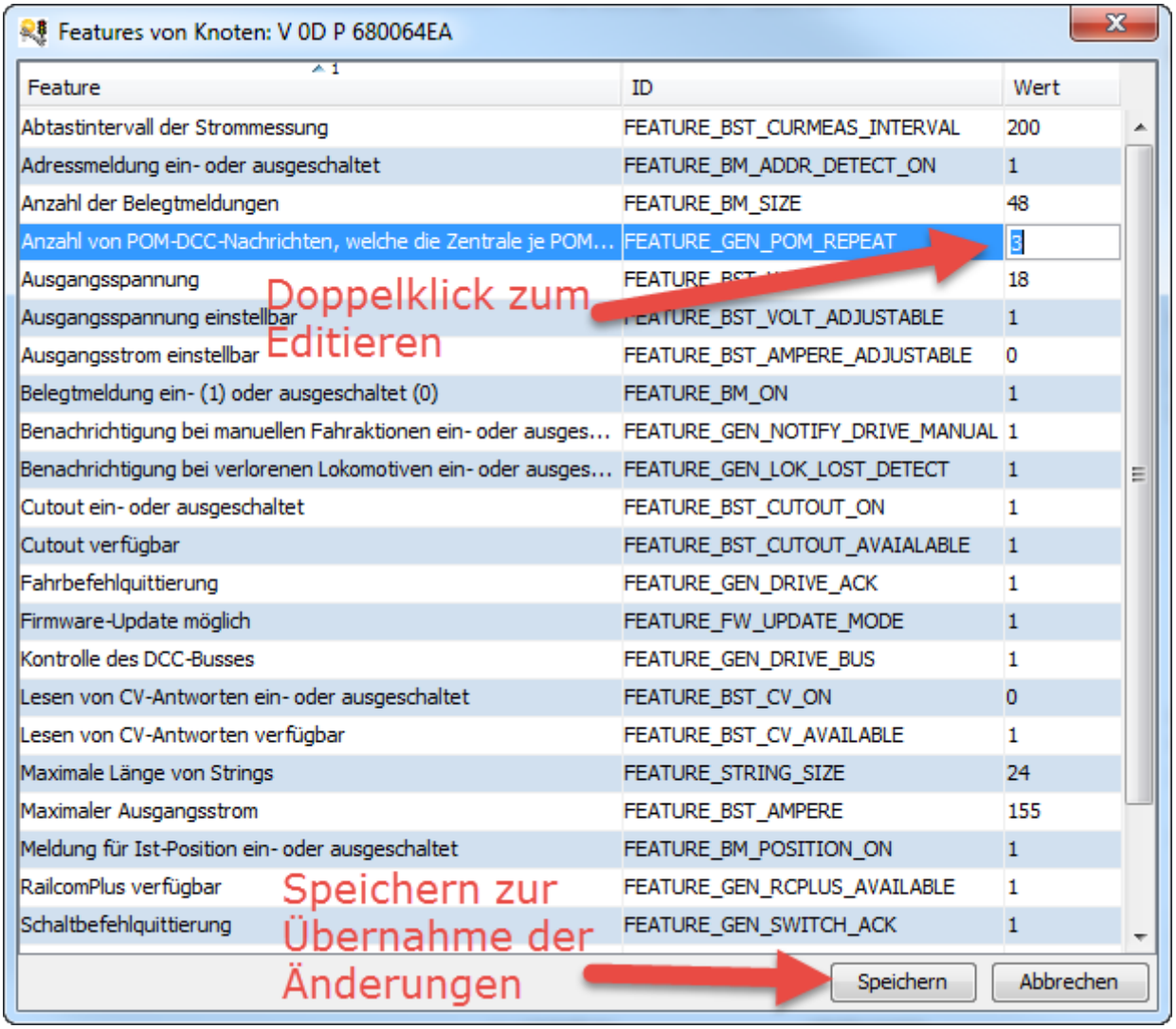
Falls auf dem Knoten und lokal kein Name gespeichert ist, wird die UniqueID des Knoten im Knotenbaum blau dargestellt.

Features ändern

Um Features auf einem Knoten zu ändern, muss der Knoten selektiert sein. Dann über das Kontextmenü auf dem Knoten den Menüpunkt **Features ...** anwählen. Im Dialog werden anschliessend alle Features des Knoten angezeigt.



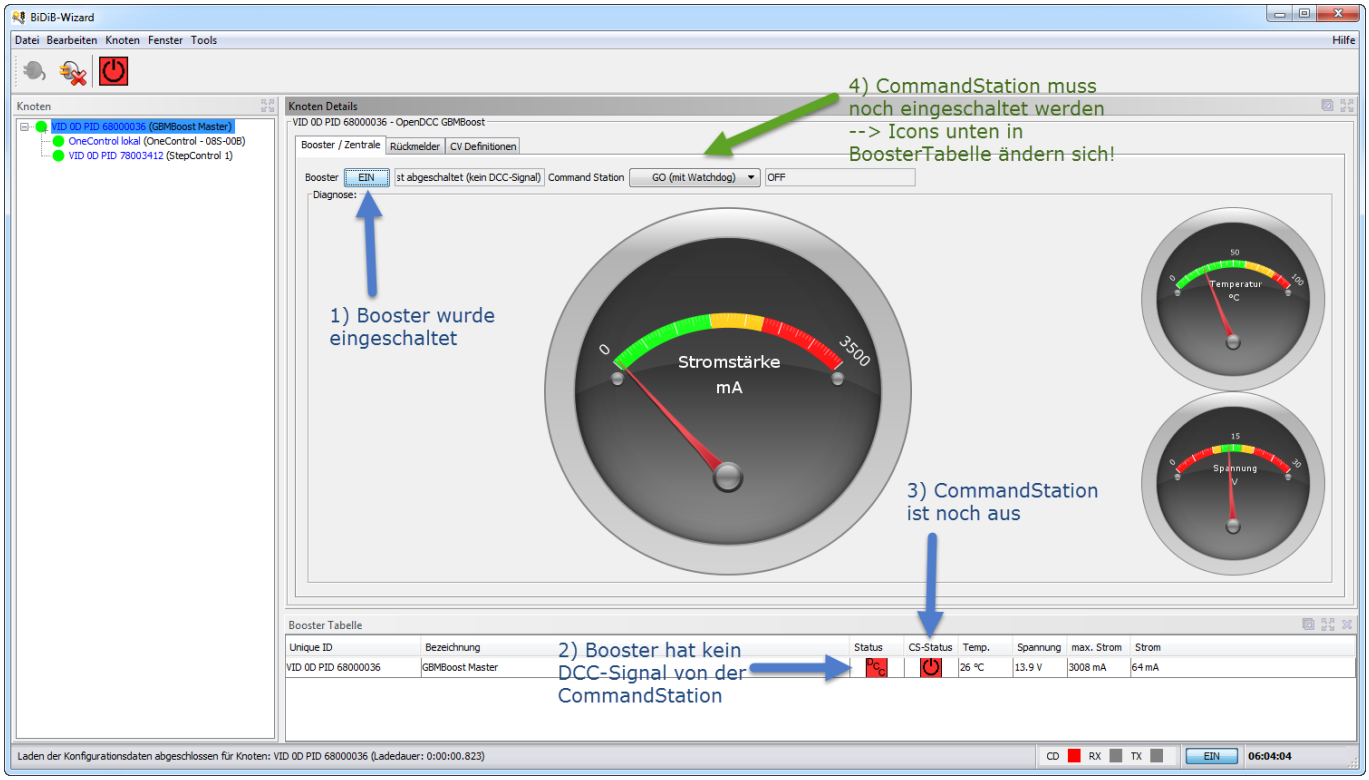
In der Spalte Wert kann der aktuelle Wert geändert werden. Durch Klick auf **Speichern** werden die angepassten Werte auf den Knoten übertragen.



Booster und Command Station

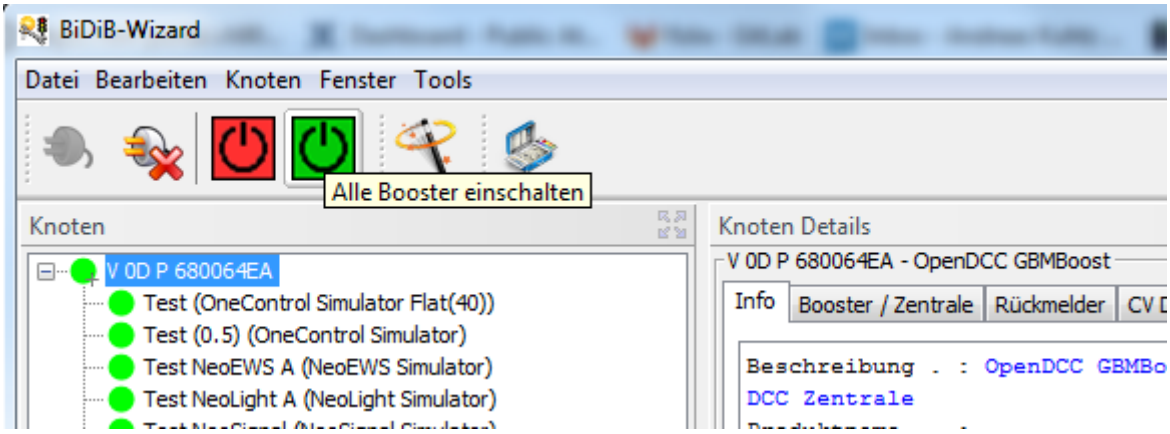
Ab Version 1.9 können der Booster und die CommandStation getrennt eingeschaltet werden. Nach dem Abschalten der Booster (z.B. über den Booster-Off-Button in der Toolbar) können die Booster über den Booster-Button *EIN* wieder einzeln eingeschaltet werden. Da aber die CommandStation (DCC-Generator) ebenfalls gestoppt wurde, schaltet sich der Booster nicht ein, weil das DCC-Signal fehlt.

Durch Klick auf den Command Station Button *GO (mit Watchdog)* wird die Command Station gestartet und dadurch auch der Booster (wenn das Feature gesetzt ist).



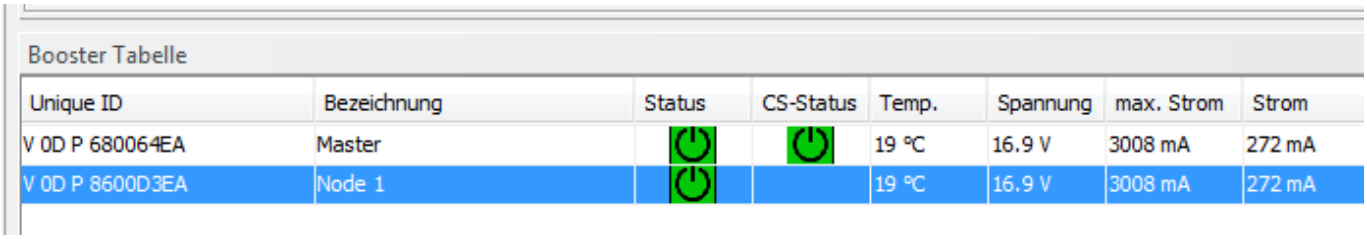
Booster Toolbar

Ab Version 1.10 ist die Booster-Toolbar erweitert um das Einschalten von der Zentrale und allen Boostern zu ermöglichen.



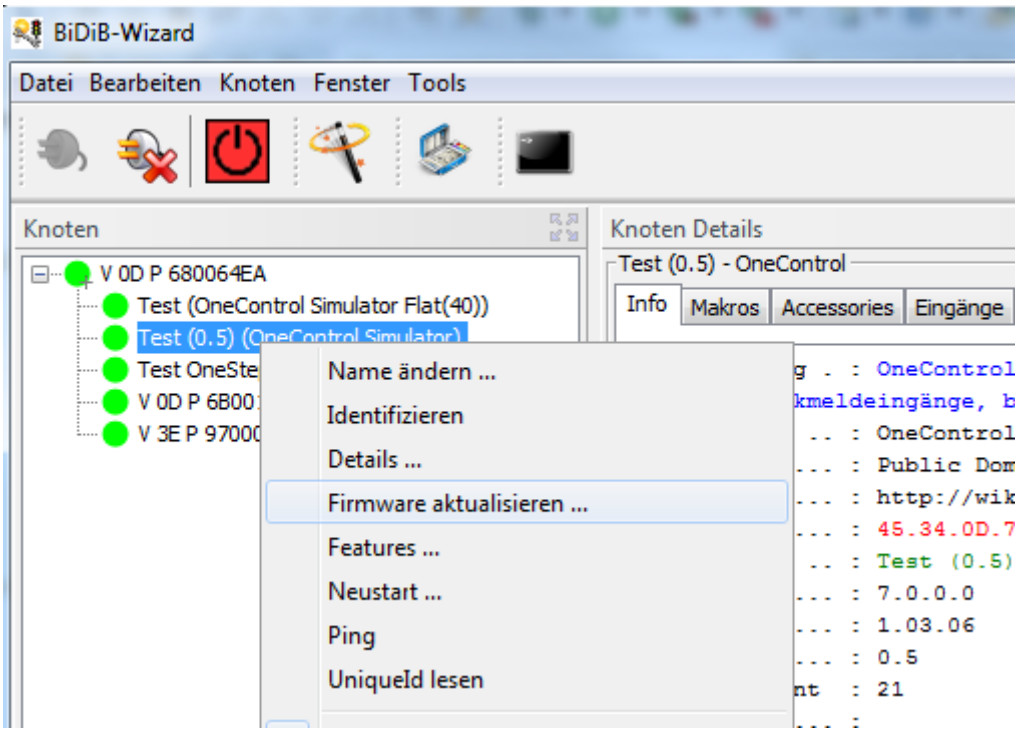
- Wahlweise den Start der Zentrale mit oder ohne Watchdog wählen.
- Wenn nur die Booster eingeschaltet werden sollen muss der *Weiter* Button gewählt werden.
- Wenn die Booster und die *Zentrale* eingeschaltet werden sollen, muss der *Weiter Booster & Zentrale* Button gewählt werden.

Anschliessend werden in der Booster-Tabelle die Zustände der Zentrale und Booster automatisch aktualisiert.



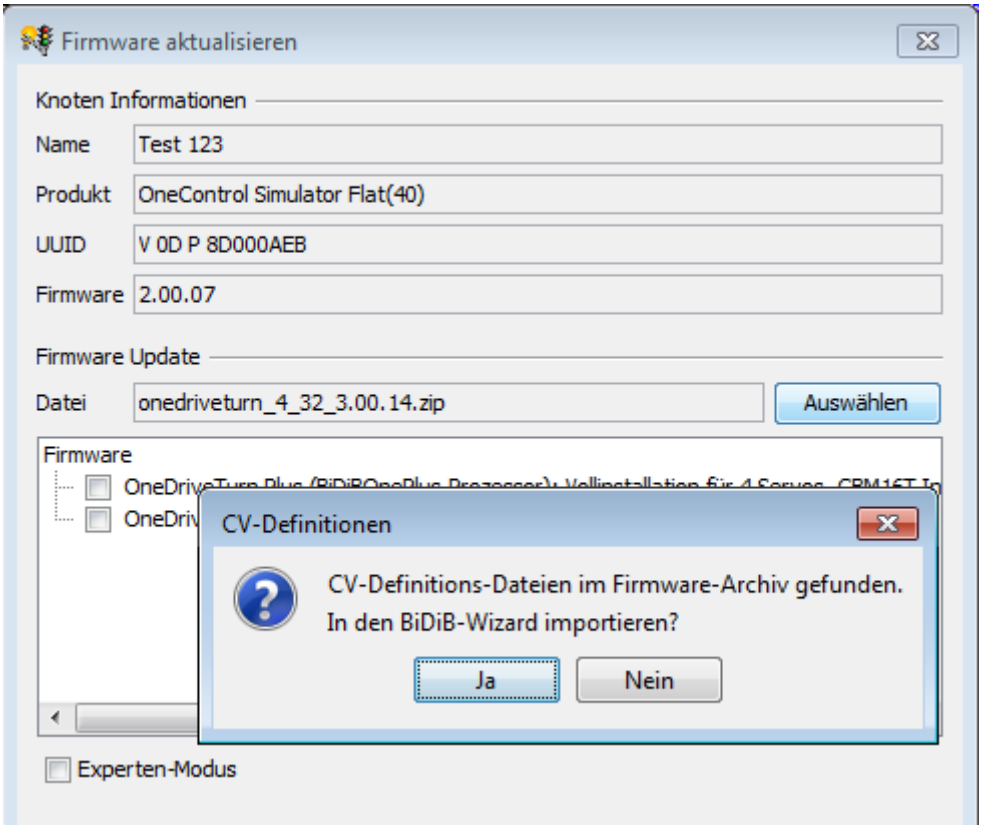
Firmware Update

Über das Kontextmenü des Knoten kann der Firmware-Update-Dialog geöffnet werden, sofern der Knoten dieses Feature unterstützt.

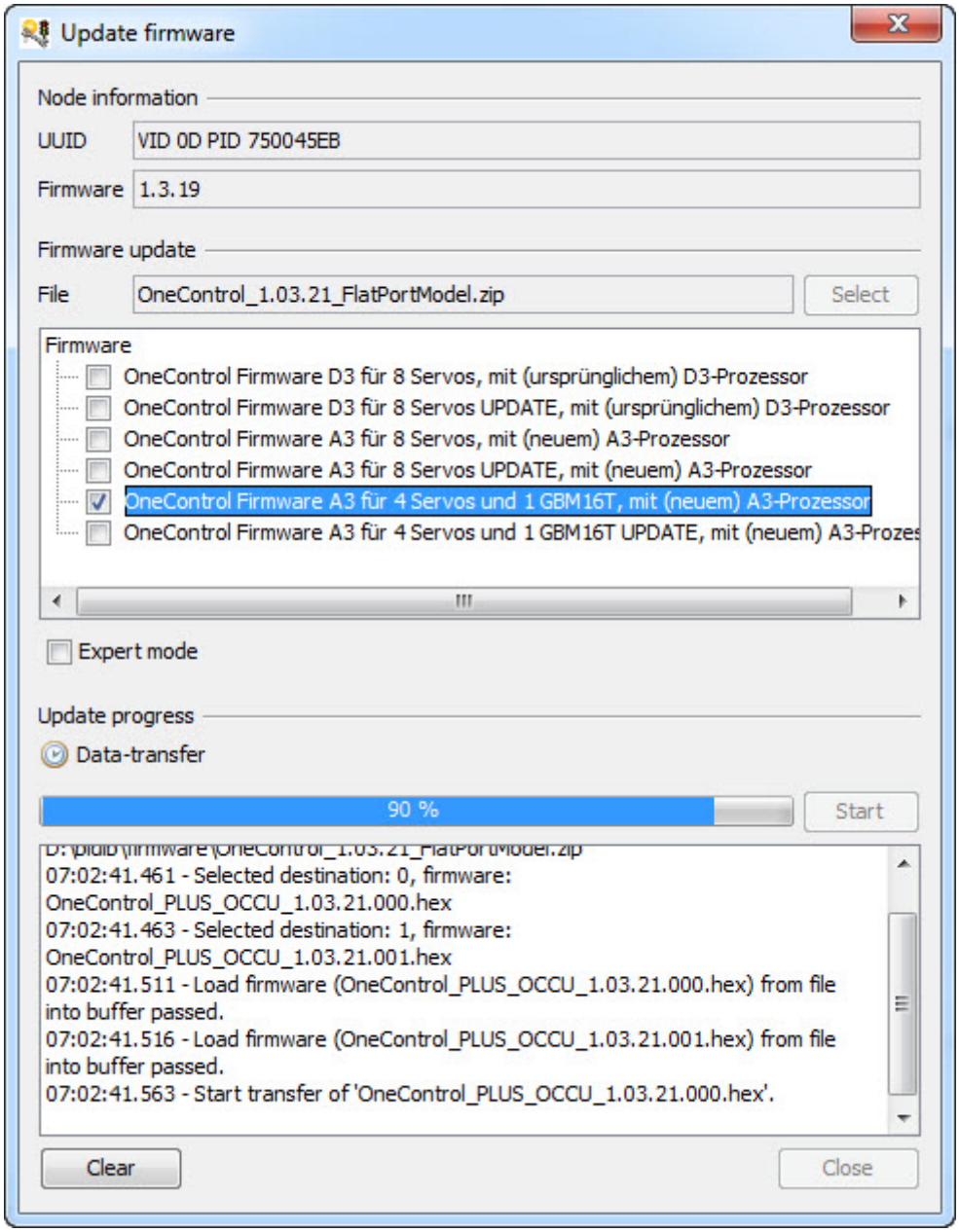


Im *Update firmware*-Dialog wird das Zip-File mit der Firmware für den Knoten ausgewählt. Anschliessend werden die verfügbaren Firmware-Pakete in einem Baum dargestellt.

Falls im Firmware-Zip neue CV-Definitionen vorhanden sind, wird der folgende Dialog angezeigt.



Aus den Paketen im Baum muss 1 Paket ausgewählt werden, welches anschliessend durch Klick auf den Button *Start* auf den Knoten übertragen wird.



Wenn der Firmware-Update beendet ist, wird der Dialog durch Klick auf *Schliessen* beendet und der Knoten wird neu gestartet.

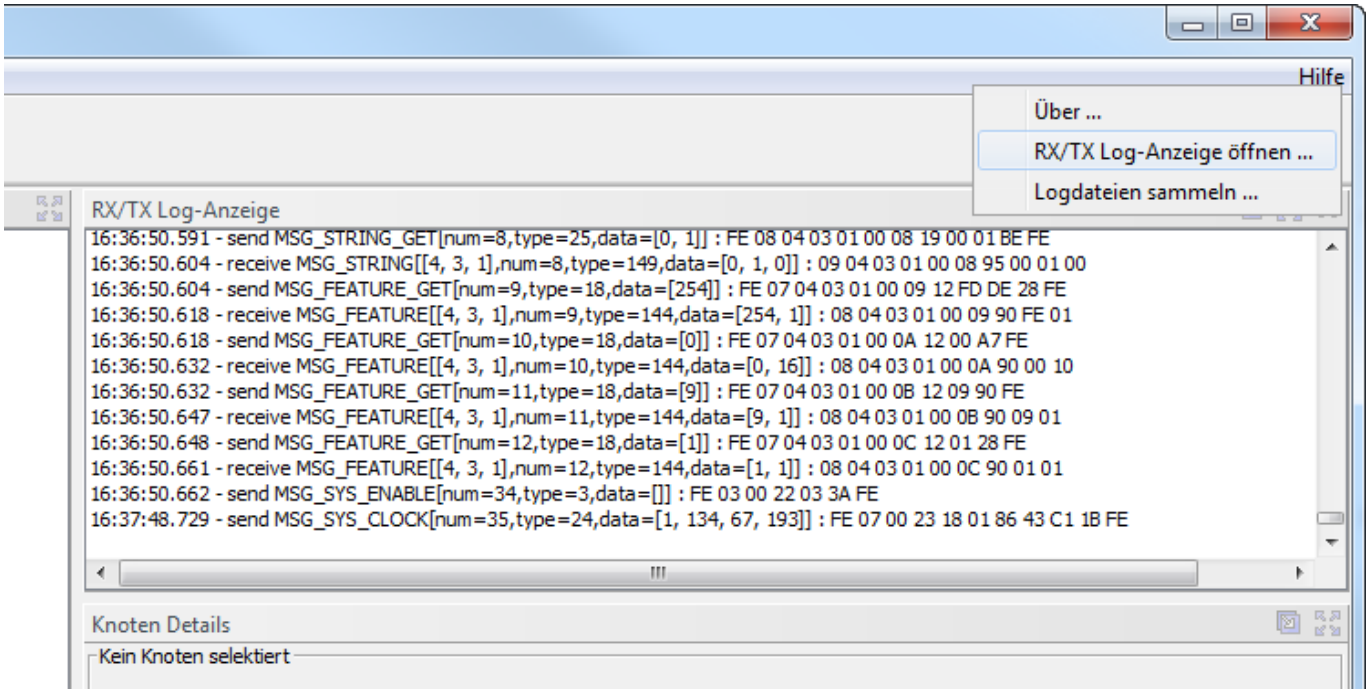
Falls ein Knoten nach dem Update nicht mehr reagiert, muss die Baugruppe mit folgenden Schritten mit dem Bootloader gestartet werden:

- Abstecken, Ausschalten.
- ID-Taste drücken und halten.
- Einschalten, jetzt erst loslassen.
- Anstecken.
- Die Baugruppe meldet sich als Bootloader. Jetzt kann man Firmware-Update machen.

—–

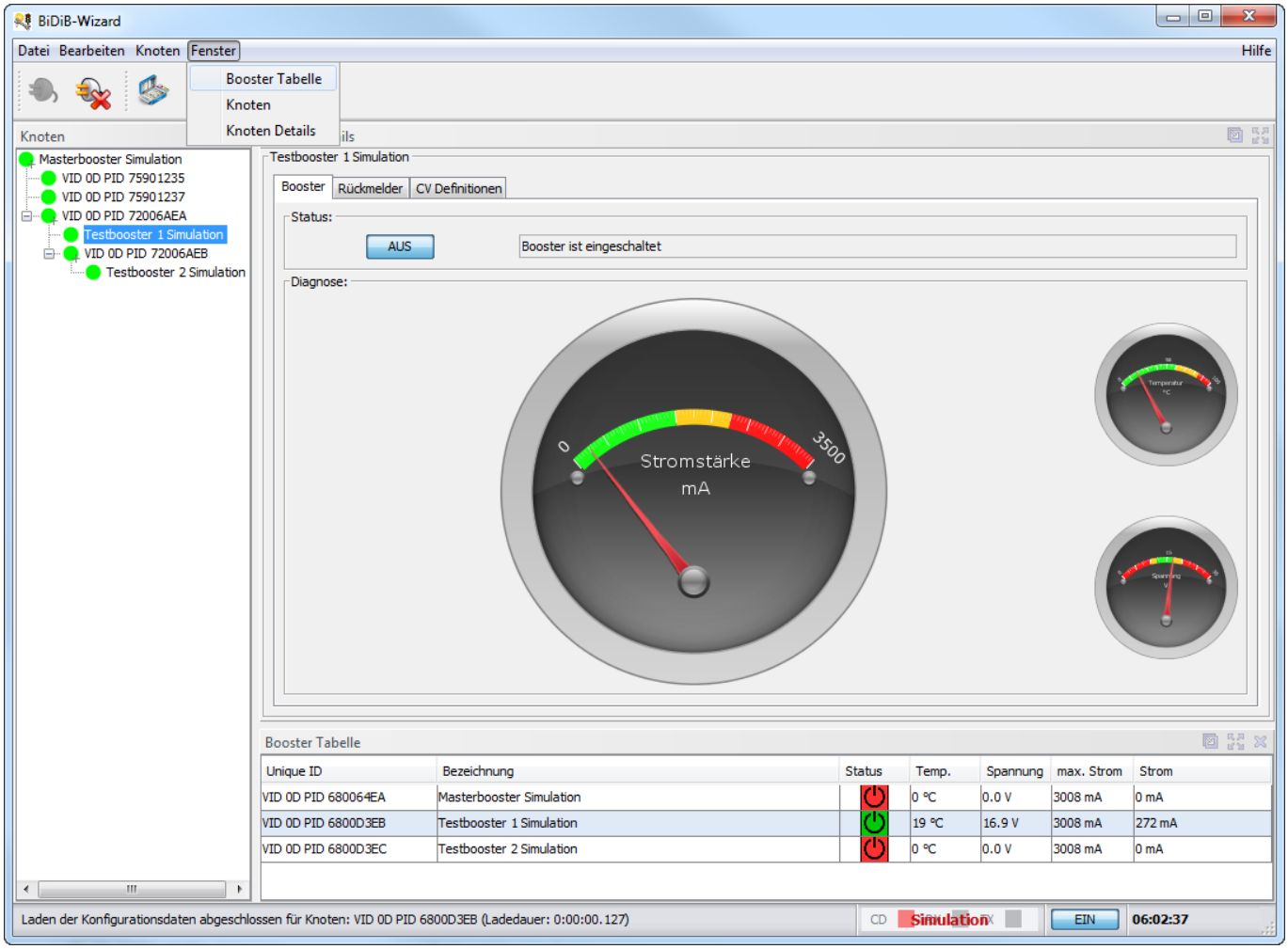
RX/TX-Log-Anzeige

Ab Version 1.7 gibt es einen integrierten Logviewer um den Meldungsverkehr zwischen dem BiDiB-Wizard und dem GBMBoost anzuzeigen. Über das Menü *Hilfe > RX/TX Log-Anzeige öffnen …* kann das Fenster geöffnet werden.



Booster Tabelle

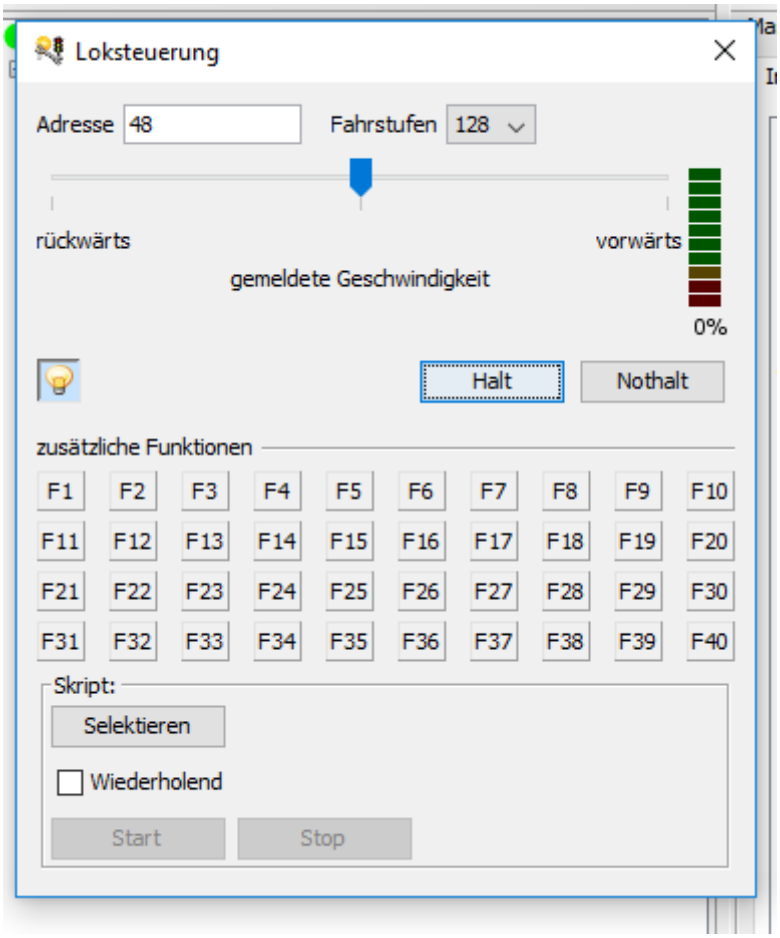
Ab Version 1.7 gibt es die Booster Tabelle welche den Zustand der Booster im BiDiB-System anzeigt. Über das Menü *Fenster > Booster Tabelle* kann das Fenster geöffnet werden. Die Booster Tabelle wird standardmässig nach dem Start eingeblendet. Dieses Verhalten kann über das Menü *Bearbeiten > Einstellungen > Checkbox „Booster-Tabelle standardmässig anzeigen“* deaktiviert werden.



Lok / Car steuern

Einzelne Loks oder Cars können über die *Loksteuerung* gesteuert werden, welche über das Kontextmenü auf dem Zentralen-Knoten (Command Station) geöffnet werden kann. Bei Steuerung von Cars muss die BiDiB-RF-Basis ausgewählt werden und dort über das Kontextmenü der Dialog geöffnet werden.

Nach der Eingabe der DCC-Adresse und der Auswahl der Fahrstufen kann die Geschwindigkeit über den Slider eingestellt werden. Eine zurückgemeldete Geschwindigkeit (Speed-Meldung vom Dekoder) wird unterhalb des Slider angezeigt.



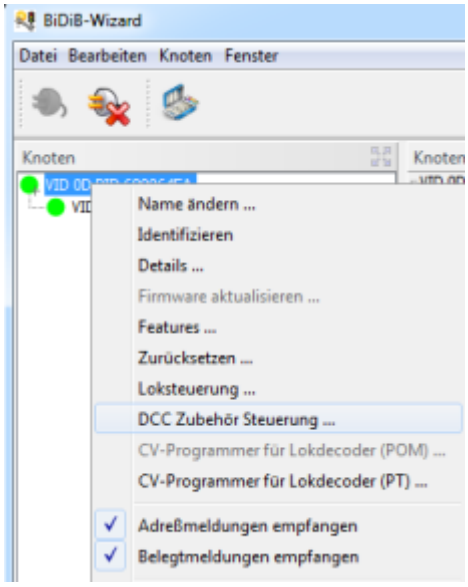
Falls ein Dekoder einen Wert für „Behälter 1“ mit der DYN_STATE-Meldung zurückliefert, wird dieser Wert in der Balkenanzeige am rechten Rand angezeigt.

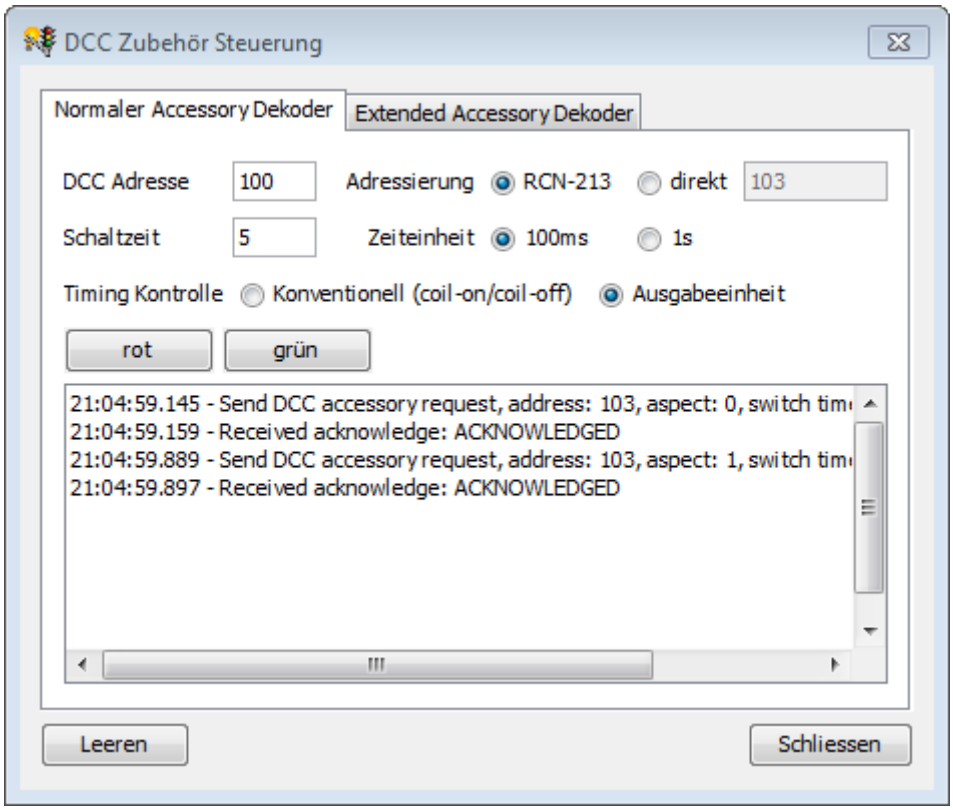
Ab **Wizard-1.11 Build 3473**:
Der Geschwindigkeitsregler unterstützt folgende Tasten-Aktionen wenn er selektiert ist:

- `Leertaste, Space`: Stop
- `Cursor-rechts/links`: Fahrstufe erhöhen/reduzieren
- PageUp/PageDown: Fahrstufe in grösseren Schritten erhöhen/reduzieren
- `Pos1/Ende`: max. Fahrstufe

DCC-Accessory-Decoder

Der BiDiB-Wizard bietet ab Version 1.7 Unterstützung für das Schalten von DCC-Accessory-Decodern (Schaltdecoder, z.B. OpenDecoder2) oder der Kehrschleifenmoduls des GBM16T an.





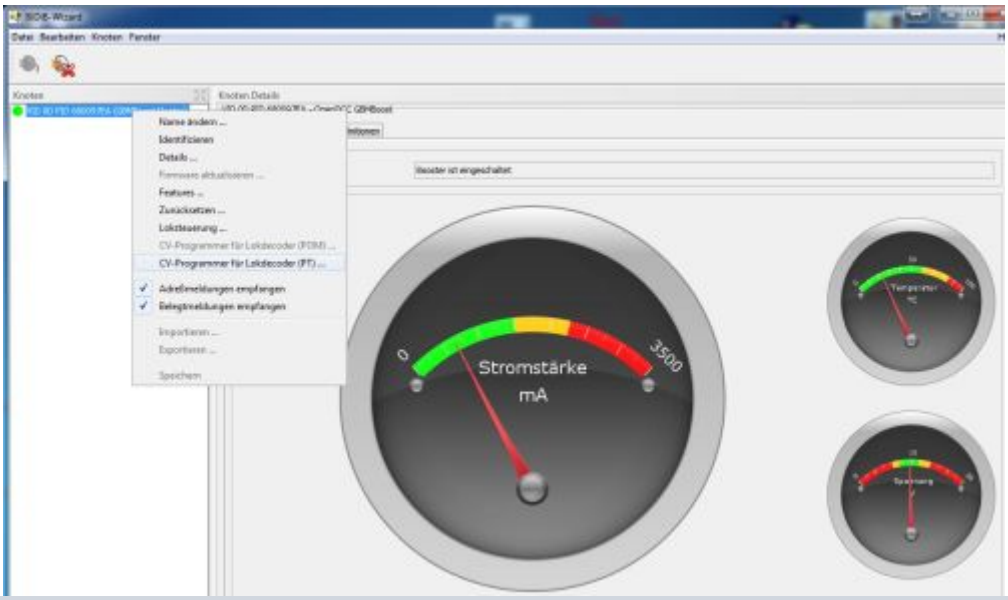
Die Adresse welche mit der Multimaus geschaltet wird ist um +1 verschoben zur Adresse, welche mit dem Wizard benutzt wird, z.B. Wizard/Monitor: 200 –> Multimaus 201

Der CV-Programmierer

Mit Hilfe es BiDiB-Wizard kann man über das Programmiergleis am GBMboost die DCC-Decoder auslesen und editieren. Eine ausführliche Erklärung zu dem Programmiergleis und dessen Warnhinweise finden Sie unter dem Thema "GBM - Programmiergleis".

Klicken Sie auf den GBMboost Master in der Knotenliste und öffnen Sie mit einem rechtsklick den CV-

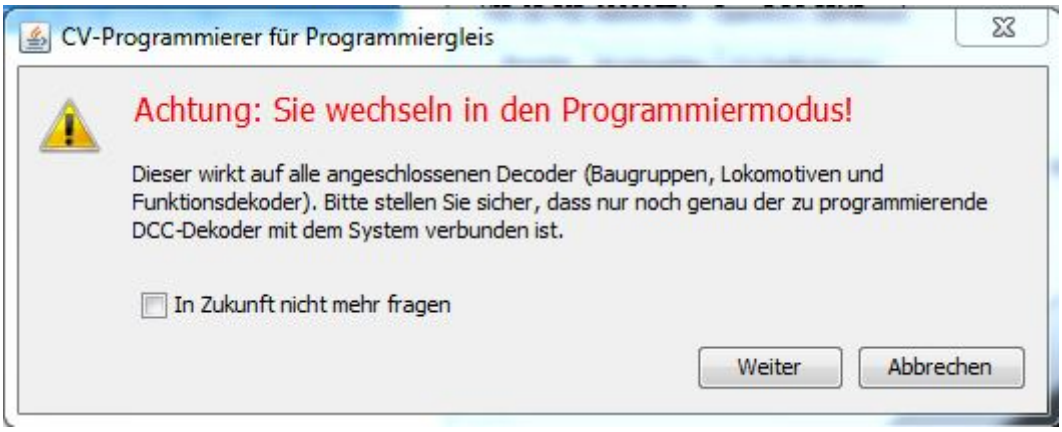
Programmer für Lokdecoder (PT).

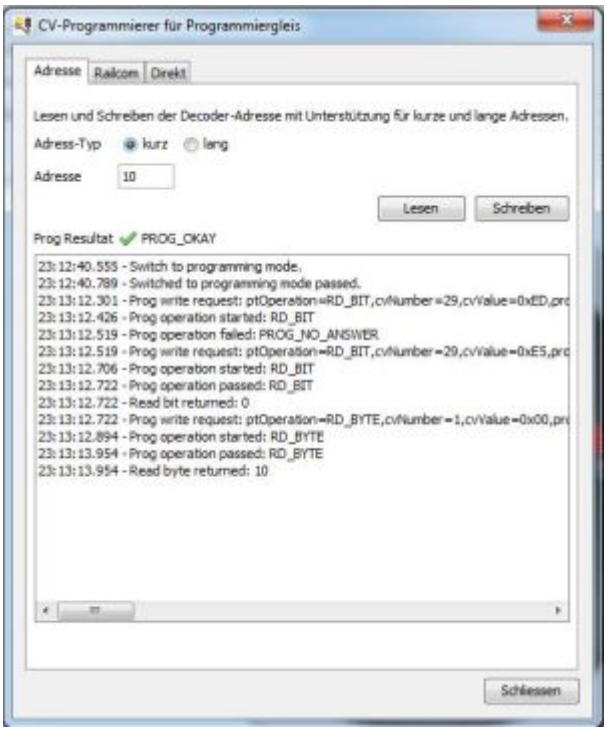


Mit Hilfe eines Warnhinweises werden Sie darauf aufmerksam gemacht, dass mit dem GBMboost Master nur die programmierende Lok verbunden sein solle. Alle weiteren DCC-Decoder, müssen abgesteckt bzw. entfernt werden. Das betrifft auch die verbundenen BiDiB-Knoten die über den BiDiBus mit dem GBMboost in Verbindung stehen. Das kann durch das abstecken des Buskabels vom GBMboost Master erfolgen.

Über das Programmiergleis werden bei nicht entfernten Loks bzw. Baugruppen, auch dessen CV-Werte verstellt.

Die Programmiergleis-Funktion kann hier nicht unterscheiden!





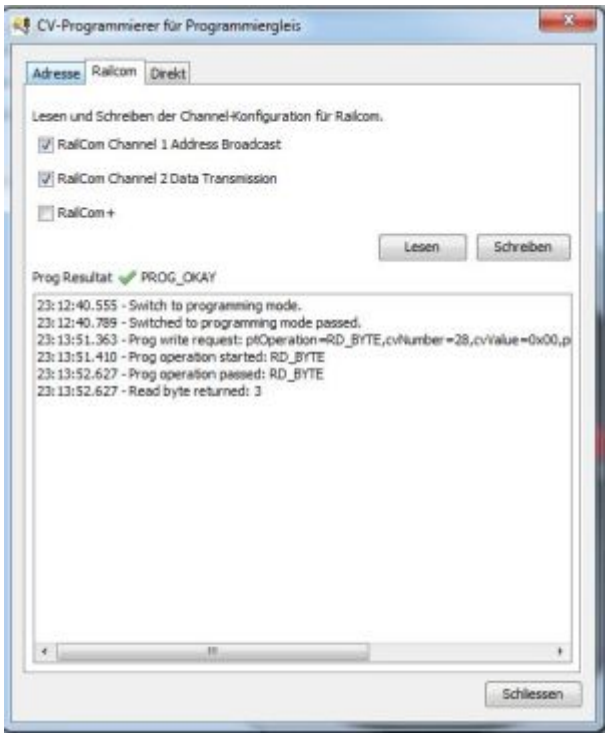
Nach dem Bestätigen und entfernen der anderen Decoder und Baugruppen, öffnet sich das Tool für die Programmiergleis-Funktion.

Im Fenster „Adresse“ kann man die Decoder Adresse „kurz“ und „lang“ von dem Decoder auslesen bzw. editieren und zurückschreiben. Beim Lesen wird automatisch erkannt ob es sich um eine kurze oder lange Adresse handelt.

Bei dem aufgeführten Beispiel wurde von dem Decoder der Defaultwert 3 ausgelesen und auf den neuen Wert 10 geändert.

Ab jetzt reagiert die Lokomotive auf die Adresse 10.

Der erfolgreiche Lese bzw. Schreibvorgang wird durch einen grünen Hacken (Prog_Okay) angezeigt.



Ein schnelles editieren der wichtigsten Railcom-Einstellungen eines Lokdecoders ermöglicht das Fenster „Railcom“.

Für den Anwender sind hier die Einstellungen des CV28 benutzerfreundlich aufgelistet.

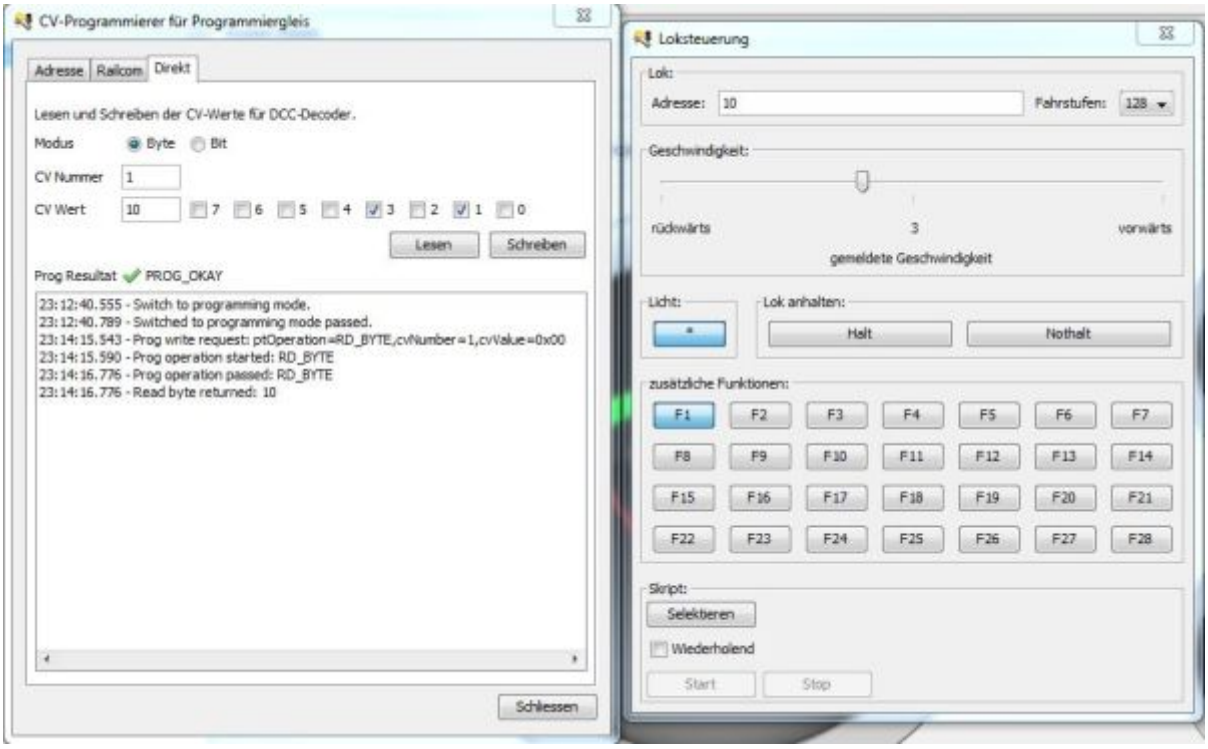
Mit setzten der Kästchen kann man die einzelnen Railcom-Channels 1 und 2 aktivieren bzw. deaktivieren. Die Auswahl Railcom+ ist eine Zusatzoption für Railcom-Plus kompatible Lokdecoder.

Weitere Informationen zu dem Thema RailCom® finden Sie im Wiki unter dem Thema "RailCom®".

Im Fenster **„Direkt“** besteht die Möglichkeit das Lesen und Schreiben aller möglichen CVs des DCC-Decoders. Dafür verwenden Sie als Hilfe die CV-Liste Ihres Decoders und editieren das betroffene CV.

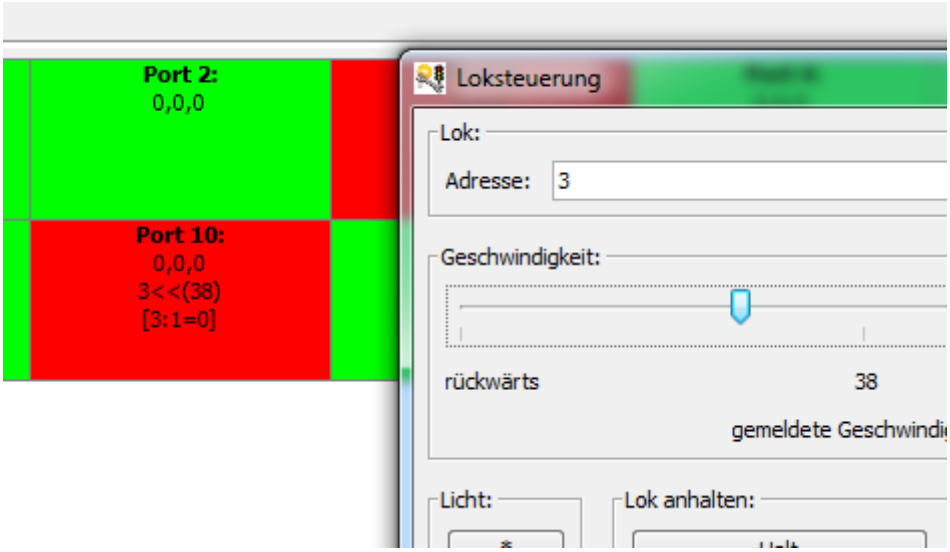
Das Tool ermöglich die Byte und Bit Schreibweise und ist somit kompatible zu den unterschiedlichen CV-Beschreibungen der jeweiligen DCC-Decoder.

Um einen CV-Wert zu lesen oder schreiben, muss die CV-Nummer eingegeben werden. Anschliessend wird der *Lesen*-Button freigeschaltet. Sobald ein Wert im Eingabefeld CV-Wert eingegeben wird oder die Checkboxen geklickt werden, wird auch der *Schreiben*-Button freigeschaltet. Falls ein ungültiger Wert eingegeben wird, erscheint ein Fehler-Marker (kleines Icon mit weissem Kreuz auf rotem Grund). Der Grund des Fehlers wird als Tooltip beim Positionieren des Mauszeiger über diesem Fehler-Marker dargestellt.



Im BiDiB-Wizard können gleichzeitig mehrere Fenster geöffnet werden. In der Abbildung ist das Fenster **CV-Programmer** und das Fenster **Loksteuerung** geöffnet. Diese doppelte Anzeige ermöglicht das Editieren eines CV-Wertes und das gleichzeitige Testen diese Änderung über die Loksteuerung.

Belegtmelder



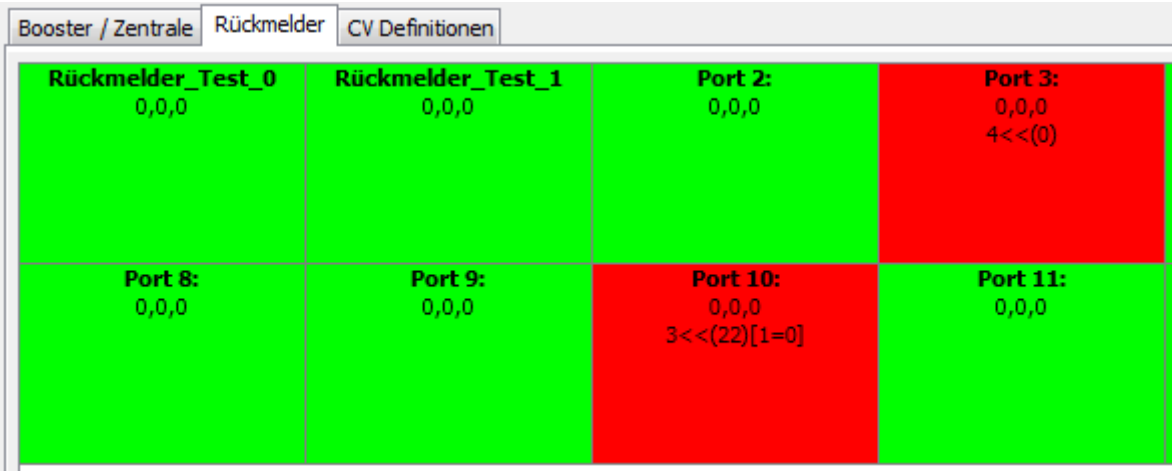
Im Bild oben ist der Decoder 3 im Prüfstand angeschlossen, darum kommt immer 1=0 (also keine fehlerhaften Pakete) aber es kann dort auch ein anderer Wert kommen wenn man auf der Anlage fährt.

In den Belegtmeldern wird:

- in der ersten Zeile ist der Inhalt der Confidence-Meldung angezeigt.

- in der 2. Zeile wird die Lokadresse, Aufgleisrichtung (« oder ») und die Geschwindigkeit angezeigt.
- in der 3. Zeile wird QOS (MSG_BM_DYN_STATE –> 1: Signalqualität) angezeigt wenn der Decoder es unterstützt.

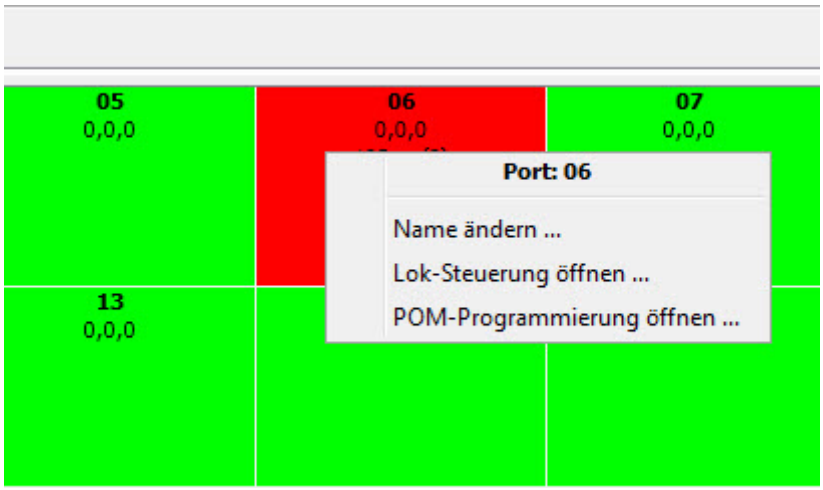
Die DynStatus-Daten (Verschmutzung) werden jetzt in der gleichen Zeile wie die Adresse, Aufgleisrichtung und Geschwindigkeit dargestellt. Von der Höhe her sollten auch 4 Decoderdaten passen.



In den Belegtmeldern wird folgendes angezeigt:

- in der ersten Zeile ist der Inhalt der Confidence-Meldung (valid, freeze, signal).
- in der 2. Zeile wird die Lokadresse, Aufgleisrichtung (« oder ») und die Geschwindigkeit, dahinter die QOS (MSG_BM_DYN_STATE –> 1: Signalqualität) falls der Decoder diese Information liefert.
- in der ersten Zeile ist der Inhalt der Confidence-Meldung (VOID, FREEZE, NOSIGNAL).

Ab **Wizard-1.11 Build 3473**:
In den Belegtmeldern kann über das Kontextmenü der Lok-Dialog und der POM-Editor geöffnet werden. Dabei wird die erste Lokadresse als Defaultwert verwendet.



CV-Programmierer für POM

Mit dem CV-Programmierer für POM können Lok-Decoder über POM-Befehle (Programming-On-Main, Hauptgleisprogrammierung) gelesen oder beschrieben werden, sofern der Decoder dies unterstützt.

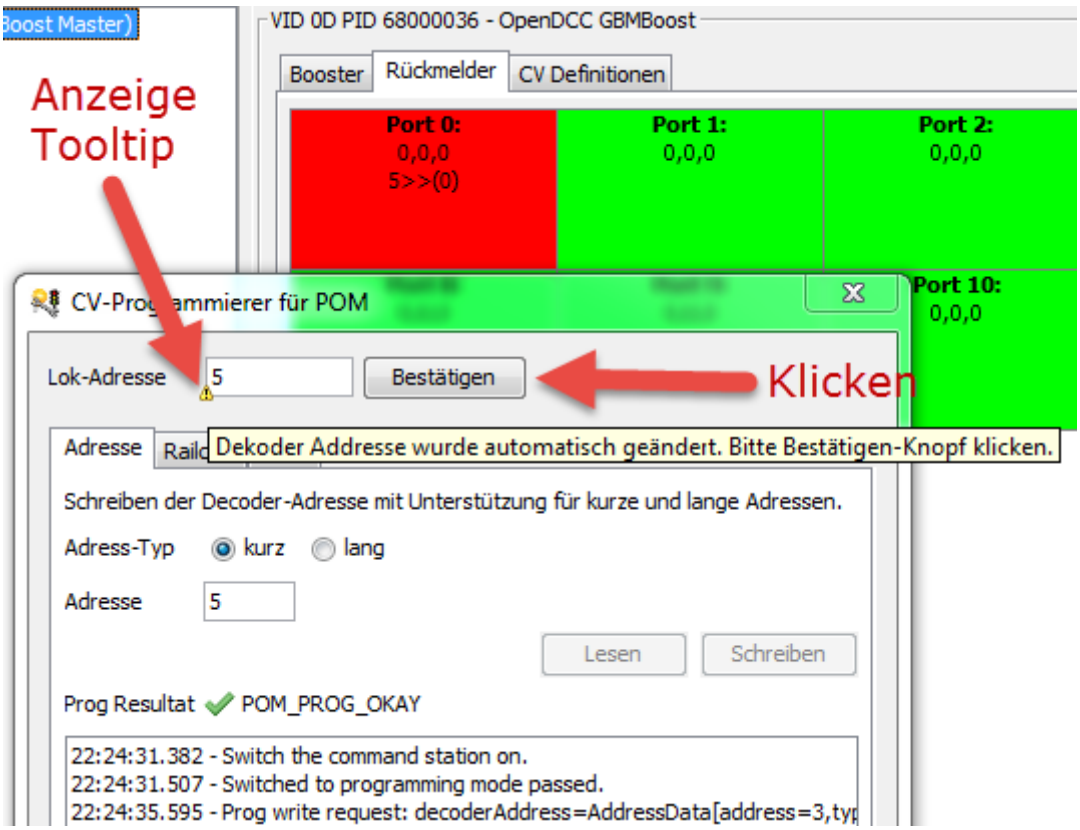
Adresse ändern

Um die Lok-Adresse zu ändern muss die Adresse der Lok im Feld Lok-Adresse eingegeben werden. Dann sollte zuerst versucht werden die Adresse zu lesen. Eine erfolgreich gelesene Adresse wird im Feld Adresse angezeigt.

Schreiben der neuen Adresse

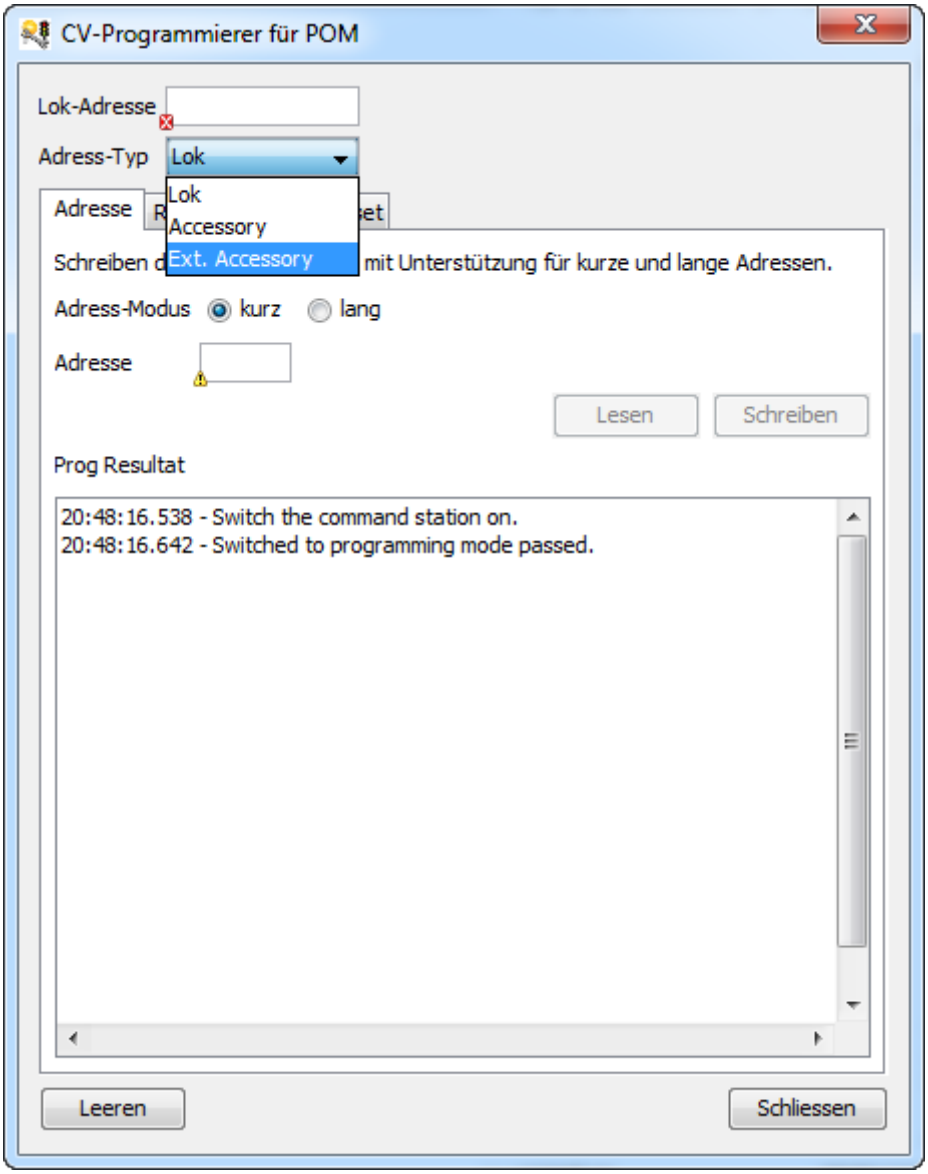
Eine neue Adresse wird durch Eingabe der neuen Adresse im Feld *Adresse* und anschliessendem Klick auf den *Schreiben*-Button über POM in den Decoder geschrieben. Die Anwort des Dekoder wird entweder nicht gesendet oder erfolgt evtl. auf die neue Adresse. Dies führt zu einem Timeout im Wizard.

Der Wizard erkennt dies und versucht den Lok-Decoder auf der neuen Adresse anzusprechen, allerdings nur falls die *kurze Adresse* verändert wurde. Ist dies erfolgreich, wird die neue Adresse als Lok-Adresse eingetragen. Diese automatische Anpassung muss der Anwender durch klicken des *Bestätigen*-Button akzeptieren um weitere Schreib- oder Lese-Aktionen ausführen zu können. Nach dem Bestätigen wird der Button wieder unsichtbar.



POM auf DCC-Schaltdekoder (DCC Accessory Decoder)

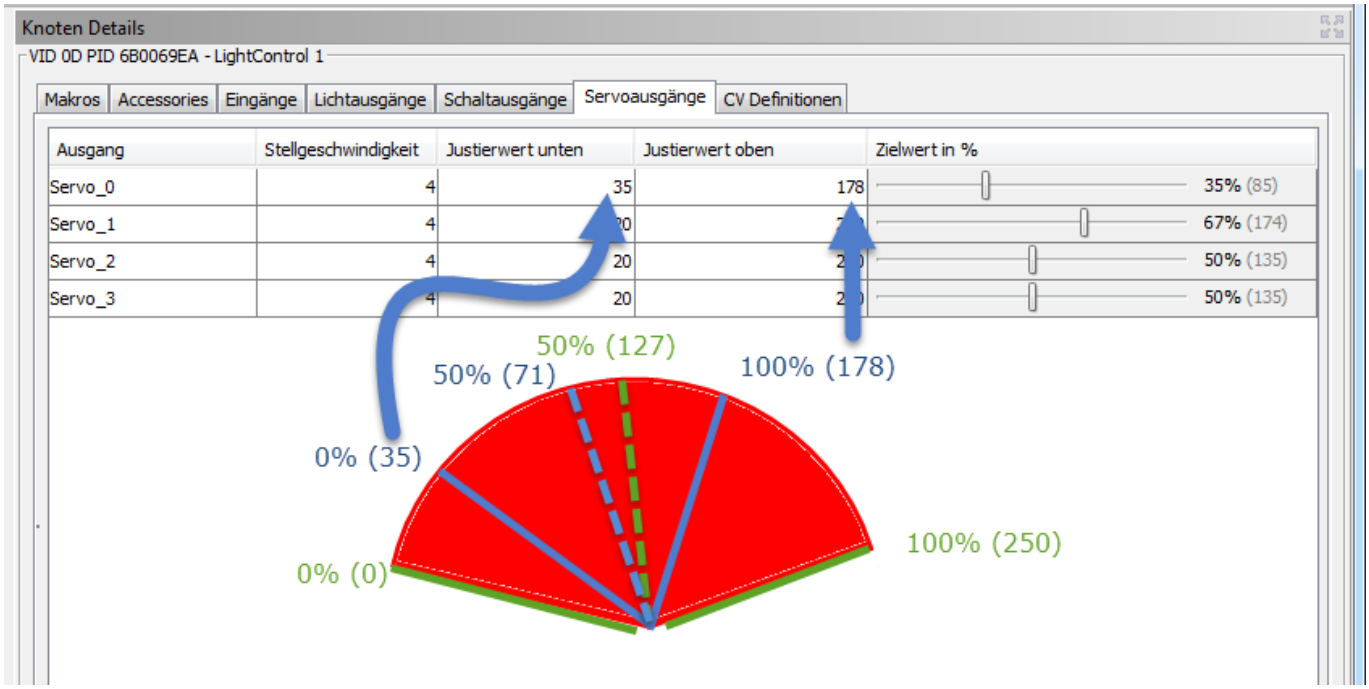
Um einen DCC-Schaltdekoder per POM zu programmieren muss der Address-Typ auf *Accessory* gestellt werden.



Um einen DCC-Schaltdekoder mit POM zu programmieren muss die Firmware des GBM16T erweitert werden. Deshalb muss diese Funktion in den Wizard-Einstellungen durch die PowerUser Checkbox freigeschalten werden.

Servo einstellen

Justierwerte einstellen



Ein Servo hat einen Stellbereich (der rote Bereich). Dieser Bereich ist der maximale Bereich der mit der LC angesteuert werden kann. Dabei ist der Justierwert unten = 0 und der Justierwert oben = 250.

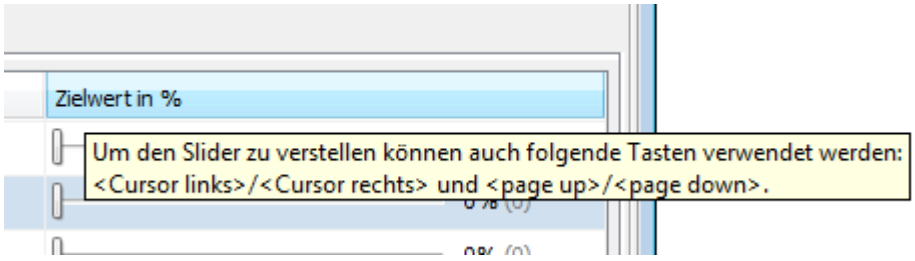
Über den Slider in der Spalte *Zielwert in %* kann man den Bereich abfahren (0% ist jetzt 0 und 100% ist jetzt 250). Der Zielwert 50% ist in diesem Fall bei der absoluten Position 127 (250 - 0 / 2, die grün gestrichelte Linie).

Wenn man den gesammten Stellbereich einschränken will, dann muss man zuerst die beiden Justierwerte ermitteln zwischen denen der Servo sich bewegen darf. Wenn man den unteren Justierwert auf 35 setzt, dann ist der untere Anschlag neu bei der blauen Linie. Neu ist also 0% bei dem Wert 35, darunter kommt man nicht mehr. Der Servo kann also nur noch zwischen 35 (entspricht 0%) und 250 (entspricht 100%) bewegt werden. Der Zielwert 50% ist neu bei der absoluten Position 107 (250 - 35 / 2).

Wird jetzt noch der obere Justierwert auf den Wert 178 gestellt, dann bewegt sich der Servo nur noch zwischen den absoluten Positionen 35 und 178 (d.h. 100% ist neu bei 178). Der Zielwert 50% ist neu bei der absoluten Position 71 (178 - 35 / 2, die blau gestrichelte Linie).

Servoposition einstellen

Um die Servo-Position einzustellen, können neben der Maus auch die Tasten `Cursor-links` und `Cursor-rechts` verwendet werden. Um direkt an den unteren oder oberen Wert zu fahren, können die Tasten `Page-Down` und `Page-Up` verwendet werden. Ein entsprechender Tooltip als Hilfe wird über der Spalten-Bezeichnung angezeigt.



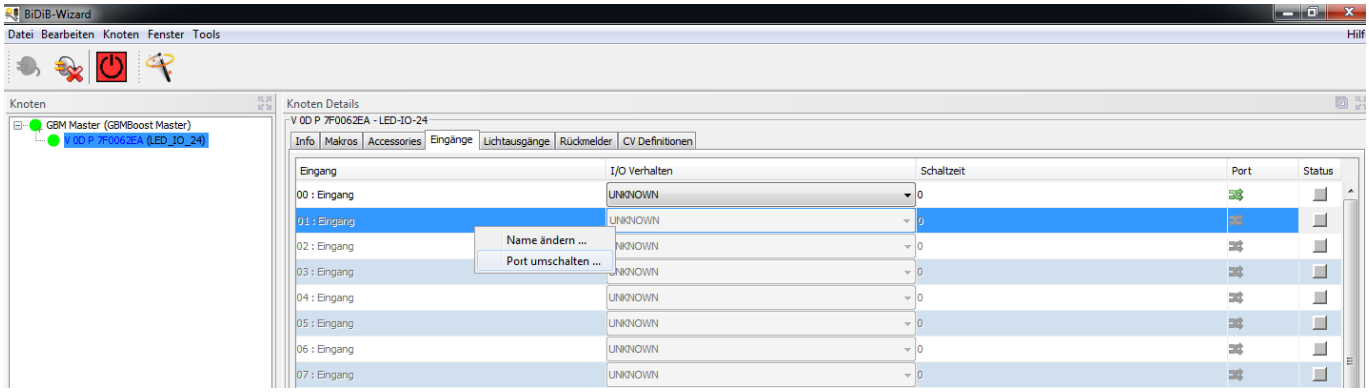
Portumschaltung

Auf der Baugruppe `LED-IO-24` kann jeder einzelne der 24 Ports von LED Ausgang auf Eingang umgeschaltet werden. Auf der Baugruppe `OneControl` können die GPIO-Ports zwischen Eingang und Schaltport umgeschaltet werden.

Die Umschaltung erfolgt mit Hilfe der BiDiB-Tools z.B. BiDiB-Wizard.

LED Ausgang -> Eingang

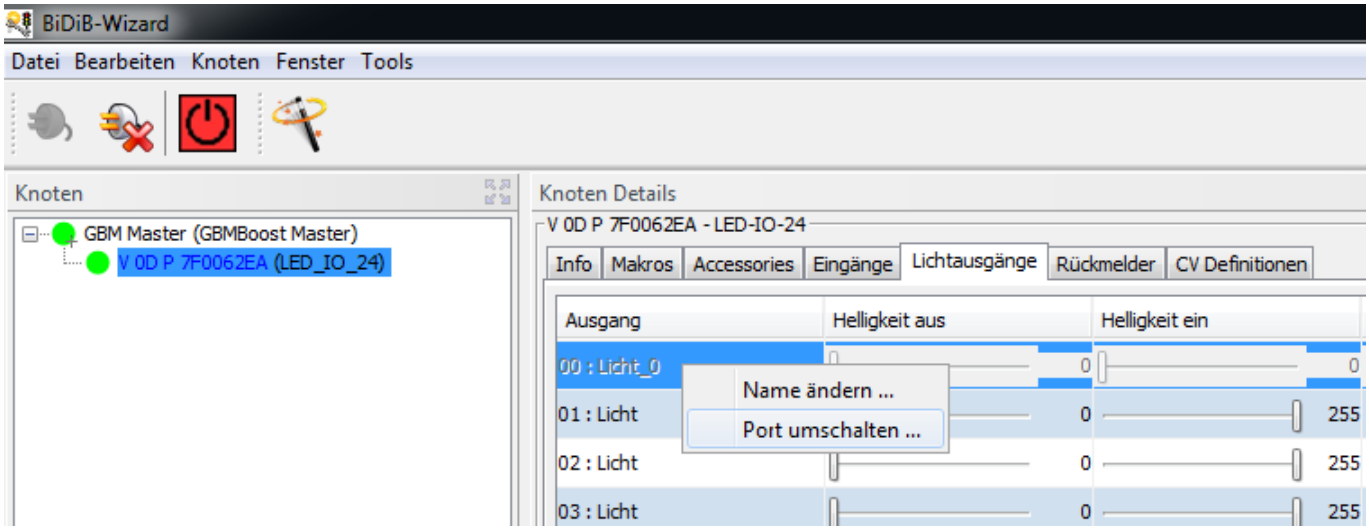
Wenn der Port aktuell als LED Ausgang aktiv ist im BiDiB Wizard auf den Reiter Eingang wechseln und beim gewünschten Port rechts klicken. Im Popup Menü auf „Port umschalten“ klicken und die Umschaltung bestätigen.



Ein auf Eingang umgestellter Port kann zusätzlich über die CV 171 - 173 von Eingang auf Belegtmelder umgestellt werden.

Eingang -> LED Ausgang

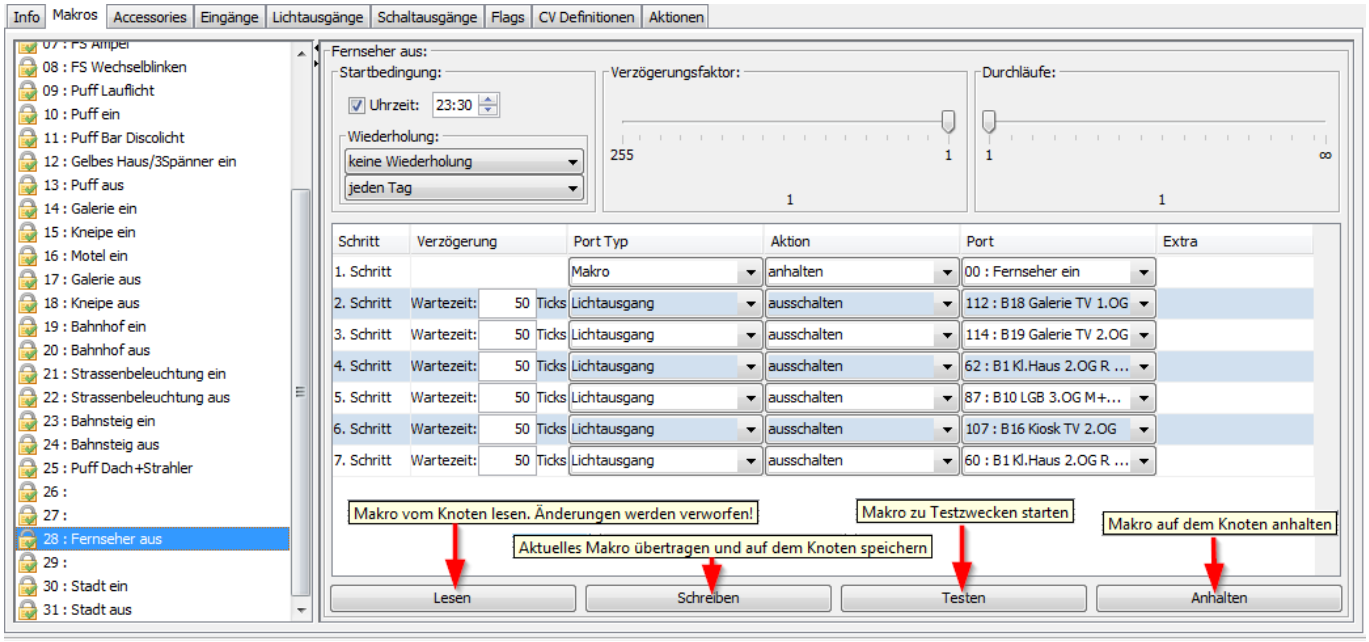
Wenn der Port aktuell als Eingang aktiv ist im BiDiB Wizard auf den Reiter Lichtausgänge wechseln und beim gewünschten Port rechts klicken. Im Popup Menü auf „Port umschalten“ klicken und die Umschaltung bestätigen.



Makro erstellen

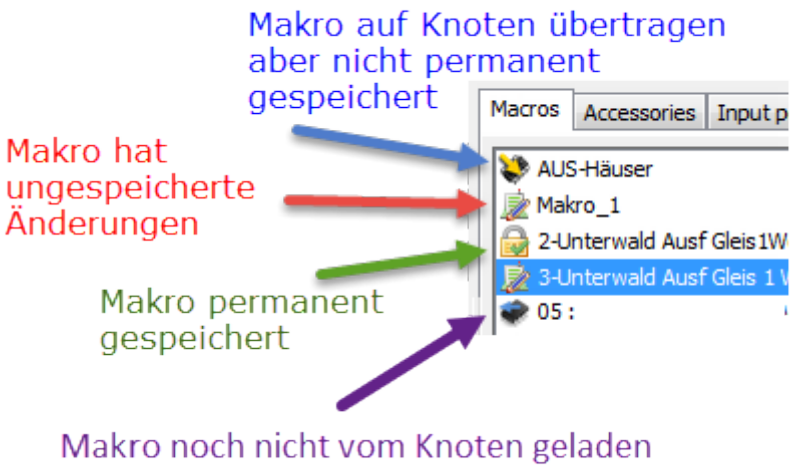
Um ein Makro zu erstellen muss der Knoten selektiert sein. Der Name des Makro kann durch Doppelklick in der Makroliste oder über das Kontextmenü geändert werden.

Wenn ein Makro selektiert ist, werden im rechten Teil des Fensters die Schritte angezeigt. Mit Hilfe des Kontextmenü können Schritte eingefügt oder gelöscht werden. Die nachfolgende Abbildung zeigt den Makroeditor:



Ein Makro muss auf den Knoten übertragen werden, bevor es genutzt werden kann. Jetzt kann man es testen und erproben. Erst nach permanenter Speicherung ist es dauerhaft auf dem Knoten abgelegt und steht beim Neustart zur Verfügung.

In der Makroliste wird der aktuelle Status der Speicherung pro Makro angezeigt.



Ab Build 3116 gibt es einen neuen initialen Makrostatus. Im Wizard werden die Makros erst vom Knoten gelesen, wenn ein Makro bearbeitet wird, indem es in der Makroliste selektiert wird. Um unterscheiden zu können ob ein Makro schon vom Knoten gelesen wurde, wird dieser initiale Status angezeigt.

Ein Makro welches nicht permanent gespeichert ist, geht beim Unterbruch der Versorgungsspannung verloren!

Auf Eingangswert warten

Die LightControl und OneControl bietet die Möglichkeit in einem Makro den Zustand eines Eingangs abzufragen und falls der Eingang den angegeben Zustand (0 oder 1) nicht hat, so lange zu warten bis der Eingang den Zustand erreicht oder das Makro gestopped wird. Die nachfolgende Abbildung zeigt die Konfguration um das Makro im Schritt 2 so lange anzuhalten bis der Input_0 den Wert 0 hat.

Step	Delay	Port Type	Action	Port
1. Step	Delay: 0 Ticks	Switch port	Turn off	Test Power 8
2. Step		Input port	Value = 0	Input_0
3. Step	Delay: 20 Ticks	Switch port	Turn on	Test Power 8

Kritischer Abschnitt

Die Makroengine bietet die Möglichkeit das Abbrechen/Stoppen eines Makro von aussen zu unterbinden, falls eine Schrittfolge nicht unterbrochen werden darf.

Schritt	Verzögerung	Port Typ	Aktion	Port	Extra
1. Schritt		krit. Bereich	Anfang		
2. Schritt	Wartezeit: 0 Ticks	Servoausgang	anfahren	Weiche_1	Zielwert: 10 %
3. Schritt		Servo-Bewegung abwa...		Weiche_1	
4. Schritt	Wartezeit: 0 Ticks	Schaltausgang	einschalten	Ausgang_10	
5. Schritt		krit. Bereich	Ende		

- Anfang / BIDIB_MSYS_BEGIN_CRITCAL: Eine Stopanforderung an dieses Makro wird nicht beachtet.
- Ende / BIDIB_MSYS_END_CRITCAL: Eine Stopanforderung an dieses Makro wird wieder beachtet.

D.h. nach BEGIN_CRITCAL macht das Makro sein Ding fertig bis zum Schritt END_CRITICAL. Basta.

Accessory Notify

Die Makroengine bietet die Möglickeit, das erfolgreiche Ende der Auführung eines Accessory an das Steuerungs-Programm zu melden bevor das Accessory durchgelaufen ist.

Normalerweise wird die „Accessory fertig durchgelaufen“-Meldung immer nach dem Ende der Ausführung des Accessory gesendet. Dies dient dazu den kritischen Vorgang der Accessory-Ausführung überwachen zu können. Falls nämlich ein Fehler auftritt wird der Fehler zurückgemeldet und nicht ok. Diese „Ende“-Mitteilung wird pro Ausführung eines Accessory genau 1x gesendet.

Falls man erreichen will, dass diese Benachrichtigung schon vor dem eigentlichen Ende der Ausführung des Accessory an das Steuerungs-Programm gesendet wird (weil z.B. alle kritischen Schaltvorgänge schon durchgeführt sind), dann kann man den Befehl „Accessory notify“ verwenden. Wenn man das „Accessory notify“ mehr als 1x im Makro drin hat, dann hat es keine Auswirkung mehr nachdem das 1. „Accessory notify“ gesendet wurde, denn der „Befehl“ wird nicht mehr gesendet.

Dazu wird unter Port Typ der „Accessory notify“ ausgewählt. In der Spalte Aktion kann bestimmt werden, ob die Meldung gesendet wird, wenn der selektierte Port **zum Zeitpunkt der Abarbeitung dieses Schritt** den Wert = 0 oder Wert = 1 hat.

Flag abfragen

Mit dem Makroschritt `Flag abfragen` wird die Ausführung eines Makro anhand eines gesetzten oder nicht gesetzten Flag pausiert. D.h. wenn die Bedingung nicht erfüllt ist, wartet das Makro in diesem Schritt solange, bis die Bedingung erfüllt ist.

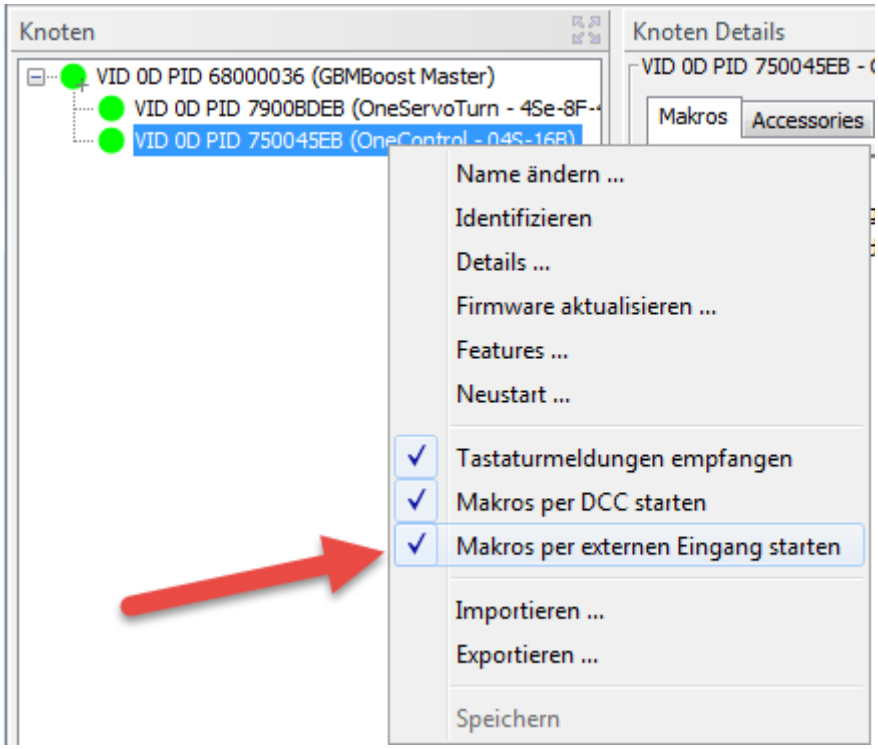
Der Makroschritt `Flag abfragen 1` pausiert das Makro solange, bis das Flag gesetzt ist. Der Makroschritt `Flag abfragen 0` pausiert das Makro solange, bis das Flag nicht (mehr) gesetzt ist.

Starten per externem Eingang

Die LightControl und OneControl bieten die Möglichkeit Makros von einem Eingang (Input) starten zu lassen. Dabei wird die Nummer des Eingang auf die Nummer des Makro gemappt, also Eingang 1

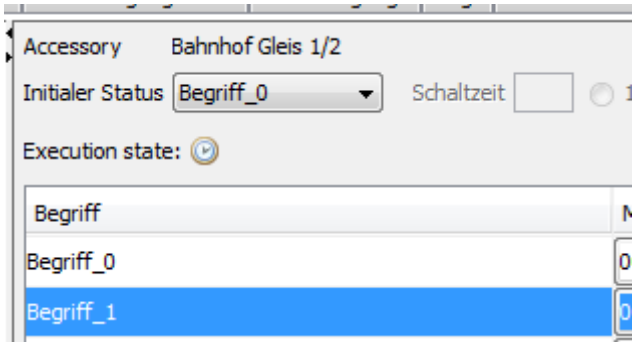
startet Makro 1, usw.

Dazu muss auf dem Knoten das Feature *FEATURE_CTRL_MAC_START_MAN* aktiviert sein. IM Wizard erfolgt dies über das Kontext-Menü auf dem Knoten.

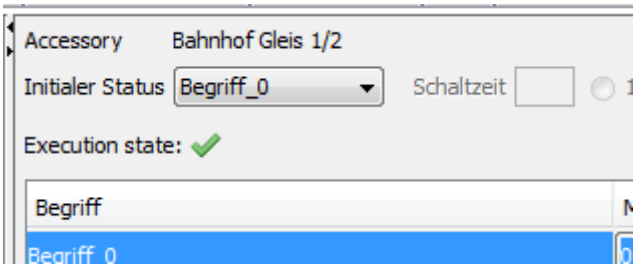


Accessory ausführen

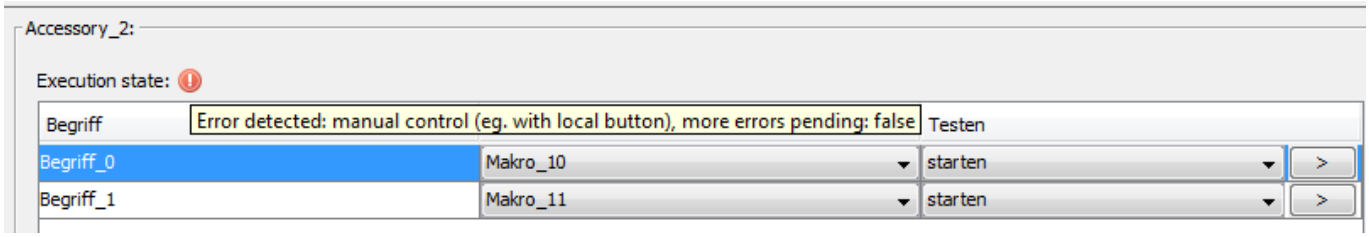
Bei der Ausführung eines Accessory wird der Ausführung-Status angezeigt. Solange das Accessory ausgeführt wird, wird eine Uhr als Symbol angezeigt:



Wenn das Accessory erfolgreich beendet wurde (ohne Fehler), dann wird dies durch ein grünes Häckchen signalisiert:



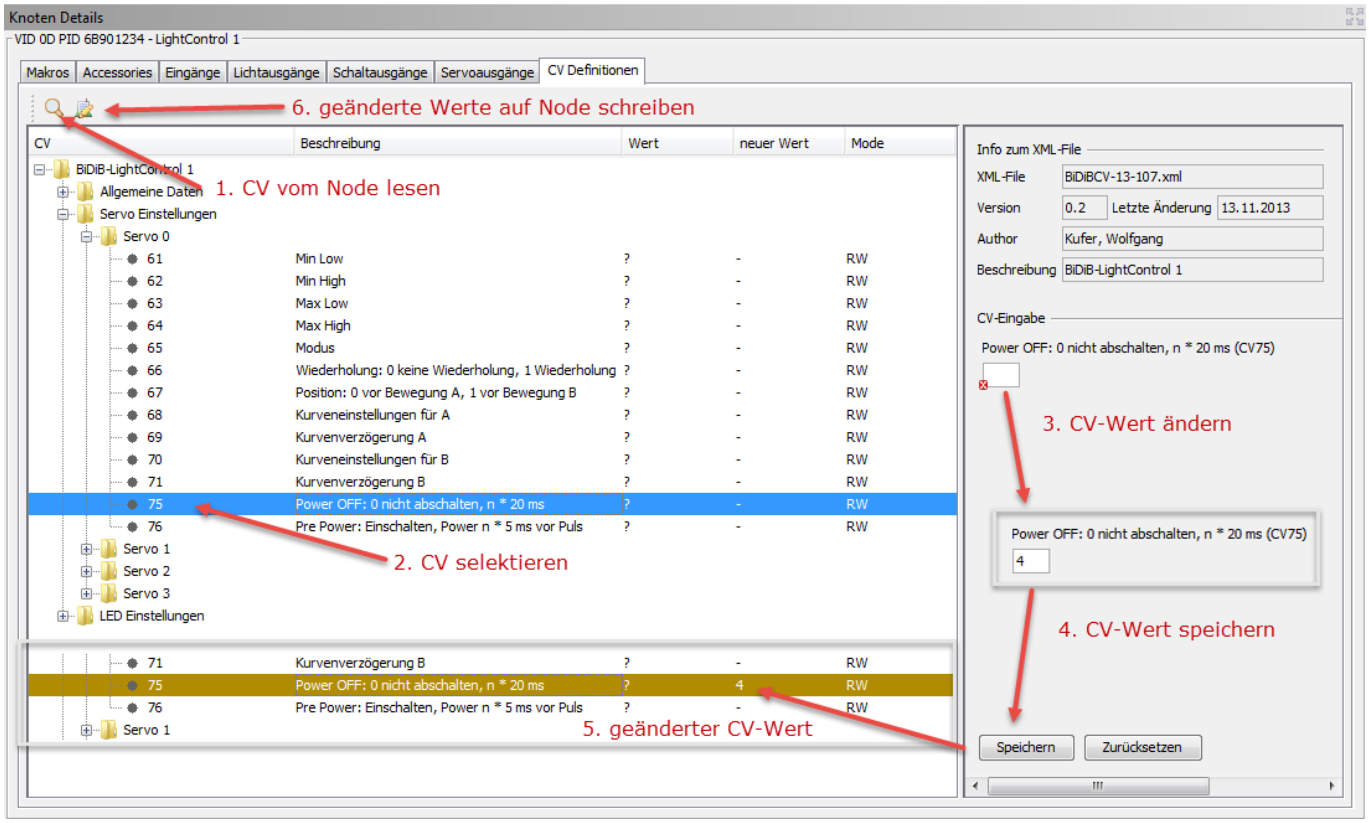
Fehler bei der Ausführung eines Accessory werden mit einem *Fehler-Icon* dargestellt. Im Tooltip des *Fehler-Icon* werden genauere Informationen über den Grund des Fehlers dargestellt.



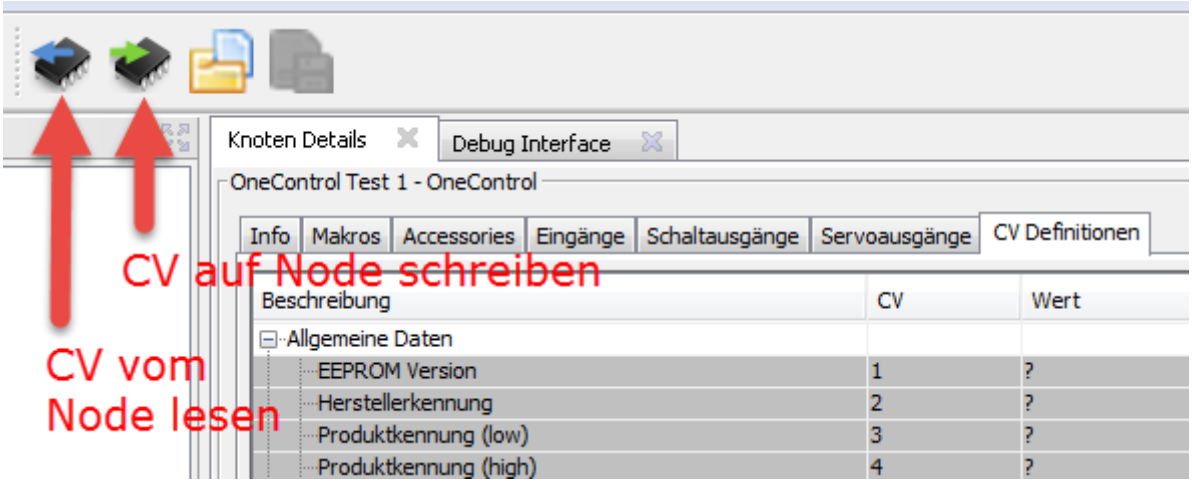
CV-Werte anpassen

CV-Werte der Knoten lassen sich über das Tab *CV-Defintionen* einstellen.

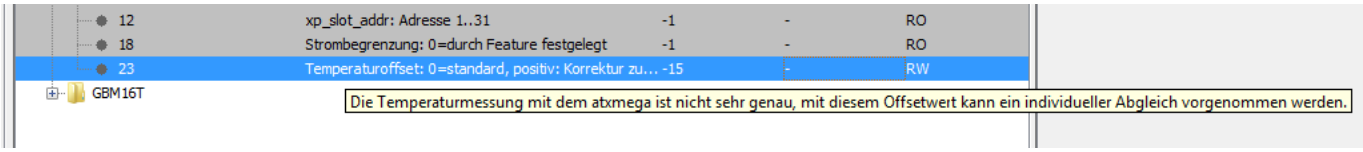
Hinweis: In der Abbildung sind in der Spalte Wert nur Fragezeichen drin weil die Abbildung im Demo-Modus erstellt wurde. Nach dem Laden der CV stehen dort die richtigen Werte drin.



Beim Wizard-1.8 und neuer sind die Buttons zum Lesen und Schreiben der CV-Werte vom Knoten in die Toolbar gewandert.



Falls bei einer CV weitere Informationen als Hilfetext hinterlegt sind, werden diese als Tooltip dargestellt.



Um CV-Werte anzupassen sind folgende Schritte notwendig:

- CV auswählen
- CV-Wert anpassen
- geänderten CV-Wert speichern. Es können auch mehrere Werte verändert und gespeichert werden, bevor die Übertragung auf den Knoten erfolgt.
- geänderte CV-Werte auf den Knoten übertragen

In einigen Fällen werden geänderte CV-Werte erst nach einem Neustart des Knoten übernommen.

Für Bit-Werte kann ebenfalls mehr Information als Hilfetext hinterlegt werden, welche als Tooltip bei den Checkboxen angezeigt wird.

Benutzerdefinierte CV-Dateien

Der BiDiB-Wizard liefert aktuelle CV-Definitionen mit. Es besteht aber (ab Build 1589) auch die Möglichkeit eigene oder **angepasste CV-Definitionen** zu verwenden. Dabei werden folgende Verzeichnisse durchsucht:

1. <user.home>**.BiDiBWizard**\data\BiDiBNodeVendorData (z.B. C:\Users\admin\.BiDiBWizard\data\BiDiBNodeVendorData)
2. <Installations-Verzeichnis>\data\BiDiBNodeVendorData (z.B. C:\Program Files\BiDiB-Wizard-1.7-SNAPSHOT\data\BiDiBNodeVendorData)
3. Interne Suche in Programm-Archiven

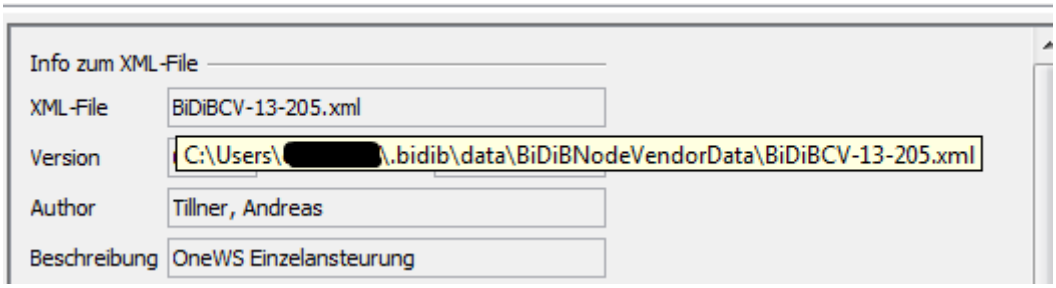
Ab **BiDiB-Wizard-1.9 Build 2507** werden folgende Verzeichnisse durchsucht:

1. <user.home>**.bidib**\data\BiDiBNodeVendorData (z.B. C:\Users\admin\.bidib\data\BiDiBNodeVendorData)
2. <Installations-Verzeichnis>\data\BiDiBNodeVendorData (z.B. C:\Program Files\BiDiB-Wizard-1.9-SNAPSHOT\data\BiDiBNodeVendorData)
3. Interne Suche in Programm-Archiven

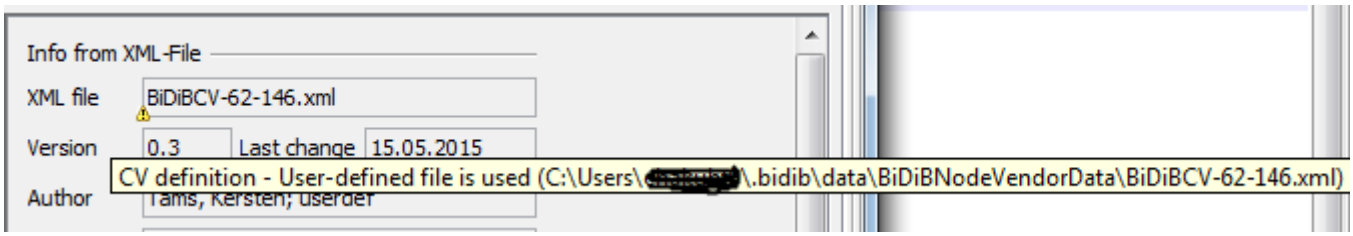
Wenn das **Verzeichnis für XML-Dateien** nicht auf den Defautwert *<user.home>\.bidib* verweist, werden im 1. Schritt die Dateien in diesem Verzeichnis unter *data\BiDiBNodeVendorData* gesucht.

Sobald eine passende CV-Definition gefunden wurde, wird diese verwendet und die Suche in den weiteren Verzeichnissen abgebrochen.

Der Wizard zeigt ab Build 2545 den Pfad zur verwendeten Template-Datei im Tooltip auf dem Textfeld XML-File an.

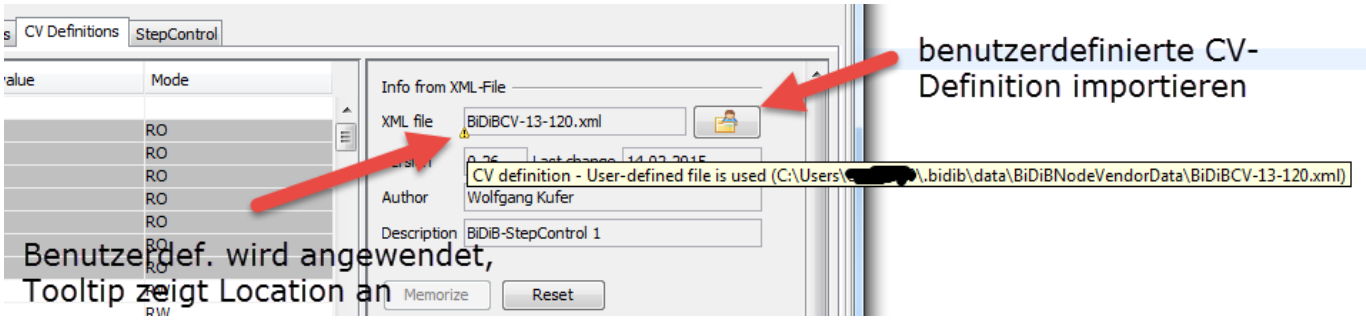


Der Wizard zeigt ab Build 2937 einen Warnungs-Marker an, wenn eine benutzer-definierte CV-Definition verwendet wird. Im Tooltip auf dem Warnungs-Marker wird der Pfad zum XML-File angezeigt.



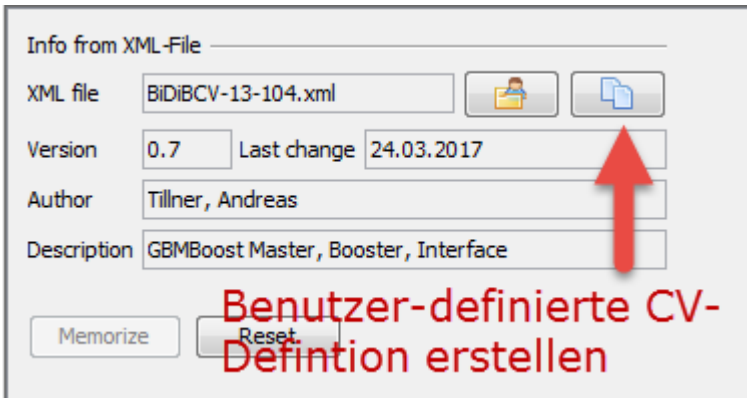
Import von benutzerdefinierten CV-Definitionen

Der Wizard erlaubt ab Build 2967 den Import von benutzerdefinierten CV-Definitionen über einen Button im UI.



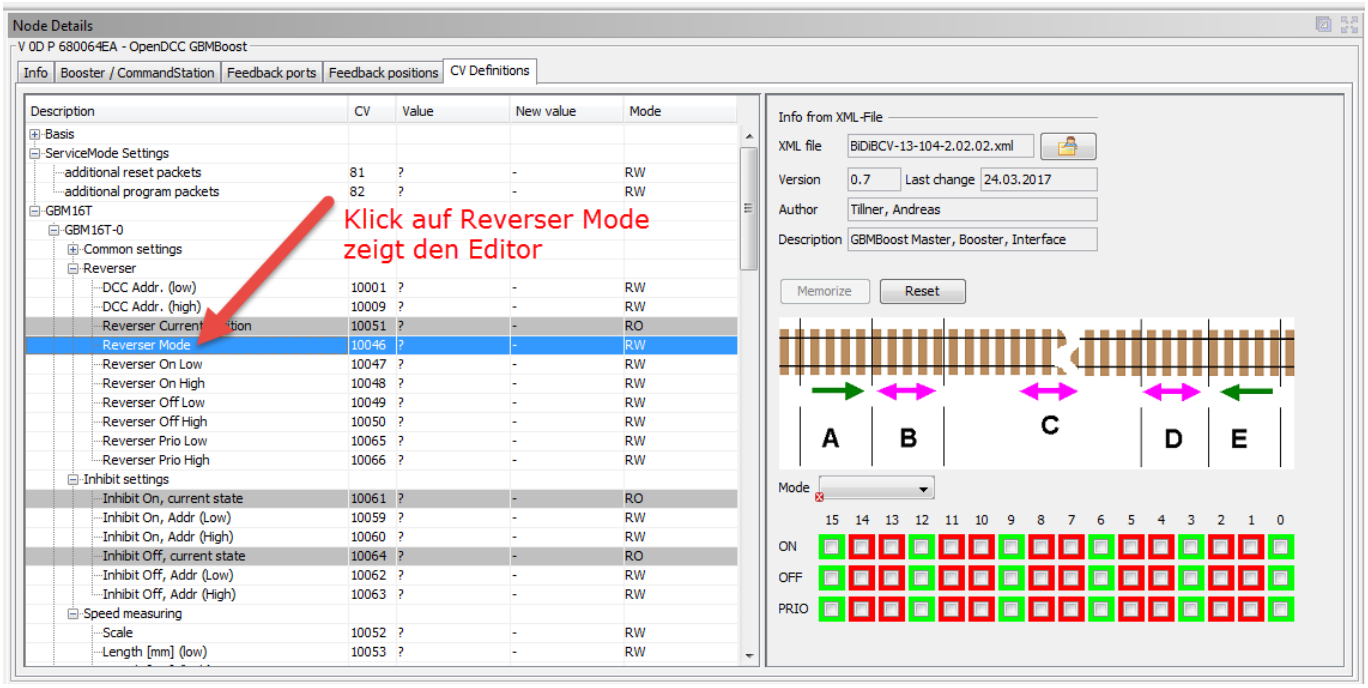
Erstellen von benutzerdefinierten CV-Definitionen

Der Wizard erlaubt ab Build 3335 das Erstellen von benutzerdefinierten CV-Definitionen über einen Button im UI. Dabei wird die interne CV-Definitionen des Wizard in das Benutzer-Verzeichnis kopiert, damit sie dort angepasst werden kann.



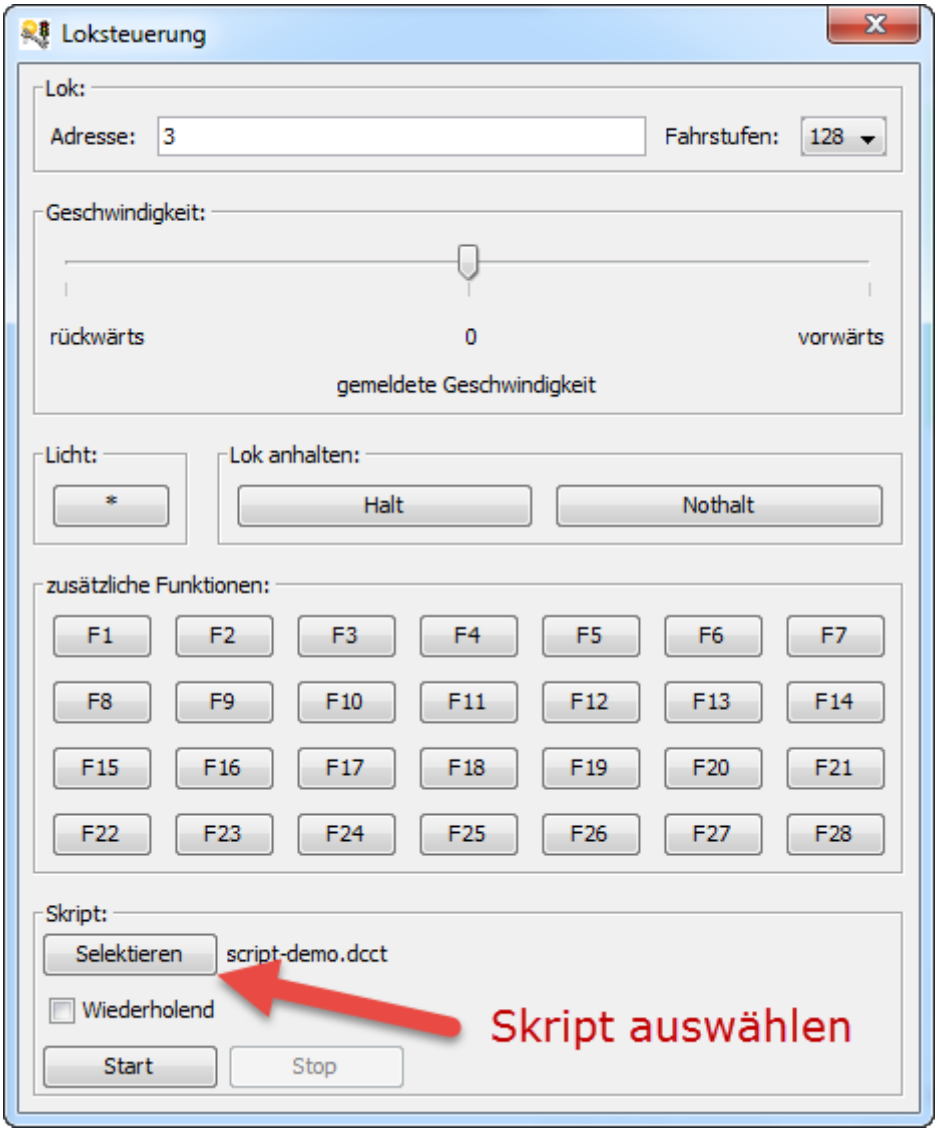
Kehrschleifen / Reverser Editor

Beim Klick auf die CV von Reverser Mode, Reverser On/Off/Prio wird der Editor für die Kehrschleife angezeigt.



Loksteuerung skripten

Der Wizard bietet die Möglichkeit die Loksteuerung mit einem einfachen Skript zu automatisieren. Dazu kann in der Loksteuerung ein Skript ausgewählt werden. Ein Demo-Skript liegt im Installationsverzeichnis unter *data/script*.

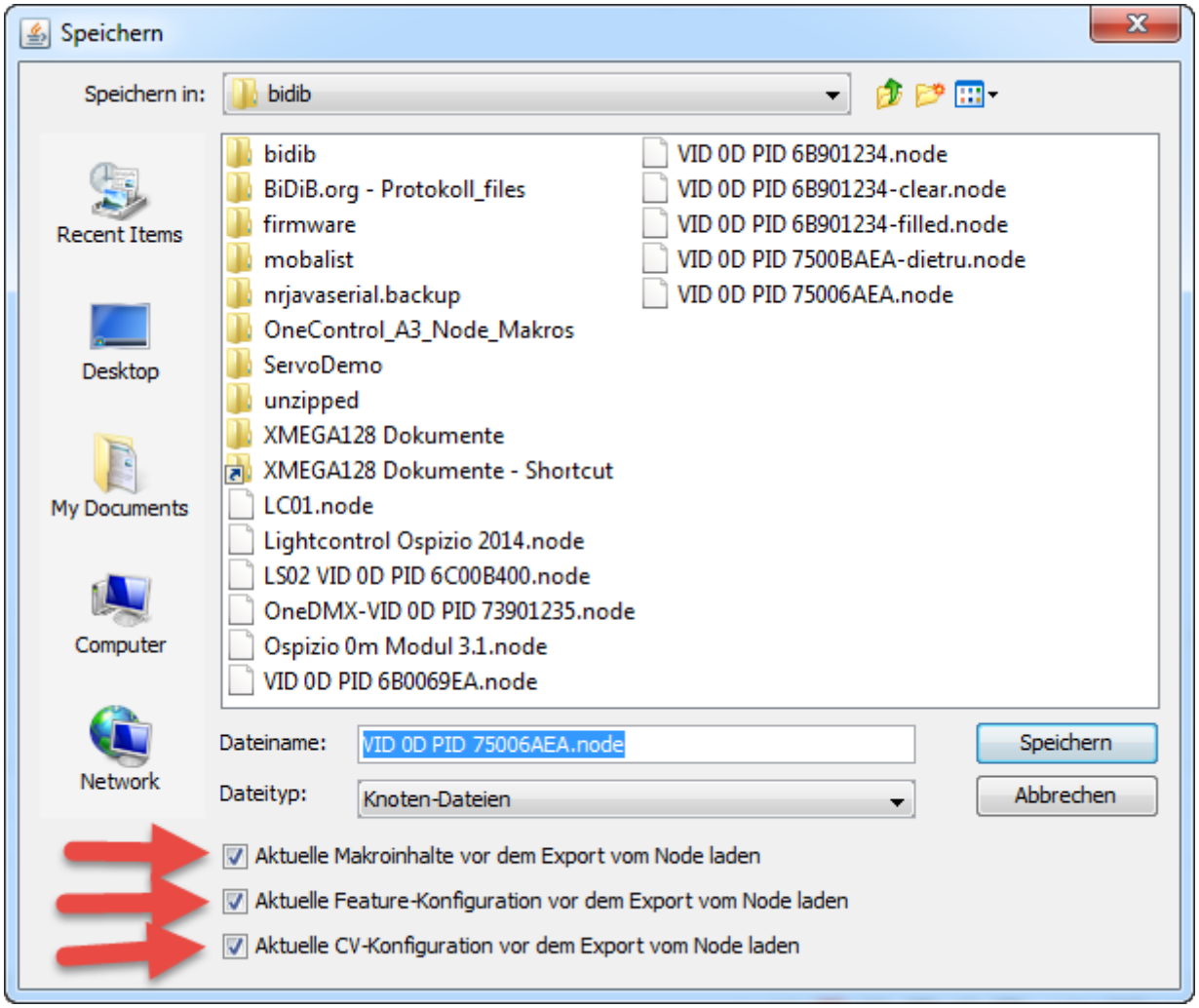


Die Kommandos für das Skripting der Loksteuerung sind hier beschrieben.

Export Knoten

Der Knoten-Export wird über das Kontextmenü des ausgewählten Knoten gestartet: `Exportieren …`

Beim Exportieren eines Knoten ist es wichtig, dass geänderte CV-Werte auch exportiert werden. Dazu müssen die 3 Checkboxen wie im nachfolgenden Bild aktiviert sein.



Der Wizard 1.12 verwendet standardmässig ein anderes Format für den Knoten-Export und Import. Man kann das in den Einstellungen > Experimentelle Eigenschaften > Java-Serialisierung für Knotenexport verwenden auf das alte Format umschalten. Anschliessend den Wizard neu starten.

Import Knoten

Nach einem Update kann die alte Konfiguration wieder in den Knoten geladen werden. Dabei werden auch Makros und Accessories restored. Vorsicht: ein Import überschreibt auch Servoeinstellungen, dadurch kann ein Servo unbeabsichtigt weit umlaufen. Es daher empfohlen, beim Import die Servos abzustecken und erst nach Überprüfung der Einstellungen die Servos wieder anzustecken.

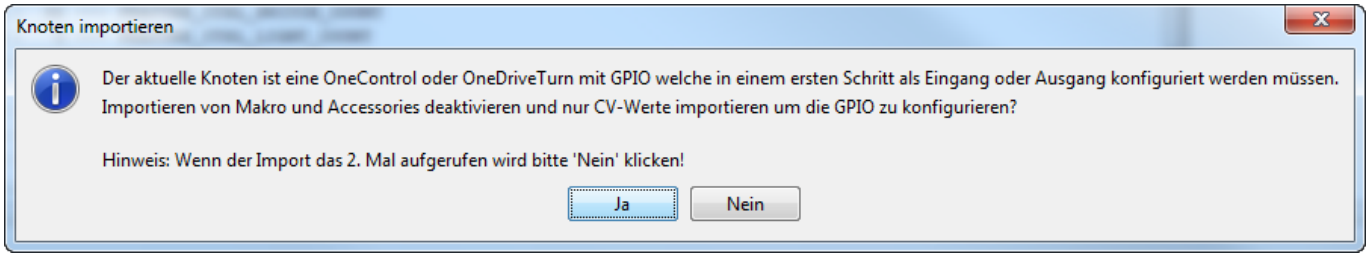
Import OneControl

OneControl: Die GPIO der OneControl werden durch Anpassungen der CV als Eingang oder Ausgang konfiguriert. Wenn ein GPIO-Port umkonfiguriert wird, dann muss ein Reset (Zurücksetzen) der OneControl durchgeführt werden, bevor die neue Port-Konfiguration aktiviert wird. Falls dies nicht erfolgt, können Makros nicht importiert werden, welche die angepasste Konfiguration erwarten. Wenn z.B. ein GPIO-Port auf Eingang (Input) konfiguriert ist, das Makro diesen Port aber als Ausgang

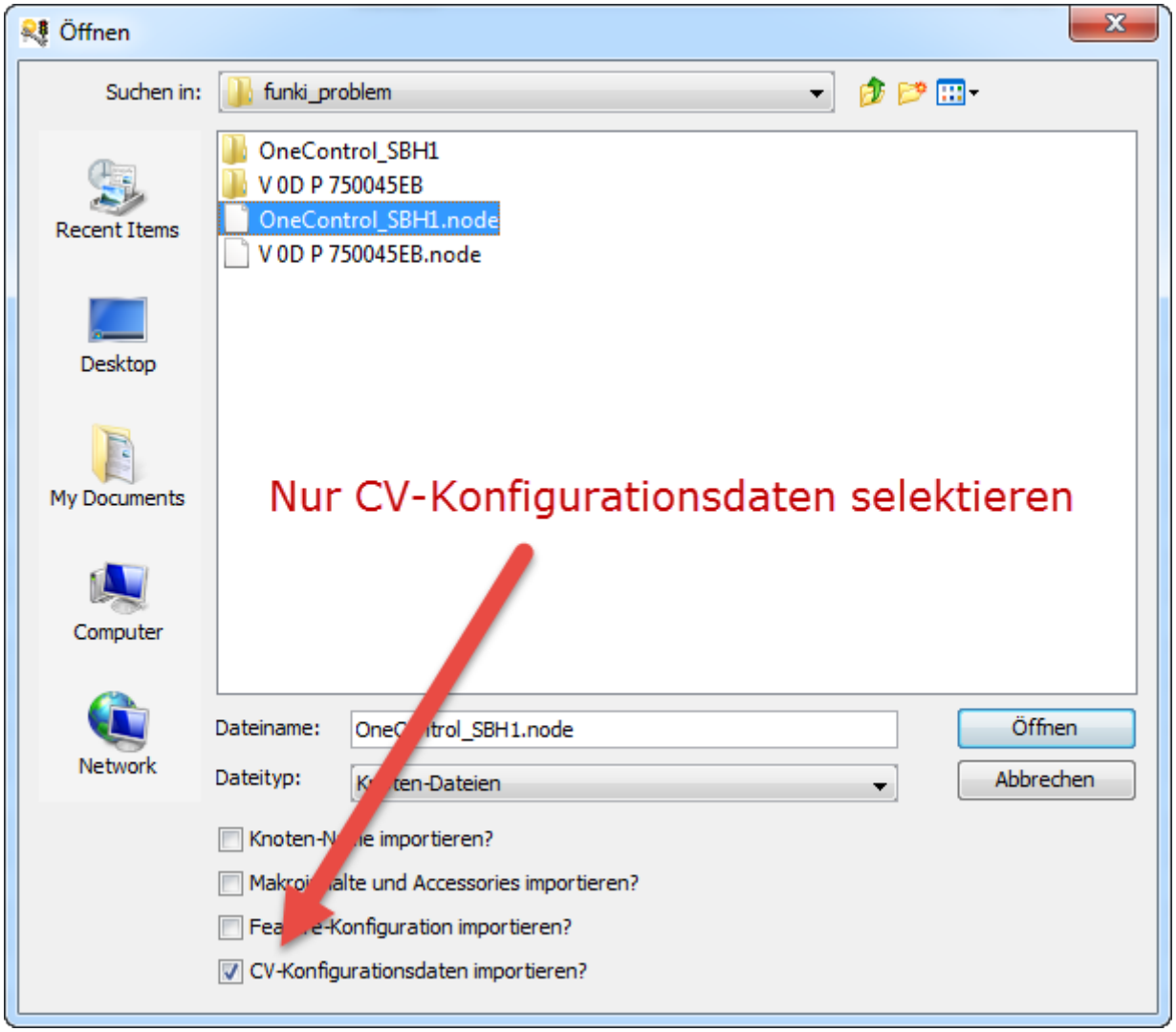
erwartet, dann schlägt der Import dieses Marko fehl.

Aus diesem Grund müssen beim Importieren der OneControl mit veränderten GPIO-Ports aus der Node-Sicherung zuerst nur die CV importiert werden, anschliessend ein *Zurücksetzen* ausgeführt werden und in einem weiteren Schritt nachdem sich die OneControl neu angemeldet hat können die restlichen Einstellungen wiederhergestellt werden.

Ab Wizard-1.9 Build 2293 wird ein Dialog angezeigt, welcher auf das Thema aufmerksam macht. Wird der Import zum 1. Mal aufgerufen, kann der Dialog mit *Ja* bestätigt werden, um die entsprechenden Einstellungen vorzunehmen.

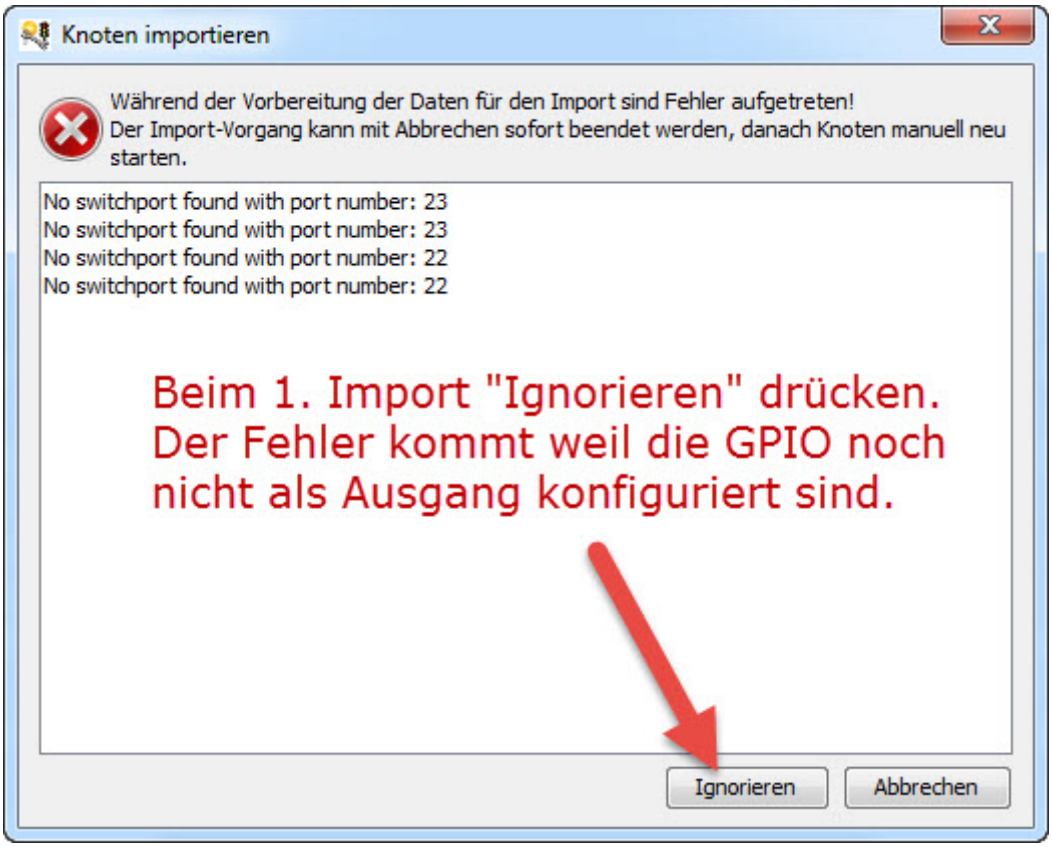


Auswahl der Sicherung mit den selektierten Checkboxen.

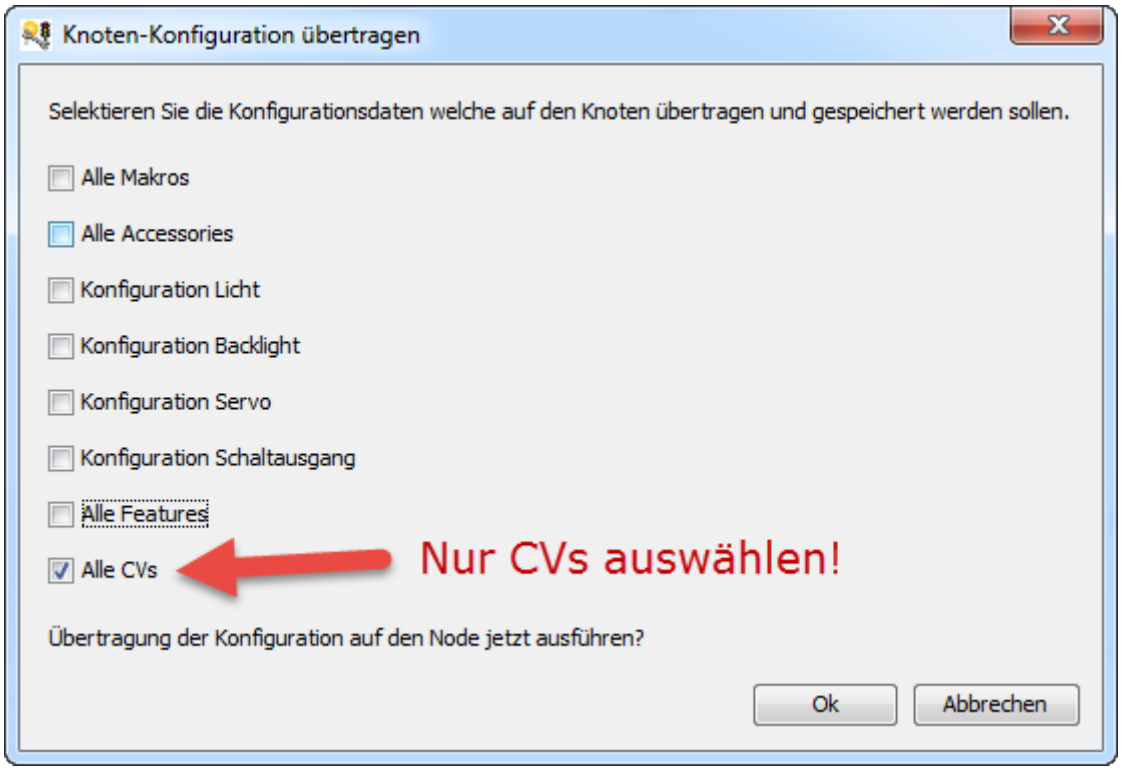


Wenn GPIO noch nicht korrekt konfiguriert sind kommt eine Fehlermeldung, welche in diesem Fall

ignoriert werden kann.



Wie oben beschrieben im ersten Schritt nur die CVs importieren:

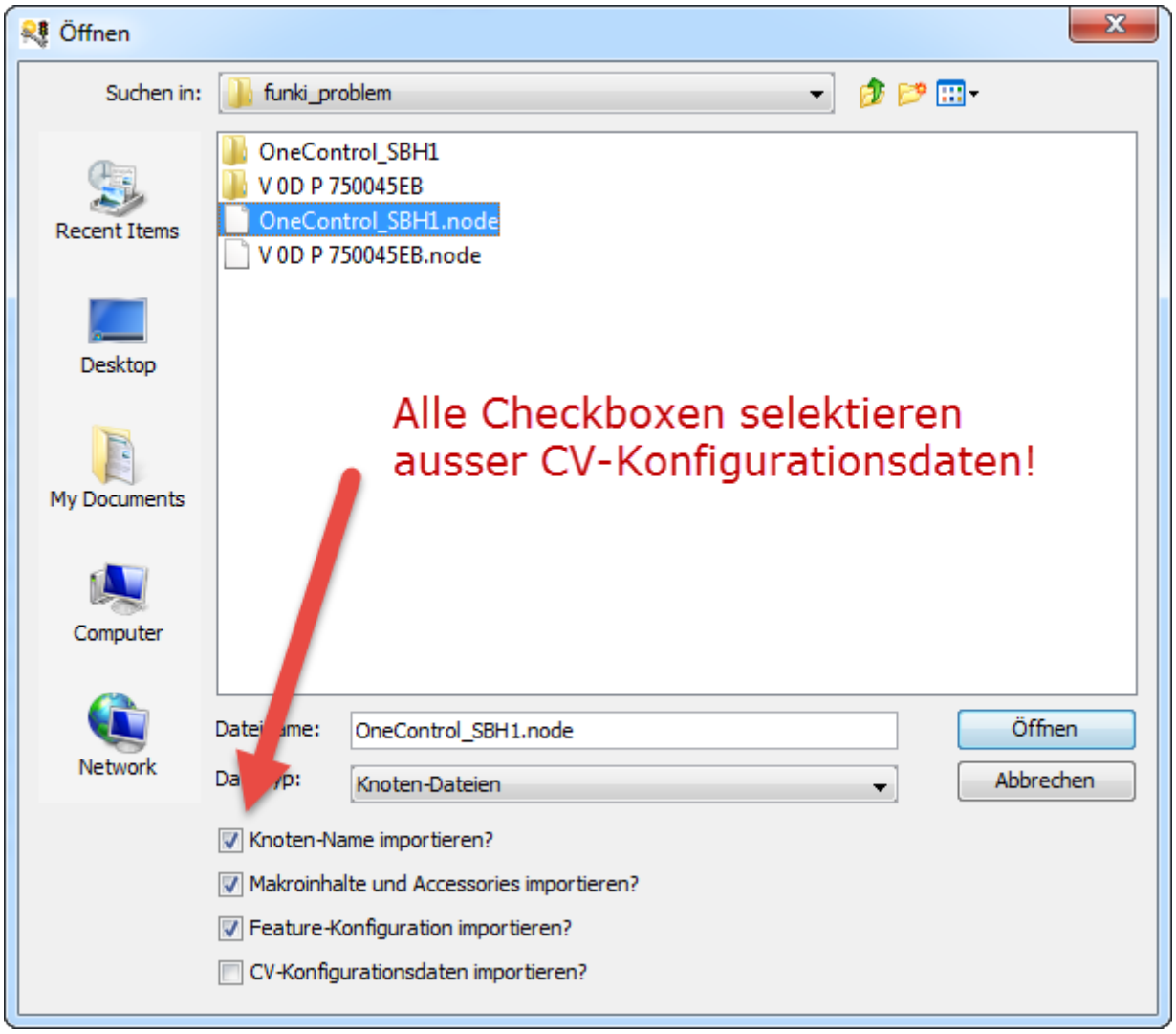


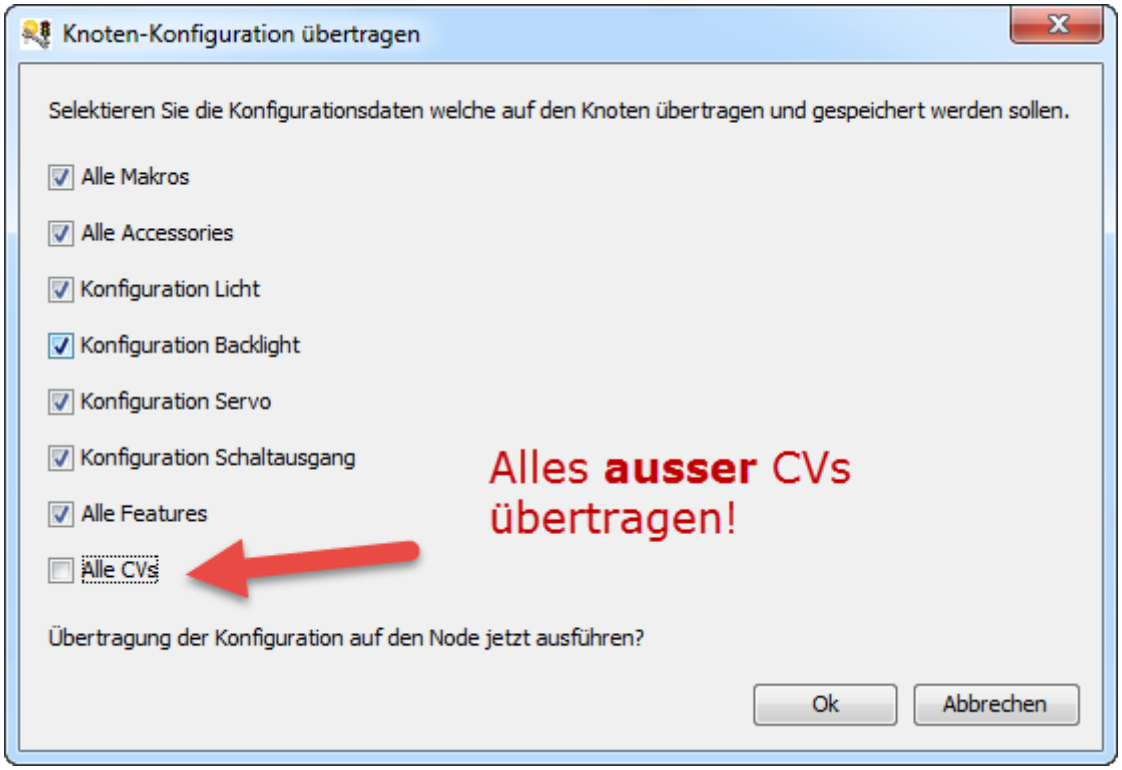
Knoten neu starten

Anschliessend muss der Knoten neu gestartet werden. Dazu muss über das Kontextmenü ein ***Neustart*** (*Reset*) auf der OneControl ausführt werden.

Nachdem die OneControl wieder am Bus ist, in einem zweiten Schritt den Rest der Konfiguration in die OneControl importieren.

Dazu die gleiche Auswahl der Sicherung mit allen selektierten Checkboxen ausser den CV verwenden.

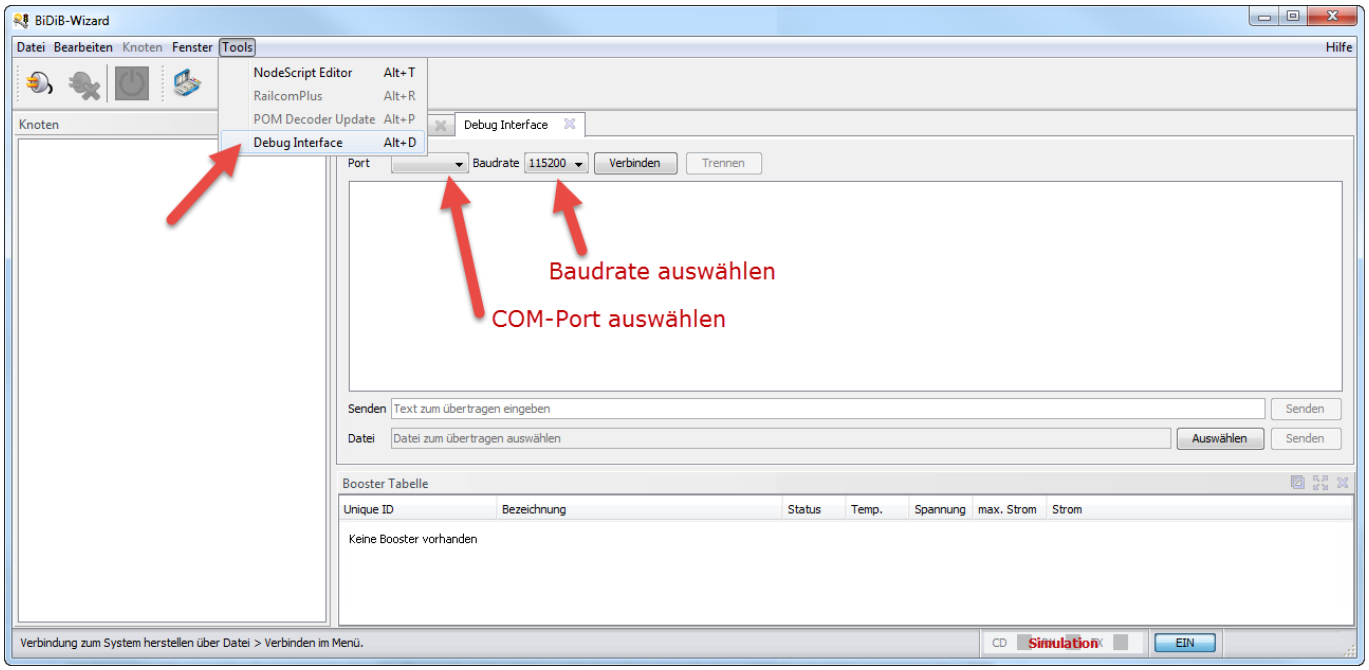




Debug Interface

Ab dem Wizard-1.9 ist ein direkter Zugriff auf die Debug-Schnittstelle möglich statt hterm zu benutzen. Dazu müssen die Schnittstelle und die Baudrate aus den entsprechenden Comboboxen ausgewählt werden.

Wenn der Wizard auf einem ARM-Prozessor läuft, muss in den **Einstellungen** der *Serielle Port Treiber* auf **RXTX** gestellt werden!



Anschliessend wird der Port über 'Verbinden' geöffnet. Durch Eingabe von '?' wird in den meisten Fällen eine Information des Knotens angezeigt.

Für die Übertragung einer Firmware-Datei muss 'f' oder 'e' in das Testfeld 'Senden' eingegeben und gesendet werden. Anschliessend muss die Datei ausgewählt und dann durch Klick auf 'Senden' gesendet werden.

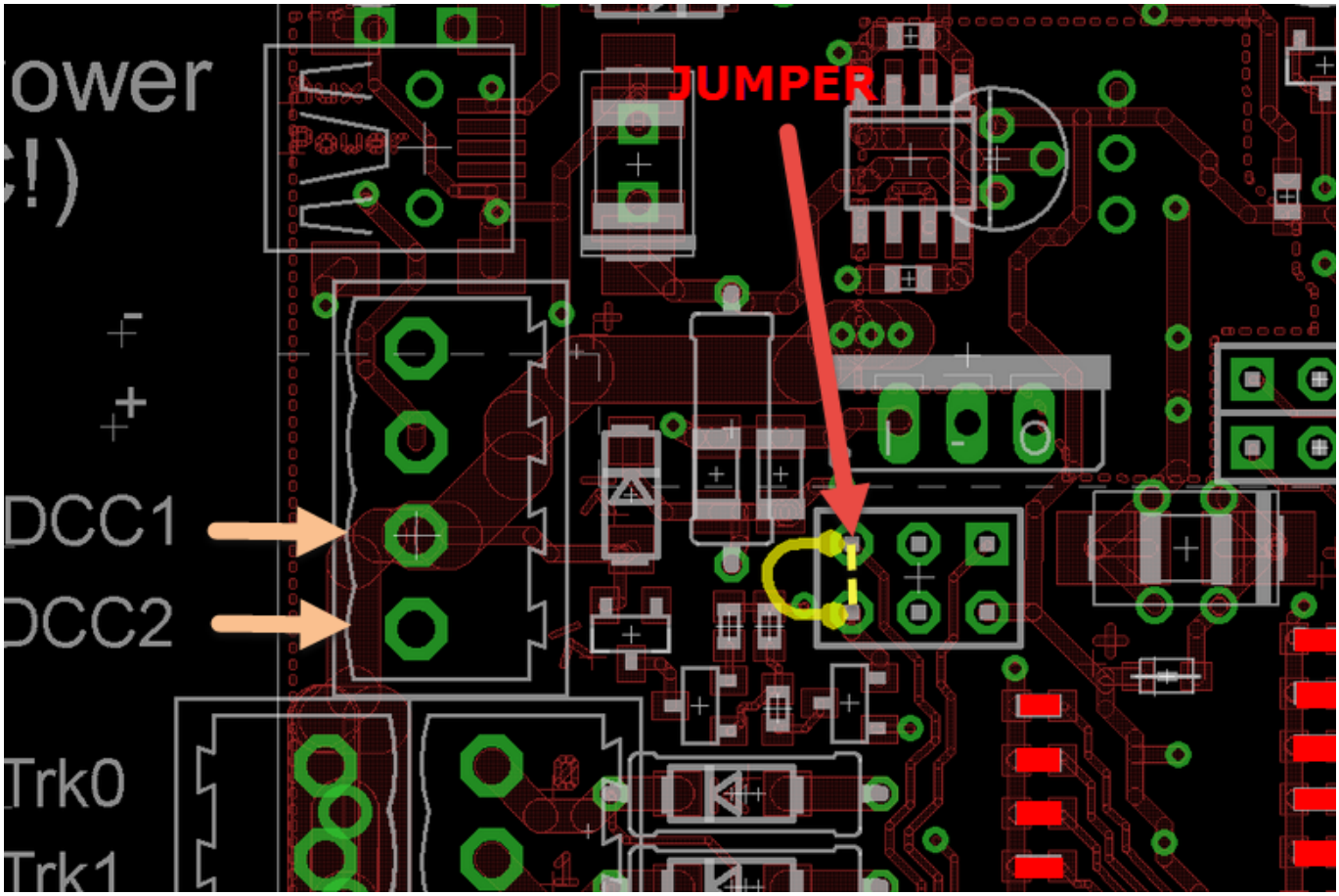
Update vom GBM16T mit dem BiDiB-Wizard (Entwicklungsstand vom Wiki)

Für das Update des GBM16T wird ein FTDI-Kabel (3.3V) benötigt, welches auf den Debug-Port gesteckt werden muss. Siehe auch Logging über FTDI-Kabel.

Tools > Entwickler-Tools > Debug Interface

- **Während dem Einschalten den Ident-Schalter gedrückt halten**.
- Mit Baudrate **19200** verbinden und im Textfeld Senden das **?** eingeben > **Senden** (geht auch mit Enter).
- Ausgabe im Logfenster: **Trackproc_Bootloader V0.01** zeigt an, dass der Bootloader aktiv ist.
- Eingabe von **f** und Senden –> Bootloader quittiert mit einem „.“ und erwartet jetzt ein Flash-File.
- Auswahl von gbm16t_2.04.04.hex aus denm ZIP-File über **Auswählen bei Datei**, anschliessend **Senden** bei Datei –> Bootloader quittiert mit vielen „.“
- Eingabe von **e** und Senden –> Bootloader quittiert mit einem „.“ und erwartet jetzt ein Eeprom-File.
- Auswahl von gbm16t_2.04.04.eep aus denm ZIP-File über **Auswählen bei Datei**, anschliessend **Senden** bei Datei –> Bootloader quittiert mit vielen „.“
- Abschliessend Verbindung trennen und Spannung entfernen, damit der GBM16T neu bootet.

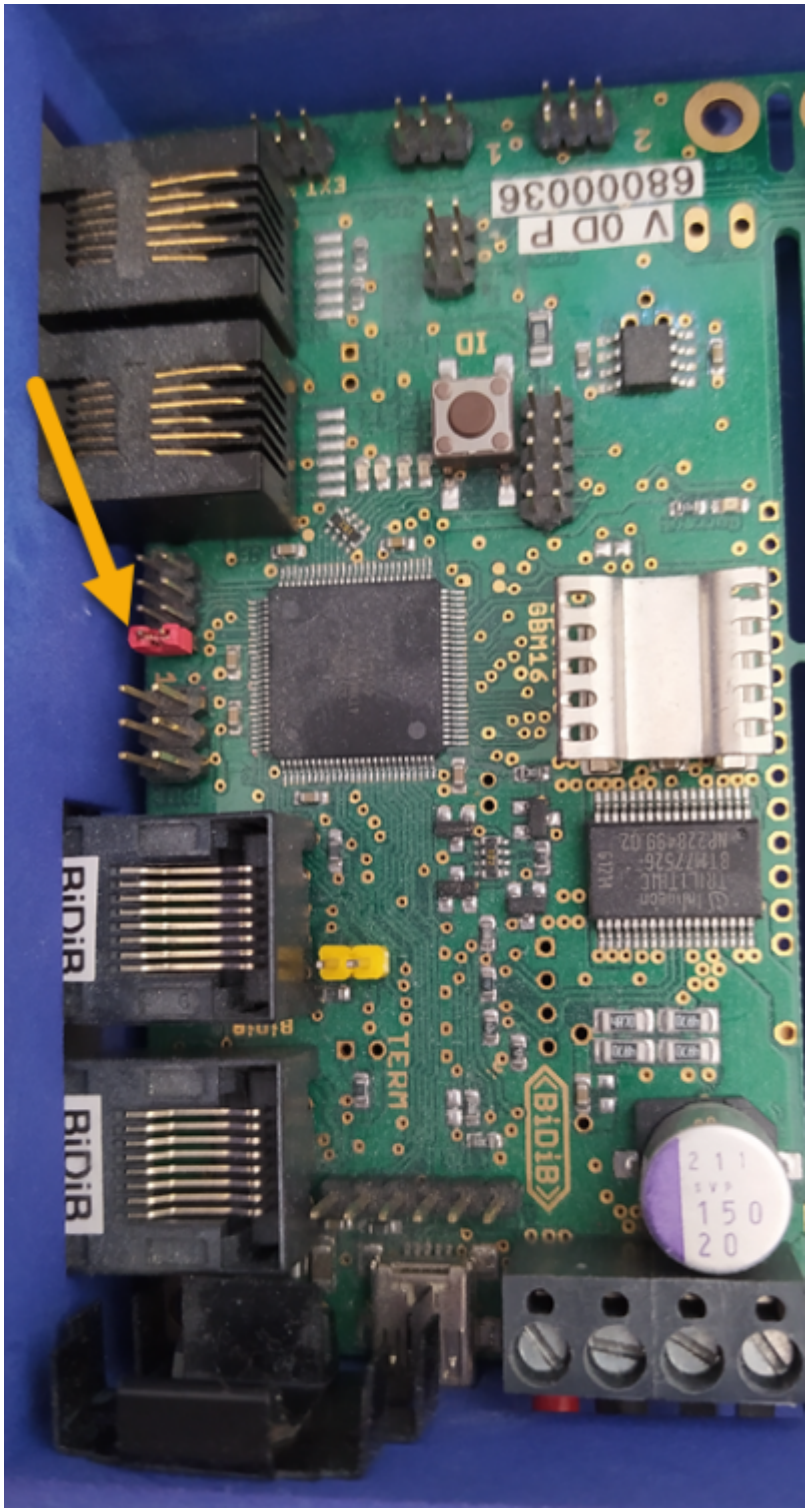
Einfacherer Weg: Reset des GBM16T mit Jumper



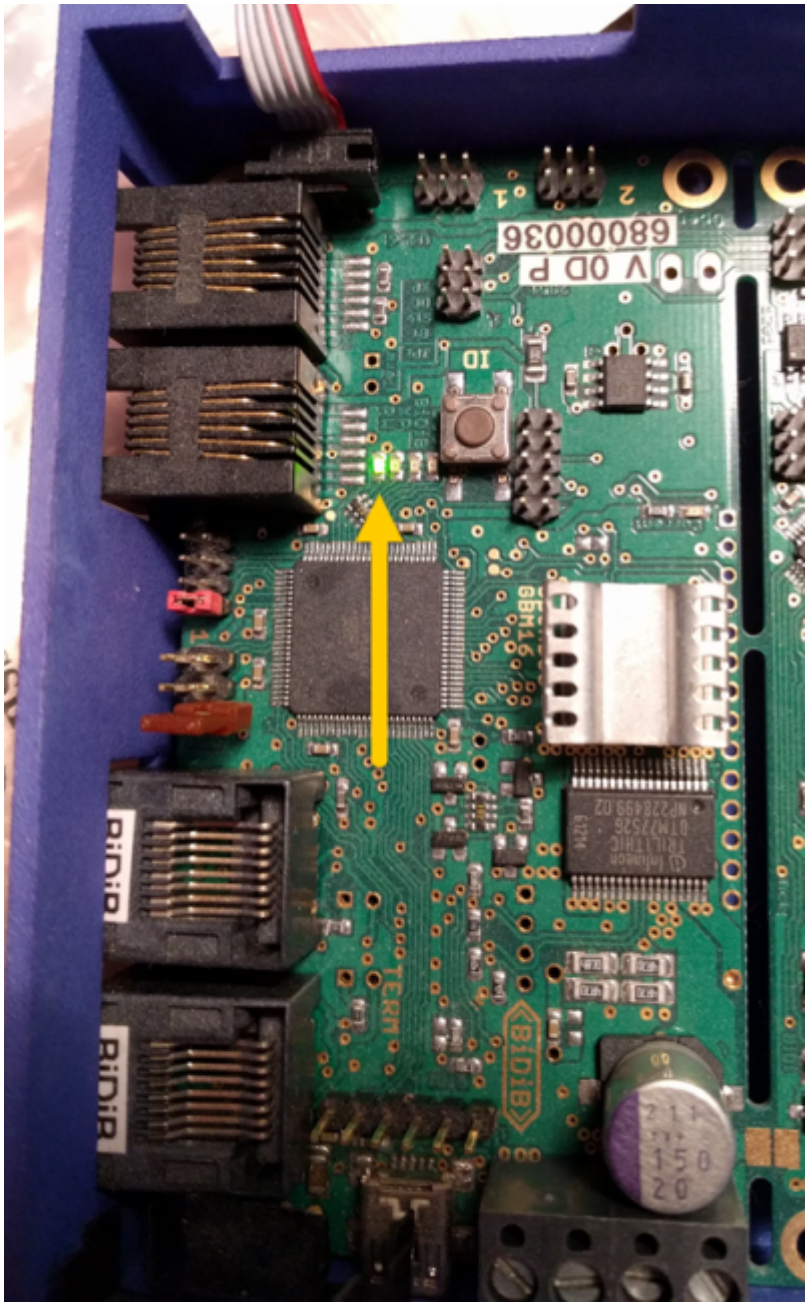
- FTDI-Kabel am GBM16T and PC einstecken, Debug-Interface verbinden mit **19200**,
- Jumper stecken (alle LEDs auf dem GBM16T werden dunkel ...),
- Knopf gedrückt halten,
- Jumper ziehen und dann Knopf loslassen müsste in den BL mode führen.
- Ausgabe im Logfenster beim Loslassen des Knopf: **Trackproc_Bootloader V0.01** zeigt an, dass der Bootloader aktiv ist.

Falls der GBM16T nach dem Update einen Nightrider auf POW, BiDiB und den anderen 2 LEDs hat, kann normalerweise der Ident-Taster gedrückt werden um den Kalibrierungsmodus zu verlassen. Dieser Hinweis per Nightrider deutet darauf hin, dass der GBM16T in den Kalibrierungsmodus gewechselt hat.

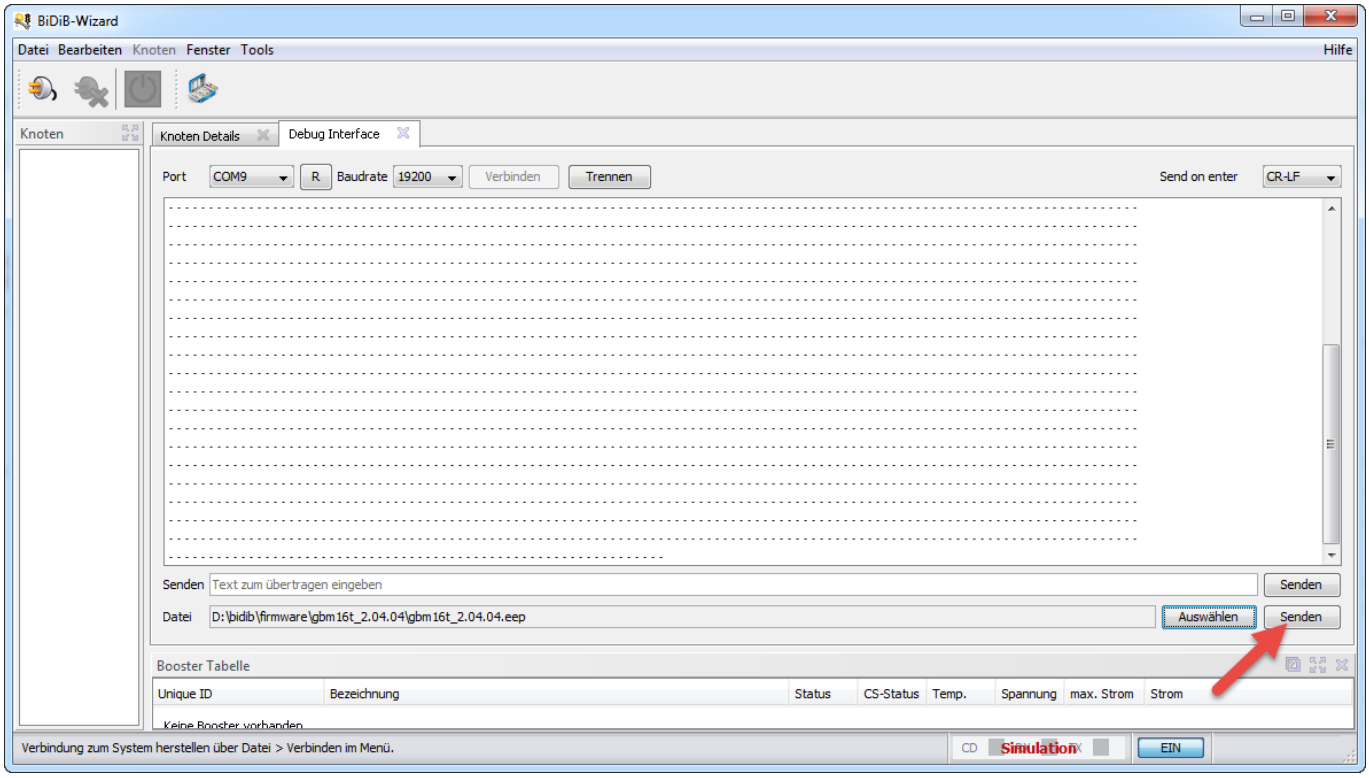
Update vom GBMboost Master mit dem BiDiB-Wizard (Entwicklungsstand vom Wiki)

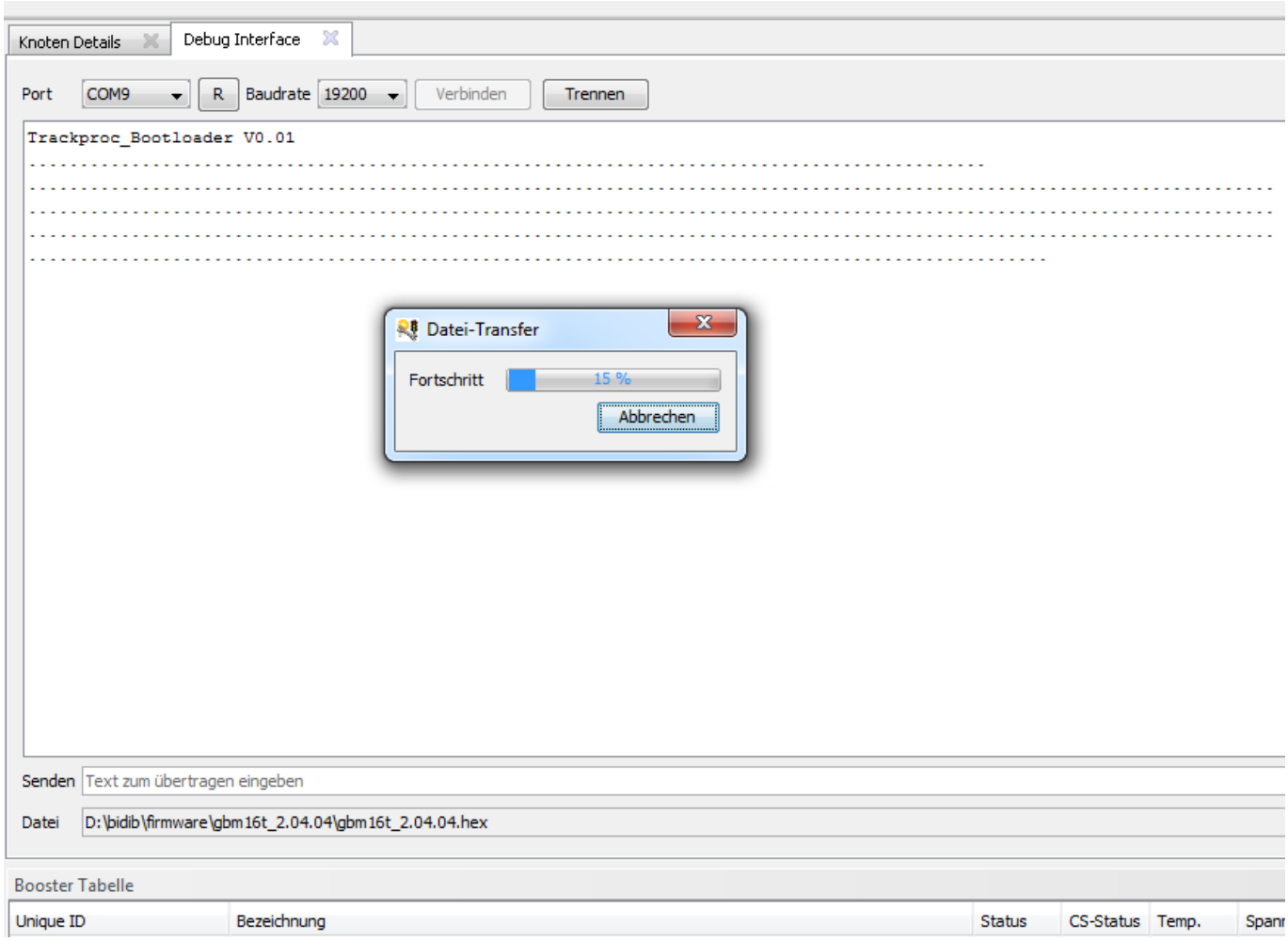


- **Jumper J54 Position J3** setzen! (siehe markierter Jumper im Bild oben)
- Während dem Einschalten den **Ident-Schalter gedrückt** halten.
  Achtung: Anstecken des USB-Kabel startet den GBMboost schon ...
- Nur die LED **PW** leuchtet ...



- Wizard starten und über Tools > Entwickler-Tools > Debug Interface das Debug Interface öffnen.
- Mit Baudrate **19200** verbinden und im Textfeld Senden das **?** eingeben > **Senden** (geht auch mit Enter).
- Ausgabe im Logfenster: **GBMboost-Bootloader V0.02.01** zeigt an, dass der Bootloader aktiv ist.
- Eingabe von **f** und Senden –> Bootloader quittiert mit einem „.“ und erwartet jetzt ein Flash-File.
- Auswahl von gbmboost_master_2.02.07.000.hex aus denm ZIP-File über **Auswählen bei Datei**, anschliessend **Senden** bei Datei –> Bootloader quittiert mit vielen „.“
- Eingabe von **e** und Senden –> Bootloader quittiert mit einem „.“ und erwartet jetzt ein Eeprom-File.
- Auswahl von gbmboost_master_2.02.07.001.hex aus denm ZIP-File über **Auswählen bei Datei**, anschliessend **Senden** bei Datei –> Bootloader quittiert mit vielen „.“
- Abschliessend Verbindung trennen und Spannung und Jumper entfernen, bevor der GBMboost Master neu bootet.

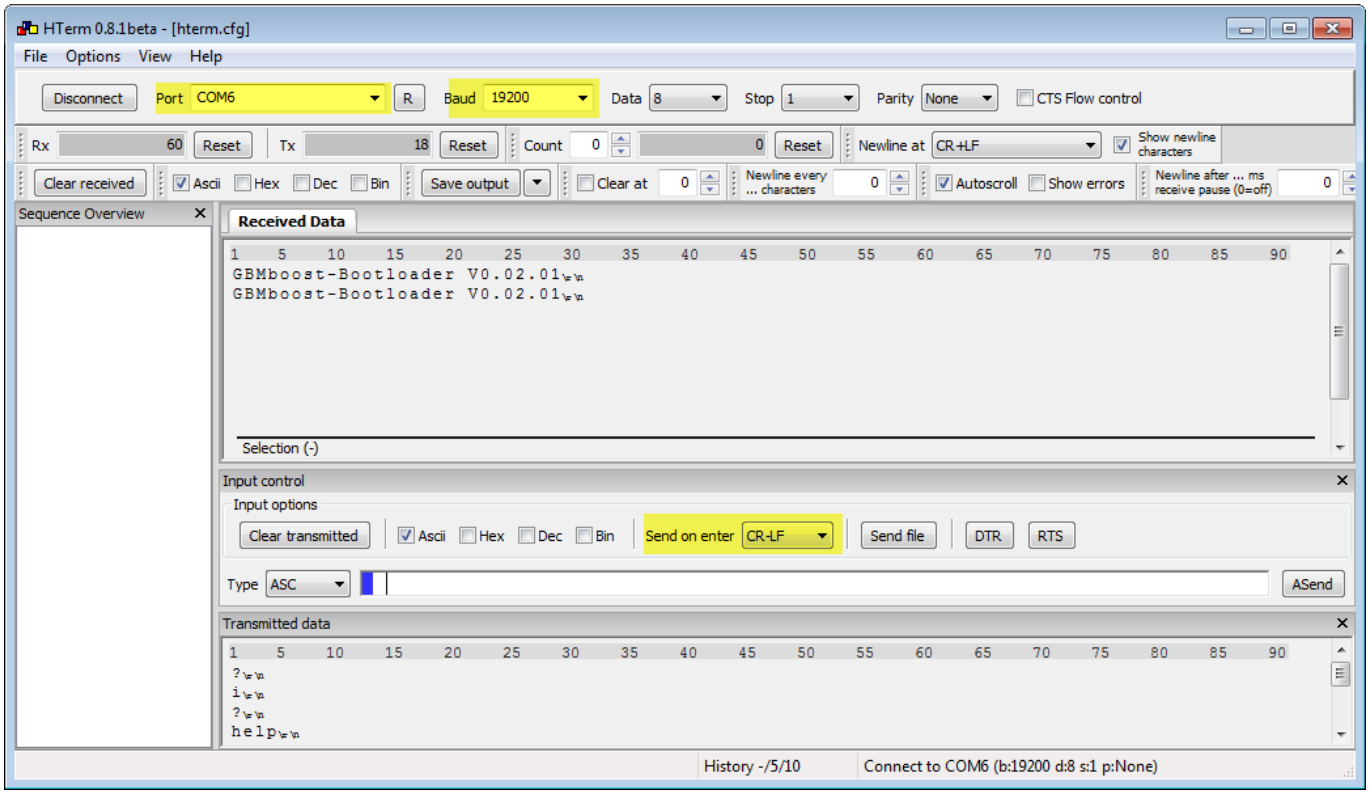




Update vom GBMboost Master mit hterm

Die Vorbereitung des GBMboost Master sind wie im vorherigen Abschnitt.

- Hterm starten und nach Auswahl des Port und der Baudrate 19200 mit Connect verbinden.
- Nach der Eingabe des '?' und Enter zum Senden antwortet der Bootloader mit dem Text „GBMboost-Bootloader“.



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

Permanent link:
https://forum.opendcc.de/wiki/doku.php?id=wizard&rev=1579074728

Last update: **2020/01/15 08:52**