

BiDiB-Monitor V 0.4.6.2

Dieses Programm basiert auf den Definitionen des BiDiB®-Bussystems von Wolfgang Kufer.

<http://www.bidib.org/>



Das ursprüngliche Ziel war es, die am Bus angeschlossenen Knoten zu erfassen, die Softwarestände zu prüfen, die Features eines Knotens anzuzeigen und einen Knoten innerhalb der Busstruktur zu „finden“.

Inzwischen sind einige Funktionen dazugekommen:

- Belegt Meldungen werden angezeigt. Incl. Adressen, soweit unterstützt.
- Features können geändert werden.
- Jeder Knoten kann mit einer kurzen Beschreibung versehen werden.
- Modelbahnuhr mit einstellbarer Geschwindigkeit / Anfangs- u. Endzeit / Wiederholung.
- Firmware-Update über den BiDiB-Bus.
- Booster ein-/ausschalten, Anzeige Stromverbrauch bei den Meldern und als Tabelle.
- Kleine Lok-Decoder-Verwaltung.
- Fahrpult für Vor-/Rückwärts, Halt, Nothalt und bis zu 28 Funktionstasten.
- CV-Verwaltung für die BiDiB-Knoten (**nicht** für Lok-Dekoder).
- Kehrschleife eines GBM16T über CV-Verwaltung konfigurieren.

Version vom 24.11.2013

Autor: Andreas Tillner

Supportanfragen bitte über das OpenDCC-Forum

<http://www.opendcc.de/forum.html>

eMail: a.tillner@gmx.de

This Tool is subject of the GNU general public license 2,

that is available at the world-wide-web at

<http://www.gnu.org/licenses/gpl.tx>

RailCom® und RailComPlus® sind eingetragene Warenzeichen der Firmen Lenz GmbH, Giessen bzw. ESU, Ulm

Inhalt

1	Installation.....	5
2	Aufruf.....	6
3	Konfiguration.....	7
3.1	Schnittstelle.....	7
3.2	Einstellungen des Monitors.....	8
3.2.1	BiDiB Log wird angezeigt	8
3.2.2	Alle Knoten bei DISCONNECT / RESET löschen.....	8
3.2.3	Technische Ansicht (Knoten)	8
3.2.4	Technische Ansicht (Features)	9
3.3	Benutzer Einstellungen.....	9
3.3.1	Menu mit BiDiB-Messages	9
3.3.2	Simulation ohne COM-Verbindung	10
3.3.3	Initiale Geschwindigkeit „0“ einschalten.....	10
3.3.4	Anzahl der gleichzeitig offenen Fahrpulte.....	11
3.3.5	DCC zusammen mit Booster ein- u. ausschalten.....	11
3.3.6	Watchdog aktiv.....	12
3.4	Einstellungen der BiDiB Logdatei	12
3.4.1	BiDiB-Log in Datei schreiben	12
3.4.2	Besser lesbare Darstellung der Messages	12
3.4.3	Technische und “lesbare” Darstellung der Messages	12
4	BiDiB-Funktionen.....	13
4.1	BiDiB-Menu	13
4.1.1	Verbinden	13
4.1.2	Verbindung lösen.....	13
4.1.3	Log-Bereich löschen.....	13
4.1.4	Refresh Knotenliste	13
4.1.5	Technische Ansicht	13
4.1.6	Listen	14
4.2	BiDiB-Kommandos.....	18
4.3	Funktionen im Bereich „BiDiB-Knoten“	19
4.3.1	Anzeige der Daten eines Knoten	19
4.3.2	Statusanzeige der Meldeeingänge eines Knoten	20
4.3.3	Statusanzeige der Switcheingänge eines Knoten	22

4.3.4	Features ändern	23
4.3.5	Software Update einspielen	25
4.3.6	Eigenschaften eines Knoten	28
4.3.7	Kontextmenü eines Knoten	31
4.4	Booster	34
4.4.1	Stromverbrauch.....	34
5	DCC-Steuerung	35
5.1	Decoder eingeben	35
5.2	Steuerung einer Lok	36
6	Funktionen für bestimmte Knoten/Module.....	37
6.1	Kehrschleife eines GBM16T konfigurieren.....	37
6.1.1	Welcher Melder ist mit welchem Abschnitt verbunden?	38
6.1.2	DCC-Adresse des GBM16T festlegen.....	39
6.1.3	Reverser Mode – Wodurch wird die Kehrschleife ausgelöst?.....	39
6.1.4	ON-Melder setzten	39
6.1.5	OFF-Melder setzten	40
6.1.6	Prioritätsmelder setzten.....	40
6.1.7	Inhibit On konfigurieren	40
6.1.8	Inhibit Off konfigurieren.....	41
7	POM / xPOM.....	42
7.1	POM via Decoder-ID	43
7.2	Operation „Bit schreiben“	43
8	Modelbahn-Uhr	44
8.1	Einstellungen	44
8.2	Uhr-Funktionen	45
8.2.1	Starten mit „Endlos“ eingeschaltet.	45
8.2.2	Starten mit „Endlos“ ausgeschaltet.....	45
8.2.3	Starten mit „Endlos“ ausgeschaltet und „Repeat“ aktiviert.	46
9	CV-Verwaltung für BiDiB-Knoten.....	47
9.1	CV's lesen.....	49
9.1.1	Alle CV's lesen.....	49
9.1.2	CV-Gruppen lesen.....	49
9.1.3	Einzelne CV lesen.....	49
9.2	CV's ändern.....	50

9.2.1	Allgemeine Datentypen.....	50
9.2.2	Datentypen für bestimmte Knoten	53
9.3	CV's in den Knoten schreiben.....	54
9.3.1	Alle CV's in BiBiB-Knoten schreiben	54
9.3.2	CV-Gruppe in BiBiB-Knoten schreiben	54
9.3.3	Einzelne CV in BiBiB-Knoten schreiben	54
9.4	CV's in Datei schreiben.....	55
9.5	CV's aus Datei lesen.....	55
9.6	CV-Definitionsdatei (XML)	56
9.6.1	Einem Knoten eine CV-Definitionsdatei zuweisen	56
9.6.2	Format der CV-Definitionsdatei.....	57
9.6.3	Template erstellen.....	61
9.6.4	Baumstruktur definieren	62
10	LogFile Analyse	64
10.1	LogFile-Analyse einblenden.....	64
10.2	LogFile-Analyse anwenden.....	65
10.2.1	Logdatei auswählen.....	65
10.2.2	Programmauswahl.....	65
10.2.3	Einzelschritt	66
10.2.4	Massenhafte Abarbeitung der Zeilen.....	67
11	Anhang A: Die XML-Dateien	68
11.1	BiDiB-Messages.xml	68
11.2	DCCManufacturer.xml	69
11.3	BiDiB-Decoder.xml.....	69
11.4	BiDiBProductID.xml	69
11.5	Products_<VendorID>.bidib	69
11.5.1	Beispiel:	69
11.6	FeatureCodes.bidib	70
11.6.1	Beispiel	70
11.7	BiDiB-NodeProperties.xml.....	71
11.8	XML-Datei zur CV-Definition	72
11.9	Format der XML-Datei zur Speicherung der CV-Werte	73
12	Anhang B: Versionsstand.....	74
13	Anhang C: Bekannte Probleme.....	80










1 Installation

VORAUSSETZUNGEN

Ab der Version V 0.4.0.1 basieren einige Funktionen auf „**.NET 4.0 Extendet**“.

Das „**.NET 4.0 Client Profile**“ reicht **NICHT** mehr, um den Monitor zu starten.

Entpacken der Datei BiDiB-Monitor.zip von <http://www.fichtelbahn.de/files/BiDiB-Monitor.zip> in ein Verzeichnis Ihrer Wahl.

Name	Änderungsdatum	Typ	Größe ▾
 data	30.06.2013 08:43	Dateiordner	
 de	30.06.2013 08:41	Dateiordner	
 de-DE	30.06.2013 08:41	Dateiordner	
 en	30.06.2013 08:41	Dateiordner	
 BiDiBLib.dll	07.06.2013 17:00	Anwendungser...	15 KB
 BiDiB-Monitor.exe	30.06.2013 08:36	Anwendung	1.110 KB
 BiDiBNodePlugIn.dll	26.05.2013 14:15	Anwendungser...	5 KB
 BiDiBPlugIn-Interface.dll	07.06.2013 17:00	Anwendungser...	6 KB
 ObjectListView.dll	02.06.2013 09:31	Anwendungser...	398 KB

Wichtig ist nur, dass alle Dateien im selben Verzeichnis stehen.

Die Unterverzeichnisse *de*, *de-DE* und *en* sind wichtig für die mehrsprachige Benutzeroberfläche ab Version *0.1.2.0*.

Ab Version *0.3.0.3* liegen alle XML-Dateien in dem Unterverzeichnis *data*.

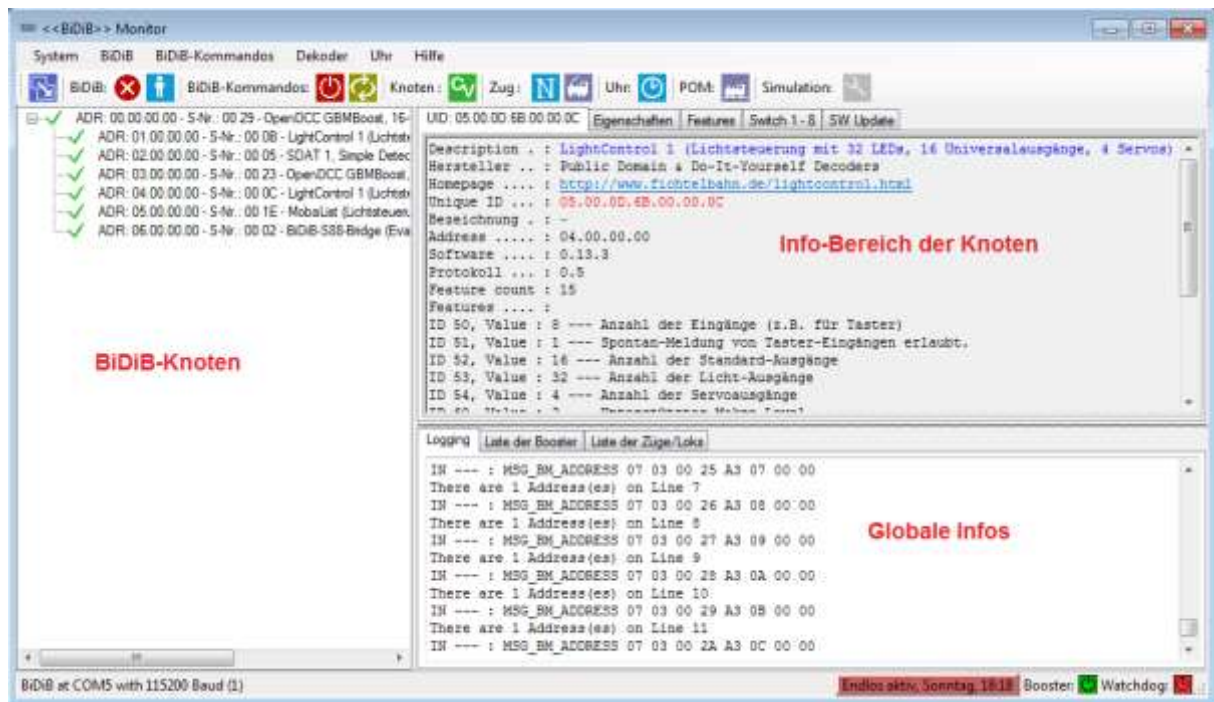
Ab Version *0.4.0.1* kommen die Dateien *ObjectListView.dll* und *BiDiBNodePlugIn.dll* dazu.

Ab Version *0.4.2.1* kommt die Datei *BiDiBPlugIn-Interface.dll* dazu.

2 Aufruf

Starten mit Doppelklick auf *BiDiB-Monitor.exe*.

Das Hauptfenster ist in drei Bereiche aufgeteilt.

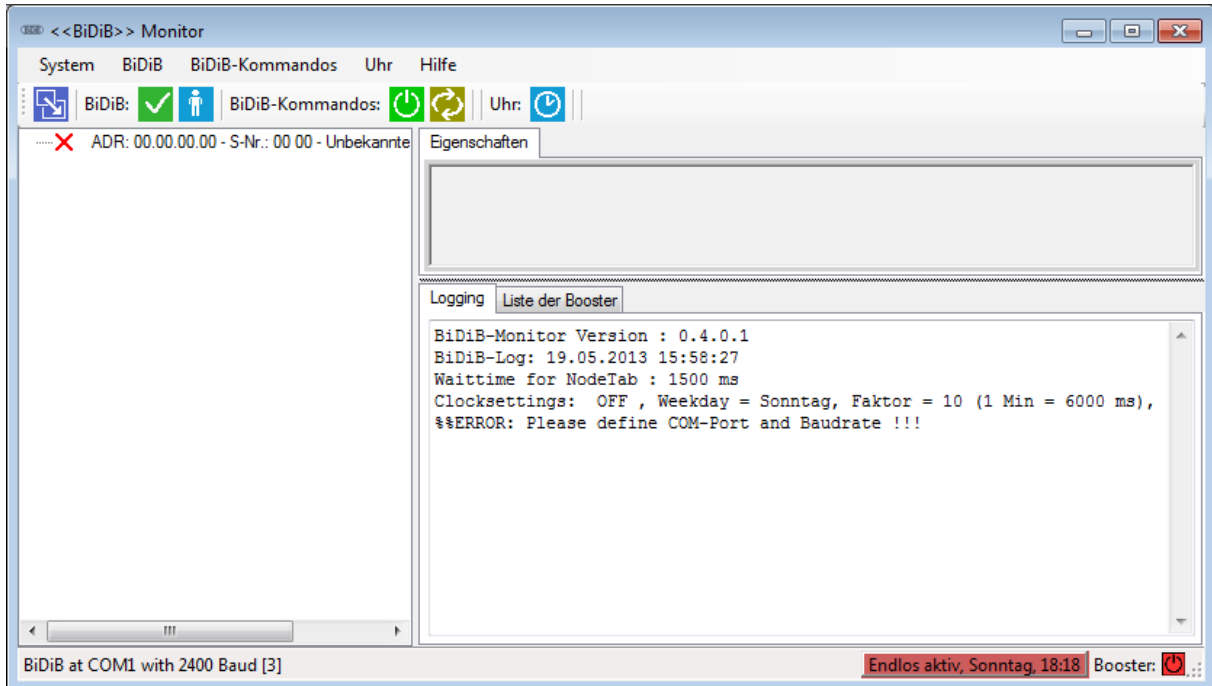


- Info-Bereich: Anzeige der Detaildaten der Knoten, Infolisten, Melder, usw.
- Globale Infos: Anzeige globaler Daten: BiDiB-Kommunikation, Booster-Liste, Decoder-Liste.
- BiDiB-Knoten: Alle gefundenen BiDiB-Knoten in einer Baum-Struktur.

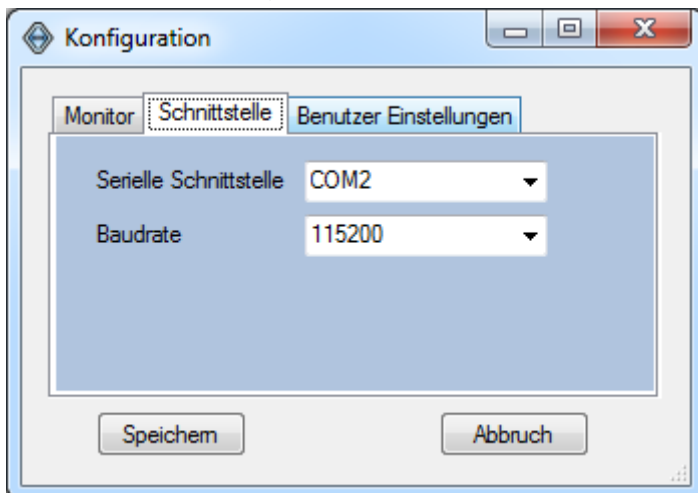
3 Konfiguration

3.1 Schnittstelle

Beim ersten Aufruf wird überprüft, ob eine serielle Schnittstelle konfiguriert ist. Ist dies nicht der Fall wird eine Meldung im Log-Bereich ausgegeben. In der Statuszeile steht dann „BiDiB at COM1 with 2400 Baud“



Über das Menu „System -> Konfiguration“ dann die Schnittstelle und Baudrate (wird noch nicht automatisch ermittelt) zum BiDiB-Interface einstellen und speichern.



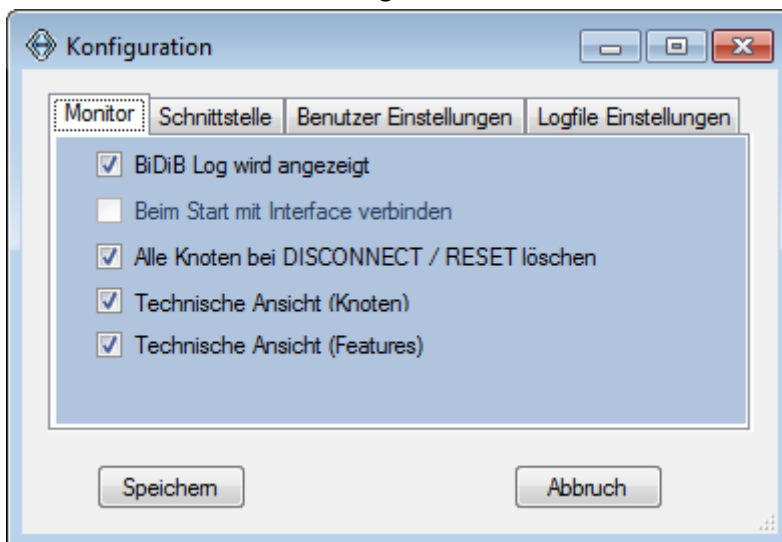
Wenn an COM-Port oder Baudrate Änderungen vorgenommen und per „Save“ bestätigt werden, öffnet sich folgendes Fenster:



Wenn hier auf OK geklickt wird, werden die neuen Einstellungen gespeichert, die Verbindung zum BiDiB-Interface wird **unterbrochen** und mit den geänderten Einstellungen neu aufgebaut.

3.2 Einstellungen des Monitors

Diese Einstellungen werden in der Registry des Computers abgespeichert und werden beim nächsten Aufruf vom Monitor wieder eingelesen.



3.2.1 BiDiB Log wird angezeigt

Wenn der Haken hier gesetzt ist, werden alle empfangenen und gesendeten BiDiB-Messages im Log-Bereich der Benutzeroberfläche ausgegeben (default).

3.2.2 Alle Knoten bei DISCONNECT / RESET löschen

Wenn der Haken hier gesetzt ist, wird bei DISCONNECT oder RESET der Baum mit den BiDiB-Knoten gelöscht.

3.2.3 Technische Ansicht (Knoten)

Hier wird zwischen der „Technischen“ und der „Anwenderfreundlichen“ Ansicht der Knotenliste umgeschaltet.

Diese Einstellung wird gespeichert und beim nächsten Start wieder verwendet.

3.2.4 Technische Ansicht (Features)

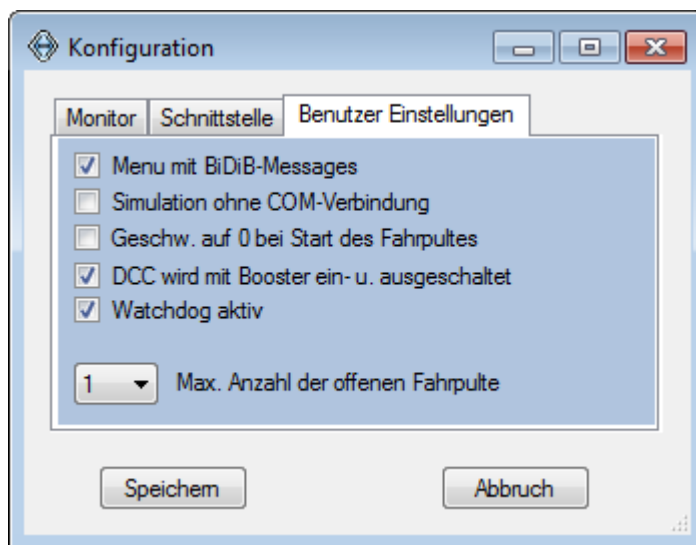
Hier wird zwischen der „Technischen“ und der „Anwenderfreundlichen“ Ansicht der Features umgeschaltet.

Diese Einstellung wird gespeichert und beim nächsten Start wieder verwendet.

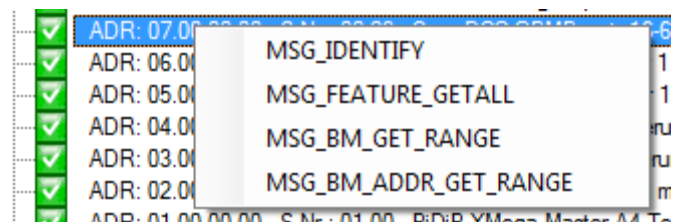
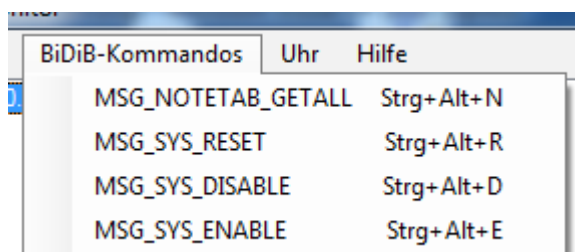
3.3 Benutzer Einstellungen

3.3.1 Menu mit BiDiB-Messages

Diese Einstellung wird in der Registry des Computers abgespeichert und beim nächsten Aufruf vom Monitor wieder eingelesen.

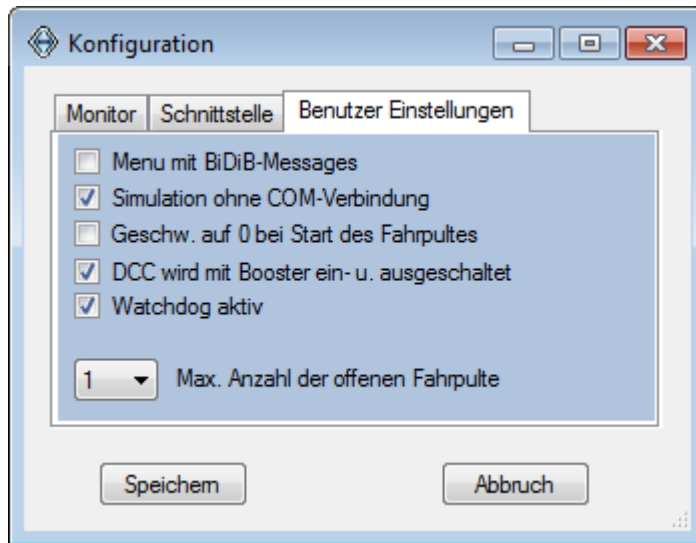


Wenn der Haken bei „Menu mit BiDiB-Messages“ gesetzt wird, werden das Kontextmenu der Knoten (siehe 4.3.7) und das Menu unter „BiDiB-Kommandos“ (siehe 4.2) nicht in sprechender Form, sondern als BiDiB-Messages dargestellt:



Diese Einstellung ist **NICHT** dynamisch. Der Monitor muss neu gestartet werden, damit diese Einstellung verwendet wird.

3.3.2 Simulation ohne COM-Verbindung

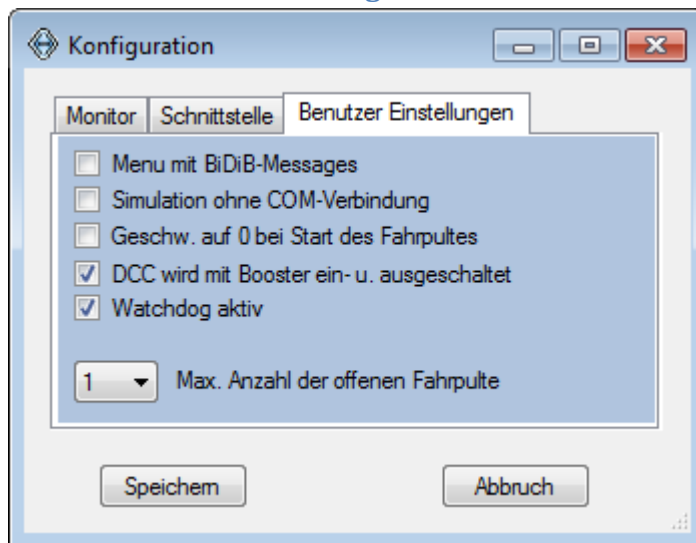


Wird der Haken bei „Simulation ohne COM-Verbindung“ gesetzt, können alle Funktionen ausgeführt werden, ohne dass Messages an das Interface gesendet werden.

Die COM-Schnittstelle wird NICHT geöffnet.

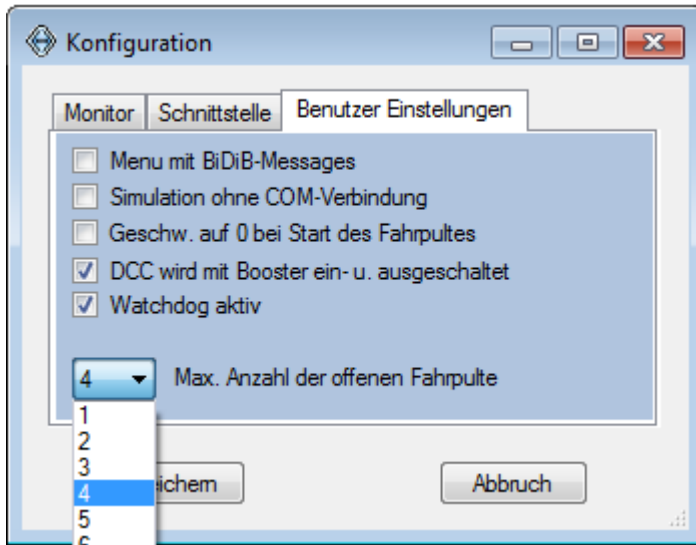
Achtung: Diese Einstellung wird NICHT gespeichert und ist bei jedem Start des Monitors wieder ausgeschaltet.

3.3.3 Initiale Geschwindigkeit „0“ einschalten



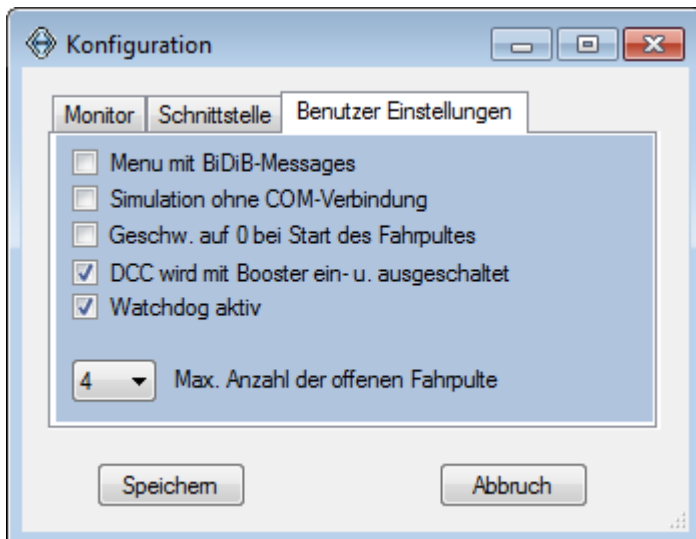
Ist der Haken bei „Geschw. auf 0 bei Start des Fahrpultes“ gesetzt, wird an den ausgewählten Decoder das Kommando für die Geschwindigkeitsstufe 0 gesendet.

3.3.4 Anzahl der gleichzeitig offenen Fahrpulte



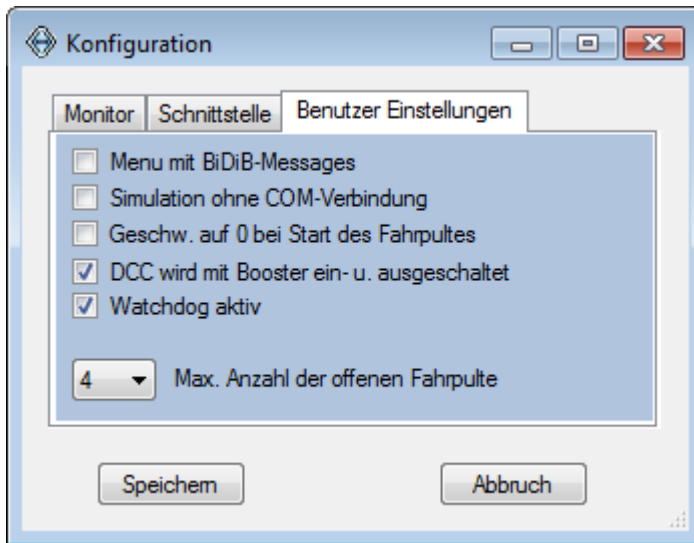
Hier kann die Anzahl der maximal gleichzeitig offenen Fahrpulte eingestellt werden.

3.3.5 DCC zusammen mit Booster ein- u. ausschalten



Wird der Haken hier gesetzt, wird mit dem Booster auch das DCC-Signal ein- u. Ausgeschaltet.

3.3.6 Watchdog aktiv

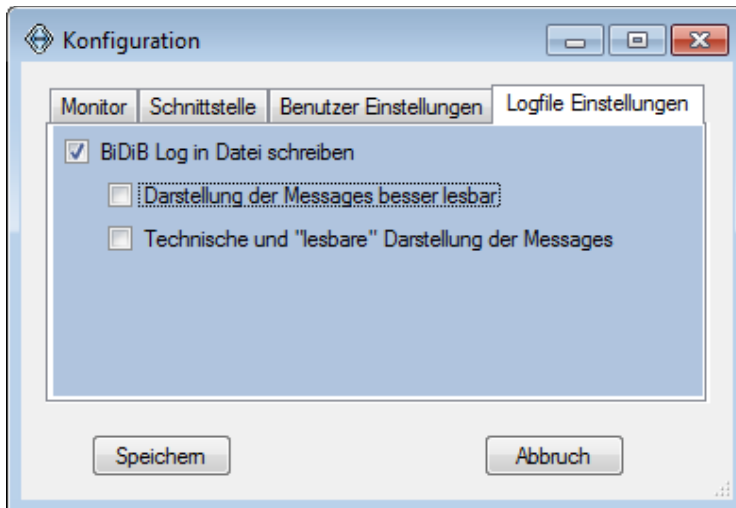


Wird der Haken bei „Watchdog aktiv“ gesetzt, wird das Feature 101 der GBMBoost ausgewertet und der Watchdog entsprechend gestartet. Ist der Haken nicht gesetzt, wird das Feature ignoriert und der Watch nicht gestartet.

Siehe auch die „OpenDCC GBM in Aktion“ unter

http://www.fichtelbahn.de/files/OpenDCC_GBM_in_aktion.pdf

3.4 Einstellungen der BiDiB Logdatei



3.4.1 BiDiB-Log in Datei schreiben

Wenn der Haken hier gesetzt ist, werden alle empfangenen und gesendeten BiDiB-Messages in eine Log-Datei geschrieben. Bei jedem Start des BiDiB-Monitors wird eine neue Datei mit dem Namen „BiDiB-Monitor-YYYYMMTT-HH.MM.SS.log“ im Verzeichnis „Eigene Dokumente“ angelegt.

Beispiel: *BiDiB-Monitor-20120916-13.48.24.log*

3.4.2 Besser lesbare Darstellung der Messages

Die protokollierten Messages werden folgendermaßen dargestellt:

```
[20131030081013604] Input : |05 00 13 90 00 00 73 FE |
```

```
[20131030081013604] IN ---: MSG_FEATURE          Cnt: 05, Addr: 00, Nr.: 019, Msg: 0x90, Param: (0x00, 0x00)
```

3.4.3 Technische und "lesbare" Darstellung der Messages

Die protokollierten Messages werden folgendermaßen dargestellt:

```
[20131030183912915] Input : |05 00 1D 90 0C 0A 55 FE |
```

```
[20131030183912915] IN ---: MSG_FEATURE          05 00 1D 90 0C 0A
```

```
[20131030183912915] IN ---: MSG_FEATURE          Cnt: 05, Addr: 00, Nr.: 029, Msg: 0x90, Param: (0x0c, 0x0a)
```

4 BiDiB-Funktionen

4.1 BiDiB-Menu



4.1.1 Verbinden

Verbindung zum Interface wird aufgebaut (Strg + Umschalttaste + C). Feature des Interface und Knotentabelle wird abgefragt.

4.1.2 Verbindung lösen

Verbindung zum Interface wird beendet (Strg + Umschalttaste + D).

4.1.3 Log-Bereich löschen

Der Inhalt des Log-Bereiches wird gelöscht (Strg + Umschalttaste + X)

4.1.4 Refresh Knotenliste

Mit <F5> wird ein Refresh auf die Ansicht der Knotenliste durchgeführt.

4.1.5 Technische Ansicht

Mit <F4> oder durch Anklicken, der unten beschriebenen Icons in der Menüleiste, wird die „Technische Ansicht“ aus- bzw. eingeschaltet.



Schaltet die „beschreibende Ansicht“ ein.



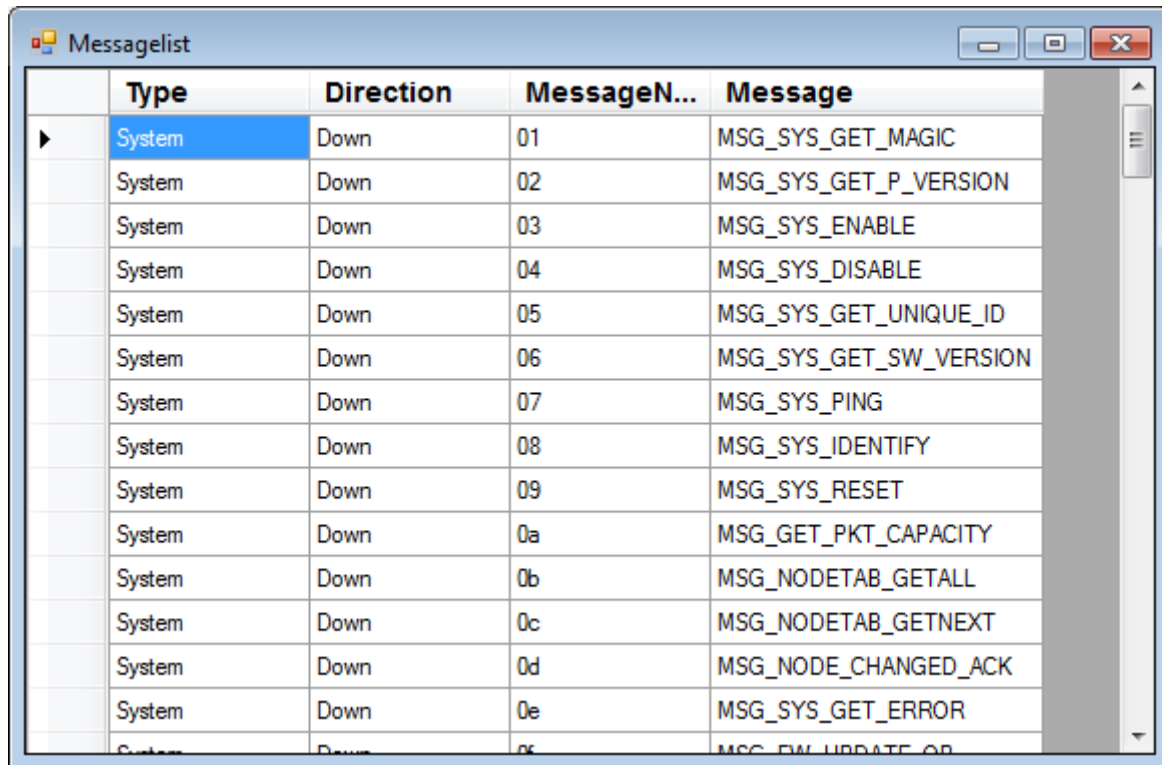
Schaltet um auf die „technische Ansicht“

Die durch <F4>, bzw. die Icons eingestellte Ansicht, wird nicht permanent in der Registry gespeichert.

4.1.6 Listen

4.1.6.1 Liste der Messages

Die Liste aller definierten BiDiB-Messages wird als Tabelle ausgegeben. Quelle ist die Datei *BiDiB-Messages.xml*.

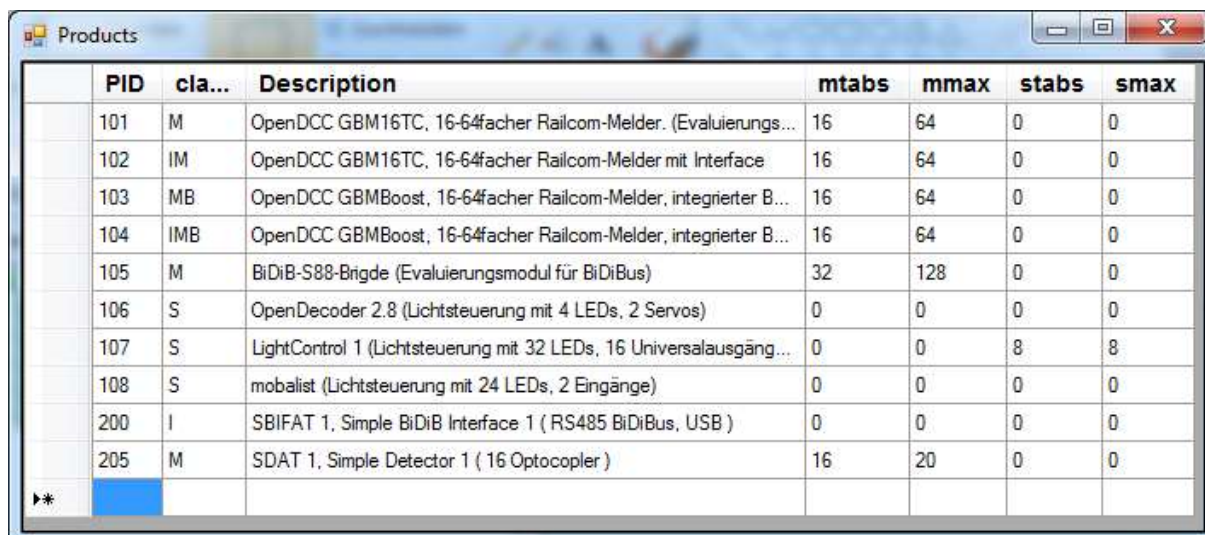


The screenshot shows a window titled 'Messagelist' with a table containing the following data:

	Type	Direction	MessageN...	Message
▶	System	Down	01	MSG_SYS_GET_MAGIC
	System	Down	02	MSG_SYS_GET_P_VERSION
	System	Down	03	MSG_SYS_ENABLE
	System	Down	04	MSG_SYS_DISABLE
	System	Down	05	MSG_SYS_GET_UNIQUE_ID
	System	Down	06	MSG_SYS_GET_SW_VERSION
	System	Down	07	MSG_SYS_PING
	System	Down	08	MSG_SYS_IDENTIFY
	System	Down	09	MSG_SYS_RESET
	System	Down	0a	MSG_GET_PKT_CAPACITY
	System	Down	0b	MSG_NODETAB_GETALL
	System	Down	0c	MSG_NODETAB_GETNEXT
	System	Down	0d	MSG_NODE_CHANGED_ACK
	System	Down	0e	MSG_SYS_GET_ERROR
	Custom	Down	0f	MSG_FW_UPDATE_OK

4.1.6.2 Liste der Produkte

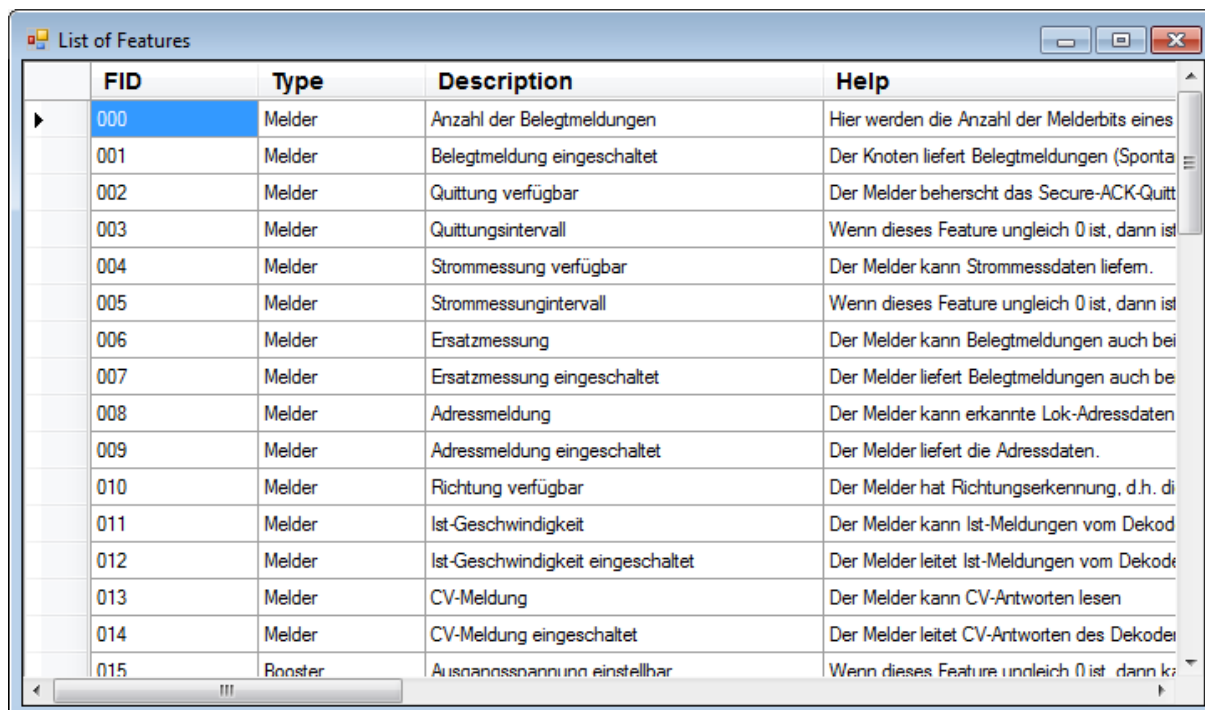
Die Liste der definierten Produkte wird als Tabelle ausgegeben. Produkte, die hier aufgelistet sind, werden in der Detailansicht eines Knotens mit der entsprechenden Beschreibung dargestellt. Quelle ist die Datei [BiDiBProductID.xml](#).



	PID	cla...	Description	mtabs	mmax	stabs	smax
	101	M	OpenDCC GBM16TC, 16-64facher Railcom-Melder. (Evaluierungs...	16	64	0	0
	102	IM	OpenDCC GBM16TC, 16-64facher Railcom-Melder mit Interface	16	64	0	0
	103	MB	OpenDCC GBMBoost, 16-64facher Railcom-Melder, integrierter B...	16	64	0	0
	104	IMB	OpenDCC GBMBoost, 16-64facher Railcom-Melder, integrierter B...	16	64	0	0
	105	M	BiDiB-S88-Bridge (Evaluierungsmodul für BiDiBus)	32	128	0	0
	106	S	OpenDecoder 2.8 (Lichtsteuerung mit 4 LEDs, 2 Servos)	0	0	0	0
	107	S	LightControl 1 (Lichtsteuerung mit 32 LEDs, 16 Universalausgäng...	0	0	8	8
	108	S	mobalist (Lichtsteuerung mit 24 LEDs, 2 Eingänge)	0	0	0	0
	200	I	SBIFAT 1, Simple BiDiB Interface 1 (RS485 BiDiBus, USB)	0	0	0	0
	205	M	SDAT 1, Simple Detector 1 (16 Optocopler)	16	20	0	0
▶▶							

4.1.6.3 Liste der Features

Die Liste der definierten Features wird als Tabelle ausgegeben. Features, die hier aufgelistet sind, werden in der Detailansicht eines Knotens mit der entsprechenden Beschreibung dargestellt. Quelle ist die Datei [BiDiBFeatures.xml](#).



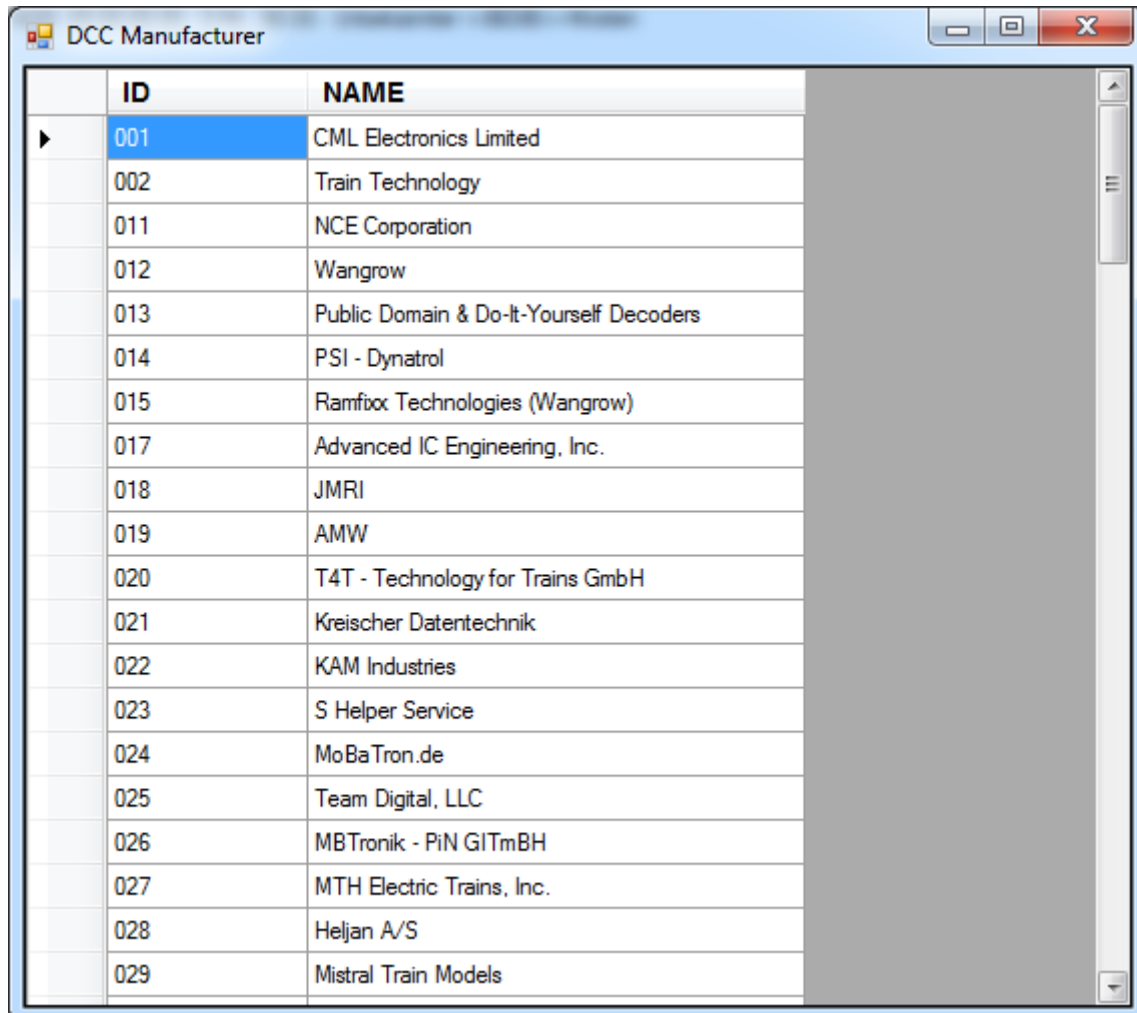
	FID	Type	Description	Help
▶	000	Melder	Anzahl der Belegtmeldungen	Hier werden die Anzahl der Melderbits eines
	001	Melder	Belegtmeldung eingeschaltet	Der Knoten liefert Belegtmeldungen (Sponta
	002	Melder	Quittung verfügbar	Der Melder beherrscht das Secure-ACK-Quitt
	003	Melder	Quittungsintervall	Wenn dieses Feature ungleich 0 ist, dann ist
	004	Melder	Strommessung verfügbar	Der Melder kann Strommessdaten liefern.
	005	Melder	Strommessungsintervall	Wenn dieses Feature ungleich 0 ist, dann ist
	006	Melder	Ersatzmessung	Der Melder kann Belegtmeldungen auch bei
	007	Melder	Ersatzmessung eingeschaltet	Der Melder liefert Belegtmeldungen auch bei
	008	Melder	Adressmeldung	Der Melder kann erkannte Lok-Adressdaten
	009	Melder	Adressmeldung eingeschaltet	Der Melder liefert die Adressdaten.
	010	Melder	Richtung verfügbar	Der Melder hat Richtungserkennung, d.h. di
	011	Melder	Ist-Geschwindigkeit	Der Melder kann Ist-Meldungen vom Dekod
	012	Melder	Ist-Geschwindigkeit eingeschaltet	Der Melder leitet Ist-Meldungen vom Dekod
	013	Melder	CV-Meldung	Der Melder kann CV-Antworten lesen
	014	Melder	CV-Meldung eingeschaltet	Der Melder leitet CV-Antworten des Dekode
	015	Rooster	Ausgangsspannung einstellbar	Wenn dieses Feature ungleich 0 ist, dann k

4.1.6.4 Liste der DCC-Hersteller

Die Liste der DCC-Hersteller wird als Liste ausgegeben. Quelle ist die XML-Datei

DCCManufacturer.xml.

Stand der Liste ist die NMRA-Website <http://www.nmra.org/standards/DCC/mfgnumbers.html> vom 27.10.2012.



	ID	NAME
▶	001	CML Electronics Limited
	002	Train Technology
	011	NCE Corporation
	012	Wangrow
	013	Public Domain & Do-It-Yourself Decoders
	014	PSI - Dynatrol
	015	Ramfixx Technologies (Wangrow)
	017	Advanced IC Engineering, Inc.
	018	JMRI
	019	AMW
	020	T4T - Technology for Trains GmbH
	021	Kreischer Datentechnik
	022	KAM Industries
	023	S Helper Service
	024	MoBaTron.de
	025	Team Digital, LLC
	026	MBTronik - PiN GITmBH
	027	MTH Electric Trains, Inc.
	028	Heljan A/S
	029	Mistral Train Models

4.1.6.5 Liste der BiDiB-Knoten

Inventory (Strg + Umschalttaste + I). Im Info-Bereich werden alle bekannten Knoten mit den gefundenen Daten aufgelistet.

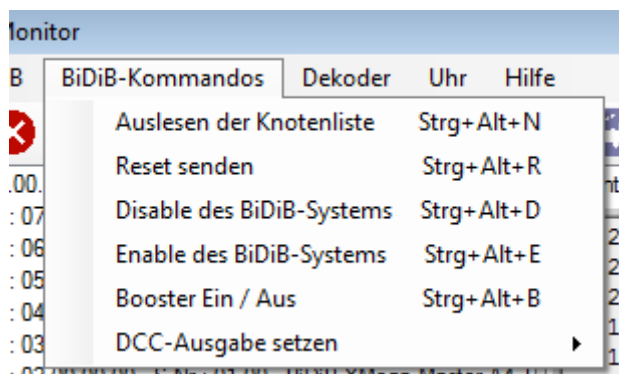
```
Inventory
Description . : OpenDCC GBMBoost, 16-64facher Railcom-Melder, integrierter Booster, Interface
Hersteller .. : Public Domain & Do-It-Yourself Decoders
Unique ID ... : D2.00.0D.68.00.00.29
Bezeichnung . : Master Interface
Address ..... : 00.00.00.00
Software .... : 2.0.5
Protokoll ... : 0.5
Feature count : 23
ID 0, Value : 16 --- Anzahl der Belegtmeldungen
ID 1, Value : 1 --- Belegtmeldung eingeschaltet
ID 2, Value : 1 --- Quittung verfügbar
ID 3, Value : 1 --- Quittungsintervall
ID 8, Value : 1 --- Adressmeldung
ID 9, Value : 1 --- Adressmeldung eingeschaltet
ID 10, Value : 1 --- Richtung verfügbar
ID 13, Value : 1 --- CV-Meldung
ID 14, Value : 1 --- CV-Meldung eingeschaltet
```

Beispiel:

```
Description . : OpenDCC GBMBoost, 16-64facher Railcom-Melder, integrierter Booster, Interface
Hersteller .. : Public Domain & Do-It-Yourself Decoders
Unique ID ... : D2.00.0D.68.00.00.29
Bezeichnung . : Master Interface
Address ..... : 00.00.00.00
Software .... : 2.0.5
Protokoll ... : 0.5
Feature count : 23
ID 0, Value : 16 --- Anzahl der Belegtmeldungen
ID 1, Value : 1 --- Belegtmeldung eingeschaltet
ID 2, Value : 1 --- Quittung verfügbar
ID 3, Value : 1 --- Quittungsintervall
ID 8, Value : 1 --- Adressmeldung
ID 9, Value : 1 --- Adressmeldung eingeschaltet
ID 10, Value : 1 --- Richtung verfügbar
ID 13, Value : 1 --- CV-Meldung
ID 14, Value : 1 --- CV-Meldung eingeschaltet
ID 11, Value : 1 --- Ist-Geschwindigkeit
ID 12, Value : 10 --- Ist-Geschwindigkeit eingeschaltet
ID 17, Value : 1 --- Cutout verfügbar
ID 18, Value : 1 --- Cutout eingeschaltet
ID 19, Value : 8 --- Abschaltzeit normal
ID 20, Value : 30 --- Abschaltzeit Inrush
ID 21, Value : 1 --- Ausgangsstrom einstellbar
ID 22, Value : 155 --- max. Ausgangsstrom
ID 23, Value : 0 --- FEATURE_BST_CURMEAS_INTERVAL
ID 26, Value : 0 --- Booster Autostart verhindern
ID 27, Value : 0 --- Booster lokale Kontrolle verhindern
ID 107, Value : 1 --- Kontrolle des DCC-Busses
ID 108, Value : 1 --- Zentrale kann verlorene Loks melden
ID 109, Value : 1 --- Zentrale meldet Handbedienung
-----
Description . : BiDiB-S88-Bridge (Evaluierungsmodul für BiDiBus)
Hersteller .. : Public Domain & Do-It-Yourself Decoders
Unique ID ... : 40.00.0D.69.00.00.02
Bezeichnung . : -
Address ..... : 07.00.00.00
Software .... : 0.3.1
Protokoll ... : 0.4
Feature count : 4
ID 0, Value : 128 --- Anzahl der Belegtmeldungen
ID 1, Value : 1 --- Belegtmeldung eingeschaltet
ID 2, Value : 1 --- Quittung verfügbar
ID 3, Value : 0 --- Quittungsintervall
```

4.2 BiDiB-Kommandos

Unter dem Menu „BiDiB-Kommandos“ können BiDiB-Messages direkt an das Interface geschickt werden.



Hinter den Menüpunkten verbergen sich folgende Messages:

- **MSG_NOTETAB_GETALL** (Strg + Alt + N)
- **MSG_SYS_RESET** (Strg + Alt + R)
- **MSG_SYS_DISABLE** (Strg + Alt + D)
- **MSG_SYS_ENABLE** (Strg + Alt + E)
- **MSG_BOOST_ON / OFF** (Strg + Alt + B)
- **MSG_CS_SET_STATE** (siehe unten)

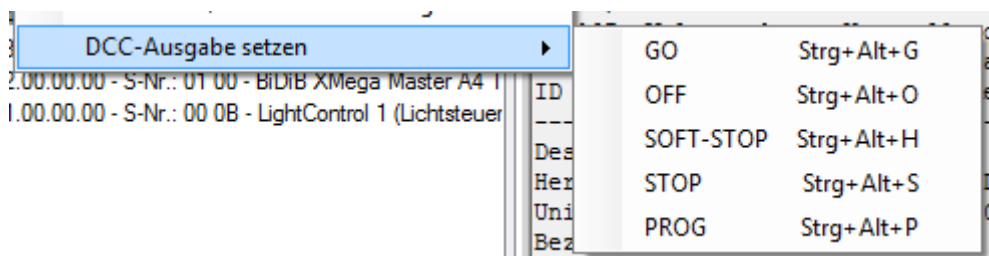
Bei **MSG_NOTETAB_GETALL** wird die Knotentabelle gelesen und an jeden Knoten noch folgende Messages geschickt und entsprechend ausgewertet:

- **MSG_GET_SW_VERSION**
- **MSG_GET_P_VERSION**
- **MSG_FEATURE_GETALL**

Die gemeldeten Knoten werden in der Baumstruktur eingetragen bzw. bei bereits vorhandenen Knoten, die evtl. noch fehlenden Infos ergänzt.

Bei **MSG_BOOST_ON / OFF** wird der Booster ein- bzw. ausgeschaltet. Je nach dem, welchen Zustand er vorher hatte.

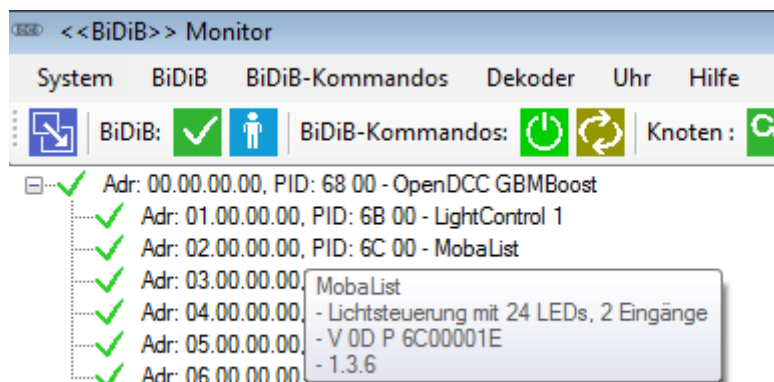
Unter „**DCC-Ausgabe setzen**“ kann der entsprechende DCC-Status eingestellt werden.



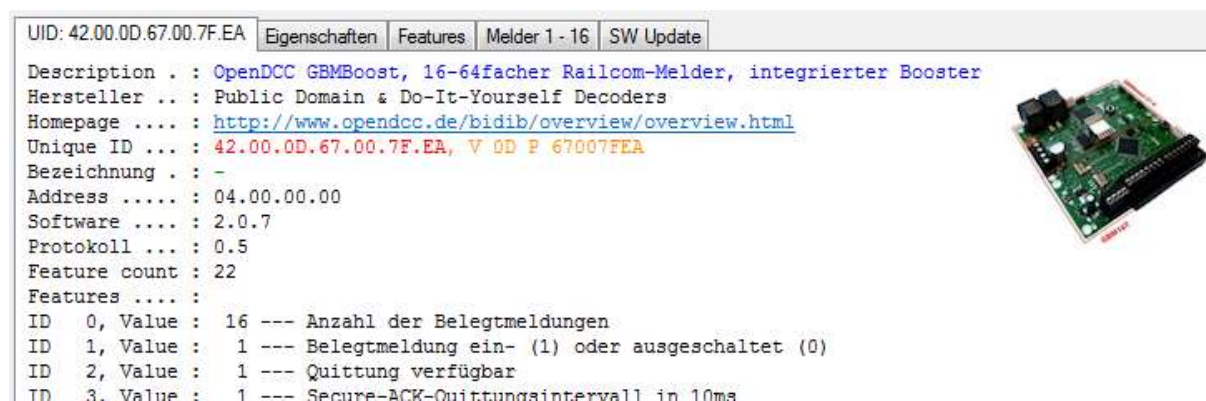
4.3 Funktionen im Bereich „BiDiB-Knoten“

4.3.1 Anzeige der Daten eines Knoten

Wenn Sie mit der Maus über die Knoten in der Baumstruktur fahren, wird die Beschreibung, die UID und die Version der Firmware, des jeweiligen Knoten als „ToolTip“ angezeigt.



Durch Anklicken eines der Knoten in der Baumstruktur mit der linken Maustaste, werden im „Info-Bereich“ alle vorhandenen Daten des Knotens im ersten Tab aufgelistet:



„Description“ wird aus der Datei [BiDiBProducts.xml](#) anhand des PID-Anteils und der VendorID gelesen. Wird die gemeldete Kombination in der Datei nicht gefunden, wird hier „Unbekannter <<BiDiB>>-Knoten“ angezeigt.

Die Angabe „Bezeichnung“ ist keine BiDiB-Information (siehe 4.3.6.2).

Das angezeigte Bild muss in dem Verzeichnis `<wo der Monitor liegt>\data\images` liegen.

Der Dateiname muss folgendes Format haben:

[Bidib-<vendorID in int>-<\(ProduktID\) in int>.png](#)

Diese Dateinamen entsprechen denen in RocRail. Die bei RocRail hinterlegten Bilder können verwendet werden.

4.3.2 Statusanzeige der Meldeeingänge eines Knoten

Wenn der angeklickte Knoten Belegt Meldungen liefert (Auswertung der Feature-ID 0), wird eine entsprechende Karteikarte angelegt „Melder 1 – n“ und dort eine Liste der möglichen Meldereingänge angezeigt.

Die Tabelle mit den Meldereingängen wird in Abhängigkeit der Features des jeweiligen Knotens erstellt.

4.3.2.1 Belegtmeldung

Rot zeigt dass der Melder belegt ist.

TC, e D e D e D e D

UID: 40.00.0D.CD.00.00.02 Melder 1 - 16

0. Melder	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
*																

```

Input : |07 05 00 19 A2 08 08 11 A8 FE |
IN --- : 07 05 00 19 A2 08 08 11 .....MSG_BM_MULTIPLE
OUT -- : FE 07 05 00 18 21 08 08 11 34 FE .....MSG_BM_MIRROR_MULTIPLE
Input : |05 05 00 1A |
Input : ||
Input : |A1 08 F1 FE |
IN --- : 05 05 00 1A A1 08 .....MSG_BM_FREE
OUT -- : FE 05 05 00 19 23 08 AB FE .....MSG_BM_MIRROR_FREE
Input : |05 05 00 1B A0 08 9E FE 07 05 00 1C A2 08 |
IN --- : 05 05 00 1B A0 08 .....MSG_BM_OCC
OUT -- : FE 05 05 00 1A 22 08 8B FE .....MSG_BM_MIRROR_OCC
Input : |08 11 7A FE |
IN --- : 07 05 00 1C A2 08 08 11 .....MSG_BM_MULTIPLE
OUT -- : FE 07 05 00 1B 21 08 08 11 7A FE .....MSG_BM_MIRROR_MULTIPLE

```

4.3.2.2 Adressmeldung

Wenn auf dem Knoten das Feature 9 gesetzt ist, wird die Tabelle der Melder um eine Adresszeile erweitert.

```

Interface | Eigenschaften | Features | Melder 1 - 16
Description . : OpenDCC GBM16TC, 16-64facher Railcom-Melder. (Evaluierungsmodul, Stand-Alone)
Unique ID ... : C0.00.0D.65.00.01.00
Bezeichnung . : Interface
Address ..... : 00.00.00.00
Software .... : 1.3.6
Protokoll ... : 0.4
Feature count : 9
Features .... :
ID 14, Value : 1 --- CV-Meldung eingeschaltet
ID 13, Value : 1 --- CV-Meldung
ID 10, Value : 1 --- Richtung verfügbar
ID 9, Value : 1 --- Adressmeldung eingeschaltet
ID 8, Value : 1 --- Adressmeldung
ID 3, Value : 1 --- Quittungsintervall
ID 2, Value : 1 --- Quittung verfügbar
ID 1, Value : 1 --- Belegtmeldung eingeschaltet
ID 0, Value : 16 --- Anzahl der Belegtmeldungen

```

Werden also, wie in diesem Beispiel, 16 Belegtmeldungen mit Adressmeldung und Aufgleisrichtung, vom Knoten „gemeldet“, sieht die Tabelle so aus.

UID: C0.00.0D.68.00.01.00		Eigenschaften	Features	Melder 1 - 16															
►	Melder	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
	Adresse		00090 00012 00009		00070														
	Richtung		>> >> >>		>>														
*																			

Aufgleisrichtung: << = Rückwärts, >> = Vorwärts, AA = Accessory-Adresse, EA = Extended Accessory

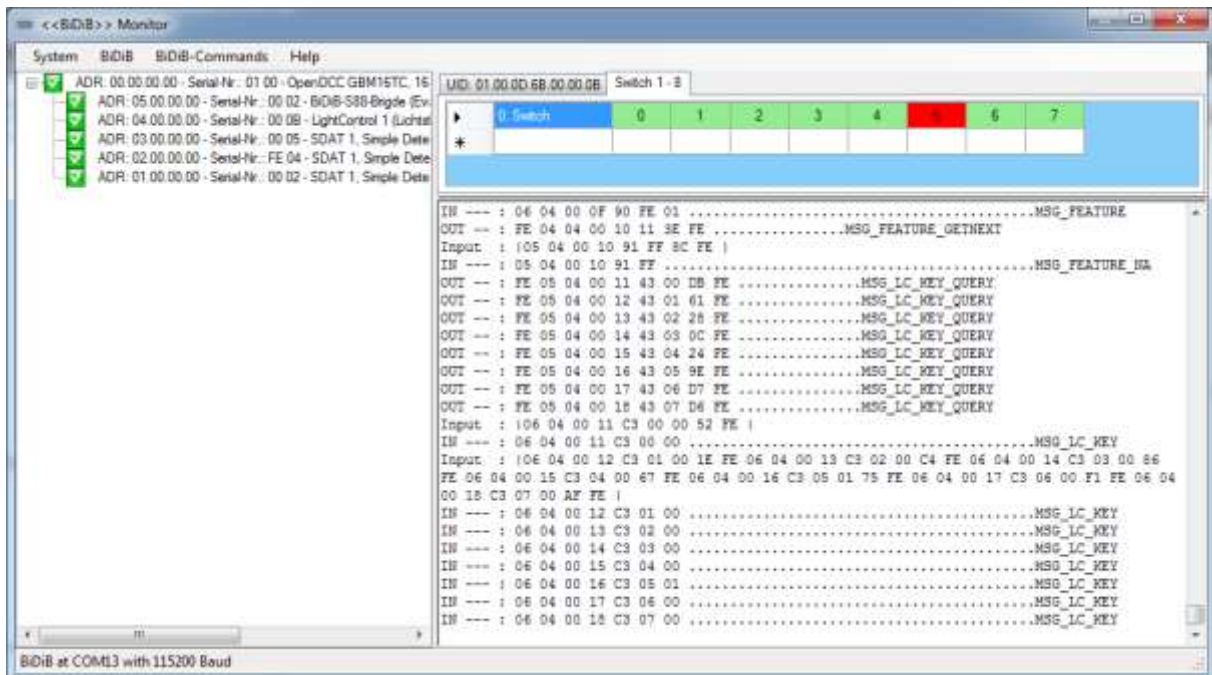
Hier belegen die Decoder mit der Adresse 9, 12 u. 90 den Abschnitt 1 mit Aufgleisrichtung „vorwärts“ (>> = Vorwärts, << = Rückwärts). Und ein Decoder mit Adresse 70 den Abschnitt 3.

Die BiDiB-Message MSG_BM_ADDRESSE liefert bis zu 4 Adressen.

4.3.3 Statusanzeige der Switcheingänge eines Knoten

Wenn der angeklickte Knoten Switcheingänge liefert (Auswertung der Feature-ID 50), wird eine entsprechende Karteikarte angelegt „Switch 1 – n“ und dort eine Liste der möglichen Switcheingänge angezeigt.

ACHTUNG: Für diese Funktion muss die Datei *BiDiBProductID.xml* erweitert werden (siehe 11.4). Rot zeigt dass der Switch betätigt ist.



Danach sieht die Feature-Liste dann so aus.

```
Description . : OpenDCC GBM16TC, 16-64facher Railcom-Melder. (Evaluierungsmodul, Stand-Alone)
Hersteller .. : Public Domain & Do-It-Yourself Decoders
Unique ID ... : C0.00.0D.65.00.01.00
Address ..... : 00.00.00.00
Software .... : 1.3.6
Protokoll ... : 0.4
Feature count : 9
Features ..... :
ID 14, Value : 1 --- CV-Meldung eingeschaltet
ID 13, Value : 1 --- CV-Meldung
ID 10, Value : 1 --- Richtung verfügbar
ID 9, Value : 1 --- Adressmeldung eingeschaltet
ID 8, Value : 1 --- Adressmeldung
ID 3, Value : 1 --- Quittungsintervall
ID 2, Value : 1 --- Quittung verfügbar
ID 1, Value : 1 --- Belegtmeldung eingeschaltet
ID 0, Value : 16 --- Anzahl der Belegtmeldungen
```


4.3.5 Software Update einspielen

Wenn ein BiDiB-Knoten das Feature 254 (Firmware-Update-Mode) auf 1 gesetzt hat, kann über den BiDiB-Bus ein Update der Firmware durchgeführt werden.

4.3.5.1 Firmware

Der BiDiB-Monitor legt dann für diesen Knoten eine weitere Karteikarte an.

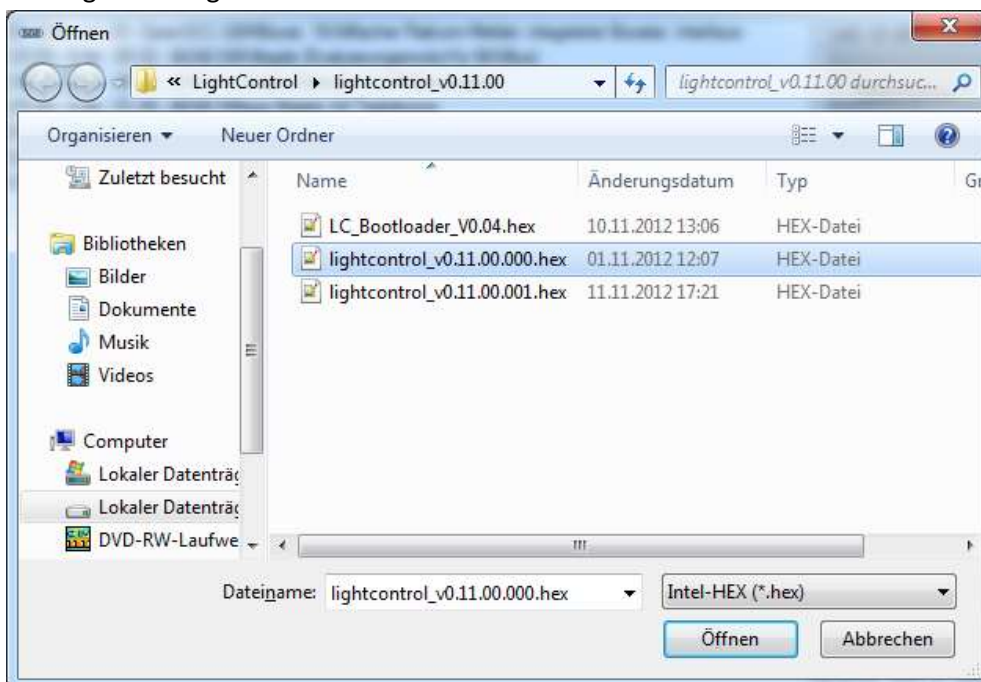
The screenshot shows the 'SW Update' tab in the BiDiB-Monitor software. At the top, there are tabs for 'Eigenschaften', 'Features', 'Switch 1 - 8', and 'SW Update'. The 'SW Update' tab is active, displaying the following information:

- Description:** LightControl 1 (Lichtsteuerung mit 32 LEDs, 16 Universalausgänge, 4 Servos)
- Hersteller:** Public Domain & Do-It-Yourself Decoders
- Unique-ID:** 05.00.0D.6B.00.00.18
- SW-Version:** 0.12.3

Below this information is a 'Start Update' button. To the left of this button is a 'Zielspeicher' (Target Storage) section with two options: 'Flash (Programmspeicher)' and 'EEPROM'. A 'HEX-Datei' button is located below the SW-Version field. At the bottom of the tab is a large button labeled 'Update-Mode beenden und Node neu starten'. At the very bottom of the window is an 'Update:' progress bar.

Hier werden die Bezeichnung, der Hersteller, die Unique-ID und die Version der zurzeit eingespielten Software angezeigt.

Bis auf die Dateiauswahl (HEX-Datei), sind alle weiteren Bedienelemente gesperrt. Durch Mausklick auf den Button „Hex-Datei“ öffnet sich ein Dateiauswahlfenster. Es können nur Dateien mit der Endung .HEX ausgewählt werden.



Hier wird eine Datei für den Lightcontrol ausgewählt. Der Monitor erkennt an dem Dateinamen mit der Endung .000.hex das es sich um eine Datei für den Flash-Speicher handelt. Der entsprechende „Zielspeicher“ wird automatisch ausgewählt.

Der Button „Update starten“ wird freigegeben und seine Farbe ändert sich in „grün“.

UID: 05.00.0D.6B.00.00.18 | Eigenschaften | Features | Switch 1 - 8 | SW Update

Description: LightControl 1 (Lichtsteuerung mit 32 LEDs, 16 Universalausgänge, 4 Servos)

Hersteller: Public Domain & Do-It-Yourself Decoders

Unique-ID: 05.00.0D.6B.00.00.18

SW-Version: 0.12.3

HEX-Datei: D:\Data\Eisenbahn\Digital-Technik\OpenDCC\LightControl\V0.12.03\lightcontrol_v0.12.03.000.hex

Zielspeicher:
☒ Flash (Programmspeicher)
☐ EEPROM

Start Update

Update-Mode beenden und Node neu starten

Update:

Durch Anklicken von „Update starten“ beginnt das Einspielen der Firmware.

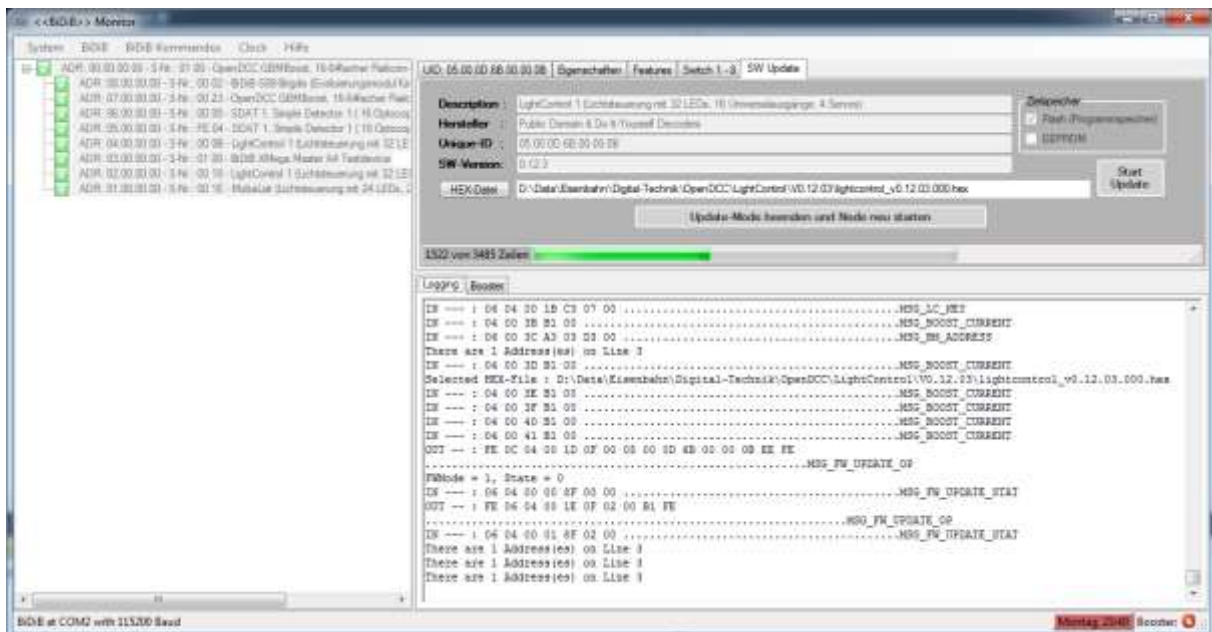
ACHTUNG: Um den Update-Vorgang so sicher wie möglich zu machen, sind jetzt **alle** Bedienelemente des Monitors deaktiviert. Auch der Knotenbaum und das Menu.

Am unteren Rand der Karteikarte wird die Anzahl der Zeilen angezeigt, die an den Knoten übertragen wurden.

Im Logbereich wird nur der Beginn der „Update-Kommunikation“ angezeigt.

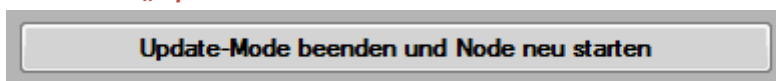
```
Selected HEX-File : D:\Data\Eisenbahn\Digital-Technik\OpenDCC\LightControl\lightcontrol_v0.11.00\lightcontrol_v0.11.00.000.hex
OUT -- : FE 0C 02 00 19 0F 00 01 00 0D 6B 00 00 0B 60 FE
.....MSG_FW_UPDATE_OP
IN --- : 06 02 00 00 8F 00 00 .....MSG_FW_UPDATE_STAT
OUT -- : FE 06 02 00 1A 0F 02 00 0D FE
.....MSG_FW_UPDATE_OP
IN --- : 06 02 00 01 8F 02 00 .....MSG_FW_UPDATE_STAT
```

Die einzelnen BiDiB-Kommandos der Übertragung werden NICHT im Log-Bereich mitgeschrieben.



Nach dem alle Zeilen erfolgreich an den Knoten übertragen wurden, wird am unteren Rand der Karteikarte eine entsprechende Meldung ausgegeben.

Der Button „**Update-Mode beenden und Node neu starten**“ wird aktiviert.



ACHTUNG: Diesen Button bitte **nur** anklicken, wenn keine weiteren Firmware-Komponenten eingespielt werden müssen.

4.3.5.2 EEPROM

Das Einspielen der Daten für den EEPROM-Bereich des Prozessors verläuft wie das Einspielen der Firmware.

Beim Auswählen der .HEX-Datei erkennt der Monitor an der Endung .001.hex das es sich um Daten für den EEPROM-Bereich handelt.

Auch hier wird, durch Klick auf „Update starten“, die Übertragung gestartet und die Zeilen hochgezählt.

Alle Bedienelemente sind gesperrt.

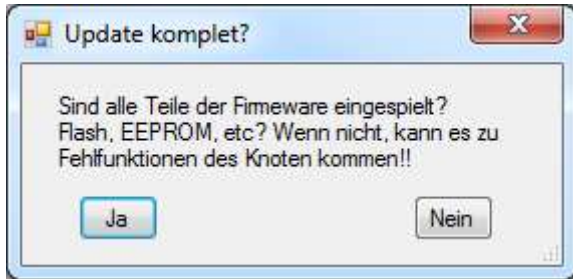
Der Button „**Update-Mode beenden und Node neu starten**“ wird aktiviert.

ACHTUNG: Diesen Button bitte **nur** anklicken, wenn keine weiteren Firmware-Komponenten eingespielt werden müssen.

4.3.5.3 Update beenden

Erst wenn alle Komponenten der Firmware eingespielt sind sollten Sie den Butten

Anklicken. Zu Sicherheit kommt noch mal eine Abfrage, ob wirklich alle Teile der Firmware eingespielt wurden.



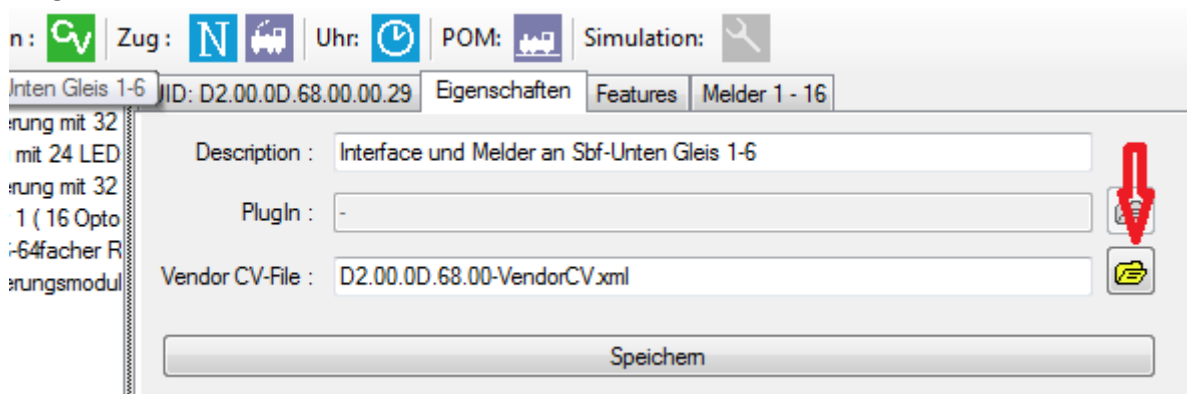
Wenn Sie hier auf „Ja“ klicken führt der Knoten einen Neustart durch. Bei „Nein“ kommen Sie zurück auf die SW-Update-Karteikarte.

4.3.6 Eigenschaften eines Knoten

Diese „Eigenschaften“ beziehen sich NICHT auf BiDiB-Funktionalitäten. Sie dienen lediglich der besseren Verwalt- und Lesbarkeit. Die Eigenschaften werden in der Datei *BiDiB-NodeProperties.xml* gespeichert (siehe 11.7).

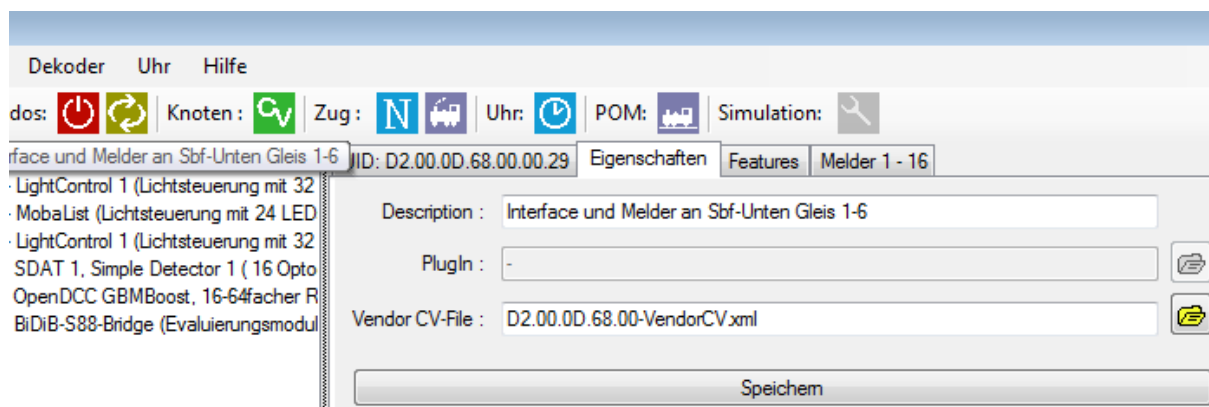
4.3.6.1 CV-Definitionsdatei zuweisen

Jedem Knoten kann hier eine eigene CV-Definitionsdatei zugewiesen werden. Ist dies nicht der Fall wird vom Monitor automatisch die Default-Datei eingetragen (siehe 9.6). Durch Mausklick auf das markierte Icon öffnet sich ein Dateidialog mit dem die XML-Datei ausgewählt werden kann.



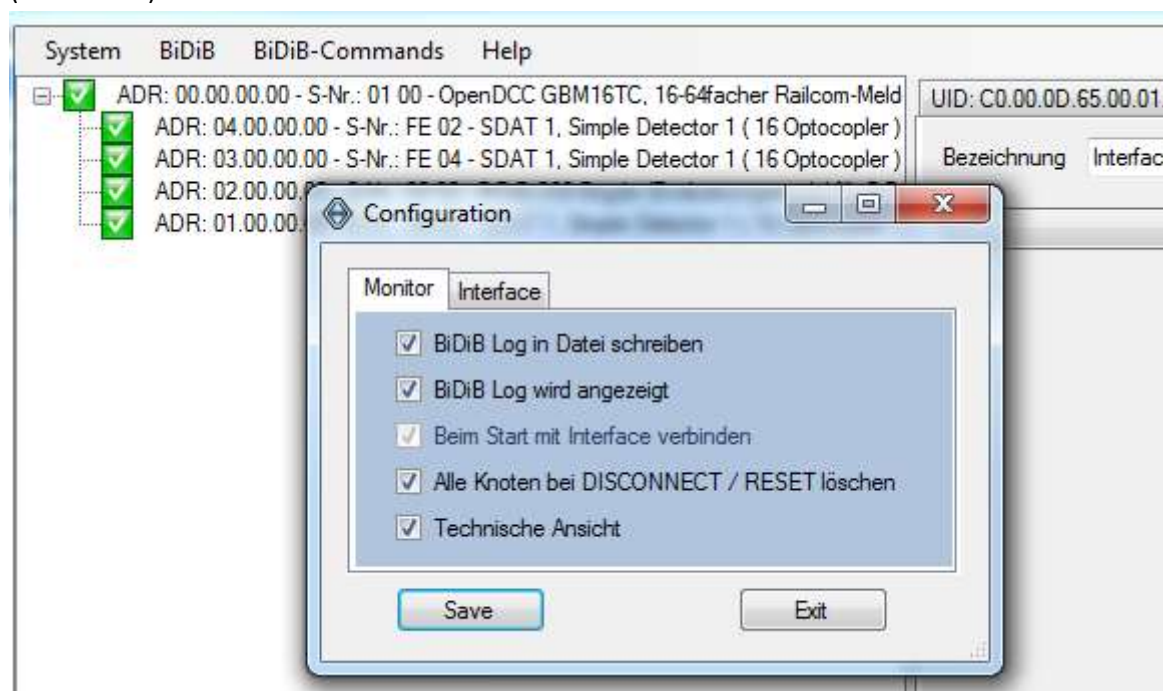
4.3.6.2 Beschreibende „Bezeichnung“ eines Knoten

Um die Knotenliste für den Anwender etwas besser lesbar zu machen, kann man jeder UID einen beschreibenden Text zuordnen. Damit kann der Melder, der im Schattenbahnhof 2, die Gleise 1 bis 4 überwacht, z.B. als „Melder Sbf 2 – Gleis 1 – 4“ beschrieben werden. Das dazu notwendige Eingabefeld findet man auf der Karteikarte „Eigenschaften“.

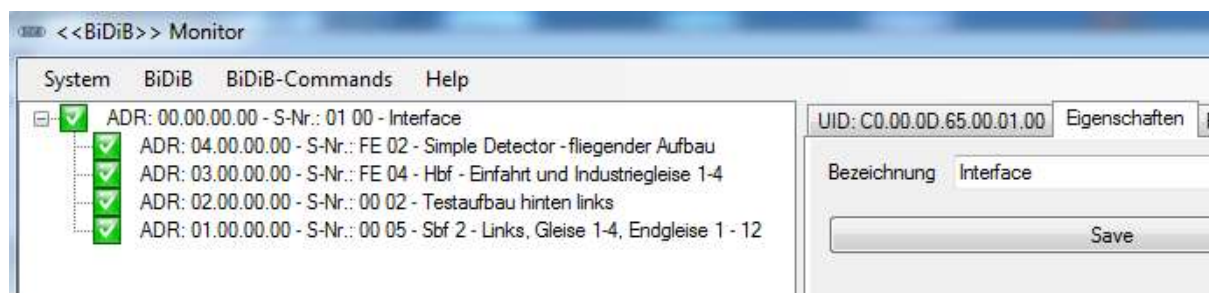


Anklicken von „Save“ weist sie der UID des Knoten zu und wird damit, bei Adresswechsel, immer dem richtigen Knoten zugeordnet.

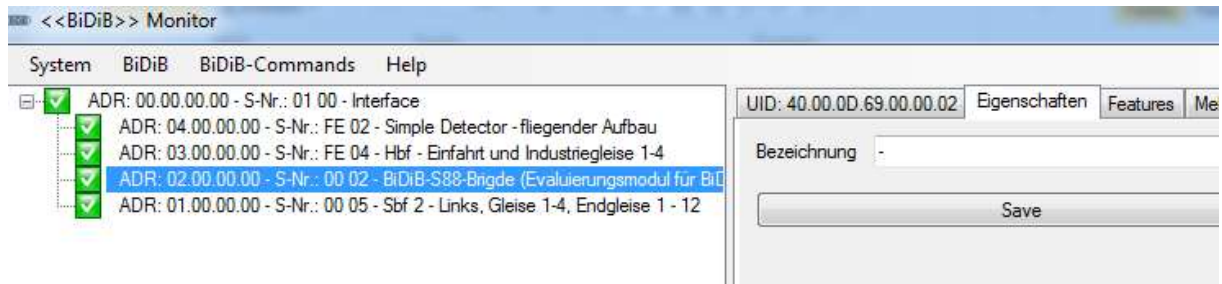
Wie unter 3.2.3 beschrieben, kann in der Konfiguration, die „Technische Ansicht“ ab- bzw. eingeschaltet werden. Oder über <F4> bzw. die Icons in der Menüleiste, direkt gewechselt werden (siehe 4.1.5).



Ist die „Technische Ansicht“ aktiviert, wird hinter der S-Nr. die Bezeichnung der Hardware angezeigt. Mit deaktivierter „Technischer Ansicht“ wird, wie in diesem Beispiel zu sehen, die Kurzbeschreibung hinter der S-Nr. ausgegeben.



Wird, als Bezeichnung, ein einfacher Bindestrich (-) eingegeben, wird unabhängig von der Einstellung der Konfiguration, immer der „Technische Name“ für diese UID ausgegeben.



4.3.7 Kontextmenü eines Knoten

Durch Anklicken eines der Knoten in der Baumstruktur mit der rechten Maustaste öffnet sich ein Menu mit Funktionen, die für diesen Knoten zur Verfügung stehen.

Diese Menu kann, je nach Knoten sehr unterschiedlich aussehen. Nur die, unter 4.3.7.1 und 4.3.7.2 beschriebenen Menüpunkte werden für alle Knoten angezeigt.

4.3.7.1 Knoten identifizieren

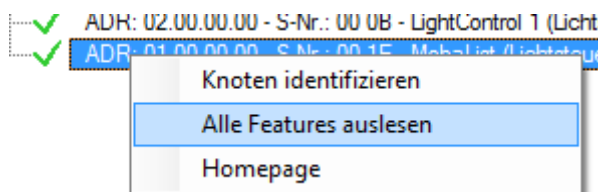
Durch Anklicken von „Knoten identifizieren“ wird die Message **MSG_IDENTIFY** an den Knoten geschickt. Je nach Identify-Zustand, wird dem Eintrag des Knotens in der Baumstruktur ein anderes Icon vorangestellt und die Schriftfarbe wechselt zwischen rot und schwarz.



Dieser Icon-/Farb-Wechsel wird auch beim Drücken der Identify-Taste am Knoten durchgeführt.

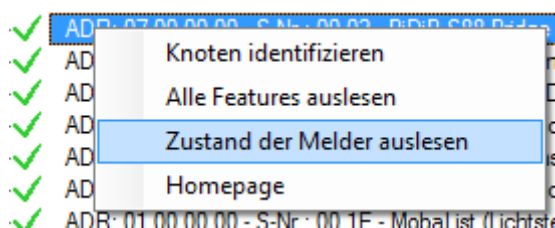
4.3.7.2 Alle Features auslesen

Durch Anklicken von „Alle Features auslesen“ wird die Message **MSG_FEATURE_GETALL** an den Knoten geschickt. Die Antworten werden entsprechend ausgewertet (**MSG_FEATURE_GETNEXT**)



4.3.7.3 Zustand der Melder auslesen

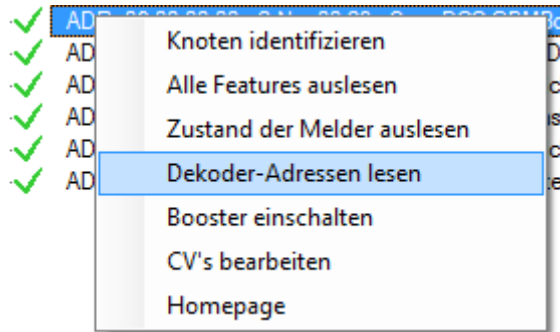
Dieser Menüpunkt steht nur zur Verfügung, wenn es sich bei dem Knoten um einen Belegtmelder oder Melder mit Momentkontakten handelt. Durch Anklicken von „Zustand der Melder auslesen“ wird die Message **MSG_BM_GET_RANGE** an den Knoten geschickt. Die Antworten werden entsprechend ausgewertet (**MSG_BM_MULTIPLE**)



4.3.7.4 Dekoder-Adressen lesen

Dieser Menüpunkt steht nur zur Verfügung, wenn es sich bei dem Knoten um einen Belegtmelder handelt der auch das Feature 9 für „Adressmeldung eingeschaltet“ gesetzt hat.

Durch Anklicken von „Dekoder-Adressen lesen“ wird die Message **MSG_BM_ADDR_GET_RANGE** an den Knoten geschickt. Die Antworten werden entsprechend ausgewertet (**MSG_BM_ADDRESS**)



4.3.7.5 Booster ein- / ausschalten

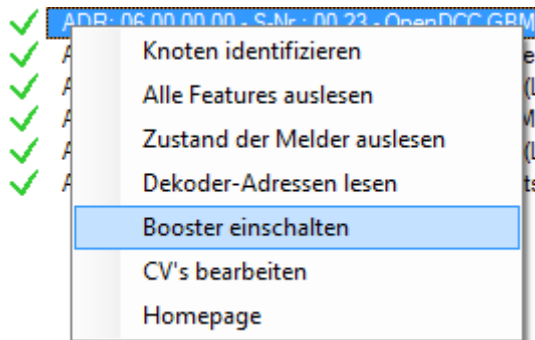
Dieser Menüpunkt steht nur zur Verfügung, wenn es sich bei dem Knoten um einen Booster handelt. Es wird nur der Booster des ausgewählten Knotens ein- bzw. ausgeschaltet.

Der Menüpunkt ändert sich dynamisch in Abhängigkeit vom Einschaltzustand des Boosters.

„Booster einschalten“ bzw. „Booster ausschalten“.

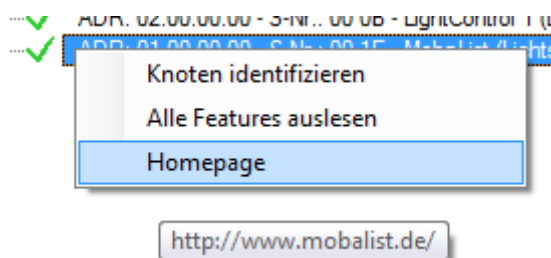
Durch Anklicken von „Booster einschalten“ wird die Message **MSG_BOOST_ON** an den Knoten geschickt. Durch Anklicken von „Booster ausschalten“ wird die Message **MSG_BOOST_OFF** an den Knoten geschickt.

Die Antworten werden entsprechend ausgewertet (**MSG_BOOST_STAT**)



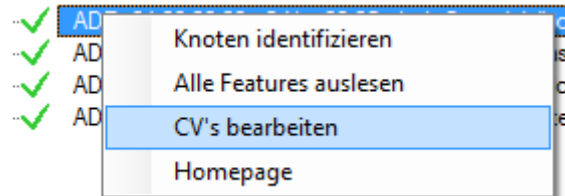
4.3.7.6 Homepage aufrufen

Dieser Menüpunkt wird nur eingeblendet, wenn in der Datei **BiDiBProductID.xml** der Eintrag für die Homepage mit einer vollständigen Homepage-Adresse gefüllt ist. Der „Tooltip“ zeigt die Adresse.



4.3.7.7 Configurationsvariablen (CV) des Knoten bearbeiten

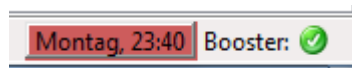
Dieser Menüpunkt wird nur eingeblendet, wenn in dem Unterverzeichnis „[data\BiDiBNodeVendorData](#)“ eine entsprechende XML-Datei mit den CV-Definitionen für die UID des Knoten liegt (siehe Kapitel 9).



4.4 Booster

Die Booster Funktionalität wird über den Menüpunkt „Booster Ein / Aus“ (siehe 4.2) ein- bzw. ausgeschaltet.

Der Status des Booster wird in der unteren, rechten Ecke durch ein Icon dargestellt.



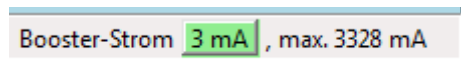
Dabei haben die Icons folgende Bedeutung (entspricht dem Rückgabewert von MSG_BOOST_STAT):

Icon	Code	Bedeutung
	0x00	Booster ist abgeschaltet (auf Grund Host-Befehl)
	0x01	Booster ist abgeschaltet (wegen Kurzschluss)
	0x02	Booster ist abgeschaltet (wegen Übertemperatur)
	0x03	Booster ist abgeschaltet (wegen fehlender Netzspannung)
	0x04	Booster ist abgeschaltet und es liegt eine Einschalt-Anforderung vor.
	0x05	Booster ist abgeschaltet (auf Grund lokalem Tastendruck).
	0x06	Booster ist abgeschaltet (auf Grund fehlendem DCC-Signal)
	0x80	Booster ist eingeschaltet.
	0x81	Booster ist eingeschaltet und läuft in der Strombegrenzung.
	0x82	Booster ist eingeschaltet und ist im kritischen Temperaturbereich.
	0x83	Booster ist eingeschaltet und es liegt eine Stop-Anforderung vor.
	0x84	Booster ist eingeschaltet (auf Grund lokalem Tastendruck).

4.4.1 Stromverbrauch

Wenn der Booster das Features 23 größer 1 gesetzt hat, wird der aktuelle Stromverbrauch in der Karteikarte der Melder, unten, rechts angezeigt.

Der max. Strom wird aus dem Feature 22 des Knoten gelesen.



In einem Tab im Logbereich werden alle gefundenen Booster in einer Liste dargestellt.

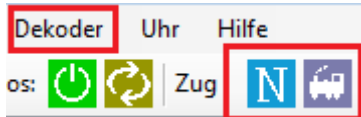
#	Unique ID	Beschreibung	Status	max. Strom	Strom	Strom
1	D2.00.0D.68.00.01.00	-		1984 mA	0 mA, 0%	<div></div>
2	42.00.0D.67.00.00.23			48 mA	3 mA, 6%	<div></div>

5 DCC-Steuerung

Zum Testen der DCC-Funktionen wurde ein kleines Fahrpult zu Dekodersteuerung eingerichtet.

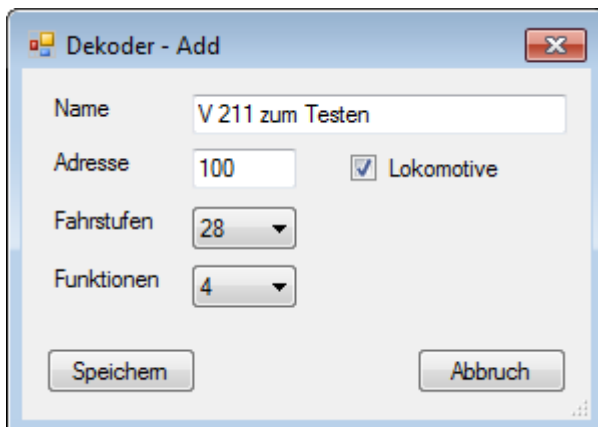
ACHTUNG: Diese Funktion steht nur zur Verfügung, wenn die Unique-ID des Interface im Byte ClassID1 das Bit 4 gesetzt hat (1: Knoten enthält DCC-Signalerzeugung für Fahren, Schalten).

Wenn die obere Bedingung erfüllt ist, werden in der Menü- und Iconleiste die entsprechenden Bedienelemente freigeschaltet. Hier rot umrandet.




5.1 Decoder eingeben

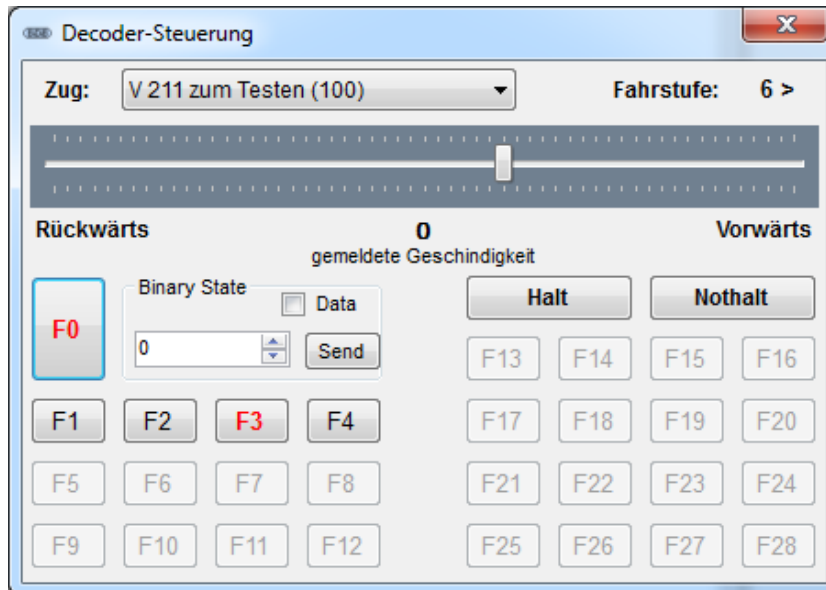
Das entsprechende Fenster wird durch Anklicken von **N**, über den Menüpunkt „Dekoder->Lokomotive“ oder Kurzwahltaste „<Alt>-L“ geöffnet.



- | | |
|-------------|--|
| Name: | Beschreibung des Dekoders |
| Adresse: | DCC-Adresse |
| Lokomotive: | Haken setzen wenn es sich um einen Lokdecoder handelt. |
| Fahrstufe: | DCC Fahrstufen des Dekoders. |
| Funktionen: | Anzahl der Dekoder-Funktionen |

5.2 Steuerung einer Lok

Das eigentliche Fahrpult wird durch Anklicken von  geöffnet. Es ist selbsterklärend und das Konzept aus anderen Anwendungen bekannt.



Über den dunkel hinterlegten Schieberegler wird die Fahrstufe und Fahrtrichtung der Lok eingestellt.

Fahrstufe und Richtung wird oben, rechts angezeigt.

Die Anzahl der bedienbaren Funktionstasten entspricht den, unter 5.1 eingegebenen, Funktionen. Bei einer betätigten Funktionstaste wechselt die Beschriftung in **rote, fette** Schrift.

Halt : Die Lok stoppt mit dem voreingestellten Bremsweg
 Nothalt : Die Lok stoppt sofort.

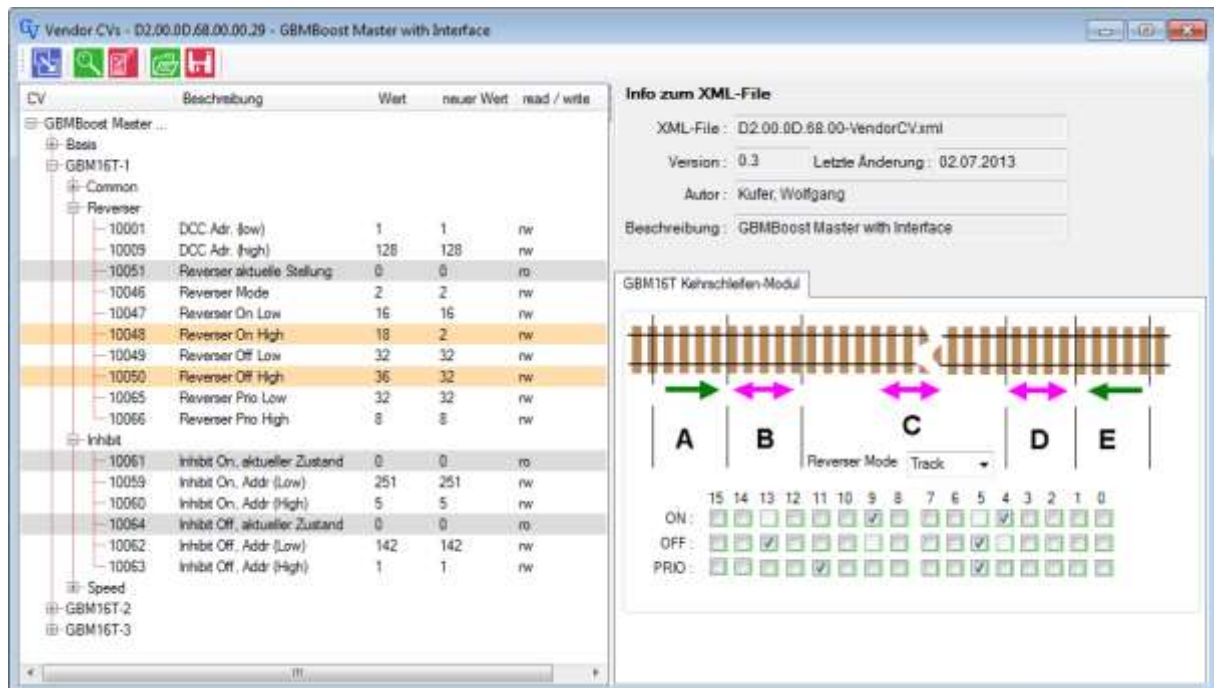
6 Funktionen für bestimmte Knoten/Module

6.1 Kehrschleife eines GBM16T konfigurieren

Die technischen Hintergründe dazu und weitere Möglichkeiten der Konfiguration über das Debug-Interface werden auf http://www.opendcc.de/s88/gbm_bidi/gbm_reverser.html beschrieben.

Bei Aufruf der CV-Verwaltung für einen GBMBoost, findet man in der Baumstruktur, unter den Bezeichnungen „GBM16T-1“, „GBM16T-2“ und „GBM16T-3“ auch die CV's für die GBM16T-Module. Die CV's für die Kehrschleife des jeweiligen GBM16T befinden sich im Zweig „Reverser“ und „Inhibit“.

Bei einem Mausklick auf eine der CV's, wird geprüft, ob die CV für die Kehrschleifen-Einstellungen gültig ist. Wenn ja, öffnet sich im unteren, rechten Bereich eine Eingabemaske in der die Einstellungen für die Kehrschleife eines GBM16T vorgenommen werden können.



Die CV's der GBM16T sind abhängig vom GBM16T der angesprochen werden soll. Für den ersten GBM16T beginnen sie bei 10000, für den zweiten bei 20000 und für den dritten bei 30000.

Offset + CV	Type in XML	Beschreibung
1	Int	DCC-Adresse des GBM16T (Lowbyte)
9	Int	DCC-Adresse des GBM16T (Highbyte)
51	Byte	Aktuelle Stellung der Kehrschleife (read only)
46	GBM16TReverser	Was löst die Kehrschleife aus: Off, DCC, Track, Track & DCC
47	GBM16TReverser	On Melder Bit 0-7 (Lowbyte)
48	GBM16TReverser	On Melder Bit 8-15 (Highbyte)
49	GBM16TReverser	Off Melder Bit 0-7 (Lowbyte)
50	GBM16TReverser	Off Melder Bit 8-15 (Highbyte)
62	GBM16TReverser	Prio Melder Bit 0-7 (Lowbyte)
63	GBM16TReverser	Prio Melder Bit 8-15 (Highbyte)

In den folgenden Abschnitten wird beschrieben wie die Werte geändert werden können.

Die neuen Werte sind damit aber noch nicht an den BiDiB-Knoten übertragen (siehe 9.3)

6.1.1 Welcher Melder ist mit welchem Abschnitt verbunden?

Um leichter zu erkennen, welcher Melder für welchen ON/OFF/PRIO Abschnitt zuständig ist, werden die Belegtmeldungen des GBM16T als Hintergrundfarbe der Checkboxes dargestellt.

The screenshot shows a control interface with a 'Reverser Mode' dropdown menu set to 'Track'. Below it is a grid of checkboxes organized into three rows: 'ON', 'OFF', and 'PRIO'. Each row has 16 columns corresponding to segments 15 down to 0. The checkboxes for segment 13 are highlighted in red, while the others are green. Some checkboxes have blue checkmarks.

Wenn man also mit einem Fahrzeug, das eine Belegtmeldung auslöst (Lok, Wagon mit Wiederstandachse, usw.), über die Abschnitte der Kehrschleife fährt, ändert sich

die Hintergrundfarbe der Checkbox von grün nach rot. Hier im Bild der Melder 13.

ACHTUNG: Der aktuelle Status der Belegtmelder wird beim ersten Aufruf **NICHT** in die Maske übernommen. Man muss also die Meldung neu auslösen. Dazu Fahrzeug aus Abschnitt fahren und wieder einfahren lassen.

6.1.2 DCC-Adresse des GBM16T festlegen

Durch Anklicken von CV 1 (DCC Adr. low) oder 9 (DCC Adr. High) öffnet sich im Bereich rechts/unten eine Eingabemaske:

CV	Bezeichnung	Wert 1	Wert 2	Modus
10001	DCC Adr. (low)	252	252	rw
10009	DCC Adr. (high)	3	3	rw
10051	Reverser aktuelle Stellung	0	0	ro
10046	Reverser Mode	2	2	rw
10047	Reverser On Low	16	16	rw
10048	Reverser On High	18	18	rw
10049	Reverser Off Low	32	32	rw
10050	Reverser Off High	36	36	rw
10065	Reverser Prio Low	32	32	rw
10066	Reverser Prio High	8	8	rw
10061	Inhibit On, aktueller Zustand	0	0	ro
10059	Inhibit On, Addr (Low)	0	0	rw
10060	Inhibit On, Addr (High)	0	0	rw
10064	Inhibit Off, aktueller Zustand	0	0	ro

Beschreibung : GBM16T

CV Eingabe (low/high)

Integerwert
1020

DCC Adr. (high)(10009)
3

DCC Adr. (low)(10001)
252

In dem Feld „Integerwert“ kann die gewünschte DCC-Adresse eingegeben werden. Die CV's werden automatisch mit den entsprechenden Werten gefüllt. Diese Adresse wird benötigt, wenn der Reverser-Mode auf „DCC“ oder „Track & DCC“ steht.

6.1.3 Reverser Mode – Wodurch wird die Kehrschleife ausgelöst?

In der Combobox „Reverser Mode“ wird ausgewählt, wodurch die Kehrschleife ausgelöst wird.

Reverser Mode: Track

- Off
- DCC
- Track
- Track & DCC

Es kann zwischen „Off“, „DCC“, „Track“ und „Track & DCC“ gewählt werden. Dies entspricht im Debug-Interface dem Befehlen [RDCC \[0|1\]](#) und [RT \[0|1\]](#).

6.1.4 ON-Melder setzen

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ON :	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
OFF :	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PRIO :	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Über die Checkboxes in der Zeile ON werden die Bits für die entsprechenden Melder gesetzt. Die Checkboxes für ON können nur gesetzt

werden, wenn sie nicht schon als OFF-Melder angeklickt sind. Ein Melder, der als ON-Melder gesetzt ist, kann also nicht mehr als OFF-Melder angeklickt werden. Und umgekehrt.

Dies entspricht im Debug-Interface dem Befehl [RA 1 <melder>](#) bzw. [RM 1 <melder>](#).

6.1.5 OFF-Melder setzen

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ON :	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
OFF :	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PRIO :	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Über die Checkboxes in der Zeile „OFF“ werden die Bits für die entsprechenden Melder gesetzt. Die Checkboxes für OFF können nur gesetzt

werden, wenn sie nicht schon als ON-Melder angeklickt sind. Ein Melder, der als OFF-Melder gesetzt ist, kann also nicht mehr als ON-Melder angeklickt werden. Und umgekehrt.

Dies entspricht im Debug-Interface dem Befehl `RA 0 <melder>` bzw. `RM 0 <melder>`.

6.1.6 Prioritätsmelder setzen

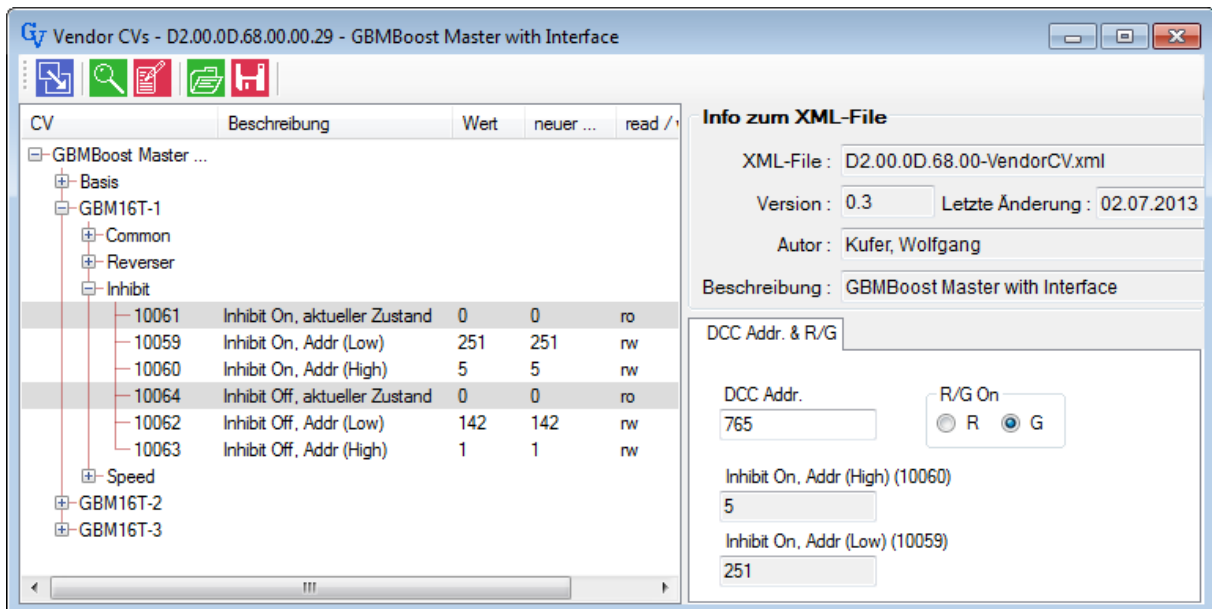
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ON :	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
OFF :	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PRIO :	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Über die Checkboxes in der Zeile „PRIO“ werden die Bits für die entsprechenden Melder gesetzt.

Dies entspricht im Debug-Interface dem Befehl `RA 2 <melder>` bzw. `RM 2 <melder>`.

6.1.7 Inhibit On konfigurieren

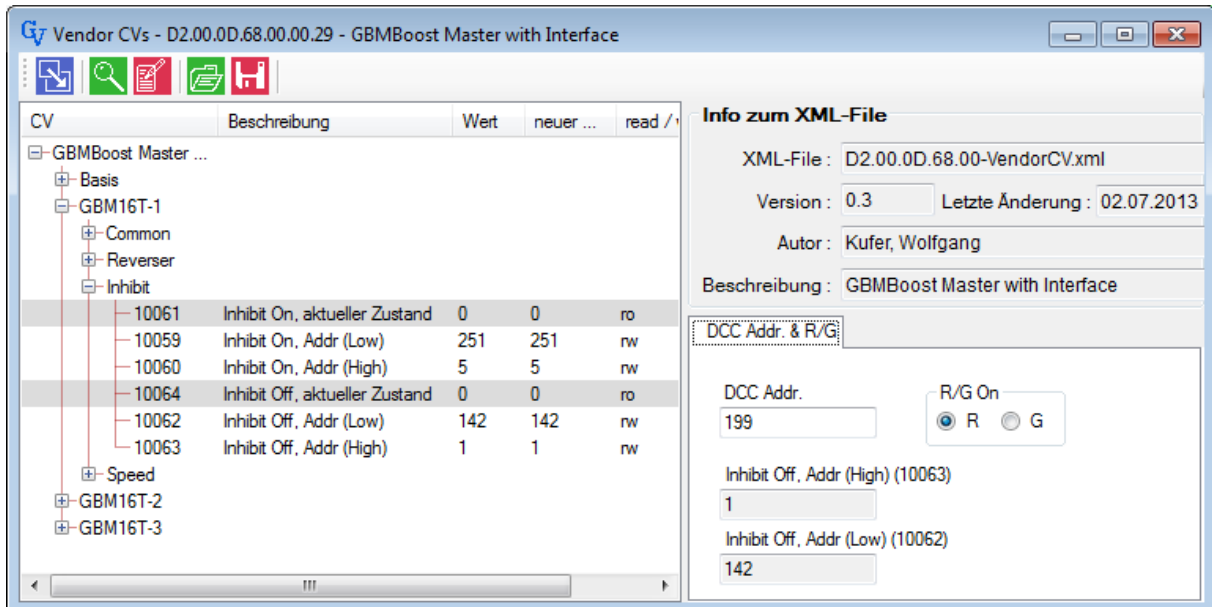
Durch Anklicken eines der CV von „Inhibit ON“ öffnet sich im rechten, unteren Bereich eine Maske mit den Eingabefeldern für die DCC Adresse und der Auswahl, ob bei R oder G ausgelöst werden soll. Diese Eingaben werden automatisch in die CV-Werte umgerechnet.



Dies entspricht im Debug-Interface dem Befehl `RI 0 <DCC-Adr.> [R|G]`

6.1.8 Inhibit Off konfigurieren

Durch Anklicken eines der CV von „Inhibit OFF“ öffnet sich im rechten, unteren Bereich eine Maske mit den Eingabefeldern für die DCC Adresse und der Auswahl, ob bei R oder G ausgelöst werden soll. Diese Eingaben werden automatisch in die CV-Werte umgerechnet.



Dies entspricht im Debug-Interface dem Befehl `RI 1 <DCC-Adr.> [R|G]`

7 POM / xPOM

ACHTUNG: Diese Funktion steht nur zur Verfügung, wenn die Unique-ID des Interface im Byte ClassID1 das Bit 4 gesetzt hat (1: Knoten enthält DCC-Signalerzeugung für Fahren, Schalten).

Wenn die obere Bedingung erfüllt ist, werden in der Menü- und Iconleiste die entsprechenden Bedienelemente freigeschaltet. Hier rot umrandet.



Über das neue Fenster können die 10 Parameter der BiDiB-Message MSG_CS_POM eingestellt und anschließend durch Klick auf den Senden-Button an das Interface gesendet werden. Die aktuellen Werte dieser 10 Byte stehen im unteren Feld und werden dynamisch den Eingaben angepasst.

Eine Beschreibung findet sich auf www.bidib.org unter http://www.bidib.org/protokoll/bidib_command_station.html#T453

Die Eingaben sind auch hier (hoffentlich) weitestgehend selbsterklärend:

- Aktion : POM Locomotive, Accessory, Extended Accessory oder via Decoder-ID (DID)
- Adresse : DCC-Adresse des Dekoders.
- CV : Nummer der CV die bearbeitet werden soll.
- POM /XPOM : bei POM ist der Adressierbare CV-Bereich bei 0 – 1023, bei XPOM liegt er bei 24 Bit.
- Operation : Die vier in DCC definierten Zugriffarten.
- Wert : Dieser Wert wird bei den Schreiboperationen in den CV geschrieben.

7.1 POM via Decoder-ID

Wenn die Aktion „POM via Decoder DID“ ausgewählt wird, wird die Adressauswahl ausgeblendet. Dafür sind jetzt fünf Auswahlfelder zusehen, mit deren Hilfe die Decoder-ID eingegeben werden kann.

The screenshot shows the 'MSG_CS_POM' window with the following settings:

- Aktion: POM via Decoder DID
- DID[0]: 0, DID[1]: 0, DID[2]: 0, DID[3]: 0
- MID: T4T - Technology for Trains GmbH
- CV: 1
- Operation: ☒ POM, ☐ XPOM (railcom)
- Operation: ☒ Byte schreiben, ☐ Bit schreiben, ☐ Byte lesen, ☐ Block lesen
- Wert: 0
- Hex display: 00 00 00 00 14 03 00 00 00 00
- Buttons: Senden, Exit

7.2 Operation „Bit schreiben“

Mit der Auswahl der Operation „Bit schreiben“ wird der Eingabebereich für das den Wert geändert. Im Auswahlbereich „Bit“ wird das Bit ausgewählt, in das geschrieben werden soll.

Die Checkbox hinter „Wert : „ gibt an, ob eine 1 (Haken gesetzt) oder 0 geschrieben werden soll.

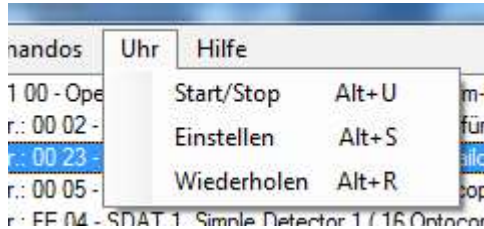
The screenshot shows the 'MSG_CS_POM' window with the following settings:

- Aktion: POM Locomotive
- Adresse: 211
- CV: 29
- Operation: ☒ POM, ☐ XPOM (railcom)
- Operation: ☐ Byte schreiben, ☒ Bit schreiben, ☐ Byte lesen, ☐ Block lesen
- Wert: ☒ (checked)
- Bit: ☐ 7, ☐ 6, ☐ 5, ☐ 4, ☒ 3, ☐ 2, ☐ 1, ☐ 0
- Hex display: D3 00 00 00 00 02 1C 00 00 FB
- Buttons: Senden, Exit

Hier wird also in CV 29 des Dekoders 211 das Bit 3 auf 1 gesetzt.

8 Modelbahn-Uhr

Alle Funktionen der Modelbahnuhr werden über den Menüpunkt „Uhr“ oder über die Kurzwahlkosten gesteuert.



<Alt>-u startet bzw. stoppt die Uhr.

<Alt>-s öffnet das Konfigurationsfenster der Uhr (siehe 8.1)

<Alt>-r aktiviert bzw. deaktiviert den Repeat-Modus der Uhr (siehe 8.2.3).

8.1 Einstellungen

Über das Menu „Uhr“->„Set“ oder über die Tastenkombination <Alt>-S wird das Konfigurationsfenster der Uhr aufgerufen. Diese Einstellungen werden in der Registry gespeichert und beim nächsten Start wieder eingelesen.



Die Einstellungen sind weitestgehend selbsterklärend.

Für Start- und Endezeit können jeweils die Stunden und Minuten eingestellt werden. Bei Wochentag der Tag der Woche für den die Startzeit gilt.

Der Faktor gibt die „Beschleunigung“ der Zeit an. Er ist von 1 bis 31 einstellbar. 10 bedeutet, die Uhr läuft 10-mal so schnell als real. Also 1 Min. dauert nicht 60, sondern nur 6 Sekunden.

Faktor 30 also 1 Min = 2 Sek.

Ist der Haken bei „Endlos“ gesetzt wird nur die Startzeit und der Wochentag als Startwert genommen. Die Uhr läuft Endlos (siehe 8.2.1).

8.2 Uhr-Funktionen

8.2.1 Starten mit „Endlos“ eingeschaltet.

Wenn das „Endlos“-Flag in der Konfiguration gesetzt ist, läuft die Uhr, beginnend bei der Startzeit mit dem eingestellten Faktor, bis der Anwender die Uhr stoppt.

Bei gestarteter Uhr wechselt der Anzeigehintergrund von Rot auf Grün und die Uhr beginnt zu laufen. Im Log-Bereich werden die Startparameter und die entsprechenden BiDiB-Meldungen ausgegeben.

Endlos aktiv, Montag, 23:58

```
Clocksettings: ON , Weekday = Montag, Faktor = 25 (1 Min = 2400 ms), Start = 23:57, Ende = 00:02
OUT -- : FE 07 00 39 18 3A 97 40 D9 B8 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 3A 18 3B 97 40 D9 6E FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 3B 18 00 80 41 D9 71 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 3C 18 01 80 41 D9 7B FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 3D 18 02 80 41 D9 C4 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 3E 18 00 80 40 C0 5F FE .....MSG_SYS_CLOCK
Clocksettings: OFF , Weekday = Dienstag, Faktor = 25 (1 Min = 2400 ms), Start = 23:57, Ende = 00:02
```

Endlos aktiv, Dienstag, 00:02

Die gestoppte Uhr wird wieder mit Rotem Hintergrund dargestellt.

8.2.2 Starten mit „Endlos“ ausgeschaltet.

Wenn das „Endlos“-Flag in der Konfiguration nicht gesetzt ist, läuft die Uhr, beginnend bei der Startzeit mit dem eingestellten Faktor, bis die Endzeit erreicht wird. Ist die Endzeit kleiner als die Startzeit, läuft die Uhr in den nächsten Tag. Die Anzeige wechselt beim Start auf grünen Hintergrund und am Ende wieder auf Rot.

Montag, 23:58

```
Clocksettings: ON , Weekday = Montag, Faktor = 25 (1 Min = 2400 ms), Start = 23:57, Ende = 00:02
OUT -- : FE 07 00 17 18 3A 97 40 D9 1D FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 18 18 3B 97 40 D9 B6 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 19 18 00 80 41 D9 A9 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 1A 18 01 80 41 D9 7F FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 1B 18 02 80 41 D9 C0 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 1C 18 02 80 41 C0 44 FE .....MSG_SYS_CLOCK
Clocksettings: OFF , Weekday = Dienstag, Faktor = 25 (1 Min = 2400 ms), Start = 23:57, Ende = 00:02
```

Dienstag, 00:02

8.2.3 Starten mit „Endlos“ ausgeschaltet und „Repeat“ aktiviert.

Es gilt das gleiche Verhalten wie unter 8.2.2. Wird aber jetzt die Endzeit erreicht, beginnt die Uhr automatisch wieder bei der Startzeit und wiederholt dies so lange, bis der Anwender die Uhr von Hand stoppt.

ACHTUNG: Wenn Einstellungen an Startzeit, Endzeit, Faktor oder Wochentag während laufender Uhr geändert werden, werden diese bei der nächsten Wiederholung als neue Startparameter angewendet.

Ob „Repeat“ aktiviert oder deaktiviert ist, wird NICHT in der Registry gespeichert. Es ist bei jedem Start des Monitors deaktiviert.

Der Beginn jedes Durchlaufes wird durch eine neue „Clocksettings“-Zeile angezeigt.

```

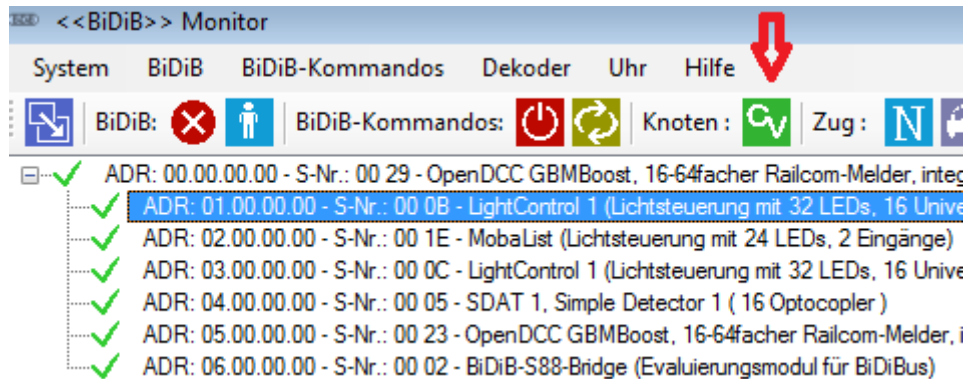
Repeat aktiv, Montag, 23:57 ...
Clocksettings: ON , Weekday = Montag, Faktor = 25 (1 Min = 2400 ms), Start = 23:57, Ende = 00:02
OUT -- : FE 07 00 1D 18 3A 97 40 D9 D2 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 1E 18 3B 97 40 D9 04 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 1F 18 00 80 41 D9 1B FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 20 18 01 80 41 D9 5D FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 21 18 02 80 41 D9 E2 FE .....MSG_SYS_CLOCK
Clocksettings: ON , Weekday = Montag, Faktor = 25 (1 Min = 2400 ms), Start = 23:57, Ende = 00:02
OUT -- : FE 07 00 22 18 3A 97 40 D9 1B FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 23 18 3B 97 40 D9 A3 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 24 18 00 80 41 D9 0E FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 25 18 01 80 41 D9 B6 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 26 18 02 80 41 D9 67 FE .....MSG_SYS_CLOCK
Clocksettings: ON , Weekday = Montag, Faktor = 25 (1 Min = 2400 ms), Start = 23:57, Ende = 00:02
OUT -- : FE 07 00 27 18 3A 97 40 D9 F0 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 28 18 3B 97 40 D9 5B FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 29 18 00 80 41 D9 44 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 2A 18 01 80 41 D9 92 FE .....MSG_SYS_CLOCK
OUT -- : FE 07 00 2B 18 00 80 40 C0 EF FE .....MSG_SYS_CLOCK
Clocksettings: OFF , Weekday = Dienstag, Faktor = 25 (1 Min = 2400 ms), Start = 23:57, Ende = 00:02
Repeat aktiv, Dienstag, 00:01 ...

```

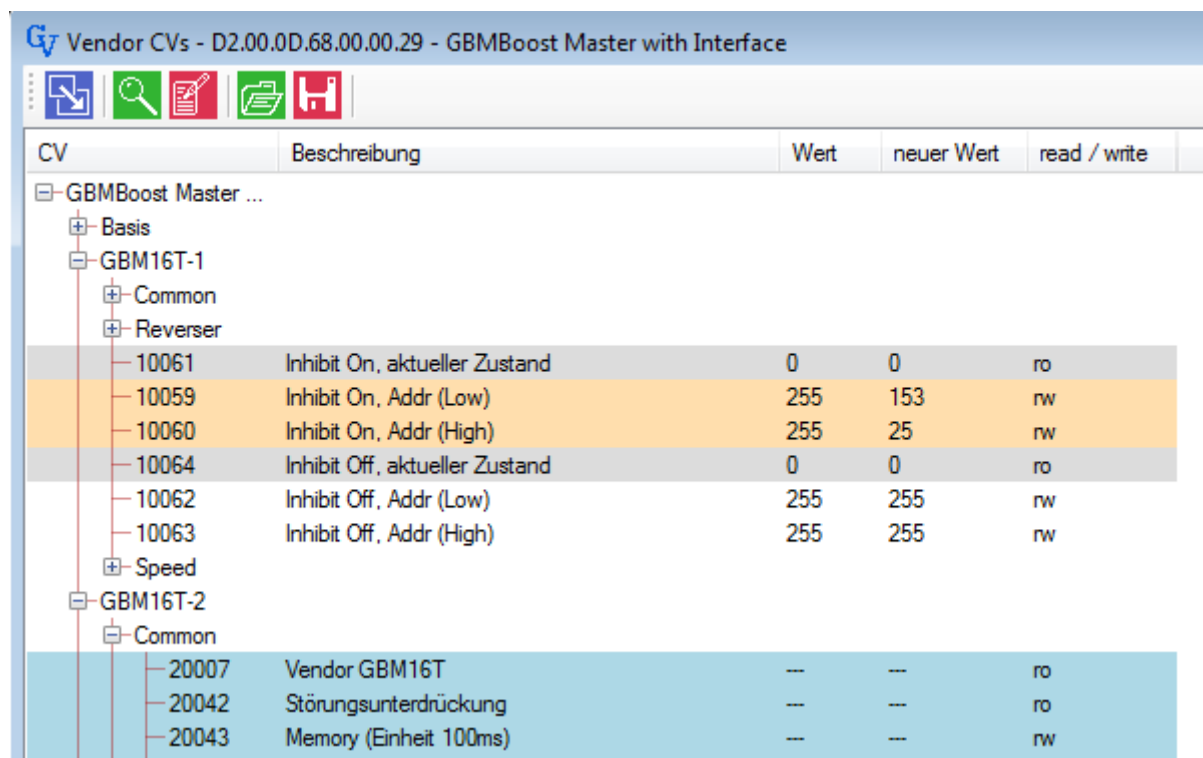
9 CV-Verwaltung für BiDiB-Knoten

Wenn der Monitor für einen BiDiB-Knoten eine CV-Definitionsdatei findet (siehe 9.6) wird im Kontextmenu des Knoten der Menüpunkt „CV's Bearbeiten“ eingeblendet.

Wird der Knoten im Baum mit linken Maustaste ausgewählt und eine CV- Definitionsdatei wurde gefunden ist die CV-Verwaltung auch über das, hier markierte Icon aufrufbar.



Im linken Teil des Fensters werden die CV's in einer Baumstruktur dargestellt.



Die hinterlegten Farben haben folgende Bedeutung:

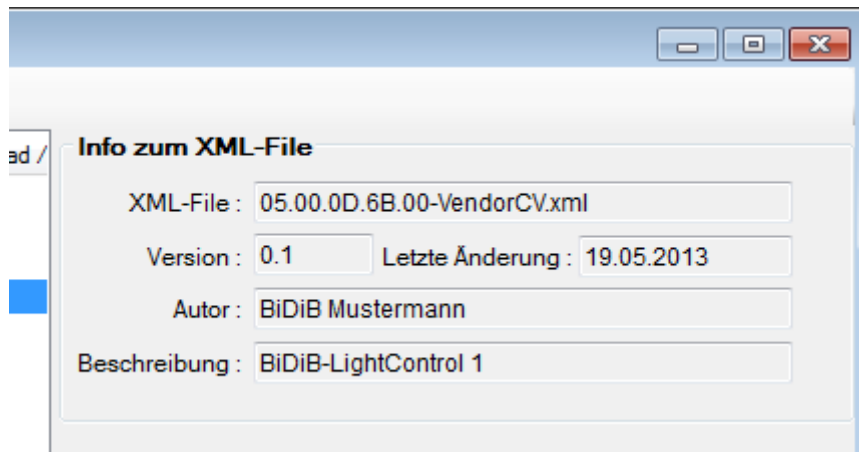
Grau = CV ist Read only

Gelb = CV ist Write only

Blau = Timeout beim Lesen der CV

Orange = Der Wert der CV wurde geändert aber noch nicht an den Knoten übertragen.

Im rechten Teil stehen einige Informationen zu der verwendeten XML-Datei.




The screenshot shows a window titled "Info zum XML-File" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains four text input fields arranged vertically, each with a label to its left. The first field is labeled "XML-File:" and contains the text "05.00.0D.6B.00-VendorCV.xml". The second field is labeled "Version:" and contains "0.1", followed by a label "Letzte Änderung:" and a field containing "19.05.2013". The third field is labeled "Autor:" and contains "BiDiB Mustermann". The fourth field is labeled "Beschreibung:" and contains "BiDiB-LightControl 1".

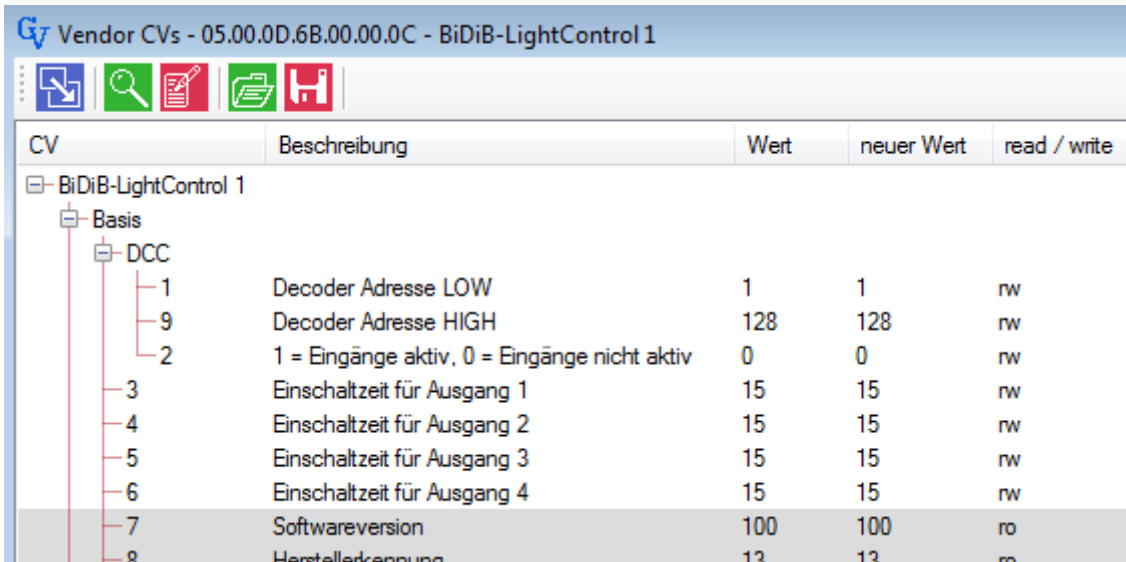
Info zum XML-File	
XML-File :	05.00.0D.6B.00-VendorCV.xml
Version :	0.1
Letzte Änderung :	19.05.2013
Autor :	BiDiB Mustermann
Beschreibung :	BiDiB-LightControl 1

9.1 CV's lesen

Es werden nur die CV's vom Knoten gelesen bei dehnen in der Spalte „read/write“ ein „ro“ oder „rw“ steht.

9.1.1 Alle CV's lesen

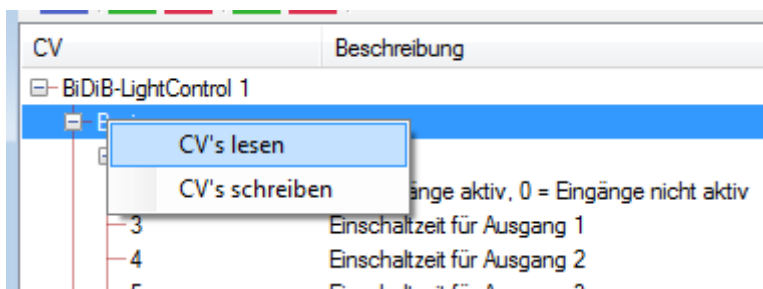
Über den Button  werden ALLE CV's des Knoten gelesen und in den Spalten „Wert“ und „neuer Wert“ eingetragen:



CV	Beschreibung	Wert	neuer Wert	read / write
BiDiB-LightControl 1				
Basis				
DCC				
1	Decoder Adresse LOW	1	1	rw
9	Decoder Adresse HIGH	128	128	rw
2	1 = Eingänge aktiv, 0 = Eingänge nicht aktiv	0	0	rw
3	Einschaltzeit für Ausgang 1	15	15	rw
4	Einschaltzeit für Ausgang 2	15	15	rw
5	Einschaltzeit für Ausgang 3	15	15	rw
6	Einschaltzeit für Ausgang 4	15	15	rw
7	Softwareversion	100	100	ro
8	Herstellerkennung	13	13	m

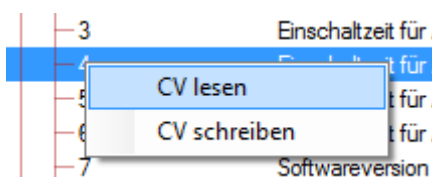
9.1.2 CV-Gruppen lesen

Durch Anklicken von „CV's lesen“ im Kontextmenu (rechtsklick) eines Knoten im Baum, werden alle CV's gelesen, die unterhalb dieses Knoten liegen. Die gelesenen Werte werden in den Spalten „Wert“ und „neuer Wert“ eingetragen.



9.1.3 Einzelne CV lesen

Durch Anklicken von „CV lesen“ im Kontextmenu (rechtsklick) einer CV im Baum, wird die CV gelesen. Der gelesene Wert wird in den Spalten „Wert“ und „neuer Wert“ eingetragen.



9.2 CV's ändern

Je nach Type, der in der XML-Definition einer CV mit dem Tag `<type></type>` wird, ergeben sich andere Eingabemöglichkeiten.

Für alle beschriebenen Typen gilt:

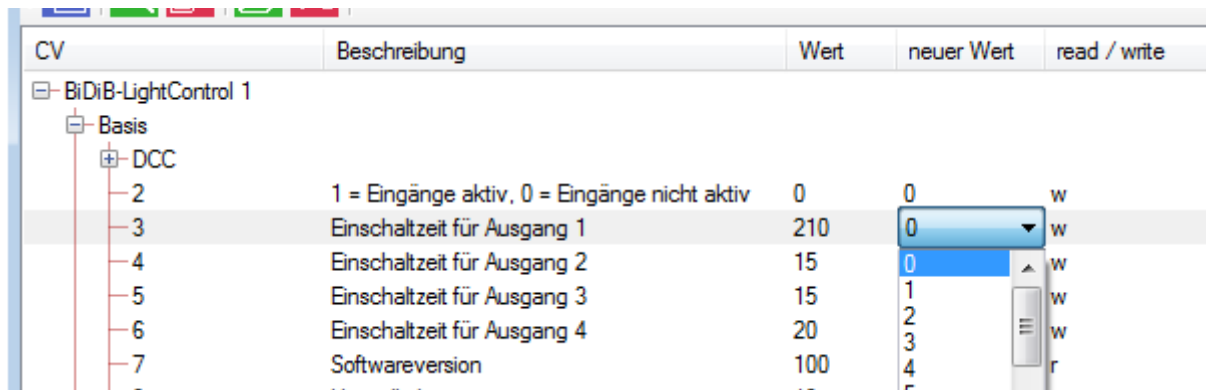
Ein geänderter Wert wird NICHT automatisch an den Knoten übertragen. Dies muss, wie unter 9.2.1.2 beschrieben erfolgen.

9.2.1 Allgemeine Datentypen

9.2.1.1 CV's vom Type „Byte“

Wenn in der Spalte „read / write“ ein „rw“ oder „wo“ steht kann eine CV vom Type „Byte“ geändert werden.

Per Doppelklick auf den Wert der CV in der Spalte „neuer Wert“ öffnet sich eine Auswahlliste.



CV	Beschreibung	Wert	neuer Wert	read / write
BiDiB-LightControl 1				
Basis				
DCC				
2	1 = Eingänge aktiv, 0 = Eingänge nicht aktiv	0	0	w
3	Einschaltzeit für Ausgang 1	210	0	w
4	Einschaltzeit für Ausgang 2	15	0	w
5	Einschaltzeit für Ausgang 3	15	1	w
6	Einschaltzeit für Ausgang 4	20	2	w
7	Softwareversion	100	4	r

Hier kann nun ein neuer Wert ausgewählt werden. Mit den Tasten <enter>, <tab> oder einem Mausklick außerhalb der Auswahlliste wird der Wert übernommen. Und in der Spalte „neuer Wert“ angezeigt.

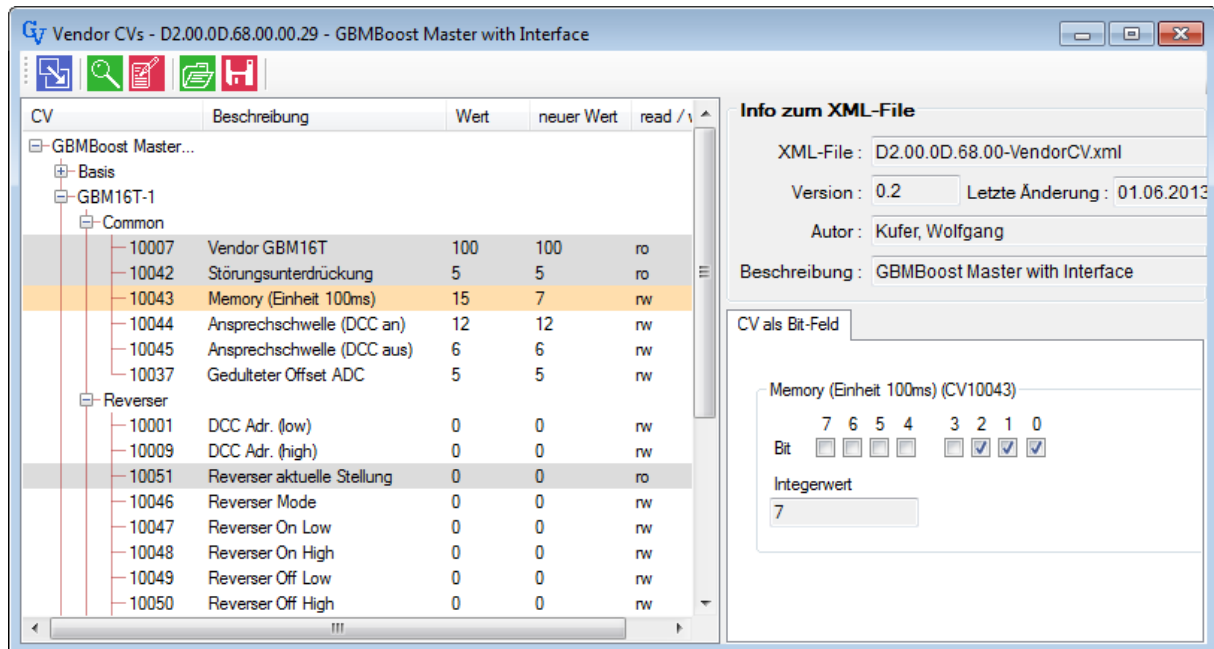
Wenn „Wert“ und „neuer Wert“ unterschiedlich sind, ändert sich die Hintergrundfarbe dieser Zeile in orange.

Der neue Wert ist damit aber noch nicht an den BiDiB-Knoten übertragen.

9.2.1.2 CV's vom Type „Bit“

Wenn in der Spalte „read / write“ ein „rw“ oder „wo“ steht kann eine CV vom Type „Bit“ geändert werden.

Ein Mausklick auf die Zeile mit der CV öffnet im rechten, unteren Bereich eine Eingabemaske für das Bitmuster.



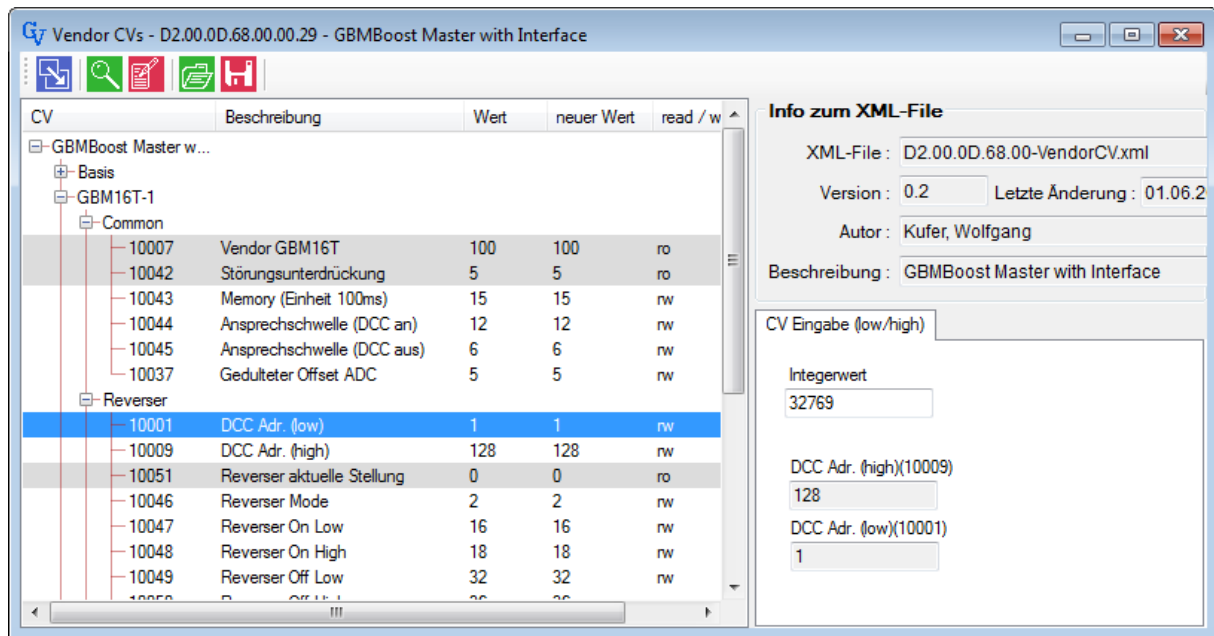
Durch Anklicken der Checkboxes wird der Wert im Feld „Integerwert“ und in der Spalte „neuer Wert“ automatisch geändert. Weitere Möglichkeiten, zur Steuerung der Eingabe von Bitwerten sind unter 9.6.2.3 beschrieben.

Der neue Wert ist damit aber noch nicht an den BiDiB-Knoten übertragen.

9.2.1.3 CV's vom Type „Int“

Wenn in der Spalte „read / write“ ein „rw“ oder „wo“ steht kann eine CV vom Type „Int“ geändert werden.

Ein Mausklick auf die Zeile mit der CV, öffnet im rechten, unteren Bereich eine Eingabemaske für den Int-Wert. Ein Int- Wert setzt sich aus zwei Bytewerten zusammen (Low-Byte und High-Byte). Er besteht also aus zwei CV's.



Im Eingabefeld „Integerwert“ kann eine beliebige Zahl von 0 – 65535 eingegeben werden. Wird der Wert außerhalb dieses Bereiches eingegeben, wird eine Fehlermeldung angezeigt und der letzte gültige Wert wird wieder eingetragen.

Wie ein CV Wert vom Type „Int“ definiert wird, steht unter 9.6.2.5 .

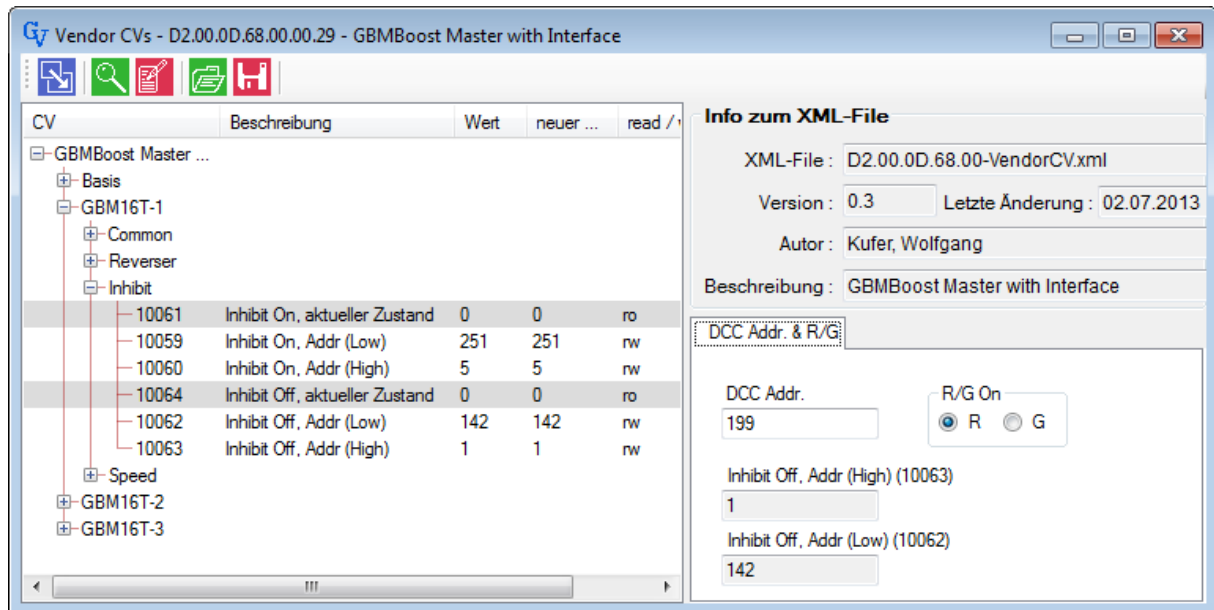
Der neue Wert ist damit aber noch nicht an den BiDiB-Knoten übertragen.

9.2.1.4 CV's vom Type „DCC_ADDR_RG“

Wenn in der Spalte „read / write“ ein „rw“ oder „wo“ steht kann eine CV vom Type „DCC_ADDR_RG“ geändert werden.

Ein Mausklick auf die Zeile mit der CV, öffnet im rechten, unteren Bereich eine Eingabemaske für den DCC_ADDR_RG -Wert. Ein DCC_ADDR_RG - Wert setzt sich aus zwei Bytewerten zusammen (Low-Byte und High-Byte). Er besteht also aus zwei CV's. Das entspricht 16 Bit.

Die eingegebene DCC-Adresse wird in den oberen 15 Bit abgebildet (Bit 1 – 15). Über die Auswahl zwischen R und G wird Bit 0 auf 1 (G) oder 0 (R) gesetzt.



Im Eingabefeld „DCC Addr.“-Wert kann eine beliebige Zahl von 0 – 32766 eingegeben werden. Ein Wert außerhalb dieses Bereiches löst eine Fehlermeldung aus und der letzte gültige Wert wird wieder eingetragen.

Wie ein CV Wert vom Type „DCC_ADDR_RG“ definiert wird, steht unter 9.6.2.6 .

Der neue Wert ist damit aber noch nicht an den BiDiB-Knoten übertragen.

9.2.2 Datentypen für bestimmte Knoten


9.2.2.1 CV's vom Type „GBM16TReverser“ – Kehrschleife des GBM16T

Der Type „GBM16TReverser“ stellt nicht wirklich einen Datentype dar und ist **nur** für einen **GBM16T** zugelassen. Der Monitor prüft ab, ob die CV zu den CV's gehört, die zur Konfiguration des Kehrschleifenmoduls eines GBM16T benötigt werden. Und ob die UID zu einem GBMBoost passt.

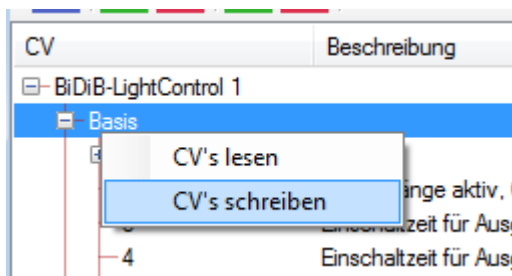
Beschreibung zur Konfiguration der Kehrschleifen siehe 6.1

9.3 CV's in den Knoten schreiben

9.3.1 Alle CV's in BiDiB-Knoten schreiben

Über den Button  werden **ALLE** geänderten Werte an den an den BiDiB-Knoten übertragen. Dazu wird überprüft, ob es einen Unterschied zwischen der Spalte „Wert“ und „neuer Wert“ gibt. Ausnahme sind die CV's des Knoten die auf „read only“ stehen. Die, vom BiDiB-Knoten bestätigten Werte, werden in der Spalte „Wert“ eingetragen.

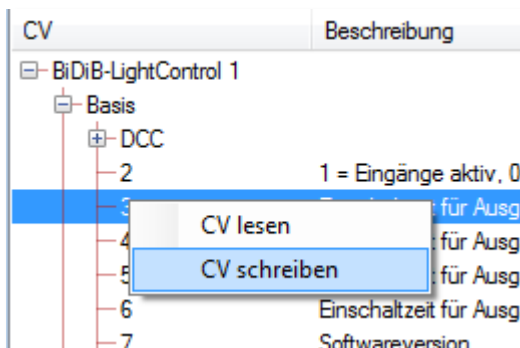
9.3.2 CV-Gruppe in BiDiB-Knoten schreiben



Durch Anklicken von „CV's schreiben“ im Kontextmenu (rechtsklick) eines Knoten im Baum, werden alle geänderten Werte, die unterhalb dieses Baumknoten liegen, an den BiDiB-Knoten geschickt. Dazu wird überprüft, ob es einen Unterschied zwischen der Spalte „Wert“ und „neuer Wert“ gibt.


Ausnahme sind die CV's des Knoten die auf „read only“ stehen. Die, vom BiDiB-Knoten bestätigten Werte, werden in der Spalte „Wert“ eingetragen.

9.3.3 Einzelne CV in BiDiB-Knoten schreiben




Durch Anklicken von „CV schreiben“ im Kontextmenu (rechtsklick) einer CV im Baum, wird der Wert dieses CV aus Spalte „neuer Wert“ an den BiDiB-Knoten geschickt. Aber nur wenn es einen Unterschied zwischen der Spalte „Wert“ und „neuer Wert“ gibt und die CV schreibbar ist („wo“ oder „rw“). Der, vom BiDiB-Knoten bestätigte Wert, wird in der Spalte „Wert“ eingetragen.

9.4 CV's in Datei schreiben

Alle CV's und ihre Werte können über den Button  in eine Datei geschrieben werden. Es öffnet sich ein Dialog-Fenster zur Auswahl der Datei. Es handelt sich um eine einfache XML-Datei (Beschreibung unter 11.9).

9.5 CV's aus Datei lesen

Ein, wie unter 9.4 beschrieben, abgespeicherte CV-Werteliste kann über den Button  wieder eingelesen werden.

ACHTUNG: Diese eingelesenen Werte werden nur in Spalte „neuer Wert“ eingetragen und **NICHT** automatisch an den BiDiB-Knoten gesendet.

9.6 CV-Definitionsdatei (XML)

9.6.1 Einem Knoten eine CV-Definitionsdatei zuweisen

Um die CV's eines Knoten bearbeiten zu können, müssen die Angaben zu den CV's in einer XML-Datei abgelegt werden. Diese Dateien werden in dem Verzeichnis „`data\BiDiBNodeVendorData\`“ unterhalb des Verzeichnisses in dem der BiDiB-Monitor liegt, gesucht.

Der Monitor sucht dabei als erstes nach der Datei, die in der Produktdefinition in *BiDiBProductID.xml* unter `<vendorcvfile></vendorcvfile>` eingetragen ist.

Dieser Eintrag darf aus einem Pfad bestehen, der unterhalb des Verzeichnisses „`data\BiDiBNodeVendorData\`“ liegt und muss immer mit dem Dateinamen enden.

Beispiele:

```
<vendorcvfile>Lightcontrol.xml</vendorcvfile>
```

Gesucht wird nach „`data\BiDiBNodeVendorData\Lightcontrol.xml`“

```
<vendorcvfile>LC\Lightcontrol.xml</vendorcvfile>
```

Gesucht wird nach „`data\BiDiBNodeVendorData\LC\Lightcontrol.xml`“

Wenn der Eintrag nur aus einem Bindestrich besteht `<vendorcvfile>-</vendorcvfile>` oder die Datei nicht vorhanden ist, wird im Verzeichnis „`data\BiDiBNodeVendorData\`“ automatisch nach einer Datei mit einem Dateinamen gesucht, der sich wie folgt zusammensetzt:

BiDiBCV-<VendorID>-<PID>.xml

Dabei sind `<VendorID>` und `<PID>` als Integer (ohne führende Nullen) zu verwenden.

Die Datei für eine LightControl hat also folgenden Namen:

BiDiBCV-13-107.xml

9.6.2 Format der CV-Definitionsdatei

ACHTUNG: Alle XML-Tags und Attribute sind Case-Sensitiv. Groß- und Kleinschreibung muss beachtet werden.

`<VendorCV></VendorCV>` ist also nicht das Gleiche wie `<vendorcv></vendorcv>`.

Die XML-Datei besteht aus drei Teilen.

Im ersten Teil werden Informationen zur Version der Datei hinterlegt.

Im zweiten können (optional) Templates für mehrfach vorkommende CV-Blöcke definiert werden.

Im dritten Abschnitt wird die Baumstruktur festgelegt und die CV's definiert.

Alle Teile müssen innerhalb von `<VendorCV></VendorCV>` stehen.

```
<?xml version="1.0" encoding="utf-8" ?>
<VendorCV>
  <Version>
    ...
  </Version>
  <Templates>
    ...
  </Templates>
  <CVDefinition>
    ...
  </CVDefinition>
</VendorCV>
```

9.6.2.1 Informationsteil der XML-Datei

```
<Version Version="0.2" Lastupdate="20130517" Author="BiDiB Mustermann"
        Pid="102" Vendor="13" Description="GBMBoost Master with Interface">
</Version>
```

Die Attribute **Version**, **Author** und **Description** können beliebigen Text enthalten.

Pid und **Vendor** enthalten die entsprechenden Teile der UID in dezimalen Werten.

Lastupdate gibt das Datum der letzten Änderung in **YYYYMMTT** an.

9.6.2.2 CV definieren

Eine `<CV></CV>` Definition bildet immer einen Endknoten in der Baumstruktur.

Eine CV wird mit folgendem XML-Block definiert

```
<CV Number="1" Type="Byte" Min="-" Max="-" Low="-" High="-" Values="-"
    Mode="rw" Rebootneeded="true"
    <Description Lang="de-DE" Text="Parser Modus"/>
    <Description Lang="en-EN" Text="Parser Mode" />
</CV>
```

Number	Die Nummer der CV
Type	Datentype: „Byte“, „Int“, „Bit“, „GBM16TReverser“
Min	Minimalwert für diese CV (bei Type „Byte“)
Max	Maximalwert für diese CV (bei Type „Byte“)
Low	Bei Type=Int wird hier die untere (low Byte) CV eingetragen.
High	Bei Type=Int wird hier die hohe (high Byte) CV eingetragen.
Values	Gültige Werte für diese CV. Die Werte werden durch „;“ getrennt.
Mode	ro = CV nur lesbar, rw = CV schreib- u. lesbar, wo = CV nur schreibbar
Rebootneeded	Änderung dieser CV wird erst bei Neustart des Knoten wirksam.

Gültige Werte vorgeben :

Bei die Typen „Bit“ und „Byte“, können dem Anwender, mit **min**, **max** und **values** Werte für die CV vorgegeben werden.

Wenn **values** eine, durch „;“ getrennte, Liste von Werten enthält, werden **min** und **max** nicht berücksichtigt.

9.6.2.3 CV-Type „Bit“

Ist der Type „Bit“ für die CV angeben, öffnet sich beim Anklicken des CV in der Baumstruktur eine Eingabemaske in der jedes zugelassene Bit einzeln gesetzt werden kann.

Optional, kann für jedes Bit eine Beschreibung in Deutsch und Englisch hinterlegt werden.

Beispiele für Type=„Bit“:

Min und **Max** werden hier nicht berücksichtigt.

In **Values** kann eine Bitmaske vorgegeben werden. Ein Integerwert zwischen 0 und 255 gibt an, welche Bits im Eingabefeld freigegeben sind.

Values="1" nur Bit 0 kann auf 0 oder 1 gesetzt werden.

Values="5" nur Bit 0 und 2 können auf 0 oder 1 gesetzt werden.

```
<CV Number="54" Type="Bit" Min="-" Max="-" Low="-" High="-" Values="5" Mode="rw">
    <Description Lang="de-DE" Text="Messstrecke aktiviert"/>
    <Description Lang="en-EN" Text="measuring track is aktive"/>
    <Bitdescription Bitnum="1" Lang="de-DE" Text="Messstrecke aktiv"/>
    <Bitdescription Bitnum="1" Lang="en-EN" Text="measuring track aktive"/>
    <Bitdescription Bitnum="3" Lang="de-DE" Text="Speed Meldung an Host senden"/>
    <Bitdescription Bitnum="3" Lang="en-EN" Text="send speed message to host"/>
</CV>
```

9.6.2.4 CV-Type „Byte“

Der Type „Byte“ kann durch einen Doppelklick in der Spalte „Neuer Wert“ geändert werden. Es öffnet sich dort eine ComboBox mit den gültigen Werten.

Beispiele für `Type="Byte"`:

Das Auswahlfeld zur Änderung des Wertes enthält nur die Werte von 3 bis 20.

```
<CV Number="51" Type="Byte" Min="3" Max="20" Low="-" High="-" Values="-" Mode="rw">
  <Description Lang="de-DE" Text="Test CV"/>
  <Description Lang="en-EN" Text="Text CV"/>
</CV>
```

Das Auswahlfeld zur Änderung des Wertes enthält nur die Werte von 4, 6, 13 und 20. `min` und `max` werden ignoriert.

```
<CV Number="51" Type="Byte" Min="3" Max="20" Low="-" High="-" Values="4;6;13;20"
  Mode="rw" >
  <Description Lang="de-DE" Text="Test CV"/>
  <Description Lang="en-EN" Text="Text CV"/>
</CV>
```

Das Auswahlfeld zur Änderung des Wertes enthält nur die Werte von 0 bis 255.

```
<CV Number="51" Type="Byte" Min="-" Max="-" Low="-" High="-" Values="-"
  Mode="rw">
  <Description Lang="de-DE" Text="Test CV"/>
  <Description Lang="en-EN" Text="Text CV"/>
</CV>
```

9.6.2.5 CV-Type „Int“

Der Type „Int“ setzt sich immer aus zwei CV's zusammen, die dann Low-Byte und High-Byte bilden. Beim Klick auf eines der beiden CV wird geprüft, ob die Partner CV vorhanden ist. Wenn ja, wird im rechten, unteren Bereich eine Eingabemaske angezeigt

Beide CV's müssen mit den Attributen `Low` und `High` aufeinander verweisen.

```
<CV Number="1" Type="Int" Min="-" Max="-" Low="1" High="9" Values="-" Mode="rw">
  <Description Lang="de-DE" Text="DCC Adresse (low)" />
  <Description Lang="en-EN" Text="DCC Address (low)" />
</CV>
<CV Number="9" Type="Int" Min="-" Max="-" Low="1" High="9" Values="-" Mode="rw">
  <Description Lang="de-DE" Text="DCC Adresse (high)" />
  <Description Lang="en-EN" Text="DCC Address (high)" />
</CV>
```

9.6.2.6 CV-Type „DCC_ADDR_RG“

Der Type „DCC_ADDR_RG“ setzt sich immer aus zwei CV's zusammen und bilden Low-Byte und High-Byte. Beim Klick auf eines der beiden CV wird geprüft, ob die Partner CV vorhanden ist. Wenn ja, wird im rechten, unteren Bereich eine Eingabemaske angezeigt

Dieser Type arbeitet ähnlich wie „Int“. Das untere Bit der Lowbyte CV wird aber extra über zwei Radio-Button gesetzt.

Der gültige Eingabebereich wird über die Attribute **min** und **max** der Lowbyte CV bestimmt.

Beide CV's müssen mit den Attributen **low** und **high** aufeinander verweisen.

```
<CV Number="59" Type="DCC_ADDR_RG" min="0" max="2040" low="59" high="60" values="-"
mode="rw">
  <Description Lang="de-DE" Text="Inhibit Off, Addr (Low)"/>
  <Description Lang="en-EN" Text="Inhibit Off, Addr (Low)"/>
</CV>
<CV Number="60" Type="DCC_ADDR_RG" min="0" max="2040" low="59" high="60" values="-"
mode="rw">
  <Description Lang="de-DE" Text="Inhibit Off, Addr (High)"/>
  <Description Lang="en-EN" Text="Inhibit Off, Addr (High)"/>
</CV>
```

9.6.2.7 CV-Type „GBM16TReverser“

Dieser Type ist nur für einen GBM16T zugelassen. Siehe 9.2.2.1 .

```
<CV Number="46" Type="GBM16TReverser" min="0" max="255" low="-" high="-" values="-"
mode="rw" >
  <Description Lang="de-DE" Text="Kehrschleife Modus" />
  <Description Lang="en-EN" Text="Reverser Mode"/>
</CV>
```

9.6.3 Template erstellen

Es kommt bei BiDiB-Knoten vor, dass für bestimmte Komponenten immer wieder die gleichen CV-Strukturen aufgebaut werden müssen. Zum Beispiel für die 32 LED-Ausgänge der Lightcontrol.

Um dies zu vereinfachen, wurde die Idee der Templates (danke Wolfgang) geboren.

Alle Templates müssen innerhalb der XML-Tags `<Templates></Templates>` definiert werden.

Die nächste Ebene legt den Namen des Templates fest.

Beispiel:

```
<Templates>
  <Template Name="LED">
    <CV>
      ...
    </CV>
    <CV>
      ...
    </CV>
  </Template>
  <Template Name="Servo">
    <CV>
      ...
    </CV>
  </Template>
</Templates>
```

Hier werden die beiden Templates „LED“ und „Servo“ definiert.

Innerhalb eines Templates kann wieder eine beliebige Baumstruktur definiert werden (siehe 9.6.4).

Auch Templates können innerhalb von Templates verwendet werden.

9.6.4 Baumstruktur definieren

Die Tags innerhalb `<CVDefinition></CVDefinition>` bestimmen die Gruppierung innerhalb der Baumstruktur.

9.6.4.1 Knoten im Baum

Jedes `<Node></Node>` -Tag bildet einen Knoten im Baum.

Um einem Knoten einen beschreibenden Namen zu geben, muss innerhalb von `<Node></Node>` mit dem Tag `<Nodetext Lang="de-DE" Text="Servo 0"/>` und den Attributen **Lang** und **Text** der Name des Knoten definiert werden.

```
<Node>
    <Nodetext Lang="de-DE" Text="Allgemeine Einstellungen"/>
    <Nodetext Lang="en-EN" Text="Common settings"/>
</Node>
```

Achtung: Wenn das Tag `<Nodetext></Nodetext>` fehlt, bekommt der Knoten im Baum den Namen „Node“ !!

9.6.4.2 Template verwenden

Um ein Template zu verwenden muß `<Node>` die Attribute **Offset** und **Template** enthalten.

```
<Node Offset="65" Template="Servo">
    <Nodetext Lang="de-DE" Text="Servo 0"/>
    <Nodetext Lang="en-EN" Text="Servo 0"/>
</Node>
```

Optional kann mit den Attributen **Count** und **Next** das Template wiederholt verwendet werden.

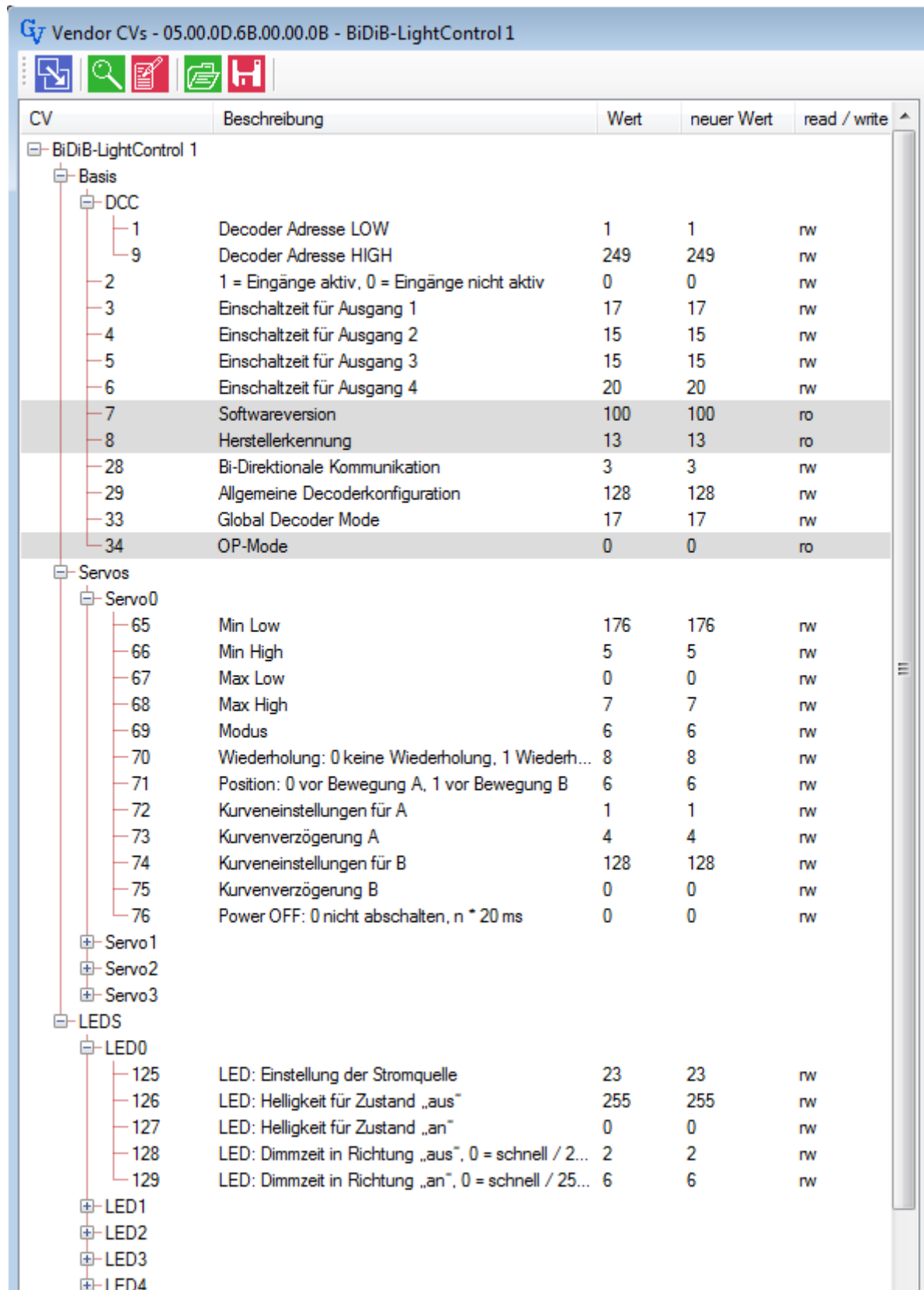
```
<Node Offset="65" Template="Servo" Count="4" Next="5">
    <Nodetext Lang="de-DE" Text="Servo 0"/>
    <Nodetext Lang="en-EN" Text="Servo 0"/>
</Node>
```

Hier wird das Template „Servo“, ab CV-Nummer 65 im Abstand von 5 viermal wiederholt.

Beispiel:

```
<CVDefinition>
    <Node>
        <Nodetext Lang="de-DE" Text="Basis"/>
        <Nodetext Lang="en-EN" Text="Basis"/>
        <Node>
            <Nodetext Lang="de-DE" Text="DCC"/>
            <Nodetext Lang="en-EN" Text="DCC"/>
            <CV>
                ...
            </CV>
        </Node>
    </CV>
    ...
    </CV>
</Node>
<Node>
    <Nodetext Lang="de-DE" Text="LED Einstellungen"/>
    <Nodetext Lang="en-EN" Text="LED settings"/>
    <Node Offset="125" Template="LED" Count="32" Next="5">
        <Nodetext Lang="de-DE" Text="LED %%d"/>
        <Nodetext Lang="en-EN" Text="LED %%d"/>
    </Node>
</Node>
</CVDefinition>
```

Für eine LightControl kann das dann so aussehen:



CV	Beschreibung	Wert	neuer Wert	read / write
BiDiB-LightControl 1				
Basis				
DCC				
1	Decoder Adresse LOW	1	1	rw
9	Decoder Adresse HIGH	249	249	rw
2	1 = Eingänge aktiv, 0 = Eingänge nicht aktiv	0	0	rw
3	Einschaltzeit für Ausgang 1	17	17	rw
4	Einschaltzeit für Ausgang 2	15	15	rw
5	Einschaltzeit für Ausgang 3	15	15	rw
6	Einschaltzeit für Ausgang 4	20	20	rw
7	Softwareversion	100	100	ro
8	Herstellereerkennung	13	13	ro
28	Bi-Direktionale Kommunikation	3	3	rw
29	Allgemeine Decoderkonfiguration	128	128	rw
33	Global Decoder Mode	17	17	rw
34	OP-Mode	0	0	ro
Servos				
Servo0				
65	Min Low	176	176	rw
66	Min High	5	5	rw
67	Max Low	0	0	rw
68	Max High	7	7	rw
69	Modus	6	6	rw
70	Wiederholung: 0 keine Wiederholung, 1 Wiederh...	8	8	rw
71	Position: 0 vor Bewegung A, 1 vor Bewegung B	6	6	rw
72	Kurveinstellungen für A	1	1	rw
73	Kurvenverzögerung A	4	4	rw
74	Kurveinstellungen für B	128	128	rw
75	Kurvenverzögerung B	0	0	rw
76	Power OFF: 0 nicht abschalten, n * 20 ms	0	0	rw
Servo1				
Servo2				
Servo3				
LEDS				
LED0				
125	LED: Einstellung der Stromquelle	23	23	rw
126	LED: Helligkeit für Zustand „aus“	255	255	rw
127	LED: Helligkeit für Zustand „an“	0	0	rw
128	LED: Dimmzeit in Richtung „aus“, 0 = schnell / 2...	2	2	rw
129	LED: Dimmzeit in Richtung „an“, 0 = schnell / 25...	6	6	rw
LED1				
LED2				
LED3				
LED4				

10 LogFile Analyse

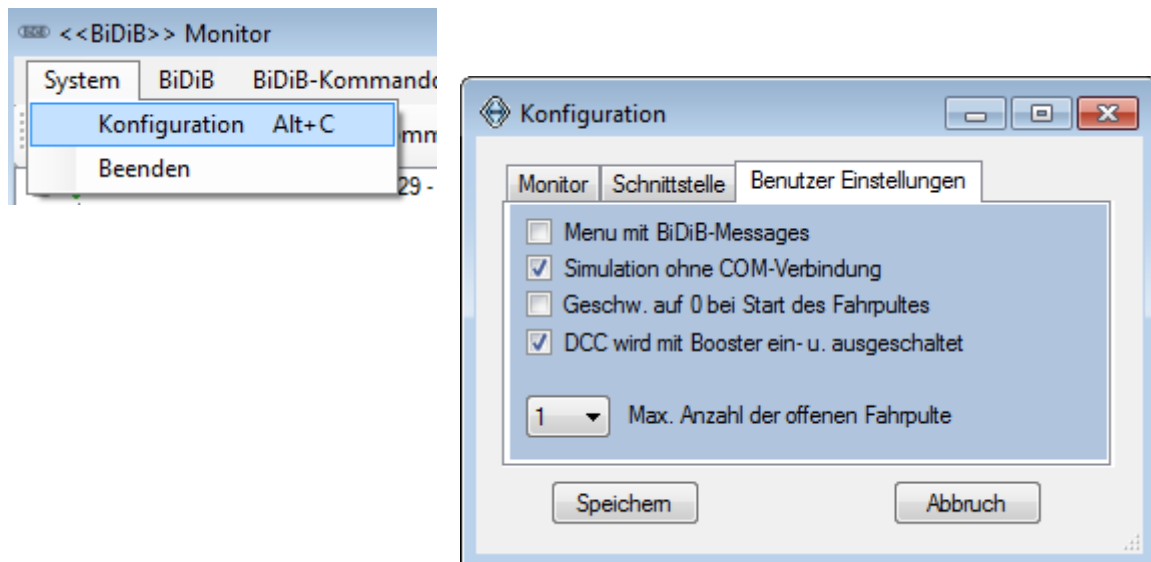
Diese Funktion kann eine Log-Datei einlesen und die dort enthaltenen Input-Zeilen als Input vom Interface auswerten. Damit können die protokollierten Abläufe im Monitor wiederholt werden, ohne dass ein Interface angeschlossen ist.

Ein Anwender, der ein Problem mit seiner BiDiB-Umgebung hat, kann den Monitor starten, die Log-Funktion in der Konfiguration aktivieren und Verbindung mit dem Interface aufnehmen. Die geschriebene Logdatei kann dann vom Support in den Monitor eingelesen und abgearbeitet werden.

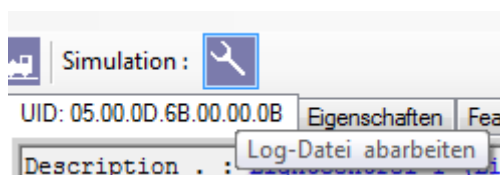
Im Monitor sind dann die Softwarestände und Features der angeschlossenen Knoten zu sehen.

10.1 LogFile-Analyse einblenden

Diese Funktion ist im Normalbetrieb ausgeblendet und kann, bei Bedarf über das Konfigurationsmenu aktiviert werden.



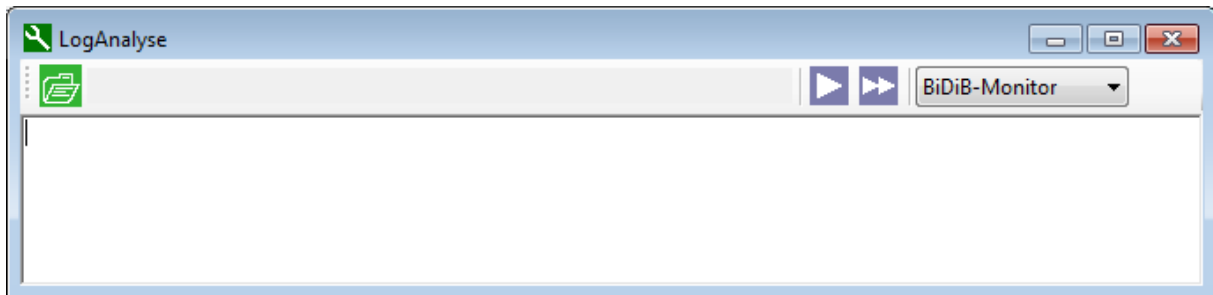
Wenn der Haken bei „Simulation ohne COM-Verbindung“ gesetzt ist, wird ein neues Icon in der Menüleiste eingeblendet.



ACHTUNG: Diese Einstellung wird NICHT gespeichert und ist beim nächsten Programmstart wieder deaktiviert.

10.2 LogFile-Analyse anwenden

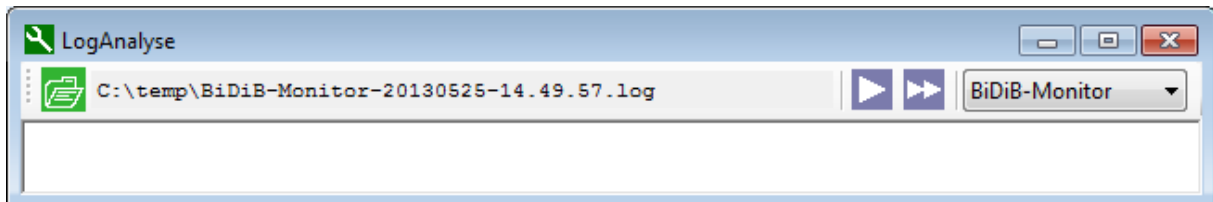
Durch Anklicken des Icons öffnet sich folgendes Fenster:



10.2.1 Logdatei auswählen



Beim Klick auf dieses Icon öffnet sich ein Dateidialog zur Auswahl der Logdatei.



10.2.2 Programmauswahl



Auswahl, von welchem Programm die Logdatei erstellt wurde.

Als Default ist hier der BiDiB-Monitor eingestellt.

Es kann aber auch einer BiDiB-Wizard-Datei eingelesen werden.

Der Wizard liest die Features eines Knoten erst, wenn der Knoten in der Liste das erste Mal angeklickt wird, Das ist KEIN Fehler, nur eine andere Vorgehensweise.

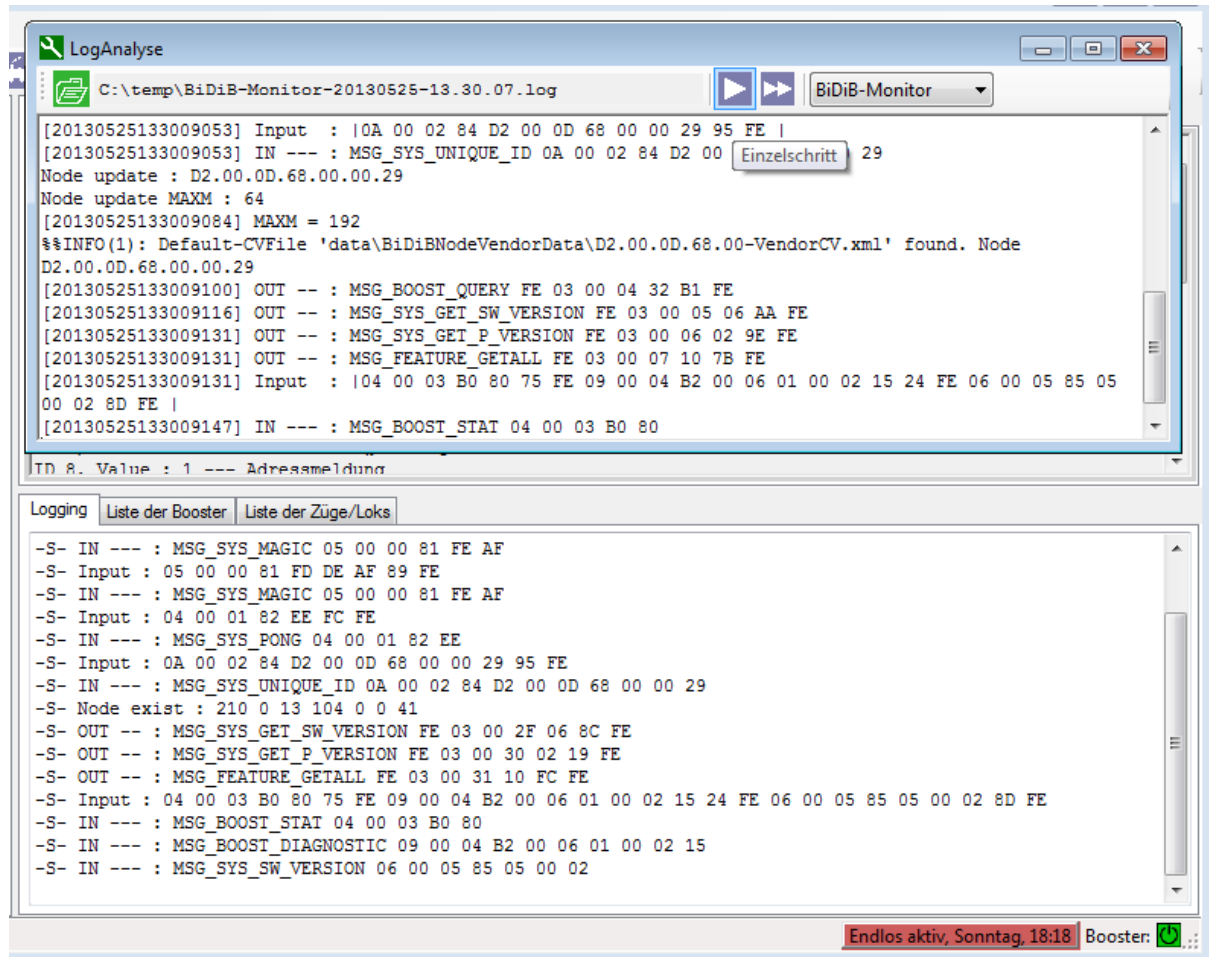
Dies führt aber im Monitor zu Problemen bis hin zum Absturz.

10.2.3 Einzelschritt



Die Zeilen der Logdatei werden Schritt für Schritt abgearbeitet.

Jede eingelesene Zeile wird in dem Fenster „LogAnalyse“ ausgegeben. Die Byte-Sequenzen in den „Input“-Zeilen werden dem Monitor als Input vom Interface übergeben. Der Monitor reagiert entsprechend mit den jeweiligen Messages an das (nicht vorhandene) Interface.



10.2.4 Massenhafte Abarbeitung der Zeilen



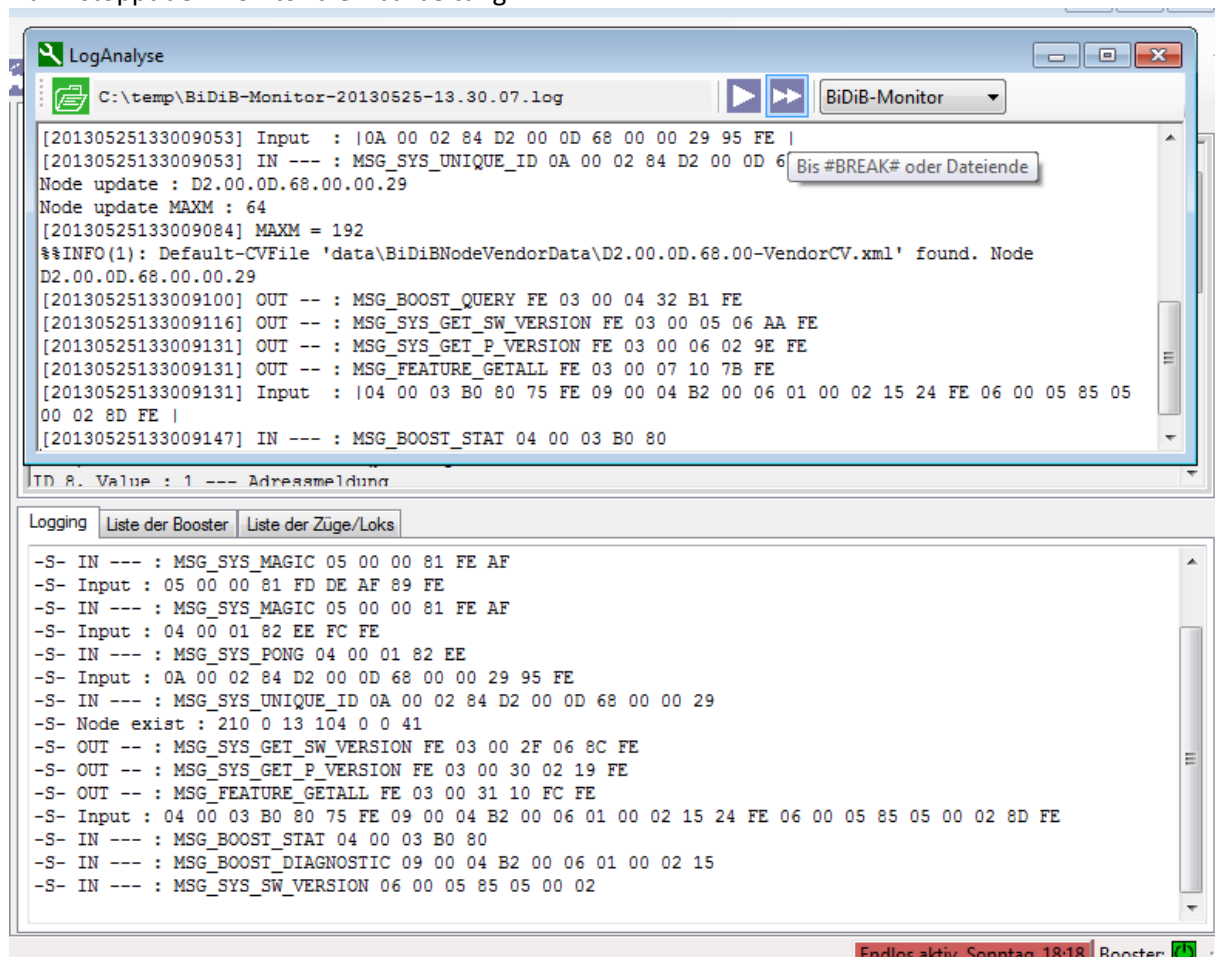
Alle Zeilen bis zum Ende der Datei oder bis zu einer Zeile mit **#BREAK#** abarbeiten.
Es werden alle Zeilen wie unter 10.2.3 abgearbeitet. Entweder bis zum Ende der Datei oder bis eine Zeile mit dem Text **#BREAK#** gefunden wird.

Eine solche Zeile kann an beliebiger Stelle in die Datei eingefügt werden. Hier ein Beispiel:

```
...
[20130203172659890] OUT -- : MSG_NODETAB_GETNEXT FE 03 00 24 0C D1 FE
[20130203172659906] Input : |04 00 23 8B FF 56 FE |
[20130203172659921] IN --- : MSG_NODE_NA 04 00 23 8B FF
[20130203172659921] OUT -- : MSG_SYS_ENABLE FE 03 00 25 03 54 FE
#BREAK#
[20130203172659952] Input : |04 00 24 B1 39 F1 FE |
[20130203172659952] IN --- : MSG_BOOST_CURRENT 04 00 24 B1 39
...
```

Die Zeile „Input : |04 00 23 8B FF 56 FE |“ wird noch übergeben.

Dann stoppt der Monitor die Abarbeitung



Hier kann man jetzt wieder per Einzelschritt weitermachen.

11 Anhang A: Die XML-Dateien

ACHTUNG: Wenn diese Daten verändert werden und die Struktur nicht eingehalten wird, kann es zu Fehlfunktionen der Anwendung kommen.

11.1 BiDiB-Messages.xml

Diese Datei enthält alle definierten BiDiB-Messages. BiDiB-Monitor verwendet diese Datei um die eingehenden Messages zu erkennen und die ausgehenden Messages zu erstellen.

Seit Version **0.4.3.0** enthält die Datei am Anfang ein Info-Block mit Versionsnummer. Dies soll sicherstellen, dass die Message-Definitionen dem aktuellen BiDiB-Standard entsprechen.

`<Version Number="0.19" LastUpdate="20130910" Author="A.Tillner"/>` Wenn diese Versionsangabe nicht mit der, im Monitor hinterlegten Version übereinstimmt, wird beim Starten des Monitors diese Warnung ausgegeben:



Der Monitor wird trotzdem gestartet.

Hier ein Beispiel für die Definition von **MSG_SYS_GET_MAGIC** und **MSG_NODE_NEW**:

```
<Msg Type="System" Direction="Down" Messagenum="01" Message="MSG_SYS_GET_MAGIC" />
...
<Msg Type="System" Direction="Up" Messagenum="8d" Message="MSG_NODE_NEW" />
```

11.2 DCCManufacturer.xml

Diese Datei enthält alle DCC-Hersteller ID's

```
<?xml version="1.0" encoding="utf-8" ?>
<DCCManufacturer>
...
    <Manufacturer Id="13" Name="Public Domain & Do-It-Yourself Decoders" />
...
</DCCManufacturer>
```

11.3 BiDiB-Decoder.xml

Diese Datei enthält alle Lok-Dekoder

```
<NewDataSet>
  <Decoder>
    <Name>V 211 zum Testen</Name>
    <Address>100</Address>
    <SpeedSteps>28</SpeedSteps>
    <Functions>4</Functions>
    <System>1</System>
    <CurSpeed>0</CurSpeed>
    <CurFunc>0</CurFunc>
  </Decoder>
</NewDataSet>
```

11.4 BiDiBProductID.xml

ACHTUNG: Die Funktion dieser Datei wurde mit der Version 0.4.4.0 geändert.

Diese XML-Datei enthält nur Information, die der Monitor zu den jeweiligen Knoten speichert. Für den Anwender sind diese Daten nicht relevant.

11.5 Products_<VendorID>.bidib

Ab der Version 0.4.5.0 wurde die Liste der Produkte in mehrere Dateien aufgeteilt.

Für jeden Hersteller gibt es eine Datei mit den jeweiligen Produkten. Für die Produkte mit der Hersteller-ID 13 („Public Domain & Do-It-Yourself Decoders“) ist dies z.B. [Products_13.bidib](#).

Die Dateien liegen alle in dem Unterverzeichnis [data\BiDiBProducts](#).

11.5.1 Beispiel:

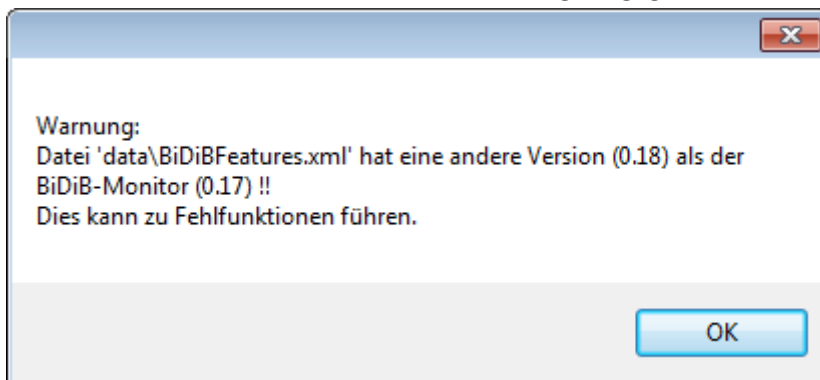
```
<BiDiB>
  <Products>
    <Product ManufacturerId="13" ProductTypeId="114" Name="OneHub">
      <Documentation Language="en-US" Text="BiDiBus Hub"
        Description="BiDiBOne OneHub, 8x input"
        Url="http://wiki.fichtelbahn.de/doku.php?id=onehub"/>
      <Documentation Language="de-DE" Text="BiDiBus Hub"
        Description="BiDiBOne OneHub, 8 Eingänge"
        Url="http://wiki.fichtelbahn.de/doku.php?id=onehub"/>
    </Product>
    ...
  </Products>
</BiDiB>
```

11.6 FeatureCodes.bidib

Diese Datei enthält die Definitionen der BiDiB-Features der einzelnen Knotentypen.
Wird dem Monitor ein Feature gemeldet und ist dieses Feature in dieser Datei definiert, wird das Feature bei den Eigenschaften des Knotens im Klartext angezeigt

Seit Version **0.4.3.0** enthält die Datei am Anfang ein Info-Block mit Versionsnummer. Dies soll sicherstellen, dass die Feature-Definitionen dem aktuellen BiDiB-Standard entsprechen.

`<Version Number="0.19" LastUpdate="20130910" Author="A.Tillner"/>` Wenn diese Versionsangabe nicht mit der, im Monitor hinterlegten Version übereinstimmt, wird beim Starten des Monitors diese Warnung ausgegeben:



Der Monitor wird trotzdem gestartet.

11.6.1 Beispiel

```
<?xml version="1.0" encoding="utf-8" ?>
<BiDiB> <!-- SchemaVersion="1.0" xmlns="http://www.bidib.org/schema/bidib" -->
  <Version Number="0.19" LastUpdate="20131004" Author="A.Tillner"/>
  <FeatureCodes>
    <FeatureCode Id="0" Name="FEATURE_BM_SIZE" >
      <Documentation Language="de-DE" Description="Anzahl der Belegtmeldungen"/>
      <Documentation Language="en-US" Description="Number of occupancy feedback
        channels."/>
    </FeatureCode>
    ...
    <FeatureCode Id="8" Name="FEATURE_BM_ADDR_DETECT_AVAILABLE" >
      <Documentation Language="de-DE" Description="Adressmeldung"/>
      <Documentation Language="en-US" Description="loco address data available "/>
    </FeatureCode>
  </FeatureCodes>
</BiDiB>
```

11.7 BiDiB-NodeProperties.xml

Diese Datei enthält die, in der Karteikarte „Eigenschaften“ eingegebenen Daten.

Diese Datei kann jederzeit gelöscht werden (ACHTUNG: Daten gehen dabei verloren).

Der BiDiB-Monitor legt die Datei wieder neu an.

```
<Node>
  <UID>0.0.0.0.0.0.0</UID>
  <FreeName>Dummy Node</FreeName>
</Node>
<Node>
  <UID>40.00.0D.CD.00.FE.04</UID>
  <FreeName>Simple Detector - fliegender Aufbau</FreeName>
</Node>
<Node>
  <UID>40.00.0D.69.00.00.02</UID>
  <FreeName>Sbf 2 - Links, Gleise 1-4, Endgleise 1 - 12</FreeName>
</Node>
```

11.8 XML-Datei zur CV-Definition

Die Datei muss folgendes Format haben (siehe 9.6):

```
<?xml version="1.0" encoding="utf-8" ?>
<VendorCV>
  <Version Version="0.4" Lastupdate="20130820" Author="Tillner, Andreas"
    Pid="104" Vendor="013" Description="GBMBoost Master">
  </Version>
  <Templates>
    <Template Name="LED">
      <CV>
        ...
      </CV>
    </Template>
  </Templates>
  <CVDefinition>
    <Node>
      <Nodetext Lang="de-DE" Text="Allgemeine Daten"/>
      <Nodetext Lang="en-EN" Text="Basis"/>

      <CV Number="1" Type="Byte" Min="0" Max="255" Low="-" High="-"
        Values="-" Mode="ro" >
        <Description Lang="de-DE" Text="Parser Modus" />
        <Description Lang="en-EN" Text="Parser Mode"/>
      </CV>
      <CV Number="2" Type="Byte" Min="0" Max="255" Low="-" High="-"
        Values="-" Mode="ro" >
        <Description Lang="de-DE" Text="DCC Herstellerkennung" />
        <Description Lang="en-EN" Text="DCC Vendor ID"/>
      </CV>
      ...
    </Node>
  </CVDefinition>
</VendorCV>
```


11.9 Format der XML-Datei zur Speicherung der CV-Werte

Beispiel für eine Datei mit CV-Werten einer LightControl:

```
<?xml version="1.0" standalone="yes"?>
<SaveCV>
  <MasterData>
    <UID>05.01.0D.6B.00</UID>
  </MasterData>
  <CVS>
    <CV>1</CV>
    <Value>1</Value>
  </CVS>
  <CVS>
    <CV>2</CV>
    <Value>0</Value>
  </CVS>
  .
  .
  .
  <CVS>
    <CV>119</CV>
    <Value>6</Value>
  </CVS>
  <CVS>
    <CV>123</CV>
    <Value>0</Value>
  </CVS>
</SaveCV>
```

12 Anhang B: Versionsstand

01.05.2012 - V 0.0.0.3

- Initiale Freigabe

08.06.2012 - V 0.0.0.4

- Probleme **0001** behoben.
- Probleme **0002** behoben.
- **MSG_NODE_NEW** und **MSG_NODE_LOST** Implementiert.
- **BiDiBFeatures.xml** mit den Feature-ID's der bekannten Knotentypen (11.6).
- **MSG_SYS_ENABLE** als Kommando per Menu dazu (4.2).
- Anzeige der Features in der Detailansicht eines Knotens.
- Ausgabe aller bekannten Features als Liste (4.1.6.3).
- Farbige Darstellung der Detailansicht eines Knotens.
- Im TreeView wird der aktuelle Status des Knoten durch ein Icon angezeigt.
- Zusätzliches **MSG_SYS_ENABLE** nach dem Initialen einlesen der ersten Nodetab (Bug **0003**)

12.06.2012 - V 0.0.0.5

- Problem **0004** behoben.
- New: Menüpunkt „Eigenschaften“ im Kontextmenu der Knoten (Work in Progress).
- New: Das Feld <FID> in der XML-Datei **BiDiBFeatures.xml** wurde auf drei Stellen erweitert. Die Features für Bootloader und Zentrale sind hinzugekommen.

16.09.2012 - V 0.0.1.1

- New: In der Konfiguration können einige Einstellungen für die Kommunikation mit dem BiDiB-Interface vorgenommen werden (siehe 3.2).
- New: Das XML-File **BiDiBProductID.xml** wurde um vier Properties-Felder erweitert (siehe 11.4).
- New: Anzeige der Meldereingänge (siehe 4.3.2)
- New: Anzeige der Switcheingänge (siehe 4.3.3)

18.09.2012 - V 0.0.1.2

- Problem **0005** ist mit der Firmware 1.3.6 für die GBM16C Hardware behoben (W.Kufer)
- Problem **0007** behoben.
- Problem **0008** behoben.
- New: <F5> führt ein Refresh auf die Ansicht der Knoten aus.

21.09.2012 - V 0.0.1.4

- Problem **0009** behoben. Wartezeit von 1s nach jedem **MSG_SYS_RESET** (Danke an W.Kufer)
- Problem **0010** behoben. Parameter eingefügt.
- Problem **0011** behoben. Sortierfunktion abgeschaltet.
- Problem **0012** behoben. IOException wird entsprechend abgefangen. (Danke an W.Kufer)
- New: Inventory listet jetzt auch die Features auf (siehe 4.1.6.4).
- New: Daten der COM-Schnittstelle können im laufenden Programm geändert und angewendet werden (siehe 3.1)

29.09.2012 - V 0.0.2.1

- Problem **0013** behoben. Umstellung der seriellen Schnittstelle auf Eventhandling mit blockadefreien Leseoperationen.
- New: Features eines Knoten können geändert werden (siehe 4.3.4).
- New: Kontextmenü um spontane Abfrage der Belegt Melder erweitert (siehe 4.3.7.3).
- New: Die „Technische Ansicht“ der Knoten kann in eine „beschreibende Ansicht“ umgeschaltet werden (siehe 4.3.6.2).
- New: Die freie „Bezeichnung“ aus 4.3.6.2 wurde in die Inventory-Liste und Auflistung der Knoten-Eigenschaften aufgenommen.

05.10.2012 - V 0.0.2.2

- Problem **0014** behoben. Neue Log-Datei wird angelegt und verwendet.
- New: Die „Technische Ansicht“ kann über Taste <F4> ein- und ausgeschaltet werden.

18.10.2012 - V 0.0.2.3

- New: Messages in XML eingetragen **MSG_SYS_CLOCK**, **MSG_BM_CONFIDENCE**, **MSG_BM_GET_CONFIDENCE** und **MSG_LC_WAIT**.
- New: Auswertung und Anzeige von Adresse und Richtung (siehe 4.3.2.2) der Belegt Meldung **MSG_BM_ADDRESS**.

28.10.2012 - V 0.0.3.5

- Problem **0006** behoben: Ist durch den Neuaufbau des TreeView behoben.
- Problem **0015** behoben: Wenn CRC ein MAGIC oder ESC war wurde dies nicht maskiert.
- Problem **0016** behoben: Schnittstelle auf Hardware-Handshake eingestellt (Danke Kersten Tams).
- Problem **0017** behoben: Belegt- bzw. Freimeldungen für unbekannte, nicht in der Datei **BiDiBProductID.xml** definierte, PID's führen zum Absturz (Danke Kersten Tams).
- Problem **0018** behoben: Aufgleisrichtung wurde falsch ausgewertet und angezeigt. „<<“ (0b10nnnnnn) und „AA“ (0b01nnnnnn) waren vertauscht.
- Problem **0019** behoben: Produkterkennung wurde auf die VenderID und das zweite PID-Byte erweitert. Dazu wurde das XML-File **BiDiBProductID.xml** um einige Felder erweitert (siehe 11.4) (Danke W.Kufer).
- New: Systemuhr mit einstellbarer Geschwindigkeit / Anfangs- u. Endzeit / Wiederholung (siehe 8).
- New: Anhand der DCC-Hersteller ID wird der Hersteller im „Info-Bereich“ ausgegeben.
- New: Liste der DCC-Hersteller in XML-File **DCCManufacturer.xml** und Ausgabe der List (siehe 4.1.6.4).

31.10.2012 - V 0.0.3.7

- Problem **0020** behoben: **MSG_NODETAB_COUNT** wird erst 1500 ms nach dem **MSG_SYS_ENABLE** an das Interface gesendet. Solange **MSG_NODETAB_COUNT** den Wert 0 liefert, wird alle 1500 ms die Abfrage der Nodetabelle mit **MSG_NODETAB_GETALL** wiederholt.
- New: In der Logdatei wird für jede Message ein Zeitstempel geschrieben.

19.11.2012 - V 0.1.0.1

- Problem **0021** und **0023** behoben: Initialisierung der internen Knotenliste nach einem **MSG_SYS_RESET** wurde angepasst.
- Problem **0022** behoben: Darstellung der Switches/Eingänge wird jetzt auch in der „beschreibenden Ansicht“ aktualisiert.
- Problem **0024** behoben: **MSG_SYS_ENABLE** wird zusätzlich nach dem ersten einlesen einer gültigen Nodetabelle gesendet.
- Problem **0025** behoben: Nach dem die Feature-Tabelle eines Knoten eingelesen wurde, werden, in Abhängigkeit der Features, alle Melder/Adressen/Eingänge abgefragt.
- Problem **0026** behoben: Es werden alle Bytes an die serielle Schnittstelle übergeben.
- Problem **0028** behoben: Falsche Index-Verarbeitung bei Meldungen von Knoten die nicht im Baum angeklickt sind.
- New: Alle gemeldeten Adressen in einem Meldeabschnitt werden in der Tabelle dargestellt.
- New: Firmware-Update über BiDiB-Bus

22.11.2012 - V 0.1.0.2

- Problem **0029** behoben: Eintrag in XML-File **BiDiBProductID.xml** für MobaList angepasst und Aufbau der Tabelle für den Status der Eingänge angepasst.

04.12.2012 - V 0.1.2.0

- Problem **0030** behoben: Adresse des BiDiB-Knoten wurde bei der Auswertung von `MSG_BM_ADDRESS` nicht richtig verarbeitet, wenn es sich um einen Knoten am Bus handelt.
- Problem **0031** behoben: Eingehende Meldungen für einen Knoten werden nur ausgewertet, wenn alle Informationen für den Knoten eingelesen sind.
- New: Menüstruktur und Bedienung ist jetzt Sprachabhängig. Auf Systemen mit deutscher Sprache wird eine Deutschsprachige Oberfläche angezeigt, sonst Englisch.
- New: Über eine Einstellung in der Konfiguration kann das Menü „BiDiB-Kommandos“ und das Kontext-Menü der Knoten wieder auf eine Darstellung mit BiDiB-Messages umgestellt werden.
- New: Kontextmenü um spontane Abfrage der Adressen (`MSG_BM_ADDR_GET_RANGE`) erweitert (siehe 4.3.7.4).

05.12.2012 - V 0.1.2.6

- Problem **0032** behoben: Firmware-Update muss auf Vollständigkeit bestätigt werden.
- New: Booster-Funktionen werden unterstützt. On/Off per Menü. Statusanzeige unten rechts.

05.12.2012 - V 0.1.2.7

- Problem **0033** behoben: Abspeichern der beschreibenden „Bezeichnung“ bei neuen Knoten.
- Problem **0034** behoben: Speed wird auch ausgewertet, wenn Knoten noch nicht erfasst ist.
- New: Anzeige des gemeldeten Stromverbrauchs im Melder-Tab, unten, rechts.
- New: Weniger aufwendige Formatierung der Log-Ausgabe und damit schnelleres Scrollen.
- New: Um den Zugriff zu beschleunigen wurde der Schlüsselwert UID in der Datei `BiDiB-NodeProperties.xml` auf das, intern verwendete, HEX-Format umgestellt.

11.02.2013 - V 0.2.2.2

- Problem **0035** behoben: Bei Unterbrechung des Datenstroms von der COM-Schnittstelle wurde die Maskierung von 0xfe durch ein <ESC> nicht sauber erkannt (Danke an W.Kufer und A.Ostertag)
- Problem **0036** behoben: Speedmeldung wird sauber erfasst.
- New: Kleine Liste aller Booster und die Anzeige des aktuellen Stromverbrauchs als Balken.
- New: Simulationsmode zum Arbeiten ohne COM-Verbindung zu einem Interface.
- New: Logfile-Analyse. Eine Log-Datei kann als Eingabe-Simulation eines Interfaces dienen und damit eine BiDiB-Session offline wiederholen.

ACHTUNG: Für die Folgenden Neuerungen muss auf dem GBMBoost eine Firmware > 2.0.0 eingespielt sein.

- New: Möglichkeit, Lokdecoder zu definieren und zu Steuern (Geschwindigkeit, Funktionen FL – F28, Halt, NotHalt).
- New: Die Parameter für die Message `MSG_CS_POM` können in einem Fenster zusammengestellt und an das Interface geschickt werden.

15.02.2013 - V 0.2.2.3

- Problem **0039** behoben: Falsche Adressauswertung bei `MSG_BM_ADDRESS` führte zu falscher Auswertung von `MSG_BM_SPEED` (Danke Christoph)
- Problem **0040** behoben: Falsche Auswertung von `MSG_BM_SPEED` wenn das untere Adressbyte 0x00 ist (Danke Christoph).

03.03.2013 - V 0.2.2.6

- Problem **0041** behoben: Tabelle der Lok-Dekoder wird nur noch einmal dargestellt und nicht bei jedem Verbindungsaufbau eine neue aufgebaut.
- Problem **0042** behoben: Sequenz der Messagenummer beginnt, bei neuen Knoten mit 0.
- Problem **0043** behoben: `BiDiB-Messages.xml` auf den aktuellen Stand gebracht.

08.03.2013 - V 0.2.2.7

- Problem **0044** behoben: Die gemeldete Geschwindigkeit wird im Fahrpult angezeigt.
- Problem **0045** behoben: Bei POM Read Byte wird der Rückgabewert im POM-Fenster angezeigt.

08.03.2013 - V 0.2.2.8

- Problem **0046** behoben: Möglicher Absturz bei POM ByteRead beseitigt.
- New: Bei POM Read Byte wird der Rückgabewert in Dezimal-, Hex- u. Binär-Wert dargestellt.

08.03.2013 - V 0.2.2.9

- Problem **0047** behoben: Absturz bei Änderung der Melderzahl und neu verbinden beseitigt. (Danke Michael)

24.03.2013 - V 0.3.0.0

- New: Teile des Monitors wurden in die Library BiDiBLib.dll ausgelagert. Der Quellcode dieser Library und eine kleine Beispielanwendung stehen im ZIP-Archiv zur Verfügung.

28.04.2013 - V 0.3.0.3

- Update: Es können wieder mehrere Fahrpult-Fenster gleichzeitig geöffnet werden. Die maximale Anzahl kann über die Konfiguration eingestellt werden.
- New: Fenstergröße und -position beim Schließen werden gespeichert und beim nächsten Start wieder verwendet.
- New: XML-Dateien liegen ab dieser Version, in dem Unterverzeichnis „data“.
- Problem **0050** behoben: Textkonvertierung wurde angepasst.

30.04.2013 - V 0.3.0.5

- New: In der Konfiguration kann eingestellt werden, ob DCC mit dem Booster ein- und ausgeschaltet wird.
- New: Vor einem Firmware Update wird das DCC-Signal abgeschaltet.
- Problem **0051** behoben: Pfad zu den XML-Dateien angepasst „data*.xml“

25.05.2013 - V 0.4.0.4

- New: CV's der BiDiB-Knoten können in einer XML-Datei definiert und im Monitor bearbeitet werden.
- New: Die Bezeichnung der Features kann in die „technische Ansicht“ umgeschaltet werden.
- New: Wenn eine Homepage in der Produkt-XML-Datei eingetragen ist, kann sie per Kontextmenu oder in der Info-Darstellung aufgerufen werden.
- Problem **0049** behoben: Es wurde ein Timeout von 3 Sek. eingeführt. Wenn der Knoten in dieser Zeit nicht antwortet, gibt der Monitor eine entsprechende Fehlermeldung aus.
- Problem **0052** behoben: Wenn ein Knoten abgemeldet wird, werden im Infobereich die Daten des Interfaces angezeigt.
- Problem **0053** behoben: Konfiguration der COM-Schnittstelle bleibt erhalten wenn Simulationsmode aktiviert wird.

03.07.2013 - V 0.4.2.1

- New: CV's Verwaltung mit weiteren Eingabemöglichkeiten für CV's mit Low/High (int), bitweise Eingabe und die Kehrschleife eines GBM16T.
- New: CV-Definitionsdatei mit der Möglichkeit Templates für sich wiederholende CV-Strukturen zu verwenden.
- New: Watchdog des GBMBoost wird unterstützt.
- New: Unterstützung von weiteren Busebenen (OneHub)

07.07.2013 - V 0.4.2.3

- New: Die CV Verwaltung wurde um den Type „DCC_ADDR_RG“ erweitert.
- Problem **0054** behoben: Registrykey wurde nicht geschrieben.

27.07.2013 - V 0.4.3.1

- New: Die Dateien *BiDiB-Messages.xml* und *BiDiBFeatures.xml* sind mit einer Versionsnummer versehen. Das soll sicherstellen, dass sie dem aktuellen Stand des BiDiB-Standards entsprechen.
- Update: Die Features werden Aufsteigend nach Feature-Nummer aufgelistet.
- Problem **0055** behoben: Abfrage in CV-Verwaltung korrigiert. „ro“ ist jetzt für alle Datentypen „read only“.
- Problem **0056** behoben: HEX-Datei wird direkt nach dem Ende des Upload freigegeben.
- Problem **0057** behoben: XML-Datei wird generiert, wenn sie noch nicht existiert.
- Problem **0058** behoben: Bei leerer Nodetabelle des Interface läuft Timer jetzt immer wieder an. Bis ein Knoten gemeldet wird.
- Problem **0059** behoben: Pfad zu DCCManufacturer.xml korrigiert.
- Problem **0060** behoben: Datenstrukturen der Eingänge werden auch aufgebaut, wenn nur das Feature 53 eine Anzahl von Eingängen meldet, diese aber nicht in der *BiDiBProductID.xml* angegeben sind.
- Problem **0061** behoben: Datenstrukturen der Belegtmelder werden auch aufgebaut, wenn nur das Feature 0 eine Anzahl von Meldern meldet, diese aber nicht in der *BiDiBProductID.xml* angegeben sind.
- Problem **0062** behoben: Der Analyser kann jetzt Logfiles vom Monitor („Input : „) und die Ausgabe des Log-Windows („IN --- : „) als Eingabe verwenden.

20.08.2013 - V 0.4.3.6

- Problem **0063** behoben: *MSG_NODETAB_COUNT* für Interfaceknoten (HUB), die nicht auf der obersten Ebene liegen, wurde falsch ausgewertet.
- Problem **0064** behoben: *MSG_NODE_CHANGE_ACK* wurde an den falschen Knoten geschickt.
- Problem **0065** behoben: In BiDiBLib.dll die Längenabfrage auf 64 Byte berichtigt.

03.10.2013 - V 0.4.4.0

- New: Bei den Daten eines Knotens wird, wenn vorhanden, ein Bild angezeigt. Siehe 4.3.1.
- New: Die Knotenerkennung wertet das neue Feature *FEATURE_RELEVANT_PID_BITS* aus.
- New: *MSG_SYS_ERROR* wird als „sprechende“ Meldung im Log ausgegeben. Bei Errorcode 0x01 wird auch der übertragene Text angehängt.
- Update: Neue Features und Messages in die XML-Dateien aufgenommen.
- Update: CV-Verwaltung (Siehe 9)
 - Der Name der Default-Datei eines Knoten, wurde in *BiDiBCV-<VendorID>-<PID>.xml* geändert.
 - Für die CV-Type „Bit“ kann für jedes Bit eine Beschreibung hinterlegt werden.
 - Die Baumknoten in der CV-Verwaltung können aus einem beliebigen Text gebildet werden.
- Update: Änderung der meisten XML-Dateien. Zur Vorbereitung auf ein einheitliches Format einiger XML-Dateien, wurden Attribute eingeführt.
- Update: In BiDiBLib.dll wurden die Methoden zur Auswertung der Datei *BiDiBMessages.xml* an das neue Format angepasst.
- Problem **0066** behoben: Bytecode für Message *MSG_LC_MACRO_PARA_GET* wurde *BiDiBMessages.xml* in auf **0x4c** geändert..
- Problem **0067** behoben: Das initialisieren der Liste der Melder wurde überarbeitet (Danke Hr.Tams)
- Problem **0068** behoben: Die Auswertung des Feature *FEATURE_CTRL_INPUT_COUNT* überarbeitet.

27.10.2013 - V 0.4.5.0

- New: Unterstützung des reduzierten Bootloaders (BIDIB_BOOT_MAGIC) auf Node und Interface (Danke Hr. Tams)
- New: Per ToolTip werden die wichtigsten Informationen eines Knoten in der Baumstruktur angezeigt.
- New: Die XML-Datei *BiDiBFeature.xml* wurde durch *FeatureCodes.bidib* ersetzt (abgestimmtes XML-Format)
- New: Die XML-Datei *BiDiBProducts.xml* wurde geteilt (je Hersteller ein File) und durch Dateien im abgestimmten Format ersetzt. Die XML-Dateien liegen jetzt im Unterverzeichnis *BiDiBProducts*.
Dateiname: *Products_<VendorID>.bidib*
- Update: CV-Verwaltung (Siehe 9.6.4)
 - Mit den Node-Attributen **Count** und **Next** kann mit einem Template, wie in einer Schleife, viele gleiche Baumknoten angelegt werden.
 - Mit dem CV-Attribute **Rebootneeded="Yes"** wird angegeben, das bei Änderung dieser CV, eine Reboot des Knoten nötig ist.
 - Um eine gemeinsame XML-Struktur zu erreichen, können Baumknoten jetzt nur noch mit dem XML-Tag `<Node></Node>` angegeben werden.
- Update: Anzeige der PID in der Baumansicht angepasst. Feature **FEATURE_RELEVANT_PID_BITS** wird berücksichtigt.
- Update: Ausgabe im Log-File kann auf ein, besser lesbares, Ausgabeformat eingestellt werden.
- Problem **0069** behoben: Jedem neuen Knoten wird eine **MSG_SYS_GET_MAGIC** geschickt.
- Problem **0070** behoben: Wechselnde Anzahl der Switch-Eingänge wird berücksichtigt.
- Problem **0071** behoben: Auswertung von BIDIB_BOOT_MAGIC angepasst.
- Problem **0072** behoben: Bei **FEATURE_RELEVANT_PID_BITS** <> 16 wird der Name und Beschreibung des Knoten noch mal im XML gesucht.
- Problem **0073** behoben: Semaphore in Methode **ExtractMsg()** der BiDiBLib.dll. Dies muss von der Methode in **ProceedMsg()** wieder zurückgesetzt werden.
- Problem **0074** behoben: Ausgabe im Debug-Mode hatte Message-Counter für den Knoten hochgezählt.
- **Removed: Bisheriges Format der CV-XML-Dateien (ohne Attribute) wird nicht mehr unterstützt.**

02.11.2013 - V 0.4.5.1

- Update: CV-Attribute **Rebootneeded="true"** wird mit den Werten **"true"** und **"false"** angegeben, nicht mehr **"Yes"** und **"No"**
- Update: Neues Icon für OneOC

18.11.2013 - V 0.4.6.0

- Problem **0075** behoben: Knotentabelle wird auch gelesen, wenn Hub erst später eingeschaltet wird.
- Problem **0076** behoben: Richtungsbit im Fahrpult wird beibehalten, wenn Geschwindigkeit auf 0 geht.

24.11.2013 - V 0.4.6.2

- Update: Meldungsfenster, wenn keine Antwort auf **MSG_SYS_GET_MAGIC** erfolgt.
- Problem **0077** behoben: Richtungsbit im Fahrpult korrigiert.

13 Anhang C: Bekannte Probleme

V 0.0.0.3

Bug	Beschreibung
0001	Wenn per Kontextmenu <code>MSG_FEATURE_GETALL</code> an das Interface geschickt wird, wird nach dem Empfang von <code>MSG_FEATURE_NA</code> noch ein <code>MSG_NODETAB_GETALL</code> an das Interface geschickt.
0002	<code>MSG_NODE_NA</code> wird falsch interpretiert. Endlosschleife durch Anfrage an den nicht mehr vorhandenen Knoten bei Parameter < 255.
0003	<code>MSG_SYS_IDENTIFY_STATE</code> von den Knoten, wird nicht an den BiDiB-Monitor weitergegeben. Erst nach dem der BiDiB-Monitor noch mal ein <code>MSG_SYS_ENABLE</code> an das Interface geschickt hat, wird der Status übertragen.

V 0.0.0.4

Bug	Beschreibung
0004	Anzahl der Knoten in der Nodetabelle wird nicht richtig ausgewertet. Daher werden weniger Knoten angezeigt als tatsächlich in der Tabelle sind.

V 0.0.0.5

Bug	Beschreibung
0005	<code>MSG_NODE_NEW</code> und <code>MSG_NODE_LOST</code> werden nicht empfangen oder vom Interface nicht gesendet (wird geprüft), wenn ein Knoten ein- bzw. abgeschaltet wird.
0006	Wenn die Nodetabelle neu eingelesen wird, wird nicht geprüft, ob die UID ↔ Adresse Zuordnung noch stimmt. Es kann daher zu falschen Anzeigen kommen.

V 0.0.1.1

Bug	Beschreibung
0007	<code>MSG_NODE_NEW</code> und <code>MSG_NODE_LOST</code> werden vom Interface empfangen aber nicht korrekt abgearbeitet. Knoten werden nicht sauber gelöscht. Wieder angeschlossene Knoten sind daher noch in der verketteten Liste enthalten und werden nicht neu hinzugefügt. UID und Adresse stimmen dann nicht immer überein.
0008	<code>MSG_NODE_CHANGE_ACK</code> wird an das Interface geschickt. Die Version der Nodetabelle ist aber falsch von <code>MSG_NODE_NEW</code> bzw. <code>MSG_NODE_LOST</code> übernommen.

V 0.0.1.2

Bug	Beschreibung
0009	Bei <code>MSG_SYS_RESET</code> wird der Bus für 1s „Still gelegt“. Der Monitor hat diese Zeit nicht abgewartet. (Danke W.Kufer)
0010	<code>MSG_SYS_PING</code> wird ohne Parameter abgeschickt.
0011	Die Tabellen der Meldereingänge können, durch Anklicken der Spaltenüberschriften sortiert werden. Dies führt bei der ersten Spalte dazu, dass die Informationen der Gruppiereten Spalten („0. ...“) nicht mehr zusammen gehören.
0012	Wenn die COM-Schnittstelle auf einen Port konfiguriert ist, an der kein Gerät angeschlossen ist, stürzt das Programm ab. (Danke W.Kufer)

V 0.0.1.4

Bug	Beschreibung
0013	Weiterhin Absturz beim Wechsel der COM-Schnittstelle.

V 0.0.2.1

Bug	Beschreibung
0014	Löschen des Log-Bereiches mit <STRG-Umschalttaste-X> führt zu Fehlermeldung beim Zugriff auf die neue Log-Datei.

V 0.0.2.2

Bug	Beschreibung
0015	Fehler bei der Übertragung einer Message, wenn eines der Zeichen 0xfe (Magic) oder 0xfd (esc) ist.

V 0.0.2.3

Bug	Beschreibung
0016	RTS/CTS stehen nicht zur Verfügung (Danke Kersten Tams)
0017	Absturz, wenn PID eines Knoten in der Datei BiDiBProductID.xml nicht vorhanden ist, aber Belegt- oder Freimeldungen von diesem Knoten kommen (Danke Kersten Tams)
0018	Aufgleisrichtung wird falsch ausgewertet und angezeigt.
0019	Produkterkennung entspricht nicht der BiDiB-Norm. Es wird nur das erste Byte der PID ausgewertet. VendorID und das zweite PID-Byte werden ignoriert. (Danke W.Kufer)

V 0.0.3.5

Bug	Beschreibung
0020	MSG_NODETAB_COUNT wird falsch ausgewertet und damit die Nodetabelle zu früh ausgelesen (Danke Karl Heinz Bury).

V 0.0.3.7

Bug	Beschreibung
0021	Nach MSG_SYS_RESET und neuem Einlesen der Busstruktur bleibt das Icon vor dem Interface rot (Danke W.Kufer).
0022	Status der Switch/Eingänge werden in der „beschreibende Ansicht“ nicht dargestellt. In der „technischen Ansicht“ funktioniert es.
0023	Die Monitor-Interne Liste der Knoten wird bei MSG_SYS_RESET und disconnect nicht sauber gelöscht. Daher kommt es danach zur Darstellung von nicht aktuellen Werten (Danke Frank).
0024	Nach MSG_SYS_ENABLE wird zwar 1 Sec. nach MSG_SYS_RESET gesendet. Dies reicht aber nicht um es an die Knoten, die noch nicht angemeldet sind, weiter zugeben.
0025	Nach einem MSG_SYS_ENABLE werden die Melder und Eingänge von Knoten nicht Initial abgefragt (Danke Frank).
0026	Die Maskierung von 0xfd und 0xfe wird zwar nach den Korrekturen von Problem 0015 durchgeführt, die „verlängerten“ Messages werden aber nicht vollständig übertragen.
0027	Bei den Einstellungen der Uhr wird immer Montag und nicht der zurzeit eingestellte Wochentag angezeigt.
0028	Absturz, bei eingehenden Belegt Meldungen in der „beschreibenden Ansicht“, wenn der, im Baum angeklickte Knoten, nicht der meldende ist.

V 0.1.0.1

Bug	Beschreibung
0029	Absturz bei Abfrage der Eingänge des MobaList (Danke Jürgen).

V 0.1.0.2

Bug	Beschreibung
0030	Auswertung von MSG_BM_ADDRESS liefert nicht alle Adressen wenn die Message von einem Unterknoten und nicht vom Interface kommt.
0031	Absturz bei früher Meldung eines Belegt Zustandes oder einer Adresse, wenn der Knoten noch nicht im Monitor erfasst ist.

V 0.1.2.0

Bug	Beschreibung
0032	Firmwareupdate: Knoten wird nach jedem Update-Schritt (Flash, EEPROM, etc.) in den Normalmodus geschaltet. Dies kann zu Fehlfunktionen des Knoten führen, wenn neue Firmware auch neue EEPROM-Daten benötigt (Danke W.Kufer)

V 0.1.2.6

Bug	Beschreibung
0033	Abspeichern der beschreibenden Bezeichnung unter der Karteikarte „Eigenschaften“ endet mit Fehlermeldung.
0034	Speed Meldung kann zu einer Endlosschleife führen.

V 0.1.2.7

Bug	Beschreibung
0035	Sporadische CRC-Fehler bei empfangenen Messages (Danke W.Kufer u. A. Ostertag)
0036	Speed Meldung kann zu einer Endlosschleife führen.

V 0.2.2.0

Bug	Beschreibung
0037	CheckBox der „Simulation ohne COM-Verbindung“ merkt sich nicht den eingestellten Zustand.
0038	Möglicher Absturz im Simulationsmode, wenn die COM-Schnittstelle angesprochen wird. (Danke Mirko)

V 0.2.2.2

Bug	Beschreibung
0039	Bei Aufgleisrichtung „rückwärts“ werden die Speedmeldungen nicht ausgewertet. (Danke Christoph)
0040	Wenn das untere Adress-Byte einer DCC-Adresse 0x00 ist wird die MSG_BM_SPEED nicht ausgewertet (Danke Christoph)

V 0.2.2.3

Bug	Beschreibung
0041	Die Karteikarte der Loks wird bei jedem Verbindungsaufbau neu aufgebaut.
0042	Die Sequenz der Messagezählung beginnt bei neuen Knoten mit 1, sollte aber mit 0 starten.
0043	Die Datei <i>BiDiB-Messages.xml</i> ist nicht auf dem aktuellen Stand.
0044	Die gemeldete Geschwindigkeit wird im Fahrpult nicht angezeigt
0045	POM Rückgabewert bei ByteRead wird nicht angezeigt.
0046	POM bei der Auswertung des ByteRead kann zum Absturz führen.
0047	Absturz, wenn sich die Anzahl der Melder an einem Knoten ändert und der Monitor dann neu Verbindung aufnimmt.

V 0.3.0.0

Bug	Beschreibung
0048	Boosterstatus 06 (Off, because missing DCC) wird nicht ausgewertet.
0049	Der Monitor hängt, wenn ein Knoten während des Firmware-Updates nicht mehr Antwortet.
0050	Eine „(,“ im beschreibenden Decoder-Text führt zu einem Absturz. (Danke Michael).

V 0.3.0.3

Bug	Beschreibung
0051	Listen der Messages, Features und Produkte werden nicht angezeigt.

V 0.3.0.5

Bug	Beschreibung
0052	Bei <code>MSG_NODE_LOST</code> werden die Info's des gelöschten Knotens weiter angezeigt. (Danke Andreas)
0053	Einschalten des Simulationsmodus, setzt Schnittstelle auf COM1 zurück.

V 0.4.0.4

Bug	Beschreibung
0054	Umschaltung auf „Technische Ansicht“ der Feature wird nicht gespeichert.

V 0.4.2.3

Bug	Beschreibung
0055	Alle CV-Typen, außer „Byte“, können trotz mode=ro verändert werden.
0056	FW-Update behält das .hex-File im Zugriff
0057	„BiDiB-Decoder.xml“ wird beim Verlassen des Monitors nicht sauber geschrieben, wenn Datei noch nicht existiert.
0058	Bei leerer Nodetabelle des Interfaces wird ein Timer gestartet. Nach Ablauf wird das Lesen der Nodetabelle wiederholt. Bis die Tabelle nicht mehr leer ist. Dieser Timer verursacht einen Absturz.
0059	DCCManufacturer.xml wurde beim auflisten nicht gefunden.
0060	Wenn <stabs>0</stabs> und <smax>0</smax> in „BiDiBProductID.xml“ auf 0 stehen, der Knoten aber trotzdem Eingänge meldet, kommt es zum Absturz.
0061	Wenn <mtabs>0</mtabs> und <mmax>0</mmax> in „BiDiBProductID.xml“ auf 0 stehen, der Knoten aber trotzdem Belegtmelder meldet, kommt es zum Absturz.
0062	Im Logfile-Analyser wurden die Zeilen „Input : “ und „IN --- : “ beide als Eingangsdaten vom Interface ausgewertet. Also doppelt.

V 0.4.3.1

Bug	Beschreibung
0063	Knoten hinter HUB wird nicht aus der Nodetab gelesen
0064	MSG_NODE_NEW und MSG_NODE_LOST funktionieren nicht bei Knoten hinter HUB
0065	Absturz, bei ausgehenden Messages von mehr als 64 Byte Länge.

V 0.4.3.6

Bug	Beschreibung
0066	Der Bytecode für die Messages MSG_LC_MACRO_GET und MSG_LC_MACRO_PARA_GET ist identisch (0x4a)
0067	Als Folge von Bug 0060 wurde die Melder-Liste nicht richtig initialisiert. (Danke Hr. Tams)
0068	Switch-Eingänge aller Knoten werden falsch ausgewertet und der Status nicht angezeigt (Danke Christoph).

V 0.4.4.0

Bug	Beschreibung
0069	Einem neuen Knoten wurde kein MSG_SYS_GET_MAGIC geschickt.
0070	Eine wechselnde Anzahl von Switch-Eingängen wurde nicht korrekt verarbeitet.
0071	Der „Simple“ Bootloader mit Magic 0xb0 0x0D wurde nicht erkannt (Danke Hr. Tams).
0072	Produktbezeichnung wird bei Feature FEATURE_RELEVANT_PID_BITS <> 16 nicht angezeigt.
0073	Bei zu schneller Folge eingehender Messages, wird die Auswertroutine mehrfach aufgerufen. FiFo kommt „durcheinander“.
0074	Bei aktivierter Debug-Ausgabe, kommt es zu einem Fehler in der Messagesequenz bei Knoten die neu angemeldet werden. Nach Message 0 kommt sofort Message 2.

V 0.4.5.0

Bug	Beschreibung
0075	MSG_NODETAB_GETALL wurde bei einem Hub nicht per Timer nach 1500 ms wiederholt. Dadurch wird beim nachträglichen Einschalten die Knotentabelle nicht abgefragt.
0076	Fahrpult: Wenn Geschwindigkeit auf 0 geht, wurde die Richtung immer auf „Vorwärts“ gestellt. Damit unter Umständen das Licht umgeschaltet.

V 0.4.6.0

Bug	Beschreibung
0077	Richtungsbit bei der Decoder-Steuerung im Fahrpult vertauscht.