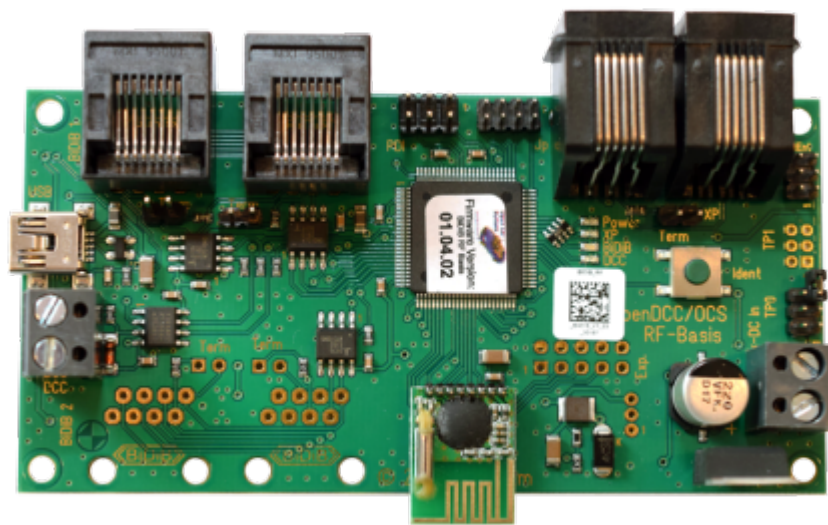


# Die BiDiB RF-Basis



Die RF-Basis ist eine Weiterentwicklung des DCC-RF-Booster. Im Unterschied zu diesem, unterstützt sie unterschiedliche digitale Signaleingangsquellen. Sie kann ihr Steuersignal aus dem BiDiB – Bus, vom DCC Gleisanschluss oder einer seriellen USB Verbindung beziehen. Zur lokalen Ansteuerung verfügt sie zusätzlich über eine Xpressnet Schnittstelle. Für den Betrieb der Basis mit Xpressnet Schnittstelle ist ein externes 14-18 V Netzteil notwendig. Die Basis ist als SMD vorbestückter Bausatz komplett programmiert im Fichtelbahn Shop erhältlich. Je nach genutzter Betriebsart müssen vom Anwender noch Anschlussbuchsen, Funkmodul und der THT Spannungsregler bestückt werden.

**Zum Betrieb der RF-Basis ist eine Cardecoder Firmware ab V0.3.0 nötig!**

**Die Baugruppe wurde abgekündigt!**



Die BiDiB RF-Basis V1.0 ist nicht mehr zum Kauf verfügbar! Die Baugruppe wurde durch die RF-Basis V2.0 abgelöst. **Support, Produktpflege und Firmwareupdates sind weiterhin verfügbar!**

**Technische Daten:**

Betriebsspannung mit Xpressnet Option:	14 bis 18 V / 1 A
Betriebsspannung ohne Xpressnet Option:	5 V / 100 mA
Steuersignal:	BiDiB digital oder
	DCC digital Gleissignal, intern galvanisch getrennt oder
	USB serielles Terminal
	bis zu 5 Lenz / Roco / Fleischmann Xpressnet Lokmäusen © / Handregler möglich
Xpressnet:	Host mit maximal 500 mA, Kurzschlussfest

Funk Ausgangssignal:	2,4 GHz moduliert mit RFM7x Modul übertragen
Funk Kanäle:	80 im 2,4 GHz Band
Funk Sendeleistung:	bis 3,2 mW in 4 Stufen wählbar
GBM16T Anschluss:	2 (1 direkt, 1 extern)

# 1. Betriebsarten der RF-Basis

## 1.1 BiDiB RF-Basis

Als BiDiB RF-Basis ist das Fahren, das Schalten der Funktionen F0 – 28 sowie die Programmierung der Decoder per POM Kommandos möglich (Lesen und Schreiben). Die Basis holt dabei von allen **einmal** angesteuerten Fahrzeugdecoder die Rückmeldungen ab und leitet diese zum Host weiter.



Im BiDiB Umfeld steht an der RF-Basis ein zusätzlicher interner TP0 Port zum Anschluss eines **GBM16T** Belegtmelder zur Verfügung.

**Hinweis:** die RF-Basis verfügt über einen zweiten GBM16T Port TP1. Für diesen ist in der Hardware Version 1 der Basis zu wenig Platz um eine normales Anschlusskabel stecken zu können. Soll dieser Port trotzdem genutzt werden, kann an dieser Stelle, für den TP1 eine um 90° abgewinkelte Stifteleiste eingesetzt werden.

## 1.2 DCC RF-Basis

Vom DCC Eingang können die Fahrzeugdecoder mit 28 und 128 Fahrstufen-Fahrbefehlen, per Funktionsbefehl für die Funktionen F0 – 28 und per POM Programmierbefehlen angesteuert werden. Decoderrückmeldungen werden nicht an die DCC Zentrale ausgegeben!



Die DCC Funktionalität der RF-Basis steht nur zur Verfügung, wenn **keine BiDiB** Verbindung besteht!

## 1.3 USB RF-Basis

Hierfür wird zwischen PC und Basis eine USB Verbindung zur seriellen Kommunikation (**115200 Baud 8N1**) hergestellt. Die Kommunikation selber erfolgt mit ASCII (Text) Kommandos zwischen PC und Basis. Dies ist auf allen gängigen Betriebssystemen mit einem gängigen ASCII Terminal möglich.



Eine Übersicht der möglichen Kommandos erhält man im Terminal mit H<Enter>.

Am USB Eingang ist das Fahren, das Schalten der Funktionen F0 – 28 sowie die Programmierung der der Decoder (Lesen und Schreiben der CV) möglich. Es werden alle Fahrzeugrückmeldungen im Klartext am USB Terminal ausgegeben.

---

## 1.4 Xpressnet RF-Basis

Die RF-Basis stellt dem Anwender ein Xpressnet Host zum Anschluss von Xpressnet Geräten zur Verfügung. Die mögliche Anzahl ist durch die 12V/500mA Stromversorgung des Buses auf ca. 5 Geräte begrenzt. Die Versorgungsspannung des Buses ist dabei kurzschlussfest ausgeführt. Mit am Bus angeschlossenen Handreglern können die Fahrzeuge gesteuert und per POM programmiert werden (Schreiben).



Xpressnet Zubehör- und Weichenbefehle sowie Service Mode Programmierung werden an der RF-Basis nicht unterstützt.

Eine Steuerung am Xpressnet ist zusätzlich zum BiDiB oder USB, sowie auch primär als „Stand-Alone“ Lösung möglich. Besteht eine DCC Verbindung, können nur Fahrzeuge vom Xpressnet gesteuert werden, die in der laufenden DCC Sitzung noch nicht per DCC gesteuert wurden!

---

# 2. Bestückungsoptionen

## 2.1 Versorgungsspannung

Für den Betrieb der Basis ist eine externe Stromversorgung notwendig. Hierfür gibt es zwei unterschiedliche Varianten, mit Xpressnet Option bzw. ohne.



Mit Xpressnet Option benötigt die Basis eine Betriebsspannung von 14-18 V / 500 mA DC an der Versorgungsklemme K9 (im Bild links), weiterhin muss der 12 V Spannungsstabilisator IC5 (im Bild rechts) bestückt werden.



Ohne Xpressnet kann die Basis an K9 oder an der USB Buchse K19 mit Betriebsspannung versorgt werden. Hierbei reicht eine externe Versorgungsspannung von 5V / 100 mA DC aus. An der USB Buchse K19 kann sowohl ein 5V Netzteil mit Mini-USB Stecker als auch eine aktive PC USB Verbindung die Spannungsversorgung übernehmen.



## 2.2 BiDiB Bauteile

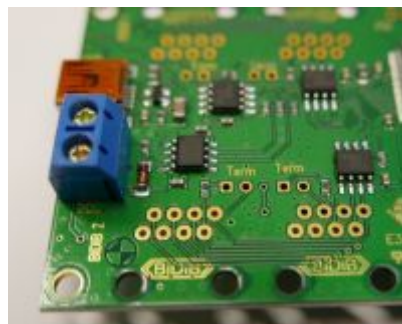


Zum BiDiB Betrieb müssen die BiDiB Buchsen K10 und K14 sowie die BiDiB Stiftleisten K1 und K13 zur Terminierung des Bus am letzten Teilnehmer bestückt werden. Die Basis nutzt als Node hier den BiDiB Port 1! Port 2 bleibt unbestückt und ist zukünftigen Erweiterungen vorbehalten.



## 2.3 DCC Bauteile

Für den DCC Betrieb der Basis muss nur die DCC Eingangsklemme K2 bestückt werden.



## 2.4 USB Bauteile

Für die Betriebsart USB RF – Basis brauchen keine extra Bauteile bestückt werden.

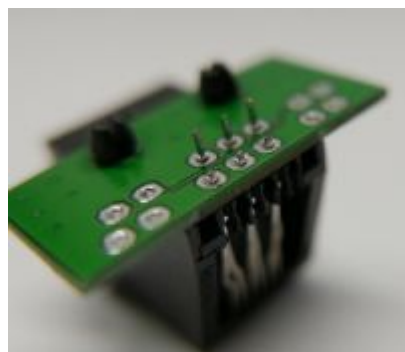


## 2.5 Xpressnet Bauteile



Zum Betrieb von Xpressnet Geräten an der Basis ist die 12V Stromversorgung für Xpressnet (siehe 4.1.) herzustellen. Die Hardware Version 1 der RF-Basis benötigt den XP Anschlusssatz (siehe Bild links). Dieser besteht aus einer RJ12 Xpressnet Buchse, einer zweipoligen Stiftleiste zur Terminierung, einer Adapterplatine und vier weiteren zweipoligen Stiftleisten zur Montage der Adapterplatine.

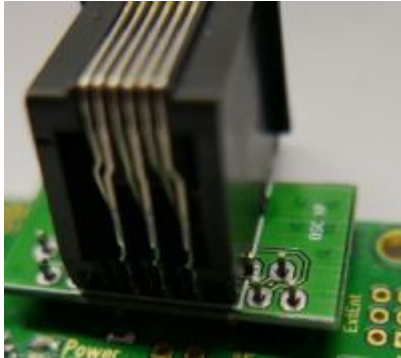
Zuerst wird die Xpressnet Buchse auf die Adapterplatine montiert. Dabei ist zu beachten, dass die Buchse von der weiß beschrifteten Seite der Adapterplatine montiert wird! (Bild rechts weiß „OSC XP“)



Vor dem Verlöten sollten die Anschlüsse der Xpressnet Buchse gekürzt werden. Im Bild links sind die äußeren schon gekürzt, die inneren noch nicht (man erkennt den Unterschied).



Jetzt werden die vier zweipoligen Stiftleisten mit dem langen Ende nach unten in die äußeren Positionen der Basis Platine gesteckt und von unten verlötet.



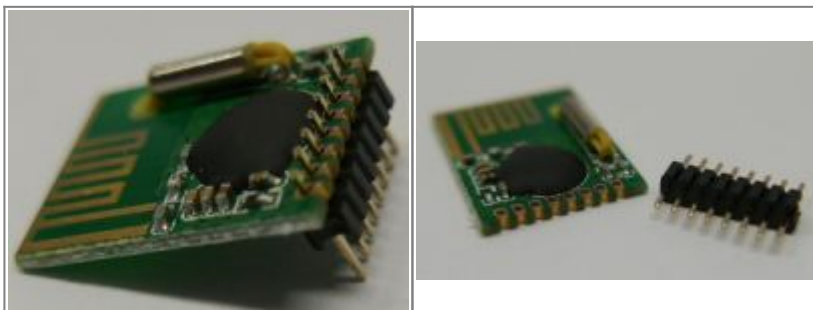
Danach die Adapterplatine mit der aufgelöteten Xpressnet Buchse auf die Stiftleisten aufsetzen. Dabei passen je die inneren beiden Stifte nur sehr eng am Buchsengehäuse vorbei. Diese beiden müssen vorsichtig mit wenig Lötzinn verlötet werden, ohne dabei das Plastik Gehäuse der Buchse zu beschädigen.

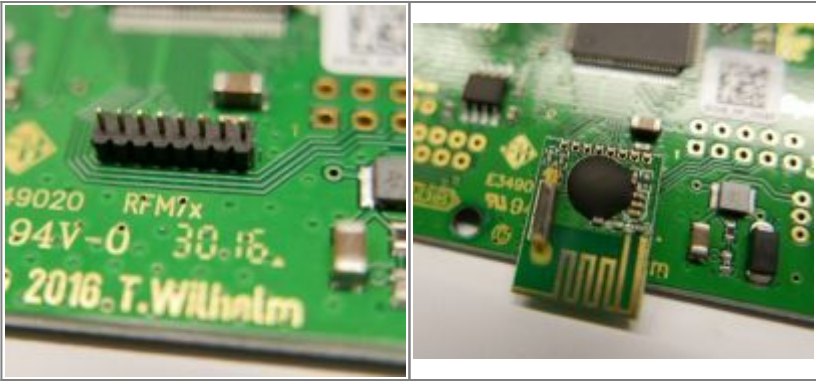
Zum Schluss wird noch die Stiftleiste für den XP Term Jumper eingelötet.



## 2.6 Funkmodul und Systemstiftleisten

Für die Funkübertragung wird ein RFM70, RFM73 oder RFM75 Funkmodul der Firma HopeRF verwendet. Ein Mischbetrieb unterschiedlicher Module (Fahrzeuge und Basis) ist möglich. Die RFM7x Module gibt es mit (RFM7xD) und ohne (RFM7x und RFM7xS) Stiftleiste. Für die RF-Basis können beide Varianten genutzt werden.





Die RF-Basis kann optional noch mit mehreren Stiftleisten bestückt werden. Darüber lassen sich externe Verbindungen zum Programmieren und Konfigurieren herstellen sowie GBM16T Belegtmelder anschließen. Im Bild rechts sind dies (von links nach rechts) PDI Programmierport, Jumperblock, externer Notastaster und Status LED Block und ganz rechts der TP0 GBM16T Anschluss. Auf der Platine gibt es noch eine weitere Position für einen GBM16T Anschluss (TP1). Hier kann bei Bedarf extern noch ein zweiter GBM16T angeschlossen werden (auf der Platine ist der Steckplatz für zwei GBM16T Stecker nebeneinander zu eng ).



### 3. Stückliste

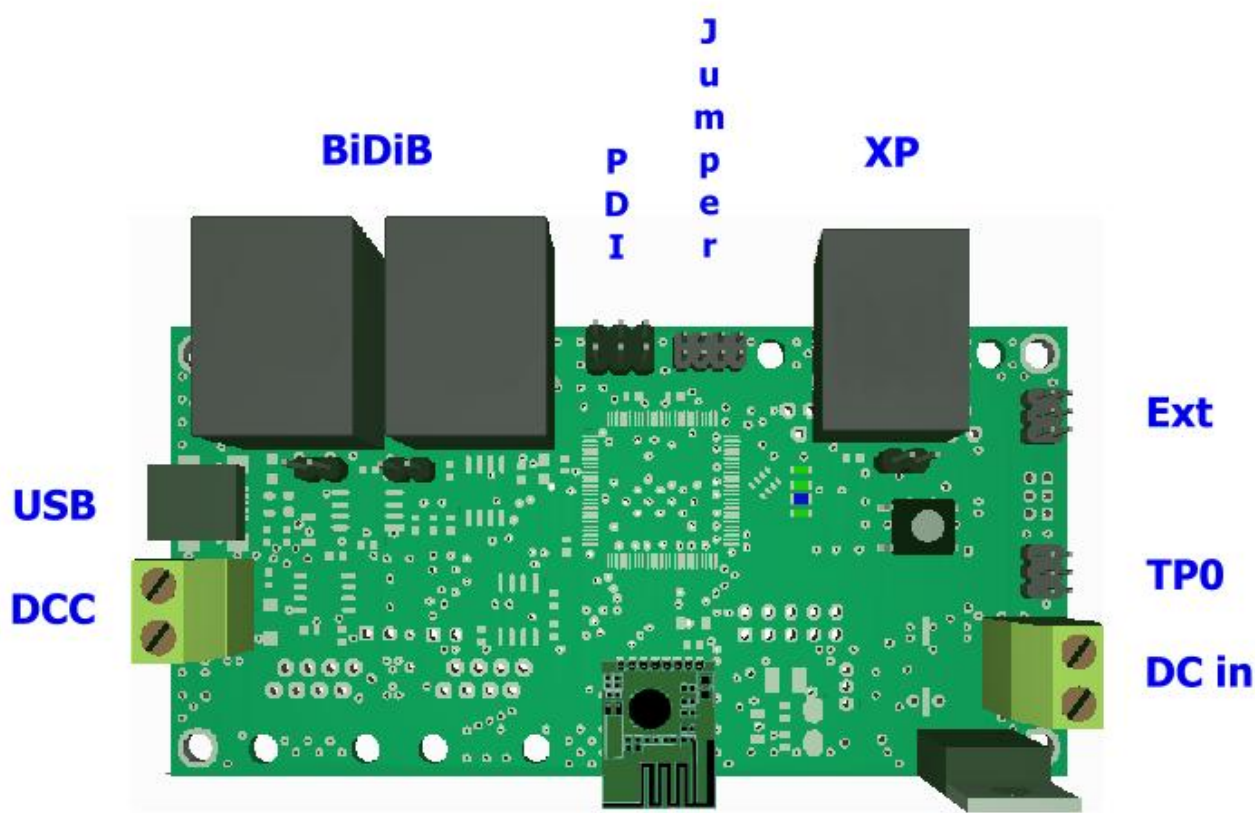
Die Stückliste zum Vervollständigen der RF-Basis Platine, ist nach Varianten sortiert. Es müssen nur die Varianten bestückt werden, die auch genutzt werden sollen.

Position	Bauteil	Bezeichnung	Beschreibung	Variante
1	K9	AKL 057-02	ANSCHLUSSKLEMME_R5,08_P2	alle
2	K12	MPE 150-3-006	Stiftleiste 2x03_2.54 PDI Programmierstecker	alle
3	K18	MPE 150-3-008	Stiftleiste 2x04_2.00 Konfigurationsjumper	alle
4	IC9	RFM75	RFM75 HopeRF 2,4 GHz Funkmodul	alle
5	IC9a	SL 1X08G 1,27	Stiftleiste 1x08_1.27 für RFM75 oder RFM75S	alle
6	IC9a	SL 1X10G 1,27	Alternativ zu Position 5 (muss gekürzt werden)	alle
7	K17	MPE 150-3-006	Stiftleiste 2x03_2.00 ext Notaus + Status LED	Optional alle
8	K10	MEBP 8-8S	Modulare Einbaubuchse RJ 45	BiDiB
9	K14	MEBP 8-8S	Modulare Einbaubuchse RJ 45	BiDiB
10	K1	MPE 087-1-002	Stiftleiste 1x02_2.54 BiDiB Bus Term JP	BiDiB
11	K13	MPE 087-1-002	Stiftleiste 1x02_2.54 BiDiB Bus Term JP	BiDiB
12	K16	MPE 150-3-006	Stiftleiste 2x03_2.00 TP0 GBM16T Anschluss	Optional BiDiB
13	K2	AKL 057-02	ANSCHLUSSKLEMME_R5,08_P2	DCC
14	IC5	LM 2940 CT12	12 V / 1 A Spannungsregler	XP
15	K3	MEBP 6-6S	Modulare Einbaubuchse RJ 12	XP
16	K3a	XP PCB	Adapterplatine für Xpressnet Buchse K3	XP

Position	Bauteil	Bezeichnung	Beschreibung	Variante
17	K3b	MPE 087-1-002	4x Stiftleiste 1x02_2.54 für K3a	XP
18	K5	MPE 087-1-002	Stiftleiste 1x02_2.54 XP Bus Term JP	XP

## 4. Anschlüsse und Funktionen

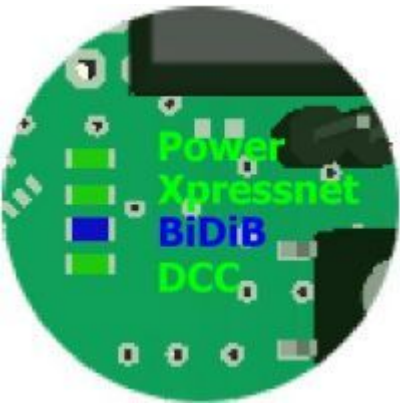
### 4.1 Übersicht



DCC	DCC Eingang, wenn verwendet mit DCC Boosterausgang (Gleis) verbinden
USB	Mini USB Anschluss für Terminal Verbindung oder 5V DC Versorgung
BidiB	BiDiB Anschlüsse, am letzten Busteilnehmer TERM Jumper stecken
PDI	Programmierschluss zur Erstprogrammierung des Mikrocontroller
Jumper	Funktions- und Boot Jumper Block
XP	Xpressnet Host Ausgang
Ext	Anschluss für externen Notastaster und Status LEDs
TP0	GBM16T Anschluss
DC in	DC Versorgungseingang



## 4.2 LED Anzeigen



Auf der RF-Basis befinden sich vier Status LEDs. Diese zeigen unterschiedliche Betriebszustände und Fehler direkt an.

LED	Aktion	Bedeutung
Power	gedimmt eingeschaltet	Betriebsspannung vorhanden
Power	blinkt schnell	Ident aktiv (Taster gedrückt oder vom Host aktiviert)
Power	blinkt	Notstopp aktiviert
Xpressnet	aus	normaler Status
Xpressnet	an	Bootloader aktiviert
Xpressnet	blinkt	Xpressnet Datenverkehr
BiDiB	an	mit dem BiDiB verbunden
DCC	flackert	DCC Signal liegt an
alle	blinken	kein EEPROM File vorhanden
Xpressnet + Power	blinken	keine Seriennummer vorhanden

## 4.3 BiDiB und TP0 Anschluss



oben).

Die RF-Basis wird am BiDiB wie jeder übliche Node angeschlossen. Ist die Basis der letzte Teilnehmer am Bus, müssen die beiden BiDiB TERM Jumper gesteckt werden. Die Konfiguration der Basis ist mit den bekannten BiDiB Tools in der je aktuellen Version möglich. Für den BiDiB Betrieb ist eine externe Stromversorgung der Basis nötig (siehe



Wird die Basis in einem BiDiB Umfeld betrieben, steht dem Anwender ein zusätzlicher interner TP0

Port zum Anschluss eines GBM16T Belegtmelder zur Verfügung.

---

## 4.4 DCC Anschluss



Soll eine DCC Zentrale als Steuereingang verwendet werden, wird von der DCC Klemme an der RF-Basis eine Verbindung zum DCC Gleis hergestellt.

**Beachte, in dieser Betriebsart teilen sich Lokomotiven und Autos einen gemeinsamen DCC Adressbereich!**

Die Konfiguration der RF-Basis kann am BiDiB mit den BiDiB Tools oder per USB Verbindung in einem PC Terminal durchgeführt werden.

Fällt das DCC Signal aus (Notaus, Kurzschluss am Gleis oder DCC Verbindung unterbrochen), werden alle Fahrzeuge per Funk angehalten. Damit wird also ein Notaus vom DCC auch an die Fahrzeuge weitergegeben. Dies zeigt eine blinkende Power LED an. Notaus/Notstopp kann an der Basis von einem Xpressnet Eingabegerät, dem externen Notastaster oder durch Wiedereinschalten des DCC Signal aufgehoben werden.

Die DCC Funktionalität der RF-Basis wurde als Ersatz für den Selbstbau DCC-RF-Booster implementiert. Hiermit wird dem Anwender eine SMD vorbestückte Hardware zur Verfügung gestellt, so dass nicht mehr zwingend eine SMD Basisplatine zum Betrieb des Carsystem selbst bestückt werden muss. Am DCC werden keine Fahrzeugrückmeldungen ausgegeben. Diese kann man aber am seriellen USB Terminal mitlesen bzw. nutzen.

Für den DCC Betrieb ist eine externe Stromversorgung der Basis nötig.

---

## 4.5 Xpressnet Anschluss



Die RF-Basis verfügt über einen Xpressnet Host Anschluss. Dieser ist für die Versorgung von Xpressnet Geräten mit insgesamt 500 mA Stromaufnahme ausgelegt. Dies reicht im Normalfall für 5 Handregler aus, die an der Xpressnet Buchse über handelsübliche Xpressnet Verteiler angeschlossen werden können. Wird der Bus auf der Anlage verlängert, eventuell mit externen Anschlussdosen, muss der Bus an der letzten Anschlussdose terminiert werden.

Am Xpressnet kann die Basis als Baugruppe für den autonomen Betrieb mit einem Handregler betrieben werden. In dieser Konfiguration erhält der Anwender eine autonome Lösung ohne PC

Anbindung und kann seine Fahrzeuge alle separat von Hand steuern und per POM programmieren. Hierbei muss allerdings beachtet werden, dass die Firmware einiger Xpressnet Handregler das Ändern der CV 1, 17 und 18 per POM unterbindet. Zum ändern der Fahrzeugadresse muss also eine andere Programmierquelle genutzt werden (z.B. Terminal am PC).

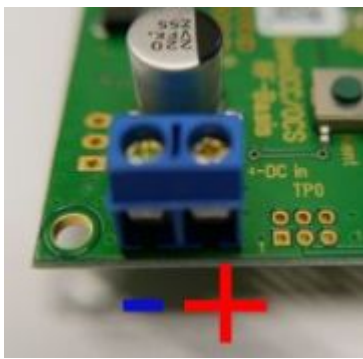
Die Basis unterstützt am Xpressnet das Schalten aller 28 Funktionen, DCC Fahrbefehle mit 28 oder 128 Fahrstufen und alle POM Programmierbefehle (Schreiben).

**Beachte, 14 Fahrstufen werden nicht unterstützt. Verwendet man 14 Fahrstufen, leuchtet beim Fahrstufenwechsel zyklisch die Fahrzeugbeleuchtung auf.**

Für den Xpressnet Betrieb ist eine externe Stromversorgung der Basis nötig.

---

## 4.6 DC Anschluss



Dies ist der primäre Anschluss für die Versorgungsspannung der RF-Basis. Die Basis benötigt hier eine Eingangsgleichspannung von 14-18 V mit maximal 1 A. Die Anschlusspolarität ist im Bild rechts gekennzeichnet. Die Basis hat einen Verpolschutz, eine verpolte Versorgungsspannung führt also nicht zur Beschädigung der Basis, sie funktioniert so lediglich nicht.

Die Angabe zum Netzteil ist abhängig von den verwendeten Funktionen der Basis. Wird kein Xpressnet benötigt, reicht eine 5V/100 mA Versorgungsspannung zum Betrieb der Basis aus.

---

## 4.7 EXT Anschluss



Der „ExtEnt“ Anschluss der RF-Basis kann für eine externe Statusanzeige der RF-Basis und für einen Notastaster genutzt werden. Alle nötigen Vorwiderstände sind auf der Basisplatine schon vorhanden, so dass LEDs und Taster direkt ohne weitere Bauteile angeschlossen werden können.

Der Taster schaltet bei jeder Betätigung zwischen Notstopp und Betrieb um. Die grüne LED leuchtet im normalen Betrieb, die rote LED beim Notstopp. Die Notstopp Funktion und damit diese LEDs lassen sich auch vom BiDiB, DCC, Xpressnet und USB Terminal schalten. Beachte, es gibt Kriterien, die ein Wiedereinschalten von einem anderen Terminal als dem auslösendem verhindern (z.B. bei einer Kabelunterbrechung am BiDiB).

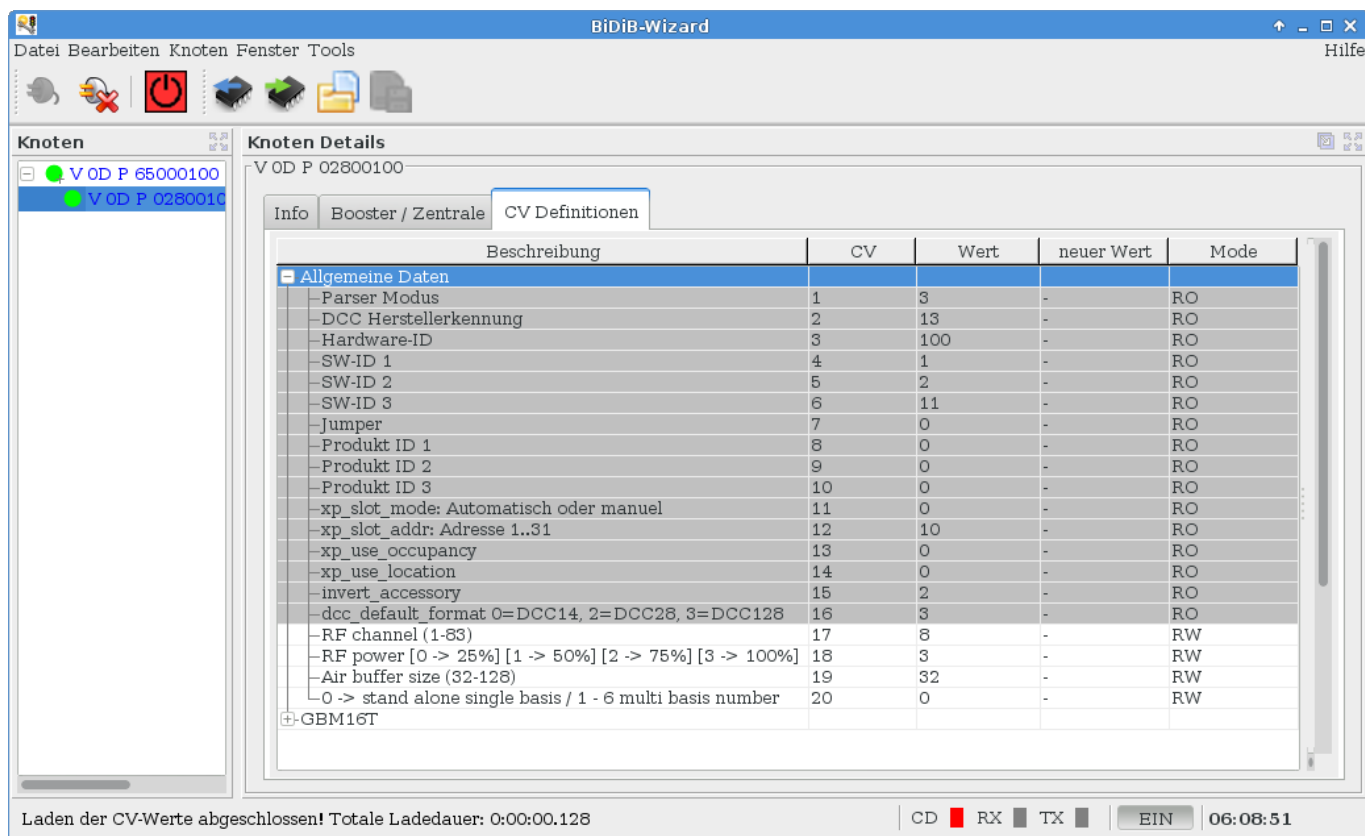
Im [Fichtelbahn Shop](#) steht ein Bausatz mit Notastaster und LEDs zum Anschluss an dieser Stelle zur Verfügung.

## 5. Die RF-Basis als BiDiB Basis

Die RF-Basis wird als klassischer Node am BiDiB angeschlossen. Dabei sind die Regeln zur Bus-Terminierung zu beachten. Die Basis meldet sich am Bus als DCC Generator an und kann so wie ein normaler DCC Generator für die digitale Gleisansteuerung genutzt werden.

Die BiDiB Tools **BiDiB-Wizard** und **BiDiB-Monitor** unterstützen dabei die speziellen Fähigkeiten der RF-Basis als Carsystem Basis.

### 5.1 RF-Basis CV Konfiguration am BiDiB



The screenshot shows the BiDiB-Wizard software interface. The main window is titled "BiDiB-Wizard" and has a menu bar with "Datei", "Bearbeiten", "Knoten", "Fenster", and "Tools". Below the menu bar is a toolbar with icons for file operations and node management. The left pane shows a list of nodes, with "V OD P 65000100" and "V OD P 02800100" visible. The right pane shows the "Knoten Details" for the selected node "V OD P 02800100". The "CV Definitionen" tab is active, displaying a table of CVs (Configuration Variables) for the node. The table has columns for "Beschreibung", "CV", "Wert", "neuer Wert", and "Mode". The CVs are listed in a tree structure under "Allgemeine Daten".

Beschreibung	CV	Wert	neuer Wert	Mode
<b>Allgemeine Daten</b>				
- Parser Modus	1	3	-	RO
- DCC Herstellerkennung	2	13	-	RO
- Hardware-ID	3	100	-	RO
- SW-ID 1	4	1	-	RO
- SW-ID 2	5	2	-	RO
- SW-ID 3	6	11	-	RO
- Jumper	7	0	-	RO
- Produkt ID 1	8	0	-	RO
- Produkt ID 2	9	0	-	RO
- Produkt ID 3	10	0	-	RO
- xp_slot_mode: Automatisch oder manuel	11	0	-	RO
- xp_slot_addr: Adresse 1..31	12	10	-	RO
- xp_use_occupancy	13	0	-	RO
- xp_use_location	14	0	-	RO
- invert_accessory	15	2	-	RO
- dcc_default_format 0=DCC14, 2=DCC28, 3=DCC128	16	3	-	RO
- RF channel (1-83)	17	8	-	RW
- RF power [0 -> 25%] [1 -> 50%] [2 -> 75%] [3 -> 100%]	18	3	-	RW
- Air buffer size (32-128)	19	32	-	RW
- 0 -> stand alone single basis / 1 - 6 multi basis number	20	0	-	RW
GBM16T				

At the bottom of the window, there is a status bar with the text "Laden der CV-Werte abgeschlossen! Totale Ladedauer: 0:00:00.128" and a set of buttons for "CD", "RX", "TX", "EIN", and a timer showing "06:08:51".

Neben den BiDiB Features gibt es in der RF-Basis auch noch einige RF-Basis spezifische Konfigurationsvariablen (CV) und CVs von optional angeschlossenen Baugruppen wie dem GBM16T.

Im Bild grau hinterlegt der nur lesbare Bereich der Info CV. Darunter hier noch einmal größer:

-dcc_default_format 0=DCC14, 2=DCC28, 3=DCC128	16	3
-RF channel (1-83)	17	8
-RF power [0 -> 25%] [1 -> 50%] [2 -> 75%] [3 -> 100%]	18	3
-Air buffer size (32-128)	19	32
-0 -> stand alone single basis / 1 - 6 multi basis number	20	0

die vier CV zum konfigurieren der RF-Basis Funktionen.

**„RF channel“ CV17 (default 8)**, hier kann ein Funkkanal im 2,4GHz Funkband zum Betrieb der Basis eingestellt werden. Dieser muss bei Basis und allen Fahrzeugen identisch gewählt werden. Es ist aber möglich, verschiedene separate Carsysteme nebeneinander auf unterschiedlichen Funkkanälen gleichzeitig zu betreiben. Allerdings sollte hier ein Abstand von mehreren Kanälen gewählt werden, um eine Gegenseitige Beeinflussung der Systeme zu verhindern. Das OpenCarSystem verwendet default den Funkkanal 8 für alle Baugruppen, dies muss der Anwender mit einem Carsystem also auch nicht ändern.

**„RF power“ CV18 (default 3)**, bietet die Möglichkeit die Sendeleistung der RF-Basis in vier Stufen an die Anlagengröße anzupassen. Hier muss der Anwender mit einer Basis und einer Funkzelle normalerweise keine Änderung machen. Bei sehr kleinen Anlagen (z.B. in Maßstab N) könnte man aber zum Energie sparen in den Fahrzeugen die Sendeleistung sowohl in der Basis wie auch in den Fahrzeugen verringern. Eine weitere Anwendung hierfür ist ein künftiger Mehrzellen Betrieb (siehe CV20).

**„Air buffer size“ CV19 (default 32)**, gibt die maximal gleichzeitig von der Basis selbst per Funk aktualisierten Fahrzeuge an. Hier sind Werte zwischen 32 und 128 möglich. Ein Wert von 128 bedeutet nicht, dass man „nur“ maximal 128 Fahrzeuge gleichzeitig betreiben kann, sondern dass die Basis die Fahr- und Funktionsdaten für die letzten 128 Fahrzeuge im internen Speicher vorhält. Wenn man mehr Fahrzeuge betreiben will, obliegt der Host Software die Aufgabe, die Fahrzeugdaten entsprechend häufig über den Bus zu aktualisieren. Dieser Wert sollte aus Gründen der Performance nicht erheblich größer als die maximal benutzte Anzahl von Fahrzeugen sein.

*Beachte: ist der Wert kleiner als die benutzte Anzahl von Fahrzeugen, fahren nach einem Notstop nicht alle Fahrzeuge sofort wieder mit ihrer alten Geschwindigkeit weiter, sondern nur die, welche im internen Basis Speicher vorhanden sind, alle anderen erst dann, wenn sie von der Host Software neu angesprochen wurden!*

**„Basis Nummer“ CV20 (default 0)**, es wird die Möglichkeit geben, einen Mehrzellen Funkbetrieb für größere Anlagen aufzubauen. Dabei sind bis zu 6 unterschiedliche Funkzellen nebeneinander möglich. Bei einem Mehrzellen Betrieb sollte die Sendeleistung jeder Funkzelle nur so groß gewählt werden, dass genau die Fläche dieser Zelle abgedeckt ist. Eine großflächige Überschneidung der Funkzellen würde die Systemperformance deutlich senken. CV20 gibt die Nummer der Funkzelle an, es dürfen sich nicht zwei Zellen mit der gleichen Nummer überlappen. Ist CV20 = 0 arbeitet das System im Einzel Basis Modus mit nur einer Funkzelle. Dies sollte für die meisten Anlagen im Heim- und Hobbybereich auch völlig ausreichen, die maximale Reichweite einer Funkzelle beträgt ca. 6m um die RF-Basis herum. Dies ist natürlich vom Anlagenaufbau und den Ausbreitungsmöglichkeiten der Funkwellen auf der Anlage abhängig (z.B. Kückendrahtgestell als Gebirge bedämpft vorzüglich die Funkwellen).

Anschließend folgt noch der CV-Bereich von maximal zwei optional angeschlossenen GBM16T Belegtmeldern. Deren Konfiguration kann im Handbuch zum GBM16T bzw. dem BiDiB Wiki nachgeschlagen werden. Ist ein GBM16T angeschlossen, gibt es im Anschluss an die „CV Information“ im BiDiB Wizard noch einen weiteren „Reiter“ mit der Belegtanzeige vom GBM16T.

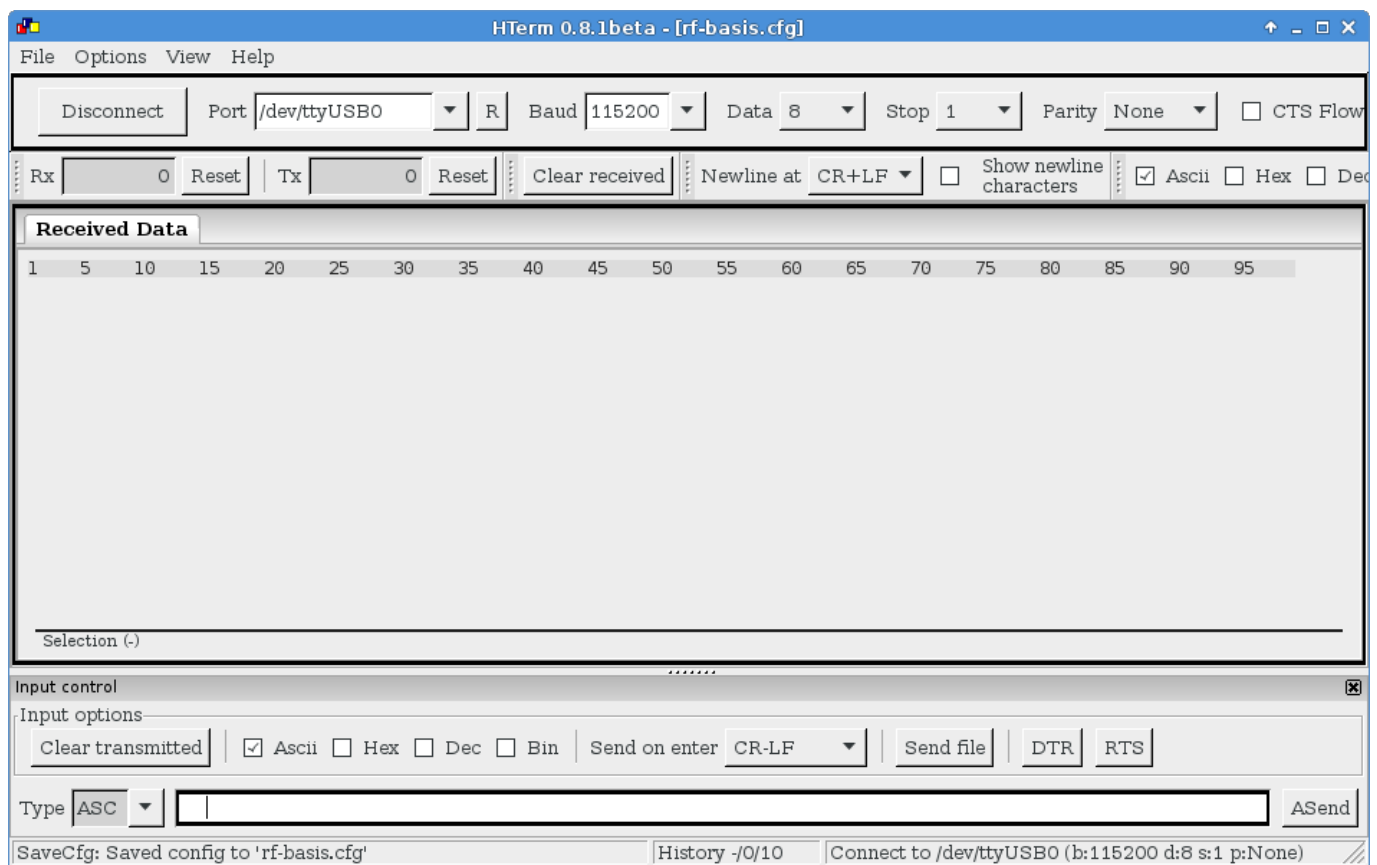


## 6. Die RF-Basis als USB Basis

Die RF-Basis verfügt über einen Mini USB Anschluss. An diesem kann sie mit seriellen Kommandos gesteuert werden. Dies kann im einfachsten Fall eine Terminal Software (Hyperterminal, HTerm, CuteCom u.ä.) sein. Der Anwender kann hier auch eigene Anwendungen programmieren bzw. nutzen, die über die serielle Schnittstelle mit der Basis kommunizieren.

Im folgenden wird die Kommunikation am Beispiel von **HTerm** gezeigt. Dies ist ein beliebtes Terminal und steht sowohl unter MS Windows wie auch unter Linux zur Verfügung. Unter Mac OS kann man hierfür **CoolTerm** verwenden. Achtung, unter IOS gibt es eine SSH Client APP die auch Hterm heißt und hierfür nicht geeignet ist!

Weiterhin kann man das „debug“ Interface vom **BiDiB Wizard** (in allen drei OS) als Terminal zum Betrieb am USB Port nutzen.



HTerm muss nicht installiert werden, es reicht die Programmdatei zu starten. Das Programmfenster kann man an seine eigenen Anforderungen anpassen und so übersichtlich gestalten. Im Bild die Ansicht vom Autor hier sind schon die nicht benötigten Fenster ausgeblendet und die Menüeinträge im oberen Bereich auf die für uns interessanten Werte sortiert.

Um die Basis nutzen zu können, sind folgende Einstellungen nötig bzw. von Vorteil:

- der Port, an dem die Basis angeschlossen ist (Windows Nutzer haben hier „COM\_x“ Ports zur

Auswahl), mit dem „R“ rechts daneben kann man die Portauswahl aktualisieren, Linux Nutzer müssen eventuell den „ttyUSB\_x“ von Hand eintragen (kann man einfach überschreiben).

- Daneben die Baudrate, die Basis arbeitet am USB mit 115200 Baud
- dann die Übertragungsparameter (8N1): Data 8, Stop 1, Parity: None

zur Erhöhung des Anzeigekomfort:

- „Newline at:“ CR+LF
- „Show newline characters“ : kein Haken
- „ASCII“: Haken setzen

im unteren Bereich unter „Input options“:

- „ASCII“: wieder Haken setzen
- „Send on enter“: CR-LF

Diese Grundeinstellungen für den Betrieb der RF-Basis mit HTerm kann man sich unter „File“ „Konfiguration sichern“ abspeichern. Man erzeugt sich damit eine Datei „my\_config.cfg“. Wenn man die Dateiendung „.cfg“ HTerm zuordnet, kann man HTerm durch doppelklicken auf seine config-Datei (oder eine Verknüpfung darauf) zukünftig gleich automatisch mit den passenden Einstellungen starten.

**Hier** ist eine fertige HTerm cfg für dieses Projekt erhältlich, in der man nur noch seinen Port einstellen muss.

Im oberen Bereich bekommt man so die Antworten der RF-Basis zu sehen, in der unteren Eingabezeile werden die Kommandos zur RF-Basis eingegeben. Mit <Enter> werden diese dann automatisch zur Basis übertragen.

*Hinweis für Nutzer anderer Terminal Software: die Basis akzeptiert auch nur <CR> als Zeilenende beim senden. Hinweis für BiDiB Nutzer: diese Einstellungen sind identisch mit den Einstellungen für die BiDiB debug Ausgaben.*

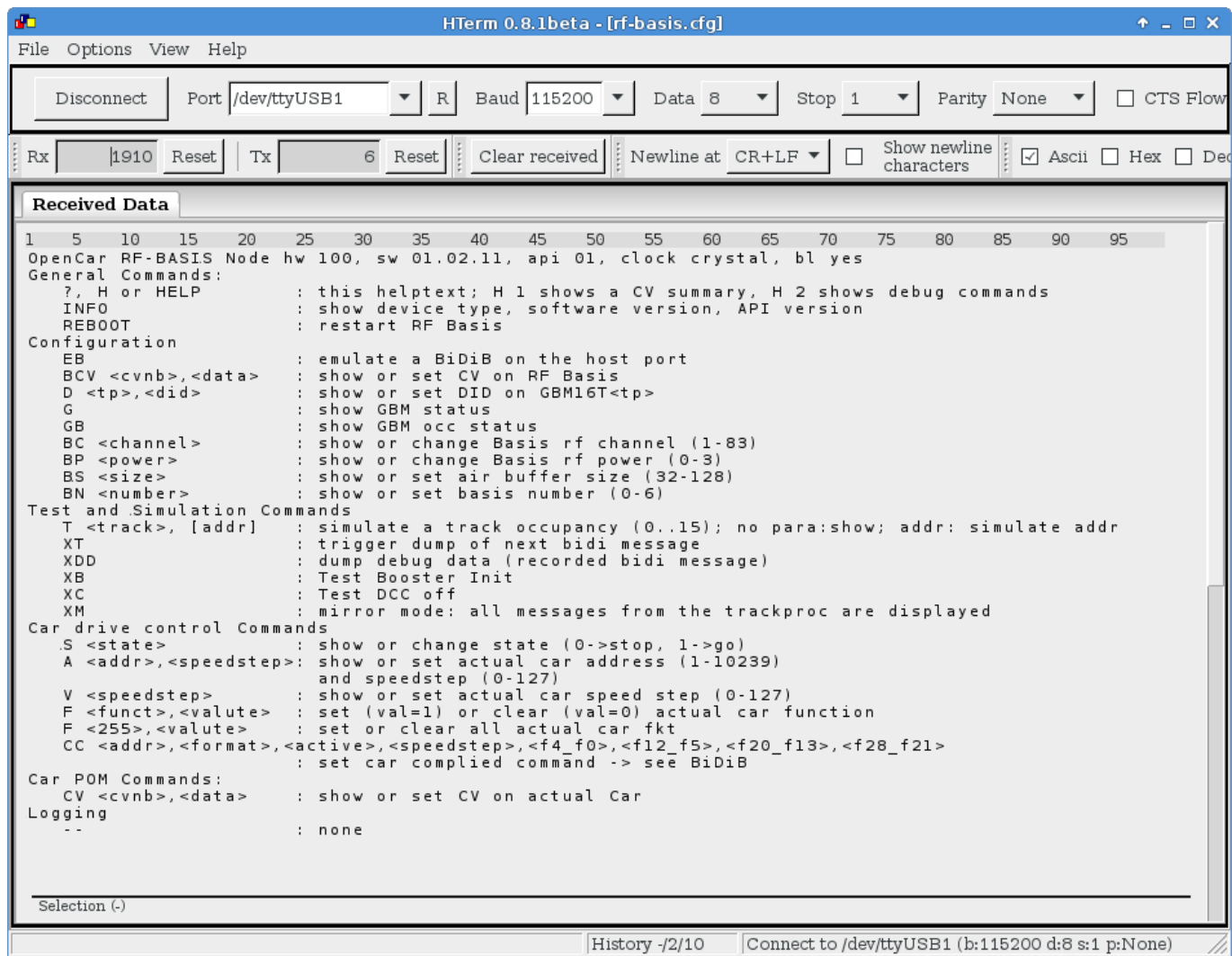
---

## 6.1 Allgemeine USB Befehle

Zum Test auf korrekte Verbindung zur Basis, bzw. zur Ausgabe des Funktionsumfang der RF-Basis am USB, kann man die „Hilfe“ Anzeige nutzen. Dazu in der unteren Eingabezeile:

<b>? &lt;Enter&gt;</b>
oder: <b>h &lt;Enter&gt;</b>
oder: <b>H &lt;Enter&gt;</b>
oder: <b>Help &lt;Enter&gt;</b>

eingeben. Mit „<Enter>“ ist hier das einmalige drücken der Entertaste gemeint. Man erkennt, dass die Basis sowohl mit Groß- wie auch mit Kleinbuchstaben angesprochen werden kann. Die Ausgabe sieht dann folgendermaßen aus:



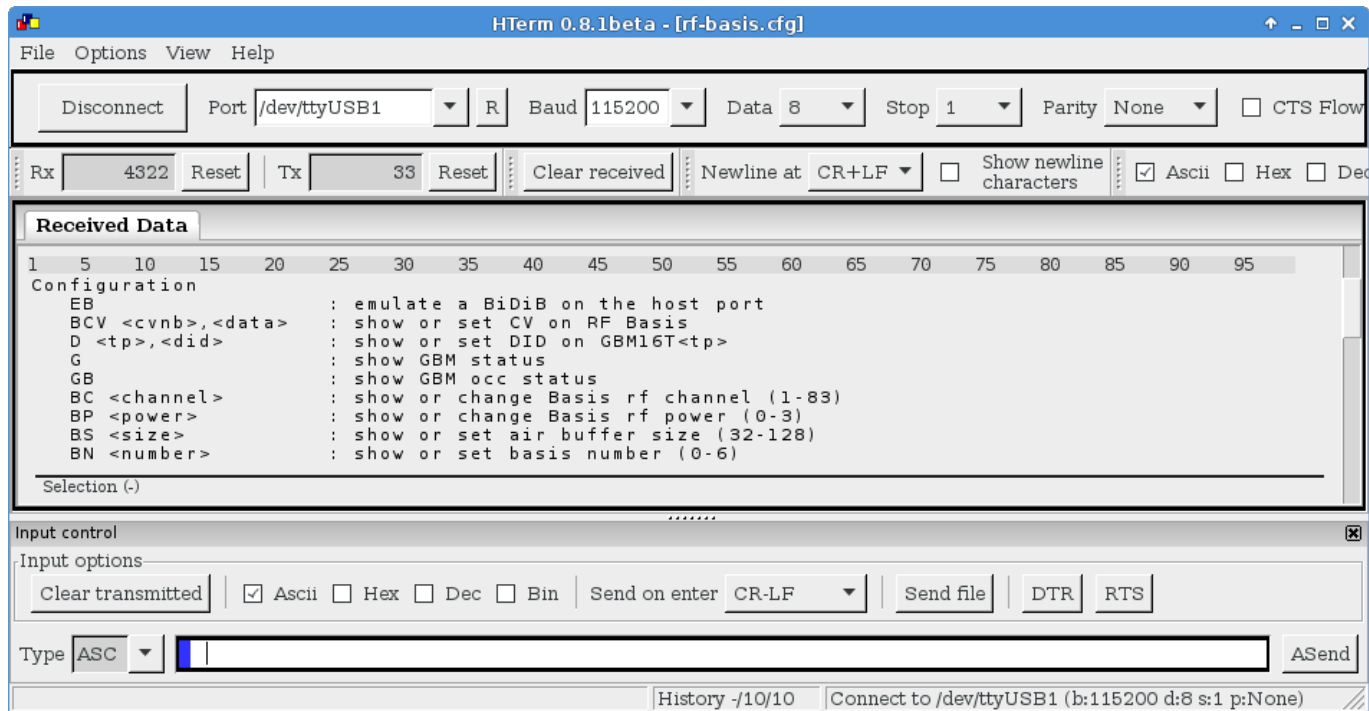
The screenshot shows the HTerm 0.8.1beta terminal window with the title bar "HTerm 0.8.1beta - [rf-basis.cfg]". The window has a menu bar (File, Options, View, Help) and a toolbar with various settings like Port (/dev/ttyUSB1), Baud (115200), Data (8), Stop (1), Parity (None), and CTS Flow. Below the toolbar is a status bar showing Rx (1910) and Tx (6) counts, a "Reset" button, and a "Clear received" button. The main area displays the "Received Data" from the RF-Basis device, which includes a list of commands and their descriptions, such as "General Commands", "Configuration", "Test and Simulation Commands", "Car drive control Commands", and "Car POM Commands". The status bar at the bottom shows "History -2/10" and "Connect to /dev/ttyUSB1 (b:115200 d:8 s:1 p:None)".

```
OpenCar RF-BASIS Node hw 100, sw 01.02.11, api 01, clock crystal, bl yes
General Commands:
? , H or HELP      : this helptext; H 1 shows a CV summary, H 2 shows debug commands
INFO              : show device type, software version, API version
REBOOT            : restart RF Basis
Configuration
EB                : emulate a BiDiB on the host port
BCV <cvnb>,<data>  : show or set CV on RF Basis
D <tp>,<did>       : show or set DID on GBM16T<tp>
G                : show GBM status
GB               : show GBM occ status
BC <channel>      : show or change Basis rf channel (1-83)
BP <power>        : show or change Basis rf power (0-3)
BS <size>         : show or set air buffer size (32-128)
BN <number>       : show or set basis number (0-6)
Test and Simulation Commands
T <track>,<addr>   : simulate a track occupancy (0..15); no para:show; addr: simulate addr
XT               : trigger dump of next bidi message
XDD              : dump debug data (recorded bidi message)
XB               : Test Booster Init
XC               : Test DCC off
XM               : mirror mode: all messages from the trackproc are displayed
Car drive control Commands
.S <state>        : show or change state (0->stop, 1->go)
A <addr>,<speedstep> : show or set actual car address (1-10239)
                  : and speedstep (0-127)
V <speedstep>     : show or set actual car speed step (0-127)
F <funct>,<valute> : set (val=1) or clear (val=0) actual car function
F <255>,<valute>   : set or clear all actual car fkt
CC <addr>,<format>,<active>,<speedstep>,<f4_f0>,<f12_f5>,<f20_f13>,<f28_f21>
                  : set car complied command -> see BiDiB
Car POM Commands:
CV <cvnb>,<data>   : show or set CV on actual Car
Logging
--               : none
```

In der ersten Zeile erfolgt die Infoausgabe, danach in einzelne Rubriken unterteilt die verschiedenen Kommando Möglichkeiten.

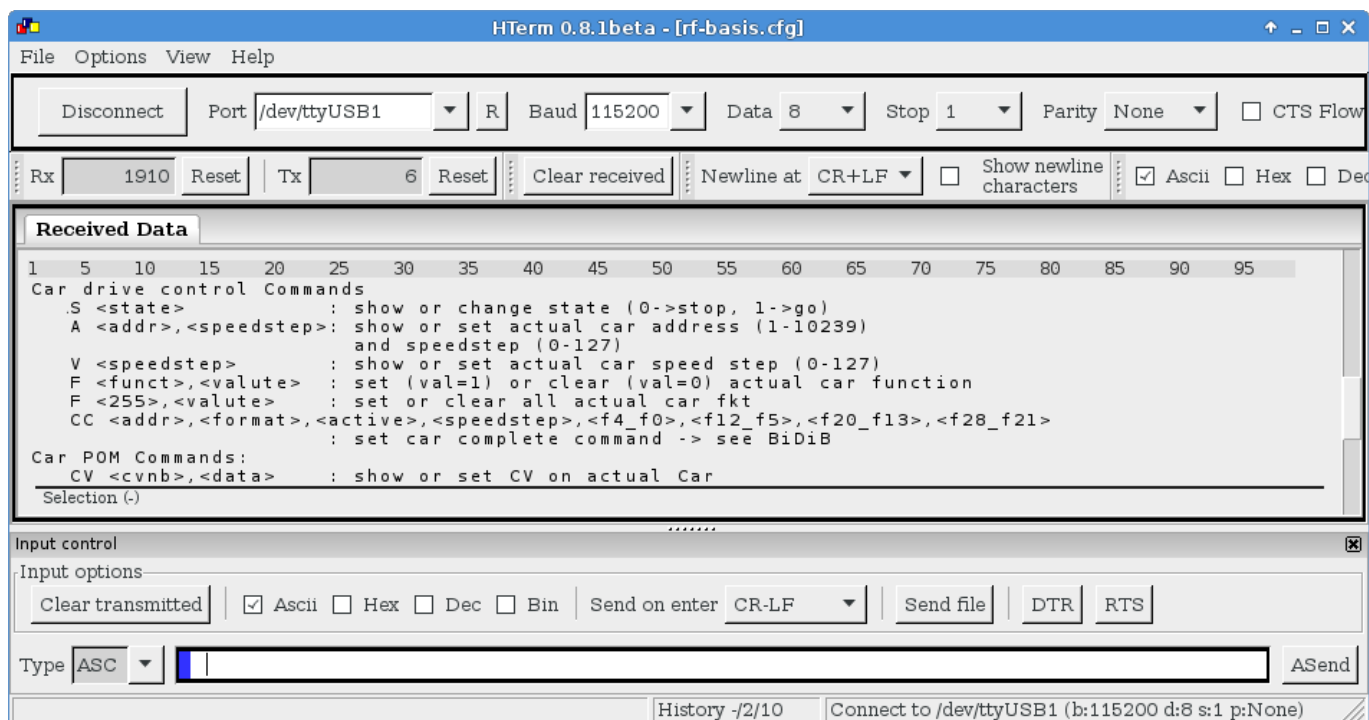
## 6.2 Basis Konfiguration am USB

Identisch zum BiDiB besteht am USB Bus auch wieder die Möglichkeit, die RF-Basis mit speziellen Einstellungen zu konfigurieren.



Im Bereich „Configuration“ kann die globale Konfiguration der Baugruppe RF-Basis angepasst werden. Hier sind nicht nur Einstellungen zum RF Betrieb, sondern auch einige BiDiB-spezifische zu finden. Diese resultieren aus dem auch über USB funktionierendem „Debug Mode“ der RF-Basis. Für die RF-Basis relevant sind die letzten vier Kommandos. Diese sind identisch mit der vergleichbaren Einstellung am BiDiB System und können genauer [hier](#) nachgeschlagen werden.

## 6.3 Carsteuerung am USB



Diese Kommandos sind zum Steuern der Fahrzeuge am USB Anschluss vorgesehen. Am USB

Anschluss steht der komplette Funktionsumfang der RF-Basis zur Verfügung. Im Folgenden werden hier die einzelnen Befehle erläutert.

Der „state“ (Status) Befehl: **S<enter>** in der Eingabezeile gibt den aktuellen Status zurück. Ein **S0<enter>** stoppt alle Fahrzeuge (Notstop), ein **S1<enter>** schaltet wieder in den Normalbetrieb zurück.

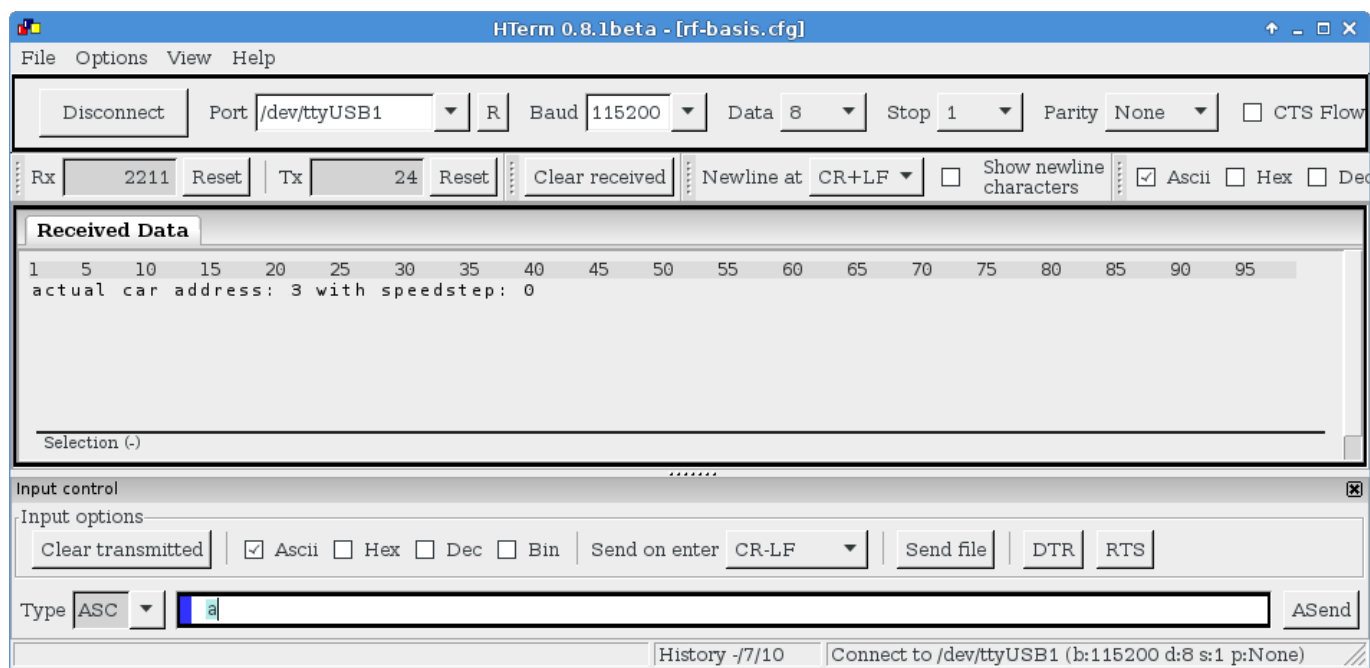
Das „S“ Kommando ist also identisch mit dem Notastaster, dem Notstopp am Handregler, dem Stopp am BiDiB oder der DCC Gleisabschaltung am DCC Gleis. Ein Stopp Befehl vom USB Anschluss kann von allen zuvor genannten Quellen auch wieder auf „aktiv“ (System „Go“ / Normalbetrieb) zurück geschaltet werden.

Um ein Fahrzeug steuern zu können, muss der Basis dessen Adresse bekannt gegeben werden. Die Carsteuerung arbeitet grundsätzlich mit dem von DCC her bekannten langen Adressraum. Es lassen sich also alle Fahrzeuge zwischen 1 und 10239 adressieren. Es wird nicht zwischen kurzer oder langer Adressierung unterschieden.

*ACHTUNG, aus Kompatibilitätsgründen unterscheiden die Fahrzeuge (Cardecoder) schon kurze und lange Adressierung! Die Fahrzeuge verwenden also immer die Adresse (lang oder kurz), die in CV29 im Fahrzeug auf aktiv gesetzt ist.*

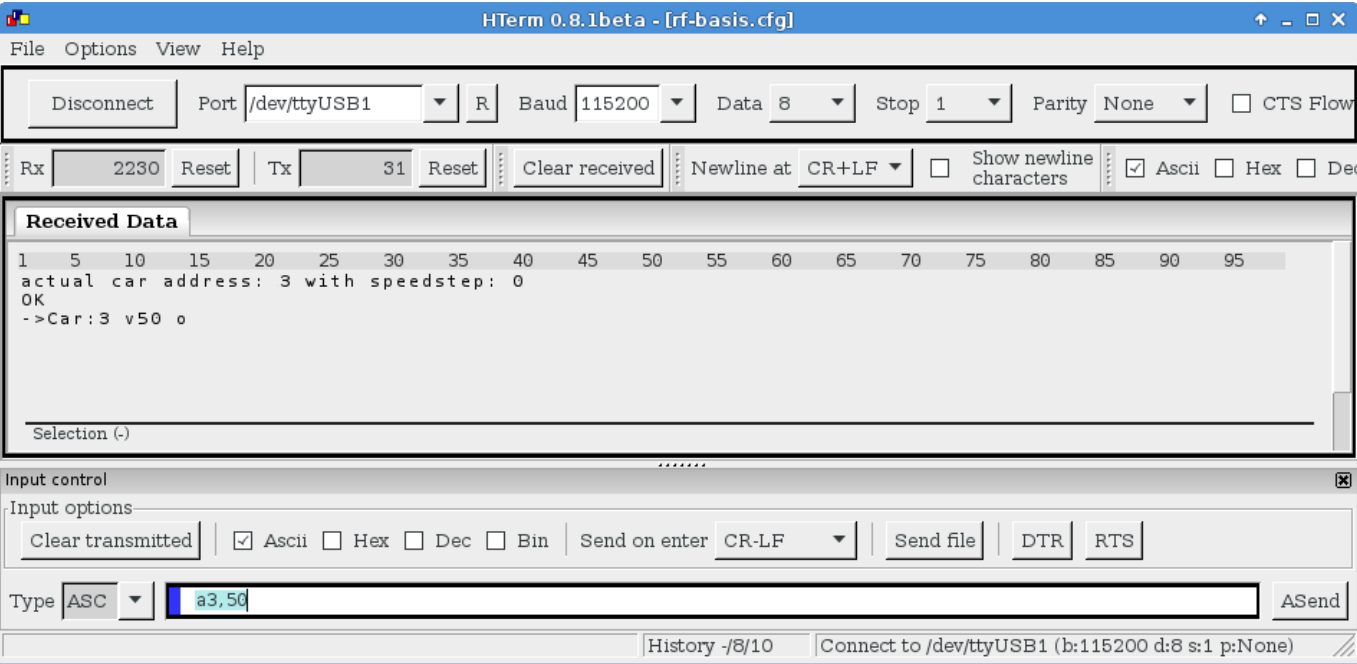
## Adressen

Die RF-Basis merkt sich am USB die Adresse vom letzten über USB angesteuerten Fahrzeug. Diese kann man sich mit **a<enter>** anzeigen lassen:



Man bekommt dann die gerade am USB aktive Adresse und die Fahrstufe dieser Adresse angezeigt. Mit dem „a“ Befehl, kann man auch die Adresse und Fahrstufe vom Fahrzeug setzen:





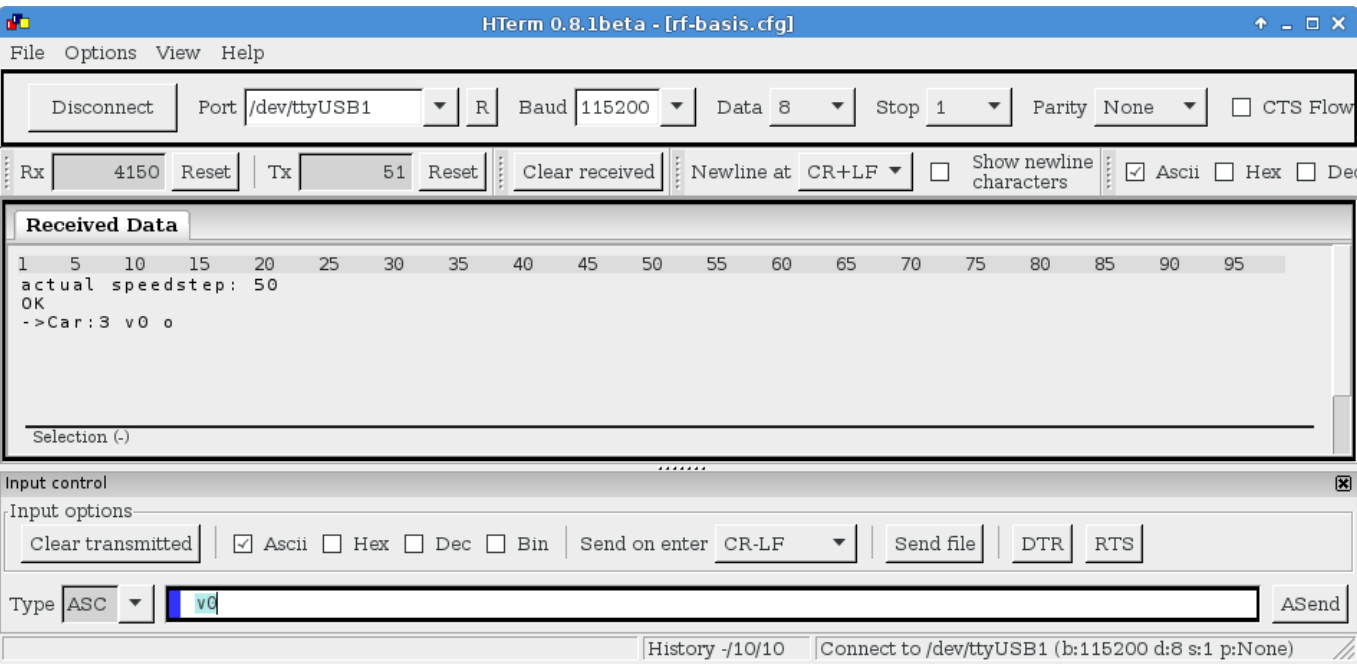
Mit: **a3,50<enter>** (siehe Bild) wird Fahrzeug Adresse 3 mit Fahrstufe 50 angesteuert. Man bekommt dies von der Basis folgendermaßen bestätigt:

**→Car: 3 v50 o**

was folgende Bedeutung hat:

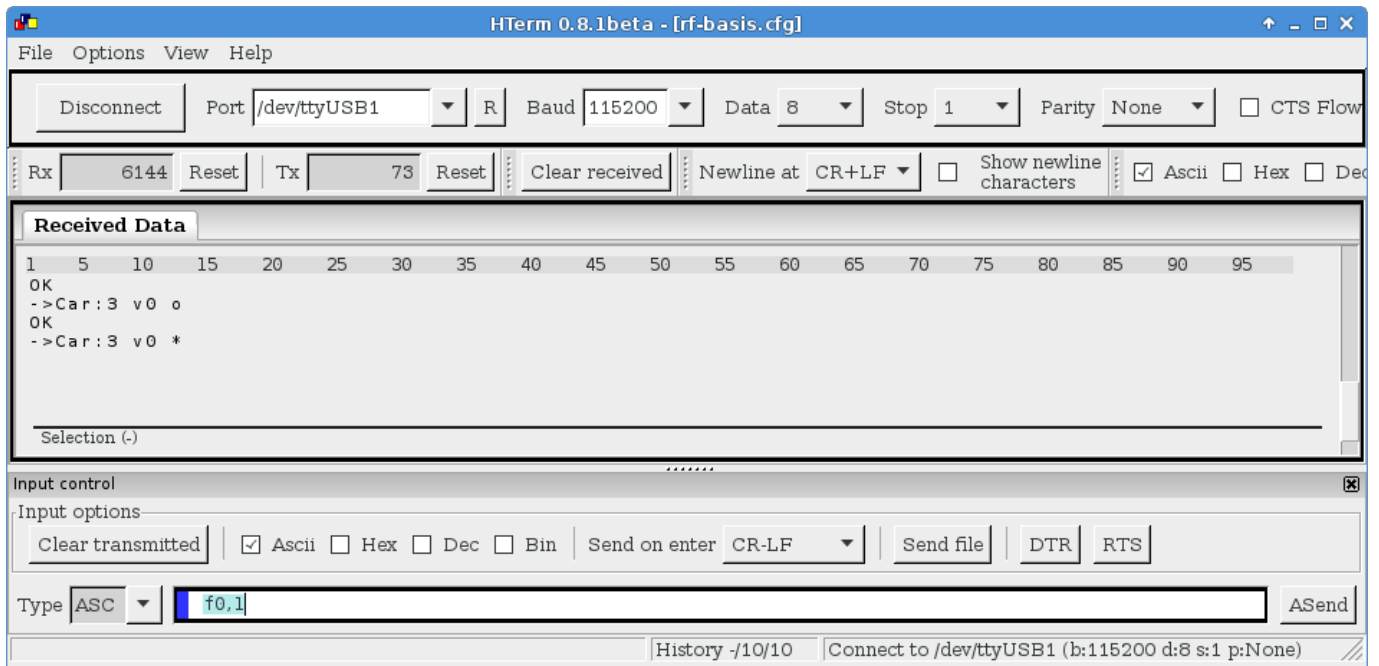
<b>→Car:</b>	es handelt sich um eine Nachricht zum Fahrzeug
<b>3</b>	Adresse 3
<b>v</b>	Fahrtrichtung ist vorwärts (^ für rückwärts, beides ist beim Car ohne Bedeutung)
<b>50</b>	die gesendete Fahrstufe
<b>o</b>	Fahrzeuglicht ist aus (* an der Stelle bedeutet Licht ist an)

## Fahrstufen



Ein **v0<enter>** setzt die Fahrstufe vom aktuellen Fahrzeug auf 0. Man kann das „v“ Kommando also ohne Adressangabe zur Geschwindigkeitsänderung nutzen. **v<enter>** gibt die aktuelle Geschwindigkeit von aktuellem Fahrzeug als Information zurück. Im Bild zu sehen, der Wechsel von Fahrstufe 50 auf 0.

## Funktionen



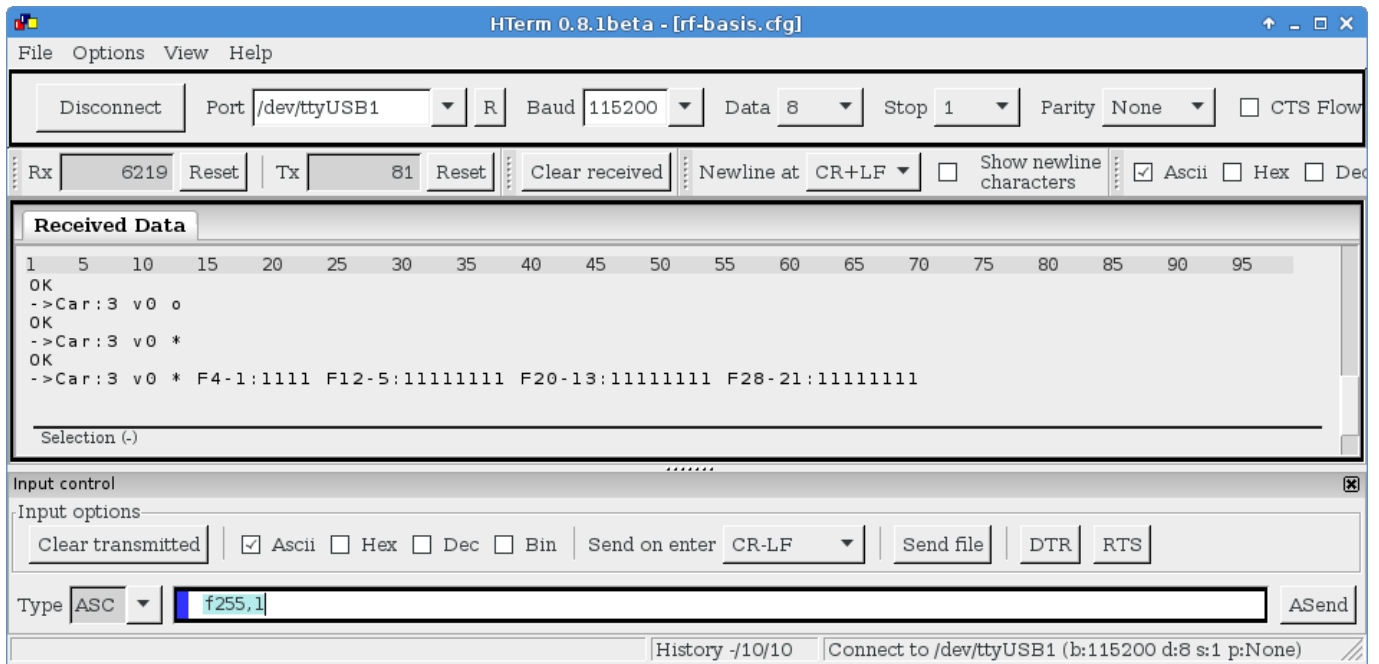
Analog zur Fahrstufe, können alle Funktionen vom Fahrzeug geschaltet werden. Dies kann einzeln, alle auf einmal oder mit dem langen Komplettbefehl (siehe weiter unten) auch ausgewählt erfolgen. Um das Fahrzeuglicht vom aktuellen Fahrzeug einzuschalten, ist folgender Befehl nötig:

**f0,1<enter>**

Als Bestätigung bekommt man wieder die bekannte Antwort (siehe weiter oben). Mit:

**f0,0<enter>**

schaltet man das Licht wieder aus. Dieser Befehl ist für alle 29 (0-28) Funktionen am Cardecoder so möglich. Wird als Funktionsnummer 255 verwendet, schaltet dies je alle Funktionen am Cardecoder ein/aus. Dieser Befehl ist zum Test beim Fahrzeugbau bzw. zum einfachen Abschalten aller Funktionen im Betrieb vorgesehen.



Ist eine beliebige Funktion zwischen 1 und 28 aktiv, erweitert sich die Ausgabe der →Car Meldung um diese Funktionen. Der Bereich mit den Funktionen wird zur Übersichtlichkeit nur angezeigt, wenn dort auch eine Funktion aktiv ist.

## Kompletter Datensatz

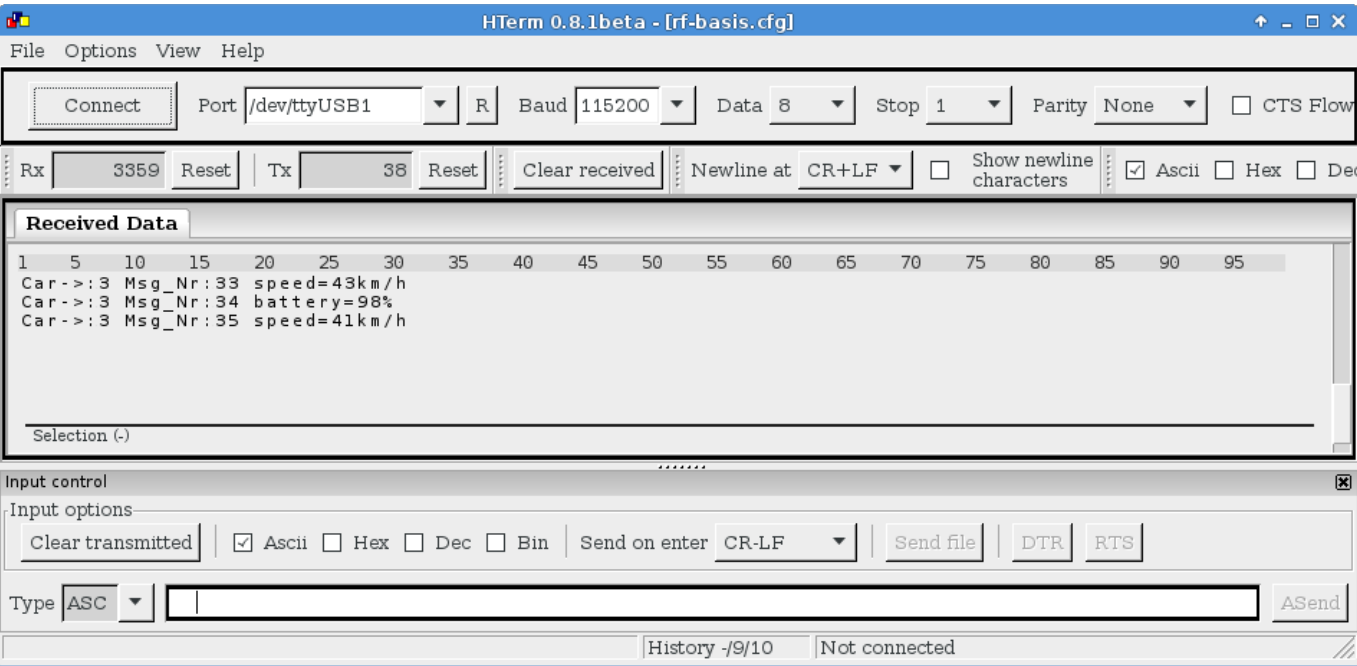
Für eigene Software Projekte ist noch ein „kompletter Car Datensatz“ Befehl implementiert. Hier können Adresse, Fahrstufe und alle Funktionen komplett in einem Befehl übertragen werden:

```
CC<Adresse>,<Format>,<Aktiv>,<Fahrstufe>,<f4-0>,<f12-5>,<f20-13>,<f28-21>,<enter>
```

Dieser Befehl ist vom Format identisch mit dem BiDiB Befehl zur Fahrzeugansteuerung. Der genaue Aufbau kann bei Interesse in der [BiDiB Protokoll Dokumentation](#) nachgeschlagen werden.

## 6.4 Rückmeldungen am USB

Sobald ein Fahrzeug einmalig angesteuert wurde, sendet es bestimmte Fahrzeugparameter zur Basis zurück. Interessant für den Anwender sind dabei die Meldungen zum aktuellen Akkustand, der aktuellen Geschwindigkeit und der momentanen Position des Fahrzeuges.



Das Format dieser Rückmeldungen ist ähnlich dem von gesendeten Befehlen an die Fahrzeuge und hat folgenden Aufbau:

Car→ 3 Msg\_Nr:33 speed=43km/h

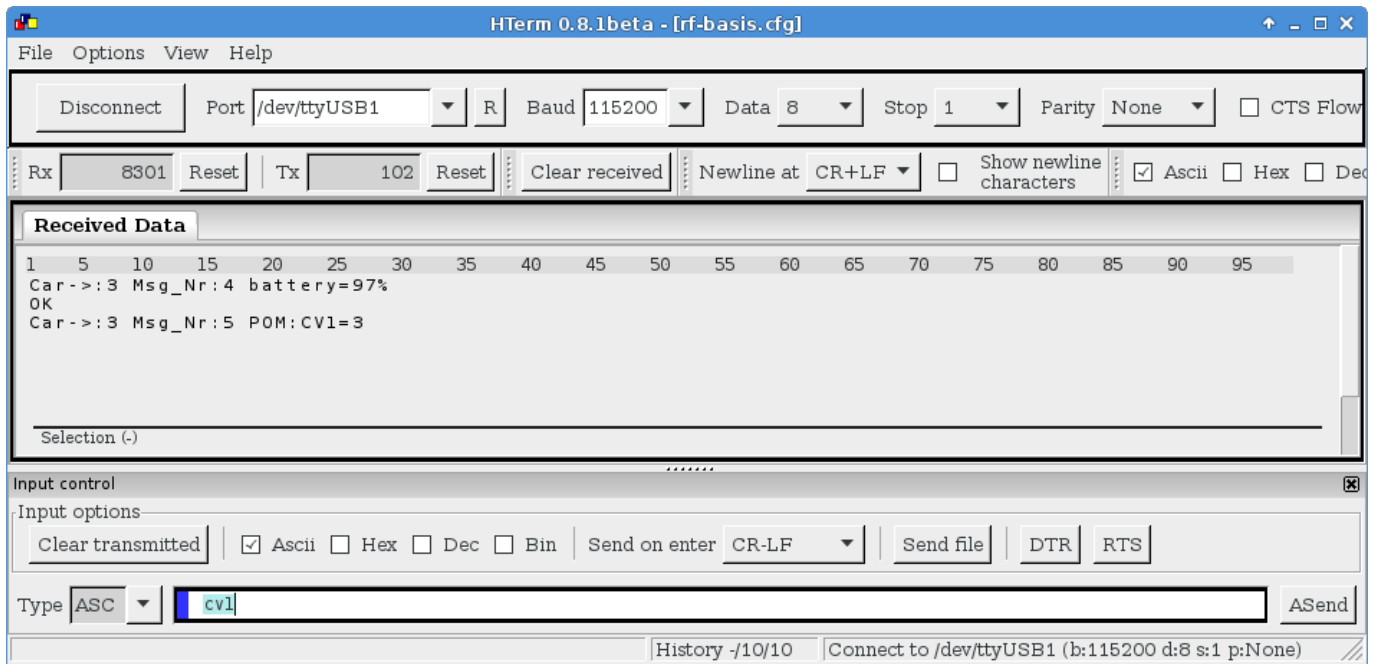
mit folgender Bedeutung:

Car→ 3	Nachricht von Fahrzeug 3
Msg_Nr:33	es ist die 33ste Nachricht von diesem Fahrzeug
speed=43km/h	die Geschwindigkeit beträgt aktuell 43km/h
battery=98%	der aktuelle Akku Füllstand ist 98%

Beachte, die Häufigkeit der Akku- und Geschwindigkeitsmeldungen wird in BiDiB Feature festgelegt. Diese Parameter verwendet die Basis auch in allen anderen Betriebsarten, sie sind nicht doppelt abgelegt.

## 6.5 Decoder Programmierung am USB

Die Fahrzeuge können am USB Anschluss komplett programmiert und ausgelesen werden. Dies betrifft sämtliche CV, auch CV1,17,18 was von diversen Handreglern nicht per POM unterstützt wird. Zum Programmieren, muss das Fahrzeug 1x adressiert werden (z.B.: **a3,0<enter>** für Fahrzeug 3). Ab diesem Zeitpunkt werden die Rückmeldungen vom Fahrzeug abgeholt und am Terminal ausgegeben. Ein CV Lesen, bzw. die Bestätigung eines CV Schreiben erfolgt auch mit Rückmeldungen vom Fahrzeug, das ist ohne eine Adressierung des gewünschten Fahrzeug nicht möglich. Ein Lesen der CV1 ( **CV1<enter>** )gibt dann folgende Ausgabe im Terminal:

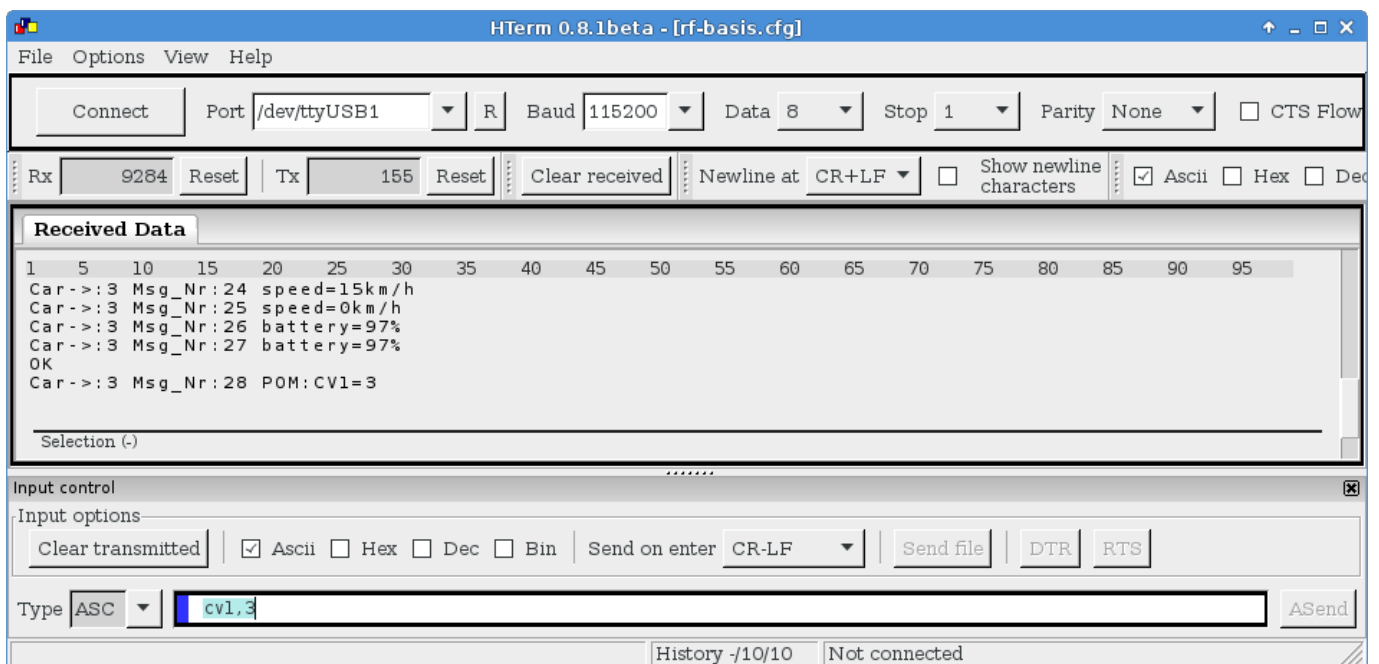


Wir bekommen hier als POM Antwort CV1 ist 3 (logisch das ist die Fahrzeugadresse und wir haben ja bei Fahrzeug 3 angefragt) zurück gemeldet.

Schreiben lässt sich eine CV folgendermaßen, hier wieder am Beispiel von CV1:

**CV1,3<enter>**

dies schreibt in CV1 eine 3 (die ist eh schon 3, weil wir schreiben ja an Fahrzeug 3 😊). Im Terminal sieht das so aus:



Nach diesem Prinzip lassen sich alle CV vom Cardecoder lesen und schreiben. Versucht man einen ungültigen Wert zu schreiben, wird dies mit einem Fehler quittiert.

*Hinweis: Eine Änderung der langen DCC Adresse ist systembedingt immer insofern problematisch,*



*dass hier die Adresse in zwei CVs geteilt ist. Wird die erste CV geschrieben, hätte der Decoder dann vorübergehend eine „Zwischenadresse“, die sich aus „alt zweite CV“ und „neu erste CV“ zusammensetzt. Um dies zu umgehen, müsste man entweder den Decoder vorübergehend auf eine kurze Adresse umstellen, damit die lange ändern und dann wieder auf die lange zurück schalten, oder eben die zweite Programmierung an die „Zwischenadresse“ senden. Beides ist unnütz aufwendig und somit auch fehleranfällig. Aus diesem Grund übernimmt der Cardecoder ein Schreiben in CV17 erst nach einem Neustart oder einem darauf folgendem Schreiben in CV18. Man kann also CV17 und CV18 in dieser Reihenfolge nacheinander an die alte Fahrzeugadresse schreiben, ohne extra Aufwand über kurze Adressierung bzw. Zwischenadresse.*

## 7. Die RF-Basis als DCC Basis

Die Hardware der RF-Basis ermöglicht einen Betrieb der Basis von einem **digitalem DCC Gleissignal** gesteuert (siehe Betriebsarten). Hiermit ersetzt diese Hardware den alten DCC-RF-Booster.

Wird die Basis mit einem DCC Gleissignal gesteuert, sind einige Einschränkungen beim Funktionsumfang gegeben. Zum einen **teilen** sich dann **Bahn und Autos** einen **gemeinsamen DCC Adressbereich**. Es ist also nicht möglich gleichzeitig eine Lokomotive und ein Auto mit z.B. der Adresse 3 zu betreiben. Jede Adresse kann nur 1x entweder für einen Zug oder für ein Straßenfahrzeug verwendet werden.

Am DCC Anschluss gibt die Basis auch keinerlei Rückmeldungen zur DCC Zentrale aus. Fahrzeugmeldungen können aber optional und zusätzlich an einem USB Terminal (siehe USB) mitgelesen werden.

Ein Betrieb am DCC Gleis ist nur möglich, wenn keine BiDiB Verbindung besteht. Ist die Basis am BiDiB angemeldet, deaktiviert sie automatisch den DCC Eingang! Eine gleichzeitige Steuerung aus beiden Signaleingangsquellen ist nicht möglich.

Von der DCC Zentrale können die Fahrzeuge **mit 28 und 128 Fahrstufenbefehlen**, allen Funktionsbefehlen und POM Schreibbefehlen angesteuert werden. Ein POM Lesen und „Service Mode Programmiergleis“ Programmieren ist nicht möglich. Die Programmierung von CV1,17,18 per POM wird unterstützt, sofern die verwendete DCC Zentrale dies zulässt.

Wird das DCC Gleis abgeschaltet (Notstopp am DCC), geht auch die RF-Basis in den Notstopp Status und hält alle Fahrzeuge an. Sobald die Gleisansteuerung wieder zugeschaltet wird, fahren dann auch die Straßenfahrzeuge wieder weiter.

Die Basis Konfiguration ist in dieser Betriebsart nicht möglich, sie kann aber am USB Terminal oder BiDiB durchgeführt werden und steht dann auch in dieser Betriebsart zur Verfügung.

Wird an der RF-Basis die DCC Ansteuerung genutzt, kann gleichzeitig mit einem Handregler am Xpressnet Anschluss ein weiteres **nicht vom DCC angesteuertes Fahrzeug** gesteuert werden. Entscheidend ist hier, dass eine DCC Zentrale zyklisch die Befehle an ein Fahrzeug wiederholt. Wenn jetzt gleichzeitig ein Befehl an das selbe Fahrzeug vom Xpressnet kommt, wird dieser sofort wieder von der DCC Wiederholung überschrieben. Die Basis weiß dann also nicht mehr welcher von den beiden Befehlen an das Fahrzeug gesendet werden soll. Es gibt an einem DCC Gleissignal keine Möglichkeit der Zentrale zu sagen „Hallo jetzt regelt ein Handregler, du hast jetzt Pause“ wie das z.B.

am BiDiB oder zwischen mehreren Handreglern möglich ist.

Trotz dieser Einschränkungen, ist eine Steuerung per Xpressnet bei aktiver DCC Ansteuerung erlaubt, auch wenn dies nur mit noch nicht am DCC adressierten Fahrzeugen möglich ist. So lassen sich einzelne Fahrzeuge auf einer DCC gesteuerten Anlage immer noch manuell fahren oder testen. Sobald ein Fahrzeug allerdings einmal vom DCC angesteuert wurde, ist es dort im Wiederholpeicher enthalten und kann bis zu dessen Neustart nicht mehr per Xpressnet gesteuert werden. Dies wird einem an einem Handregler angezeigt (bei der Multimaus blinkt dann das Loksymbol).

---

## 8. Die RF-Basis als Xpressnet Basis

Xpressnet Handregler können in allen Betriebsarten der RF-Basis zusätzlich oder auch ohne diese autonom genutzt werden. Bei der Ansteuerung vom Xpressnet gelten die gleichen Bedingungen wie am DCC Gleis.

Es können die Fahrzeuge mit **28 und 128 Fahrstufenbefehlen, allen Funktionsbefehlen und POM Schreibbefehlen** angesteuert werden. Ein POM Lesen und „Service Mode Programmiergleis“ Programmieren ist nicht möglich. Die Programmierung von CV1,17,18 per POM wird unterstützt, sofern der verwendete Handregler dies zulässt.

---

## 9. Links

- [Handbuch zur RF Basis Version V1.3](#)
- [Shop - BiDiB RF-Basis](#)
- [HTerm](#)
- [CoolTerm](#)
- [Firmware RF-Basis V01.07.03](#)
- [BiDiB Seriennummern Generator](#)

---

### Wiki Home Open Car System

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

Permanent link:  
<https://forum.opendcc.de/wiki/doku.php?id=ocs:rmbasis:bidibrfbasis&rev=1591617033>

Last update: **2020/06/08 13:50**

