

# LightControl V1.1

„Handbuch für den Aufbau des Löt-Bausatz“

# Inhaltsverzeichnis

Einleitung.....	3
1. LightControl .....	7
1.1 elektrische Parameter / Funktionen.....	7
1.2 Bauteilliste .....	7
2. Aufbaubeschreibung der Basisversion .....	9
1.Schritt: die Spannungsversorgung Teil1 .....	9
2.Schritt: die Spannungsversorgung Teil2 .....	10
3.Schritt: der Prozessor .....	10
3. Aufbaubeschreibung der verschiedenen Funktionen .....	12
3.1 DCC /ACK und Railcom - Interface.....	12
3.2 BiDiB – Interface.....	13
3.3 8 Meldeeingänge (im Basisbausatz enthalten) .....	14
3.4 16 Powerausgänge .....	14
3.5 4x Servo-Ausgänge .....	14
3.6 32x LED-Ausgänge .....	15
4. Firmware aufspielen / flashen.....	16
Variante 1 .....	18
Variante 2 / Firmware Update .....	20
5. Statusanzeigen .....	22
6. Terminierung des BiDiBus .....	23
6.1 Vorgehensweise der Terminierung .....	23
7. Hinweis .....	24



## Einleitung

Diese Anleitung bezieht sich auf den BiDiB / DCC-Decoder "LightControl" von **OpenDCC** und **Fichtelbahn**. Der Decoder versteht sich nicht als kommerzielles Fertigprodukt, sondern ist eine Entwicklungshilfe oder Bausatz für technisch interessierte Modellbahner zum Eigenbau.

### Hier noch einmal ein klarer Hinweis:

Der Decoder und diese Anleitung wurden sorgfältig geprüft und nach bestem Wissen erstellt. Für die hier dargebotenen Informationen wird kein Anspruch auf Vollständigkeit, Aktualität, Qualität und Richtigkeit erhoben. Es kann keine Verantwortung für Schäden übernommen werden, die durch das Vertrauen auf die Inhalte dieser Anleitung, dem Decoder oder deren Gebrauch entstehen.

Die Software des Decoders kann auf unserer Internetseite gedownloadet werden darf von jedem benutzt, erweitert und verbessert werden.

Eine kommerzielle Nutzung der Software oder Teile daraus sind nicht erlaubt!

### Gewährleistung

Die Verwendung dieser Betriebsanleitung ist nur für den Nachbau und den Eigenbedarf des beschriebenen Bausteins erlaubt. Eine anderweitige Nutzung bedarf der schriftlichen Einwilligung des Verfassers. Für den Nachbau und dessen Funktionen des beschriebenen Bausteins übernimmt der Verfasser keinerlei Haftung. Für die Einhaltung bestehender Vorschriften und dem vorschriftsmäßigen Einsatz des Produkts ist der Betreiber alleine verantwortlich.

### Hinweis:

**RailCom®** und **RailComPlus®** sind eingetragene Warenzeichen der Firma Lenz Elektronik GmbH und ESU electronic solutions ulm GmbH & Co. KG. Zur Erhöhung der Lesbarkeit des Textes haben wir darauf verzichtet, bei jeder Verwendung des Begriffes darauf zu verweisen.

## Wissenswertes zur BiDiB-Seriennummer / Unique-ID



Das Bild zeigt eine **Unique-ID** auf einem BiDiBus Device.

Alle SMD bestückten BiDiB-Baugruppen die über den Fichtelbahn-Shop bezogen werden und am BiDiBus hängen werden mit einer aufgespielten BiDiB-Seriennummer ausgeliefert. Diese Seriennummer ist ein Bestandteil der Unique-ID und deshalb wird auch auf der Rückseite des Bausteins diese komplette Unique-ID abgedruckt.

### Was ist eine Unique-ID?

Die vom Hersteller in dem Baustein fest programmierte, eindeutige Kennung, bestehend aus 16 Bit Herstellerkennung und 32 Bit herstellerspezifischer Nummer (z.B. Produktindex und Seriennummer).

**V** = VID (Herstellerkennung)

**OD** = Selbstbauprojekte

**P** = PID (Produktkennung)

**6800** = GBMboost Master

**6700** = GBMboost Node

**6B00** = LightControl

**CCEA** = Seriennummer

Die Angaben der Unique-ID sind in HEX.

### Wozu dient die Unique-ID?

Die Unique-ID ist eine absolut einmalige Nummer, mit dieser Nummer wird eine Baugruppe unabhängig von ihrem Einbauort und ihrem Platz am Bus gefunden.

Das heißt: Das BiDiB-System führt ein Art 'Telefonbuch', unter welchem Anschluss welche Baugruppe erreicht werden kann. Das Hostprogramm vergibt dann Namen für die einzelnen Anschlüsse. Die Unique-ID ist hier das Verbindungsglied zwischen der Bezeichnung am PC und der Baugruppe.

#### Beispiel:

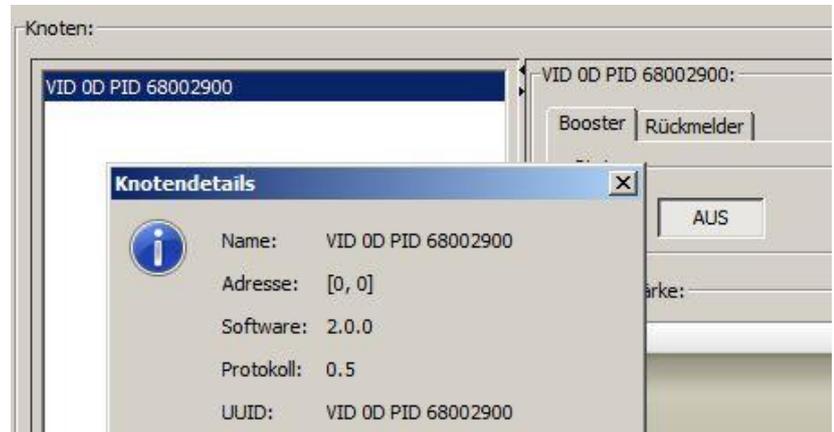
Eine LightControl sei unter dem Bahnhof verbaut und habe die Unique-ID **OD 6B001234**. Im Hostprogramm wird sie als HBF-West bezeichnet. Das BiDiB System meldet dem Hostprogramm: 0D6B001234 findest Du unter Anschluss 3. Wenn man jetzt in HBF-West was umstellt, guckt das Hostprogramm im Telefonbuch nach: aha, ich soll 3 anrufen.

**Damit hat man im Normalfall nichts mehr mit Adressen und Dipschalter zu tun.**

## Warum sollte man diese Nummer aufkleben? Für was könnte man diese Angabe gebrauchen?

Alle Tools aber auch die aktuellen PC-Steuerprogramme kommunizieren und verwalten die Bausteine über diese Nummer. Es gibt jetzt keine "DCC - Adresse XX" mehr, sondern der Befehl wird an den Knoten \*Unique-ID X\* und dessen Port gesendet.

Der Umkehrschluss ist, dass für die Konfiguration einer neuen Aktion an dem Ausgang eines Decoders (z.B. LightControl) oder der Zuordnung von einem Melder im Gleisbild (z.B. GBM), die passende Unique-ID des BiDiB-Bausteins dem Programm mitgeteilt werden muss.



## Hardware-Seriennummer:

Die aufgeklebte Seriennummer mit S/N auf der Rückseite der Baugruppe ist eine Hardware-Seriennummer und hat nichts mit der Unique-ID Seriennummer für die BiDiB-Baugruppen zu tun. Mit dieser Nummer ist Ihr Baustein bei uns registriert. Im Servicefall bitte die komplette Nummer angeben.  
z.B.: 50131240001000023



## Wo wird die Unique-ID auf dem Baustein gespeichert?

Das Verfahren hier ist etwas zweigeteilt. In der Flash/EEPROM-Firmwaredatei ist die VID und PID bereits integriert und wird auf den Baustein übermittelt, jedoch ist noch keine Seriennummer enthalten. Die **Firmwaredateien** können daher problemlos auf den Prozessor geladen werden oder auch **aktualisiert** werden. Eine bereits vorhandene **Seriennummer wird dabei nicht überschrieben**. Wenn der Baustein beim Starten noch keine geladene Seriennummer erkennt, blinkt ein Fehlercode und der Betrieb wird blockiert.

**Ohne Seriennummer läuft die Firmware nicht!**

*(Ausnahme beim GBMboost Master: hier wird eine Notfall-S/N erzeugt mit der Kennung: 0100. Diese S/N kann jederzeit mit dem Nachschieben der gültigen S/N überschrieben werden)*

Die Seriennummer wird immer als dritter Akt der Inbetriebnahme nachgeschoben und landet als erstes im EEPROM des Bausteins.



### **Bei jedem Neustart führt der Baustein folgende Überprüfung aus:**

Er schaut in dem User Signatur Bereich, ob schon eine Seriennummer vorhanden ist. Ist keine Seriennummer vorhanden wird die Seriennummer aus dem EEPROM hier abgelegt und der Baustein startet seinen Betrieb. Ist im Bereich der USER Signatur schon eine Seriennummer vorhanden wird die neuaufgespielte Seriennummer vom EEPROM verworfen und die schon hinterlegte Seriennummer aus dem User Signatur Bereich verwendet.

Die hinterlegte Seriennummer im USER Signatur Bereich bleibt auch erhalten bei einem CHIP ERASE und kann nur gelöscht werden mit einem USER Signatur ERASE.

Falls man die Seriennummer auf einem BiDiBus tauglichen Device tauschen müsse, dann funktioniert das nur über ein vorheriges USER Signatur ERASE.

Das hat aber auch zur Folge, dass vorhandene Kalibrierungswerte verloren gehen, weil diese ebenfalls in diesem Bereich gesichert werden.

**Dies sollte nicht ohne speziellen Grund durchgeführt werden.**

### **Ich habe aus Versehen die aufgedruckte Seriennummer gelöscht. Wie kann diese wieder aufgespielt werden?**

1. Man zieht eine neue Seriennummer über den Generator und spielt diese nach dem löschen des USER Signatur Bereich auf das BiDiBus Device.

**Wichtig:** Jetzt nicht vergessen den bestehenden Aufkleber durch die neue Unique-ID zu ersetzen.

Link zum Generator: [http://www.opendcc.de/elektronik/bidib/opendcc\\_bidib.html](http://www.opendcc.de/elektronik/bidib/opendcc_bidib.html)

2. Man möchte die vorhandene (aufgedruckte) Unique-ID weiterverwenden, weil diese schon in den Gleisbildern meines PC-Programms hinterlegt wurde.

In diesem Fall kontaktieren Sie den Fichtelbahn-Support **mit der Angabe der aufgedruckten Unique-ID Nummer und der zugehörigen Hardware-Seriennummer.**

Dies funktioniert nur bei den SMD-vorbestückten Bausteinen! Bei eigenständig gezogenen Seriennummern müssen Sie in der Historie Ihrer gezogenen Seriennummern im Generator, nach der passenden Nummer suchen.

# 1. LightControl

## 1.1 elektrische Parameter / Funktionen

- **Eingangsspannung 9-18V AC/DC**
- **Schaltregler mit 3A** Ausgangsleitung (durch Sicherung auf 1A begrenzt)
- **16** frei programmierbare **Schaltausgänge** mit je 300mA
- **32 LED-Ausgänge** mit einstellbarer Stromquelle u. Helligkeitsregelung über PWM
- **4 Servoausgänge**, Stromversorgung der Servo schaltbar
- **8** universelle **Eingänge**
- **DCC-Eingang**, railcom®
- **BiDiBus-Anschluß**
- **Universalansteuerport** für USB (FTDI)
- **Platinengröße:** 80mm x 100mm

## 1.2 Bauteilliste

### Basisversion:

(Spannungsversorgung, Prozessor u. Systemteile):

1x	L1	<b>SLF12575 33µH</b>	SMD-Spule
1x	L2	<b>Bead 40Ohm 600mA</b>	Filter
1x	D9	<b>SSA33L</b>	SMD-Diode
2x	C37, C38	<b>15pF / 0603</b>	SMD-Kondensator
1x	C36	<b>22µF Bauform A</b>	SMD-Tantal
1x	B1	<b>B80C1500</b>	Gleichrichter
4x	LED5-LED8	<b>LED-0603 gelb, rot, grün</b>	Leuchtdiode
1x	RN8	<b>BCN16 1,0kOhm</b>	SMD-Parallelwiderstand
4x	C3,C6,C16,C17	<b>100nF / 0805</b>	SMD-Kondensator
10x	C24,C25,C34		
	C35,C27-C32	<b>100nF / 0603</b>	SMD-Kondensator
1x	C4	<b>1000µF/35V</b>	Elektrolytkondensator
1x	C8	<b>470µF/16V</b>	Elektrolytkondensator
1x	C5	<b>100µF/16V</b>	Elektrolytkondensator
1x	C33	<b>SMD TAN.22/20</b>	Tantal-Kondensator
1x	IC1	<b>LM2596S</b>	Schaltregler 3A
1x	IC4	<b>LF 33 CV</b>	Spannungsregler 3,3V
1x	IC7	<b>ATXMEGA128A1</b>	Mikrocontroller
1x	Q1	<b>HC49-SMD 8MHz</b>	Quarz
1x	F1	<b>PFRA 050</b>	Multifuse
1x	S1	<b>Taster 9314</b>	SMD -Taster
1x		<b>Jumper 2,54 SW</b>	Jumper
1x		<b>SL 1x36G 2,54</b>	Stiftleiste
1x	X2	<b>AKL055-03</b>	Anreihklemme 3polig
1x	J11	<b>PSS 254/2</b>	Printstecker
1x		<b>V FI343</b>	Aufsteckkühlkörper
1x	R37	<b>1,0kOhm 0805</b>	SMD-Widerstand
1x	R38	<b>3,3kOhm 0805</b>	SMD-Widerstand

#### Upgrade Servo:

1x	R24	<b>10kOhm 0603</b>	SMD-Widerstand
1x	RN5	<b>BCN16 1,0kOhm</b>	SMD-Parallelwiderstand
1x	RN3	<b>BCN16 1000Ohm</b>	SMD-Parallelwiderstand
1x	R5	<b>10Ohm 2W Axial</b>	Widerstand
1x	C19	<b>47µF / 16V SMD</b>	Elektrolytkondensator
4x	C1, C2, C7, C9	<b>47µF / 6 1206</b>	Kondensator
1x	C14	<b>100nF / 0603</b>	SMD-Kondensator
1x	IC8	<b>74VHCT 541D</b>	Buffer
4x	Q2, Q4, Q5, Q6	<b>IRLML 5203</b>	Mosfet-Transistor
1x	JP9	<b>Jumper</b>	Jumper
1x		<b>SL 1x20G 2,54</b>	Stiftleiste

#### Upgrade Power-Ausgänge:

2x	IC2, IC6	<b>ULN2803</b>	Treiber
2x	SV1, SV2	<b>WSL16G</b>	Wannenstecker 16polig

#### Upgrade LED-Ausgänge:

2x	R34, R36	<b>10k Poti SMD</b>	SMD-Poti
2x	U\$1, U\$2	<b>TLC5941</b>	LED-Treiber
2x	R33, R35	<b>1kOhm 0603</b>	SMD-Widerstand
2x	C22, C23	<b>100nF 0603</b>	SMD-Kondensator
2x	SV3, SV4	<b>WSL 34G</b>	Wannenstecker 32polig

#### Upgrade Eingänge:

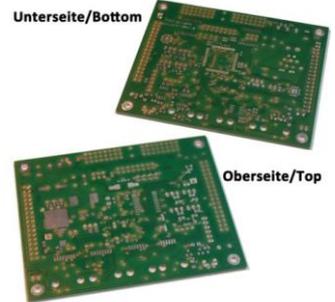
1x	SV5	<b>WSL16G</b>	Wannenstecker 16polig
2x	RN1, RN2	<b>BCN16 1,0kOhm</b>	SMD-Parallelwiderstand

#### Upgrade DCC-Interface:

1x	OC1	<b>PC354</b>	Optokoppler
1x	R20	<b>330Ohm 1206</b>	SMD-Widerstand
1x	R21	<b>150Ohm 0603</b>	SMD-Widerstand
1x	R27	<b>100Ohm 0805</b>	SMD-Widerstand
3x	R22, R23, R26	<b>10kOhm 0603</b>	SMD-Widerstand
1x	R32	<b>1,5kOhm 0805</b>	SMD-Widerstand
1x	R19	<b>1kOhm 0805</b>	SMD-Widerstand
2x	C15, C11	<b>100nF 0603</b>	SMD-Kondensator
1x	C13	<b>100µF / 35V</b>	Elektrolytkondensator
2x	D1, D11	<b>4,7V MiniMelf</b>	SMD-Z-Diode
3x	D2, D4, D5	<b>BAV199</b>	SMD-Diode
3x	T1, T2, T4	<b>BC860</b>	SMD-Transistor
1x	T3	<b>BC850</b>	SMD-Transistor
1x	IC3	<b>ADUM 1201</b>	Optokoppler
1x	X1	<b>AKL 055-02</b>	Anreihklemme 2polig

#### Upgrade BiDiB-Interface:

2x	X8, X9	<b>MEBP 8-8G</b>	RJ45 Buchse
1x	IC5	<b>ISL83075EIBZA</b>	Transceiver
2x	R7	<b>10kOhm 0805</b>	SMD-Widerstand
1x	C10	<b>100nF 0805</b>	SMD-Kondensator
1x	R30	<b>120Ohm 1206</b>	SMD-Widerstand
1x	J6	<b>Jumper RM2,54</b>	Jumper RM2,54 (Terminierung)
1x		<b>SL 1x20G 2,54</b>	Stiftleiste



## 2. Aufbaubeschreibung der Basisversion

Für den Aufbau ist nicht viel nötig. Ein LötKolben, mit feiner Lötspitze, Lötzinn maximal 1 mm stark, Flussmittel, eine feine Pinzette, ein Seitenschneider, ein Ohmmeter und eventuell etwas Entlötlitze. Zum programmieren ist ein AVR JTAG Programmer und die passende Software dafür notwendig. Ich verwende Atmels AVR-Studio und einen „AVRICE mkII“.



Falls Sie einen Bausatz von Fichtelbahn bezogen haben, können Sie den beiliegenden Positionsplan zum auffinden der Bauteile verwenden. Auf der Bausatzstückliste finden Sie zu jedem Bauteil die dazugehörige Bauteilnummer.

### 1.Schritt: die Spannungsversorgung Teil1



Als erstes bestücken wir auf der Oberseite der Platine den Gleichrichter **B1**. Folgend **F1**, **X2**, **C4** und auf der Rückseite der Platine den SMD Kondensator **C16**.

*Bei Versorgung der LightControl über eine externe Quelle (**empfohlen**) bleiben die beiden Lötbrücken **SJ6** und **SJ7** offen. Wenn der Baustein über die DCC Spannung gespeist werden soll müssen die beiden Brücken geschlossen werden und **X2** nicht bestückt.*

*Die LightControl kann mit **9V–18V Gleichstrom und Wechselstrom** versorgt werden. Mit Hilfe des Gleichrichters liegt immer die passende Polarität am Schaltregler. Auch wenn Sie den Baustein mit einer Gleichspannung versorgen kann der Gleichrichter bestückt werden, somit erhalten Sie einen **Verpolungsschutz**. Wenn Sie den Baustein mit einer Gleichspannung versorgen, sollte an **X2-3** die Masse (GND) angeschlossen werden.*

**Wichtig: Achten Sie bei der DC-Variante auf die richtige Polarität an den Klemmen X2 (X2-2 und X2-3).**

Folgend **C17**, die Diode **D9** (Einbaurichtung beachten), Spule **L1** und Stützkondensator **C8**. Der Schaltregler **IC1** kann liegen und stehend je nach verwendeter Bauart montiert werden (Einbaurichtung beachten). Bei der stehenden Variante den Kühlkörper nicht vergessen.

--> **stehende Variante LM2596 5V:**

Das Lötpad **R38** kann mit Lötzinn geschlossen werden und **R37** wird nicht bestückt.

--> **liegende Variante LM2596-ADJ (im Bausatz enthalten):**

Für diese Variante müssen die beiden Widerstände **R37** und **R38** eingebaut werden.



### Kontrolle:

Zum testen, wird an der Klemme X2 eine 9-18 Volt AC oder DC Spannung angelegt. An der Spule L1 sollte jetzt eine Spannung von **5 Volt** messbar sein.

**Folgen Sie dem nächsten Bauabschnitt nur bei erfolgreicher Spannungsmessung!!**



## 2.Schritt: die Spannungsversorgung Teil2



Weiter geht es mit dem 3,3V Spannungsversorgung für den Mikrocontroller. Hierzu erst die beiden 100nF SMD-Kondensatoren **C6** und **C3** auf der Rückseite verlöten. Danach folgt der linear Regler **IC4** (Einbaurichtung beachten) und der Stützkondensator **C5**.

**Fertig ist die Spannungsversorgung für LightControl.**

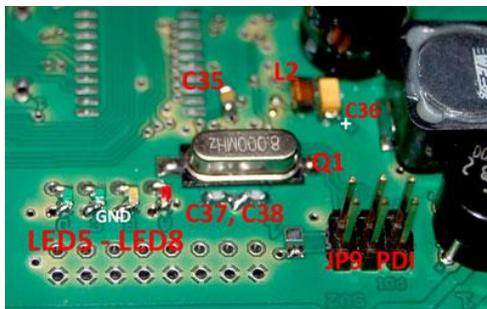
### Kontrolle:

Zum testen, kann der 3,3V Printstecker auf Position **J11** eingelötet werden. Hier sollten jetzt 3,3V messbar sein, gegeben falls kontrollieren Sie nochmals alle verlöteten Bauteile auf Ihre Richtigkeit.

**Folgen Sie den nächsten Bauabschnitt nur bei erfolgreicher Spannungsmessung!!**



## 3.Schritt: der Prozessor



Wir beginnen mit den Status-Leds **LED5 bis LED8**, hier auf die richtige Einbaurichtung achten (**Kathode in Richtung Pfostenstecker / GND**).

Die richtige Einbaurichtung kann mit dem Multimeter (Diodentester) ermittelt werden. Löten Sie erst die Kathode ein (Richtung Pfostenstecker und testen sie mit dem Multimeter (rote Messspitze) an der Anode, ob die LED leicht leuchtet.



Für die Referenzspannung am ATXMEGA muss noch **C36** und die Spule **L2** bestückt werden. Danach die restlichen Bauteile **C37**, **C38** und **Q1** (hier entsteht der Takt für den ATXMEGA) sowie die Programmierstiftleiste „PDI“ **JP9**.

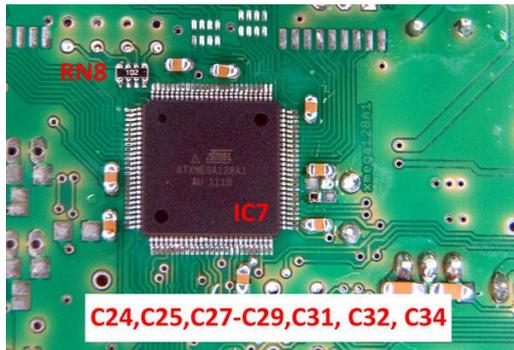
Bei dieser Stiftleiste muss man den Plastiksockel etwas verschieben so, dass die Stifte auf der Rückseite nicht aus der Platine schauen. Hier wird später noch ein Treiberbaustein bestückt.

Weiter geht es mit dem Tantalkondensator **C33** (Polung beachten), Taster **S1** und Demo-Jumper **JP2** und der Netzwerkwiderstand **RN8** (Bild auf der nächsten Seite).

### Kontrolle:

Die Platine mit GND verbinden und mittels anlegen von 3,3 VDC über einen bedrahteten Widerstand von 100 Ohm Spannung zuführen. Streichen Sie einfach mit dem Widerstand entlang der Pads des unbestückten Mikrocontrollers. Sie können den Test auch mit einem Ohmmeter durchführen.





Jetzt wird die ATXmega 128A1 **IC7** bestückt.  
Zuerst exakt auf einen Eck-Pin ausrichten, dann gegenüber anlöten. Nochmals die Pin1-Orientierung prüfen (nicht dass der Käfer im Eifer des Gefechtes falsch rum drauf kommt).

### In sechs Schritten ist der Käfer auf der Platine:

- 1) Prozessor auf der Platine fixieren mit ein paar gegenüberliegenden Pins.
- 2) Die Pins kräftig mit Flussmittel tränken.
- 3) einfach alle Pins unter sparsamer Zugabe von Lötzinn anlöten indem man mit dem LötKolben von einer Seite zur anderen Pinseite zieht.
- 4) das überschüssige Zinn am Ende der Lötfahrt mit einer LötSauglitze oder Pumpe aufsaugen.
- 5) den letzten "mini" Rest mit dem LötKolben nach außen ziehen
- 6) Alle Pins unter einer Lupe auf eventuelle Kurzschlüsse prüfen. Notfalls nochmals wiederholen und nicht das Flussmittel vergessen!!



#### kleiner Tipp:

Auf [fichtelbahn.de](http://fichtelbahn.de) gibt es unter LightControl / Wissenswertes einen kleinen Videoclip indem man das Handling sehen kann.

Nachdem der große schwarze Käfer verlötet ist kommen noch die Abblockkondensatoren **C24, C25, C27, C28, C29, C30, C31, C32, C34, C35** (C30 und C35 sind auf der Oberseite zu finden)

### Kontrolle:

Nachdem die ATXmega - Pins nochmal mit der Lupe auf Kurzschluss untersucht wurden (mind. zweimal checken, unter verschiedenen Blickwinkeln!) wird mittels eines strombegrenzten Netzteiles Spannung angelegt und der Leerlaufstrom überwacht.

**Dieser sollte ca. um die 10mA liegen.**



### Vorgehensweise:

- 1) drehen Sie Spannung auf den Wert 0
- 2) schließen Sie den Baustein an ihr regelbares Netzteil
- 3) erhöhen Sie die Spannung langsam bis sie die 9V-18V erreicht haben, dabei achten Sie auf das Amperemeter!

### Hinweis:

Im Einschaltmoment liegt der Strom etwas über den 10mA und fällt dann zurück. Sollten Sie mehrere 100mA messen dann trennen Sie sofort den Baustein von der Versorgung und überprüfen nochmals die letzten Punkte auf flachen Einbau bzw. Kurzschlüsse.

**Folgen Sie den nächsten Bauabschnitt nur bei erfolgreicher Strommessung!!**

### 3. Aufbaubeschreibung der verschiedenen Funktionen

#### Kommunikationsinterfaces:

- DCC-, ACK- und Railcom
- BIDIB

#### Funktionen:

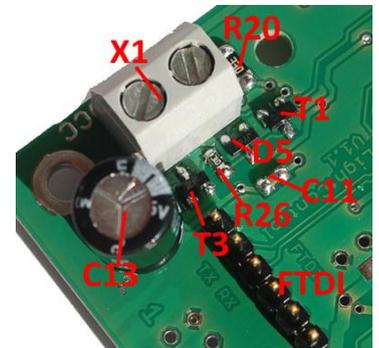
- Eingänge für interne Funktionen und Rückmelder
- 16x Powerausgänge für Effekte und Magnetartikel
- 4x Servo-Ausgänge
- 32x LED-Ausgänge

#### 3.1 DCC /ACK und Railcom - Interface

Wir beginnen mit den Kleinsignal SMD-Transistoren **T1** und **T3** auf der Oberseite und **T2** und **T4** auf der Unterseite. Der **T3** ist ein BC850 (NPN) Transistor nicht verwechseln mit den drei anderen!

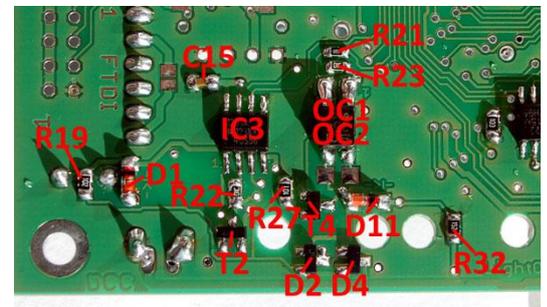
Von den drei BAV199 SMD-Dioden befindet sich die **D5** auf der Oberseite und die **D2** und **D4** auf der Unterseite.

Die zwei MiniMelf 4,7V-Zener Dioden **D1** und **D11** sind ebenfalls auf der Unterseite. Hier gibt es noch den Optokoppler **IC3** und **OC1** oder **OC2**. Zum Schluss bleiben uns noch die beiden SMD-Abblockkondensatoren **C15** und **C11**, der Stützkondensator **C13** (Polung beachten), eine Klemme **X1** und ein paar SMD-Widerstände:



Oberseite: **R20** und **R26**

Unterseite: **R19, R22, R23, R21, R27** und **R32**



## 3.2 BiDiB – Interface

Die Schnittstelle BiDiB für die adresslose Kommunikation besteht aus sehr wenigen Teilen.

Auf der Oberseite befinden sich die beiden RJ45-Buchsen **X8** und **X9** wobei die Platine für eine SMD oder für eine DIL-Variante vorbereitet ist. Die drei SMD-Widerstände **R7**, **R15** und **R16** befinden sich auf der Unterseite des Bausteins. Der RS-485 Transceiver **IC5** besitzt noch einen Abblockkondensator **C10**.



### Wichtig:

**Je nach eingebauten RS-485 Transceiver müssen die beiden Solder Jumper unterschiedlich geschlossen werden:**

**3,3V:** R10 geschlossen und R11 offen --> beim MAX3075E und SN65HVD12DR

**5V:** R10 offen und R11 geschlossen --> beim SN65HVD22 und SP3085EEN-L

### Hinweis:

**Bei einer Hotplug-Funktion oder einer weitläufigen BiDiBus Verkabelung (z.B. Turnhalle) empfehlen wir noch eine Schutzdiode SM712 mit zu bestücken.**

(gibt es auch im Fichtelbahn-Shop)

Die Bestückung der **SM712 D44** erfolgt über die beiden Pads **D7** und **D8** auf das Masse Via.



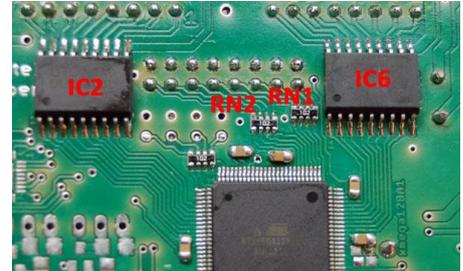
Der Widerstand **R30** und die Stiftleiste **J6** wird für die BiDiBus Terminierung benötigt und sollte ebenfalls bestückt werden. **Weitere Informationen dazu im Kapitel 6.**



### 3.3 8 Meldeeingänge (im Basisbausatz enthalten)

Diese acht Eingänge können für Aktivierung / Auslösung von internen Effekten aber auch zur Rückmeldung an das PC-System verwendet werden.

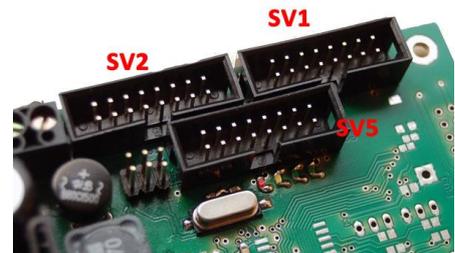
Bei Verwendung müssen die beiden Parallelwiderstände RN1 und RN2 sowie der Pfostenstecker SV5 bestückt werden.



### 3.4 16 Powerausgänge

Diese 16 Ausgänge sind geeignet für Lichteffekte mit Glühbirnen, zum schalten von größeren Verbrauchern **max. pro Ausgang 300mA** oder auch für Magnetartikeln.

Bei der Verwendung müssen die beiden ULN2803D's IC2 und IC6 sowie die beiden Pfostenstecker SV1 und SV2 eingelötet werden.



### 3.5 4x Servo-Ausgänge

Die vier Servo-Ports können für Modellbahneffekte oder auch für Servo-Weichen verwendet werden.

Bei Benutzung dieser Funktion müssen auf der Oberseite der Parallelwiderstand RN3, R24 und das Buffer IC IC8 bestückt werden. Beim Stützkondensator C19 (47µF/16) kann eine SMD oder Radial Variante verwendet werden. Dieser Kondensator sollte aber die **ESR-Eigenschaft** besitzen.

Bei dem 1 Ohm Widerstand R5 oder R6 kann zwischen einem Axial oder SMD gewählt werden. Der letzte Einbau auf der Oberseite sind die fünf Stiftleisten JP3, JP4, JP5, JP6 und JP9/J12.



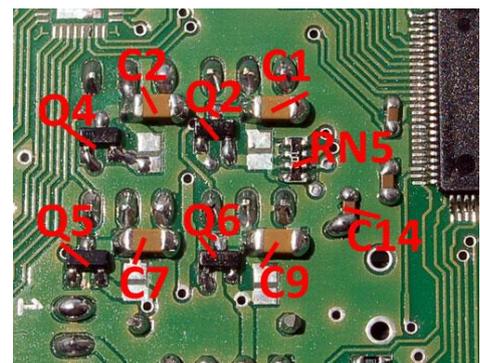
#### ACHTUNG Kurzschlussgefahr:

Beim JP9/JP12 kann die 5V Servoversorgung extern gespeist werden oder eine 5V Versorgung abgegriffen werden. Für eine interne 5V Servo- versorgung muss der Pin2 und der Pin3 verbunden werden.

Auf der Rückseite müssen die vier Mosfets Q2, Q4, Q5, und Q6 bestückt werden.

**Hinweis:** Verwendet werden können hier drei Typen: SI2309, NTR1P02LTG oder IRLML5203.

Den Abblockkondensator C14 fürs IC8, einen weiteren Parallelwiderstand RN5 und die vier 47µF Stützkondensatoren in SMD C1, C2, C7, C9 für die Servo-Ports.



### 3.6 32x LED-Ausgänge

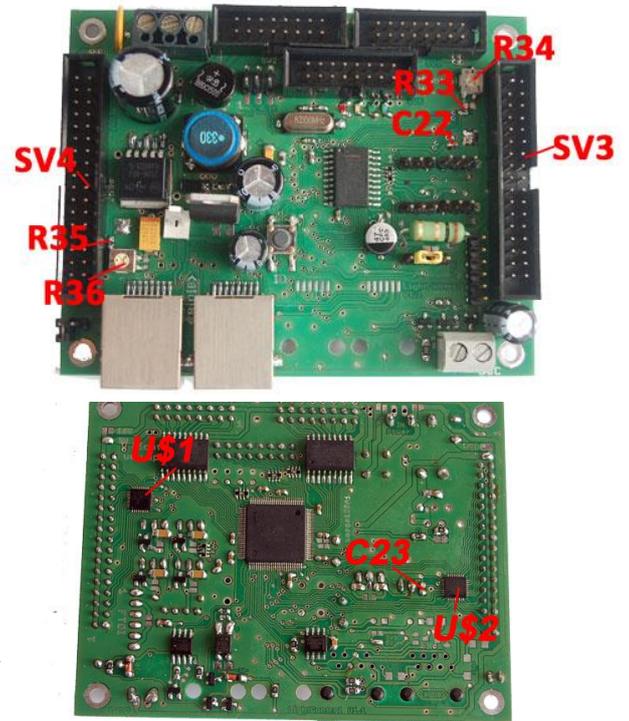
Diese 32 LED-Ausgänge können für Signalbilder sowie auch für Beleuchtungseffekte verwendet werden. Zuallererst beginnen wir mit den Bestücken der beiden TLC's 5941 U\$1 und U\$2.

Das verlöten eines QFN-Gehäuse verlangt ein geschicktes und ruhiges Händchen.

(vergewissern Sie sich der richtigen Einbaurichtung, Punkt auf dem Bauteil und 1 auf der Platine)

Am einfachsten kann man ein QFN-Gehäuse mit einem passenden Halter am Heißluftfön verlöten. Ich möchte aber Ihnen auch einen Weg zeigen mit einem normalen SMD-LötKolben:

1. Platinen-Päds und Baustein-Päds mit Flussmittel bestreichen.
2. Mit dem LötKolben etwas Lötzinn auf die Platine u. Baustein auftragen (Päds müssen eben bleiben).
3. Bauteil platzieren und eventuell fixieren mit einem TesaStreifen.
4. Nochmals die Einbaurichtung und richtige Position prüfen!
5. Das Center-Pad von der Rückseite verlöten, dabei mit dem Finger (zwischen Finger und Bauteil ein Stück Pappe wegen der Hitze klemmen) gegen das Bauteil drücken und auf der Rückseite genügend Lötzinn in das Center-Pad laufen lassen. Das Bauteil darf kurzzeitig schon mal kräftig heiß werden.
6. TesaStreifen vorsichtig entfernen und das Bauteil sollte jetzt festsitzen.
7. Die Päds kräftig mit Flussmittel tränken und mit dem LötKolben und etwas Lötzinn von rechts nach links über die Päds ziehen.
8. optische Kontrolle auf Kurzschlüsse unter einer Lupe
9. ohmsche Kontrolle auf Kontaktsicherheit. Man sollte zwischen einen Port und einem anderen Port einen MOhm-Wert messen können.



#### kleiner Tipp:

Auf [fichtelbahn.de](http://fichtelbahn.de) gibt es unter LightControl / Wissenswertes einen kleinen Videoclip indem man das Handling sehen kann.

Jetzt werden noch die beiden 100nF SMD Kondensatoren C22 und C23 bestückt und die beiden SMD Widerständen R33 und R35.

Mit den beiden Potis R34 und R36 kann man die Grundhelligkeit des jeweiligen Treibers verändern (Stromquelle).

Zuletzt noch den Sockel SV3 und SV4 und fertig sind die LED-Ausgänge.

Neben den 34pin Wannensteckern gibt es noch die Lötbrücken SJ10, SJ11, SJ12 und SJ13, damit kann man die LED-Versorgungsspannung festlegen.

Wir empfehlen für eine Versorgung **ohne Vorwiderstand die 5V Spannung.**

Bei mehreren Leuchtdioden in Reihe pro Ausgang benötigt man die 18V Variante.

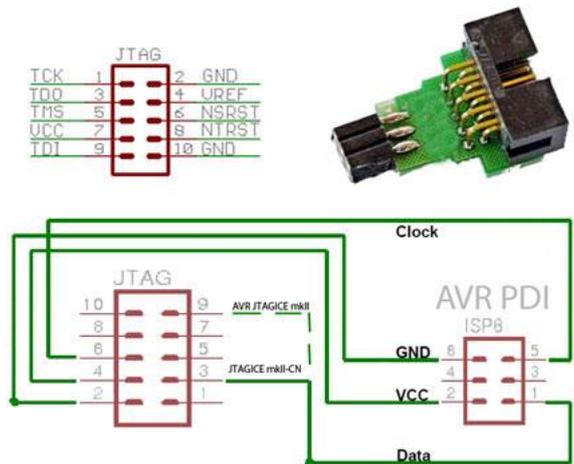
Hier sollte der LED-Treiber mit Vorwiderständen betrieben werden.

## 4. Firmware aufspielen / flashen

Die ATXmega werden mittels PDI bespielt, das ist eine Zweidraht-Schnittstelle. Der bisher übliche SPI-Adapter (wie z.B. ponyprog) kann nicht verwendet werden.

### Verwendet werden kann:

- **AVRISPmkII:** Hierzu ist eine aktuelle Version von AVR Studio erforderlich, vor Benutzung unbedingt im Programmiermenu die Firmware des AVRISP updaten.
- **STK600:** Bei STK600 wird vom 6-poligen blauen PDI-Stecker eine 1:1 Verbindung auf das Board gezogen. Wichtig: Am STK600 muss der VTARGET-Jumper geöffnet werden!
- **JTAGICE mkII und JTAGICE mkII-CN**  
Bei dem AVR JTAGICE mkII ist die Data (PDI) mit dem JTAG Pin 9 zu verbinden.  
Bei dem JTAGICE mkII-CN (Clone) ist die Data (PDI) mit dem JTAG Pin 3 zu verbinden.



### Der DIAMEX ALL AVR:



Der **DIAMEX ALL AVR** ist ein sehr preiswerter Programmierer für das komplette BiDiB-Projekt. Über die PDI-Schnittstelle können all unsere ATX-Prozessoren programmiert werden.

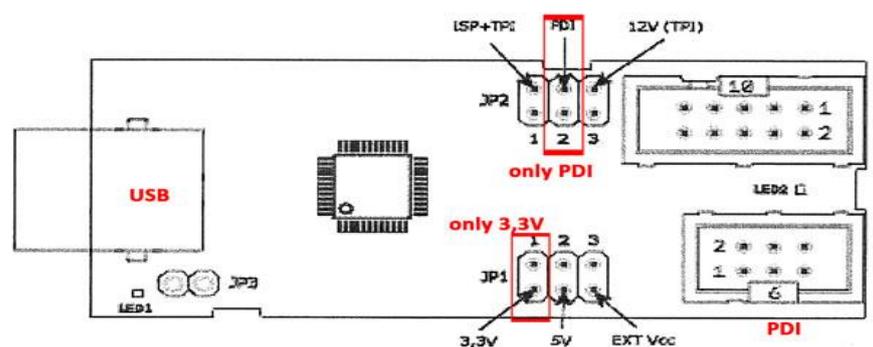
Bezugsquelle Reichelt - [Artikel-Nr.: DIAMEX ALL AVR](#)  
(ist eine Empfehlung wert)

Der Programmierer ist genial, weil er viele Anwendungsbereiche abdecken kann und auch die programmierende Baugruppe mit Spannung versorgen könnte. Deshalb ist vor der Programmierung absolut wichtig, dass die Jumperstellungen überprüft werden um eine Zerstörung des Programmers und Prozessors zu vermeiden.

**Es darf nur ein Jumper auf PDI und 3,3V stecken.** Eine externe VCC Versorgung ist nicht empfohlen, weil bei Missachtung von Erdpotenzialen (anliegendes DCC Signal) der Programmierer zerstört wird, **deshalb den Jumper EXT VCC nicht stecken.**

### Wichtig:

**Den Jumper nicht auf 5V stecken, der Prozessor würde diese Betriebsspannung nicht überleben, seine Betriebsspannung liegt bei 3,3V.**



### Hinweis zur Stromversorgung während des Programmierens mit dem PDI Adapter:

Der LightControl-Baustein benötigt zum flashen seine eigene 9V–18V Spannungsversorgung und wird nicht über den Programmierer gespeist.

### Zur Firmware:

Für alle Baugruppen, welche BiDiB unterstützen, ist die Vergabe einer eindeutigen Produktkennung zwingend erforderlich anhand einer **Seriennummer**.

(siehe das erste Kapitel Einleitung / BiDiB-Seriennummer)

Diese Seriennummer kann ohne großen Aufwand auf folgender Seite generiert werden:

**OpenDCC - BiDiB Seriennummer**

Um eine Seriennummer zu erstellen oder einzusehen, benötigen Sie einen gültigen Zugang zum Forum. Bitte melden Sie sich nun mit den Anmeldedaten des Forums an.

Username:

Password:

[http://www.opendcc.de/elektronik/bidib/opendcc\\_bidib.html](http://www.opendcc.de/elektronik/bidib/opendcc_bidib.html)

Um den Generator benutzen zu können, ist eine Anmeldung erforderlich. Hierbei ist die Benutzerkennung und das Passwort vom OpenDCC-Forum zu verwenden.

Laden Sie die aktuelle Firmware für die LightControl von unserer Webseite und entpacken Sie das ZIP-File auf Ihren Rechner.



Verwenden Sie bei einem Update nur die Datei mit der Bezeichnung: **lightcontrol\_X\_update\_X.000.hex**.

Damit bleiben alle Makros und Einstellungen, die Sie auf der LightControl konfiguriert haben erhalten und werden nicht von neuen Defaultwerten überschrieben. **Sie müssen auch nur dieses Update-File auf die LightControl mit dem BiDiB-Wizard Tool übertragen.**

Die vier anderen Files sind zwei unterschiedliche Softwarepakete für die **Erstinbetriebnahme bzw. für einen Firmware-Wechsel**. Bei Verwendung dieser Dateien werden Ihre gespeicherten Einstellungen mit den Default-Werten überschrieben (Werkseinstellungen). Hier müssen beide Files (\*.000.hex und \*.001.hex) überspielt werden.



In Anleitung „Inbetriebnahme der LightControl“ im Kapitel 5.4 in der ist der Unterschied zwischen den beiden Softwarepaketen „Standard“ und „Signals“ erklärt.

(Stichwort: unterschiedliche Makroanzahl)

## Zum Bootloader:

Der Bootloader ist ein Bestandteil der LightControl-Firmware und kann nicht weggelassen werden. Der Bootloader sorgt im Betrieb dafür, dass die Firmware Makros oder andere Einstellungen speichern kann. Die LightControl verfügt aber auch über einen doppelten Bootloader (FTDI oder BiDiB). Dieser wird aktiviert, wenn während des Einschaltens der LightControl der Identify-Taster (ID) gedrückt gehalten wird. Die Firmware startet dann den Bootloader, wobei der Demo-Jumper festlegt, welche Schnittstelle Verwendung findet:

- Demo-Jumper gesteckt, Boot über die FTDI-Schnittstelle:**  
 Hier findet die gleiche Schnittstelle wie für das Debuginterface Verwendung, allerdings ist die Baudrate auf 19200 (8N1) eingestellt. Der Boot über die FTDI-Schnittstelle wird aktiviert, wenn man vor dem Einschalten der LightControl den Programmierertaster drückt und gedrückt hält. Wenn man die Taste wieder loslöst, meldet sich der Bootloader mit 'LC\_Bootloader V?.??' über die FTDI-Schnittstelle. Nun kann man Befehl an den Bootloader senden, wobei jede Eingabe mit <cr> (Enter-Taste) abgeschlossen wird.  
*... weitere Infos auf [opendcc.de](http://opendcc.de) oder in der Anleitung "Wie erstelle ich ein Makro"*
- Demo-Jumper offen, Boot über die BiDiBus-Schnittstelle:**  
 Der Bootloader meldet sich selbständig am BiDiBus an und erwartet ein Firmware-Update. Dieses kann dann über die BiDiBus-Schnittstelle mit dem **BiDiB-Wizard Tool** durchgeführt werden.  
*... weitere Infos unter [Programmierung Variante 2](#)*

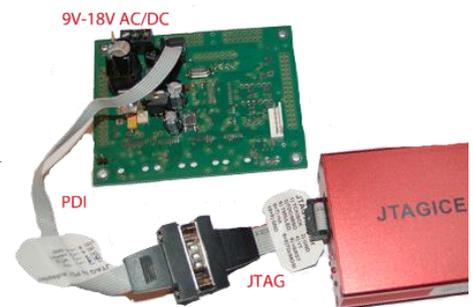
## Die Vorgehensweise zur Prozessor Programmierung:

### Variante 1

Der **Bootloader** und die **Firmware** von der LightControl wird mit dem Programmer auf die LightControl übertragen.

#### Schritt 1:

Verbinden Sie Ihre LightControl über die **PDI-Schnittstelle** mit dem Programmer und öffnen Sie die Programmiersoftware. In unserem Beispiel ist dies das AVR Studio 4.19.



#### Schritt 2:

Als erstes müssen Sie das Device auswählen "ATXmega128A1" und den Programmier Modus "PDI" wählen. Mit klick auf Read Signatur wird vom µC auf der LightControl eine Kennung ausgelesen und in dem Feld angezeigt. Dies ist ein wichtiger Hinweis für eine fehlerfreie Kommunikation. Kann die Signatur nicht ausgelesen werden, müssen Sie nochmals Ihre Verbindung zum µC (Adapterkabel), ihre Stiftkontakte auf der LightControl Platine überprüfen.

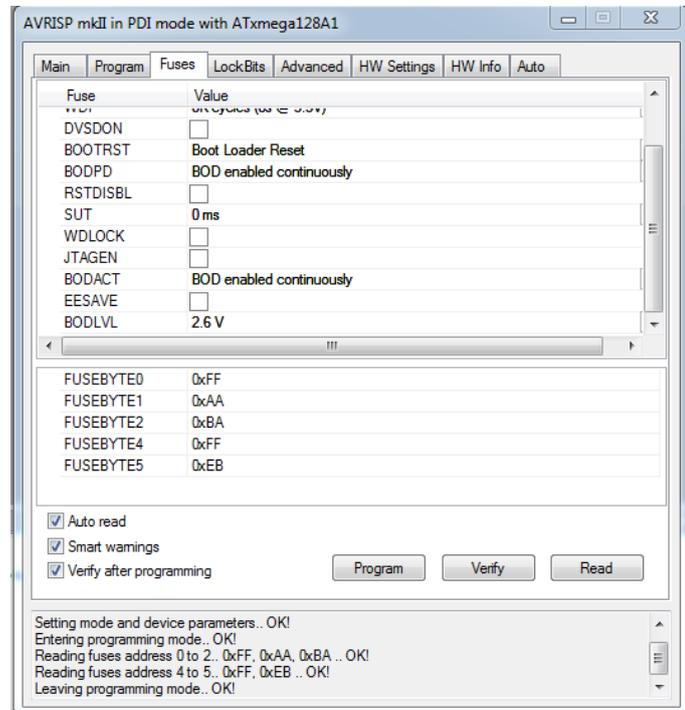
**Hinweis:** Die LightControl benötigt Ihre eigene Spannungsversorgung und wird nicht über den Programmer versorgt.



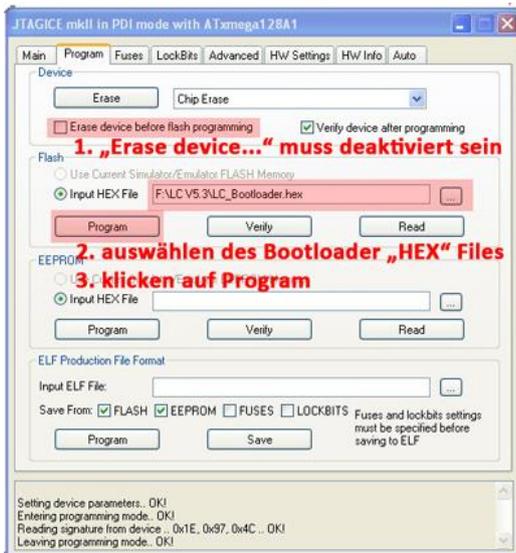
### Schritt 3:

Bevor ein Bootloader oder eine Firmware geflasht werden kann muss der ATXmega die richtigen FUSE Einstellungen bekommen. Dazu gehen Sie auf den Punkt FUSES und geben für die FUSEBYTE 0 bis 5 folgende Werte ein:

- FuseByte 0: 0xFF**
- FuseByte 1: 0xAA**
- FuseByte 2: 0xBA**
- FuseByte 4: 0xFF**
- FuseByte 5: 0xEB**



### Schritt 4:



Das Kästchen "Erase device before flash programming" muss deaktiviert sein sonst wird später beim laden der Firmware der bestehende Bootloader gelöscht. Jetzt wählen Sie das Bootloader-HEX File aus Ihrem Downloadordner aus und mit Klick auf "Program" wird der Bootloader auf die LightControl übertragen.

### Kontrolle:

Sie können den Programmiervorgang abbrechen und mit einem Neustart der LightControl und anschließenden drücken auf den Taster S1 wird der Bootloader gestartet. Dieser findet keine Firmware und bleibt je nach DEMO-Jumper im Boot-Modus für BiDiB oder FTDI stehen. Sein Zustand wird durch ein leuchten der roten LED signalisiert.



### Schritt 5:

Nun kann die eigentliche Firmware auf die LightControl übertragen werden. Dazu wählen Sie im Punkt FLASH das dazugehörige Flash-File und im Punkt EEPROM das eeprom-File. Mit einen **Klick auf Program vom Flash** und einen **Klick auf Program vom eeprom** wird die Firmware auf die LightControl übertragen.

### Kontrolle:

Falls Sie das eeprom vergessen haben zu übertragen, wird das die LightControl mit einem blinken aller Status-LEDs melden.



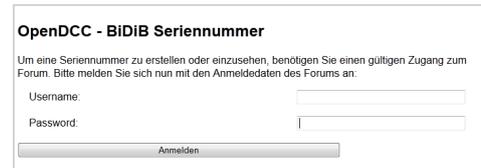
Das blinken der beiden mittleren LED's (gelb und grün) signalisiert Ihnen, dass die Seriennummer noch fehlt. Mit dieser Seriennummer meldet sich die LightControl am BiDiB-Knoten an und somit ist die LightControl eindeutig im Netz zu finden. Diese Seriennummer ist vergleichbar mit einer MAC-Adresse eines Computernetzwerkes und darf im kompletten Netz nur einmalig auftreten.



### Schritt 6:

Als letzten Schritt generieren wir uns eine kostenlose Seriennummer mit dem BiDiB Seriennummer Tool auf [http://www.opendcc.de/elektronik/bidib/opendcc\\_bidib.html](http://www.opendcc.de/elektronik/bidib/opendcc_bidib.html)

Dazu ist eine Registrierung im Opendcc Forum notwendig. Mit Auswahl des passenden Produkts und einer Bemerkung kann das Seriennummer File generiert werden. Diese Datei kann gespeichert werden auf Ihrem PC mit klick auf das gefärbte Kürzel **.eep**.



OpenDCC - BiDiB Seriennummer

Um eine Seriennummer zu erstellen oder einzusehen, benötigen Sie einen gültigen Zugang zum Forum. Bitte melden Sie sich nun mit den Anmeldeinformationen des Forums an:

Username:

Password:

Anmelden

### Schritt 7:

Die Seriennummer ist ebenfalls ein eeprom-File das im Punkt EEPROM ausgewählt werden muss und auf die LightControl übertragen wird mit **Klick auf Programm**.

Die erfolgreiche Programmierung wird auf der LightControl nach dem trennen des Programmers mit einem leuchten der ersten grünen LED signalisiert.

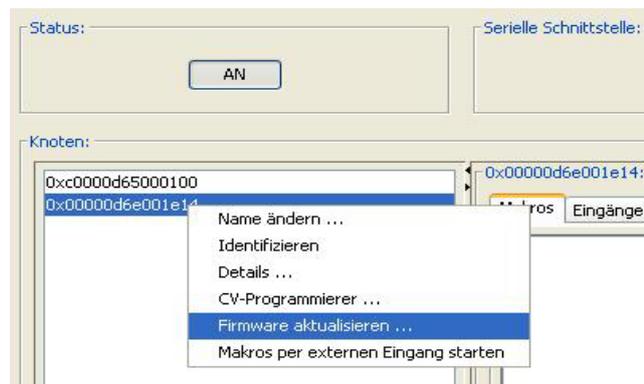


## Variante 2 / Firmware Update

Die LightControl kann auch direkt über das **BiDiB-Wizard Tool** geflasht werden. Diese Funktion kommt auch bei einem späteren **Firmware Update** zum Tragen.

### Der Vorteil ist:

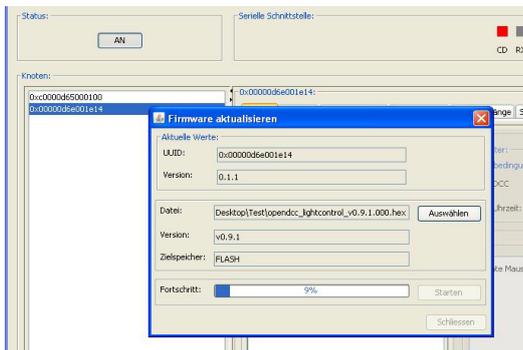
Die LightControl kann im eingebauten Zustand über das Tool aktualisiert werden, man spricht auch von einer Fernwartung. In der Knotenliste von dem BiDiB-Wizard Tool finden Sie Ihre LightControl und alle weiteren BiDiB-Komponenten, für eine Konfiguration oder einem Firmware Update.



**Wenn Sie zum ersten Mal** die LightControl flashen, benötigt der  $\mu$ C noch einen Bootloader. Diesen müssen Sie mit einem Programmer aufspielen (*siehe Variante1 Schritt 1 bis Schritt 4*), oder Sie haben schon einen ATXmega erworben mit aufgespieltem Bootloader im Fichtelbahn-Shop. Im letzteren Fall ist kein Programmer notwendig.

Die notwendige Firmware und Seriennummer wird mit dem **BiDiB-Wizard Tool** übertragen. Die aufzuspielenden Firmware - Dateien müssen in folgender Schreibweise vorliegen:

lightcontrol.hex	[FLASH]	-->	lightcontrol_v*.*.*.000.hex
lightcontrol.eep	[EEProm]	-->	lightcontrol_v*.*.*.001.hex
_0207fe*****.eep	[S/N]	-->	_0207fe*****_serial.001.hex



#### Schritt 5:

Die LightControl wird an die Versorgungsspannung gelegt und mit dem BiDiBus verbunden.

Bei erstmaligem flashen sind alle Status-LED's aus.

#### Schritt 6:

Das **BiDiB-Wizard Tool** wird geöffnet und in der Knotenliste sollte die Seriennummer des Interfaces sichtbar sein. Bei drücken des Tasters auf der LightControl erleuchtet die rote Programmier-LED und im Tool erscheint eine neue Nummer ... **die LightControl**.

#### Ergänzung:

Kommt es zu keiner Erkennung im BiDiB-Wizard Tool, dann mit gedrücktem Taster die Versorgungsspannung anlegen.

#### Schritt 7:

Mit der rechten Maustaste auf die Seriennummer der neu erschienen LightControl klicken und im Kontextmenü die Aktion "Firmware aktualisieren" auswählen.

#### Schritt 8:

Im letzten Schritt muss der Ordner ausgewählt werden mit den drei Firmware Dateien (siehe oben). Diese Dateien müssen in der Reihenfolge **opendcc...000.hex**, **opendcc...001.hex** und **\_0207fe...001.hex** übertragen werden. Mit schließen des Fensters ist die LightControl auf die neue Firmware aktualisiert und für die Inbetriebnahme bereit.

## 5. Statusanzeigen

### Demo-Jumper:



Die LightControl hat einen **SELF TEST** oder **DEMO Mode**, mit dessen Aktivierung alle Ports ein Lauflicht ausgeben. Mit dieser Funktion kann nach dem erfolgreichen Aufbau, alle Ausgänge der LightControl auf Ihre Funktion getestet werden.

#### Vorgehensweise:

Legen Sie an die LightControl eine Versorgungsspannung und verbinden Sie die LED und Power Ausgänge mit einem Leuchtmittel. Mit schließen des Jumpers "DEMO Jumper" wird unabhängig von den gespeicherten Makros ein Lauflicht an den Power- und LED-Ausgängen ausgegeben. Somit können Sie fehlerhafte Ports erkennen.

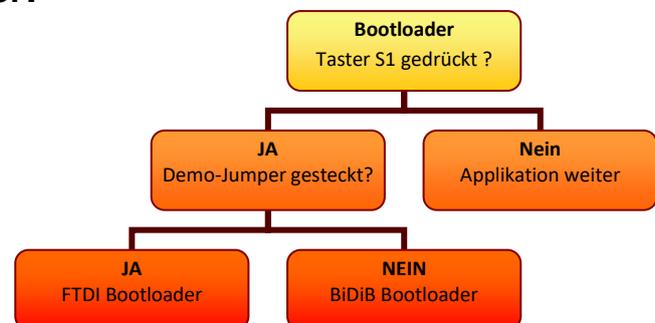
#### weiteres EXTRA:

Wenn Sie einen der acht Eingänge schließen, leuchtet unabhängig vom Lauflicht einer der Power-Ausgänge dauerhaft bis zum wieder öffnen des Eingangs. Somit kann auch die Funktionalität der Eingänge überprüft werden.

#### Wichtig:

Vergessen Sie nicht den "Demo-Jumper" nach dem Test zuziehen, bevor Sie ein Makro auf die LightControl übertragen, sonst wird das DEMO-Mode Makro gelöscht.

### Tastenkombination für Bootloader:



### Fehlercodes der Statusleds:



rote LED leuchtet

LC befindet sich im **Bootloader Modus**

alle LED's blinken

LC hat kein **EEPROM-File** geladen

gelbe und grüne LED blinken

LC hat keine **Seriennummer** geladen

erste grüne LED flimmert

LC ist **betriebsbereit**

erste grüne LED flimmert, zweite grüne LED leuchtet

LC ist **betriebsbereit** und mit **BiDiB** verbunden

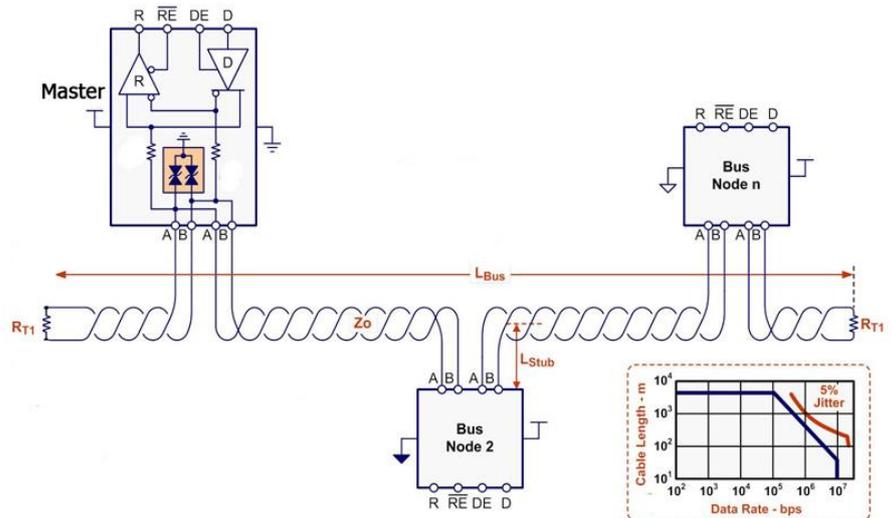
## 6. Terminierung des BiDiBus

Der **BiDiBus** besteht aus einer RS485-2 Draht Verbindung die speziell für Hochgeschwindigkeits-Datenübertragungen über große Entfernungen entwickelt worden ist und eine zunehmende Verbreitung in industriellen Anwendungen findet. Dank diesen Eigenschaften kann eine Kabellänge von über 500 Meter mit hohen Datenübertragungsraten realisiert werden.

Um bei diesen hohen Übertragungsgeschwindigkeiten und Kabellängen noch eine fehlerfreie Kommunikation zu gewährleisten ist eine Terminierung des BiDiBusses notwendig um Reflexionen zu verhindern.

Man spricht auch davon, dass man den Kommunikationsbus abschließen muss.

(RT1 in der Abbildung)



Bei kurzen Kabellängen unter 5m kann auf die Terminierung verzichtet werden.

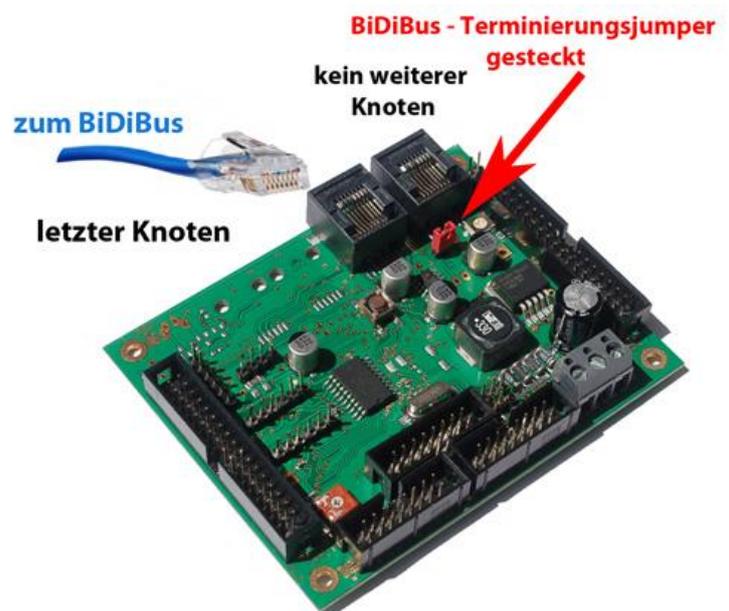
### 6.1 Vorgehensweise der Terminierung

Der BiDiBus muss an jedem Ende (vorne und hinten) mit einem Widerstand mit dem Wert von 120 Ohm abgeschlossen werden. Das bedeutet, dass jeder Knoten der am Ende des BiDiBusses seinen Platz findet (auch bei einem verzweigten Bussystem), ein Widerstand zwischen **BiDiB\_A** und **BiDiB\_B** erhalten sollte.

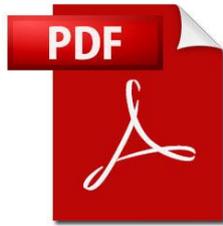
Auf allen unseren BiDiB-Knoten (GBM, LightControl, BiDiBone, s88-BiDiB-Interface....) haben wir das schon vorgesehen und somit für die Anwender vereinfacht. Der Anwender muss nur am letzten Knoten im Bus den Terminierungsjumper stecken.

#### Hinweis:

Der Jumper darf nur am letzten BiDiB-Knoten im BiDiBus (Anfang und Ende der kompletten Busleitung) gesteckt werden. Besteht der BiDiBus aus einem verzweigten Bus (Stichleitungen) mit Hilfe von BiDiBus Erweiterungsmodulen (Hubs), dann muss der Bus an allen offenen Enden abgeschlossen werden.



## 7. Hinweis



Die weiterführende Anleitung "**Inbetriebnahme und Anwendung**" beschäftigt sich mit folgenden Hauptthemen:

- Inbetriebnahme
- Funktionsweise der Makros
- Konfigurationstool "BiDiB-Wizard Tool"
- Debugschnittstelle
- Einsatz in Verbindung mit einer Modellbahnsoftware
- Firmware Update

Für Verbesserungsvorschläge und Hinweise auf Fehler bin ich sehr dankbar.

Auf die Bauanleitung bzw. irgendwelcher Software gibt es keine Haftung für irgendwelche Schäden oder Funktionsgarantie. Ich hafte nicht für Schäden, die der Anwender oder Dritte durch die Verwendung der Software oder Hardware verursachen oder erleiden. In keinem Fall hafte ich für entgangenen Umsatz oder Gewinn oder sonstige Vermögensschäden die bei der Verwendung oder durch die Verwendung dieser Programme oder Anleitungen entstehen können.

**Bei Rückfragen steht Ihnen unser Support-Forum gerne zur Verfügung!**  
(<http://forum.opendcc.de>)

## Kontakt:

**FichtelBahn**  
Christoph Schörner  
Am Dummersberg 26  
D-91220 Schnaittach

support@fichtelbahn.de



© 2013 Fichtelbahn

Alle Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung vorbehalten.  
Vervielfältigungen und Reproduktionen in jeglicher Form bedürfen der schriftlichen Genehmigung durch Fichtelbahn.  
Technische Änderungen vorbehalten.