

Beleuchtungsplatine mit DCC-Decoder für die Y-Wagen der Fa. Kühn



© Desastro

Aus der Hobbywerkstatt

Letzte Änderung: 05.06.2011

Inhaltsverzeichnis

1. Was gibt's hier zu basteln?	Seite 2
2. Benötigte Bauteile	Seite 3
3. Die Bestückung – Allgemeines	Seite 4
3.1. Bestückung der Platine	Seite 5
3.1.1 Bestückung der LEDs	Seite 6
3.2. Zuordnung der LEDs über Lötjumper	Seite 7
4. Programmierung des Controllers	
4.1. Programmieradapter und Software	Seite 10
4.2. Einspielen des Bootloaders	Seite 11
4.3. Einstellung der Fuses	Seite 12
4.4. Einspielen der Firmware	Seite 13
5. Einbau der Platinen	Seite 14
6. Liste der Configuration Variables (CV)	Seite 17
7. Schaltpläne	Seite 18

1. Was gibt's hier zu basteln?

Hierbei handelt es sich um eine Bauanleitung für eine Beleuchtungsplatine mit integriertem DCC-Decoder für die Y-Wagen der Firma Kühn.

Die Software dazu basiert auf den OpenDecoder2-Files von Wolfgang Kufer, die durch Änderungen und Erweiterungen durch mich, für einen Funktionsdecoder genutzt werden.

Des Weiteren wird ein Bootloader von „Hagen Re“ aus dem Mikrocontroller.net-Forum genutzt, um bei Änderungen der Firmware diese dann einfacher einspielen zu können.

Der integrierte DCC-Decoder hat folgende Eigenschaften:

- Kurze und lange Adressen
- Ausgänge separat dimmbar
- Funktionsmapping F1 – F12
- Richtungsabhängigkeit **aller** Ausgänge einstellbar

Auf den folgenden Seiten geht es nun um die Bestückung der Platine, die Programmierung des Mikrocontrollers mit der Firmware, die Konfiguration der LED-Anordnungen mittels Lötjumper und den Einbau der Platinen in die Wagen.

Wichtiger Hinweis

Ich übernehme keinerlei Haftung durch unsachgemäßen Umgang mit dem genutzten Werkzeug (Lötkolben etc.) oder jeglicher Art von Hardwareschäden dadurch an den Bauteilen!

Und jetzt viel Spaß beim Nachbauen!

2. Benötigte Bauteile

Präfix	Anzahl	Bauteilname	Bauform	Preis/Stk	Anbieter	Artikelnr.	Hinweise
IC1	1	ATTiny44	SO14	2,35 €	Reichelt	ATTINY 45-20SU	
T2;3;4;5	2	PUMH9	SOT 363	0,20 €	HBE-Shop	8738599	
T2;3;4;5	2	BCR 148 S	SOT 363	0,09 €	Conrad	154134 - 62	alternativ zu PUMH9!
T6	1	BSS123	SOT 23	0,05 €	Reichelt	BSS 123 SMD	
D1	1	BAW56	SOT 23	0,04 €	Reichelt	BAW 56 SMD	
D2	1	BAV70	SOT 23	0,04 €	Reichelt	BAV 70 SMD	
D3	1	ZF5,1	MiniMELF	0,05 €	Reichelt	SMD ZF 5,1	
C1	1	1µF Tantal	B	0,09 €	Reichelt	SMD TAN.1,0/35	
C2	1	100nF	0603	0,05 €	Reichelt	X7R-G0603 100N	
R4	1	150 Ohm	0603	0,10 €	Reichelt		
R2	1	560 Ohm	0603	0,10 €	Reichelt		
R3	1	10K	0603	0,10 €	Reichelt		
R1	1	22K	0603	0,10 €	Reichelt		

Achtung: Tagespreise u.a. beim Prozessor

Bei größeren Bestellposten der Widerstände verringert sich der Stückpreis deutlich.

Weiterhin werden benötigt:

- LEDs Bauform 0603 (*siehe unten*)
- Evtl. für die Lötbrücken der LED-Konfiguration 0 Ohm-Widerstände, je nachdem was einem besser gefällt.
- Vorwiderstände für die LEDs (Bauform 0603 oder 0805)

Die im Schaltplan angegebenen Werte für die Vorwiderstände sind nur Richtwerte! Bitte je nach Anzahl der in Reihe geschalteten LEDs den Wert selbst errechnen. -> [LED-Rechner](#)

<u>Links zu den Elektronikversendern:</u> <ul style="list-style-type: none"> • Reichelt • Conrad • HBE-Shop 	<u>Bezugsquellen für die LEDs:</u> <ul style="list-style-type: none"> • ledbaron (EBay) • Modellbau-Schoenwitz (EBay) • Digitalzentrale • MBZ-Weber
--	---

3. Die Bestückung - Allgemeines

Hinweis: Vor dem Bestücken des Mikrocontrollers zuerst den Bootloader und evtl. sogar gleich die passende Firmware einspielen! (siehe: „Programmierung des Controllers“)

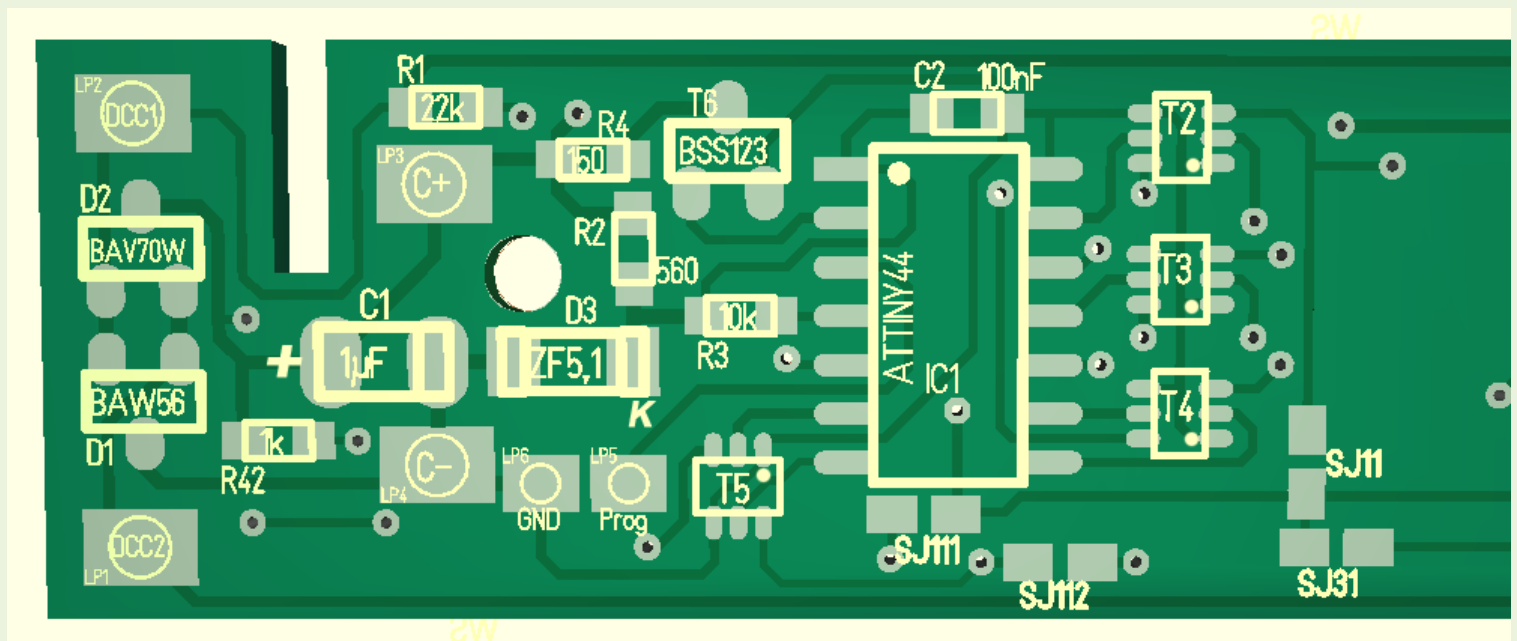
Bei dieser Platine gibt es nur eine Bestückungsvariante im Decoderteil.

Der Prozessor nutzt 8 Ausgänge zum Schalten der LEDs. Die LEDs können einzeln oder in Reihe zu 3 Stück über Lötjumper miteinander und zu einem Ausgang des Prozessors verbunden werden. Frei verfügbar sind 7 Ausgänge, da einer fest für die WC und Gangbeleuchtung genutzt wird. Wenn eine Schlußbeleuchtung, einseitig oder beidseitig gewünscht wird, so entfallen dort auch nochmal 1 oder 2 Ausgänge!

Weitere Hinweise:

- **Auf die korrekte Ausrichtung der Bauteile achten (u.a. Pin1 von T 2,3,4,5 und von IC1)**
- am besten zuerst die Doppeltransistoren T2 bis T5 bestücken, danach den Rest.
- Auch auf die Ausrichtung der Z-Diode D3 achten. Der Ring auf dem Gehäuse kennzeichnet die Kathode!
- Wenn alle Bauteile, außer der Prozessor, aufgelötet sind einen 1.Funktionstest durchführen
 1. Sichtkontrolle auf evtl. Kurzschlüsse durch ungewollte Lötbrücken!
 2. 2 Kabel an die beiden Pads (DCC1 & DCC2 am Rand) der Hauptplatine löten.
 3. die beiden Kabel mit dem DCC-Gleissignal verbinden.
 4. Digitalzentrale einschalten.
 5. Nun sollte an den Lötpins 14 (Masse) und 1 (+) des Prozessors IC1 eine Spannung von etwa 5,1-5,2 Volt zu messen sein!
 6. Jetzt kann man gleich noch die Spannung messen, die nach dem Gleichrichter anliegt (wichtig für die Berechnung der LED-Vorwiderstände). Dazu wieder das Messgerät mit Masse (Pin14 von IC3) und plusseitig von C1 verbinden und den Wert für Später aufschreiben!
 7. wenn alles ok ist, dann weiter mit dem nächsten Punkt...
- den Prozessor als letztes bestücken, nachdem er mit dem Bootloader beschrieben wurde.

3.1. Bestückung der Platine

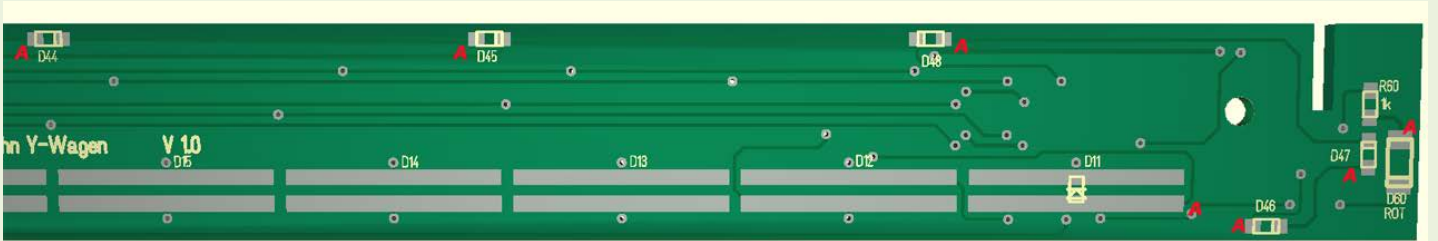
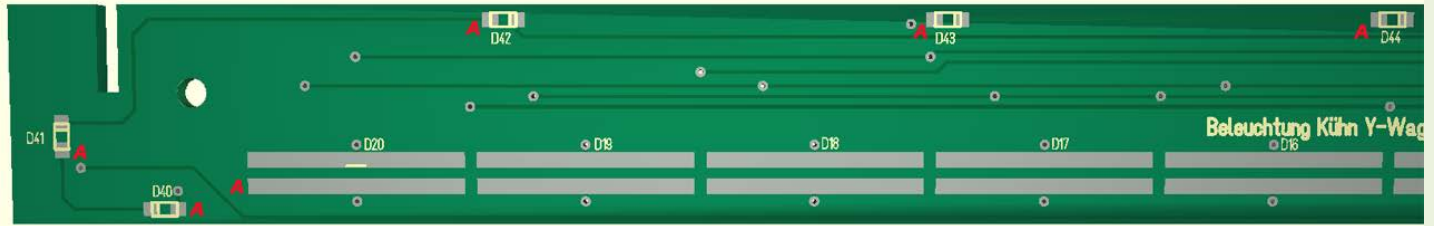


Ansicht Decoderteil

- auf die Polarität von C1 („+“ ist der Strich auf dem Bauteil!) und der Z-Diode D3 (Kathode ist mit einem Strich gekennzeichnet!) achten. Plus von C1 zeigt Richtung linkem Platinenrand, die Kathode von D3 zeigt zum Controller IC1.
- Pin1 der Doppeltransistoren T2 bis T5 ist mit einem winzigen Punkt markiert! Auf die Einbaurichtung achten!

Wenn alles, außer dem Prozessor, bestückt ist, den 1.Funktionstest (siehe Seite 4) durchführen.

3.1.1. Bestückung der LEDs



Ansicht Unterseite der Platine

- Platine so vor sich hinlegen, das die Aussparungen nach oben zeigen
- jetzt die LED's einlöten. Das „A“ auf der Platine kennzeichnet natürlich die Anode. Auf der Rückseite der LED ist eine Kennzeichnung, je nach Hersteller verschieden. Entweder ein **Dreieck** (die breite Seite ist die Anode) oder ein „T“. (dabei ist der obere Balken die Anode)

Die Bilder der Platinenansichten befinden sich nochmals in höherer Auflösung im Ordner *Bilder* des Archivs.

3.2. Zuordnung der LEDs über Lötjumper

Wie im Kapitel zuvor schon erwähnt ist es, durch die begrenzte Anzahl der Ausgänge, nicht möglich allen Abteil-LEDs einen eigenen Ausgang zuzuweisen.

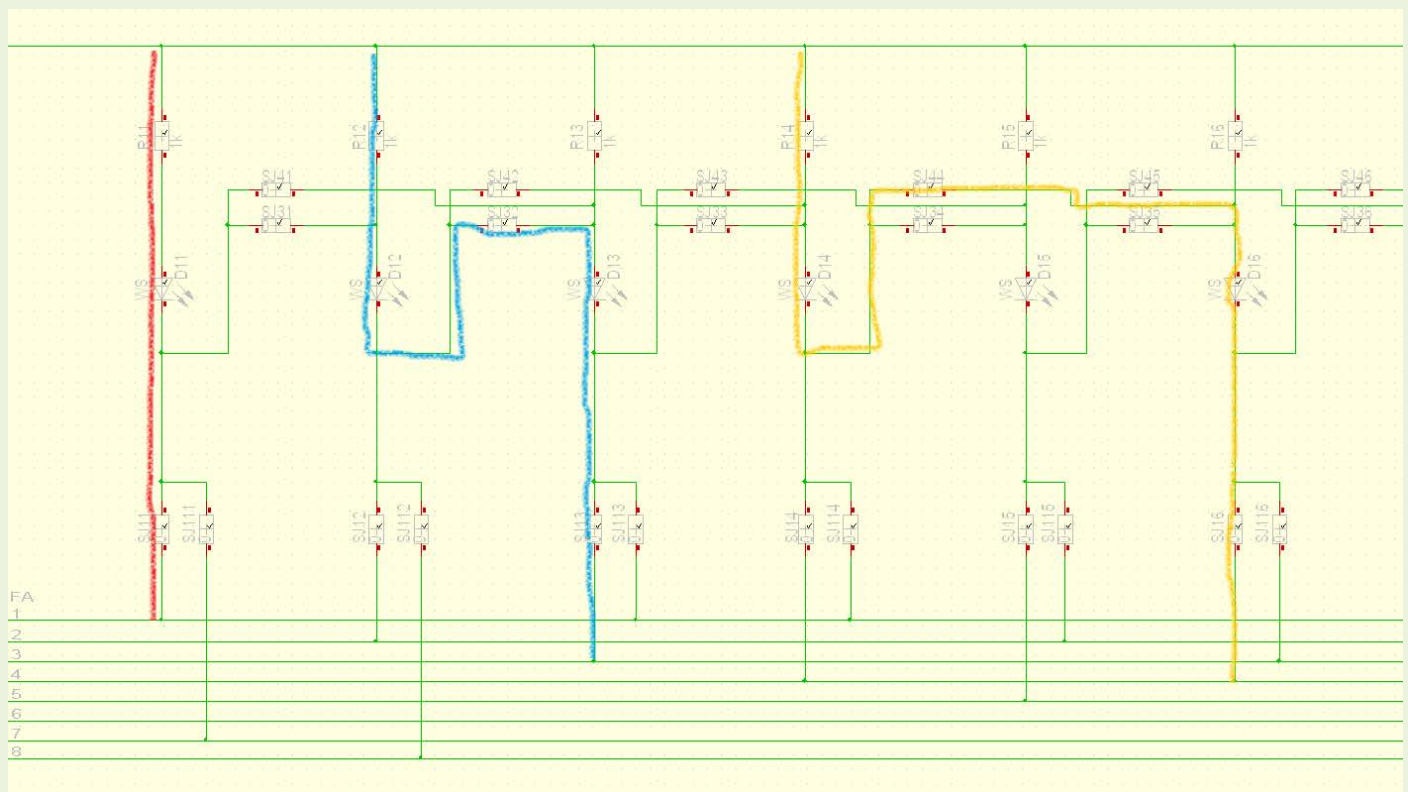
Deshalb gibt es auf der Oberseite der Platine eine Reihe von Lötbrücken (*man kann auch ein 0 Ohm Widerstand 0603 dazu nutzen*), welche zum Verbinden der LEDs zueinander in Reihe und einem Ausgang vorgesehen sind.

Diese Lötbrücken haben ein Nummerierungssystem für den jeweiligen Zweck:

- **SJ 11-20** - Verbindung der Kathode einer LED oder LED-Reihe auf einen Ausgang des μC
- **SJ 111-120** - wie SJ11-20, jedoch ein alternativer Ausgang
- **SJ 31-39** - Verbindung der Kathode einer LED zur Anode der benachbarten LED (Reihenschaltung)
- **SJ 41-48** - wie SJ31-39, nur dabei wird eine LED übersprungen

Es wird immer nur das bestückt (LEDs, Vorwiderstand, Lötbrücke) was auch benötigt wird.

Anbei noch 3 Beispiele für die Konfiguration zur Veranschaulichung.



Erläuterung auf der nächsten Seite...

Das Beispielbild befindet sich auch nochmal in Originalgröße im Ordner *Bilder (LED-Konfig-Beispiele.png)*.

Beispiel 1 (rote Linie)

Es ist nur 1 LED an einen Ausgang des μ C gelegt. Von R11, D11 und SJ11 an FA1 des Controllers. Anstelle SJ11 funktioniert natürlich auch SJ111.

Beispiel 2 (blaue Linie)

Hier sind 2 LEDs in Reihe geschaltet und auf einen Ausgang gelegt. Von R12, D12 über SJ32 auf D13 und SJ13 an FA3 des Controllers. Hier wäre auch SJ113 möglich.

Beispiel 3 (gelbe Linie)

Bei dieser Konfig wird die benachbarte LED „übersprungen“. Von R14, D14 über SJ44 auf D16 und SJ16 an FA4 des Controllers. Alternativ auch SJ116 auf FA3 möglich.

Achtung! Bitte niemals 2 Lötbrücken zu den μ C-Ausgängen **einer** LED, also z.B. SJ11 und SJ111 oder SJ16/SJ116, gleichzeitig verbinden! Durch das Aktivieren der dazugehörigen Ausgänge am Controller, könnte dieser beschädigt werden!

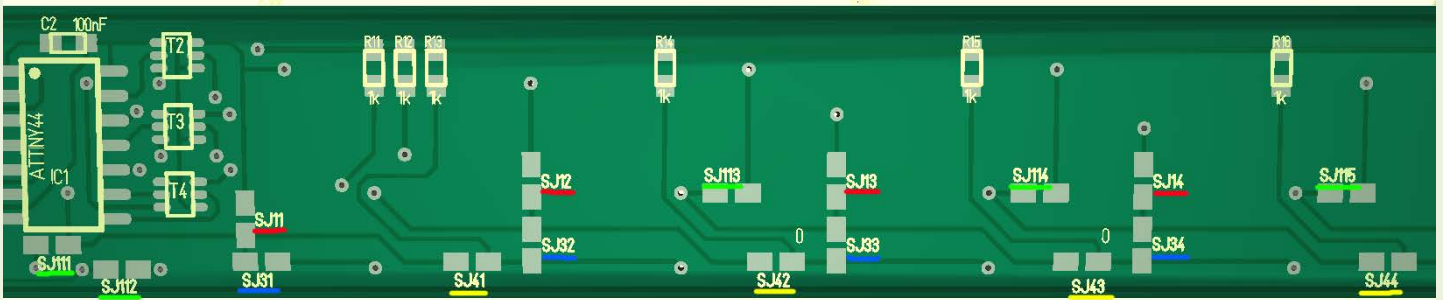
Für die Bestückung der LEDs, Vorwiderstände und Lötbrücken sind im Ordner *Bilder* weiterhin noch die Schaltplanabschnitte und die Bestückungsansichten in großer Darstellung vorhanden.

Es gibt noch viele weitere Möglichkeiten um die Abteil-LEDs miteinander zu verbinden. Aber für alle Varianten die Lötbrücken ins Layout zu bringen ist unmöglich.

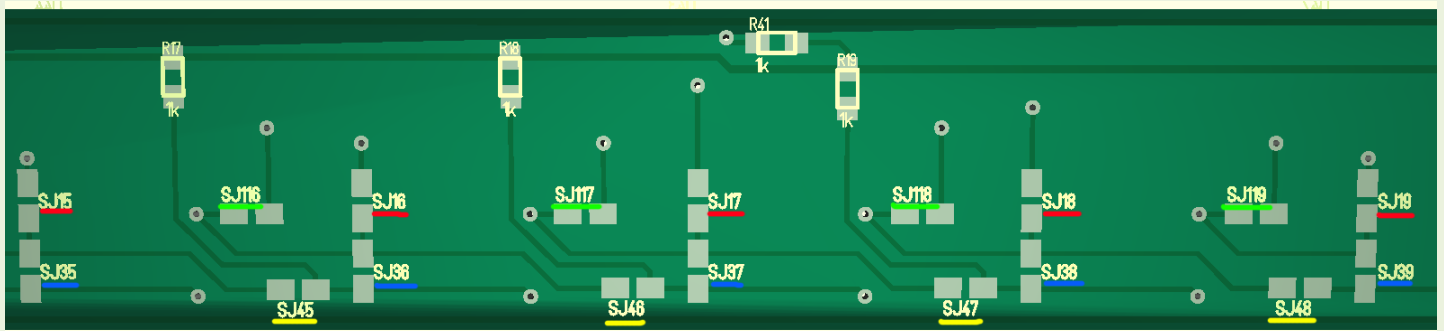
Man kann sich aber, durch die Verwendung von z.B. Kupferlackdraht, diese Möglichkeiten selber erstellen.

Beispiel: Man möchte D11 mit D15 verbinden. Dazu nimmt man ein in der Länge passenden Cu-Lackdraht und lötet ihn einmal an das linke Pad von SJ31 (Decoderteil der Platine zeigt nach links) und die andere Seite an das obere Pad von SJ34.

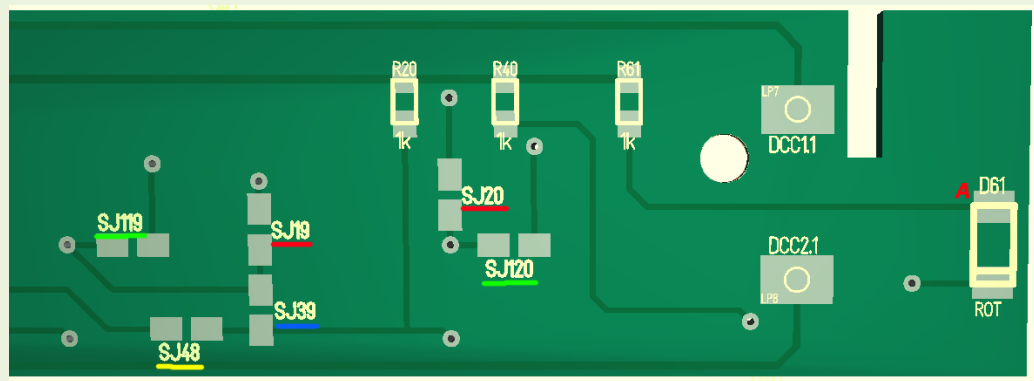
weiter auf der nächsten Seite...



Übersicht Lötbrücken Teil1



Übersicht Lötbrücken Teil2



Übersicht Lötbrücken Teil3

- Die Lötbrücken auf den Bildern sind farbig unterstrichen, analog zu der Beschreibung auf Seite 6!
- D61 ist der Anschluss für ein Schlusslicht, die Anode für die LED ist auf dem Bild markiert, auf der Platine jedoch nicht!

4. Programmierung des Controllers

Ich möchte hier nicht auf die Einzelheiten der grundlegenden Programmierung (das Brennen der Software) in den Controller eingehen, das würde den Umfang der Anleitung sprengen. ☺

Wer schon Erfahrung hat, der weiß wie es funktioniert und welche „Werkzeuge“ man dafür benötigt.

Und wer sich darin gerne einarbeiten möchte, dem kann ich [diese](#) Seite empfehlen. Dort ist alles zum Thema AVR beschrieben, mit weiterführenden Links zu Selbstbau-Programmiergeräten und Software. z.B. PonyProg.

4.1. Programmieradapter und Software

Das Programmieren des Bootloaders in den Controller sollte man vor dem Einlöten erledigen, weil es sonst eine richtige Fummelarbeit ist. Wenn man es selbst machen möchte, muss erst noch ein kleiner Adapter gebaut werden.

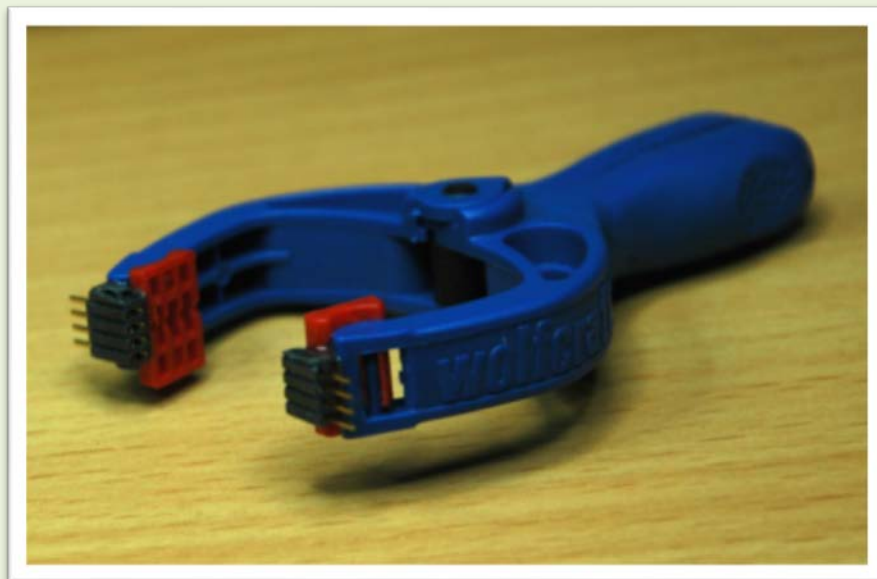
Der Controller hat in der SMD-Version einen „Beinchen-Abstand“ von 1,27mm.

Um nun den Prozessor mit dem Programmiergerät zu verbinden, braucht man einen kleinen Adapter. Dafür nimmt man einfach zwei 10-polige Buchsenleiste 1,27mm und teilt diese in 2 x 7 Pole und verbindet dann die entsprechenden Pins (6 Stück) der 2 Buchsenteile mit dem [ISP-Stecker](#) am Programmiergerät.

Welche Pins das sind erfährt man aus dem Datenblatt des [ATTiny44](#).

Das Bild zeigt nur ein Beispiel, wie man die Buchsenleisten befestigen kann um den Prozessor einfacher zu halten.

Microklemme aus dem Baumarkt; Buchsen mit 2K-Kleber befestigt.



4.2. Einspielen des Bootloaders

Nachdem man sich einen Adapter gebastelt und korrekt verdrahtet hat, kann man mit dem Einspielen des *.hex-Files für den Bootloader beginnen.

Je nach Vorhandensein des entsprechenden Brennprogramms (AVR-Studio, PonyProg o.ä., siehe auch die unter (4.) empfohlene Seite), lädt man nun aus dem Unterordner *AVRootloader/AVR* des mitgelieferten Archivs die Datei *AVRootloader_Tiny44.hex* in den Controller.

Ist diese Aktion erfolgreich verlaufen, muss man nun noch die Fuses des Controllers richtig setzen!

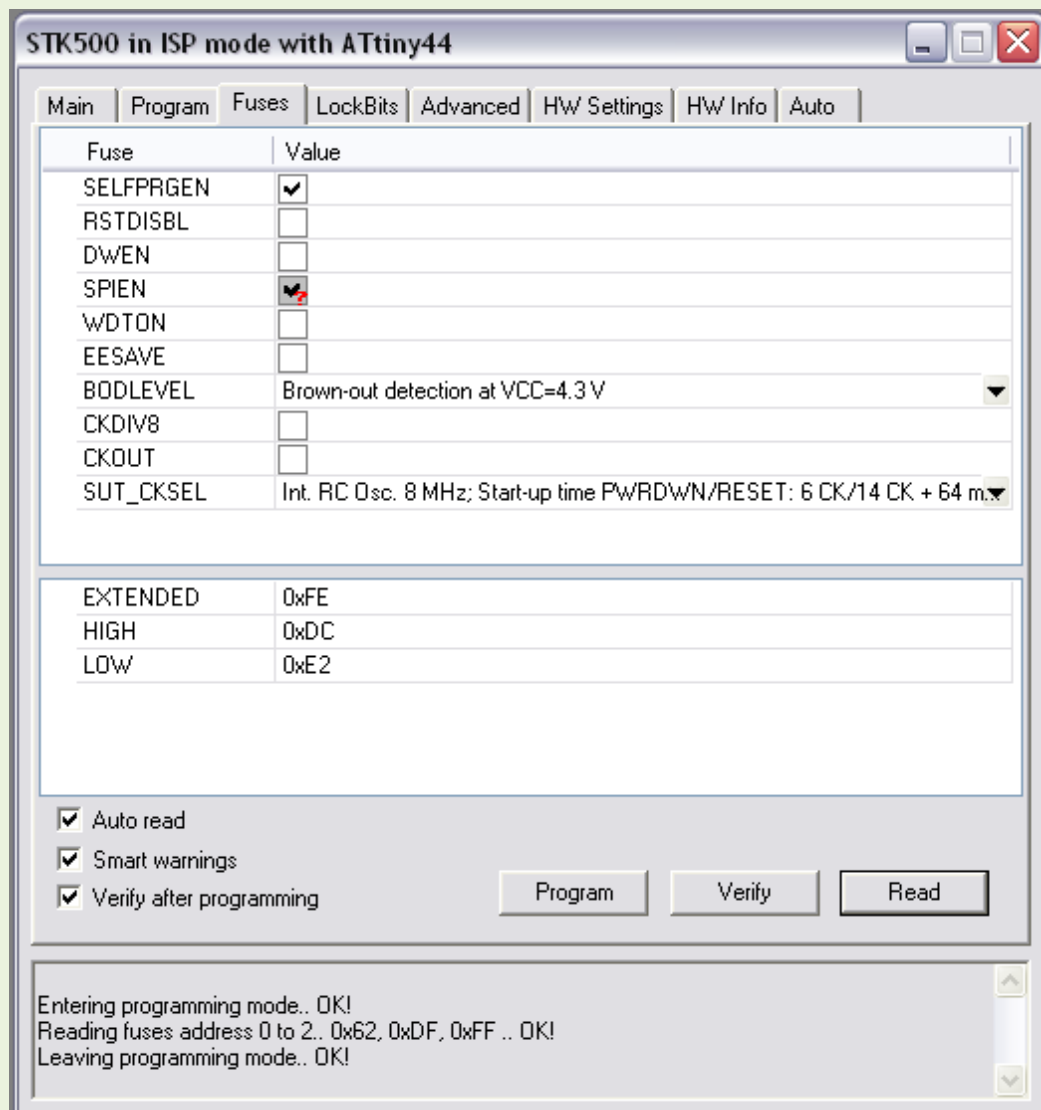
4.3. Einstellung der Fuses

Nun müssen die Fuses (sozusagen die Sicherungsschalter) des Controllers angepasst werden, damit dieser den Bootloader ausführt und später auch das Programm richtig funktioniert.

Das untere Bild zeigt die richtigen Einstellungen für den ATtiny44 mit AVR-Studio. Wer ein anderes Programm zum Brennen nutzt, kann sich der Werte die bei **EXTENDED; HIGH; LOW** stehen, bedienen.

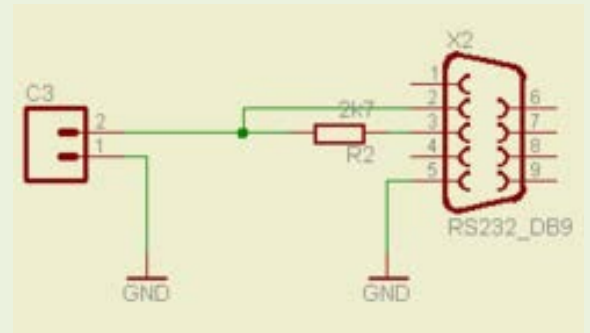
Es gibt noch eine Seite mit einem [Fuses-Calculator](#) die u.U. nützlich sein kann.

Jetzt nur noch mit „Program“ die neuen Einstellungen in den Prozessor übertragen.



4.4. Einspielen der Firmware

Zum Einspielen der eigentlichen Firmware in den Prozessor benötigt man ein PC mit serieller Schnittstelle (inwieweit USB-zu-Serial Adapter funktionieren, muss ausprobiert werden) und einen Adapter, bestehend aus einer 9-poligen Sub-D-Buchse und einem 2,7kΩ Widerstand. Der Aufbau wie rechts auf dem Bild zu sehen. Den Stecker C3 kann man nach Belieben einsetzen oder durch 2 kleine Klemmen ersetzen. Man muss ja irgendwie die 2 Pins am Prozessor kontaktieren können!



Die Hauptplatine schließt man zuerst mit den 2 Kabeln wieder an das DCC-Gleissignal an, die Zentrale aber noch ausgeschaltet oder im „STOP“-Modus belassen!

Jetzt ruft man aus dem Archiv im Unterordner *AVRootloader/Windows* die *AVRootloader.exe* auf. Man macht die Einstellungen wie auf dem unteren Bild zu sehen. In **1.** wird der entsprechende COM-Port gewählt. Unter **2.** und **3.** sucht ihr euch aus dem Archiv im Unterordner *Firmware* die entsprechende Version der **.hex* (Flash) und **.eep* (EEProm) Datei heraus und tragt diese im Abschnitt *Programmingfiles* ein.

Jetzt wird die Masse-Verbindung (GND) von dem kleinen Adapter am Serialport zum Decoder hergestellt. Ein entsprechendes Pad ist auf der Platine vorhanden. Danach die andere Verbindung des Adapters auf das „Prog“-Pad der Platine herstellen. Nun in der Anwendung auf **4.** klicken (Program) und danach die Zentrale einschalten! Nach spätestens 2 Sek. sollte das Protokollfenster des Programms erscheinen.

Wenn nun eine Erfolgsmeldung erscheint, dann wurden die 2 Dateien korrekt in den Controller geschrieben und der Decoder ist ab sofort einsatzbereit.

Sollte irgendwo im Protokoll eine rote Meldung erscheinen, dann ist was schief gelaufen. Dann nochmal alle Verbindungen prüfen, Zentrale auf STOP und von vorn beginnen.

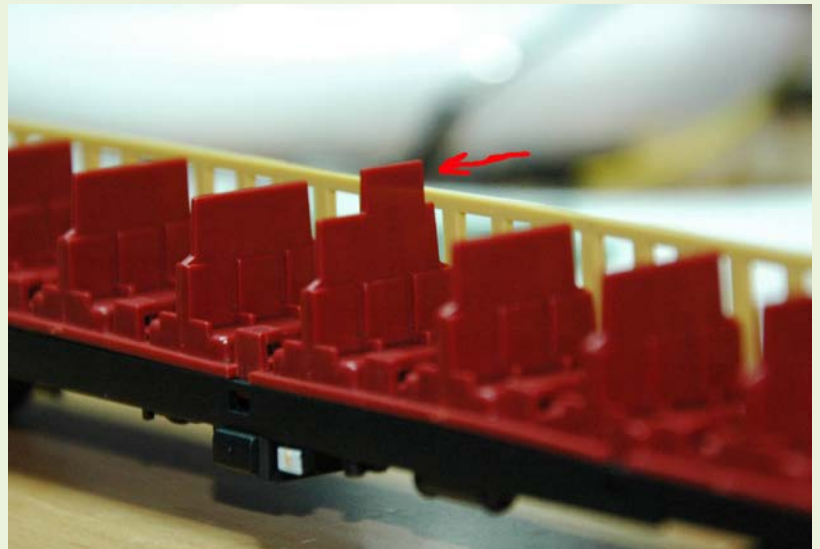


5. Einbau der Platinen

Zuerst muss das Modell des Wagens geöffnet werden, dazu den Hinweisen in der beigelegten Bedienungsanleitung folgen.

Auf den folgenden 2 Bildern ist die Inneneinrichtung der Wagen zu sehen und dort müssen 3 Abstandsstege abgeschnitten oder gekürzt werden.

Das 1. Bild zeigt den breiten Steg in der Mitte der Inneneinrichtung. Dieser wird so gekürzt, dass noch etwa 2mm über der Abteiltrennwand stehen bleibt. Somit hat später die Platine noch eine Auflagefläche.

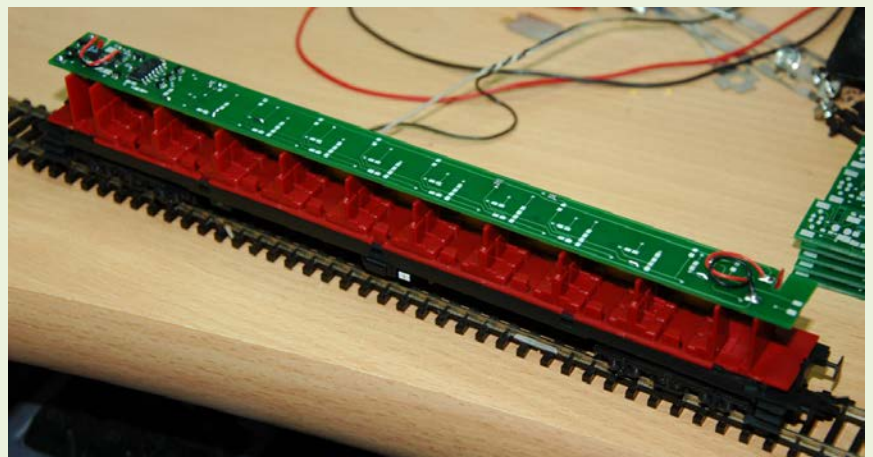


Im 2. Bild ist jeweils an den Enden der Inneneinrichtung der Steg plan abzuschneiden, welcher sich auf der **abgewandten** Seite des Seitengangs befindet!



In die 2 übrig gebliebenen Stege auf der Gangseite wird später die Platine mit den Ausfräsungen eingelegt.

Die Verbindungsleitungen für die Stromabnahme zu den Drehgestellen sind an diesen Wagen schon vorbereitet. Die Platine wird nun so gehalten, dass die 2 Ausfräsungen in Richtung der verbliebenen Stege zeigen. Jetzt fädelt man die Litzen der Stromabnahme durch die 2 Löcher in der Platine und kann dann die Platine auf die Inneneinrichtung auflegen.

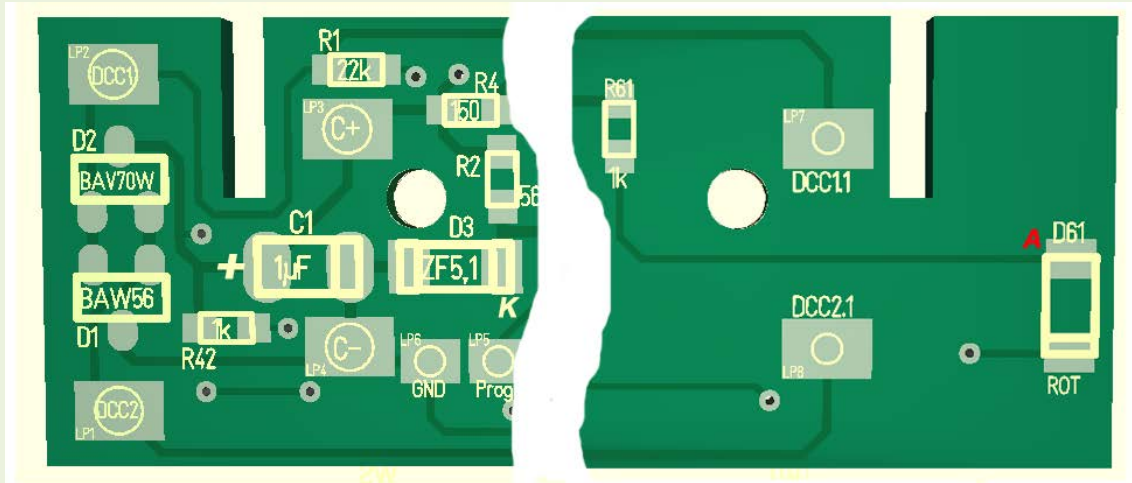


weiter auf der nächsten Seite...

Nun lötet man die Litzen an die dafür vorgesehenen Pads auf der Platine fest. Das wären in dem Fall „DCC1 / DCC1.1“ und „DCC2 / DCC2.1“. Dabei drauf achten, dass jeweils die 2 gleichfarbigen Litzen an die gleich bezeichneten Pads gelötet werden. (siehe nachfolgendes Bild)

z.B.: Rot auf DCC1 und DCC1.1!

Wird das nicht beachtet, gibt das später einen Kurzschluss!



Ansicht der vorhandenen Löt pads auf der Platine (für DCC, Pufferkondensator, Programmierung, 1xSchlußlicht)

- An die Löt pads „C+ & C-“, kann nach Belieben ein Pufferkondensator (Spannungsfestigkeit mind. 25V) angeschlossen werden. Vorhandene Verlängerungskabel können mit durch die Bohrung rechts neben der Ausfräsung geführt werden.
- Zum Einspielen einer neuen Firmware Version sind die Anschlüsse „GND“ und „Prog“ gedacht. Wenn man möchte, kann man diese durch Verlängern auf die Wagenunterseite mit einer kleinen 2-poligen Buchse verlegen.
- Ganz rechts auf der Oberseite der Platine befindet sich noch der Anschluss für ein Schlusslicht. Die Anode ist wie im Bild oben gekennzeichnet.

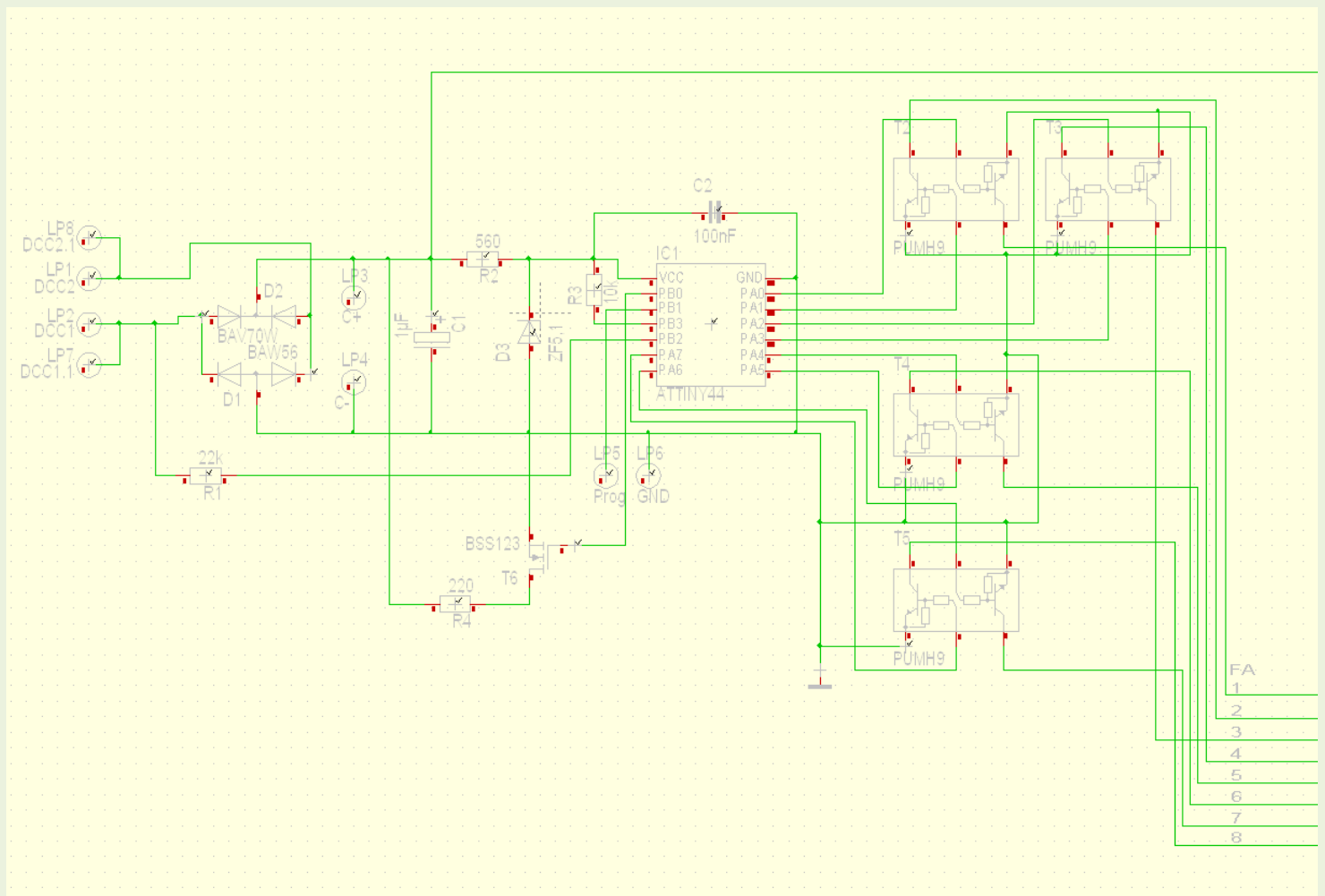
Wenn jetzt alles verbunden und richtig eingesetzt ist, nochmals eine Funktionsprobe durchführen. Läuft alles zur Zufriedenheit, kann der Wagen wieder zusammengesetzt werden.

Damit ist der Zusammenbau beendet!

6. Liste der Configuration Variables (CV)

CV-Liste Funktionsdecoder OD2_F-Dec (Basis Opendecoder2)				
CV-Nr.	Bedeutung	Wertebereich	Grundeinst.	Kommentar
1	Basisadresse	1-127	3	
7	Versionsnummer	-	128	
8	Herstellerident	-	0x0D	
17	erweiterte Adresse, Teil1	192-231	192	
18	erweiterte Adresse, Teil2	0-255	0	
29	Konfiguration Bit 0 Bit 5	0-63	0	Fahrtrichtung (0 = normal ; 1 = invers) Basis/erweiterte Adresse (0 = Basisadr.)
33	Ausgangszuordnung F0 v. Bit 0 = FA1 Bit 1 = FA2 Bit 2 = FA3 Bit 3 = FA4 Bit 4 = FA5 Bit 5 = FA6 Bit 6 = FA7 Bit 7 = FA8	0-128	0	Hinweis: Wert 4 nicht nutzen! Es wird in der Software dadurch der Ausgang 2 am µC angesteuert -> ist aber DCC-IN! FA6 - FA8 nur bei genutztem ATtiny44 dann entfällt auch die oben genannte Werteeinschränkung
34	Ausgangszuordnung F0 r.	0-128	0	Wertebereich wie CV33
35	Ausgangszuordnung F1	0-128	1	Wertebereich wie CV33
36	Ausgangszuordnung F2	0-128	2	Wertebereich wie CV33
37	Ausgangszuordnung F3	0-128	8	Wertebereich wie CV33
38	Ausgangszuordnung F4	0-128	0	Wertebereich wie CV33
39	Ausgangszuordnung F5	0-128	0	Wertebereich wie CV33
40	Ausgangszuordnung F6	0-128	0	Wertebereich wie CV33
41	Ausgangszuordnung F7	0-128	0	Wertebereich wie CV33
42	Ausgangszuordnung F8	0-128	0	Wertebereich wie CV33
43	Ausgangszuordnung F9	0-128	0	Wertebereich wie CV33
44	Ausgangszuordnung F10	0-128	0	Wertebereich wie CV33
45	Ausgangszuordnung F11	0-128	0	Wertebereich wie CV33
46	Ausgangszuordnung F12	0-128	0	Wertebereich wie CV33
47	Dimmwert FA1	1-60	20	61 = kein Dimmen !
48	Dimmwert FA2	1-60	20	
49	Dimmwert FA3	1-60	20	!! Bei Tiny45/85 nicht genutzt -> DCC-IN !!
50	Dimmwert FA4	1-60	20	
51	Dimmwert FA5	1-60	20	!! Bei Tiny45/85 ACK-Puls (wahlweise auch als Ausgang nutzbar !)
52	Dimmwert FA6	1-60	20	
53	Dimmwert FA7	1-60	20	
54	Dimmwert FA8	1-60	20	
57	Effekt_FA1 Bit 0 = 1 - Ausgang reagiert nur bei "vorwärts" Bit 1 = 1 - Ausgang reagiert nur bei "rückwärts"	0-3	0	
58	Effekt_FA2	0-3	0	Funktion wie CV57
59	Effekt_FA3	0-3	0	Funktion wie CV57
60	Effekt_FA4	0-3	0	Funktion wie CV57
61	Effekt_FA5	0-3	0	Funktion wie CV57
62	Effekt_FA6	0-3	0	Funktion wie CV57
63	Effekt_FA7	0-3	0	Funktion wie CV57
64	Effekt_FA8	0-3	0	Funktion wie CV57

7. Schaltpläne



Der Schaltplanteil mit den LED-Zuordnungen befindet sich mit in Ordner *Bilder*.