DCC-Funktionsdecoder OD2_F-Dec V1.0

Bild zeigt den funktionsfähigen Prototypen

Aus der Hobbywerkstatt

Letzte Änderung: 16.12.2010

Inhaltsverzeichnis

1. Was gibt‘s hier zu basteln?

Wie das Deckblatt schon beschreibt, handelt es sich hier um eine Bauanleitung für einen einfachen Funktionsdecoder mit bis zu 5 Ausgängen für das DCC-Format.

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

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.

Was kann das Ding denn nun alles? Die folgende Aufzählung zeigt den aktuellen Stand!

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

Das Platinchen dazu hat eine Größe von etwa 13,4 x 9,9mm.

Auf den folgenden Seiten geht es nun um die Bestückung der Platine und die Programmierung des Mikrocontrollers mit der entsprechenden Firmware.

Wichtiger Hinweis

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

Und jetzt viel Spaß beim Nachbauen!

2. Benötigte Bauteile

Anzahl	Bezeichnung	Bauform	Preis/Stk	verfügbar bei…	Artikelnr.
1	ATTiny45	SO8	1,45€	Reichelt/CSD	ATTINY 45V-20SU
2	PUMH9	SOT363	0,20€	HBE-Shop	8738599
1	BSS123	SOT23	0,05€	Reichelt	BSS 123 SMD
1	BAW56	SOT23	0,04€	Reichelt	BAW 56 SMD
1	BAV70	SOT23	0,04€	Reichelt	BAV 70 SMD
1	ZF5,1	MiniMELF	0,05€	Reichelt	SMD ZF 5,1
1	1µF Tantal	B	0,09€	Reichelt	SMD TAN.1,0/35
1	100nF	0603	0,05€	Reichelt	X7R-G0603 100N
1	220 Ohm	0603	0,10€	Reichelt	
1	560 Ohm	0603	0,10€	Reichelt	
1	10K	0603	0,10€	Reichelt	
1	22K	0603	0,10€	Reichelt	

Achtung: Tagespreise u.a. beim Prozessor

Hinweis: ATTiny45 bei CSD-Electronic ab 10Stk á 1,30€

Links zu den Elektronikversendern:

- Reichelt
- CSD-Electronic
- HBE-Shop

Öffentlicher Reichelt-Warenkorb für die dort verfügbaren Teile.

3. Die Bestückung

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

Es sind mehrere Bestückungsvarianten für den Decoder möglich!

Version 1.	Decoder mit 3 Funktionen und ACK-Puls (SJ1 offen; R4 und Q1 bestückt!)
Version 2.	Decoder mit 4 Funktionen **ohne** ACK-Puls (SJ1 geschlossen; R4, Q1 **nicht** bestückt!)
Version 3.	Decoder mit 4 Funktionen und ACK-Puls (siehe dazu Seite 7.)
Version 4.	Decoder mit 5 Funktionen **ohne** ACK-Puls (siehe dazu Seite 8.)

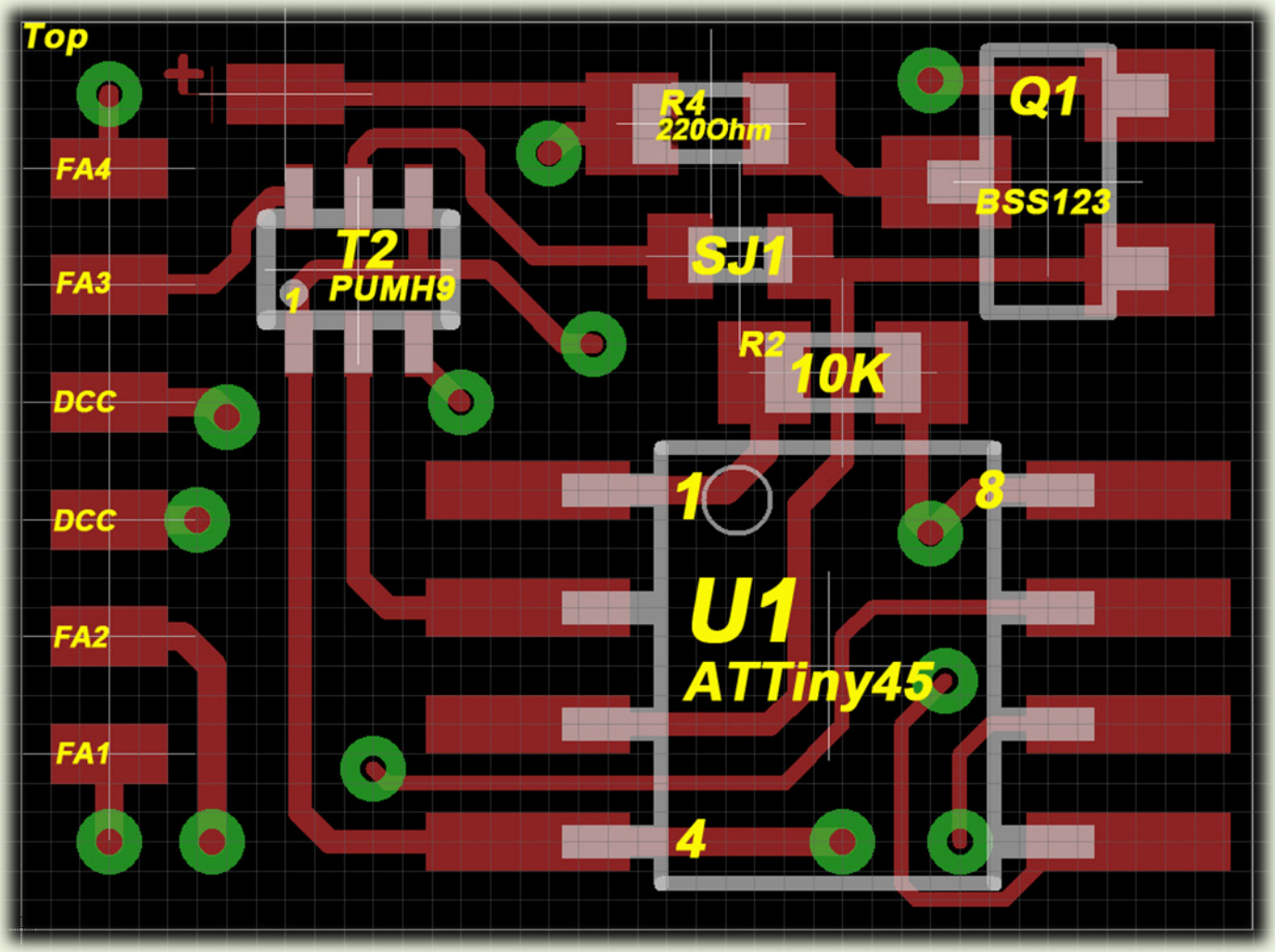
Weitere Hinweise:

- Auf die korrekte Ausrichtung der Bauteile achten (u.a. Pin1 von T1/T2 und von U1)
- am besten zuerst die beiden Doppeltransistoren T1 und T2 bestücken, danach den Rest.
- Auch auf die Ausrichtung der Z-Diode D1 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 mittleren Pads (DCC) auf der Oberseite der Platine löten (siehe Layout Seite 5).
    3. die beiden Kabel mit dem DCC-Gleissignal verbinden.
    4. Digitalzentrale einschalten.
    5. Nun sollte an den Lötpins 4 (Masse) und 8 (+) des Prozessors U1 eine Spannung von etwa 5,1-5,2 Volt zu messen sein!
    6. Jetzt kann man gleich noch die Spannung messen, welche die DCC-Zentrale aussendet (wichtig für die Berechnung der LED-Vorwiderstände). Dazu wieder das Messgerät mit Masse (Pin4 von U1) und dem Lötpad **+** (das einzelne über T2!) 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.

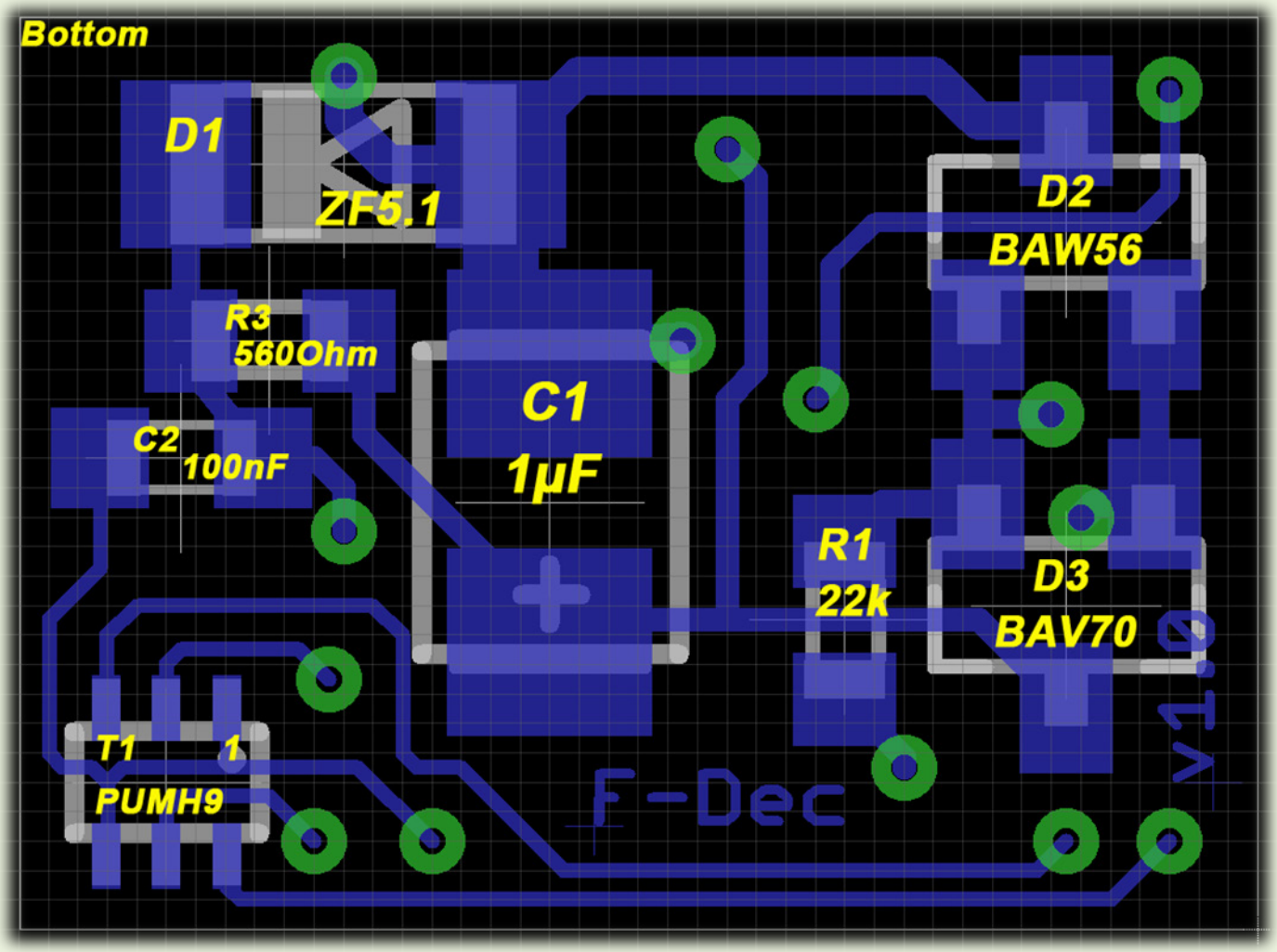
3.1. Bestückung Oberseite



Ansicht ***Bestückung oben*** (Erkennbar an den Lötpads auf der linken Seite der Platine)

- Pin1 von T2 ist mit einem winzigen Punkt gekennzeichnet!

3.2. Bestückung Unterseite



Ansicht ***Bestückung unten***

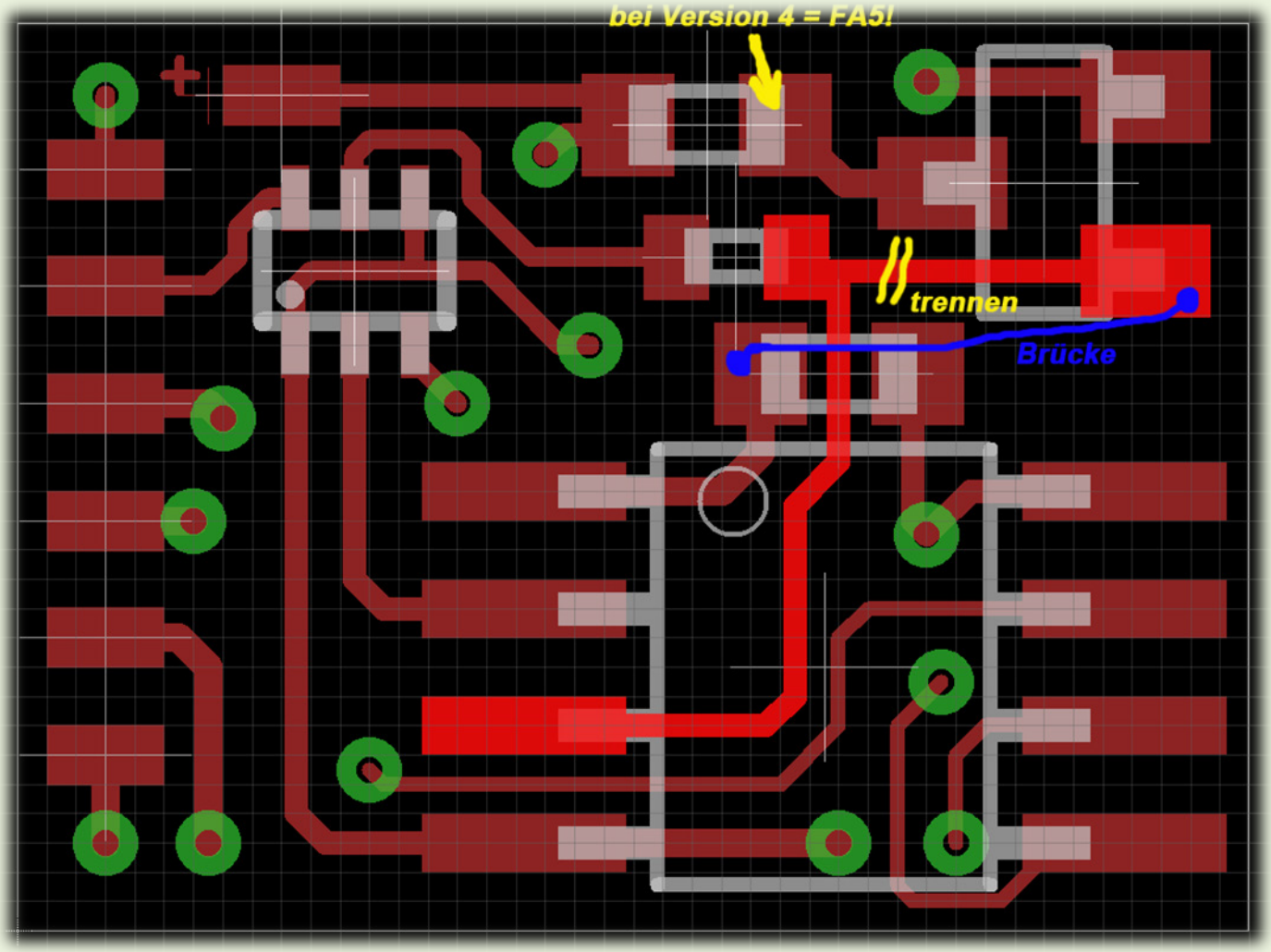
- Lage der Z-Diode, des Kondensators C1 und des Doppeltransistors T1 beachten!
- Pin1 von T1 ist mit einem winzigen Punkt gekennzeichnet!

3.3. Besondere Bestückungsvariante 1

Es ist möglich 4 Funktionsausgänge zu nutzen, ohne auf den ACK-Puls verzichten zu müssen.

Wenn man diese Option nutzen möchte, dann einfach folgendes beachten!

1. Es muss der RESET-Pin des Controllers deaktiviert werden! (siehe „Programmierung des Controllers“ -> „Einstellung der Fuses“ -> „Deaktivieren des RESET-Pin“)
2. Es werden alle Bauteile bis auf den Widerstand R2 (10K) bestückt! SJ1 geschlossen.
3. Weiterhin muss eine Leiterbahn auf der Oberseite durchtrennt und eine Drahtbrücke gesetzt werden! (siehe folgendes Bild)
4. Firmware mit der Bezeichnung OD2_F-Dec_Tiny45_V3.hex (und *_V3.eep) in den Controller schreiben.



3.4. Besondere Bestückungsvariante 2

Weiterhin ist es möglich 5 Funktionsausgänge zu nutzen, unter Verzicht des ACK-Pulses.

Wenn man diese Option nutzen möchte, dann einfach folgendes beachten!

1. Es muss der RESET-Pin des Controllers deaktiviert werden! (siehe „Programmierung des Controllers“ -> „Einstellung der Fuses“ -> „Deaktivieren des RESET-Pin“)
2. Das Durchtrennen der Leiterbahn und das Setzen der Brücke sind auch hier erforderlich! (siehe Bild auf Seite 7.)
3. Es werden alle Bauteile, bis auf R2 (10K) und R4 (220Ohm), bestückt. SJ1 wird geschlossen.
4. Nach dem Brennen der Firmware (OD2_F-Dec_Tiny45_V4.hex (und *_V4.eep)) steht nun zusätzlich an der Position des rechten Lötpads von R4 der 5. Funktionsausgang zur Verfügung. (siehe Bild Seite 7, oberer Hinweis in Gelb)

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.

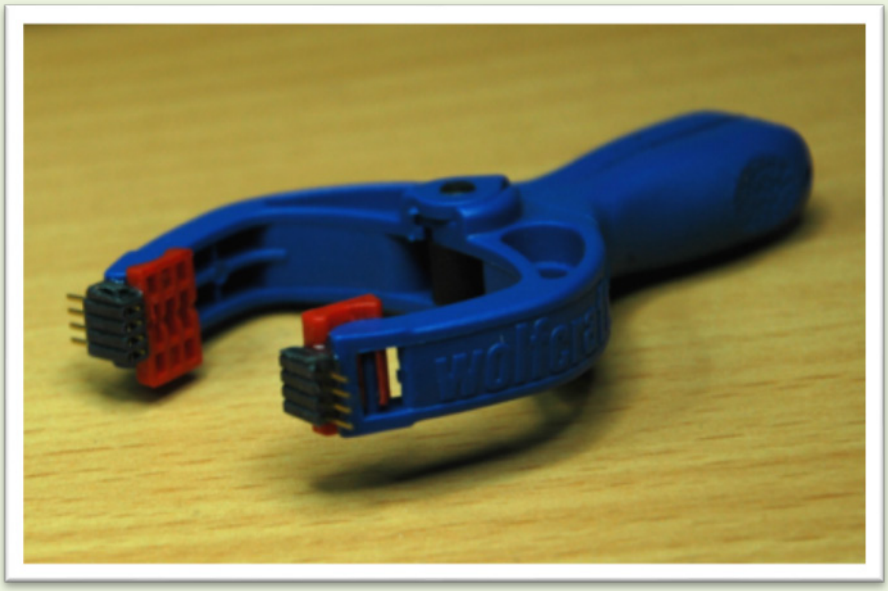
Der Controller hat in der SMD-Version einen „Beinchen-Abstand“ von 1,27mm.

Um nun den Prozessor mit dem Programmiergerät zu verbinden, braucht man einen kleinen Adapter. Dafür nimmt man einfach eine 10-polige Buchsenleiste 1,27mm und teilt diese in 2 x 4 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 ATTiny45.

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_Tiny45.hex* in den Controller.

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

4.3. Einstellung der Fuses

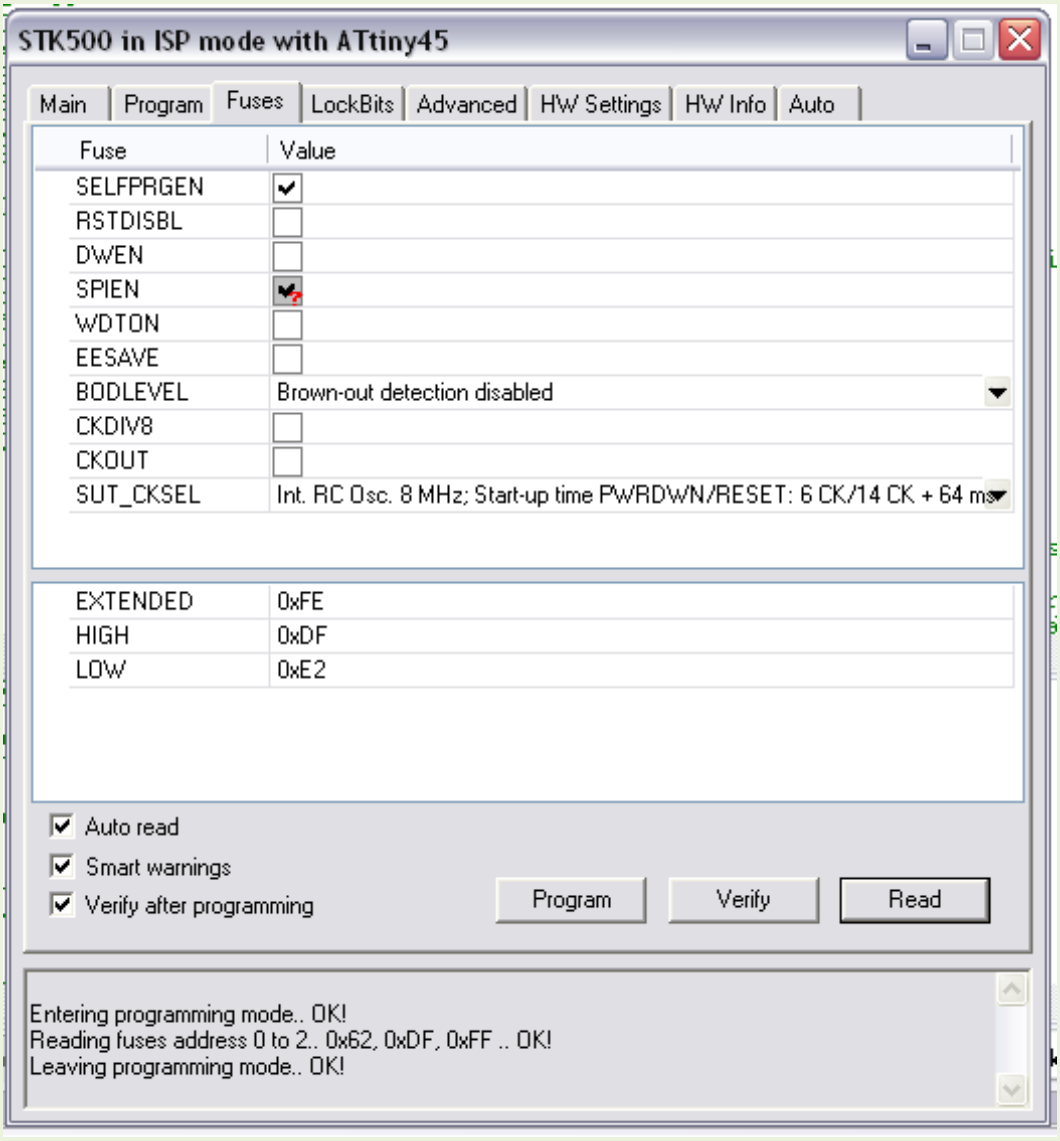
Nun müssen die Fuses (sozusagen die Sicherungsschalter) des Controllers angepasst werden, damit dieser den Bootloader ausführt und später auch das Programm richtig funktioniert.

Hinweis: Seite 11 beschreibt noch den RESET-Fuse, für die Bestückungsoption 3 und 4.

Das untere Bild zeigt die richtigen Einstellungen für den ATTiny45 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.



4.3.1. Deaktivieren des RESET-Pin

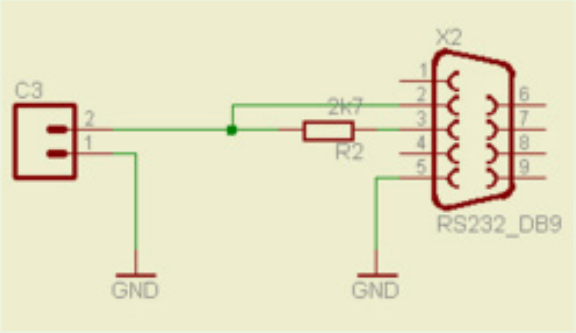
Damit die Bestückungsvariante 3 und 4 realisiert werden kann, muss der Pin 1 als normaler I/O-Pin konfiguriert werden. Normalerweise wird über Pin1 ein Reset des Prozessors ausgelöst.

Achtung: Wird dieser Pin über die Fuses deaktiviert, ist kein normales Programmieren über die ISP-Schnittstelle mehr möglich! Der Bootloader ist aber weiter funktionsfähig.

Jetzt wird zusätzlich zu den Einstellungen die auf Seite 10 beschrieben sind, das „RSTDISBL“-Fuse gesetzt! Die Werte für *EXTENDED; HIGH; LOW* ändern sich nur bei *HIGH* auf den Wert **5F**, die anderen 2 Werte bleiben gleich, wie auf dem Bild Seite 10 zu sehen.

Jetzt einfach wieder die neuen Werte in den Prozessor brennen.

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!

Nun schließt man zuerst den Decoder 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 einsprechende 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, je nach Bestückungsvariante (siehe Seite 4), heraus und tragt diese im Abschnitt *Programmingfiles* ein.

Jetzt wird die Masse-Verbindung (GND) von dem kleinen Adapter am Serialport zum Decoder hergestellt. Dazu am besten den PIN 4 am Controller U1 nutzen. Danach die andere Verbindung des Adapters auf PIN 5 des Controllers. 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 einsatz-bereit.

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. 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

6. Schaltplan