

Handbuch

Aufbau und Inbetriebnahme der

MoBaLiSt - Baugruppe

Inhaltsverzeichnis

Änderungsverzeichnis	3
1. Einleitung	4
2. Sicherheitshinweise	5
3. Aufbau	7
3.1 Die Funktion.....	7
3.2 Die Platine	8
3.3 Schaltplan	9
3.4 Bauteileliste für den MoBaLiSt.....	10
3.5 Aufbau in sieben Schritten	11
4. Inbetriebnahme	16
4.1 Versorgung der Hardware	16
4.2 Versorgung der Ausgänge 1-24.....	18
4.3 Firmware auf dem MoBaLiSt.....	20
5. Die Betriebsanzeige.....	23
6. Konfiguration der Ausgänge und Einsatz der Baugruppe	24
7. Terminierung des BiDiBus	25
7.1 Vorgehensweise der Terminierung	25
8. Sonderversion MoBaLiSt THT	26
9. Aufbau MoBaLiSt SMD mit Printsteckern.....	29
10. Aufbau MoBaLiSt SMD 64-Bit Version.....	31
10.1. Firmware beim MoBaLiSt SMD 64-Bit, ohne Bootloader.....	31
10.2. Firmware beim MoBaLiSt SMD 64-Bit, mit Bootloader.....	34

Änderungsverzeichnis

Version	Änderungsbeschreibung	Seite	geändert von	Datum
v1.0	Dokument MoBaLiSt „Handbuch“ erstellt	alles	C. Schörner	20.05.2013
v1.1	Diamex ALL AVR Programmer entfernt. Mit dem Prozessortyp vom MoBaLiSt hat der Programmer EEPROM Write Probleme	Seite 20	C. Schörner	11.10.2013
v1.2	Pin1 Erklärung für den Transceiver-Baustein ergänzt	Seite 11	C. Schörner	05.03.2014
v1.3	Programmieraablauf mit aktuelleren Abbildungen erweitert	Seite 22	C. Schörner	18.03.2014
v2.0	MoBaLiSt 64-Version eingefügt	Seite 29	W. Krügel C. Schörner	03.05.2015
v2.1	textliche Veränderungen und Erweiterungen	Alles	C. Schörner	05.05.2015
v2.2	MoBaLiSt 64-Version mit Bootloader eingefügt	ab Seite 28	W. Krügel C. Schörner	07.01.2018

1. Einleitung

Diese Bauanleitung beschreibt den Aufbau und Inbetriebnahme der MoBaLiSt - Baugruppe von Michael Volk.

Lesen Sie diese Bauanleitung **vor** Beginn des Zusammenbaus sorgfältig durch und beachten Sie die Sicherheitshinweise.

Diese Anleitung erhebt nicht den Anspruch auf ein kommerziell gefertigtes Produkt. Sie dient lediglich als Hilfe zum Aufbau des Bausatzes für versierte und interessierte Modellbahner ausschließlich für den Eigenbau.

Diese Anleitung wurde sorgfältig geprüft und nach bestem Wissen erstellt. Es kann kein Anspruch auf Vollständigkeit, Aktualität und Richtigkeit erhoben werden. Sollten Handelsnamen oder geschützte Bezeichnungen verwendet werden, so liegen alle Rechte beim Rechteinhaber.

Es wird keine Haftung für jedwede Art übernommen, die aus der Nutzung dieser Anleitung, deren Inhalte oder deren Gebrauch herleitbar wäre. Der Nutzer dieser Anleitung erklärt sich mit der Ingebrauchnahme damit einverstanden.

Die hier verwendete und teilweise beschriebene Software kann auf der Internetseite von [www.fichtelbahn](http://www.fichtelbahn.de) und www.opendcc.de als Download benutzt, erweitert und verbessert werden.

Alles Weitere zur Nutzung von Software, Hardware und Applikation, ist auf der Internetseite von **Fichtelbahn** und **OpenDCC** beschrieben. Der Nutzer und Anwender erklärt sich mit den dort beschriebenen Regelungen vorbehaltlos einverstanden.

Eine kommerzielle Nutzung der Software oder Teile daraus ist nicht statthaft!

Diese Bauanleitung darf keiner anderen Nutzung zugeführt werden, außer der bestimmungsgemäßen Anwendung zum Aufbau der MoBaLiSt - Baugruppe. Anderweitige Nutzung erfordert die Zustimmung des Autors, bzw. des Rechteinhabers der Internetseite www.fichtelbahn.de und www.opendcc.de.

2. Sicherheitshinweise

Das in dieser Bauanleitung beschriebene Modul ist ein elektrisch betriebenes Gerät. Es sind alle beim Betrieb notwendigen Vorsichtsmaßnahmen zu treffen, die mit dem Umgang mit elektrischem Strom anzuwenden sind.

Legen Sie an das Modul keinesfalls Netzspannung an.

Verwenden Sie keinesfalls Schaltnetzteile von PC's. Diese Geräte sind nicht erdfrei, d.h. es können hier betriebsbedingt an den Gleisen und angeschlossenen Geräten hohe Spannungen auftreten – Lebensgefahr!

Erden Sie keinesfalls leitfähige Teile Ihrer Modellbahnanlage!

Alle Schirmungen, Kabelschirme usw. sind ggf. wenn als notwendig erachtet auf einen gemeinsamen, erdfreien Punkt zusammen zu führen.

Das fertige Modul ist ausschließlich mit Schutzkleinspannung und Schutztrennung zu betreiben.

Modelleisenbahnen sind in der geläufigen Rechtsauffassung als Spielzeug eingestuft. Hier gelten besondere Bestimmungen.

Zur Stromeinspeisung sind ausschließlich die im Handel erhältlichen Netzspeisegeräte mit der entsprechenden Zulassung zu verwenden.

Achten Sie beim Erwerb auf die entsprechende Klassifizierung des Netzgerätes.

Näheres erfahren Sie unter www.VDE.de .

Bestimmungsgemäßer Gebrauch:

Das Modul ist dafür vorgesehen ausschließlich in Modellbahnanlagen, welche digital gesteuert werden sollen, zum Schalten und Melden eingesetzt zu werden.

Jeder andere Gebrauch ist nicht bestimmungsgemäß.

Das Modul ist nicht dafür bestimmt von Kindern unter 15 Jahren zusammengebaut, eingebaut oder betrieben zu werden.

Werkzeuge und Arbeitsmittel:

Sie benötigen:

- kleiner Seitenschneider
- Pinzette für kleinste Teile
- Lötzinn 0,5 oder 0,3 mm Durchmesser
- ggf. Flussmittel
- Reinigungsmittel, Pinsel, 100% Isopropylalkohol
- Lötsauglitze
- Lupenleuchte
- Lötkolben 30 Watt, oder besser eine thermisch geregelte Lötstation.
- Lötspitzen mit Eignung zum Verarbeiten von SMD-Bauteilen.

Keine Angst vor den kleinen SMD - Teilen, Mikrocontroller und ICs.
Es ist nur ein Durchhaltevermögen und eine ruhige Hand notwendig.....



Tipp:

schauen Sie sich ein kleines Video an, wie die „Tausendfüßler“ von Atmel problemlos von Hand gelötet werden kann:

Video

www.fichtelbahn.de

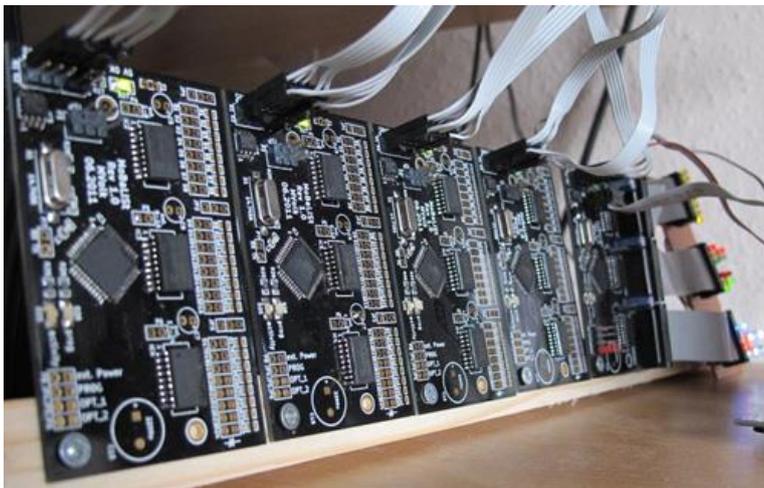
3. Aufbau

3.1 Die Funktion

Was bedeutet MoBaLiSt?

Das Selbstbauprojekt **MoBaLiSt** von Michael Volk ermöglicht die Steuerung der kompletten Beleuchtung auf einer Modellbahnanlage.

www.mobalist.de



Die Module können vorprogrammiert und StandAlone betrieben werden oder über ein Bussystem z.B. den BiDiBus an einen PC angeschlossen werden.

Mit dem BiDiB-Wizard Tool können die einzelnen Ausgänge vom PC aus konfiguriert werden und somit über ein Host-Steuerprogramm (z.B. Rocrail) verwaltet und geschaltet werden.

Die Module sind in einer **SMD-** und **THT-Version** verfügbar.

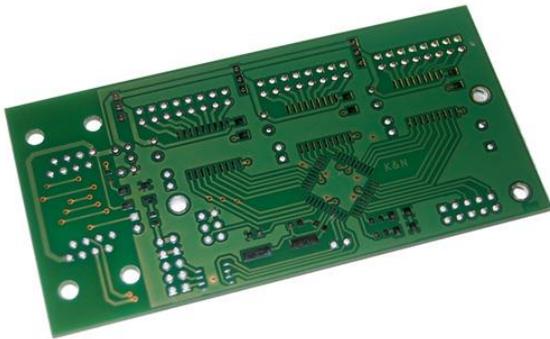
Die ausführliche Aufbauanleitung beschränkt sich auf die SMD-Version. Alle notwendigen Stücklisten und Bestückungsansichten für die THT-Version finden Sie im **Kapitel "Sonderversion MoBaLiSt THT"** in dieser Dokumentation oder auf der Homepage www.fichtelbahn.de.

Eigenschaften:

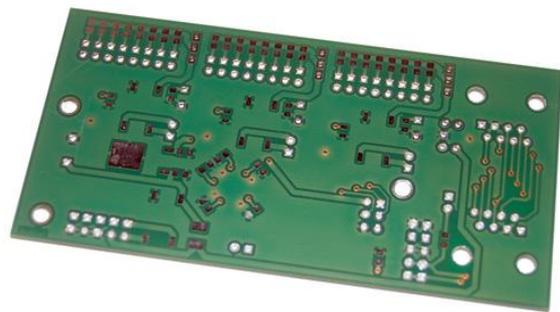
- 24 Ausgänge zum Ansteuern von LEDs und/oder Lämpchen
- PC-gesteuert per BiDiBus oder StandAlone
- alle Ausgänge sind einzeln dimmbar
- Versorgung über ein Zentrales 5V-Netzteil oder über einen MobaTrafo
- Die Ausgänge können für den LED-Betrieb mit der Bordinternen 5V-Spannung versorgt werden...
- ...oder in der SMD-Version in Blöcken zu je 8 Ausgängen mit externer Spannung versorgt werden (für Lämpchen, Relais...)
- drei Status-Leds zeigen 5V, Datenverkehr und Programmiermodus an
- 3 Jumper dienen zum Einstellen der Betriebsart (Programmiermodus, Aktionen)

3.2 Die Platine

Ansicht der Ober- und Unterseite der Platine

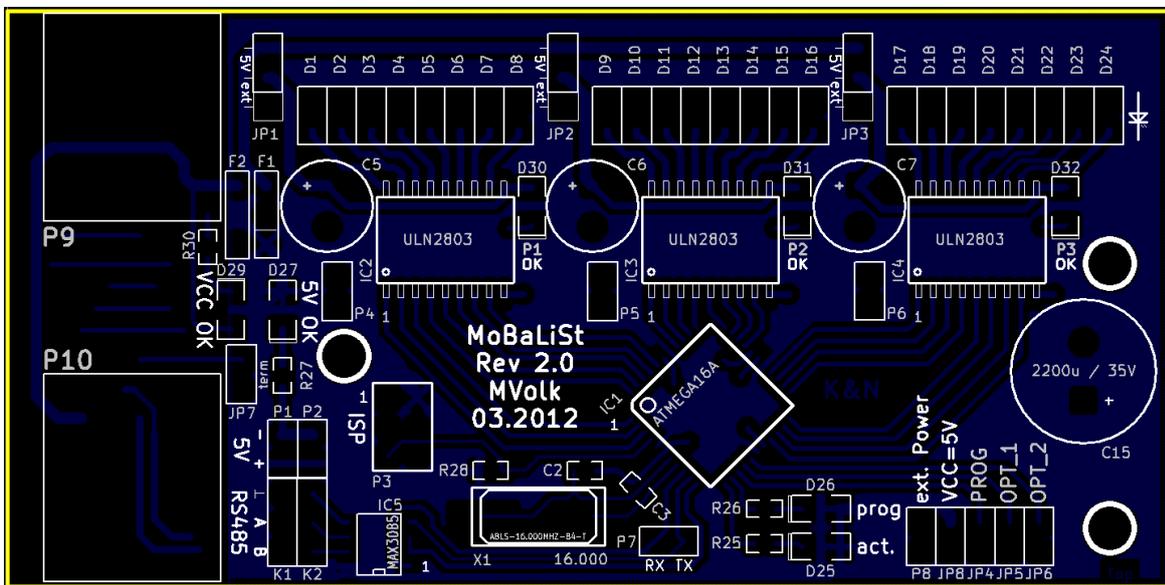


Oberseite der Platine

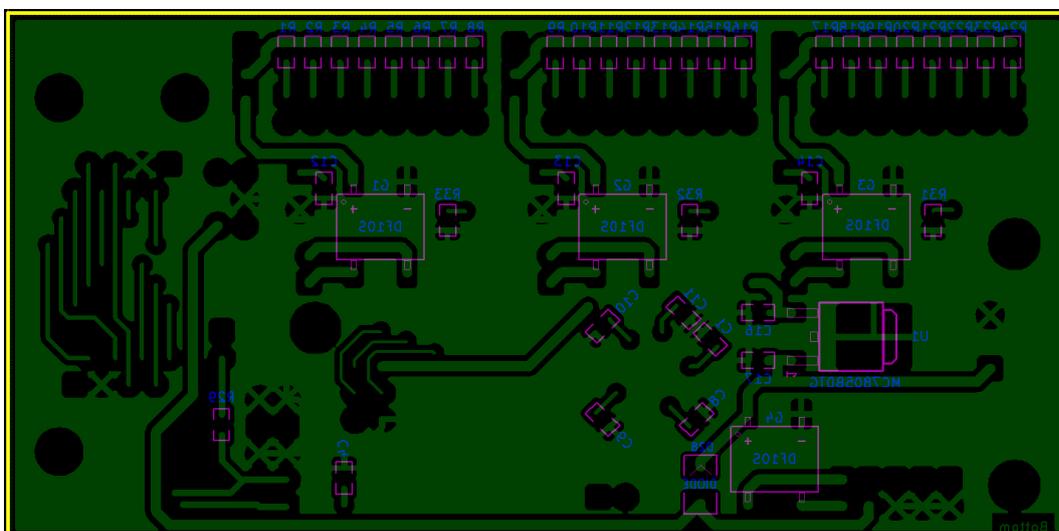


Unterseite der Platine

Placement Bestückungsseite:



Placement Lötseite:



3.4 Bauteileliste für den MoBaLiSt

Bauteil	Typ	Gehäuse	Anzahl	Note
IC 1	ATMEGA32A-AU	TQFP44	1	
IC 2-4	ULN2803ADW	SOIC18	3	
IC 5	MAX3085ECSA+	SOIC8	1	
U1	MC7805BDTG	DPAK-3	1	*1
Q1	16 MHZ	HC49/US	1	
C1, C4, C8, C9, C10, C11, C12, C13, C14, C16, C17	100nF	0805	11	
C2, C3	22pF	0805	2	
C5, C6, C7	470uF/35V 8mm	RM3,5	3	*2
<i>C5, C6, C7 Option bei LED's und 5V-Versorgung</i>	<i>100uF/50V 8mm</i>	<i>RM3,5</i>		
C15	2200uF/35V 12,5mm	RM5	1	*1
R1-R24 (bei LED-Betrieb)	220	0805	24	*4
R25, R26, R27, R30, R31, R32, R33	1k	0805	7	
R28	10k	0805	1	
R29	120	0805	1	*3
D1-D24	LED			
D25	LED gelb	1206	1	
D26	LED rot	1206	1	
D27, D29	LED grün	1206	2	
D30, D31, D32	LED grün	1206	3	*2
D28	1N4148	SOD323	1	
G1	DF10S	4-SMD	1	*1
G2, G3, G4	DF10S	4-SMD	3	*2
F1	Polyfuse 60V 2,5A		1	
F2	Polyfuse 60V 170mA		1	
PinHeader 2x8	LED		3	
PinHeader 2x3	ISP		1	
Pinheader 2x5	Power+RS485+Jumper		2	
PinHeader 1x2	ext Power		3	
P9, P10	RJ45 Bus		2	
Jumper			3	
Buchsen 2			4	
Buchsen 5			2	
Buchsen 16 (8x2) Flachband			3	

*1) 1 Stück, nur, wenn Spannungsregler auf Platine

*2) bei externe Ausgangs-Versorgung

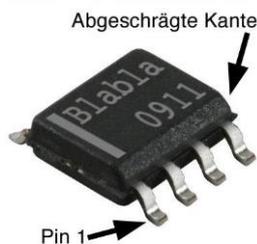
*3) ein Stück am Ende des RS485-Bus

*4) je nach Ausgangsbeschaltung

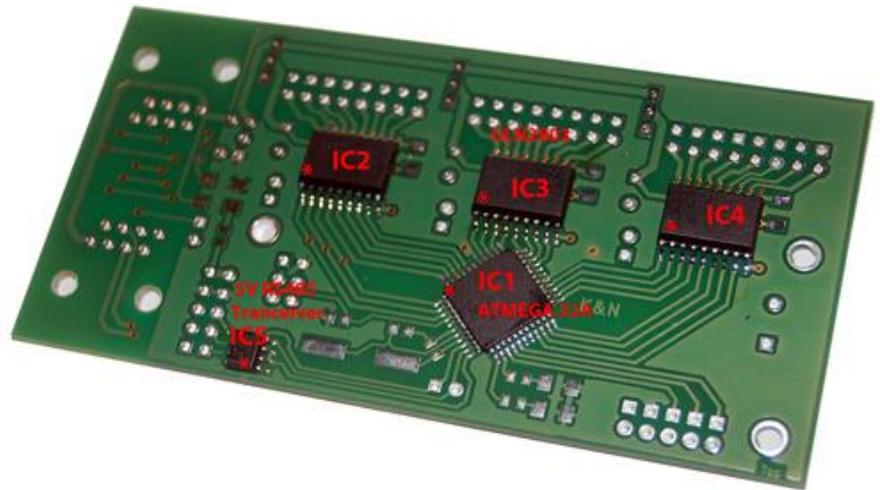
3.5 Aufbau in sieben Schritten

Schritt 1: Einbau der ICs:

Wir beginnen mit dem Auflöten des Mikrocontrollers **IC1** Atmega32A auf die Platine. Auf richtige Positionierung von **PIN 1** achten, dieser ist als kleiner Punkt auf der Oberseite des Gehäuses markiert.



Der Pin1 kann auch mit einem Längs- bzw. Querstrich, auf dem Baustein gekennzeichnet werden.



Die Vorgehensweise:

Ein Lötpad mit wenig Lötzinn versehen, den Käfer lagerichtig auflegen und verlöten. Jetzt kann noch korrigiert werden. Gegenüberliegendes Lötpad mit dem MC-PIN verlöten – schon ist alles fixiert.

Unter Zugabe von Flussmittel und wenig Lötzinn eine Seite nach der Anderen verlöten, dabei den LötKolben zügig von einer Seite zur anderen ziehen.

Das sieht schwieriger aus als es den Anschein hat, entschlossen handeln ist der Weg zum Ziel.

Auf Kurzschlüsse achten, diese ggf. mit der Löt-sauglitze entfernen.

Nicht zu lange an einzelnen Pins rumbrutzeln, sonst stirbt der Controller am Hitzeschlag!

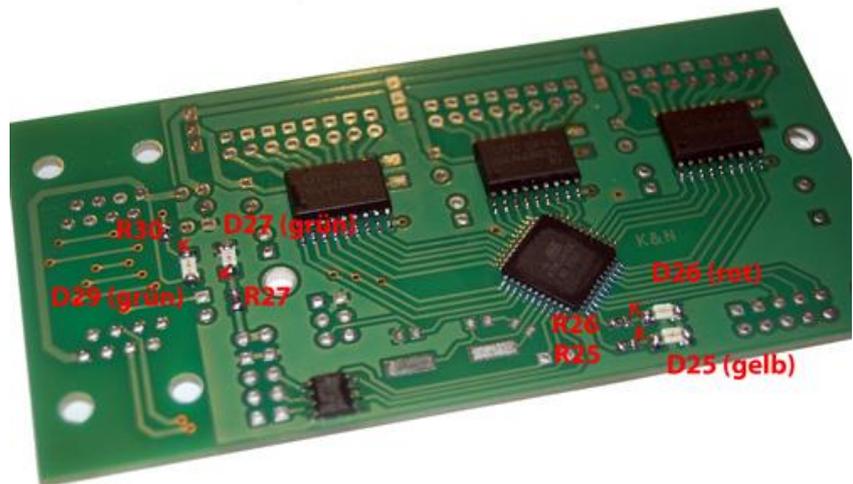
Die übrigen Lötreste und Flussmittelspuren mit Alkohol entfernen.

Im Anschluss folgen die drei Leistungstreiber ULN2803 **IC2-IC4** und der RS485 Receiver **IC5**. Hier ebenfalls auf die PIN 1 Markierung achten.

Schritt 2: Einbau der LED-Statusanzeigen

Vor dem Einlöten der SMD-LEDs, den Widerstand **R30**, **R27**, **R26** und **R25** mit 1kOhm auf die Platine löten.

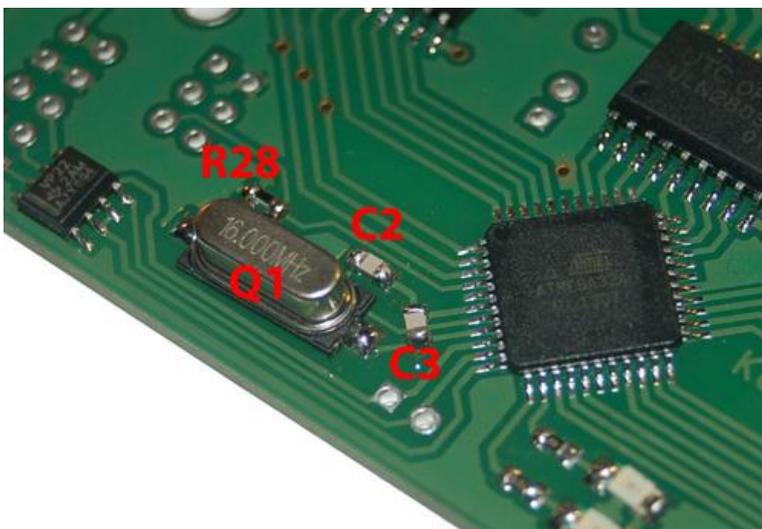
Bei der **D26** rot, **D25** gelb, **D27** grün und **D29** grün auf die richtige Einbaurichtung (Kathode zu Masse) achten. Die Kathode ist in der Abbildung mit einem K gekennzeichnet.



Kontrolle:

Die Platine mit Masse des Ohmmeters verbinden. Das Ohmmeter auf den Diodentest-Modus umstellen und mit der Messspitze die richtige Einbaulage prüfen. Bei richtiger Einbaulage leuchtet die Leuchtdiode.

Schritt 3: Einbau des Quarzes



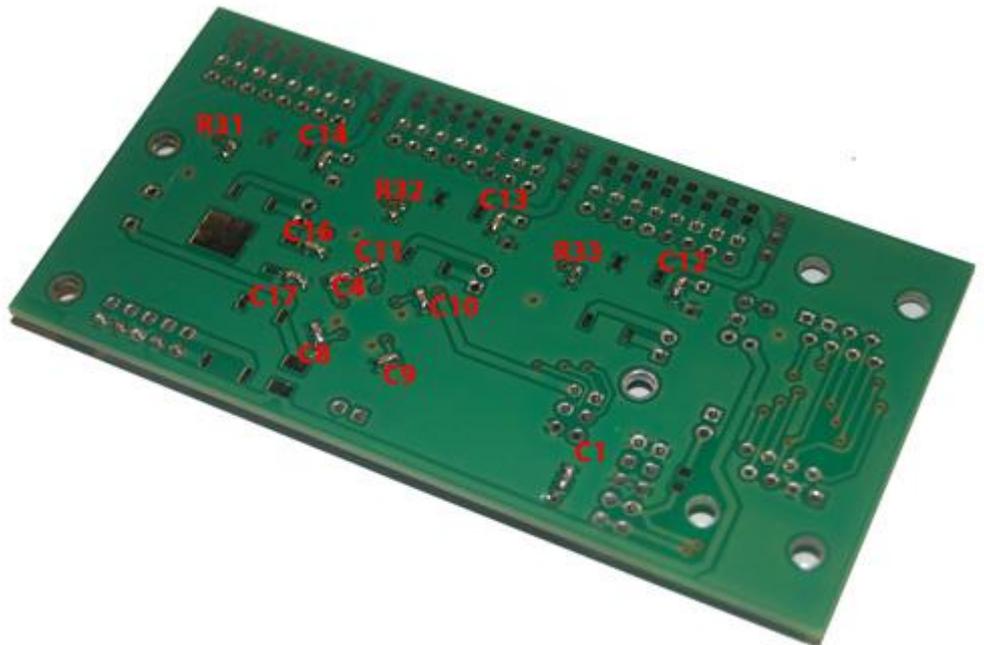
Die beiden Kondensatoren **C2** und **C3** mit 22pF für den 16Mhz Quarz **Q1** auf der Oberseite auflöten.

Neben den Quarz befindet sich der Reset-Widerstand **R28**.

Schritt 4: auf der Rückseite der Platine geht es weiter mit etwas Bauteilfutter (Abblockkondensatoren)

C1, C4, C8, C9, C10, C11, C12, C13, C14, C16 und C17 mit je 100nF.

Die drei Widerstände **R31, R32** und **R33** mit 1kOhm sind für die drei PORT-LEDS auf der Oberseite.

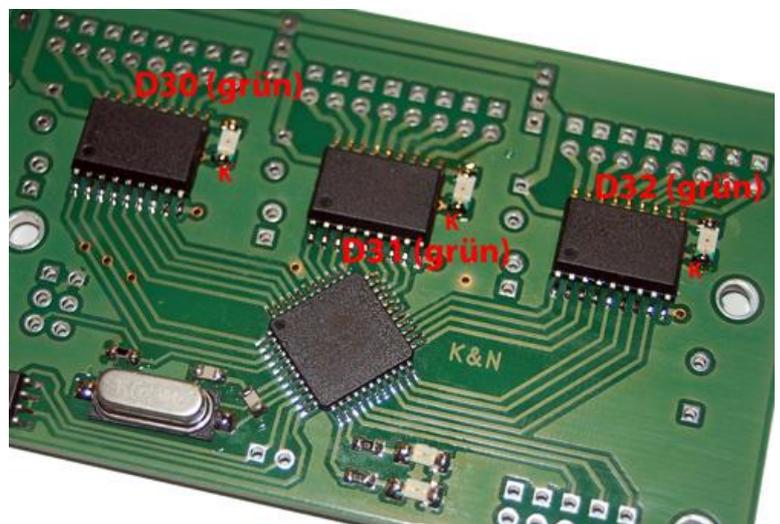


Schritt 5: die PORT-LEDS

Die Port-LEDs **D30, D31** und **D32** mit richtiger Einbaulage auf der Platine verlöten. (Abbildung kennzeichnet die Kathode mit einem K)

Bedeutung:

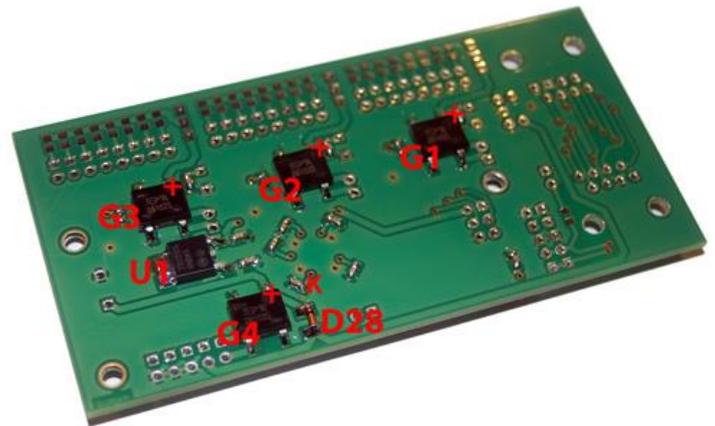
Die Portgruppen 1-3 (mit je 8 Ausgänge) können unterschiedlich mit einer anderen Versorgungsspannung über den JP1-JP3 beaufschlagt werden. Die Status-LED signalisiert die Spannungsversorgung.



Schritt 6: Aufbau der Versorgungseinheiten:

Der MoBaLiSt und die angeschlossenen Leuchtmittel können über drei Wege mit Spannungen versorgt werden:

- ➔ von einer externen Quelle (z.B. 12V Steckernetzteil)
- ➔ vom BiDiBus
- ➔ von der internen 5V Versorgung



Versorgungsvarianten für Platine (Controller) und Ausgänge (IC2-IC4)

		Platine Versorgung	Ausgänge Versorgung	Poly- fuse	Bemerkung
Variante 1	eigenst. komplette Versorgung 9-18V DC/AC	(P8)	(P8)	F1	
Variante 1	eigenst. komplette Versorgung hier zentrales 5V Netzteil 5V DC	(P1/P2)	(P1/P2)	F1	(Aufbau wie bei P8, dann Seite) 17, 18
Variante 2	externe unabhängige Spannungs- quellen für die Ausgänge 5-18V DC/AC	(P1/P2) oder (P8)	(P4/P5/P6)	NO	
Variante 3	BiDiB für die Platine	(RJ45- Buchsen)	(P4/P5/P6)	F2	

Grüner Rahmen = Interne 5V Versorgung“ von Platine zu Ausgängen per Jumper möglich, siehe auch Seite 18, oben.

Blauer Rahmen = Externe unabhängige Versorgung der Ausgänge. Per Jumper zu aktivieren, siehe auch Seite 18, unten.

Variante1:

Für eine **eigenständige komplette Versorgung** müssen die Bauteile **P8**, **G4**, **U1**, **D28** und **C15** bestückt werden. Sollen die angeschlossenen Leuchtmittel (LEDs) von dieser 5V Versorgung mitversorgt werden, dann muss noch die Multifuse **F1** aufgelötet werden.

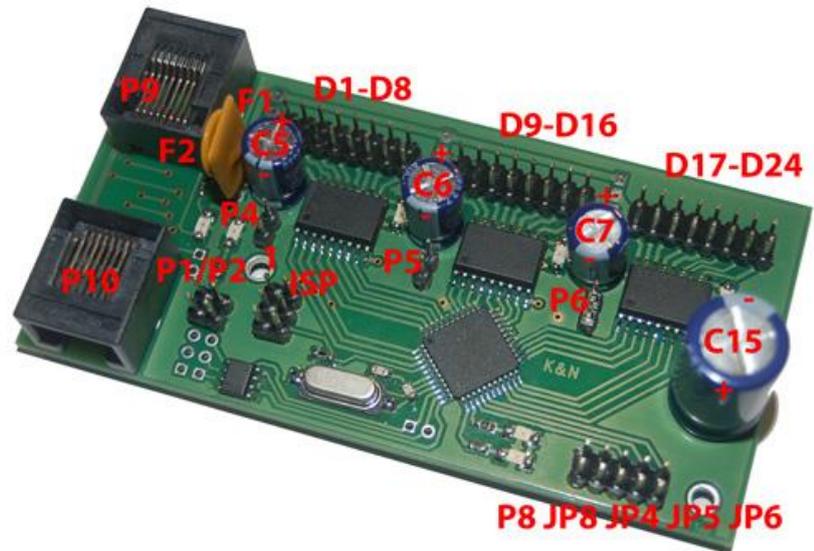
Variante2:

Werden die Leuchtmittel je Port von einer **externen unabhängigen Spannungsquelle** versorgt, dann müssen ja nach gewählten Port die Teile **G3**, **P6** und /oder **G2**, **P5** und/oder **G1**, **P4** bestückt werden. Die Multifuse **F1** darf **nicht** eingelötet werden.

Variante3:

Soll die **Baugruppe vom BiDiBus** versorgt werden, dann wird die Multifuse **F2**, **U1**, **D28** und **C15** eingelötet.

Beachten Sie aber, dass der BiDiBus max. 500mA treiben kann und somit die LEDs extern versorgt werden müssen.



Unabhängig von der Versorgungsspannung müssen die Elkos **C5**, **C6** und **C7** zu den jeweiligen Ports aufgelötet werden. Die Kapazität kann hier zwischen 100 μ F bis 470 μ F gewählt werden.

Für eine BiDiBus Kommunikation werden die RJ45 Buchsen **P9** und **P10** benötigt. Die weiteren Stiftleisten **D1-D24** sind die Ausgänge für die Leuchtmittel.

Alle weiteren Stiftleisten die auf der Abbildung zusehen sind werden zum Abschluss des Aufbaus noch bestückt. **Die Erklärung ihrer Funktionen folgt im nächsten Kapitel.**

Schritt 7: Terminierungswiderstand

Kommt der **MoBaLiSt** als letzter Teilnehmer am BiDiBus zum Einsatz, dann benötigt der Knoten einen Busabschluss.

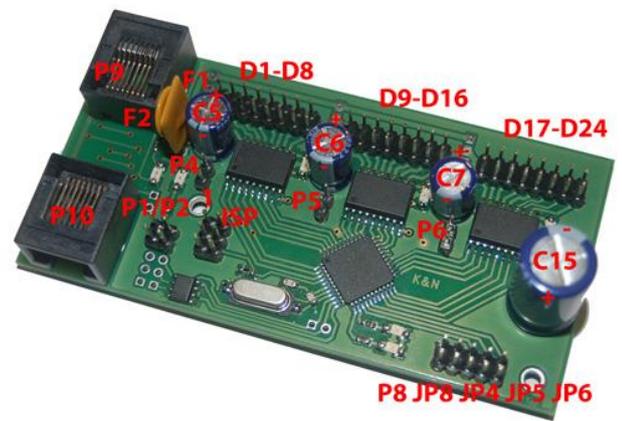
Weitere Informationen zu der Bus-Terminierung finden Sie im Kapitel 7.

Der Terminierungswiderstand **R29** mit 120 Ohm auf der Rückseite und die Stiftleiste **JP7** sollte auf jeder Baugruppe bestückt werden.



4. Inbetriebnahme

4.1 Versorgung der Hardware

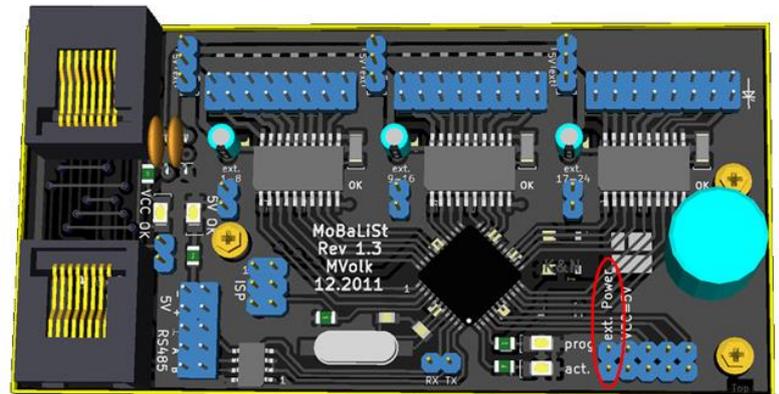


1. von einer externen 12V Quelle versorgt (Variante 1)

An die Stiftleiste **P8** wird eine Gleichspannung von einem Labornetzteil mit Strombegrenzung 100 mA und 6 Volt angelegt, die Spannung langsam bis auf 12 Volt hochdrehen.



Ist die Versorgung korrekt aufgebaut, dann erleuchtet die grüne Leuchtdiode **D27** auf der Platine und signalisiert, dass eine **5V Spannung** vorhanden ist.



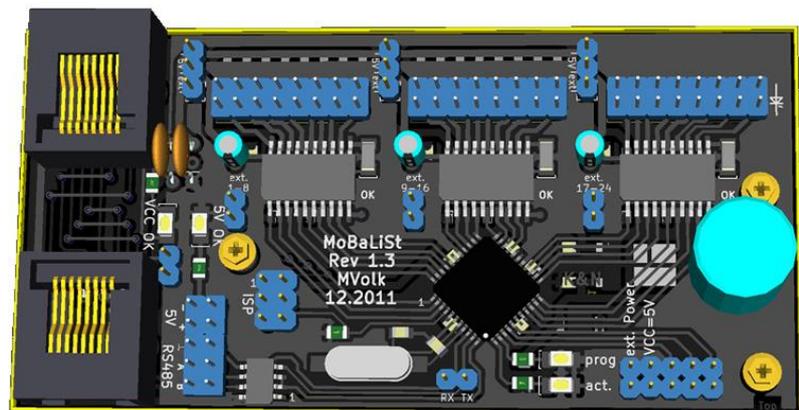
Lichttrafo

2. vom BiDiBus versorgt (Variante 3)

Verbinden die den MoBaLiSt über den BiDiBus mit dem GBMboost Master und schalten Sie den GBMboost Master ein.



Achtung:
Es darf kein Jumper auf der Stiftleiste **JP8** stecken. Der Prozessor würde sonst 12V vom Bus bekommen und das nicht überleben.



VCC über Bus



Ist die Versorgung korrekt aufgebaut, dann erleuchten die beiden grünen Leuchtdioden **D27** und **D29** auf der Platine und signalisieren, dass eine **5V Spannung** vorhanden ist.

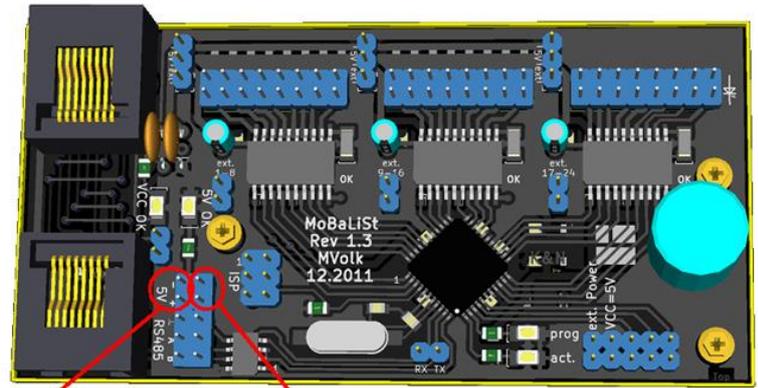
Aus Gründen der Strombelastung darf dies aber nur für den Controller erfolgen. Die Ausgänge müssen extern versorgt werden!

3. zentrales 5V Netzteil

Bei manchen Modellanlagen arbeitet ein zentrales 5V Netzteil. Dieses Netzteil könnte mehrere Platinen mit 5V versorgen.

ACHTUNG!

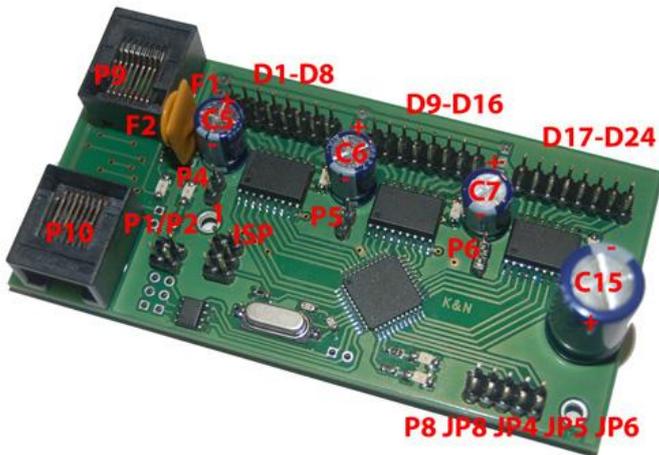
Der 5V-Eingang hat keinen Verpolungsschutz!
Bitte Polung genau prüfen!



vom 5V-Netzteil

zur nächsten Platine

Welche Bedeutung haben die weiteren Stiftleisten?



P1 / P2:

An den beiden Stiftleisten kann die interne Spannung von 5V abgegriffen werden. Hier liegen unabhängig der gewählten Versorgungsvariante immer 5V an. (zentrales 5V Netzteil)

ISP:

Das ist die Programmierschnittstelle für die Prozessorprogrammierung. Achten Sie bitte beim Aufstecken des Programmers auf die PIN1 Markierung.

P4/P5/P6:

Hier kann die unterschiedliche externe Spannung für getrennte Ausgangsgruppen eingespeist werden. Als Versorgungsspannung kann eine DC/AC Spannung von 5-18V gewählt werden.

P8:

Betriebsspannung DC/AC von 9-18V für die Versorgungsvariante 1

JP8:

Hier darf kein Jumper gesteckt werden. Stiftleiste bleibt offen.

JP4/JP5/JP6:

Der MoBaLiSt hat drei Eingänge zum Schalten von Effekten.

4.2 Versorgung der Ausgänge 1-24

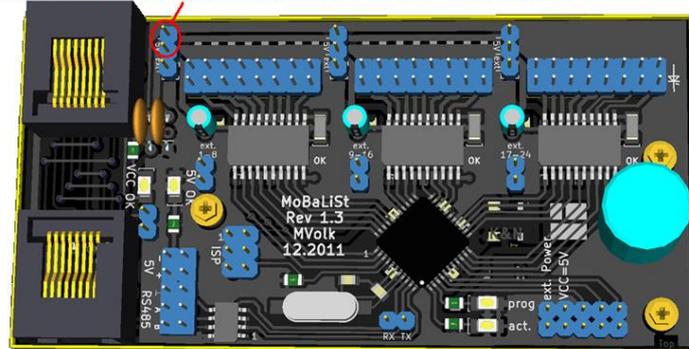
Die Ausgänge 1-24 können auf zwei Arten mit Spannung versorgt werden.

- 1. mit Bordeigene 5V
- 2. externes Netzteil bzw. Trafo 5V-18V

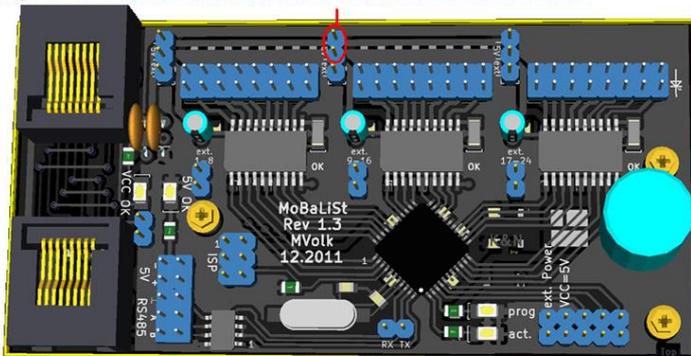
1. mit Bordeigene 5V

Wird der Controller durch ein zentrales 5V-Netzteil oder durch einen externen Trafo versorgt, können die auf dem Board vorhandenen 5V für die Ausgänge genutzt werden. Dazu sind bei den Jumper **JP1**, **JP2** und **JP3** die oberen beiden Pads mit einem Lötspunkt zu brücken. Die PolyFuse **F1** ist zu bestücken.

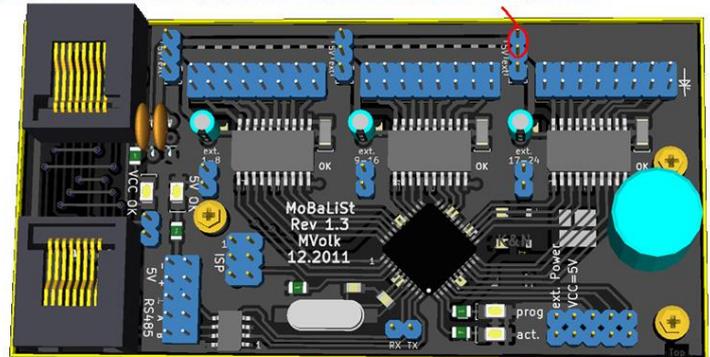
Ausgänge 1-8 mit Board-5V, Lötbrücke zwischen den oberen beiden Pads.



Ausgänge 9-16 mit Board-5V, Lötbrücke zwischen den oberen beiden Pads.



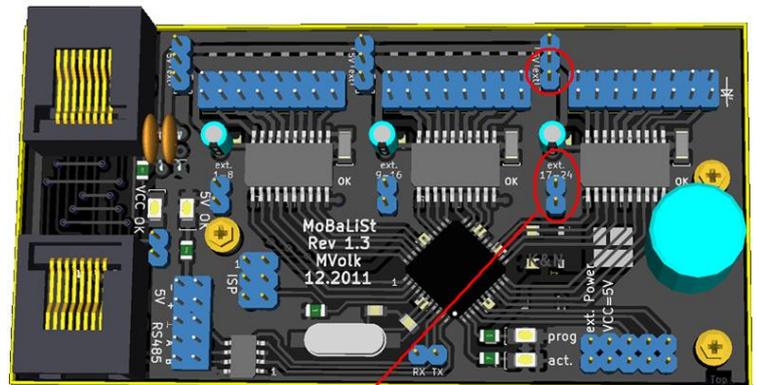
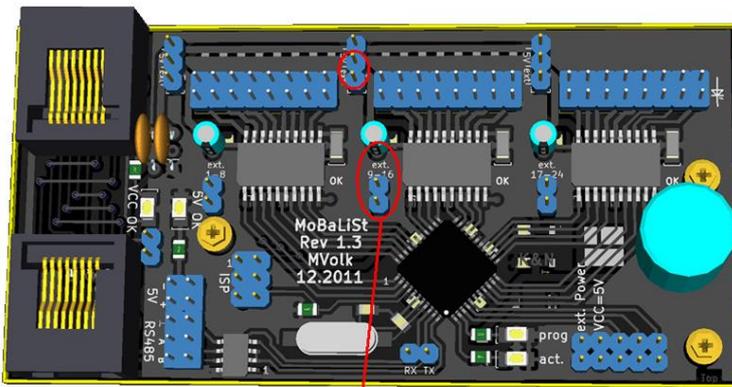
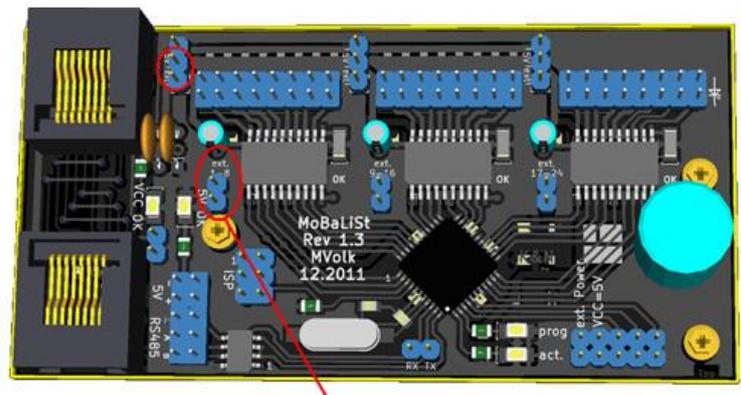
Ausgänge 17-24 mit Board-5V, Lötbrücke zwischen den oberen beiden Pads.



2. externes Netzteil

Reichen die 5V zum Betrieb der Ausgänge nicht aus, können die Ausgänge über einen externen Trafo versorgt werden. Der Trafo ist an P4, P5 und P6 anzuschließen.

Dazu sind bei den Jumper **JP1**, **JP2** und **JP3** die unteren beiden Pads mit einem Lötspunkt zu brücken. Zusätzlich sind die Gleichrichter **G1-G3** zu bestücken.

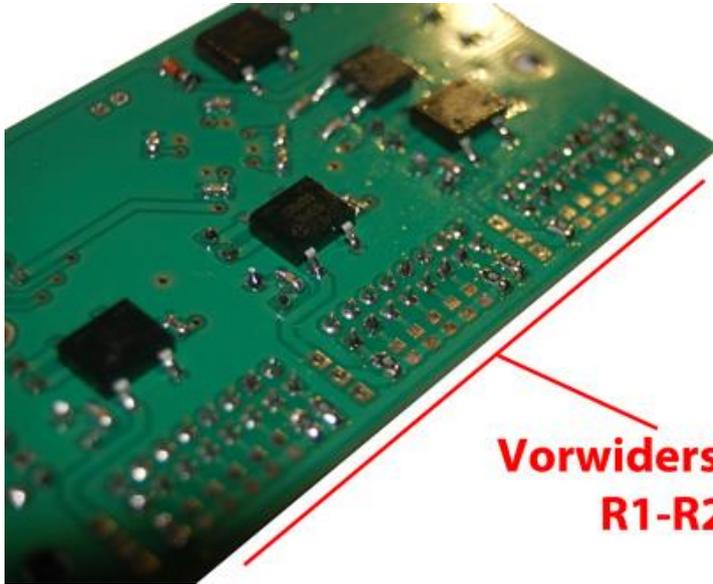


externer Trafo (Lichttrafo)

externer Trafo (Lichttrafo)

3. Vorwiderstände der Ausgänge

Je nach gewählter Versorgungsspannung variiert die Spannung der Ausgänge.
Die Vorwiderstände müssen entsprechend berechnet werden!



**Vorwiderstände
R1-R24**

Beispiele:

5V: LED 220Ohm

18V: LED 1KOhm,
Lämpchen 400hm

Am besten Sie messen die Spannung und berechnen die Vorwiderstände entsprechend. Es kann ruhig eine Nummer größer gewählt werden, da die LEDs bei voller Ansteuerung für die meisten Moba-Anwendungen zu hell sind.

Einen Rechner gibt es hier: [LED-Rechner](#)

4.3 Firmware auf dem MoBaLiSt

Schritt 1: Vorbereitung der Programmierung

1. Zum Programmieren der Prozessoreinheit benötigen Sie einen ISP-Programmer

z.B. der AVRISP-MK II und AVR Studio 4.

(Die gezeigten Screenshots der folgenden Schritte geben die Version 4 von AVR Studio wieder.)



2. Die aktuelle Firmware finden Sie auf der Homepage von OpenDCC / Fichtelbahn zum Download.

- FLASH - File
- EEPROM - File
- die OpenDCC - BiDiB Seriennummer

Diese Dateien sind gepackt. Mit einem gebräuchlichen Entpacker wie z.B. WinRar oder Winzip entpacken und am besten in einem separaten Ordner speichern.

Zur Seriennummer:

Für alle Baugruppen, welche [BiDiB](#) unterstützen, ist die Vergabe einer eindeutigen Produktkennung zwingend erforderlich anhand einer **Seriennummer**.

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

http://www.opendcc.de/elektronik/bidib/opendcc_bidib.html

OpenDCC - BiDiB Seriennummer

Neue Seriennummer generieren

Produktkennung

Bemerkung

Ihre Seriennummern

Datum	Produkt	Bemerkung	Seriennummer	
02.06.2012, 16:46	BiDiBus-S88-Bridge 1	s88-Bridge_1	:0201fe0000	.eep
14.05.2012, 20:01	LightControl 1, 32 LEDs, 16 Port, 4 Servo	LControl_1	:0207fe0000	.eep

[Abmelden](#)

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

Wie funktioniert es?

- In den Downloaddateien (Firmware) zu OpenDCC-BiDiB-Projekten ist die VID und PID bereits integriert, jedoch ist keine Seriennummer enthalten. Die Downloaddateien können daher problemlos auf den Prozessor geladen werden, eine eventuell bereits vorhandene Seriennummer wird dabei nicht überschrieben.
- Wenn die Firmware beim Starten keine geladene Seriennummer erkennt, blinkt ein Fehlercode und der Betrieb wird blockiert. **Ohne Seriennummer läuft die Firmware nicht!**
- Zu jeder Baugruppe muss mit dem Generator **einmalig** eine Seriennummer erzeugt werden.
- **Die erzeugte Seriennummer bitte notieren und auch auf der Baugruppe vermerken.**
- Die Seriennummer muss auf dem üblichen Weg (also via Programmiergerät) in die Baugruppe eingespielt werden, andere Inhalte werden dabei nicht überschrieben.

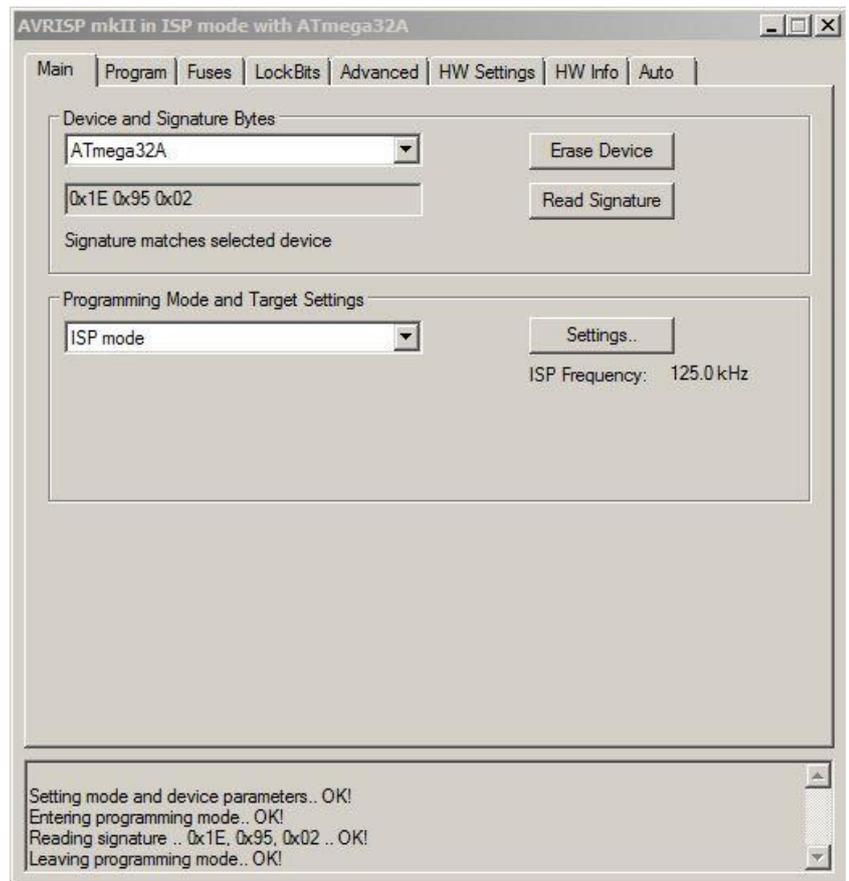
Schritt 2: Einspielen der Software:

Verbinden Sie das Modul mit der Stromversorgung und den Programmieradapter mit der ISP Stiftleiste auf der Platine, beachten Sie auf richtige Polung (PIN 1)

Starten Sie das AVR Studio und bauen Sie eine Verbindung auf.

Wählen Sie Atmega32A und lesen Sie die Signatur des Mikrocontrollers aus.

Der Programmiermodus ist auf **ISP Mode** einzustellen.

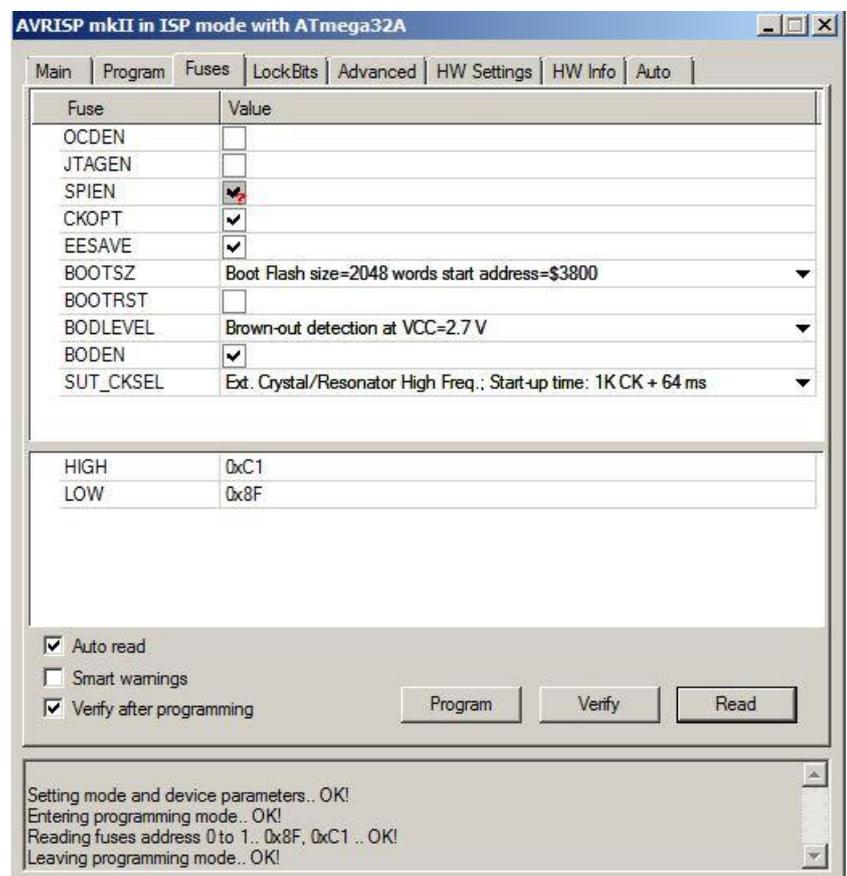


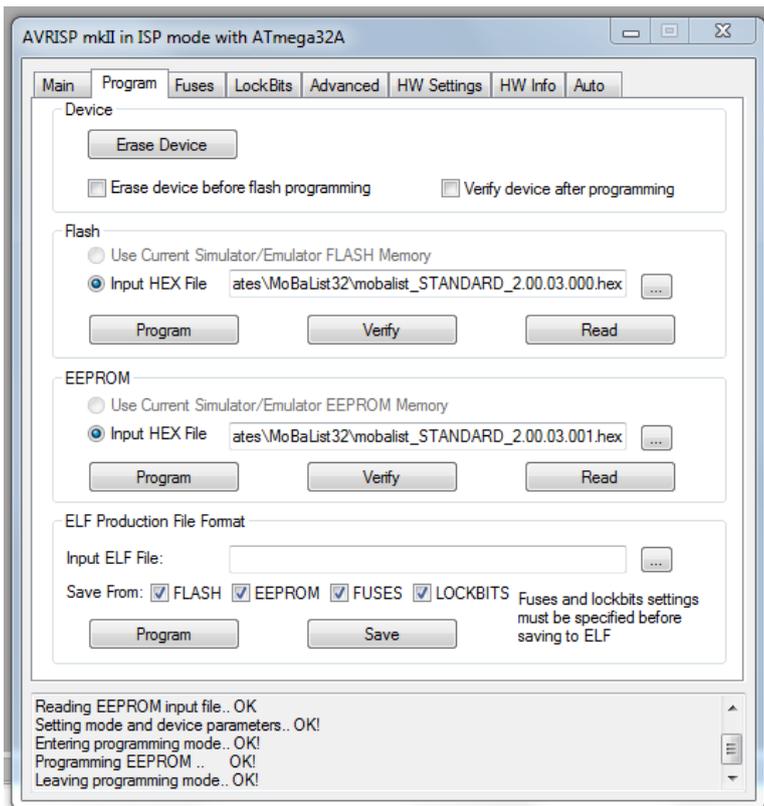
Als Nächstes werden die Fuses eingestellt. Gehen Sie hier sehr sorgfältig mit den Eingaben um. Ein „verfuster“ Atmel lässt sich nur schwer wiederbeleben.

Das **Highbyte** **0xC1**
Das **Lowbyte** **0x8F**

Jetzt klicken Sie auf die Schaltfläche **Program** um die Fuses auf die Baugruppe zu übertragen.

Ist alles ordnungsgemäß eingestellt, so erhalten Sie OK!





Dann wird die Datei „mobalist_Standard_...000.hex“ für den Flash-Speicher und die Datei „mobalist_Standard_...001.hex“ in das EEPROM übertragen.

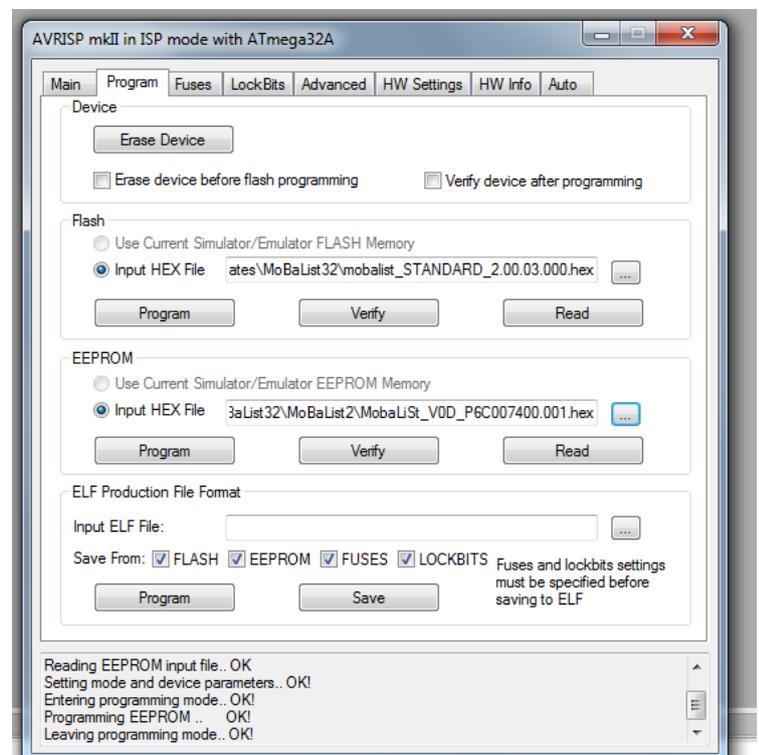
Der Programmierablauf startet mit einem Klick auf die Schaltfläche **Program**.

Ist die Programmierung erfolgreich verlaufen erhalten Sie die unteren Meldungen im Fenster.

Die LEDs **D25** und **D26** blinken.

Erst wenn Sie die BiDiB-Seriennummer als zweiten Stepp in das EEPROM eingespielt wurde, hört das Blinken auf.

War das Einspielen der Software + Seriennummer erfolgreich, meldet AVR Studio OK und die LED D26 flimmert.



5. Die Betriebsanzeige



Power VCC
(LED D27)



Power BiDiB
(LED D29)



PROG
(LED D26)

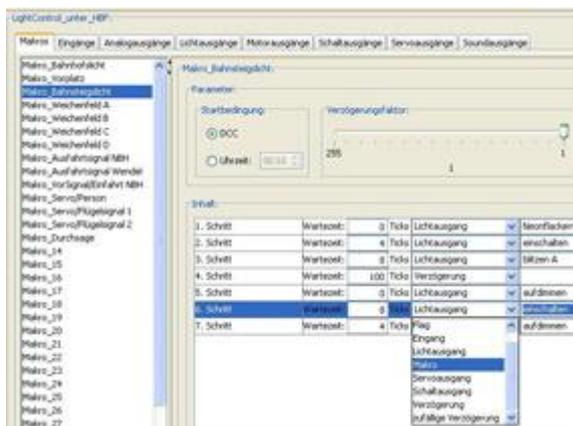


Activity
(LED D25)

Zustand der LED	Bemerkung
POWER LED aus	Keine Betriebsspannung angeschlossen
POWER LED leuchtend	Baustein wird mit Betriebsspannung versorgt
Power BiDiB LED aus	Modul ist nicht mit dem BiDiBus verbunden
Power BiDiB LED on	Modul ist mit dem BiDiBus verbunden
PROG LED flimmert	Modul ist betriebsbereit und arbeitet
PROG LED off	Modul ist nicht programmiert
PROG LED blinkt	Modul ist im Identify-Modus
Message LED aus	keine Kommunikation mit BiDiB
Message LED on	mit BiDiB verbunden und Kommunikation

6. Konfiguration der Ausgänge und Einsatz der Baugruppe

Der **MoBaLiSt** ist jetzt vollständig aufgebaut und betriebsbereit. Mit diesem Baustein ist man jetzt in der Lage eine intelligente Lichtsteuerung und Signalsteuerung mit Leuchtdioden und Glühbirnen auf der Anlage zu realisieren.



Die Ausgänge werden im **BiDiB-Wizard Tool** konfiguriert und einem Makro / Accessory zugeordnet.

Das Verfahren ist identisch mit der LightControl.

Als Hilfe für die Konfiguration kann die **LightControl Anleitung "Inbetriebnahme und Anwendung"** verwendet werden. Diese Anleitung finden Sie auf der Fichtelbahn Homepage im Downloadbereich der LightControl.

7. Terminierung des BiDiBus

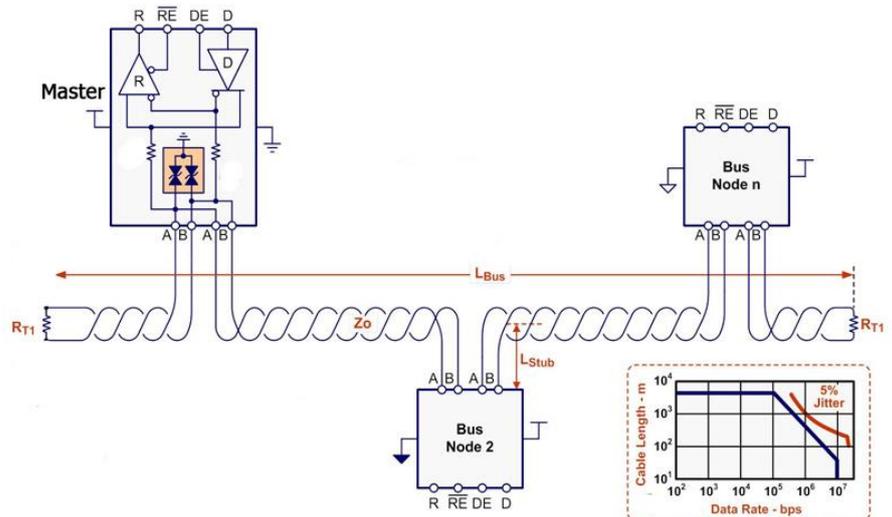
Der **BiDiBus** besteht aus einer RS485-2 Draht Verbindung die speziell für Hochgeschwindigkeits-Datenübertragungen über große Entfernungen entwickelt worden ist und eine zunehmende Verbreitung in industriellen Anwendungen findet. Dank diesen Eigenschaften kann eine Kabellänge von über 500 Meter mit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



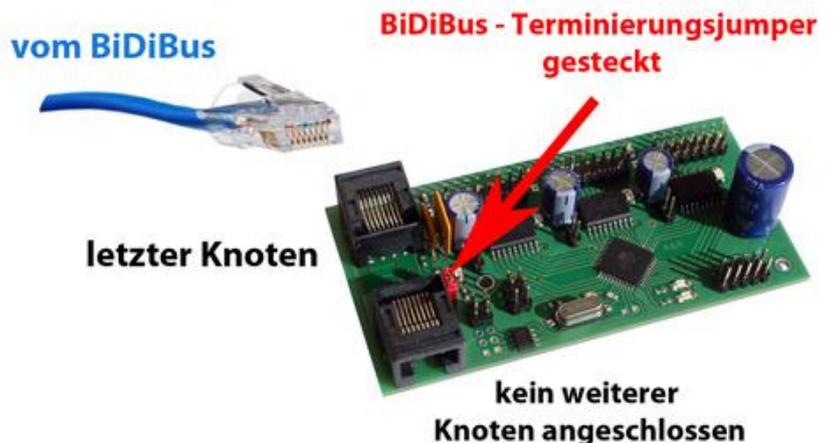
7.1 Vorgehensweise der Terminierung

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.

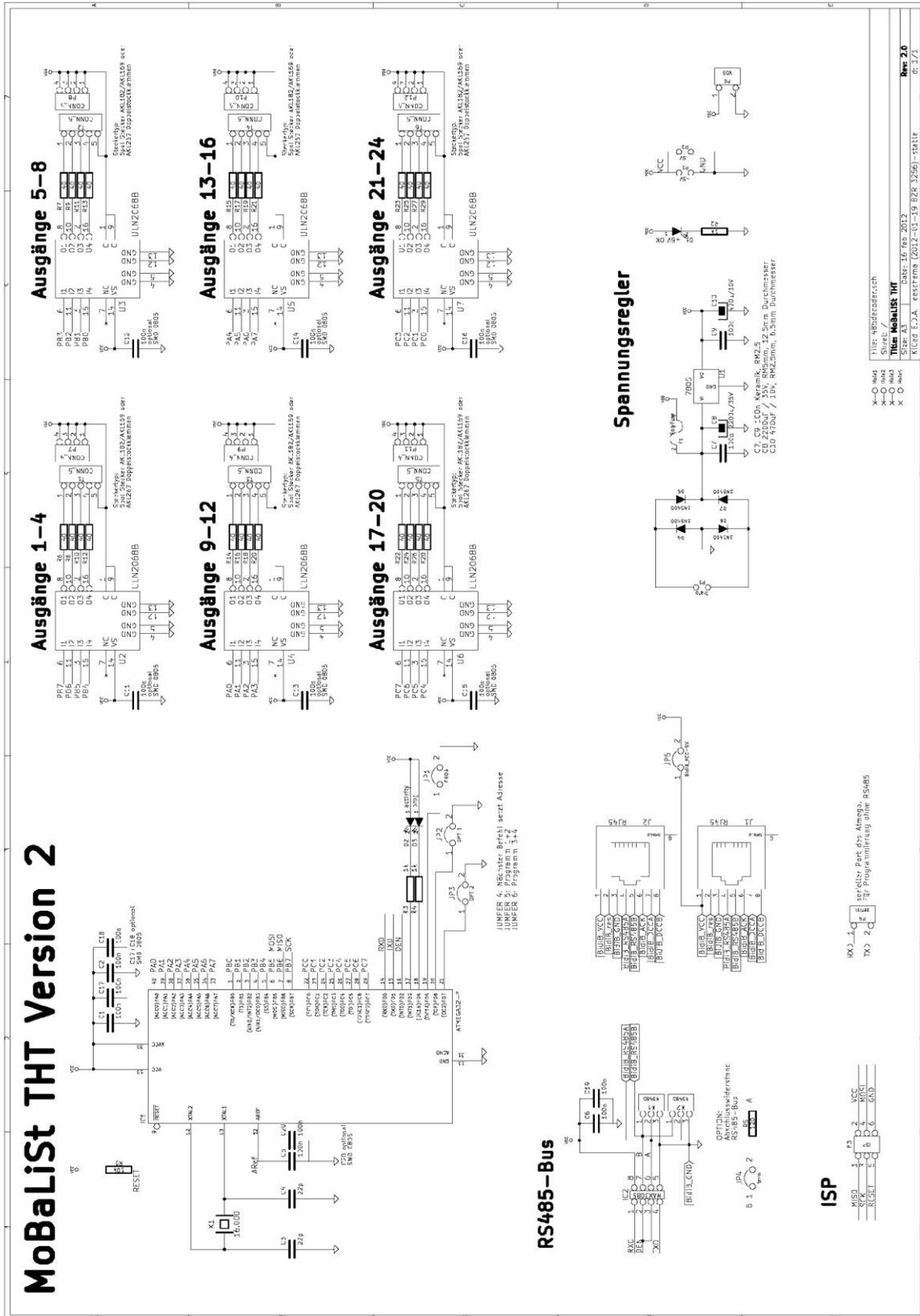
Hinweis:

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.



8. Sonderversion MoBaLiSt THT

8.1 Schaltung



MoBaLiSt THT Version 2

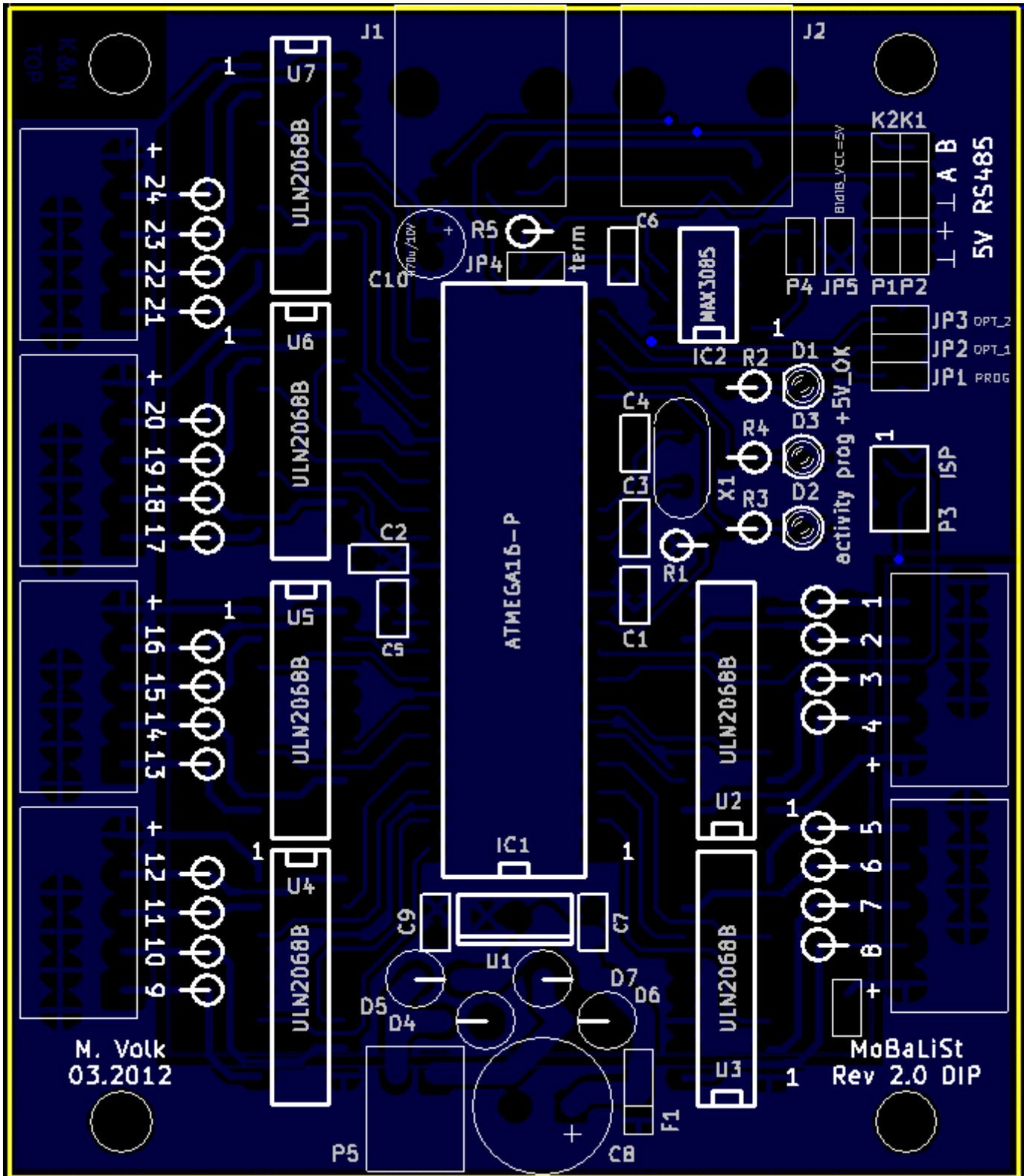
Spannungsregler

RS485-Bus

ISP

File: openDCC.dcc
 Date: 16. Feb. 2012
MoBaLiSt THT
 Star: A3 | Star: 16 Feb 2012
 Star: A3 | Star: 16 Feb 2012
 KICER E.J.A. - eStern (2012-13-19 BZR 1256) - t841e
Rev. 2.0
 0: 3/1

8.2 Bestückungsdruck



8.3 Stückliste

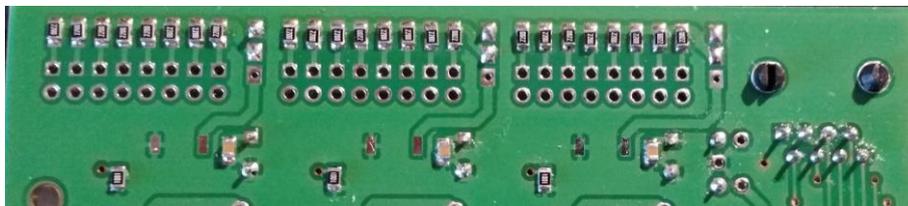
<u>MoBaLiSt BOM THT</u>				
Bauteil	Typ	Gehäuse	Anzahl	Notes
IC 1	ATMEGA32A-PU	40-DIP	1	
U2-U7	ULN2068B	16-DIP	6	
IC 5	MAX3085ECPA+	8-DIP	1	
U1	MC7805CTG	TO-220-3	1	
Q1	16 MHZ	HC49/US	1	
C1,C2,C5,C6,C7,C9	100nF	RM2.5	6	
C3,C4	22pF	RM2.5	2	
C10	470uF/10V 6.5mm	RM2.5	1	
C15	2200uF/35V 12,5mm	RM5	1	
C11-20	100nF	0805	10	*1
R6-R29	zB 1KOhm		24	*4
R2, R3, R4	1k		3	
R1	10k		1	
R5	120		1	*3
D2	LED gelb	3mm	1	
D3	LED rot	3mm	1	
D1	LED grün	3mm	1	
D4-D7	1N5400		4	
F1	Polyfuse 30V 3A		1	
P5	Spannungsversorgung		1	
Buchse zu P5			1	
T1-T6	Ausgangsbuchsen		6	
Buchse zu T1-T6			6	
<i>alternativ</i>				
T1-T6			12	
J1, J2	RJ45-Buchse		2	
*1) Optional, SMD				
*2) bei externe Ausgangs-Versorgung				
*3) ein Stück am Ende des RS485-Bus				
*4) je nach Ausgangsbeschaltung	LED 1KOhm, Lämpchen 40Ohm			

9. Aufbau MoBaLiSt SMD mit Printsteckern

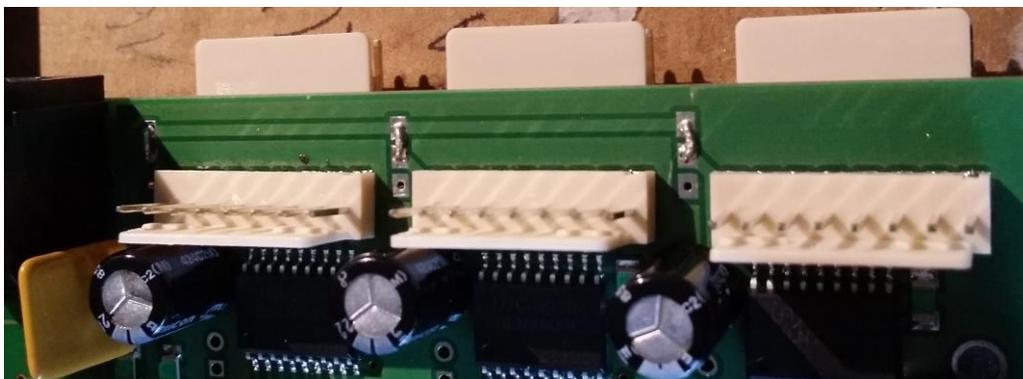
Eine interessante Variante der Anschlussmöglichkeiten ergibt sich durch den Einsatz von Printsteckern, z.B. PSS 254/8W (gewinkelt) und PSS 254/8G (gerade) in Verbindung mit dem Kupplungsleergehäuse PSK 254/8W in Crimptechnik. Auch diese Anschlussmöglichkeit kann man über Reichelt beziehen. Diese Stecker sind jeweils 8-polig und besitzen einen Verpol-schutz.

Um diese Technik anwenden zu können und dabei natürlich auch die vorgeschlagenen Vorwiderstände R1 bis R24 einbauen zu können, gehen Sie bitte nach folgenden Muster vor:

Zuerst löten Sie die Vorwiderstände komplett ein. Dann stecken Sie einen PSS 254/8W von unten in die vordere Reihe und verlöten diesen von oben. Die ersten 8 Vorwiderstände werden dabei vom PSS 254/8W verdeckt.

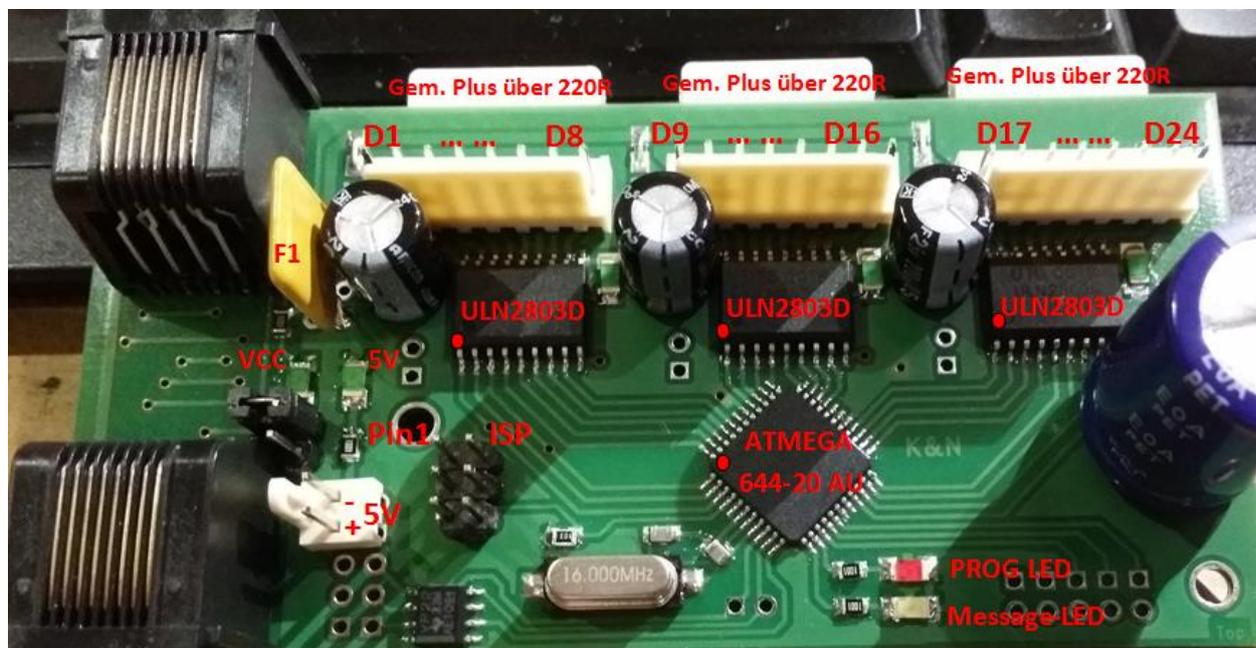


Nun stecken Sie von oben einen PSS 254/8G genau in die hintere Reihe und verlöten diesen von unten. Dies machen Sie bei allen 3 Gruppen, somit ergibt sich folgendes Bild:



Die gewinkelten Printstecker ergeben somit den gemeinsamen Pluspol über den jeweils eingesetzten Vorwiderstand und die stehenden Stecker bilden die Kontakte D1 bis D24.

Hier eine Ansicht vom MoBaLiSt64 unter Verwendung der direkten 5V Einspeisung.



10. Aufbau MoBaLiSt SMD 64-Bit Version

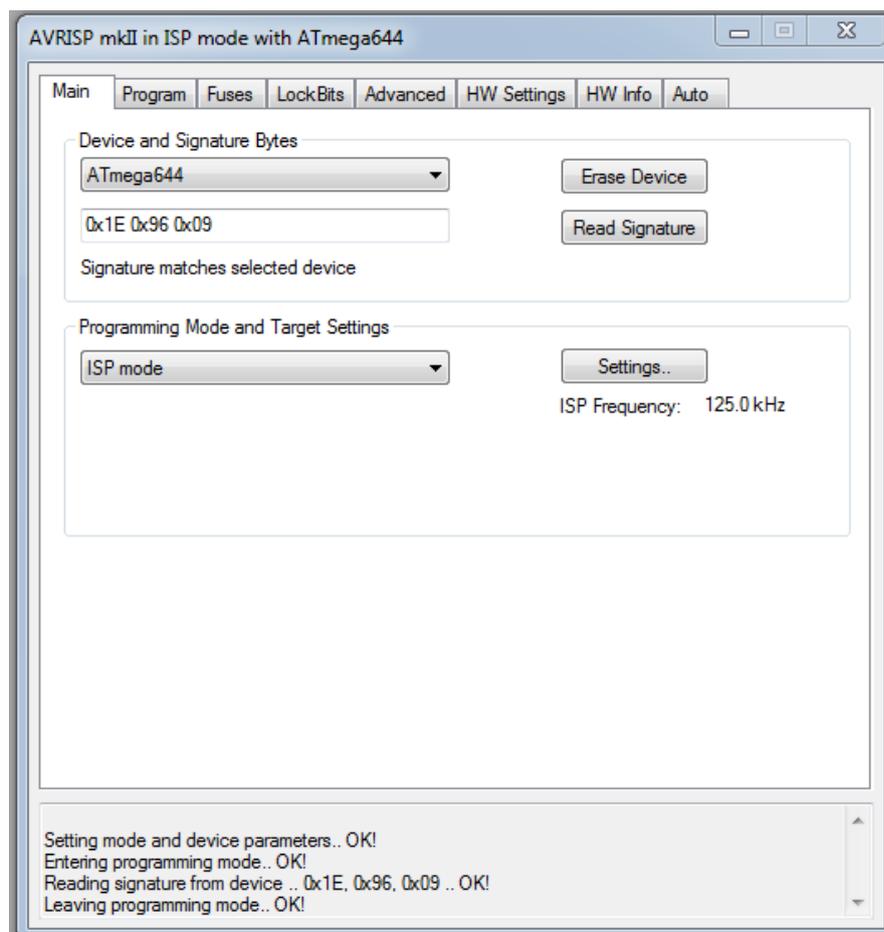
Der gesamte Modulaufbau entspricht dem Aufbau wie vorher beschrieben. Nur das Bauteil **IC1 ist ein ATMEGA 644-20 AU oder ATMEGA 644P-20AU**. Diesen kann man natürlich auch über www.reichelt.de beziehen. Also bitte im Warenkorb entsprechend austauschen.

10.1. Firmware beim MoBaLiSt SMD 64-Bit, ohne Bootloader

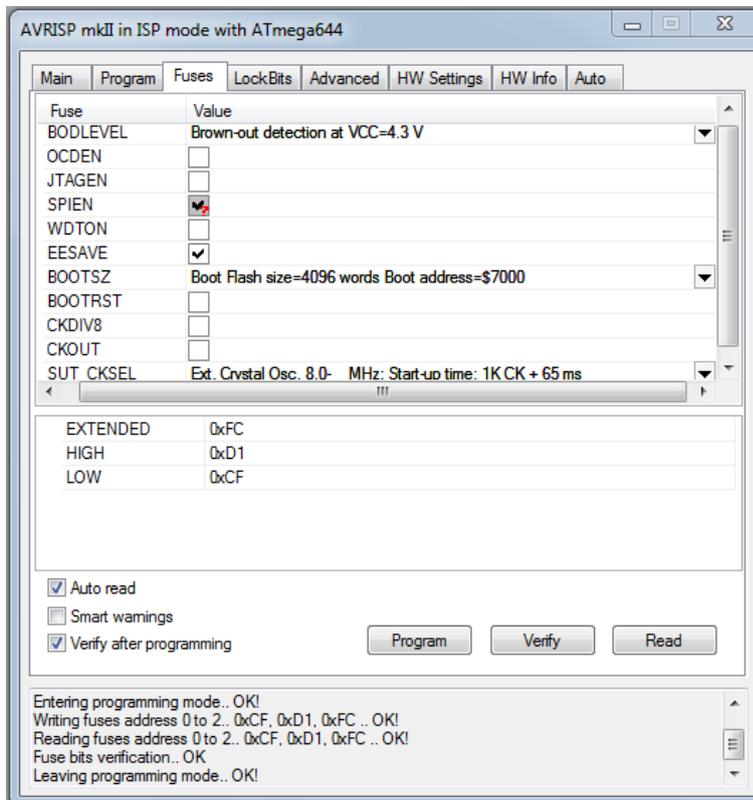
Wir beginnen so, wie im **Abschnitt 4.3** beschrieben.

Ab **Schritt 2: Einspielen der Software berücksichtigen wir folgende Änderungen**

Verbinden Sie das Modul mit der Stromversorgung und den Programmieradapter mit der ISP Stiftleiste auf der Platine, beachten Sie auf richtige Polung (PIN 1)
Starten Sie das AVR Studio und bauen Sie eine Verbindung auf. **Wählen Sie Atmega644** und lesen Sie die Signatur des Mikrocontrollers aus.
Der Programmiermodus ist auf **ISP Mode** einzustellen.



Wir lesen die Signatur aus und sollten besagtes Bild wie unter „Main“ erreichen.

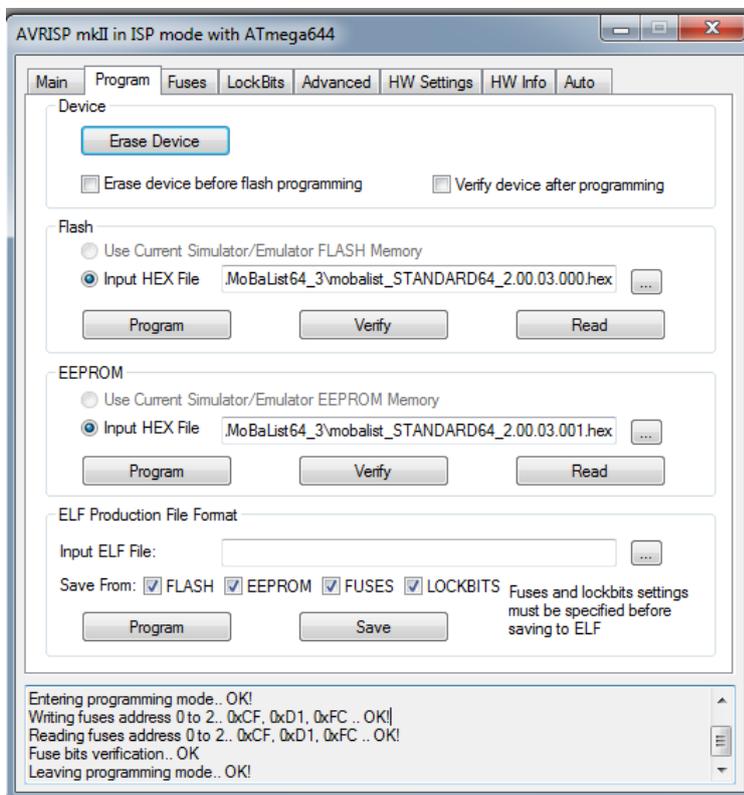


Nun stellen wir die Fuses ein:

Das **Highbyte** **0xD1**
 Das **Lowbyte** **0xCF**

Jetzt klicken Sie auf die Schaltfläche **Program** um die Fuses auf die Baugruppe zu übertragen.

Ist alles ordnungsgemäß eingestellt, so erhalten Sie **OK!**



Dann wechseln wir auf den Reiter „Program“

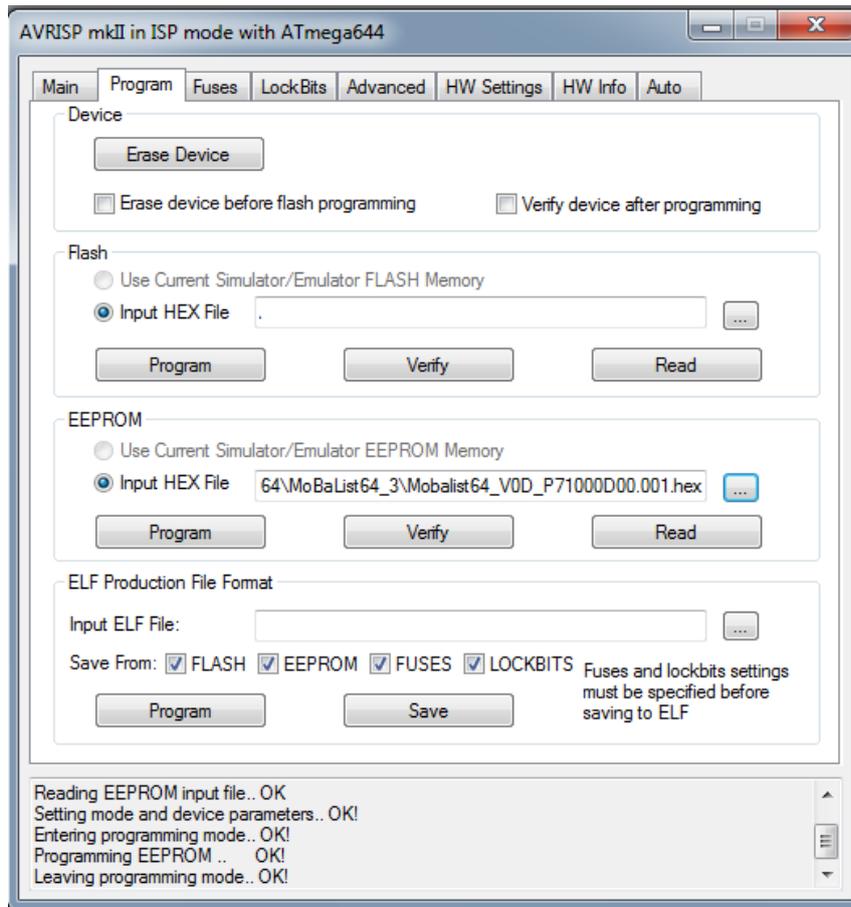
Jetzt wird die Datei „mobalist_Standard64_...000.hex“ für den Flash-Speicher und die Datei „mobalist_Standard64_...001.hex“ in das EEPROM übertragen.

Der Programmierablauf startet mit einem Klick auf die Schaltfläche **Program**.

Ist die Programmierung erfolgreich verlaufen erhalten Sie die unteren Meldungen im Fenster.

Die **LEDs D25** und **D26** blinken.

Erst wenn Sie die BiDiB-Seriennummer als zweiten Stepp in das EEPROM eingespielt wurde, hört das Blinken auf.



War das Einspielen der Firmware und Seriennummer erfolgreich, meldet AVR Studio „OK“ und die LED D26 flimmert.

10.2. Firmware beim MoBaLiSt SMD 64-Bit, mit Bootloader

Wird der Bootloader mit auf die Baugruppe geschrieben, kann zukünftig ein Update über die BiDiB-Tools erfolgen und kein externer Programmierer wird benötigt.

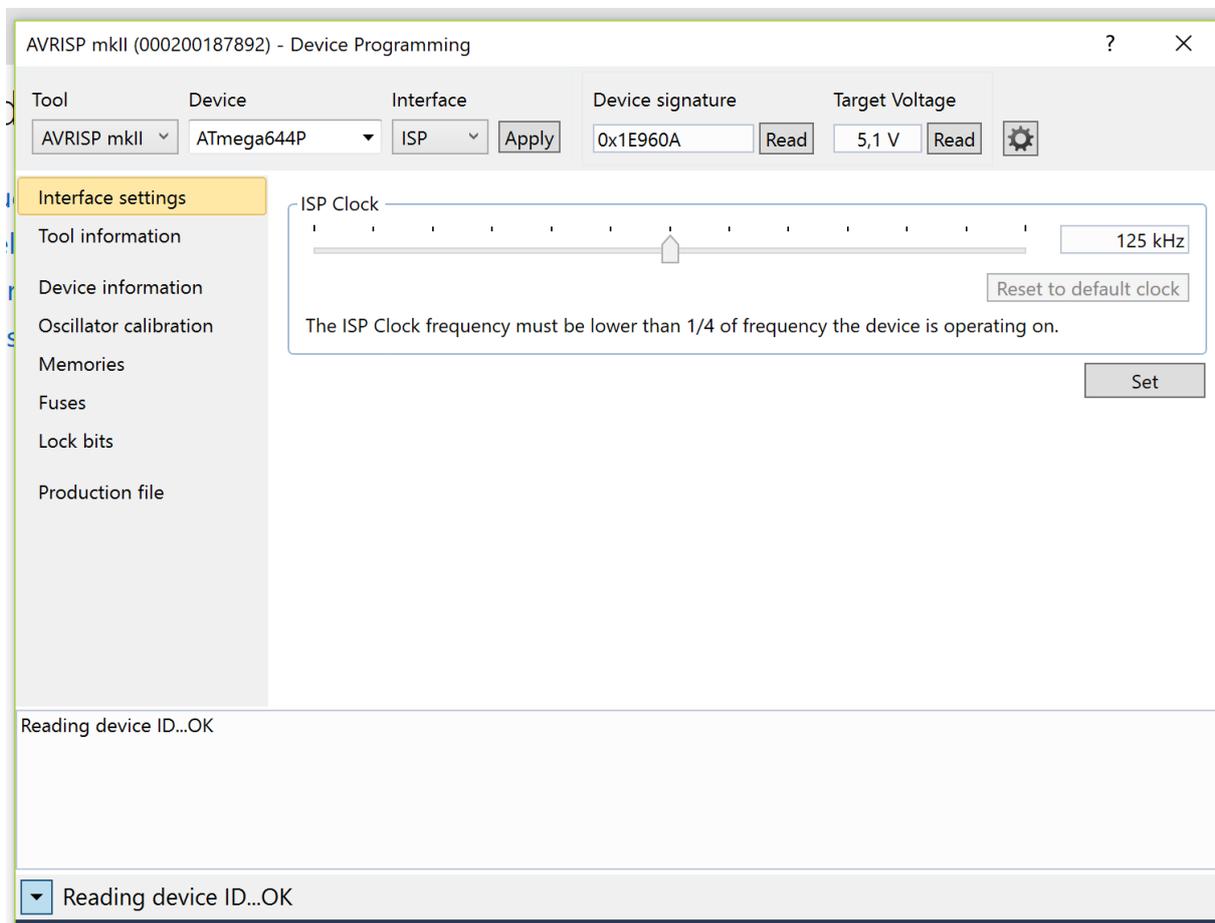
Wir beginnen so, wie im **Abschnitt 4.3** beschrieben.

Ab **Schritt 2: Einspielen der Software berücksichtigen wir folgende Änderungen**

Verbinden Sie das Modul mit der Stromversorgung und den Programmieradapter mit der ISP Stiftleiste auf der Platine, beachten Sie auf richtige Polung (PIN 1)

Starten Sie das AVR Studio und bauen Sie eine Verbindung auf. **Wählen Sie Atmega644** oder Atmega644P (je nach verwendeten Prozessor) und lesen Sie die Signatur des Mikrocontrollers aus.

Der Programmiermodus ist auf **ISP Mode** einzustellen. Hier mal gezeigt, am Beispiel vom Atmel Studio 7.



Dann wählen wir die Fuses aus. Bitte beachten Sie dabei, dass diese sich bewußt von der anderen Einstellung (wie unter Pkt. 9.1. aufgeführt) unterscheiden. Nach deren Einstellung, siehe Bild unten, werden diese per Auswahl „Program“ übertragen.

Danach sollte das Bild wie unten angegeben aussehen.

Tool	Device	Interface	Device signature	Target Voltage
AVRISP mkII	ATmega644P	ISP	0x1E960A	5,1 V

Interface settings	Fuse Name	Value
Tool information	<input checked="" type="checkbox"/> EXTENDED.BODLEVEL	Brown-out detection at VCC=4.3 V
Device information	<input checked="" type="checkbox"/> HIGH.OCDEN	<input type="checkbox"/>
Oscillator calibration	<input checked="" type="checkbox"/> HIGH.JTAGEN	<input type="checkbox"/>
Memories	<input checked="" type="checkbox"/> HIGH.SPIEN	<input checked="" type="checkbox"/>
Fuses	<input checked="" type="checkbox"/> HIGH.WDTON	<input type="checkbox"/>
Lock bits	<input checked="" type="checkbox"/> HIGH.EESAVE	<input type="checkbox"/>
Production file	<input checked="" type="checkbox"/> HIGH.BOOTSZ	Boot Flash size=4096 words Boot address=\$7000
	<input checked="" type="checkbox"/> HIGH.BOOTRST	<input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/> LOW.CKDIV8	<input type="checkbox"/>
	<input checked="" type="checkbox"/> LOW.CKOUT	<input type="checkbox"/>
	<input checked="" type="checkbox"/> LOW.SUT_CKSEL	Ext. Crystal Osc. 8.0- MHz; Start-up time: 16K CK + 65 ms

Fuse Register	Value
EXTENDED	0xFC
HIGH	0xDB
LOW	0xFF

Auto read
 Verify after programming

```

Starting operation verify registers
Verify register EXTENDED...OK
Verify register HIGH...OK
Verify register LOW...OK
Verify registers ... OK
    
```

Jetzt wechseln wir auf die Seite „Memories“ und übertragen dort die einzelnen Programmteile.

Als erstes übertragen wir in den Flash den BootLoader für den MoBaList64. Anschliessend folgt die Übertragung des ersten Teils mit der Endung ...000.hex ebenfalls in den Flash. Ist dies erfolgt übertragen wir die andere Datei ...001.hex in den EEPROM. Und zum Abschluss folgt die Datei mit der SerienNr. Ebenfalls in den EEPROM.

Ist dies erfolgreich verlaufen, sollte jetzt die rote LED D26 auf dem MoBaList deutlich flimmern und die Programmierung ist abgeschlossen.

Siehe dazu die folgenden Beispielbilder:

Tool	Device	Interface	Device signature	Target Voltage
AVRISP mkII	ATmega644P	ISP	0x1E960A	5,1 V

Interface settings

Tool information

Device information

Oscillator calibration

Memories

Fuses

Lock bits

Production file

Device

Erase Chip Erase now

Flash (64 KB)

C:\Modellbahn\Fichtelbahn\Updates\MoBaList64\Boot644P_BOARD_MOBALIST64_MOBALED_0.1.5.000.hex

Erase device before programming

Verify Flash after programming

Advanced

Program Verify Read...

EEPROM (2 KB)

Verify EEPROM after programming

Advanced

Program Verify Read...

Programming Flash...OK
Verifying Flash...OK

Tool	Device	Interface	Device signature	Target Voltage
AVRISP mkII	ATmega644P	ISP	0x1E960A	5,1 V

Interface settings

Tool information

Device information

Oscillator calibration

Memories

Fuses

Lock bits

Production file

Device

Erase Chip Erase now

Flash (64 KB)

C:\Modellbahn\Fichtelbahn\Updates\MoBaList64\mobalist_STANDARD64_2.02.4.000.hex

Erase device before programming

Verify Flash after programming

Advanced

Program Verify Read...

EEPROM (2 KB)

C:\Modellbahn\Fichtelbahn\Updates\MoBaList64\mobalist_STANDARD64_2.02.4.001.hex

Verify EEPROM after programming

Advanced

Program Verify Read...

Tool	Device	Interface	Device signature	Target Voltage
AVRISP mkII	ATmega644P	ISP	0x1E960A	5,1 V

Interface settings

Tool information

Device information

Oscillator calibration

Memories

Fuses

Lock bits

Production file

Device

Erase Chip Erase now

Flash (64 KB)

Erase device before programming

Verify Flash after programming

Advanced

Program Verify Read...

EEPROM (2 KB)

C:\Modellbahn\Fichtelbahn\Updates\MoBaList64\MoBaList64_7\Mobalist64_V0D_P71004000.001.eep

Verify EEPROM after programming

Advanced

Program Verify Read...

Programming EEPROM...OK
Verifying EEPROM...OK

Ab jetzt können Sie weitere Firmware-Updates ganz leicht über die BiDiB-Tools übertragen.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind wir sehr dankbar.

Auf die Bauanleitung und Software gibt es keine Haftung auf Schäden oder eine 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.de

Christoph Schörner

Am Dummersberg 26

D-91220 Schnaittach

support@fichtelbahn.de



© 2018 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.