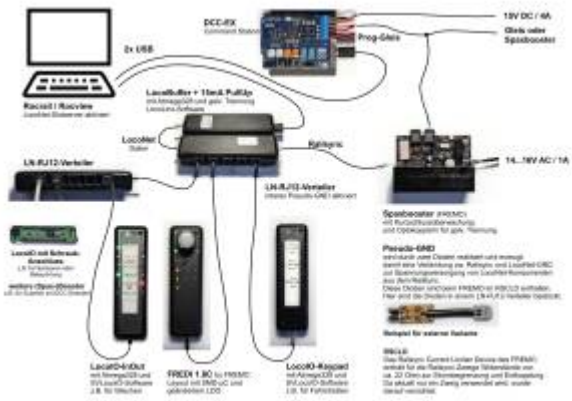
Digital-Hardware

Some parts are descibed in english here: Digital Hardware english

Hardware-Konfiguration

Ich beschäftige mich seit einiger Zeit mit der digitalen Steuerung der Modellbahn. Loks wurden mit einem Digitaldekoder ausgestattet, Technik für die Steuerung wurde gesucht, Schaltungen erstellt. Mit dem nachfolgenden Komponenten kann die Modellbahn jetzt digital gesteuert werden.



Dargestellte Komponenten sind

- USB Command Station.
- RJ12-Verteiler - 7-Ports für LocoNet® mit Pseudo-Ground
- SpaxBooster - FremoDCC-Projekt eines Boosters
- PC-Schnittstelle LocoBuffer für LocoNet®
- FREDI - FREMODCC-Fahrregler für LocoNet®
- LocoIO-Keypad für Taster-Eingaben über LocoNet®
- LocoIO-InOut für Ein-/Ausgaben über LocoNet®

Als Software kommt rocrail zum Einsatz, da dies Programm einen LocoNet®-Slotserver an Board hat, der für die Verwaltung der Fahrregler/Lokadressen notwendig ist. In Zentralen, die auch ein LocoNet® zur Verfügung stellen, ist der Slotserver dort vorhanden.

LocoNet® is a Trademark of Digitrax Inc., 2443 Transmitter Rd, Panama City, FL, 32404-3157, USA
Digtrax LocoNet Page

DCC-Hardware

Programmer

Mein erstes Projekt war ein Lok-Programmer DECPROG im Web-Archiv mit einem seriellen Anschluss. Ich habe diesen auf einer Universalplatine aufgebaut. Als Programmiersoftware kommt Prolok von Thomas Borrmann zum Einsatz. Dieses ist zwar schon etwas betagt, funktioniert aber auch unter Windows10. Mit Kenntnis der Dekoder-Variablen kann das Programm auch erweitert werden.

Command Station

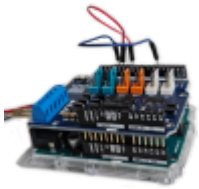
In einer Steuersoftware, wie RocRail, können die Command Stations zugefügt und z.B. mit dem Demoprogramm und dem Bildschirm-Throttle getestet werden.

SPROG3



Als Command Station hatte ich zu Beginn den SPROG3 eingesetzt. Diese ist über USB an den PC angeschlossen und erscheint als serielles Gerät im Gerätemaneger von Windows. Die DCC-Signale werden an das Gleis geführt, oder als RailSync an einen Booster, der mit dem Gleis verbunden ist. Weiteres hierzu auch unter RSCLD.

Arduino DCC



Auf Basis des Arduino Uno wurde eine Command Station aufgebaut. Dazu wird zusätzlich ein MotorShield benötigt. Die Verdratung und die Einbindung ist bei rocrail beschrieben.
Für die Software habe ich die Beispielprogramme BaseStation-Classic von DCC-EX und die Ursprungsversion von DCCpp getestet. Die zugehörige Bibliothek DCCpp kann über die Arduino-IDE eingebunden werden.

In aktuellen Versionen von Rocrail (ab Version 7014) wurde eine Option für den **Emergency-Stop** im Slot-Server eingebaut, so dass im Zusammenhang mit dem FREDI-Fahrregler und dem Slotserver im Rocrail kein Änderungen an einer Datei im DCC-Ex nicht mehr notwendig sind. Forums-Beitrag

Es ist zu beachten, dass nur Lok-Dekoder mit **128 Fahrstufen** unterstützt werden. Im rocrail sind dazu entsprechende Einträge vorzusehen, damit der Fahrregler entsprechend „dispatched“ wird. Der Forumsbeitrag war mir dabei eine gute Hilfe, als es anfangs nicht klappte. Wichtig ist hier die Protokoll-Einstellung NMRA-DCC.

Booster



Für den Anfang wäre es sicher nicht notwendig, einen Booster zu bauen, da ja die Commandstation 3A Fahrstrom liefert, aber ich wollte das System komplett aufbauen, um nicht später auf Systemschwierigkeiten zu treffen. Ich habe mit dafür den SpaXbooster des FremoDCC ausgesucht. Die PIC-Firmware ist hier zu finden. Da die Leiterplatten-CAD-Daten frei verfügbar waren (nicht die Produktionsdaten), habe ich diese in das freie CAD-System DesignSpark übertragen incl. des 100nF-Kondensators (Lok-Sound-Problem), für den auch ein Keramik-Typ ausreichen sollte. Ich musste statt des 100nF-Kondensators einen 220nF-Typ einsetzen, da das Signal zur Kurzschlusserkennung Störspitzen aufwies und nicht richtig funktionierte.

Anstelle des PIC16F628 nutze ich den PIC16F628A und den in der A-Serie genaueren internen 4MHz-Oszillator, was den Resonator spart. Dies muss aber über die Fuses (Config-Bits) beim Programmieren eingestellt werden. Da das Programm kurz ist, reicht auch ein PIC16F627(A).

Der Booster realisiert die galvanische Trennung zwischen mehreren Segmenten aber auch zwischen Loconet/Railsync und dem Gleisstromkreis.

Schaltplan
Bestückungsplan
Stückliste beim FREMO
Bestellmöglichkeit der Leiterplatte bei QSH Park für private Nutzung
Bestellmöglichkeit der Leiterplatte für bei Aisler für private Nutzung
Eine kommerzielle Nutzung ist untersagt.

Weichen-Dekoder



Als Weichendekoder habe ich derzeit die DCC-Hardware von digital-bahn.de, den Servo-Dekoder SAnD-Ei aufgebaut. An diesen habe ich einen kleinen Servo mit selbst gebautem Weichenantrieb angeschlossen. Die Versorgung erfolgt aus der Fahrstromversorgung, die an der Schiene anliegt.

Das Projekt SAnD-Ei ist zwar eingestellt, aber die Daten zur Hardware und die Hex-Dateien liegen auf der Webseite zum Download für eine mögliche eigene Verwendung.

DCC-Dekoder



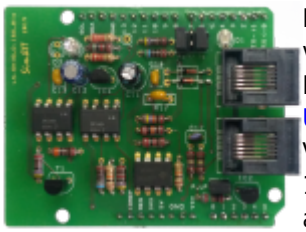
Für den Umbau der Beleuchtung in Personenwagen und für die Ansteuerung der LEDs bei Gebäuden und Signalen wurden DCC-Dekoder auf der Basis des ATtiny85 entwickelt. Die Entwicklung begann mit Digispark-Boards und Zusatzplatinen für das DCC-Signal und Ansteuertransistoren.

Eingesetzt wird dieser Dekoder u.a. bei einer neu konstruierten Lichtleiste für SCHICHT-Rekowagen, in einem Pack-Wagen und für die selbst erstellten EZMG-Signale.

Für die Synchronisation des Blinkens von Vor- und Hauptsignal wurde der DCC-Dekoder mit ATmega328 und sechs Ausgängen erweitert.

LocoNet-Hardware

LNShield-ISO für Arduino



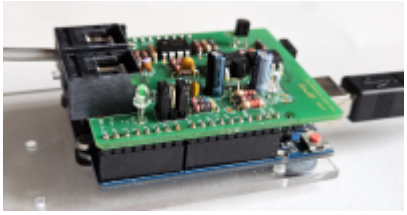


Für Experimente mit dem Arduino und den von MRRWA veröffentlichten Bibliotheken und Beispiel-Sketchen habe ich den LocoNet-Teil der Schaltung des unten beschriebenen LocoBuffer-USB DIY auf eine Arduino-Shield-Platine gebracht. Die Versorgung der LocoNet-Seite erfolgt aus dem RailSync. Die 15mA-LocoNet-Stromquelle (Pull Up) ist über eine Steckbrücke aktivierbar. Die Signalführung zum Arduino ist wie beim FremoLNShield ausgeführt, so dass für Tx das Signal D6 oder D7 verwendet werden kann. Rx liegt auf D8. Eine weitere LED kann optional für weitere Signalisierungen genutzt werden, diese ist aktuell nicht bestückt.

Ich habe die erste Platine ohne die Stacking Header aufgebaut, da erste Tests als **LocoBuffer** durchgeführt wurden. Dafür habe ich den Beispiel-Sketch LocoLinx von MRRWA geladen. Aus diesen Experimenten entstand dann der LocoBuffer-ProMini.

Die serielle Schnittstelle wird auf **57600** Baud eingestellt. Der originale Arduino UNO mit ATmega16U2 für USB unterstützt Handshake, das wurde aber nicht im Zusammenhang mit dem Shield und LocoLinx getestet. Ansonsten muss Handshake **aus**geschaltet werden, da z.B. der USB-Seriell-Wandler CH340 Handshake nicht unterstützt. Diese Einstellungen erfolgen im Betriebssystem und auch in der Steuer-Software.

Schaltplan
Bestückungsplan
Stückliste
Bestellmöglichkeit der Leiterplatte für Eigenbau bei Aisler
Eine kommerzielle Nutzung ist untersagt.



Um die Betriebsfähigkeit auch bei geringer Eingangangsspannung zu gewährleisten, wurde anstelle des 78L05 ein LDO LP 2950 ACZ5,0 eingesetzt. Dieser hat außerdem einen deutlich geringeren Ruhestrom. Als Gleichrichter-Diode kann anstelle der 1N4148 eine Schottky-Diode verwendet werden, z.B. BAT46. Bei den LEDs sollten Typen verwendet werden, die mit dem Vorwiderstand 2,2kOhm bei 5V noch ausreichend Intensität haben. Bei der in der Stückliste angegebenen grünen LED reichen für die statische Anzeige des 5V-Reglers bereits 15k als Vorwiderstand aus, um eine ausreichende Intensität zu erzeugen. Für die LocoNet-Impulsanzeige wurden die 2,2k wie im Schaltplan verwendet.

LocoBuffer

Damit Fahrregler und andere Bausteine über LocoNet mit einem PC als Zentrale kommunizieren können, wird eine Schnittstelle benötigt, welche die Signale von und zum PC zwischenspeichert und die Kommunikation mit dem LocoNet realisiert. Der erste Entwurf des **LocoBuffer** stammt von John Jabour. Diese Seite ist nur noch über das Web-Archiv verfügbar.
Dieser hatte keine galvanische Trennung und wird über die RS232-Schnittstelle angeschlossen. Auf dieser Idee basieren viele DIY-Projekte und auch kommerziell hergestellte Produkte, bei RR-CirKits gibt es einen LocoBuffer-NG zu kaufen.

Ich habe aus vorigen Entwicklungen und Erfahrungen eine neue Platine entwickelt, die einen ATmega328 und einem USB-Seriell-Wandler enthält. Alles passt in das FREDI-Gehäuse.

LocoBuffer-ProMini



Die aktuelle LocoBuffer-Variante ist bis auf Steckverbinder und LEDs in SMD ausgeführt und wurde manuell bestückt. Die Anbindung an das LocoNet wurde schaltungstechnisch vom LNShield-ISO und den bisherigen LocoBuffer-Varianten übernommen und realisiert über Optokoppler eine galvanische Trennung zwischen PC und Loconet. Wie beim LNShield des FREMO wurde die Konfigurationsmöglichkeit für Tx auf D6 oder D7 über Jumper oder Drahtbrücke beibehalten, im Sketch ist standardmäßig D6 dafür vorgesehen. Rx ist mit D8 verbunden.
Der µC-Teil des LocoBuffer wird aus der USB-Schnittstelle versorgt, der LocoNet-Teil aus dem RailSync-Signal. Für die RailSync-Spannungsversorgung gilt das gleiche wie beim FREDI-Fahrregler, es ist das Pseudo-GND wie beim RSCLD notwendig.

Eine **15mA-LocoNet-Stromquelle** (Pull Up) ist über Jumper **JP4** aktivierbar. Vier LED signalisieren die Spannungsversorgung aus RailSync und USB sowie die Kommunikation auf LocoNet und USB-Rx-Tx. Um die Betriebsfähigkeit auch bei geringer Eingangangsspannung zu gewährleisten, sollte der LDO (siehe Stückliste) anstelle des 78L05 bestückt werden, dieser hat auch einen deutlich geringeren Ruhestrom.

Als Software wird der Sketch LocoLinx von MRRWA verwendet. Tx ist als D6 und Rx als D8 definiert. Mit der Realisierung einer Verbindung vom Port D9 des Arduino und dem CTS#-Signal des FT230XS kann auch das im Sketch vorgesehenen **Hardware-Handshake** verwendet werden. Dies wird über Jumper bzw. Drahtbrücke **JP3** aktiviert. In der LocoLinx-Software ist CTS standardmäßig aktiv und wird über die Auslastung des Pufferspeichers gesteuert. Die Baudrate der seriellen Schnittstelle ist **57600**.

Je nach Beschaffbarkeit können Optokoppler in unterschiedlichen Bauformen verwendet werden. Dafür gibt es zwei Leiterplattenvarianten.

Variante 2.1 mit Optokopplern z.B. Everlight 6N136S/EL4502S oder Avago HCPL-4502 im SMD-Gehäuse mit DIP8 Gull Wing Pins
Schaltplan
Bestückungsplan
Stückliste
Bestellmöglichkeit der Leiterplatte für Eigenbau bei Aisler

Eine kommerzielle Nutzung ist untersagt.

Variante 2.2 für alternative Optokoppler HCPL-0453 im kleinen SOT-8-Gehäuse, alle anderen Bauteile sind gleich zur Variante 2.1
Schaltplan
Bestückungsplan
Stückliste
Bestellmöglichkeit der Leiterplatte für Eigenbau bei Aisler
Eine kommerzielle Nutzung ist untersagt.

Weiter unten ist die Vorgängerversion 1.0 aufgeführt, bei der die Anbindung an das LocoNet mit THT-Bauteilen realisiert ist.

Gilt für die ATmega-Varianten:
Der ATmega328 muss zuerst mit einen Bootloader programmiert werden, um danach als Arduino Pro Mini benutzt zu werden. Dazu ist ein 6-poliger ISP-Steckverbinder für einen AVR-Programmer z.B. von Pololu vorgesehen. Siehe Bootloader programmieren.
Die anschließende kann Programmierung kann ebenfalls über den über den ISP-Programmer oder auch über den USB-Anschluss mit der Arduino-IDE wie für einen Arduino Pro Mini erfolgen.
Bei den Steckbrücken für CTS und LN-Tx (D6) reicht eine Drahtbrücke. Der Taster S1 kann entfallen, dieser ist nur für Testzwecke vorgesehen.

Voraussetzung für die Stromversorgung aus dem RailSync ist wie beim Betrieb des FREDi die Verwendung des RSCLD oder zumindest der beiden Dioden für das Pseudo-GND.
Eventuell wird diese Verbindung aber bereits durch eine Zentrale realisiert.

In Windows11 sind die Treiber für den FTDI-USB-Seriell-Wandler über das Windows-Update installierbar, ansonsten sind diese auch auf der FTDI-Seite verfügbar.
Auch am Raspberry Pi wird der LocoBuffer automatisch erkannt.

Die Einstellungen im Rocrail sind hier dargestellt.

Diese Varianten hatte ich davor aufgebaut bzw. im Einsatz

HDM09USB



Mein erster LocoBuffer war ein Bausatz von H. Deloof, der HDM09USB. Vor dem Anschließen muss der Microchip-Treiber installiert werden, da HDM09USB einen PIC als USB-Seriell-Adapter verwendet. Nach dem Anschließen am PC erscheint das Gerät als USB-Seriell-Port im Gerätemanager und über die Nummer der COM-Schnittstelle können Daten von der PC-Software ans LocoNet gesendet bzw. vom LocoNet empfangen werden. Der LocoBuffer hat auch als Option die 15mA-LocoNet-Stromquelle, auch LocoNet-Pullup genannt. Diese ist aktiviert, da keine LocoNet-Zentrale verwendet wird.

Bei der Inbetriebnahme gab es jedoch Kommunikations-Probleme. Die Software meldete einen Fehler. Die Rückfrage beim Hersteller lieferte die Erklärung. Eine Eigenart des Microchiptreibers ist wohl, dass das Handshake im USB-Treiber realisiert ist und nicht die RTS-CTS-Signalisierung der virtuellen COM-Schnittstelle nutzt. Handshake muss also im Treiber und in der Software ausgeschaltet sein.

LocoBuffer-USB DIY



Da der HDMUSB09 keine galvanische Trennung vorsieht, habe ich mich nach anderen Lösungen umgesehen. Dabei bin ich auf den LocoBuffer-II von RR-CirKits gestoßen, wo auch Schaltung und HEX-Code für den PIC veröffentlicht sind. Der hier gezeigte LocoBuffer basiert auf dort gezeigten Schaltungen und einem USB-Seriell-Wandler auf einem Carrier-Board von Pololu. Damit sind keine SMD-Bauteile eingesetzt und die Bestückung ist einfach. Das ganze passt dann auch in das gleiche Gehäuse wie

beim Fahrregler FREDI. Der µC-Teil des LocoBuffer wird aus der USB-Schnittstelle versorgt, der LocoNet-Teil aus dem RailSync-Signal. Die **15mA-LocoNet-Stromquelle** (Pull Up) ist über eine Steckbrücke aktivierbar. Über LEDs werden Spannungsversorgung USB und RailSync sowie die LocoNet-Aktivität angezeigt. Für die RailSync-Spannungsversorgung gilt das gleiche wie beim Fahrregler, es ist das Pseudo-GND wie beim RSCLD notwendig. Auch bei Rocrail ist ein Version eines LocoBuffers beschrieben, der GCA85.

Schaltplan
Bestückungsplan
Stückliste
Firmware rr-cirkits LocoBuffer-II
Firmware rocrail GCA85 entspricht LB163.hex
Bestellmöglichkeit der Leiterplatte für Eigenbau bei Aisler
Eine kommerzielle Nutzung ist untersagt.

Der LocoBuffer wird mit JP1 auf **57600** Baud eingestellt JP2 und JP3 bleiben offen. Im System wird für die COM-Schnittstelle auch diese Baud-Rate festgelegt und Hardware-**Handshake ein**geschaltet. Diese Einstellungen sind dann auch in der Software-Zentrale z.B. Rocrail einzustellen.

Um die Betriebsfähigkeit auch bei geringer Eingangangsspannung zu gewährleisten, wurde anstelle des 78L05 ein LDO LP 2950 ACZ5,0 eingesetzt. Dieser hat außerden einen deutlich geringeren Ruhestrom.

LocoBuffer-ProMini V1.0



Die ist die erste Variante, die einen ATmega328 verwendet und als Software den Sketch LocoLinx von MRRWA nutzt. Die LocoNet-Schaltung wurde vom LocoBuffer-USB übernommen und ist in THT-Technik ausgeführt. Die ATmege328-Seite mit dem USB-Seriell-Wandler ist in SMD-Technik ausgeführt.
Es gelten die beim LocoBuffer-ProMini 2.x angeführten Bemerkungen zu den Jumpern und der seriellen Schnittstelle.

Um die Betriebsfähigkeit auch bei geringer Eingangangsspannung zu gewährleisten, kann ein LDO LP 2950 ACZ5,0 eingesetzt werden. Dieser hat außerdem einen deutlich geringeren Ruhestrom als ein 78L05.

Schaltplan
Bestückungsplan
Stückliste
Bestellmöglichkeit dieser tw. THT-Leiterplatte für Eigenbau bei Aisler
Eine kommerzielle Nutzung ist untersagt.

LocoIO

Der **LocoIO** entstammt einer Idee von John Jabour (locobuffer.com). Diese Seite ist nur noch über das Web-Archiv verfügbar.

LocoIO-Keypad



Für die Weichensteuerung und Sensoren will ich die Idee des **LocoIO** verwenden. LocoIO wird von Rocrail unterstützt.
Es gibt davon mehrere Nachbauten, u.a. den bei Rocrail vorgestellten GCA50. Ein handliches Gerät für den Einstieg stellt der GCA123 dar, der 16 Tasten bietet.



Ich habe die Schaltung etwas modifiziert. Die Bestückung wurde so geändet, dass Komparator und Schalttransistor, die der LocoNet-Ansteuerung dienen, in die Nähe des RJ12-Steckverbinders gebracht wurden. Die Reihenfolge der Taster S09...S16 wurde gedreht, so dass es zur LocoIO-Programmiermaske im rocrail passt. Gegenüber von S01

befindet sich jetzt S09, S09 bedient den Port 9 usw.. Als Pullup für die Taster wurden Widerstandsnetzwerke eingesetzt. Mit diesen Änderungen wurde das Keypad aufgebaut. Die Leiterplatte kann vor der Bestückung als Bohrschablone für Taster und LEDs genutzt werden.

Die Leiterplatte passt in das FREDI-Gehäuse. Die Versorgung erfolgt aus dem RailSync-Signal. Anstelle des 78L05 kann ein LDO LP 2950 ACZ5,0 eingesetzt werden. Dieser benötigt eine geringere Eingangsspannung und hat auch einen geringeren Ruhestrom.

Schaltplan
Bestückungsplan
Stückliste
Bestellmöglichkeit der Leiterplatte für Eigenbau bei Aisler
Eine kommerzielle Nutzung ist untersagt.
Firmware bei Rocrail or LocoIO

LocoIO-Keypad-Arduino



Von Daniel Guisado ClubNCaldes gibt es einen Arduino-Sketch für LocoIO der auf der Hardware Arduino-Pro-Mini aufbaut. Link zum Sketch weiter unten.



Bei der Inbetriebnahme habe ich festgestellt, dass im Sketch nicht alle Funktionen der PIC-Vers-148 umgesetzt wurden, wie sie im GCA123/GCA50 verwendet werden. Einige Optionen, die in der Sensor-Funktion fehlte, habe ich nachträglich in den Sketch eingefügt.

- Option „Active High“, die ich für die Weichensteuerung mit Rocrail nutze
- Eine weitere Funktion beim Ausgang entsprach ebenfall nicht der PIC-Version. Bei der PIC-Variante wird der Port invertiert geschaltet, wenn Contact2 aktiviert ist. Dies wurde auch nachträglich im Sketch eingefügt.
- Es wurde auch die Blink-Funktion ergänzt.

Für meine Anwendung sind beide LocoIO-Keypad-Varianten gleichwertig einsetzbar. Link zum Sketch weiter unten.

Ich habe die Schaltung und Bestückung der Anschlusseite an das LocoNet vom LocoIO-Keypad übernommen und den PIC gegen dem ATmega328 getauscht. Dafür habe ich den SMD-Typ im TQFP-Gehäuse verwendet, da der DIL-Typ lt. Webseite nicht für neue Designs verwendet werden soll. Außerdem konnte ich auf einen vorhandenen Schaltungsentwurf zurückgreifen.

Alles passt wie bisher in das FREMO-FREDI-Gehäuse des Fahrtreglers. Beim Aufbau ist folgendes zu beachten:

- Der ATmega muss zuerst mit einen Bootloader programmiert werden. Dazu ist ein 6-poliger Steckverbinder für einen AVR-Programmer z.B. von Pololu vorgesehen. Die Programmierung mit einem Arduino als Programmer ist ebenfalls mäglich. Siehe Bootloader programmieren. Dort sind die notwendigen Verbindungen beschrieben
- Die ATmega-Eingänge sind im Sketch auf PULLUP definiert worden, so dass die beiden Widerstandsnetzwerke RN1 und RN2 (in PCB-Rev. 1.3ff) nicht notwendig sind, wenn keine langen Leitungen verwendet werden. Die internen Pullup haben lt. Datenblatt einen Wert 20 … 60 kOhm. Auch bei der Verwendung der Ports als Ausgang können diese entfallen. Diese sollten eingesetzt werden, falls Taster/Sensoren extern mit längerer Leitung angeschlossen werden sollen.
- Für die Programmierung mit einem USB-Seriell-Adapter und der Arduino-Oberfläche ist ein sechpoliger FTDI-Steckverbinder wie am Arduino-Pro-Mini vorhanden.
- Die LEDs sollten vor dem FTDI-Steckverbinder in das Gehäuse eingepasst und eingelötet werden. Die

Anschlussdrähte wurden länger gelassen und dann entsprechend gebogen. Die überstehenden Anschlussdrähte der LEDs sind dann stark zu kürzen, damit sie den Steckverbinder nicht stören.
- Für den Programmierfall mit USB-Seriell-Adapter ist eine Steckbrücke **FTDI 5P** vorgesehen. Diese muss entfernt werden, wenn das Keypad gleichzeitig am Loconet und am PC über den FTDI-Stecker verbunden ist.
- **Achtung:** Im Fall der Verbindung von PC und LocoNet ist keine galvanische Trennung mehr vorhanden. Bei einem Laptop ohne weiter Masse/Schutzerde-Verbindung ist dies eher unproblematisch. Durch die Möglichkeit, den seriellen Anschluss mit USB-Seriell-Adapter auch im Betrieb am PC zu lassen, kann mit einem Terminal-Programm die Kommunikation auf dem LocoNet beobachtet werden, da die Software das im Debug-Modus an der seriellen Schnittstelle ausgibt.
- Um den Betrieb auch bei geringer Eingangsspannung zu gewährleisten, wurde anstelle des 78L05 ein LDO LP 2950 ACZ5,0 verwendet.

Variante nur Taster
Schaltplan V 1.3
Bestückungsplan V 1.3
Stückliste
Bestellmöglichkeit dieser frühen Leiterplattenvariante für Eigenbau bei Aisler
Eine kommerzielle Nutzung ist untersagt.

Variante Taster / alternativ LEDs oder Schraubklemme
In dieser Variante ist die LocoNet-Anbindung weiterhin in THT ausgeführt, aber die Möglichkeit für Schraubklemmen oder LEDs mit Vorwiderständen ist alternativ zu den Tastern vorhanden, wie bei der nachfolgend beschriebenen kompletten SMD-Variante **LocoIO-A-InOut**. Damit sind nur wenige SMD-Bauteile um den ATmega328 herum verwendet worden und wenn Ausgänge genutzt werden sollen, die Vorwiderstände.
Diese Baugruppe kann genauso wie die Variante 1.3 genutzt werden. Die SMD-Variante hat zusätzlich noch Anschlüsse für 5V und GND herausgeführt. Das war aus Platzgründen bei dieser Variante nicht möglich.

Schaltplan V 1.4
Bestückungsplan V 1.4
Stückliste
Bestellmöglichkeit dieser Leiterplattenvariante für Eigenbau bei Aisler
Eine kommerzielle Nutzung ist untersagt.

LocoIO-A-InOut





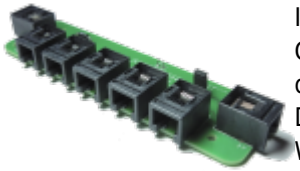
In der aktuellen Version sind weitere Bauteile als SMD ausgeführt. Außerdem kann statt des Tasters ein zweipoliger Schraubanschluss verwendet werden. Je ein Schraubanschluss pro Seite stellt 5 V für externe Geräte zur Verfügung. Durch die Pull-Up-Widerstände kann das externe Gerät auch Open-Kollektor/Drain-Ausgänge verwenden, z.B. auch entsprechende Optokoppler-Ausgänge.
Zum Schutz bei der Verwendung als Ausgänge sind Serien-Widerstände vorgesehen. Durch geeignete Anpassung dieser Widerstände ist auch der direkte Anschluss von LEDs möglich. Die Pull-Up-Widerstandsnetzwerke können bei der Nutzung als Ausgang entfallen.
Alle weiteren Bemerkungen siehe im obigen Abschnitt zum LocoIO-Keypad-Arduino.

Schaltplan V 1.5 SMD
Bestückungsplan V 1.5 SMD
Stückliste
Bestellmöglichkeit der Leiterplatte für Eigenbau bei Aisler
Eine kommerzielle Nutzung ist untersagt.

Links:
Arduino Sketch ClubNCaldes/SVLocoIO
Arduino Sketch Silverailscolo/SVLocoIO
ergänzter Arduino-Sketch forked from ClubNCaldes/SVLocoIO

Weitere Quellen:
https://www.eisenbahnfreunde99.de/index.php/tecnik/elektronik/loconet/ko-loconetio
https://wiki.rocrail.net/doku.php?id=gca50_an-de

LN-RJ12-Verteiler



Ich habe einen RJ12-Verteiler gebaut, der optional auch die zwei Dioden für das Pseudo-GND (siehe RSCLD) und einen Widerstand wie hier enthält. Der 22R-Widerstand realisiert die im RSCLD vorhandene Strombegrenzung. Siehe auch LNC162 von Digitrax.
Das Pseudo-GND kann über eine Steckbrücke zugeschaltet werden. Ein optionaler Widerstand (ich habe ihn nicht bestückt) kann bei Bedarf mit einem Jumper überbrückt werden. Das Pseudo-GND darf nur einmal in einem RailSync-Versorgungsabschnitt vorhanden sein.
Die Schaltung passt in das FREDI-Gehäuse.

Schaltplan
Bestückungsplan
Bestellmöglichkeit der Leiterplatte für Eigenbau bei Aisler
Eine kommerzielle Nutzung ist untersagt.

Low-Profile-RJ12-Steckverbinder folgender Hersteller wurden getestet:

- Econ Connect MEB6/6PL bei Voelkner
- MOLEX 95501-2661 6P6C bei diversen Distributoren

Fahrregler FREDI







Auf den Seiten des FremoDCC sind viele Informationen des FREMO-Fahrreglers FREDI von Olaf Funke zu finden. Die Version, die ich gebaut habe, ist Version 1.8. Hierfür waren Leiterplattendaten verfügbar. Aufgrund eines geschilderten Problems mit der Versorgungsspannung habe ich mir die CAD-Daten angesehen und festgestellt, dass die Leiterbahnführung der Masse und auch die Position des Kondensators C1 nicht optimal sind, was eine mögliche Ursache für Probleme bei der Versorgungsspannung sein könnte. Ich habe unter Beibehaltung der Bauteile die Leiterbahnen der Masse angepasst. Außerdem habe ich bei meiner Bestückung einen anderen LDO (LP 2950 ACZ3,3) eingesetzt, der nach Datenblatt kein Problem mit niedrigem ESR hat und so die vorhandenen Kondensatoren nutzen kann. Im TO-92-Gehäuse kann man die drei Beine entsprechend biegen, so dass er auf die Pads gelötet werden kann. Ich habe diese Lösung bevorzugt, zumal ich nur die inkrementale Version aufgebaut habe, die das beschriebene Problem nicht hat. Als Kurzhubtaster wurde TASTER 3301B von Reichelt mit schwarzem Stößel eingesetzt, der Tasterkopf wurde dann eingefärbt. Anders als auf der Abbildung wurde später als Drehgeber der STEC11B03 eingesetzt, die geschlitzte 6mm-Achse wurde gekürzt. Für die LEDs wurden Typen mit hoher Lichtleistung (ca. 2000mcd) verwendet, so dass die Widerstände von 470R auf 4,7k geändert werden können.

Bei der Inbetriebnahme am LocoNet musste ich etwas länger nach der Fehlerursache suchen. Es fehlte das Pseudo-GND. Weiteres siehe bei RSCLD. Als dies geschafft war, konnte dann die Inbetriebnahme wie in der Bauanleitung beschrieben, erfolgen.

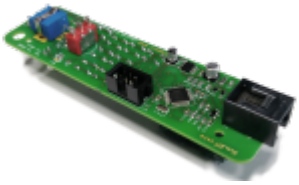
Die FREDI Schaltung und das Layout von Olaf Funke und anderen steht unter einer Creative Commons Namensnennung-Nicht-kommerziell-Weitergabe unter gleichen Bedingungen 3.0 Unported Lizenz.
The FREDI Schematic and Board by Olaf Funke and others is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.

Weitere Quellen beim FremoDCC:
Schaltplan und Stückliste entsprechend Version 1.8C

Bauanleitung
Inbetriebnahme
Mini-Bedienungsanleitung
FCalib2 - Programmierung und Firmware

leicht geänderte Layoutvariante für Schaltplan 1.8C
Bestückungsplan meines Layouts
Bestellmöglichkeit der Leiterplatte für Eigenbau bei OSH Park
Bestellmöglichkeit der Leiterplatte für Eigenbau bei Aisler
Eine kommerzielle Nutzung ist untersagt.

Neue Version mit SMD-Controller





Da auf der aktuellen Leiterplatte bereits SMD-Bauteile verwendet werden und ich keinen steckbaren IC benötige, habe ich für den Aufbau weiterer FREDIs eine Leiterplatte mit dem SMD-Typ des ATMega328 erstellt. Statt eines Quarzes wird ein Resonator gleicher Frequenz eingesetzt, die Größe der SMD-Bauteile wurde von 0805 auf 0603 geändert. Dies ist mit den mir vorhandenen Mitteln noch gut manuell lötbar. Wegen der geschilderten Schwingneigung des LM2936 bei zu niedrigem ESR habe ich einen anderen SMD-LDO eingesetzt.
Die hier vorgestellte Version der Leiterplatte hat als 3,3V-Regler den LP2951-33D im Design, der für niedriges ESR geeignet ist. Falls es Beschaffungsprobleme gibt, kann auch der dreipinnige LP 2950 ACZ3,3 auf die Pads gelötet werden, wie bereits bei der oben dargestellten 1.8C-Variante. Ein nicht mit Lötstopplack abgedeckter Teil eines Leiterzuges dient als Pad für dessen mittleres GND-Pin.

Ich habe die inkrementale und die analoge Version aufgebaut. Für den Geschwindigkeitsregler wurde ein BOURNS-Potentiometer 51AAA-B28-A18L verwendet. Unter die Montage-Mutter muss eine Isolierscheibe untergelegt werden.

Für den Betrieb muss entweder das Railsync/LocoNet ein gemeinsames Massepotenzial (Pin2-5) haben, ansonsten muss ein Pseudo-GND mit den zwei Dioden wie im RSCLD erzeugt werden. Diese sind im LN-RJ12-Verteiler als Option bestückbar.

Eine Programmierung der Firmware kann für meine Zwecke auch ohne das Tool FCalib2 nur mit AVRDUDE erfolgen:
Programmierung des FREDI mit AVRDUDE



Schaltplan
Bestückungsplan
Stückliste
Bestellmöglichkeit der Leiterplatte für Eigenbau bei Aisler
Eine kommerzielle Nutzung ist untersagt.

Sonstiges

RSCLD

Nachdem ich die Command Station und den SpaXbooster getestet hatte, war jetzt das LocoNet an der Reihe. Als erstes nahm ich den LocoBuffer in Betrieb. Hinweise hierzu unter dem entsprechenden Abschnitt.
Beim FREDi gab es Probleme. Mit normaler Spannungseinspeisung fing das Gerät erstmal an zu arbeiten. Am LocoNet jedoch nicht.
Grund: Der FREDi nutzt als Spannungsversorgung das RailSync, das am LocoNet-Stecker an den Kontakten 1 und 6 anliegt. An den Kontakten 2 und 5 liegt GND. Der FREDi hat zwar zwei Dioden als Gleichrichter. Aber wenn man die Schaltung weiter verfolgt, gibt es keine Beziehung zum GND.



Hier ist jetzt eine einfache Schaltung notwendig, die beim FremoDCC im RailSync-Current-Limiter-Device (RSCLD) RSCLD Urversion verborgen ist. Auch hier finden sich Infos MEC-Leonberg. Es sind zwei Dioden vorhanden, die zusammen mit den beiden Dioden im FREDi eine Greatz-Schaltung ergeben. Im FREDi entsteht also das Plus-Potential aus der Gleichrichtung und im RSCLD das Minus-Potential, welches mit dem

LocoNet-GND (RJ12 6-6, Pin2 und Pin5) verbunden wird, sodass eine Versorgung von LocoNet-Geräten aus dem RailSync möglich wird. Einen Hinweis zu diesem Zweck habe ich jedoch auf keiner Seite gefunden. Etwas ähnliches jedoch hier. Desweiteren sind im RSCLD Widerstände zur Strombegrenzung eingesetzt. Ich habe diese erstmal weggelassen und speise das DCC-Signal der Command Station entweder über einen Spaxbooster mit geringerer Leistung oder bei Testaufbauten direct als RailSync in eine LocoNet-Buchse (RJ12 6-6, Pin1 und Pin6) eines LN-RJ12-Verteilers ein.

Im oben beschriebenen LN-RJ12-Verteiler ist die Schaltung mit den beiden Dioden für das Pseudo-GND vorhanden.

LocoNet IR-Empänger





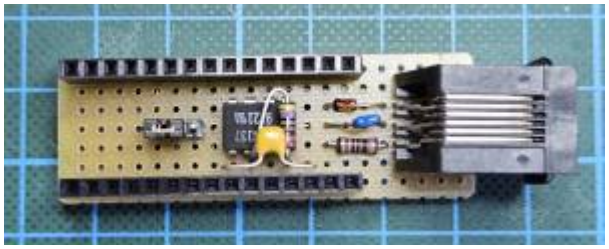
Für die Verwendung von zwei IR-Fernbedienungen von Piko (baugleich IRIS Uhlenbrock) habe ich den LocoNet-IR-Empfänger von Uhlenbrock in das LocoNet eingebunden. An den IR-Empfänger können mit den vier IR-Kanälen A bis D dann vier Fernbedienungen angeschlossen werden. Oder eine Fernbedienung kann mit den vier umschaltbaren Kanälen jeweils eine Loks bedienen. Die Zuordnung der Loks erfolgt an der Fernbedienung mit Eingabe der Lokadresse für den jeweiligen Kanal. Die angeforderte Lokadresse erhält dann vom rocrail-Slotserver einen Slot zugewiesen. Die Weichensteuerung hat bisher über diesen Weg nicht funktioniert, war aber auch nicht Ziel der Anwendung.

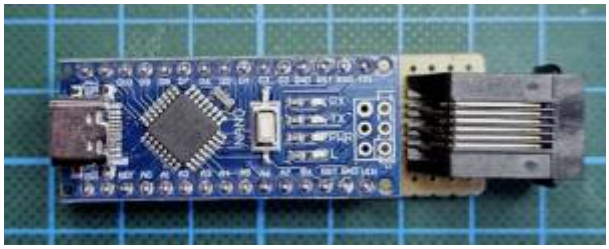
Tools

DCC-Sniffer

Für die Analyse der DCC-Pakete wurde mit einem Arduino-Nano und der Software DCC-Inspector von DCC-Ex ein Paket-Sniffer auf einer Lochrasterplatine aufgebaut. Die Schaltung eintspricht der dort gezeigten Schaltung. Durch den Optokoppler ist die galvanische Trennung zum PC realisiert.
Der Anschluss erfolgt über die Loconet-Verkabelung mit RJ12-Steckverbinder und greift dort das RailSync-Signal ab.

Der Sketch verwendet Pin D8 als Eingang. Um eventuell andere Software zu nutzen, die Pin D2 als Eingang verwendet, wurde ein Jumper für diese Umschaltung hinzugefügt. Auf eine Anzeige wurde verzichtet, die Ausgabe erfolgt über die serielle Schnittstelle mit 115200 bit/s und kann mit einem Terminalprogramm empfangen werden.

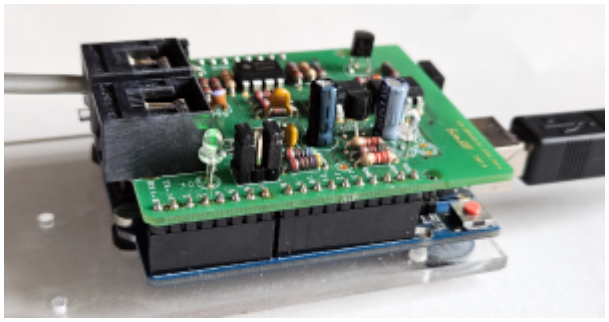




LocoNet-Monitor

Für den LocoNet-Monitor wurde das oben gezeigte galvanisch zum PC getrennte LN-Shield-ISO zusammen mit einem Arduino-Uno verwendet. Als Software wird der Beispiel-Sketch LoconNetMonitor von MRRWA verwendet. Der Anschluss an das LocoNet erfolgt über die RJ12-Verkabelung. Die LocoNet-Stromquelle auf der Platine (Pull-Up) muss deaktiviert werden.

Die Ausgabe erfolgt über die serielle Schnittstelle mit 57600 bit/s und kann mit einem Terminalprogramm empfangen werden.



From:
https://www.simandit.de/simwiki/ - **Wiki**

Permanent link:
https://www.simandit.de/simwiki/doku.php?id=modellbahn:digital

Last update: **2026/11/22 21:15**