Digital-Hardware

Some parts are descibed in english here: Digital Hardware english

Hardware-Konfiguration

Ich fange gerade an, mich etwas mit der digitalen Steuerung der Modellbahn zu beschäftigen. Ich habe einige alte Loks mit einem Digitaldekoder ausgestattet, um erstmal einen Anfang zu machen. Das Bild zeigt meinen derzeitigen Aufbau.

Dargestellte Komponenten sind

- USB Command Station.
- RJ12-Verteiler - 8-Ports für LocoNet® mit Pseudo-Ground
- SpaxBooster - FremoDCC-Projekt eines Boosters
- PC-Schnittstelle LocoBuffer für LocoNet®
- FREDI - FREMODCC-Fahrregler für LocoNet®
- LocoIO-Keypad für Weichensteuerung über LocoNet®

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.

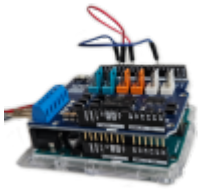
Command Station

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 wird 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

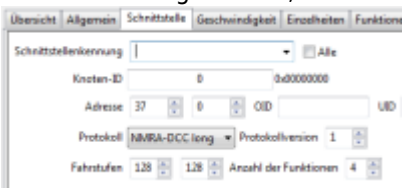


Auf Basis des Arduino Uno wurde eine Command Station aufgebaut. Dazu wird zusätzlich ein MotorShield benötigt. Die Verdratung und die Einbindung ist bei rocrail beschrieben.
Für die Software habe ich die Beispielprogramme BaseStation-Classic von DCC-EX und die Ursprungsversion von DCCpp getestet. Die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

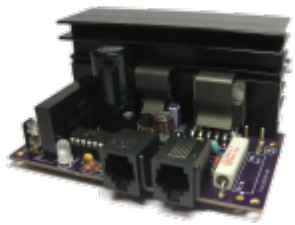
Hinweis: Beide Versionen behandeln nach meiner Erfahrung das **Emergency-Stop** des Fahrreglers nicht richtig. Ich habe von der DCC-EX-Version ein „Fork“ erstellt und in eine Datei an einer Stelle eine Änderung eingefügt. Der „Fork“ für BaseStation-Classic liegt auch auf GitHub. Für CommandStation-EX muss eine ähnliche Anpassung in der DCCEXParser.cpp vorgenommen werden, zwei Zeilen wurden auskommentiert:

```
...
        if (tspeed < 0)
            tspeed = 1; // emergency stop DCC speed
//        else if (tspeed > 0)
//            tspeed++; // map 1-126 -> 2-127
        if (cab == 0 && tspeed > 1)
...
```

Weiterhin 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.



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
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.

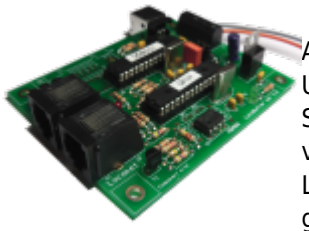
Das Projekt SAnD-Ei ist zwar eingestellt, aber die Daten liegen auf der Webseite zum Download. Ich habe die Daten für die Bestellung bei Aisler aufbereitet und auf Nachfrage kann ich auch einen Link zu einer Leiterplattenbestellung senden, falls beim Entwickler keine Platinen mehr verfügbar sind.

LocoNet-Hardware

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 LocuBuffer-USB zu kaufen.

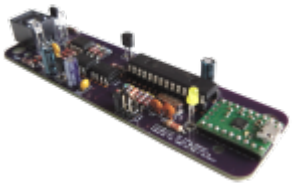
HDM09USB



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

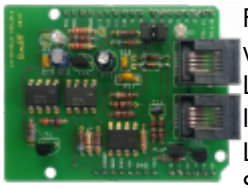
Aus der Zusammenführung des Schaltungsdesigns des LNShield-ISO bzw. Loconet-USB, dem Arduino-Pro-Mini und einem USB-Seriell-Wandler wurde der LocoBuffer-ProMini entwickelt. Die Baugruppe realisiert über die Optokoppler eine galvanische Trennung zwischen PC und Loconet wie bereits beim LocoBuffer-USB. Wie beim LNShield ist Tx auf D6 oder D7 über Jumper oder Drahtbrücke konfigurierbar. Rx liegt auf D8. Die **15mA-LocoNet-Stromquelle** (Pull Up) ist über Jumper **JP4** aktivierbar.
Als Software wird der Beispiel-Sketch LocoLinx von MRRWA verwendet. Mit der Realisierung der im Sketch vorgesehenen Verbindung vom Port D9 des Arduino und dem CTS#-Signal des FT230XS kann auch das **Hardware-Handshake** verwendet werden. Dies ist über Jumper **JP3** zuschaltbar. Die Baudrate der seriellen Schnittstelle ist **57600**.
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. Die SMD-Bestückung wurde manuell mit Lupenbrille ausgeführt.
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, wurde anstelle des 78L05 ein LDO LP 2950 ACZ5,0 eingesetzt. Dieser hat außerden einen deutlich geringeren Ruhestrom.

Der ATmega328 muss zuerst mit einen Bootloader programmiert werden, um danach als Arduino Pro Mini benutzt zu werden. Dazu ist ein 6-poliger Steckverbinder für einen AVR-Programmer z.B. von Pololu vorgesehen. Siehe Bootloader programmieren.
Anschließend kann die Programmierung über den USB-Port mit der Arduino-IDE wie für einen Arduino Pro Mini erfolgen.

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

LNShield-ISO für Arduino



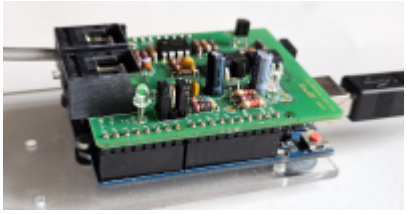
Für Experimente mit dem Arduino und den von MRRWA veröffentlichten Bibliotheken und Beispiel-Sketchen habe ich den LocoNet-Teil der Schaltung des oben beschriebenen LocoBuffer-II+USB auf eine Arduino-Shield-Platine gebracht. Die Versorgung der LocoNet-Seite erfolgt aus dem RailSync. Die 15mA-LocoNet-Stromquelle (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. 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.

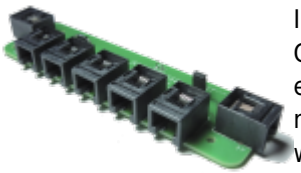
Die serielle Schnittstelle wird auf **57600** Baud eingestellt und das **Handshake** muss **aus**geschaltet werden, da der Arduino dies nicht unterstützt. Diese Einstellungen erfolgen im Betriebssystem und auch in der Steuer-Software.

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.

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. Das Pseudo-GND kann über eine Steckbrücke zugeschaltet werden. Den dort enthaltenen Widerstand habe ich derzeit nicht bestückt. Ein bestückter Widerstand 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. 1000mcd) verwendet, so dass die Widerstände von 470R auf 2,2k geändert wurden.

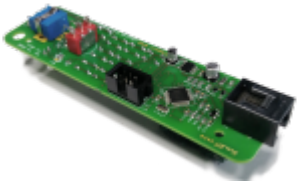
Bei der Inbetriebnahme am LocoNet musste ich etwas länger nach der Fehlerursache suchen. Es fehlte das Pseudo-GND. Weiteres siehe bei RSCLD. Als dies geschafft war, konnte dann die Inbetriebnahme wie in der Bauanleitung beschrieben, erfolgen.

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. Es 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 mit BOURNS-Potentiometer (51AAA-B28-D18L) aufgebaut.

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.

LocoIO-Keypad

Für die Weichensteuerung will ich die Idee des **LocoIO** verwenden. Dieser entstammt auch einer Idee von John Jabour und wird von Rocrail unterstützt. Diese Seite ist nur noch über das Web-Archiv verfügbar.
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. Als Pullup für die Taster wurden Widerstandsnetzwerke eingesetzt. Die Bestückung wurde so geändet, dass Komparator und Schalttransistor, die der LocoNet-Ansteuerung dienen, in die Nähe des Steckverbinders gebracht wurden. Die Reihenfolge der Taster S09…S16 wurde gedreht, so dass es zur Programmiermaske passt. Gegenüber von S01 ist jetzt S09, S09 bedient jetzt Port 9 usw.. 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 auch der LP 2950 ACZ5,0 getestet werden. Dieser ist ein LDO mit deutlich geringerem 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. Der Sketch kann hier geladen werden https://github.com/ClubNCaldes/SVLocoIO.

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 den Schaltungsentwurf der Flugsimulations-Geräte 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. Man kann aber auch Kabel an beschriftete Lötstellen anlöten für die Programmierung mit einem Arduino. Siehe Bootloader programmieren.
- Die ATmega-Eingänge sind im Sketch auf PULLUP definiert worden, so dass die beiden Widerstandsnetzwerke RN1 und RN2 (in PCB-Rev. 1.3) nicht notwendig sind. Die internen Pullup haben lt. Datenblatt einen Wert 20 … 60 kOhm.
- Für die Programmierung mit einem USB-Seriel-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 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:** In diesem Fall ist keine galvanische Trennung zwischen LocoNet und PC vorhanden. Bei einem nur über Akku betriebenen Laptop ist diese Verbindung aber 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, kann ein LDO LP 2950 ACZ5,0 verwendet werden.

Ich habe festgestellt, dass im Sketch nicht alle Funktionen der PIC-Vers-148 umgesetzt wurden, wie sie im GCA123/GCA50 verwendet werden. Als Input ist nur die Variante „Block Sensor“ umgesetzt. Eine Option, die auch in der Sensor-Funktion fehlte, habe ich nachträglich in den Sketch eingefügt. Das ist die Option „Active High“, die ich für die Weichensteuerung mit Rocrail nutze. Für meine Anwendung sind beide LocoIO-Keypad-Varianten jetzt gleichwertig.

Schaltplan
Bestückungsplan
Stückliste
Bestellmöglichkeit der Leiterplatte für Eigenbau bei Aisler
Eine kommerzielle Nutzung ist untersagt.
ergänzter Arduino-Sketch forked from ClubNCaldes/SVLocoIO

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

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 Abschnitt dazu.
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. Dort sind zwei weitere Dioden eingesetzt, 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 (RJ11 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. Zum Testen habe ich diese erstmal weggelassen und speise das DCC-Signal der SPROG3 Command Station als RailSync in den LocoNet-Steckverbinder (RJ11 6-6, Pin1 und Pin6) direkt ein.

Im oben beschriebenen RJ12-Verteiler ist diese Schaltung zusätzlich 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 wird nicht verwendet.

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

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

Last update: **2025/09/08 11:24**