Digital Hardware - english

Here are explained some parts of my development in english. The german page is here: Digital-Hardware

To enable speed controllers and other components to communicate with a PC as a control centre via LocoNet, an interface is required to buffer the signals from and to the PC and to implement communication with the LocoNet. The first draft of the **LocoBuffer** was created by John Jabour. This page is only available via the Web Archive.

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

Arduino DCC Command Station
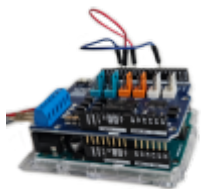
In parallel to the SPROG3, a command station was built on the basis of the Arduino Uno. This also requires a MotorShield. The wiring and the integration will be found at rocrail.
For the software I have used the sample programme BaseStation-Classic (original version). The associated library DCCpp can be integrated via the Arduino IDE.

In my experience, both versions do not handle the **emergency stop** of the speed controller correctly. I have created a 'fork' of the DCC-EX version and inserted a change in a file. The fork of BaseStation-Classic is also on GitHub. For CommandStation-EX, a similar adjustment must be made in DCCEXParser.cpp, two lines have been commented out:

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

It should also be noted that only locomotive decoders with **128 speed steps** are supported. Corresponding entries must be made in rocrail so that the speed controller is dispatched accordingly. The forums thread was a good help to me when it didn't work at first. The protocol setting is important here.
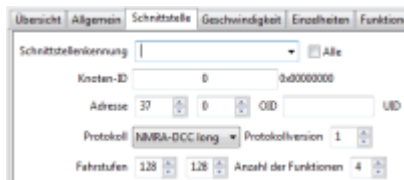

DIY parts for LocoNet

LNShield-ISO for Arduino

used as LocoBuffer
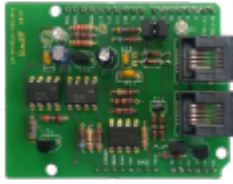
I put the circuit design of the galvanic isolated LocoNet part of the LocoBuffer-II on an Arduino shield board. The LocoNet side is supplied from the RailSync. The 15mA LocoNet power source (pull-up) can be activated via a jumper. The signal routing to the Arduino is the same as for the Fremo LNShield, so that the D6 or D7 signal can be used for Tx. Another LED can optionally be used for further signalling; this is currently not fitted. I have built the board without the stacking headers, as the first tests are intended as LocoBuffer. I have loaded the example sketch LocoLinx from MRRWA for this purpose.
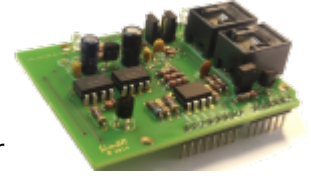
The serial interface is set to **57600** baud and the **handshake** must be switched **off** as the Arduino does not support this. These settings are made in the operating system and also in the control software.

Schematic
Assembly
Bill of Material
Order of the PCB at Aisler for personal use
Commercial use is prohibited.

To ensure operability even at low input voltages, an LDO LP 2950 ACZ5,0 is used. This also has a significantly lower quiescent current. A Schottky diode can be used as a rectifier diode instead of the 1N4148, e.g. BAT46. For the LEDs, types should be used that still have sufficient intensity with the 2.2kOhm series resistor at 5V.

LocoBuffer-ProMini

The LocoBuffer-ProMini was developed by combining the circuit design of the LNShield-ISO, the Arduino-Pro-Mini and a USB-to-serial converter. The module realises a galvanic isolation between PC and LocoNet via the optocouplers. As with the LNShield, Tx can be configured on D6 or D7 via a jumper or wire bridge. Rx is on D8. The **15mA LocoNet current source** (pull-up) can be activated via jumper **JP4**.
The example sketch LocoLinx from MRRWA is used as the software. The hardware **handshake** can also be used by realising the connection provided in the sketch from port D9 of the Arduino to the CTS# signal of the FT230XS. This can be activated via jumper **JP3**. The baud rate of the serial interface is **57600**. The LocoNet circuit was adopted from the LocoBuffer-USB and is designed using THT technology. The ATmege328 side with the USB-to-serial converter is designed using SMD technology. The SMD assembly was carried out manually with magnifying glasses.
Four LEDs indicate the power supply from RailSync and USB as well as communication on LocoNet and USB-Rx-Tx. This function is default but can be programmed in the FTDI chip with FT_Prog.

In order to ensure operability even with a low input voltage, an LDO LP 2950 ACZ5,0 was used instead of the 78L05. This also has a significantly lower quiescent current.

The ATmega328 first has to be programmed with a bootloader in order to be used as an Arduino Pro Mini. A 6-pin connector for an AVR programmer, e.g. from Pololu, is provided for this purpose. I did this programming after the SMD assembly. See programming the bootloader.
Afterwards, programming can be carried out via the USB port with the Arduino IDE as for an Arduino Pro Mini.
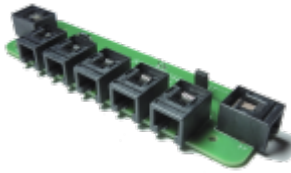
Schematic
Assembly
BOM
Order of the PCB at Aisler for personal use

Commercial use is prohibited.

LocoNet RJ12 Box

I have built an RJ12 box, which optionally also contains the two diodes for the pseudo-GND (see RailSync-Current-Limiter-Device (RSCLD) RSCLD Urversion) and a resistor like here. The pseudo-GND can be connected via a jumper. I have not currently fitted the resistor contained there. An assembled resistor can be bridged with a jumper if required. The pseudo-GND may only be present once in a RailSync supply section.
The circuit fits into the FREDI housing.

Schematic
Assembly
Order of the PCB at Aisler for personal use
Commercial use is prohibited.

Low profile RJ12 connectors of the following manufacturer are tested:

- Econ Connect MEB6/6PL at Voelkner
- MOLEX 95501-2661 6P6C at diverse distributors

LocoIO-Keypad-Arduino

I want to use the idea of **LocoIO** for the turnout control. This also comes from an idea by John Jabour and is supported by Rocrail. Unfortunately this page is only available via the web archive.

There are several replicas, including the GCA50 presented at Rocrail. The GCA123, which has 16 buttons, is a handy device for beginners.

Daniel Guisado ClubNCaldes has created an Arduino sketch for LocoIO based on the Arduino-Pro-Mini hardware. The sketch can be downloaded here https://github.com/ClubNCaldes/SVLocoIO.

I have realised that not all functions of the PIC-Vers-148 have been implemented in the Arduino sketch, as they are used in GCA123/GCA50. Only the 'Block Sensor' variant is implemented as an input. I have subsequently added an option to the sketch that was also missing in the sensor function. This is the 'Active High' option, which I use for switch control with Rocrail. For my application, both LocoIO keypad variants are now equivalent.

I have taken over the circuit and assembly of the connection side to the Loconet from the LocoIO Keypad and replaced the PIC with the ATmega328. I used the SMD type in the TQFP housing, as the DIL type should not be used for new designs according to the website. And so I was also able to use a already used circuit design.

Everything fits into the FREMO-FREDI housing of the speed controller as before. Please note the following during assembly:

- The ATmega must first be programmed with a bootloader. A 6-pin connector for an AVR programmer, e.g. from Pololu, is provided for this purpose. However, you can also solder cables to labelled solder joints for programming with an Arduino. I carried out this programming after the SMD assembly. See Programming bootloader.
- The ATmega inputs are defined as PULLUP in the sketch, so that the two resistor networks RN1 and RN2 (in

PCB-Rev. 1.3) are not necessary. According to the data sheet, the internal pull-ups have a value of 20 … 60 kOhm.

- A six-pin FTDI connector is available for programming with the Arduino interface, as on the Arduino Pro Mini with an USB-Serial-Adapter.
- The LEDs should be soldered and fitted into the housing before the FTDI connector. The overhanging terminal wires of the LEDs should then be shortened as much as possible so that they do not interfere with the connector.
- A jumper **FTDI 5P** is provided for programming purposes. This must be removed if the keypad is connected to the Loconet and also to the PC via the FTDI connector at the same time.
- **Caution:** In this case, there is no electrical isolation between LocoNet and PC. However, this connection is unproblematic with a laptop without connection to ground potential. The possibility of leaving the serial connection with USB-Seriell-Adapter on the PC during operation means that the communication on the LocoNet can be monitored with a terminal programme, as the software outputs this in debug mode on the serial interface.
- To ensure operability even at low input voltages, an LDO LP 2950 ACZ5,0 can be used.

The option of leaving the serial connection (with USB-to-Serial-converter) on the PC during operation means, that communication on the Loconet can be monitored with terminal program, as the software outputs this to the serial interface in debug mode.

Schematic
Assembly
Bill of Material
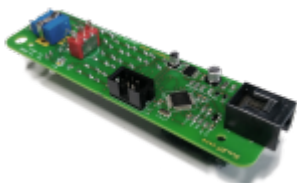Order of the PCB at Aisler for personal use
Commercial use is prohibited.
extended Arduino Sketch forked from ClubNCaldes/SVLocoIO

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

FremoDCC-Throttle FREDI

The FREDI Schematic and Board by Olaf Funke and others is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.

I have adapted the circuit based on the FREDI-1.8C of the FremoDCC with an SMD type of the ATmega328 and a different voltage regulator, as there were problems with the used LM2936. The STEC11B03 was used as the rotary encoder, the slotted 6mm axis was shortened. Types with high light output (approx. 1000mcd) were used for the LEDs, so that the resistors were changed from 470R to 2.2k.

I have assembled the incremental with STEC11B03 and the analogue version with a BOURNS potentiometer (51AAA-B28-D18L).
The Preudo-GND is required for operation at the Loconet/Railsync, as described at the RJ12 Box.

Additional information about FREDI at FremoDCC:
https://fremodcc.sourceforge.net/throttle/throttle.en.html

Sources most in german:
Commissioning
Mini user manual
FCalib2 - Programming and firmware

For my purposes, the firmware can also be programmed without the FCalib2 tool, only with AVRDUDE:

Programming of FREDI with AVRDUDE

Schematic
Assembly
BOM
Order of the PCB at Aisler for personal use
Commercial use is prohibited.

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

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

Last update: **2026/01/21 22:35**