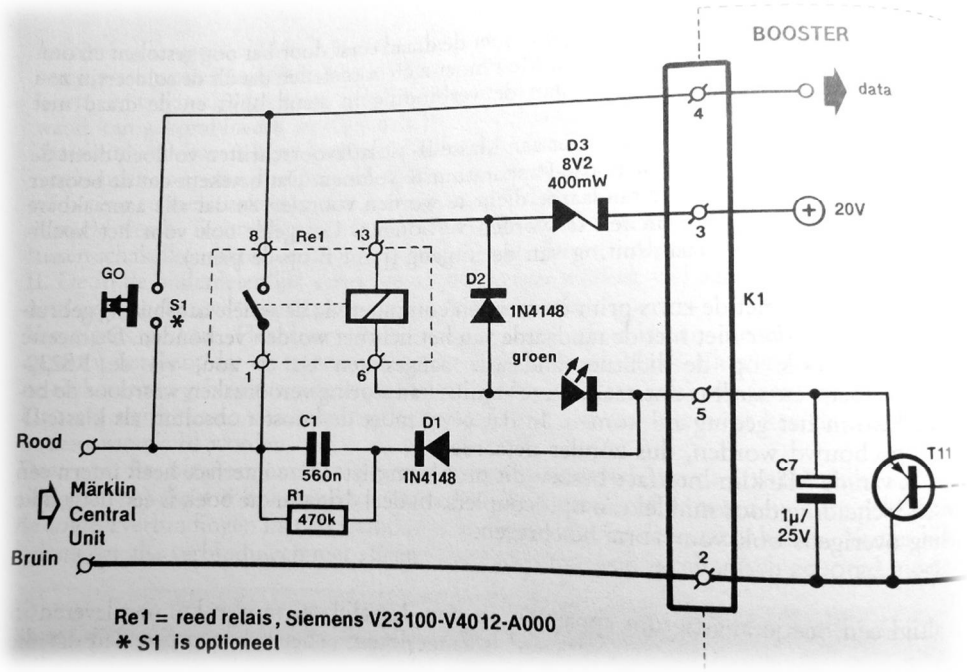
Edits Booster Interface

In the Elektor books and on internet, a couple of interfaces can be found to use the Edits booster. However, most of them require a start (and stop) button.

The one in the original Edits book can be used for automatic start. However, this schematic does not work with DCC. This is caused by a symmetrical DCC signal, while the Marklin/Motorola signal is negative biased. The C1/R1/D1 circuit does not work with DCC, the booster will only work with DCC via the Go button.



Another issue is that a data loss in the booster will also remove the ground connection of pin 5 (via T11). That means that removing data from the master booster, will also drop the relay. That is not a bad thing within this schematic, however we therefore cannot use this relay for a (real) short indication.

If we would use this relay for a short indication, the master would not send data anymore while there is a short. However, as long as there is no data, a short (T11 not switching to ground) will remain…

Other things that were noticed:

- After stopping the DCC signal, the Logic analyzer showed a rather long period with a positive voltage. The short circuit in the Edits Booster works however on the negative part of the circuit… So a short is noticed, before the voltage drops to 0V;
- T11 stops about 4.5 milliseconds after the last pulse change;
- Marklin/Motorola signal has a maximum of 2.1 milliseconds space between packages.

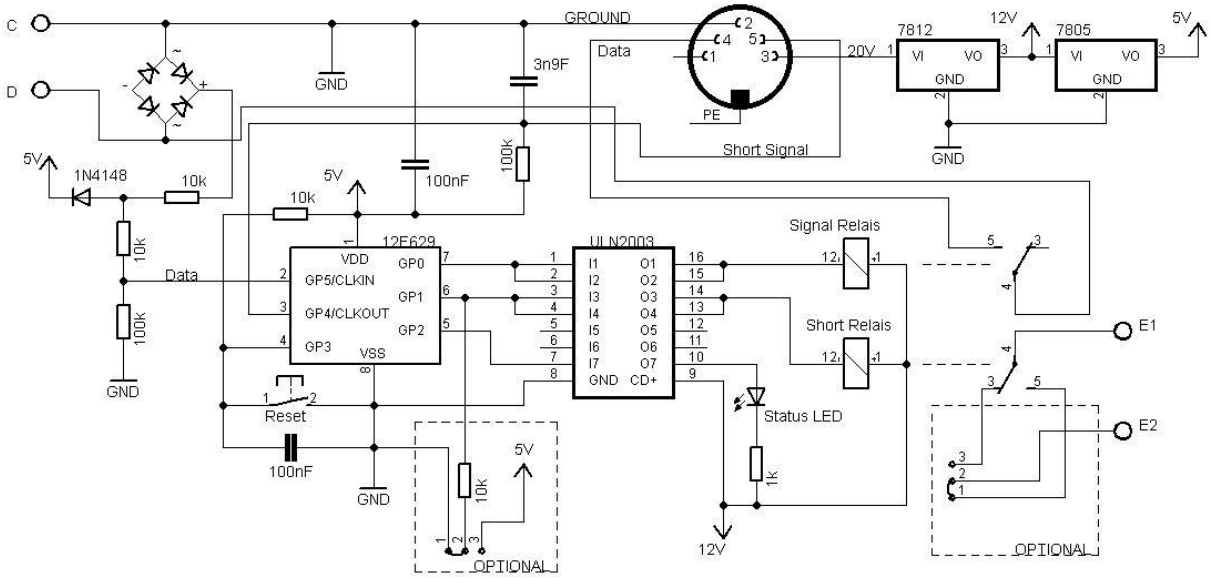
So time to think of something else…

Requirements

1. Automatic on/off, based on data
    a. Data is based on counting pulses low to high;
    b. Data line is removed from the Edits booster when no pulses are counted for 2.5 milliseconds (multiple protocols can be used);
    c. Data line is controlled by a reed relay with low current use;
    d. Status Led shows if the data is send to the booster.
2. Short indication is based on the ground/open signal from the booster.
    a. A short is only indicated when data is send to the booster, otherwise it is ignored;
    b. The Status LED will flash when a short has occurred;
    c. A (real) short must be reset manually;
    d. Short Indication is done via a relay;
    e. The short indication can be used for creating a real CDE interface;
    f. Wait at least 500 microseconds for a steady short signal, before making the short active;
    g. It can be optional set if a power loss of the Edits Booster will create a short indication.
3. Circuit is powered from Edits Booster.

Schematic

The following schematic takes care of the interface. All logic is done via a Pic 12F629 microcontroller.



The Edits booster will provide +20V. Via a 7812 and 7805 we can supply the relays at 12V and the microcontroller at 5V. It is strongly recommended to use Buck converter, instead of a 7812 to generate 12V to reduce current use by the interface..

To count the data pulses at 5V, a rectifier will output only positive voltages. The diode and resistors will limit the voltage to 5V at GP5 of the 12F629.

The short signal from the booster is stabilized via a 100k resistor to 5V and 3n9 capacitor to ground. This will provide a steady 0V (booster is working) or 5V (booster is off or short has occurred).

The reset (after a short) can be done via a push to close button and will reset the microcontroller.
To control the relays and LED, I used a driver like the ULN2003 or ULN2803 because I had that one available.

During startup, GP1 is checked if 5V is present via a 10K resistor. If so, the short relay will be normally activated and will drop off during a real short. If the E2 jumper is then selected to the NC (Normally Closed) switch, a power loss of the Edits Booster will also create a shortcut warning.
When GP1 is not connected (or connected to ground via the resistor), the short relay is only activated during a real short. The E2 jumper would then normally be set to the NO (Normally Open) switch.

A CDE interface will provide a short signal by connecting E to D. So by connecting E2 to D, we have a real CDE interface!

Firmware/Software

The software is written in Pic Micro Pascal (http://www.pmpcomp.fr/smf/). It's a wonderful freeware program. Hex file and source code is available.

Building

Since I only needed one, I did not make a PCB, but created it on breadboard. Remember to check voltages first, before installing the microcontroller and ULN2003.

Use reed relays or other low current relays!

The relays take the most current and you want to try to avoid that. The books mention a maximum use of 25mA, I did it with maximum 32mA (both relays powered) and all worked fine.

For a CDE interface it is also possible to use an optocoupler PC817 with a diode and resistor to pull E to D (instead of a relay). Check the BoosteR-CDE schematic on [Paco's website](#).