

Railuino – Hacking your Märklin



(Director's Cut)

Jörg Pleumann
Maker Faire Hannover
03.08.2013

Fahrplan

- Motivation
- Grundlagen
 - Digitale Modellbahn
 - Arduino
- Railuino
 - Arduino Hardware
 - Arduino Software
 - App für Android
- Demo

Motivation

- Android-Entwickler seit 2007
 - Derzeit Android Team Lead bei MEDION
- Modellbahner (nach längerer Pause) seit 2011
 - Wie lässt sich beides verbinden?

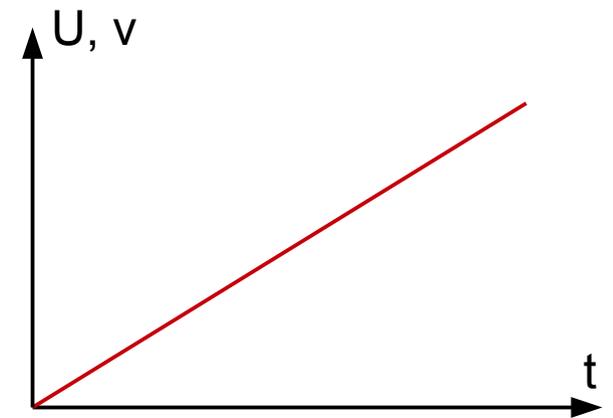
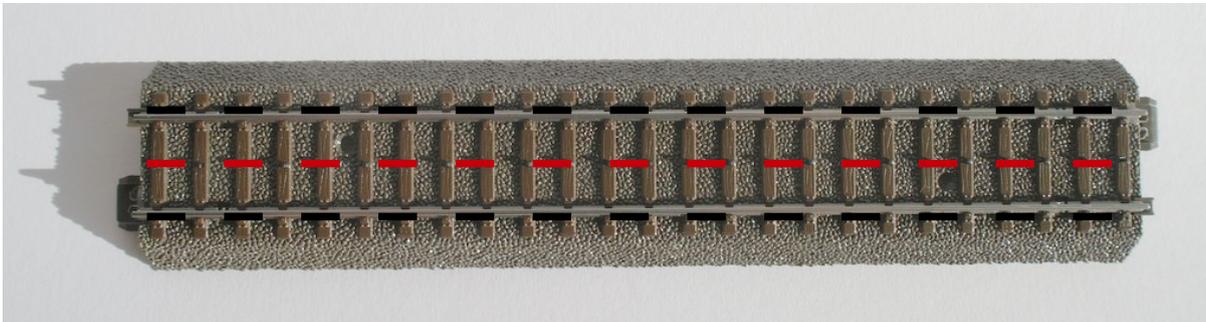


?

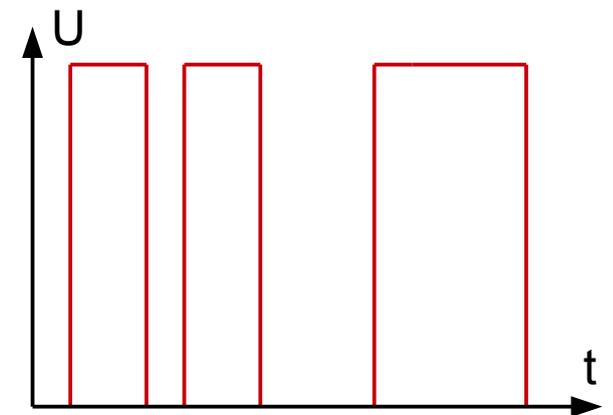
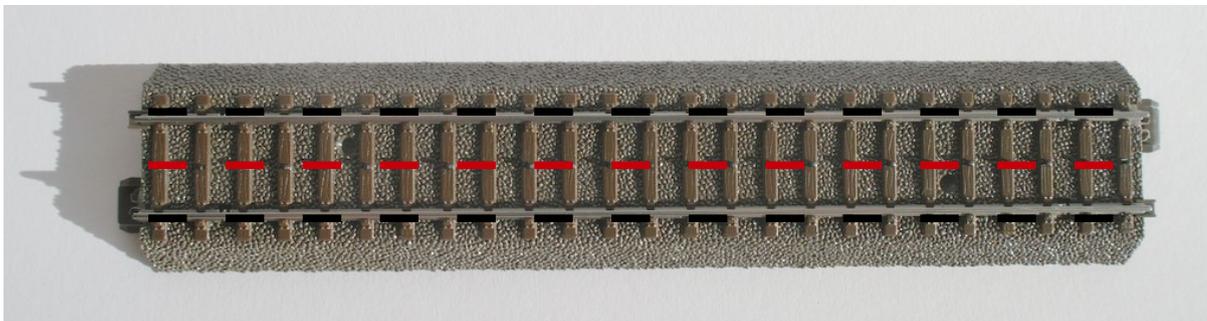


Digitale Modellbahn (I)

- Analog: AC oder DC, $v \sim U$

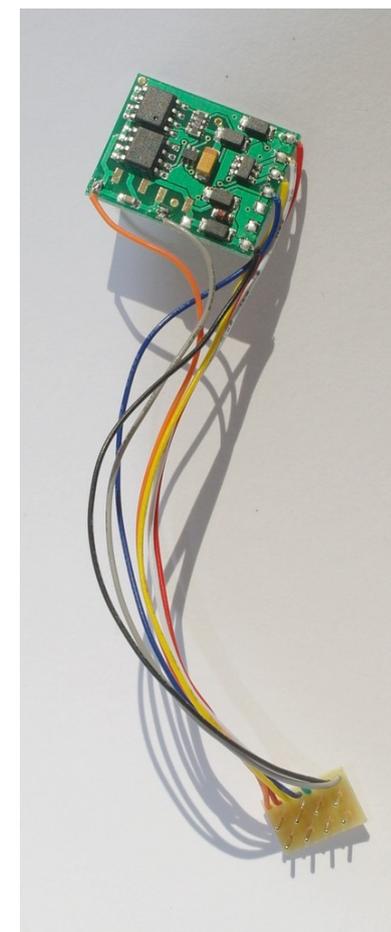


- Digital: DC plus Protokoll



Digitale Modellbahn (II)

- Viele Vorteile
 - Weniger Kabelsalat
 - Spitzenlicht hat konstante Helligkeit
- Loks und Zubehör brauchen Decoder
 - Alles hat eine eindeutige Adresse
- Verschiedene Protokolle
 - MM2 Märklin-proprietär alt
 - MFX Märklin-proprietär neu
 - DCC Offenes Protokoll der NMRA



Digitale Modellbahn (III)

- Infrarot-Fernsteuerung
 - Nur vier Lokomotiven
 - Nur MM2, fixe Adressen
- Mobile Station 2
 - MM2, MFX und DCC
 - Weichen schalten
- Central Station 2
 - Gleispläne darstellen
 - Ethernet-Anschluss, iPhone

20,- €



100,- €



500,- €

Digitale Modellbahn (IV)

- Mobile Station 2 reines Bediengerät
- Digitalsignal wird in der Anschlussbox erzeugt
- Die beiden kommunizieren via CAN-Bus
- Es gibt eine freie Buchse an der Box
- Das Protokoll ist (teilweise) offengelegt

Da muss doch etwas zu machen sein...

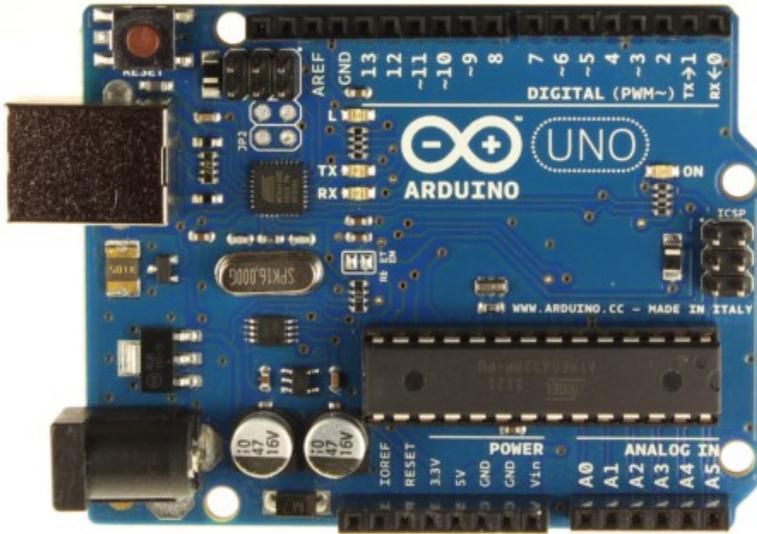


Arduino (I)

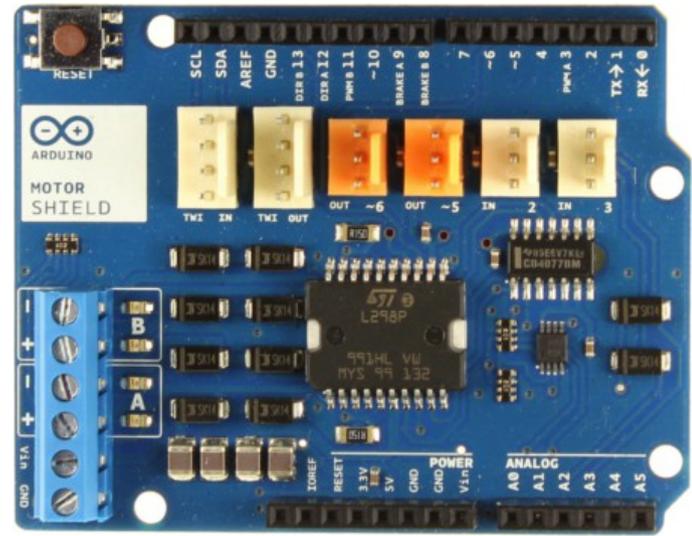
- Einfach zugängliche Microcontroller-Plattform
- Zielgruppe sind Bastler, Künstler, Schüler, ...
- Open-Source (Hardware & Software)
- Sehr populär, viel Material & Zubehör
- Sehr günstig, ab 15 Euro aufwärts
- Programmierung in Untermenge von C/C++
- Einfache Entwicklungsumgebung
- Flashen über USB-Kabel



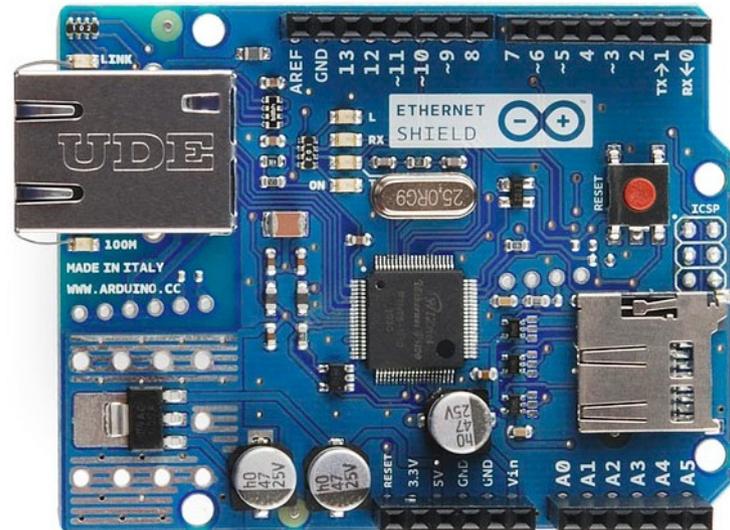
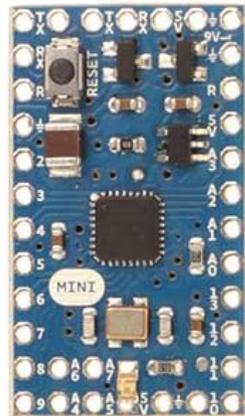
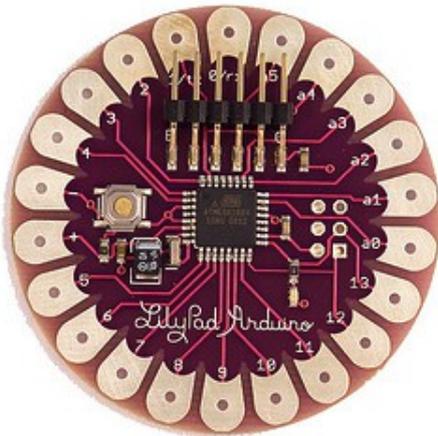
Arduino (II)



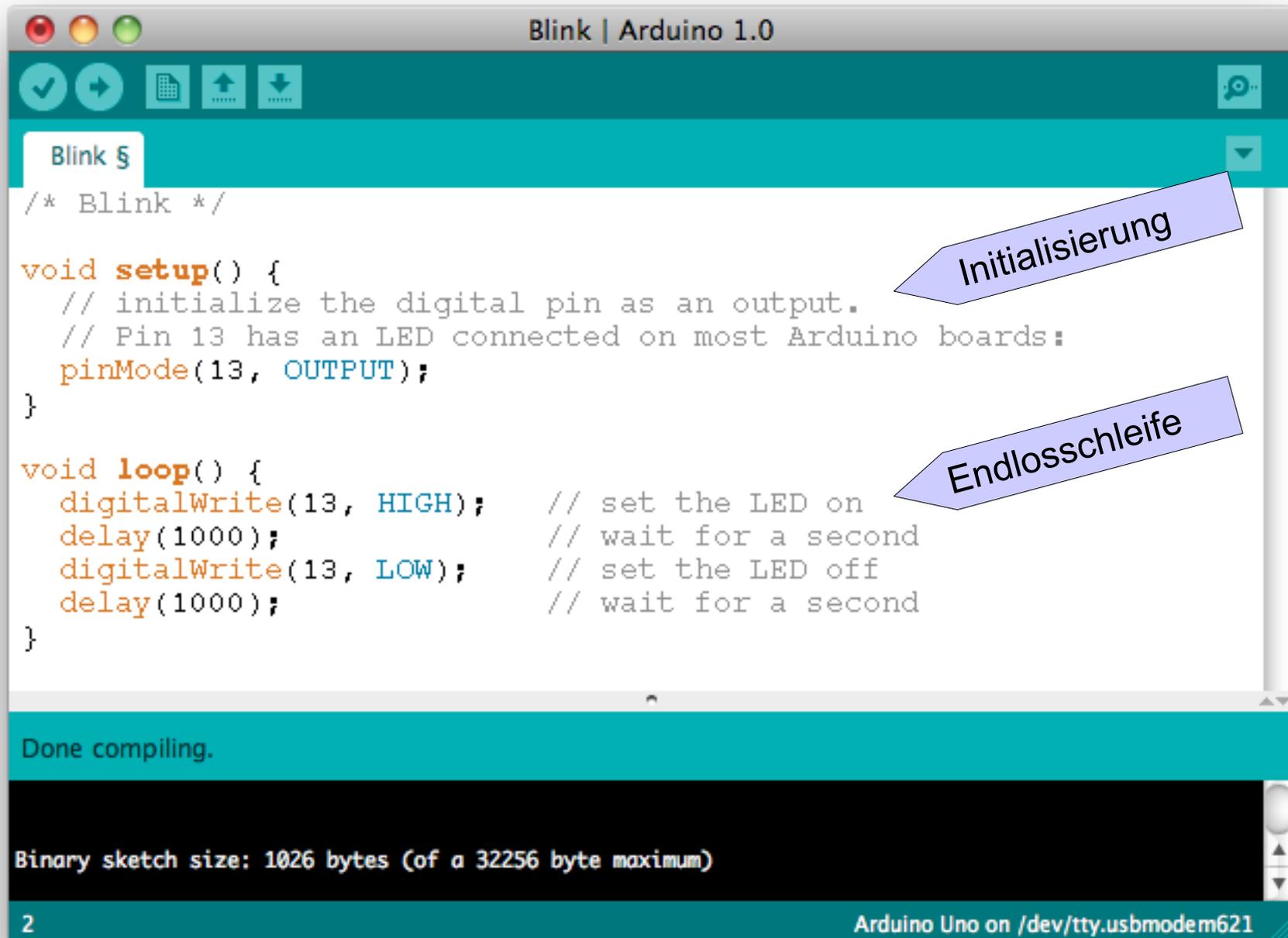
Boards



Shields



Arduino (III)



The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0". The main editor area contains the following code:

```
/* Blink */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // set the LED on  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // set the LED off  
  delay(1000);           // wait for a second  
}
```

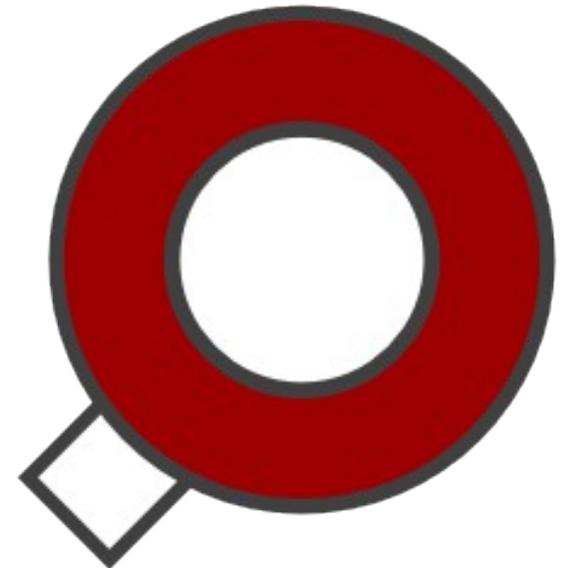
Two blue callout boxes with white text and arrowheads point to specific parts of the code:

- "Initialisierung" points to the `pinMode(13, OUTPUT);` line in the `setup()` function.
- "Endlosschleife" points to the `loop()` function.

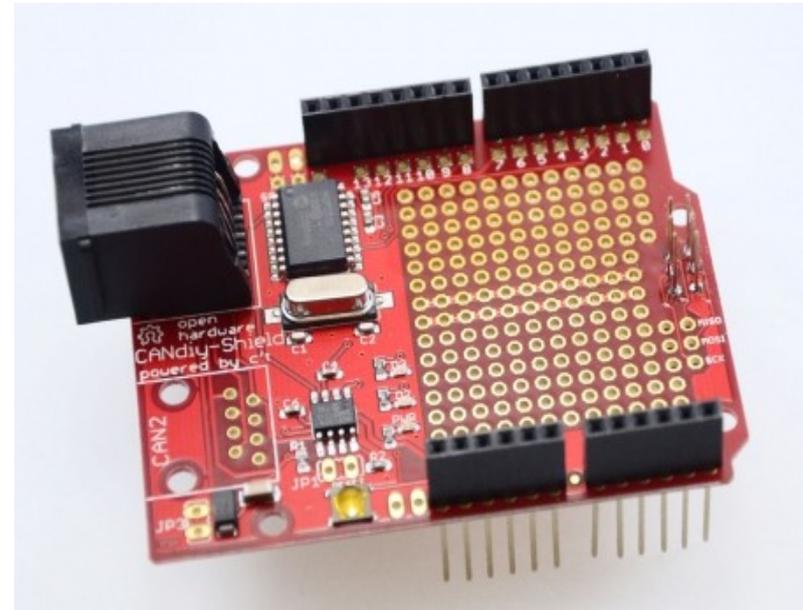
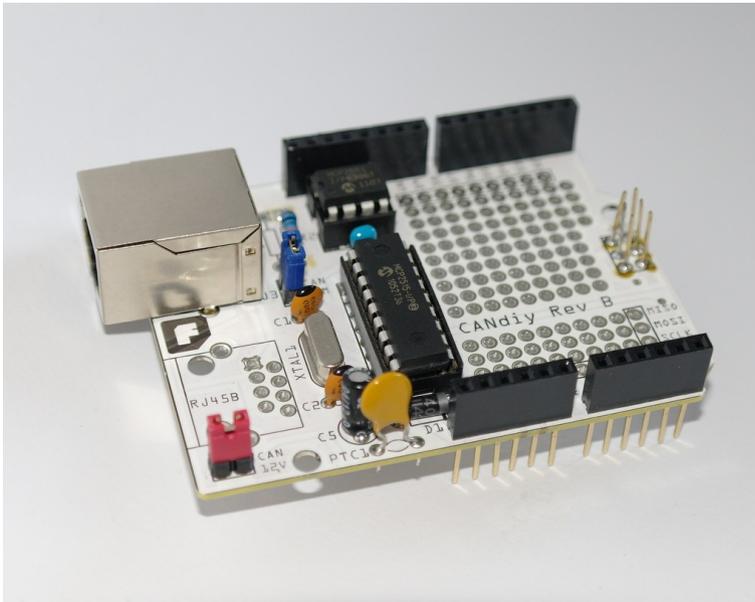
Below the code editor, a teal status bar displays "Done compiling." Below that, a black console area shows "Binary sketch size: 1026 bytes (of a 32256 byte maximum)". At the bottom of the IDE, a teal bar shows the page number "2" on the left and the connection information "Arduino Uno on /dev/tty.usbmodem621" on the right.

Railuino

- Projekt besteht aus
 - CAN Shield für Arduino
 - Bibliothek mit vielen Beispielen
 - App für Android
- Open-source
 - <http://code.google.com/p/railuino>
- Artikel in c't Hardware Hacks 1/2013



Hardware



- CANdiy – CAN do-it-yourself Shield
- Basiert auf MCP 2515 / MCP 2551
- Links: Prototyp entworfen mit Fritzing
- Rechts: Profi-Version von Watterott (15 €)

Kabel zur Anschlussbox

- RJ-45 auf 10-polig Mini-DIN
- Belegung der beiden Seiten

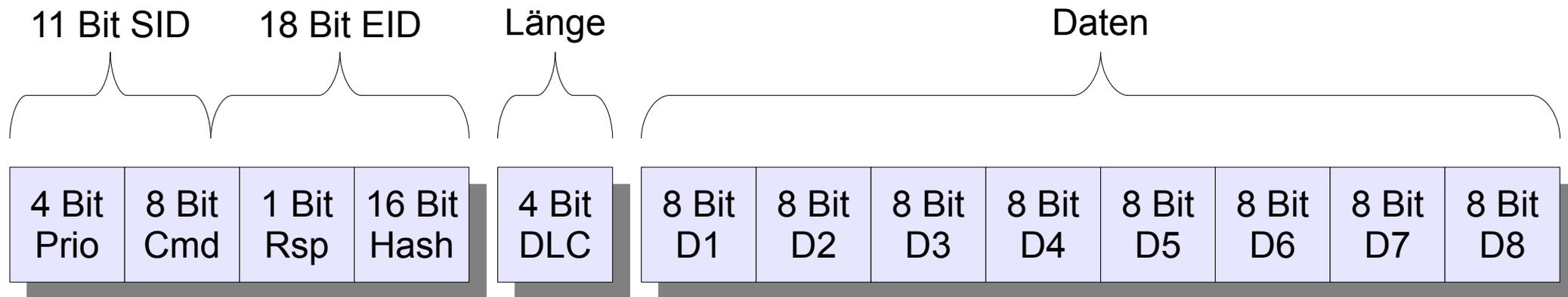
Signal	RJ45	Mini-DIN	Farbe
CAN_H	1	4	Orange-weiß
CAN_L	2	8	Orange
GND	7	2	Braun-weiß
Vcc	8	1	Braun

- Kann Strom für Arduino liefern

Der wird dann aber etwas warm :)



Protokoll (I)



- Jede Nachricht: 29 Bit Header, 8 Byte Daten

- Priorität immer $0x0$

- Hash identifiziert Gerät (z.B. $0xff10$)

- In den meisten Fällen Frage / Antwort

- „set“-Antwort entspricht Anfrage

- „get“-Antwort enthält gewünschte Daten

} Antwortbit
gesetzt

Protokoll (II)

- Die wichtigsten Kommandos sind dokumentiert
 - Steuern von Lokomotiven, Funktionen, Zubehör
 - MM2, DCC, MFX, ...
- Darüber hinaus hilft „Sniffing“
 - Mobile Station und Arduino anschließen
 - Der Arduino sieht alle Kommandos auf dem Bus

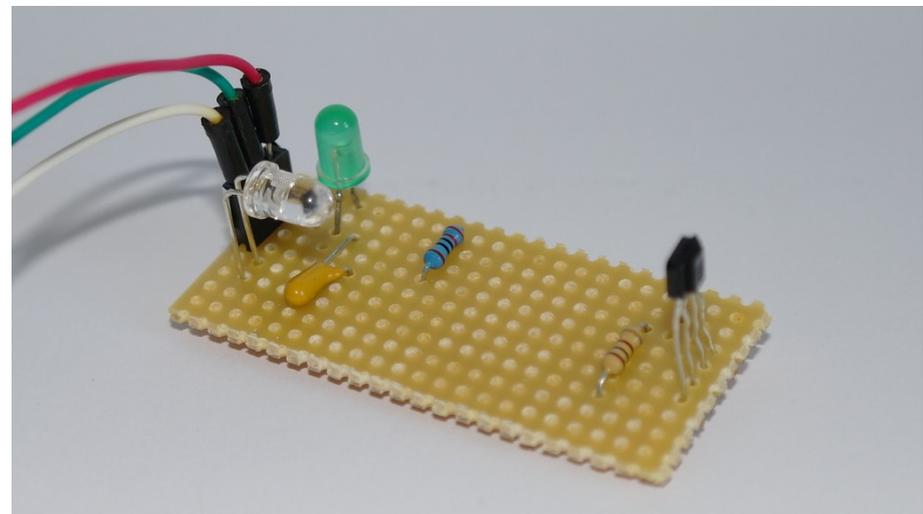
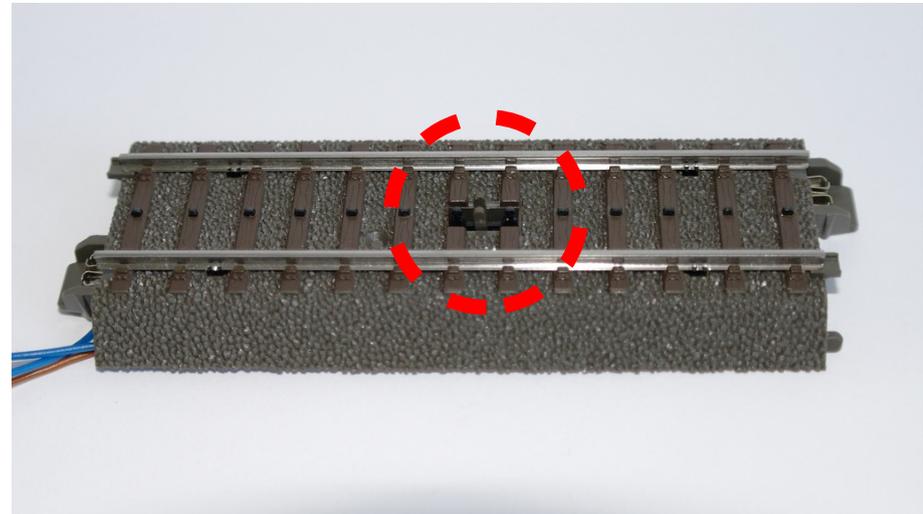
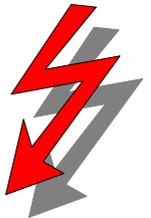
Klingt kompliziert...

Railuino-Bibliothek

- Muss in Sketch eingebunden werden
 - `#include <Railuino.h>`
- **Zentrale Klasse** `TrackController`
 - `get/setLocoDirection()`
 - `get/setLocoSpeed()`
 - `get/setLocoFunction()`
 - `get/setAccessory()`
 - `read/writeConfig()`
- **Viele Beispiel-Sketches**

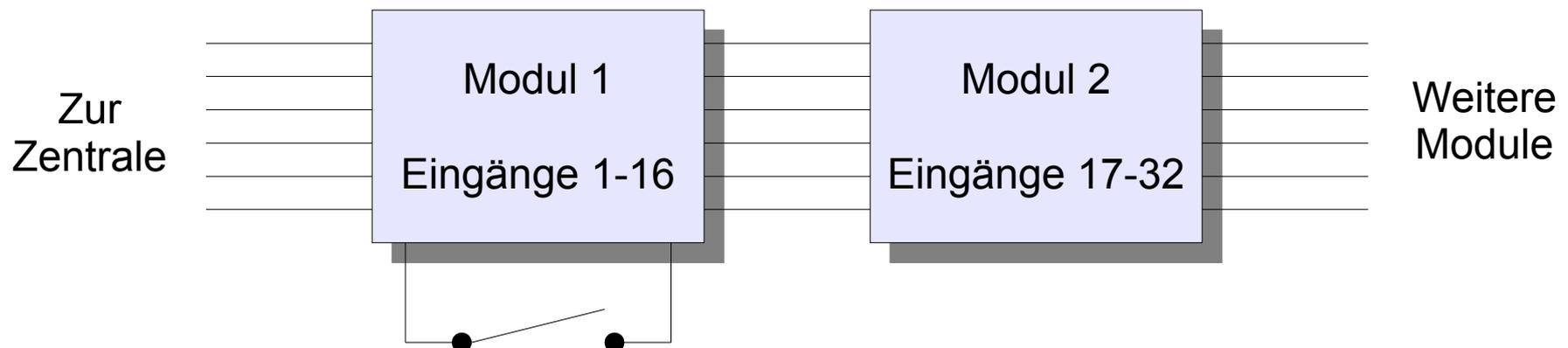
Rückmeldung (I)

- Wissen, wo ein Zug gerade ist
- Möglichkeiten
 - Schaltgleise
 - Kontaktgleise
 - Lichtschranken
 - Magnetkontakte
- Galvanische Trennung!
 - 18V an den Schienen
 - Arduino mag nur 5V



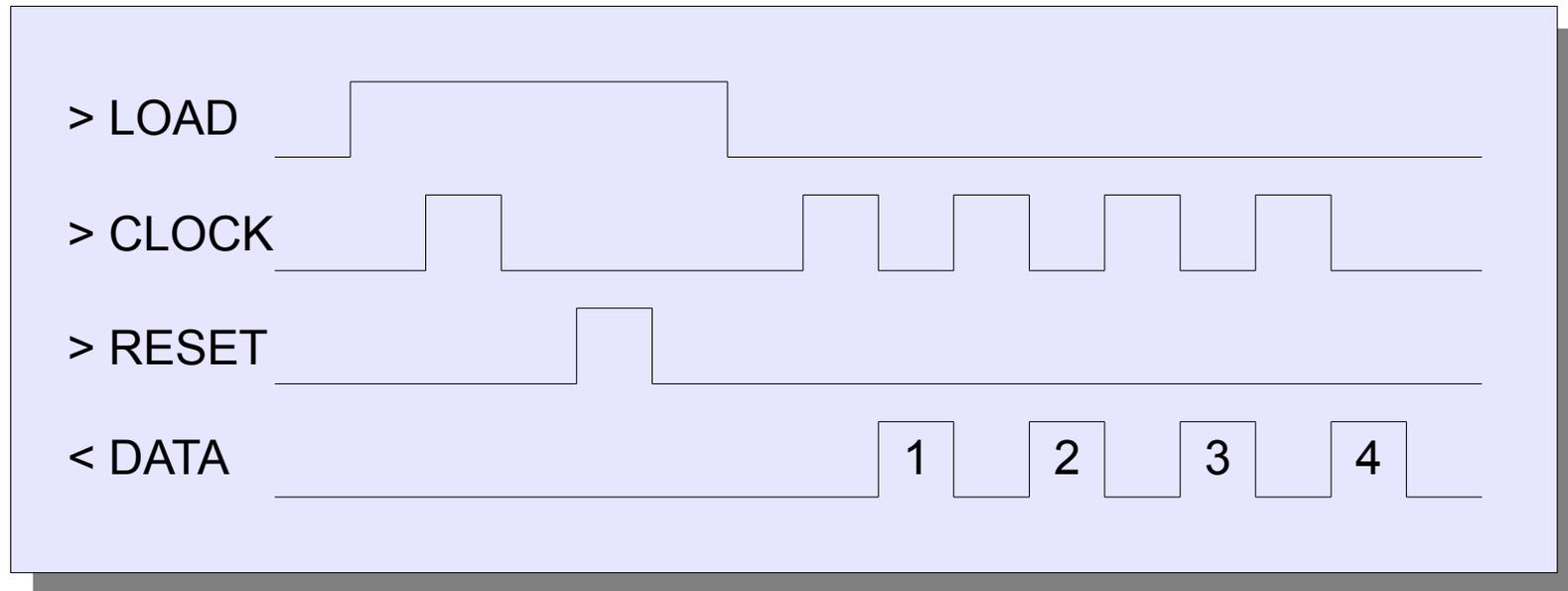
Rückmeldung (II)

- Variante 1: Schalter direkt an Arduino
 - Einfach, skaliert aber nicht beliebig
 - Man muss im richtigen Moment abfragen
- Variante 2: S88-Bus
 - Sechsadriger Standardbus für Rückmeldung
 - Parallele Eingänge, serieller Ausgang, Puffer



Rückmeldung (III)

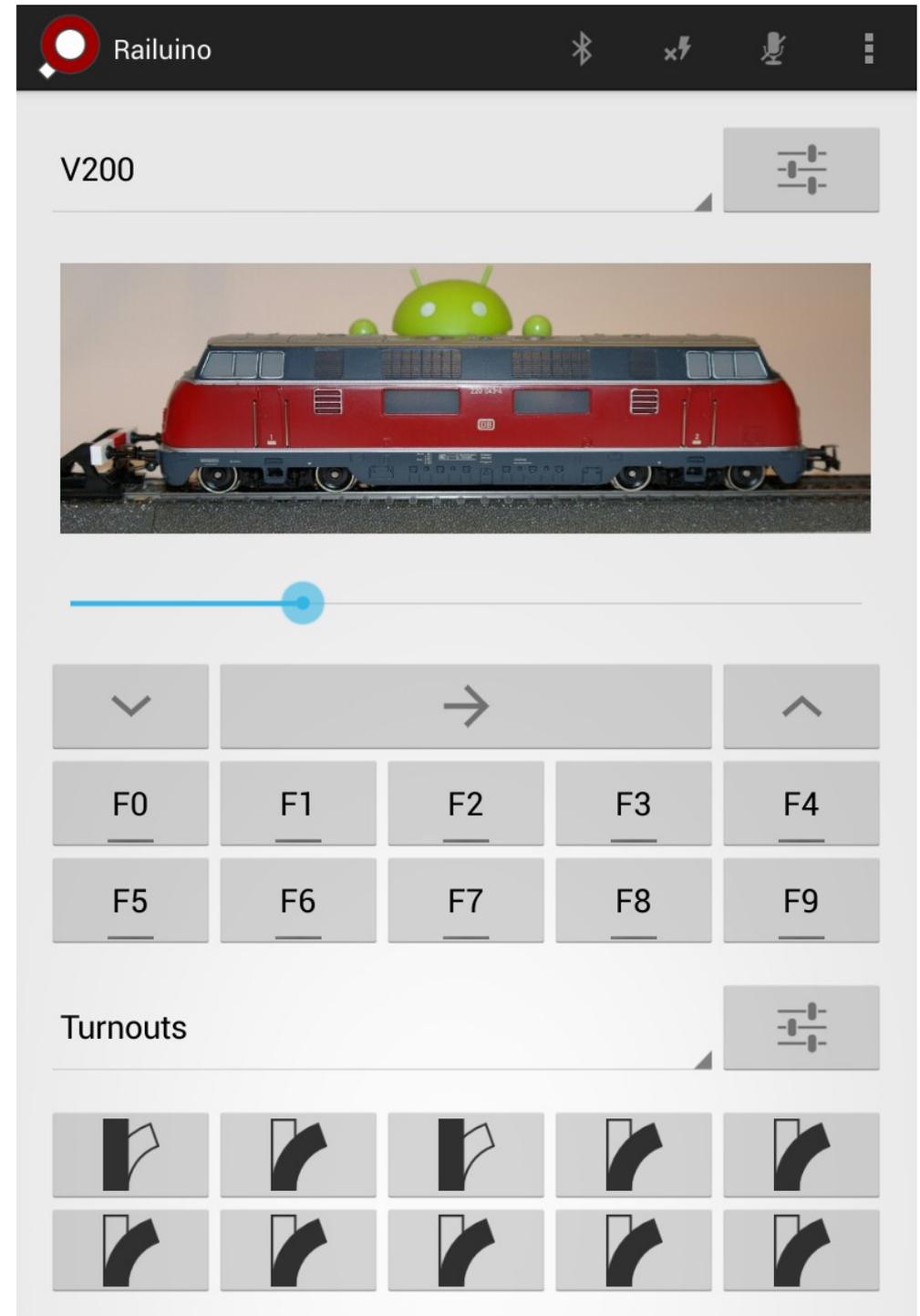
- Timing (via Google)



- Auf dem Arduino leicht zu realisieren
- Klasse `TrackReporterS88`
 - `refresh()` bzw. `getValue()`

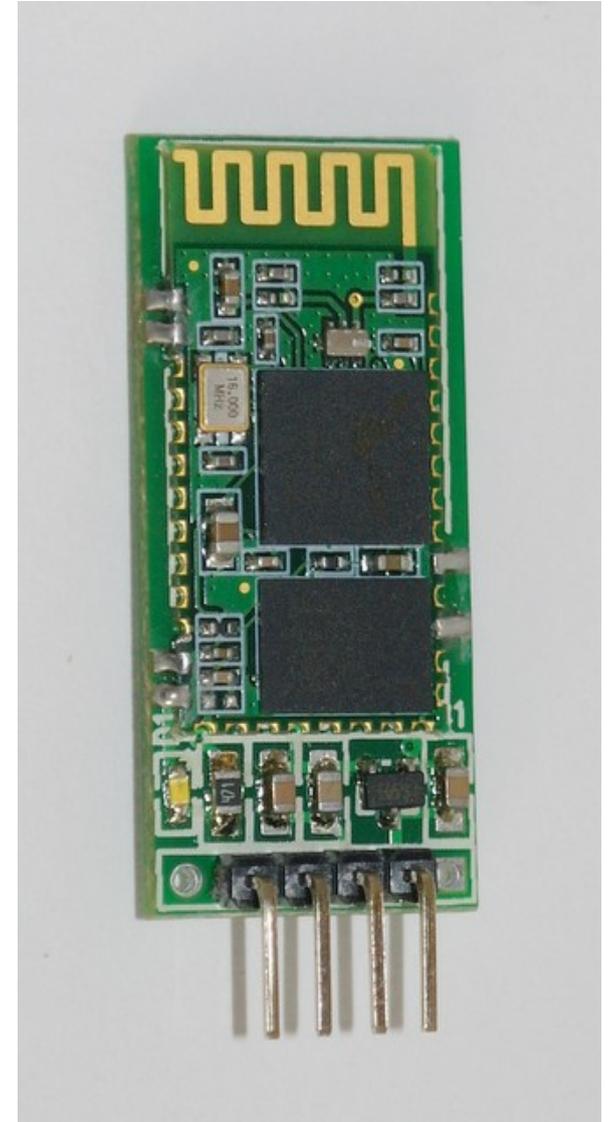
Android App

- Verwalten & Steuern
 - Lokomotiven
 - Zubehörartikel
- Einfaches UI
 - Optimiert für Google Nexus 7 Tablet
- Drahtlose Verbindung zu Arduino
- Derzeit nur Senden, kein Empfangen

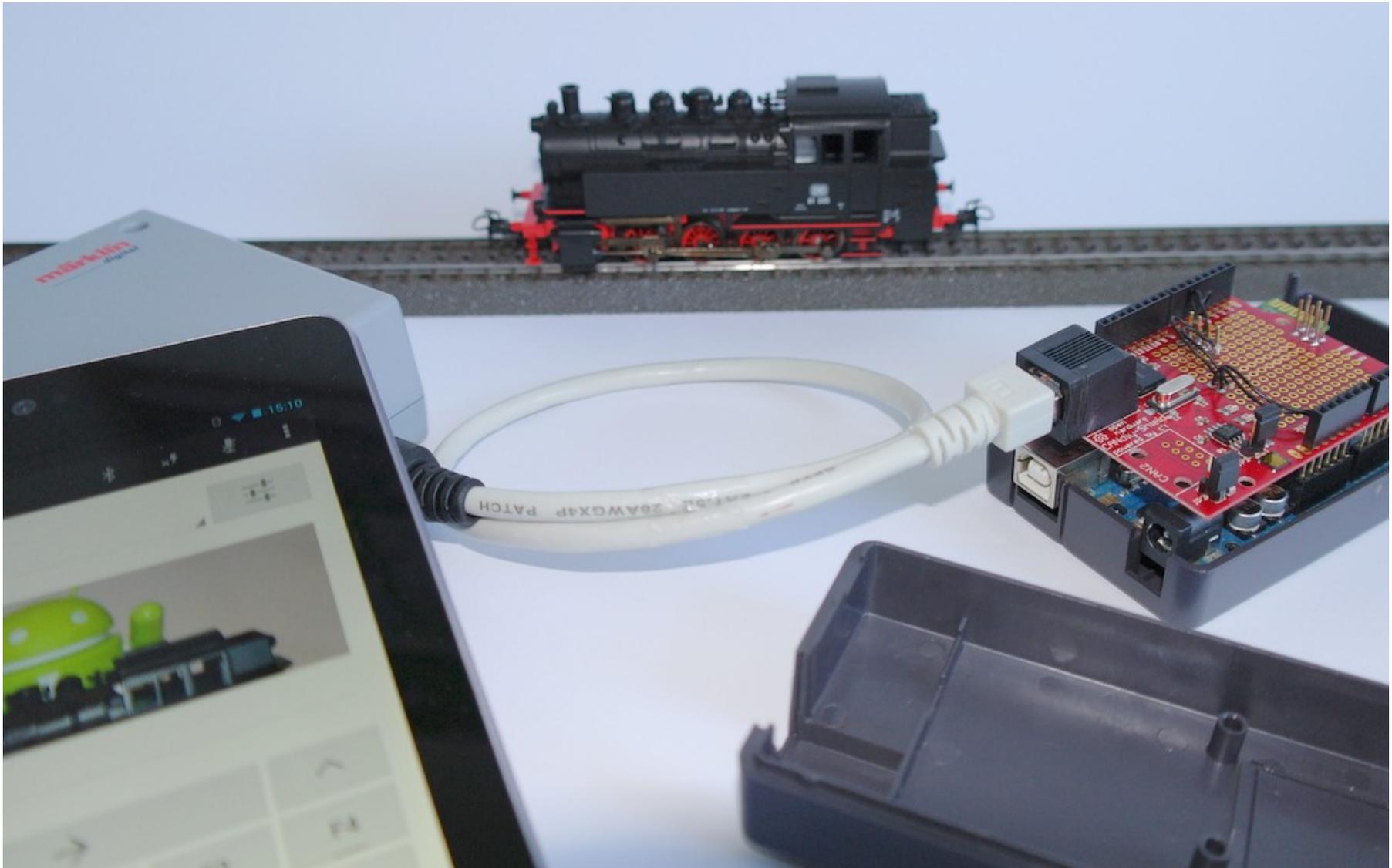


Verbindung zum Arduino

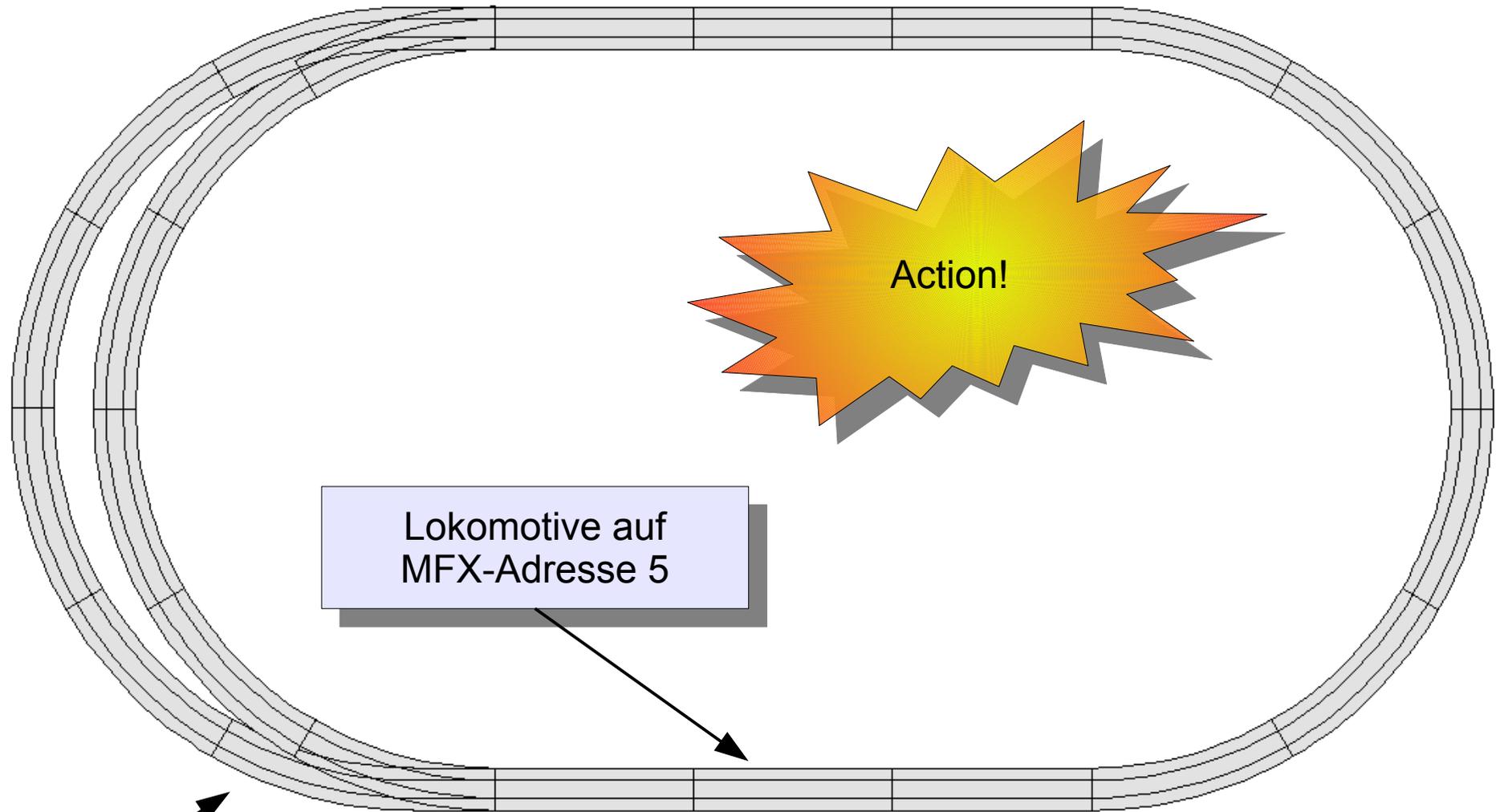
- USB
 - ADK
 - USB Host Shield
 - Android mit OTG-Adapter
- TCP/IP
 - Ethernet oder WiFi Shield
- Bluetooth
 - Günstigste Lösung (~ 10 €)
 - Serielle Schnittstelle



Die Summe der Teile (I)



Beispiele & Demo



Lokomotive auf
MFX-Adresse 5

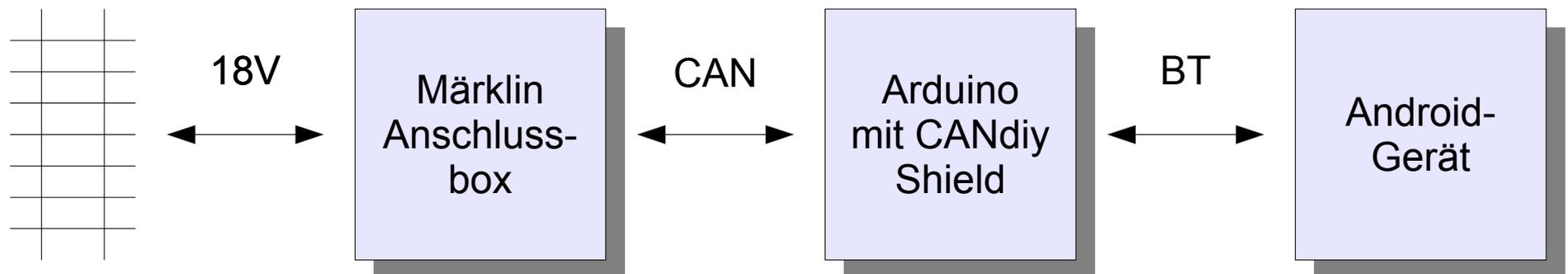
Action!

Weiche auf
MM-Adresse 3

Infrarot-Lichtschanke
an Arduino Pin A0

Zusammenfassung

- DIY-Modellbahnsteuerung
 - Primär für Märklin, aber nicht darauf beschränkt
 - Benötigt wird nur die Anschlussbox
- Autonome Anwendungsfälle
- Interaktive Anwendungsfälle
- Open-Source (LGPL, CC-BY-SA)



Danke für Euer Interesse!



<http://code.google.com/p/railuino>
joerg dot pleumann at gmail dot com