

# DCC-Dekoder

## In Bearbeitung

## DCC-Funktions-Dekoder mit ATtiny85

### Funktions-Dekoder mit Digispark-Board

[Hier](#) wurde ein Funktionsdekoder mit einem ATtiny85-Digispark-Board vorgestellt. Ich habe dafür die Hardware ergänzt, damit auch CVs in Wagen, die mit diesem Decoder ausgerüstet sind, auch auf dem Programmiergleis programmiert und zurückgelesen werden können. Der damit ausgestattete Dekoder hat dann nur noch 3 Ausgänge.

Ein ergänzter Arduino-Sketch für die 3-Port-Variante mit ACK ist dort abgelegt, ebenso ein Sketch für 4 Ports ohne CV-Lesemöglichkeit. Das Programmieren kann aber trotzdem erfolgen  
Ich empfehle in jedem Fall das dort erwähnte Upgrade auf die 300ms-Bootzeit.

Nach ersten positiven Versuchen mit einem fliegendem Aufbau habe ich eine kleine Zusatzplatine entworfen, die rückseitig mit Stiften aufgelötet wird. Hier der überarbeitete Entwurf mit der zusätzlichen Diode.

[Schaltplan](#)

[Bestückungsplan](#)

[Stückliste](#)

[Bestellmöglichkeit](#) der Leiterplatte für Eigenbau bei Aisler

Eine kommerzielle Nutzung ist untersagt.



Hier ein Einbaubeispiel der ersten Version der Platine im [Pwgs/Daa-Wagen](#). Nachträglich musste eine zusätzliche Diode eingefügt werden, damit auch mit dem Stützkondensator ein CV-Schreiben und -Lesen mit dem ACK-Signal zuverlässig funktionieren.

Die Software-Variante mit den 3 Ports wird von mir weiterentwickelt, z.B. mit Funktionsmapping für zwei Ports. Das Projekt habe ich auf [Github](#) abgelegt.

Folgende Funktionen sind aktuell konfiguriert:

F0 Taste schaltet richtungsabhängig beide Schluss-LEDs an PB0 und PB4

F1 Taste schaltet Wagenbeleuchtung an Pb1, die beiden Onboard-LEDs wurden deaktiviert.

---

### Funktions-Dekoder auf Lichtleiste



Auf der Basis der oben genannten Hard- und Software wurde eine Lichtleiste mit einem ATtiny85 aufgebaut. Diese erhält keinen Bootloader, wie das Digispark-Board. Die Programmierung erfolgt über einen ISP-Stecker. Es kann der gleiche Sketch verwendet werden. Es sind ebenfalls drei Ausgänge vorhanden und das ACK-Signal wird verwendet.

Der umgebaute Wagen mit der Lichtleiste ist [hier](#) vorgestellt. Dort sind auch die Leiterplattendaten abgelegt.

# DCC-Zubehör-Dekoder mit ATtiny85

## Entwicklungsboard

Nachdem ich den Funktionsdekoder auf Basis des ATtiny85 für die [Beleuchtung](#) meiner Wagen erfolgreich in Betrieb genommen hatte, habe ich auf gleicher Basis die Entwicklung eines Zubehördekoders begonnen. Basis war ein Arduino-Beispielsketch der [NmraDcc](#)-Bibliothek von [MRRWA](#). Aus dem DCC-Signal wird die Spannung für das Board generiert und es kann auch ein DCC-ACK-Signal erzeugt werden. Damit ist das Lesen und Schreiben der Adresse und einiger Konfigurationsdaten mit dem Programmiergleis-Anschluss möglich.

Der Dekoder hat 4 Ausgänge (Ports), die mit DCC-Befehlen aktiviert und deaktiviert werden können. Mit der Adresse+1 kann bei jedem Port ein Blinken eingeschaltet werden. Die Blinkperiode ist konfigurierbar von 0,5 ... 3,5s. PORT1 und PORT2 können als alternierend verbunden werden (z.B. Weichen, Signale) und dann besteht auch die Möglichkeit, einen Impuls definierter Länge (0,25 ... 8s) zu erzeugen, z.B. bei Weichen ohne Endabschaltung.

CV1/CV9 = Dekoder-Adresse (Modul-Adresse nach NMRA mit 4 Ports und zwei Gates)

CV1 = 6 bit LSB

x x x x x x x x  
- - +--+--+--+ - - - - - LSB 0 ... 63

CV9 = 3 Bit MSB

x x x x x x x x  
- - - - - +--+ - - - - - MSB (0 ... 7) \* 64

CV10

x x x x x x x x  
- - - - - - - + - - - - - „0“ = 4 unabhängige Ports, „1“ = PORT1/PORT2 als alternierende Ports, z.B. Signale, Weichen

CV11

x x x x x x x x  
| | | +--+--+--+ - - - - - 5 bit pulse length if PORT1 and PORT2 are alternating ports,  
number \* 256 ms (256 ms ... 7.93 s)  
+--+ - - - - - - - - - 3 bit for blinking periode in s (0.5 ... 3.5 s)

Für die Adressierung gibt es unterschiedliche Varianten. Ich habe mit Rocrail und der [DCC-Commandstation](#) von DCC-Ex die [MADA-Adressierung](#) ohne Probleme verwenden können. Andere Adressierungsarten kann ich mit dieser Commandstation nicht testen.

An einer Z21 sollte es mit der [PAPA-Adressierung](#) funktionieren, jedoch mit einem Offset von +4.

Im Video ein Beispiel für eine Signalsteuerung durch Rocrail und dem [LocoIO-Keypad](#).

Für meine Modellbahn habe ich erstmal die Signalbilder mit statischer Anzeige vorgesehen:

- HI 1 - ein grünes Licht - Fahrt mit Strecken-Höchstgeschwindigkeit
- HL 3a - ein gelbes Licht, darüber ein grünes Licht - Fahrt mit 40 km/h, dann mit Strecken-Höchstgeschwindigkeit
- HI 10 - ein gelbes Licht oben - „Halt“ erwarten
- HI 13 - ein rotes Licht - „Halt“

Die Signalbilder sind im Rocrail mit den Einstellungen für Mustern für die 4 Ausgänge erstellt worden.

Mit Aktionen können auch weitere Signalbilder (Signal-Muster + blinkende LED) erzeugt werden, wie HI 4 und HI 7. Ich habe mal testweise zwei weitere erzeugt:

- HI 6a - ein gelbes Licht, darüber ein grünes Blinklicht
- HI 9a - ein gelbes Licht, darüber ein gelbes Blinklicht

[attiny\\_signal\\_k.mp4](#)

From:

<http://simandit.de/simwiki/> - **Wiki**

Permanent link:

<http://simandit.de/simwiki/doku.php?id=modellbahn:umbauten:dcc-dekoder>

Last update: **2026/04/14 19:21**

