

DCC-Monitor mit der Hauptplatine der MobaLedLib

[Domapi](#) aus dem Stummi-Forum hat einen Arduino-DCC-Monitor entwickelt, der auch auf der MobaLedLib-Hauptplatine lauffähig ist.

Wer also schon immer wissen wollte, was so eine DCC-Zentrale an Befehlen an die Lok- und Zubehördekoder sendet und das auch noch übersichtlich in Klartext dargestellt, der ist hier genau richtig.

Warum braucht man einen DCC-Monitor?

- Es lassen sich z.B. die Anzahl von Schaltartikel-Befehlen und deren zeitliche Abstände messen und darstellen.
- PC-Steuerungssoftware für Modellbahnen sendet über die Zentrale eine ganze Reihe von Befehlen, z.B. zum Schalten von Fahrstraßen und beim Abbremsen von Loks vor Signalen. Hier ist z.B. interessant, welche Fahstufe zu welchem Zeitpunkt gesendet wird.
- Manche Zentralen senden die Befehle mehrfach, andere wiederum senden keine off-Befehle für Weichen. Mit dem Monitor kommt man all diesen Phänomenen auf die Schliche.

Neueste Sketch-Versionen, Schaltplan und benötigte Bauteile

Aktuelle **Sketch**-Versionen, weitere Informationen und Hilfe zum Programm gibt es im [Stummiforum](#).

Dort findet sich auch ein **Schaltplan** sowie ein Reichelt-Warenkorb der **benötigten Bauteile**.

Programm-Beschreibung

Im Netz findet man mehrere Arduino-Projekte, die DCC-Gleissignale auswerten und versuchen, sie in verständlichem Klartext am seriellen Monitor der Arduino-IDE darstellen. Da manche Programme nicht sauber liefen, teilweise Auswertungsfehler enthalten waren und ein paar Features fehlten, entstand hier kurzerhand ein eigener DCC-Monitor, DCC-Sniffer oder DCC-Schnüffler auf Basis der DCC-NMRA-Library.

Über ein kleines Menü lässt sich die Darstellung am seriellen Monitor des Arduino beeinflussen, also z.B. welche DCC-Befehle überhaupt angezeigt werden und ob alle Befehle oder nur diejenigen mit neuen Kommandos. So eine Zentrale wiederholt sich nämlich fortlaufend. Vor allem bei Loks werden permanent Geschwindigkeitsbefehle und die Funktionen F0 - F4 gesendet. Etwas seltener folgen die anderen Funktionsbefehle etc. So kann man sich z.B. auf Lok-Befehle oder nur auf das Zubehör fokussieren.

Schaltet man bei Loks „nur neue Lok-Pakete ...“ (Option 4) aus, wird ein endloser Stream von DCC-Befehlen angezeigt. So schnell kann man gar nicht lesen, wie die Textzeilen am Bildschirm durchlaufen.

Ganz interessant ist auch, welche und wie viele DCC-Befehle beim CV-Lesen und -Schreiben aufs Gleis gelegt werden. Am Ende des Programms kann nach Belieben die Liste der Loknamen angepasst werden. Dann wird die Ausgabe der Lokadressen nicht ganz so kryptisch.

Der Sketch funktioniert auch mit der MobaLedLib-Hauptplatinen. Dazu muss man einfach den DCC-Monitor-Sketch auf den DCC-Nano aufspielen. Und schon lassen sich auf dem seriellen Arduino-Monitor die von der Zentrale gesendeten DCC-Befehle in Klartext verfolgen. Das sollte mit **allen Hauptplatinen-Versionen** funktionieren.

Und falls ihr dann doch wieder die MobaLedLib per DCC ansteuern wollt: nicht vergessen, vorher den entsprechenden Sketch auf den DCC-Nano aufspielen, sonst funktioniert es nicht!

Zusatzfunktionen auf der MLL-Hauptplatine V1.0



Auf der Hauptplatine Version 1.0 sind einige Bauelemente vorgesehen, die auch das Lesen und Schreiben von CVs auf dem Programmiergleis ermöglichen. Hierfür muss der Dekoder (hier also der DCC-Monitor bzw. der DCC-Nano) Bestätigungen an die Zentrale senden. Dies sind 6ms lange ACK-Impulse, so genannte Acknowledgements.

In den neueren Versionen hat Hardi als *alter Schwabe* den Schaltungsteil mit dem Optokoppler CNY17 und dem Transistor weggelassen. 😞

Direkt unter dem DCC-Nano auf der Hauptplatine V1.0 befindet sich die Hardware für die Erzeugung dieser ACK-Signale für das Schreiben und Lesen von CVs.

Damit diese Signale auch wirklich auf das Gleis gegeben werden, muss man einfach eine Zeile im Sketch an die Verdrahtung der Hauptplatine anpassen, da hier das Acknowledgement-Signal am D4 Pin erzeugt wird

(im Original-Sketch wird es am A5-Pin ausgegeben); also einfach „A5“ durch „4“ ersetzen (ca. Zeile 100 im Sketch).

```
const byte DccAckPin = 4; // Arduino-Pin zur Erzeugung eines ACK-Signals
```

Dann kann man den DCC-Monitor auf der Hauptplatine V1.0 auch am Programmiergleis betreiben und CVs auslesen und schreiben.

Achtung: Dies ist nur auf der V1.0-Platine möglich, bei neueren Platinen-Versionen fehlen die Bauteile hierfür. Die notwendigen Komponenten sind im Schaltplan unten links zu sehen.

[hauptplatine_100_1-0_schaltplan.jpg](#)

Beispielausgabe auf dem seriellen Monitor

So sieht das Ganze am Bildschirm aus (beispielhafte Ausgabe):

```
NMRA DCC Monitor V 1.4
Sketch-Upload am: ...
? = Zeige Tastaturbefehle für den seriellen Arduino-Monitor
```

S t a t i s t i k

```
-----
Zeitraum [sec]           :      315
Anzahl empfangene Bytes:  140455
Gültige Kommandos       :   38626
Ungültige Kommandos     :         0
Idle-Pakete             :   12002
Geschwindigkeitsbefehle:  13723
F0 - F4 Funktionen     :   10476
F5 - F8 Funktionen     :    1494
F9 - F12 Funktionen    :    236
F13 - F20 Funktionen   :    234
F21 - F28 Funktionen   :    182
F29 - F36 Funktionen   :     26
Spezialbefehle Lok     :     20
Zubehör-Befehle        :     40
Dekoder-Reset-Befehle  :    162
Zubehör-CV-Befehle     :         0
Lok-CV-Befehle         :         0
Programmiergleisbefehle:    51
Acknowledgments        :     36
Counter Lok            :    208
Counter Acc            :         2
```

Tastaturbefehle für den seriellen Monitor:

```
1 = Anzeige Loks ein/aus           ein
2 = Anzeige Zubehör ein/aus        ein
3 = Anzeige CV-Befehle ein/aus     ein
4 = Nur neue Lok-Pakete anzeigen ein
5 = Nur neue Zubehör-Pakete anzeigen ein/aus ein
6 = Nur neue CV-Befehle ein/aus    ein
7 = Statistik anzeigen
? = Befehle anzeigen
```

```
Lok    4    P4 KPEV FLM           F0   f1   f2   f3   f4
0000-0100 1001-0000 1001-0100
Lok    23   BR 23 Trix           -->> 116
0001-0111 0011-1111 1111-0101 1101-1101
```

Lok	23	BR 23 Trix	F0	f1	f2	f3	f4
0001-0111	1001-0000	1000-0111					
Lok	3910	BR 39 105 grün	-->	115			
1100-1111	0100-0110	0011-1111	1111-0100	0100-0010			
Lok	3910	BR 39 105 grün	F0	f1	f2	f3	f4
1100-1111	0100-0110	1001-0000	0001-1001				
Lok	280	BR280 Trix	-->	104			
1100-0001	0001-1000	0011-1111	1110-1001	0000-1111			
Lok	280	BR280 Trix	f0	f1	f2	f3	f4
1100-0001	0001-1000	1000-0000	0101-1001				
Lok	80	BR 80 Trix	-->	115			
0101-0000	0011-1111	1111-0100	1001-1011				
Lok	80	BR 80 Trix	F0	f1	f2	f3	f4
0101-0000	1001-0000	1100-0000					
Lok	3918	BR 39-186 FLM	-->	104			
1100-1111	0100-1110	0011-1111	1110-1001	0101-0111			
Lok	3918	BR 39-186 FLM	F0	f1	f2	f3	f4
1100-1111	0100-1110	1001-0000	0001-0001				
Lok	100	VT10-1	f5	f6	f7	f8	
0110-0100	1011-0000	1101-0100					
Lok	60	V 60 ESU	f5	f6	F7	f8	
0011-1100	1011-0100	1000-1000					
Lok	36	S3/6 Roco Zimo	f5	f6	f7	f8	
0010-0100	1011-0000	1001-0100					
Lok	98	VT 98	f5	f6	f7	f8	
0110-0010	1011-0000	1101-0010					
Lok	5501	G 5/5 Bay.	F5	f6	f7	f8	
1101-0101	0111-1101	1011-0001	0001-1001				
Lok	1016	T16 KPEV FLM	F5	f6	f7	f8	
1100-0011	1111-1000	1011-0001	1000-1010				
Lok	10	P10 Trix	F5	f6	f7	f8	
0000-1010	1011-0001	1011-1011					
Lok	79	V80 Roco	f5	f6	f7	f8	
0100-1111	1011-0000	1111-1111					
Lok	236	V 236 Bawa Zimo	f5	f6	f7	f8	
1100-0000	1110-1100	1011-0000	1001-1100				
Lok	4417	BR 44 Trix	F5	f6	f7	f8	
1101-0001	0100-0001	1011-0001	0010-0001				
Lok	5031	BR 50-319 Roco	F5	f6	f7	f8	
1101-0011	1010-0111	1011-0001	1100-0101				
Lok	3624	S3/6 Trix	F5	F6	f7	f8	
1100-1110	0010-1000	1011-0011	0101-0101				
DCC-Adresse	6 (2 : 2)	B On					
1000-0010	1111-1010	0111-1000					
DCC-Adresse	6 (2 : 2)	B Off					
1000-0010	1111-0010	0111-0000					
DCC-Adresse	6 (2 : 2)	A On					
1000-0010	1111-1011	0111-1001					
DCC-Adresse	6 (2 : 2)	A Off					

```

1000-0010 1111-0011 0111-0001
DCC-Adresse 7 ( 2 : 3) A On
1000-0010 1111-1101 0111-1111
DCC-Adresse 7 ( 2 : 3) A Off
1000-0010 1111-0101 0111-0111
DCC-Adresse 6 ( 2 : 2) B On
1000-0010 1111-1010 0111-1000
DCC-Adresse 6 ( 2 : 2) B Off
1000-0010 1111-0010 0111-0000
DCC-Adresse 7 ( 2 : 3) B On
1000-0010 1111-1100 0111-1110
DCC-Adresse 7 ( 2 : 3) B Off
1000-0010 1111-0100 0111-0110
Lok 5 EP5 bayr. FLM -->> 118
0000-0101 0011-1111 1111-0111 1100-1101
Lok 5 EP5 bayr. FLM -->> 120
0000-0101 0011-1111 1111-1001 1100-0011
Lok 5 EP5 bayr. FLM -->> 121
0000-0101 0011-1111 1111-1010 1100-0000
Lok 5 EP5 bayr. FLM -->> 119
0000-0101 0011-1111 1111-1000 1100-0010
Lok 5 EP5 bayr. FLM -->> 118
0000-0101 0011-1111 1111-0111 1100-1101
Lok 5 EP5 bayr. FLM -->> 116
0000-0101 0011-1111 1111-0101 1100-1111
Lok 5 EP5 bayr. FLM -->> 115
0000-0101 0011-1111 1111-0100 1100-1110
Lok 5 EP5 bayr. FLM -->> 114
0000-0101 0011-1111 1111-0011 1100-1001
Lok 5 EP5 bayr. FLM -->> 115
0000-0101 0011-1111 1111-0100 1100-1110
Lok 5 EP5 bayr. FLM -->> 116
0000-0101 0011-1111 1111-0101 1100-1111
Prg CV1 Lese Bit #0
0111-1000 0000-0000 1110-0000 1001-1000
Prg Dekoder-Reset-Befehl
0000-0000 0000-0000 0000-0000
Prg CV1 Lese Bit #1
0111-1000 0000-0000 1110-0001 1001-1001
Prg Dekoder-Reset-Befehl
0000-0000 0000-0000 0000-0000
Prg CV1 Lese Bit #2
0111-1000 0000-0000 1110-0010 1001-1010
Prg Dekoder-Reset-Befehl
0000-0000 0000-0000 0000-0000
Prg CV1 Lese Bit #3
0111-1000 0000-0000 1110-0011 1001-1011
Prg Dekoder-Reset-Befehl
0000-0000 0000-0000 0000-0000
Prg CV1 Lese Bit #4
0111-1000 0000-0000 1110-0100 1001-1100

```

```
Prg Dekoder-Reset-Befehl
0000-0000 0000-0000 0000-0000
Prg CV1 Lese Bit #5
0111-1000 0000-0000 1110-0101 1001-1101
Prg Dekoder-Reset-Befehl
0000-0000 0000-0000 0000-0000
Prg CV1 Lese Bit #6
0111-1000 0000-0000 1110-0110 1001-1110
Prg Dekoder-Reset-Befehl
0000-0000 0000-0000 0000-0000
Prg CV1 Lese Bit #7
0111-1000 0000-0000 1110-0111 1001-1111
Prg Dekoder-Reset-Befehl
0000-0000 0000-0000 0000-0000
Prg CV1 Lese CV
0111-0100 0000-0000 0001-0000 0110-0100
Prg Dekoder-Reset-Befehl
0000-0000 0000-0000 0000-0000
Prg CV1 Schreibe CV = 16
0111-1100 0000-0000 0001-0000 0110-1100
Prg Dekoder-Reset-Befehl
0000-0000 0000-0000 0000-0000
```

From: <https://wiki.mobaledlib.de/> - MobaLedLib Wiki

Permanent link: <https://wiki.mobaledlib.de/anleitungen/spezial/dcc-signal-auswertung?rev=1731996476>

Last update: **2024/11/19 06:07**

