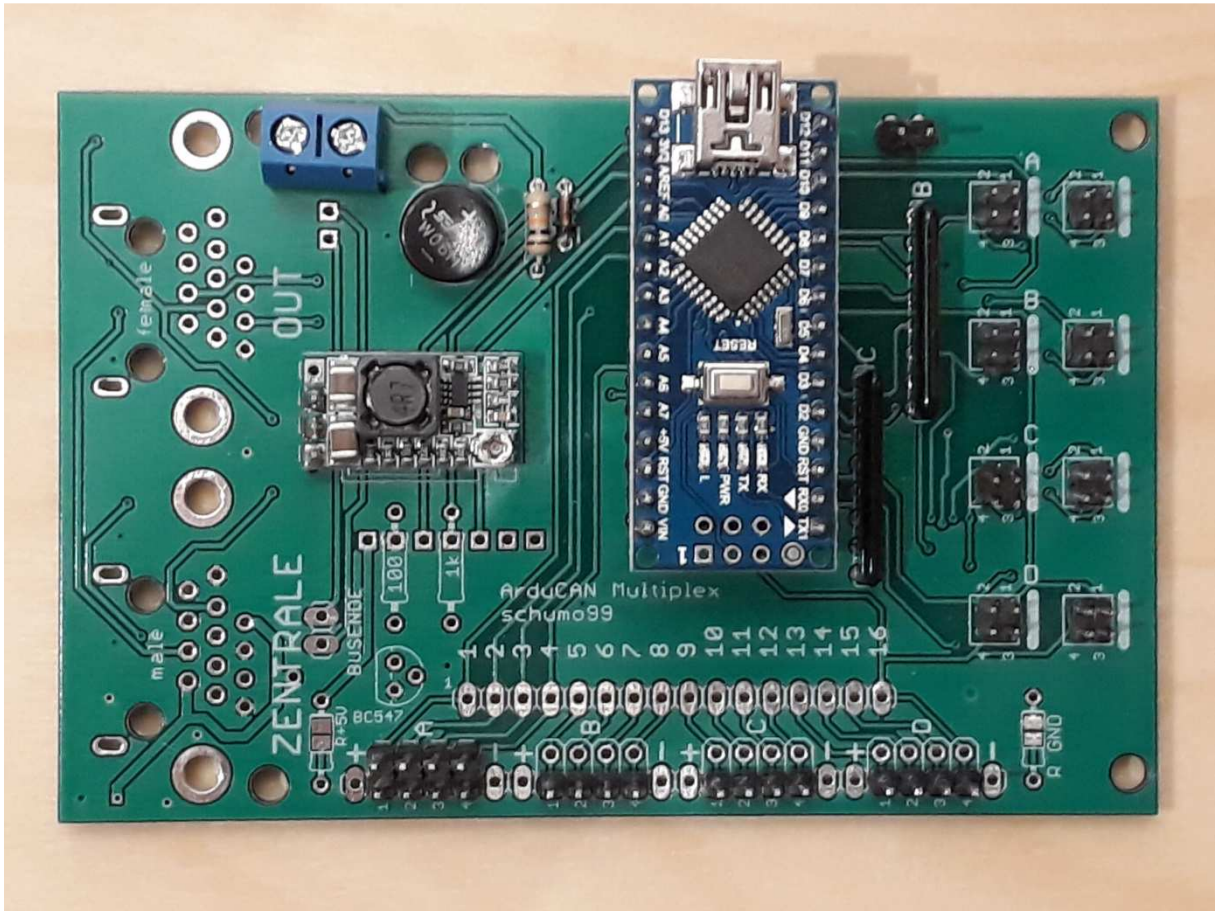
ArduCAN-Multiplex

Der ArduCAN Multiplex ist ein DCC-Decoder auf Arduino-Basis zum Eigenbau. Er steuert bis zu 48 LEDs im Multiplex-Betrieb an und kann sowohl Lichtsignale, als auch Ampeln, Lauflichter, usw. darstellen.



Die große Anzahl an steuerbaren Lichtern ist dem Charlie-/Multiplexing zu verdanken. Dabei kann man mit N Leitungen bis zu N*(N-1) LEDs ansteuern. Erreicht wird das dadurch , dass an jedem Anschluss Anode bzw. Kathode der LED sitzen kann, und die LEDs somit antiparallel an jedem möglichen Anschluss liegen. Der Dekoder teilt seine Pins in 4 Gruppen zu jede 4 Anschlüsse auf. Damit erzielt man 4*(4*3) = 48 steuerbare LEDs. (Theoretisch ist auch eine große Matrix mit 16*15=240 LEDs oder die Aufteilung auf 2 Matrixes mit jeweils 8*7=56 (also gesamt 112) LEDs denkbar. Diese Aufteilungen wurden aber derzeit nicht realisiert). Das Prinzip ist auf Wikipedia gut erklärt (https://en.wikipedia.org/wiki/Charlieplexing).

Bei der Programmierung wurde Wert darauf gelegt, dass so viel Logik wie möglich im Dekoder selbst integriert ist und nur wenige Stellbefehle über das DCC-Signal nötig sind. Daher können z.B.

> Ampelschaltungen einer ganzen Kreuzung völlig autonom erfolgen, oder auch bis zur nächsten Rot-/bzw. Grünphase. Alle Zwischenschritte werden vom Dekoder aber selbstständig übernommen.

oder auch

> Vorsignale selbstständig aufgrund von Befehlen an Hauptsignale angesteuert werden

Features:

- Bis zu 8 Signale (4 Haupt- und 4 Vorsignale) können gleichzeitig angesteuert werden
- Signalverknüpfung, d.h. der Stellbefehl für ein Hauptsignal kann auch gleich das entsprechende Vorsignal mitstellen
- Signallogik: Wenn das Hauptsignal Hp0 oder Sh1 zeigt, wird das Vorsignal am gleichen Mast ausgeschaltet
- Alle Signalbilder frei konfigurierbar
- Die Helligkeit aller LEDs ist frei konfigurierbar
- Möglichkeit alle LEDs auch Blinken zu lassen (Blinkfrequenz einstellbar)
- Auf- und Abblendmöglichkeit aller LEDs, Dunkeltastung zwischen den Signalbildern frei konfigurierbar
- Bis zu 4 Ampelkreuzungen mit jeweils Haupt-/Nebenstrasse und Fussgängerampel sowie gelbes Blinklicht für Abbieger (frei konfigurierbar, z.B. Schaltzeiten, Helligkeit, Blinken, …)
- Bis zu 4 Lauflichter (frei konfigurierbar, z:b. Geschwindigkeit, Zahl der beteiligten LEDs, Blendzeit, Helligkeit, ...)
- Simulation von Licht in Häusern: Zufallsgesteuertes Ein und Ausschalten von Licht in einzelnen Räumen, Anzahl der Räume mit Licht und Lichtwechselrate frei konfigurierbar
- DCC Adressen frei wählbar über USB-Programmierung, alternativ: DCC-Adresse lernen über Taster
- Automatisches Herstellen des letzten Zustandes bei Wiedereinschalten des DCC-Signals

Aufbau:

- Nur wenig Bauteile, komplett ohne SMD
- Vielfältige Anschlussmöglichkeiten für Signale, LEDs, … über Stiftleisten

Ansteuerung:

- Stromversorgung und Ansteuerung über DCC

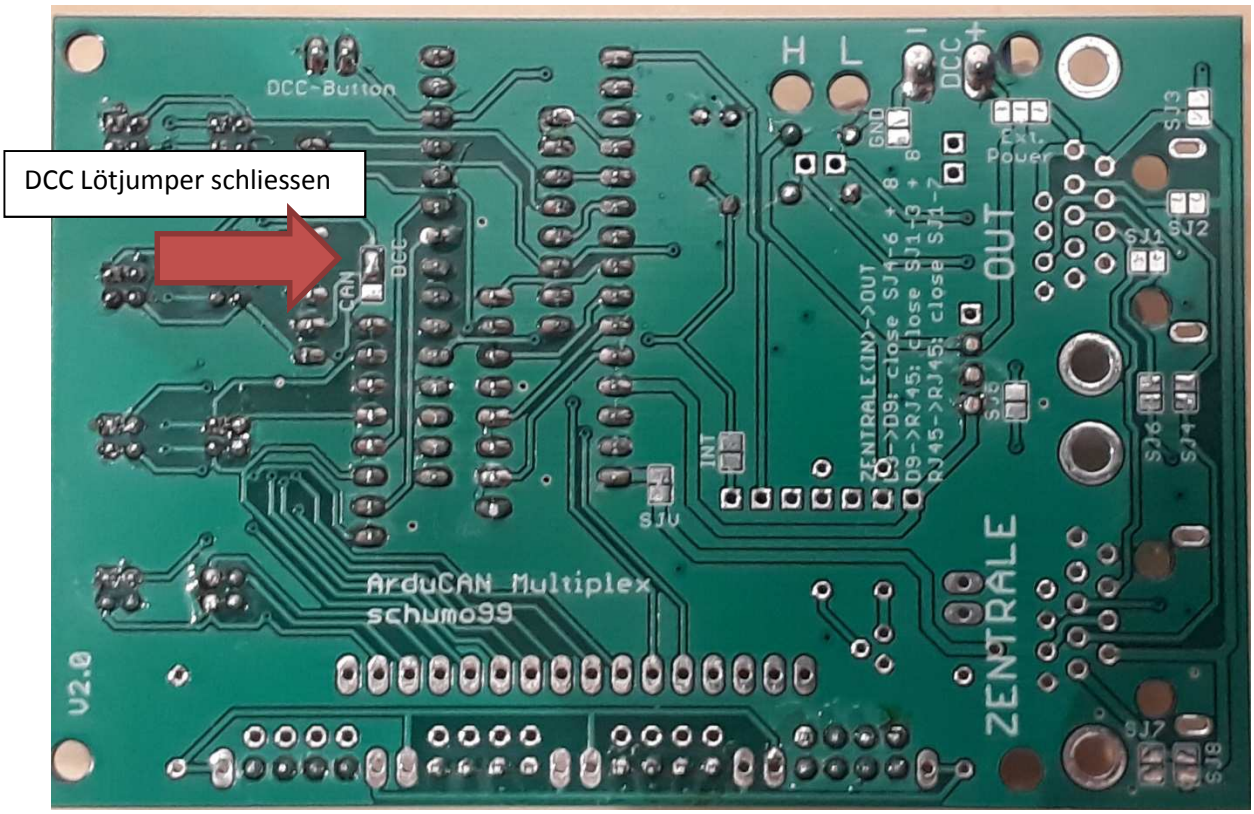
Konfiguration:

- Über integrierte USB-Schnittstelle

Der Verfasser übernimmt keine Haftung für eventuelle Schäden oder das Nichtfunktionieren der Schaltung. Hinweise und/oder Verbesserungsvorschläge sind immer erwünscht.

Aufbau Hardware

Die Bauteile gemäß nachstehender Tabelle einlöten und auf der Rückseite den Lötjumper „DCC“ schließen. Den Arduino aufstecken. Der USB-Anschluss zeigt zum Platinenrand.



Anzahl	Bauteil	Zweck
4x	Widerstandsnetzwerk SIL 8-4 150	Vorwiderstand für LEDs
1x	Brückengleichrichter B70C1500RUND	Gleichrichter für DCC-Signal
1x	DC/DC-Wandler für 5V oder alternativ IC 7805	Spannungsversorgung
1x	Widerstand 10kOhm (z.B. 1/4W 1k)	Vorwiderstand DCC
1x	Zenerdiode 4,7V z.B. ZF 4,7	Spannungsbegrenzung DCC
2x	Buchsenleiste für Arduino: z.B. BL1X20G8 2,54 Kürzen auf 15 Pins	Zum Anschluss der Arduinos

Vorschlag für externe Anschlüsse:

Anzahl	Bauteil	Zweck
1x	AKL101-02	Anschlussklemme für DCC
1x	SL 1X40G 2,54 (muss selbst geteilt werden)	Stiftleiste zum Anschluss der LEDs
1x	SL 2X20G 2,00 (muss selbst geteilt werden)	Stiftleiste zum Anschluss von Multiplex-Signales

Alle anderen auf der Platine vorgesehenen Lötjumper oder Bauteile sind für den Betrieb nicht notwendig.

Aufspielen der Software

Den Arduino (mit Prozessor ATMega328PB; das B ist wichtig!!) mit der Software programmieren. Dazu das file ArduCAN-Signal.hex einspielen. Man kann das mit beispielsweise mit avrdude machen (Download: http://stefanfrings.de/avr_tools/avrdude-6.3-mingw32.zip). Das o.g. HEX-file und die Dateien von avrdude in den gleichen Ordner kopieren, in der Kommoandozeile von Windows zu diesem wechseln und folgenden Befehl eingeben:

avrdude -carduino -patmega328p -PCOMxx –b57600 -D -F -Uflash:w:arducan-signal10.hex:i

Dabei stehen die beiden xx für die Nummer des zugewiesenen COM-Ports.

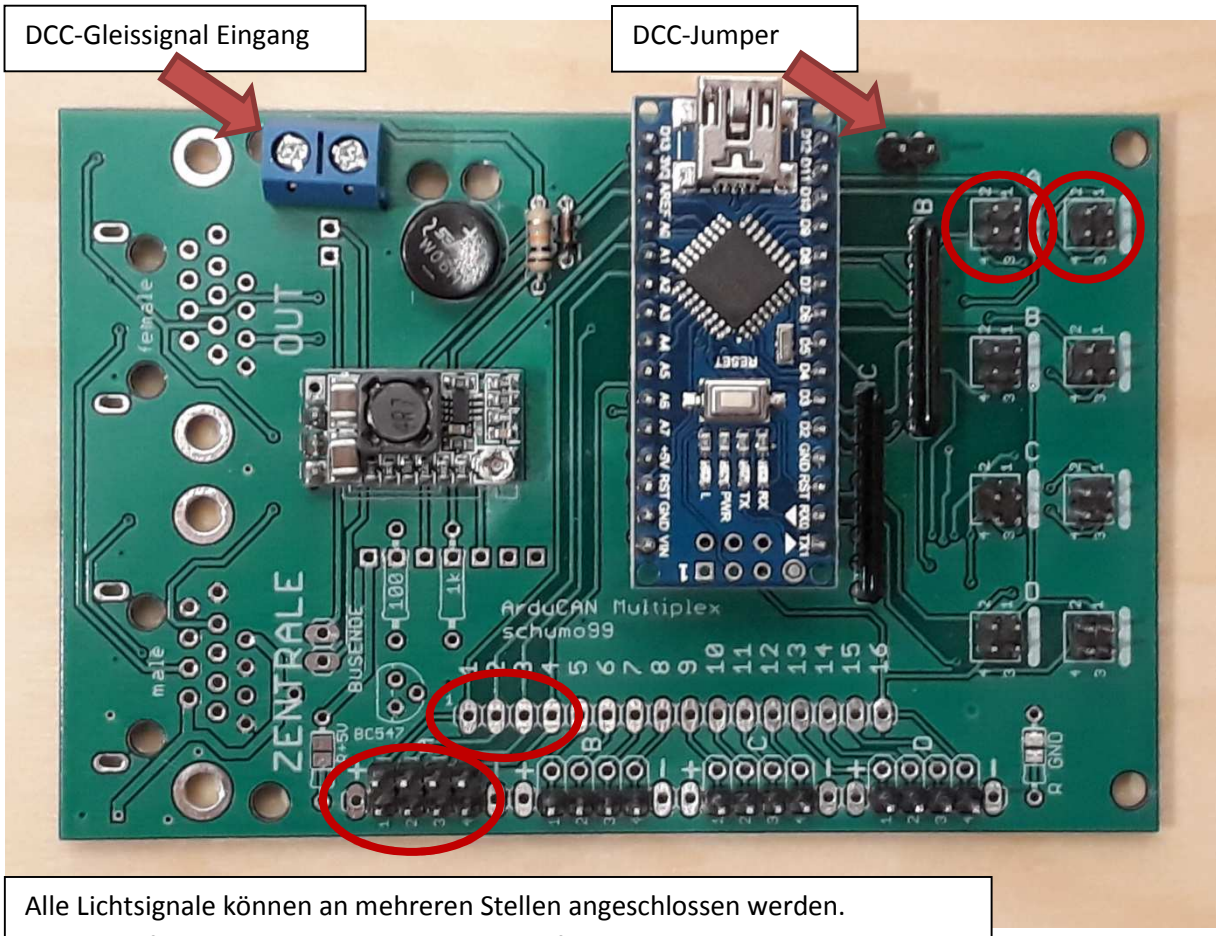
Ich biete auch an einen fertig programmierten Arduino zur Verfügung zu stellen. Das erspart unter Umständen viel Frust.

Betrieb und externe Anschlüsse

Über die polige Anschlussklemme wird das DCC-Signal eingegeben. Die Signale, Ampeln, Lauflichter oder sonstige LEDs werden an die Stiftleisten angeschlossen. Die Konfiguration sollte über die USB-Schnittstelle erfolgen, da hierüber alle Möglichkeiten zur Verfügung stehen.

Anzeigen am Dekoder selbst:

LED	Verhalten	Bedeutung
RX	aus	Normalzustand
	flackert kurz	Bei Übertragung von Einstellbefehlen vom PC
TX	aus	Normalzustand
	flackert kurz	Bei Übertragung von Statusmeldungen des Dekoders zum PC
Power/PWR	An	Indikator für Stromversorgung
LED / L	Dauernd an	Adresslernmodus: Warte auf DCC-Befehl
	Dauernd Blinkend	Prozessorfehler. Software nicht richtig geladen
	Einmalig Blinkend	DCC Zubehörbefehl empfangen



Alle Lichtsignale können an mehreren Stellen angeschlossen werden. Beispielhaft sind die identischen Anschlüsse für das Signal A rot umrandet dargestellt. Für die Anschlüsse B-D gilt dies äquivalent.

Grundeinstellung

Der Dekoder hört in der Grundeinstellung auf 16 aufeinanderfolgende Adressen. Die DCC-Basisadresse kann über den DCC-Jumper eingestellt werden. Dazu wird dieser kurz überbrückt. Der Arduino signalisiert den Adress-Lernmodus mit einem dauernden Leuchten der LED „L. Die nächste gesendete DCC-Adresse stellt die Basisadresse des Dekoders dar.

Die Grundeinstellung der Software geht vom Anschluss von Multiplexsignalen aus. Alle Anschlüsse sind daher wie folgt vorkonfiguriert:

- Multiplexsignale an allen Anschlüssen
- Signallogik eingeschaltet, d.h. bei durch das Hauptsignal verbotener Fahrt (Hp0 oder Sh1) wird kein Vorsignal am gleichen Anschluss (=Mast) gestellt.
- Haupt- und Vorsignale werden einzeln angesteuert.

Es steht eine Vielzahl an Einstellmöglichkeiten zur Verfügung, die jedoch nur über eine Verbindung des Dekoder mit dem PC mit Hilfe der USB-Schnittstelle ermöglicht wird. Für die Signalkonfiguration sind für Einsteiger zunächst folgende Befehle u.U. besonders zu empfehlen:

<table>
<tr><td><A:B></td><td>Dem Hauptsignal an A wird das Vorsignal an B zugeordnet. Dadurch kombiniert man das Vorsignal B mit dem Hauptsignal A. Ab jetzt werden Stellbefehle an A wie folgt an B weitergeleitet.

<table><tr><th>Hauptsignal A</th><th>Vorsignal B</th></tr><tr><td>Hp0/Sh1</td><td>Vr0</td></tr><tr><td>Hp1</td><td>Vr1</td></tr><tr><td>Hp2</td><td>Vr2</td></tr></table>
Statt A und B in obigem Konfigurationsbefehl sind natürlich auch andere Kommandos möglich.</td></tr>
<tr><td><Reset></td><td>Wiederherstellung der Grundeinstellung</td></tr>
</table>

USB-Schnittstelle/Verbindung mit dem PC

Wichtig: Zuerst das DCC-Signal vom Dekoder entfernen. Der Dekoder, die DCC-Zentrale und/oder der PC könnten Schaden nehmen, wenn sowohl USB, als auch DCC gleichzeitig angeschlossen sind.

Der Arduino wird mit einem USB-Kabel mit dem PC verbunden. Daraufhin sollte der Arduino erkannt und ein virtueller COM-Port zugewiesen werden. Wenn dies nicht erfolgt, kann man den nötigen Treiber leicht nachinstallieren. Bei google einfach nach „CH341SER.ZIP“ fragen

Nachdem ein COM-Port zugewiesen wurde kann man den Arduino über ein beliebiges Terminalprogramm (z.B. HTerm oder auch die Arduino DIE) ansprechen. Als Baudrate ist 115200 zu wählen.

Konfiguration über USB

Der Dekoder verfügt über 4 Ansteuerungsbereiche, die mit A, B, C und D gekennzeichnet sind. Prinzipiell ist gedacht, dass an jedem Bereich entweder:

- 1 Hauptsignal und ein Vorsignal
- 1 Ampelkreuzung (also sowohl die Haupt- als auch die Nebenstrasse inkl. Fussgängerampel)
- 1 Lauflicht mit bis zu 12 LEDs
- 1 Hausbeleuchtung mit bis zu 12LEDs

angeschlossen werden. Da es 4 Anschlussbereiche gibt, kann also alles 4x angeschlossen werden, oder auch jedem Bereich ein eigenes Ansteuerungsmuster zu Grunde gelegt werden. Für jedes Ansteuerungsmuster können dann Variablen wie Frequenz, Blenddauer, Dunkelzeit, usw. einzeln konfiguriert werden. Man kann aber auch nur ein Ansteuerungsmuster konfigurieren und das dann allen Anschlussbereichen zuordnen.

Folgende Kommandos sind möglich:

Achtung die Zeichen < und > gehören dazu!

Kommando	Bedeutung
<x:n>	Den Anschlussbereich x (A-D) wird ein Ansteuerungsmuster n zugeordnet, wobei n zwischen 0 und 3 sein muss.
<n:Tm>	Das Ansteuerungsmuster (Signaltyp) wird festgelegt, wobei für m gilt - 0: Multiplexsignal - 1: Ampel - 2: Lauflicht - 3: Hauslicht
<x:y>	Dem Hauptsignal x wird das Vorsignal y zugeordnet, d.h. Stellbefehle die x betreffen, werden auf y umgesetzt.
<x:Hp0>	Stellt das Signalbild Hp0 auf dem Signal x da (wenn x als Signal konfiguriert wurde). Möglich sind: Hp0, Hp1, Hp2, Sh1, Vr0, Vr1, Vr2, Aus

<table>
<tr><td><x:z></td><td>Schaltet die Ampel-/Lauflicht- oder Haussimulation auf dem Anschluss x (wenn x als Ampel, Lauflicht oder Hauslicht konfiguriert wurde).
<table>
<tr><th>z=</th><th>Ampel</th><th>Lauflicht</th><th>Hauslicht</th></tr>
<tr><td>+</td><td>ein</td><td>ein</td><td>ein</td></tr>
<tr><td>-</td><td>aus</td><td>aus</td><td>aus</td></tr>
<tr><td>o</td><td>Blinken Hauptstrasse</td><td>ein</td><td>Ein mit 1LED</td></tr>
<tr><td>O</td><td>Blinken Nebenstrasse</td><td>ein</td><td>Ein mit 2LED</td></tr>
<tr><td>/</td><td>Grün Hauptstarsse</td><td>ein</td><td>Ein mit 3LED</td></tr>
<tr><td>|</td><td>Grün Nebenstrasse</td><td>ein</td><td>Ein mit 4LED</td></tr>
<tr><td>#</td><td>Eine Stellung weiter</td><td>ein</td><td>Ein mit 5LED</td></tr>
<tr><td>*</td><td>n.a.</td><td>ein</td><td>Ein mit 6LED</td></tr>
</table></td></tr>
<tr><td><n:Lm></td><td>Im Signaltyp n wird die Signallogik konfiguriert, wobei für m gilt:
- 0: Signallogik aus
- 1: Signallogik ein, d.h. Vorsignal am gleichen Mast (d.h. am gleichen Anschluss) kann keine Befehle anzeigen, wenn das Hauptsignal die Fahrt verbietet</td></tr>
<tr><td><n:xxx=yyy></td><td>Im Signaltyp n wird für den Befehl xxx festgelegt, dass die LEDs yyy geschaltet werden.
xxx kann sein: Hp0, Hp1, Hp2, Sh1, Vr0, Vr1, Vr2
yyy ist eine Dezimalzahl, die sich aus dem binären Wert der zu schaltenden LEDs errechnet. LED-Nr. 0 steht für das 0.Bit, LED-Nr.11 für das 11.Bit. Sollen z.B. die LEDs 11 und 9 bei Hp0 des Ansteuerungsmusters (Signaltyps) 2 geschaltet werden, so muss man eingeben <2:Hp0=2560></td></tr>
<tr><td><n:xxx(yyy></td><td>Im Signaltyp n wird für den Befehl xxx festgelegt, dass die LEDs yyy in blinken sollen (und der Gruppe 1 angehören).</td></tr>
<tr><td><n:xxx)yyy></td><td>Im Signaltyp n wird für den Befehl xxx festgelegt, dass die LEDs yyy in blinken sollen (und der Gruppe 2 angehören).
Blinkgruppe 1 und 2 blinken genau gegengleich!</td></tr>
<tr><td><n:Dzm></td><td>Im Signaltyp n wird als Dunkelzeit (zwischen einzelnen Stellbefehlen) m Millisekunden gewählt</td></tr>
<tr><td><n:Ftm></td><td>Im Signaltyp n wird als Blendzeit (Geschwindigkeit für Auf-/Abblenden) m Millisekunden gewählt</td></tr>
<tr><td><n:LedMaxm></td><td>Für Lauf-/Hauslicht: maximal m LEDs sind angeschlossen</td></tr>
<tr><td><n:LedAnm></td><td>Für Lauf-/Hauslicht: maximal m LEDs sollen gleichzeitig mit voller Helligkeit leuchten</td></tr>
<tr><td><Zn:m></td><td>Für Ampelschaltung: Lässt die Ampelstellung n (1-10) für m Millisekunden stehen, bevor die Ampel eine Stellung weiter schaltet.</td></tr>
<tr><td><+Xn:m></td><td>Stellt die Helligkeit für LED-No n im Anschlussbereich X auf m% ein.</td></tr>
<tr><td><-Xn></td><td>Schaltet die LED-No n im Anschlussbereich X aus.</td></tr>
<tr><td><BXn:m></td><td>Lässt die LED-No n im Anschlussbereich X mit einer Frequenz von m blinken.</td></tr>
<tr><td><DCCDzm></td><td>Ignoriert doppelte DCC-Befehle innerhalb von m Millisekunden</td></tr>
<tr><td><DCCn:m></td><td>Ordnet die DCC-Adresse n dem m.Stellbefehl zu. Der Anschlussbereich</td></tr>
</table>

	an hört auf die Stellbefehle 1-8, B auf 9-17, …
<An>	Konfiguriert, ob der letzte Zustand der Signalbilder bei wiedereinschalten des Stromes wieder dargestellt werden so.. 1: ja 0: nein
<Reset>	Setzt alle Einstellungen zurück
<I>	Zeigt die gegenwärtigen Einstellungen
<T>	Testes alle angeschlossenen LEDs

Beispiele

Kommando	**Bedeutung**
<0:T0>	Ansteuerungsmuster 0 ist Multiplexampel (T0)
<2:T1>	Ansteuerungsmuster 2 ist Ampel (T1)
<1:T2>	Ansteuerungsmuster 1 ist Lauflicht (T2)
<A:0>	Anschlussbereich A ist Ansteuerungsmuster 0 zugewiesen
<B:0>	Anschlussbereich B ist Ansteuerungsmuster 0 zugewiesen
<C:1>	Anschlussbereich C ist Ansteuerungsmuster 1 zugewiesen
<D:2>	Anschlussbereich C ist Ansteuerungsmuster 2 zugewiesen
<A:B>	Das Vorsignal von B wird vom Hauptsignal A beeinflusst
<0:Hp0=10>	Die Leds 1 und 3 im Ansteuerungsmuster 0 werden beim Befehl Hp0 geschaltet, da Dezimal 10 = Binär 1010
<0:Hp2=2560>	Die Leds 12 und 10 Ansteuerungsmuster 0 werden beim Befehl Hp2 geschaltet, da Dezimal 2560 = Binär 101000000000
<0:Dz300>	Für das Ansteuerungsmuster 0 werden 300ms als Dunkelzeit zwischen auf und abblenden eingestellt.
<0:Ft25>	Für das Ansteuerungsmuster 0 werden 25ms als Blendzeit eingestellt.
<+A0:80>	Die LED-No. 0 im Anschlussbereich A wird auf 80% Helligkeit gestellt.
<+D11:20>	Die LED-No. 11 0 im Anschlussbereich D wird auf 20% Helligkeit gestellt.
<Z10:5000>	In der Ampelschaltung bleibt die Stellung 10 für 5000ms bestehen
<1:Hp0=16>	Im Anschlussmuster 1 ordnen wir dem Befehl Hp0 die LED-No. 5 zu, da 16 (dezimal) =10000 (binär)
<2:LedMax9>	Ein im Ansteuerungsmuster 2 angeschlossenes Lauflicht hat 9 LEDs
<2:LedAn4>	Bei einem im Ansteuerungsmuster 2 angeschlossenen Lauflicht sollen 4 LEDs leuchten.
<2:B10>	Die Geschwindigkeit für das Lauflicht wird auf 10 eingestellt.
<2:Dz1000>	Wenn ein Durchgang des Lauflichtes fertig ist, wird 1000 ms gewartet, bis der nächste beginnt
<A:Hp1>	Für Signale im Anschlussbereich A wird Hp1 gezeigt
<C:Vr2>	Für Signale im Anschlussbereich C wird Vr2 gezeigt
<D:+>	Ampeln im Ansteuerungsbereich D werden gestartet