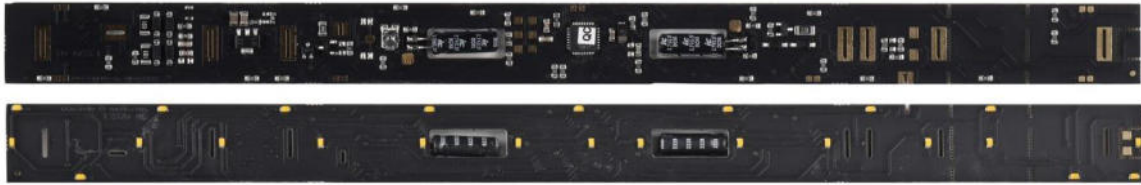


Bedienungsanleitung



Inhaltsverzeichnis

1. Einführung	Seite 2
2. Funktionsausgangszuordnung	
2.1 Funktionsausgänge	Seite 3
2.2 Funktionstastenzuordnung	Seite 4
3. Einbau	Seite 4
3.1 Steuerwagen Bybdzf 482, Bauart Halberstadt	Seite 5
3.2 Steuerwagen Bnrdzf 483, Bauart Silberling	Seite 6
3.3 Steuerwagen BDnf 738	Seite 7
3.4 Steuerwagen BDnf 740	Seite 8
3.5 Reisezugwagen Halberstädter Mitteleinstiegsw.	Seite 9
3.6 Reisezugwagen Halberstädter Seitengangw.	Seite 10
3.7 Reisezugwagen Bauart Silberling	Seite 11
4. Standard CV – Programmierung	
4.1 Adresse programmieren	Seite 12
4.2 Zweitadresse (Decodersperre)	Seite 13
4.3 Decoder-Reset	Seite 13
4.4 Funktionsmapping	Seite 14
4.5 Analog - Modus	Seite 15
4.6 Onboard Pufferspeicher (SPP)	Seite 16
4.8 Anwendererkennung	Seite 16
4.9 Andere Funktionen	Seite 16
5. Effekte für Funktionsausgänge	
5.1 Lichtintensität	Seite 17
5.2 Lichteffekte	Seite 17
5.2.1 Ein- und Ausblenden	Seite 17
5.2.2 Neonröhren - Effekt	Seite 17
5.2.3 Flackereffekt	Seite 18
5.2.4 Defekte Neonröhre - Effekt	Seite 18
5.2.5 Ein- und Ausschaltverzögerung	Seite 18
5.2.6 Zufälliges Ein- und Ausschalten	Seite 18
6. CV Tabelle	Seite 19

1. Einführung

Herzlichen Glückwunsch zum Erwerb eines TILLIG – Qualitätsproduktes. Wir möchten Ihnen hier alle nötigen Informationen an die Hand geben, um Ihre Innenbeleuchtung zu verbauen und nach Ihren Wünschen anzupassen

Dieser Typ Innenbeleuchtung kann in folgenden Wagen verbaut werden:

- Steuerwagen Bybdzf 482, Bauart Halberstadt



- Steuerwagen Bnrdf 483, Bauart Silberling



- Steuerwagen BDnf 738



- Steuerwagen BDnf 740



- Reisezugwagen "Halberstädter Mitteleinstiegswagen" (neue Konstruktion)



- Reisezugwagen "Halberstädter Seitengangwagen"



- Reisezugwagen "Silberling"



2. Funktionsausgangszuordnung

Der integrierte Decoder hat 16 Funktionsausgänge, so dass LED's komplett einzeln oder in einzelnen Gruppen geschaltet werden können.

2.1 Funktionsausgänge

Auf der folgenden Zeichnung können Sie sehen, wie die LED's angeschlossen sind.

Für diesen speziellen Typ Innenbeleuchtung nutzen wir Trennstellen, so dass die Innenbeleuchtung an die Wagen- und Inneneinrichtungsart angepasst werden kann. Dafür trennen Sie die Verbindung zwischen den zwei Lötspots einer Trennstelle mit einem scharfen Messer. Falls Sie die falsche Verbindung getrennt haben, können Sie sie durch Löten wieder verbinden.

Einzelne LED's:

AUX1=LED1; AUX2=LED2; AUX3=LED3; AUX4=LED4; AUX5=LED5; AUX6=LED6; AUX7=LED7;
 AUX8=LED8; AUX9=LED9; AUX10=LED10; AUX11=LED11; AUX12=LED12; AUX13=LED13

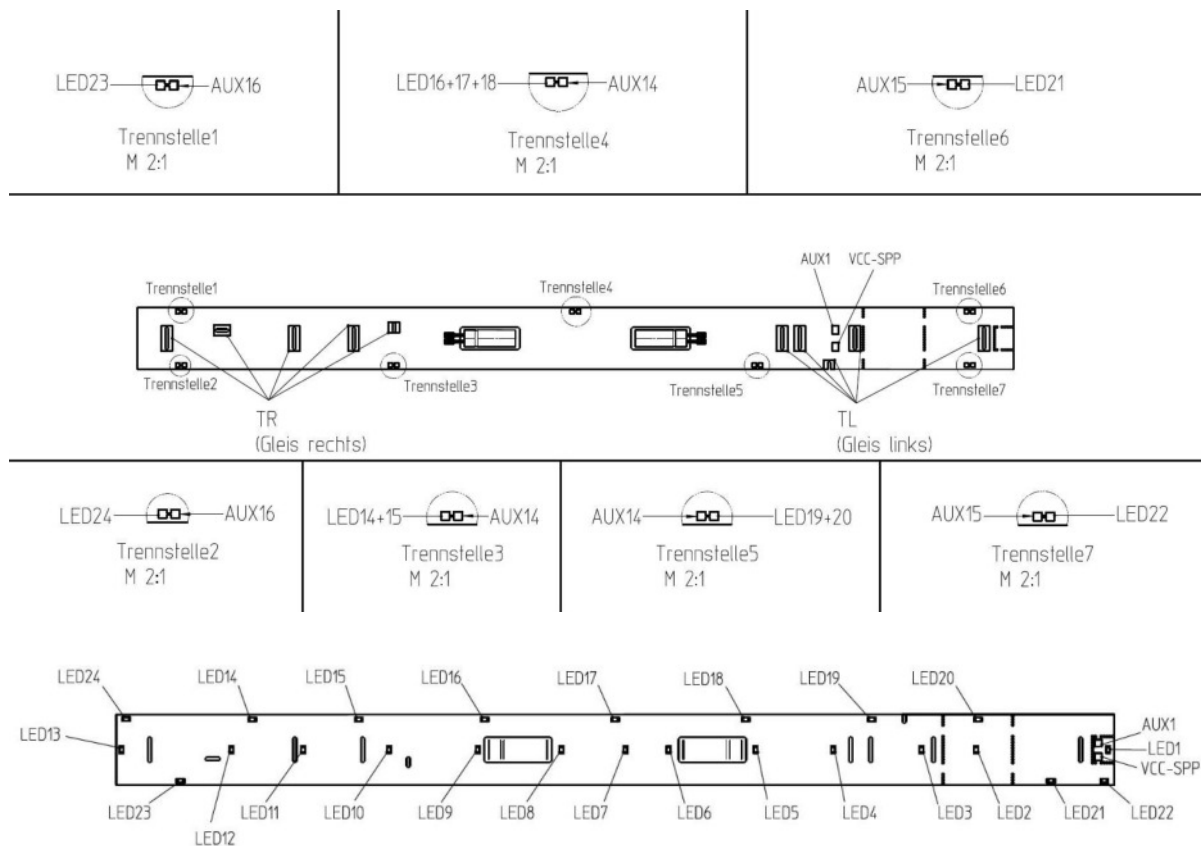
Gruppen:

AUX14= Seitengang = (Gruppe 1=LED14+15)+(Gruppe 2=LED16+17+18)+(Gruppe 3=LED19+20)

AUX15= Toilette vorn = (LED21)+(LED22)

AUX16= Toilette hinten = (LED23)+(LED24)


In Klammern stehende LED's, können durch die unten dargestellten Trennstellen separat vom Funktionsausgang getrennt werden.



2.2 Funktionstastenzuordnung

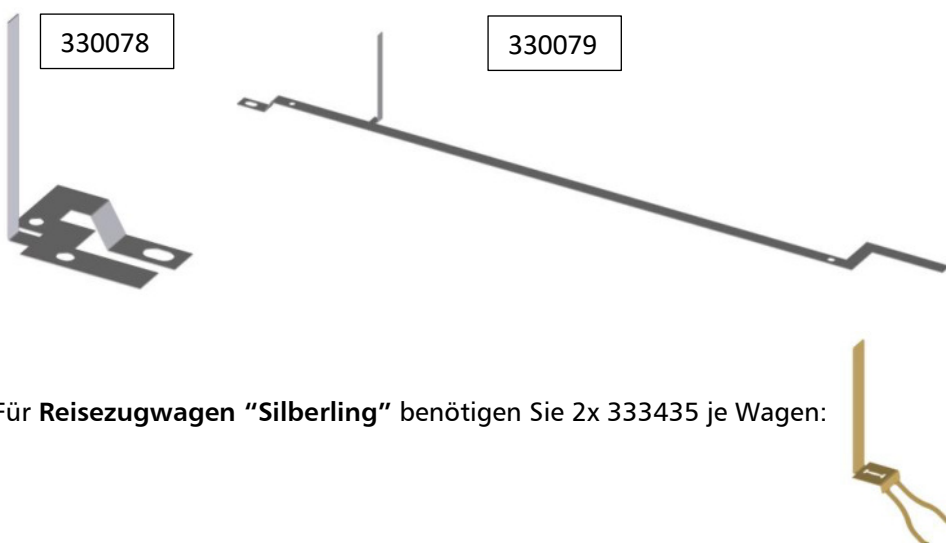
F0	Alle Funktionsausgänge an
F1	AUX1
F2	AUX2
F3	AUX3
F4	AUX4
F5	AUX5
F6	AUX6
F7	AUX7
F8	AUX8
F9	AUX9
F10	AUX10
F11	AUX11
F12	AUX12
F13	AUX13
F14	AUX14 (Seitengang)
F15	AUX15 (Toilette vorn)
F16	AUX16 (Toilette hinten)

3. Einbau

Damit der Wagen für den Einbau einer Beleuchtung vorbereitet ist, achten Sie bitte auf dieses Symbol: 

Ältere Wagen, welche nicht für den Einbau vorbereitet sind, können ebenfalls verwendet werden, allerdings müssen Sie dort die entsprechenden Kontaktbleche nachrüsten/umbauen oder Sie verwenden Litzen. **-WARNUNG-**: Hierzu ist die Demontage des Wagens notwendig. Dabei kann es zum Verlust oder Beschädigung von Teilen kommen.

Für Steuerwagen BDNf 738 und BDNf 740 benötigen Sie folgende Artikel (1x pro Wagen):



Für Reisezugwagen "Silberling" benötigen Sie 2x 333435 je Wagen:

Hinweis an Digitalbahner: Sollten Sie einen Steuerwagen umbauen, müssen Sie vor dem Einbau Ihren Funktionsdecoder sperren (siehe Kapitel 4.2 Zweitadresse). Ansonsten bekommen Sie Probleme bei der Programmierung.

3.1 Steuerwagen Bybdzf 482, Bauart Halberstadt

<p>Die Innenbeleuchtung wird oben auf der Inneneinrichtung verbaut.</p> <ol style="list-style-type: none"> 1. Nehmen Sie das Oberteil ab. Dazu dieses im Bereich der Türen leicht spreizen und nach oben abziehen. 	
<ol style="list-style-type: none"> 2. Kürzen Sie die Länge der Leiterplatte an der ersten Markierung. Nutzen Sie dafür ein scharfes Messer oder eine Trennscheibe. 	
<ol style="list-style-type: none"> 3. Trennen Sie die Trennstellen 1; 3; 4 und 5. <p>Hinweis: AUX1; AUX14 und AUX15 werden nun nicht mehr benutzt. AUX1 kann als freier Funktionsausgang verwendet werden. Außerdem kann die LED aus dem Reststück der Leiterplatte ausgeschnitten und per Litzen angelötet werden.</p>	
<ol style="list-style-type: none"> 4. Löten Sie die Leiterplatte entsprechend der Markierungen an den Kontaktblechen an. 	
<ol style="list-style-type: none"> 5. Setzen Sie das Oberteil wieder auf. 	

3.2 Steuerwagen Bnrdfz 483, Bauart Silberling

<p>Die Innenbeleuchtung wird oben auf der Inneneinrichtung verbaut.</p> <ol style="list-style-type: none"> 1. Nehmen Sie das Oberteil ab. Dazu dieses im Bereich der Türen leicht spreizen und nach oben abziehen. 	
<ol style="list-style-type: none"> 2. Kürzen Sie die Länge der Leiterplatte an der ersten Markierung. Nutzen Sie dafür ein scharfes Messer oder eine Trennscheibe. 	
<ol style="list-style-type: none"> 3. Trennen Sie die Trennstellen 1; 2; 3; 4 und 5 <p>Hinweis: AUX1; AUX14; AUX15 und AUX16 werden nun nicht mehr benutzt. AUX1 kann als freier Funktionsausgang verwendet werden. Außerdem kann die LED aus dem Reststück der Leiterplatte ausgeschnitten und per Litzen angelötet werden.</p>	
<ol style="list-style-type: none"> 4. Löten Sie die Leiterplatte entsprechend der Markierungen an den Kontaktblechen an. 	
<ol style="list-style-type: none"> 5. Setzen Sie das Oberteil wieder auf. 	

3.3 Steuerwagen BDNf 738

<p>Die Innenbeleuchtung wird auf dem Oberteil verbaut.</p> <ol style="list-style-type: none">1. Nehmen Sie das Dach ab. Dazu schieben Sie es nach vorn und heben es ab.	
<ol style="list-style-type: none">2. Kürzen Sie die Länge der Leiterplatte an der ersten Markierung. Nutzen Sie dafür ein scharfes Messer oder eine Trennscheibe.	
<ol style="list-style-type: none">3. Trennen Sie die Trennstellen 1; 3; 4 und 5. <p>Hinweis: AUX1; AUX14 und AUX15 werden nun nicht mehr benutzt. AUX1 kann als freier Funktionsausgang verwendet werden. Außerdem kann die LED aus dem Reststück der Leiterplatte ausgeschnitten und per Litzen angelötet werden.</p>	
<ol style="list-style-type: none">4. Löten Sie die Leiterplatte entsprechend der Markierungen an den Kontaktblechen an.	
<ol style="list-style-type: none">5. Setzen Sie das Dach wieder auf.	

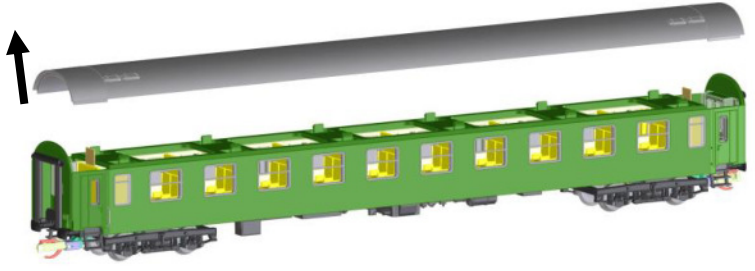


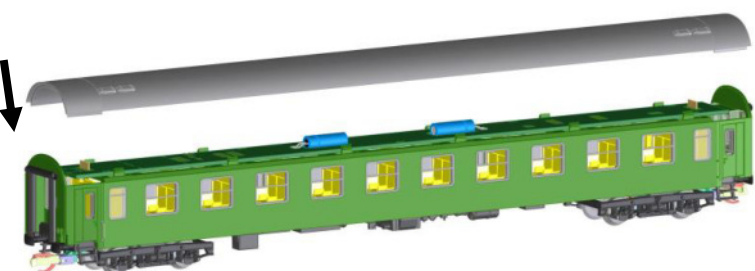
3.4 Steuerwagen BDnf 740

<p>Die Innenbeleuchtung wird auf dem Oberteil verbaut.</p> <ol style="list-style-type: none"> 1. Nehmen Sie das Dach ab. Dazu schieben Sie es nach vorn und heben es ab. 	
<ol style="list-style-type: none"> 2. Kürzen Sie die Länge der Leiterplatte an der ersten Markierung. Nutzen Sie dafür ein scharfes Messer oder eine Trennscheibe. 	
<ol style="list-style-type: none"> 3. Trennen Sie die Trennstellen 1; 3; 4 und 5. <p>Hinweis: AUX1; AUX14 und AUX15 werden nun nicht mehr benutzt. AUX1 kann als freier Funktionsausgang verwendet werden. Außerdem kann die LED aus dem Reststück der Leiterplatte ausgeschnitten und per Litzen angelötet werden.</p>	
<ol style="list-style-type: none"> 4. Löten Sie die Leiterplatte entsprechend der Markierungen an den Kontaktblechen an. 	
<ol style="list-style-type: none"> 5. Setzen Sie das Dach wieder auf. 	

3.5 Reisezugwagen "Halberstädter Mitteleinstiegswagen" (neue Konstruktion)

<p>Die Innenbeleuchtung wird oben auf der Inneneinrichtung verbaut.</p> <p>1. Nehmen Sie das Oberteil ab. Dazu dieses im Bereich der Türen leicht spreizen und nach oben abziehen.</p>	
<p>2. Für Wagen der Gattung ABy (siehe Inneneinrichtung) trennen Sie die Trennstellen 1; 3; 5 und 6</p> <p>Bei allen anderen trennen Sie die Trennstellen 1; 3; 4; 5; und 6.</p> <p>Hinweis: AUX14 wird nicht mehr benutzt. (Außer ABy)</p>	
<p>3. Löten Sie die Leiterplatte entsprechend der Markierungen an den Kontaktblechen an.</p>	
<p>4. Setzen Sie das Oberteil wieder auf.</p>	

3.6 Reisezugwagen "Halberstädter Seitengangwagen"

<p>Die Innenbeleuchtung wird auf dem Oberteil verbaut.</p> <ol style="list-style-type: none">1. Nehmen Sie das Dach ab. Dazu müssen Sie die Rastungen lösen, indem Sie ein dünnes Stück Kunststoff zwischen Dach und Oberteil schieben.	
<ol style="list-style-type: none">2. Trennen Sie die Trennstellen 2 und 7	
<ol style="list-style-type: none">3. Löten Sie die Leiterplatte entsprechend der Markierungen an den Kontaktblechen an.	
<ol style="list-style-type: none">4. Setzen Sie das Dach wieder auf.	

3.7 Reisezugwagen Bauart "Silberling"

<p>Die Innenbeleuchtung wird auf dem Oberteil verbaut.</p> <p>1. Nehmen Sie das Dach ab. Dazu ziehen Sie die Gummiwulst vorn ab, schieben das Dach nach vorn und heben es ab.</p>	
<p>2. Für Wagen der Gattung ABy (siehe Inneneinrichtung) trennen Sie die Trennstellen 1; 3; 5 und 6</p> <p>Bei allen anderen trennen Sie die Trennstellen 1; 3; 4; 5; und 6.</p> <p>Hinweis: AUX14 wird nicht mehr benutzt. (Außer ABn)</p>	<p style="text-align: center;">ABn</p> <p style="text-align: center;">Alle anderen</p>
<p>3. Löten Sie die Leiterplatte entsprechend der Markierungen an den Kontaktblechen an.</p>	
<p>4. Setzen Sie das Dach wieder auf und stecken die Gummiwulst wieder an.</p>	

4. Standard CV – Programmierung

-WARNUNG-: Gehen Sie während der Programmierung sicher, dass Sie nur die Innenbeleuchtung programmieren. Sollten Sie einen Steuerwagen umbauen, müssen Sie vor dem Einbau Ihren Funktionsdecoder sperren (siehe Kapitel 4.2 Zweitadresse). Ansonsten können Probleme bei der Programmierung auftreten.

Sie können CV's auf dem Programmiergleis (PT) oder per PoM programmieren. Das Lesen der CV's funktioniert nur auf dem Programmiergleis (nicht mit PoM).

4.1 Adresse programmieren

Der On-Board Funktionsdecoder der Innenbeleuchtung kann sowohl auf kurze (1-127), als auch auf lange Adresse (1-9999) programmiert werden. Im Auslieferungszustand ist die Innenbeleuchtung auf die kurze Adresse (CV29; bit5=0) 3 programmiert (CV1=3).

Die Adresse kann auf dem Programmiergleis (PT) oder per PoM programmiert werden, indem Sie in die CV1 die gewünschte Adresse schreiben.

Falls die lange Adresse benötigt wird, muss der Adress-Modus umgestellt werden. Dies geschieht im bit5 der CV29. Ändern Sie also den bit-Wert für den bit5 von 0 auf 1 oder addieren Sie 32 zum ausgelesenen Wert um die Lange Adresse zu verwenden. Der Decoder reagiert nun auf die in CV17 und CV18 gespeicherte Adresse.

Die lange Adresse wird nach dem folgenden Algorithmus berechnet (in unserem Beispiel Adresse 2000):

- Dividieren Sie Ihre gewählte Adresse durch 256
(Für unser Beispiel: $2000 / 256 = 7,8125 = 7$; Restwert = $0,8125 \times 256 = 208$)
- Addieren Sie 192 zum Ergebnis und schreiben den Wert in CV17
($7 + 192 = 199$ in CV17 schreiben)
- Schreiben Sie den Restwert in CV18
(208 in CV18 schreiben)

Nach dem Programmieren von CV29, CV17 und CV18 auf die gezeigten Werte, ist dem Decoder die Adresse 2000 zugeordnet.

Um auf die kurze Adresse zurück zu stellen, muss nur der bit5 der CV29 wieder Null – gesetzt werden.

4.2 Zweitadresse (Decodersperre)

Die Programmierung der Zweitadresse ist nötig für die Verwendung der Innenbeleuchtung in Steuerwagen mit Funktionsdecodern.

Wenn Sie mehrere Decoder innerhalb desselben Modells verwenden, ist es sinnvoll, eine sekundäre Adresse zu verwenden, die die Auswahl des betreffenden Decoders ermöglicht. Somit kann jeder Decoder separat programmiert werden, ohne ihn ausbauen zu müssen. Die Zweitadresse des jeweiligen Decoders muss programmiert werden, bevor ein zweiter eingebaut wird. Die Zweitadresse kann von 1-7 vergeben werden (0 bedeutet, dass keine Zweitadresse benutzt wird). Es können also bis zu 7 verschiedene Decoder in einem Modell verbaut werden.

Wenn der Wert von CV16 ungleich Null ist, akzeptieren die Decoder Programmierbefehle nur, wenn die Zweitadresse des Decoders, der programmiert werden soll, zuvor in CV15 programmiert wurde und mit dem Wert in CV16 übereinstimmt (sie sollte mit CV16 des betreffenden Decoders identisch sein).

Für die Verwendung der Zweitadresse ist es wichtig zu wissen, dass die einzige CV, welche gelesen und geschrieben werden kann, ohne die Zweitadresse zu kennen, CV15 ist. Aus diesem Grund sind die zur Verfügung stehenden Adressen auf den Bereich 1-7 beschränkt. Wenn die Sekundäradresse des Decoders vergessen wird, kann sie durch Testen wiedergefunden werden.

Es wird also immer nur der Decoder programmiert, dessen Zweitadresse dem in CV15 geschriebenen Wert entspricht. Kontrollieren Sie diesen Wert als erstes, bevor Sie mit dem Programmieren anderes CV's beginnen!

Als Beispiel: Der Steuerwagen Bybdzf 482, Bauart Halberstadt soll mit der Innenbeleuchtung ausgestattet werden. Dieses Modell hat einen integrierten Funktionsdecoder.

Das bedeutet, **bevor** Sie die Innenbeleuchtung verbauen, programmieren Sie die die Zweitadresse auf 1 (CV16=1).

Nun können Sie die Innenbeleuchtung einbauen. Diese soll die Zweitadresse 2 bekommen. Dazu muss als erstes die CV15=0 (Standartauswahl) geschrieben werden. Anschließend wird in die CV16=2 geschrieben. Somit hat die Innenbeleuchtung nun die gewünschte Zweitadresse.

Beide Decoder reagieren im Betrieb nun nach wie vor auf die kurze (primäre) Adresse 3, welche von Haus aus eingestellt ist, können aber durch die Auswahl in CV15 getrennt voneinander programmiert werden.

Das bedeutet, wenn etwas am OnBoard-Decoders des unprogrammiert werden soll, muss als erstes seine Adresse in die CV15 geschrieben werden (CV15=1). Somit wird die Zweitadresse 1 ausgewählt und alle anderen Zweitadressen sind für die Programmierung gesperrt.

Soll etwas an der Innenbeleuchtung unprogrammiert werden, wird CV15=2 geschrieben.

4.3 Decoder-Reset

Die werkseitig voreingestellten CV-Werte sind in der Spalte "Standardwert" der CV-Tabelle angegeben. Der Decoder kann jederzeit durch einen Reset auf die Standardwerte zurückgesetzt werden. Programmieren Sie einen beliebigen Wert auf CV8, um einen Reset auf die Werkseinstellungen durchzuführen.

4.4 Funktionsmapping

Jede Funktionstaste (F0 bis F16) kann dazu benutzt werden, um einen oder mehrere der 16 Funktionsausgänge (AUX) des OnBoard Decoders zu schalten. Diese Abhängigkeiten zwischen Funktionstaste und Funktionsausgang nennt man Funktionsmapping. Das Programmieren dieses Mappings erfolgt wie unten in der Tabelle dargestellt ist.

Die Funktionstasten **F0** (als Lichtfunktion bezeichnet) und **F1** werden fahrtrichtungsabhängig definiert. Es gibt also 2 CV's pro Fahrtrichtung, entspricht 4 CV's pro Funktionstaste. Die restlichen Funktionstasten (F2-F16) werden nicht der Fahrtrichtung zugeordnet, so dass lediglich 2 CV's pro Funktionstaste zu mappen sind, um sie einem Funktionsausgang zuzuordnen

AUX		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
bit		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
F-Taste	CV																
F0	33									128	64	32	16	8	4	2	1
Vorw.	34	128	64	32	16	8	4	2	1								
F0	35									128	64	32	16	8	4	2	1
Rückw.	36	128	64	32	16	8	4	2	1								
F1	37									128	64	32	16	8	4	2	1
Vorw.	38	128	64	32	16	8	4	2	1								
F1	39									128	64	32	16	8	4	2	1
Rückw.	40	128	64	32	16	8	4	2	1								
F2	41									128	64	32	16	8	4	2	1
	42	128	64	32	16	8	4	2	1								
F3	43									128	64	32	16	8	4	2	1
	44	128	64	32	16	8	4	2	1								
F4	45									128	64	32	16	8	4	2	1
	46	128	64	32	16	8	4	2	1								
F5	47									128	64	32	16	8	4	2	1
	48	128	64	32	16	8	4	2	1								
F6	49									128	64	32	16	8	4	2	1
	50	128	64	32	16	8	4	2	1								
F7	51									128	64	32	16	8	4	2	1
	52	128	64	32	16	8	4	2	1								
F8	53									128	64	32	16	8	4	2	1
	54	128	64	32	16	8	4	2	1								
F9	55									128	64	32	16	8	4	2	1
	56	128	64	32	16	8	4	2	1								
F10	57									128	64	32	16	8	4	2	1
	58	128	64	32	16	8	4	2	1								
F11	59									128	64	32	16	8	4	2	1
	60	128	64	32	16	8	4	2	1								
F12	61									128	64	32	16	8	4	2	1
	62	128	64	32	16	8	4	2	1								
F13	160									128	64	32	16	8	4	2	1
	161	128	64	32	16	8	4	2	1								
F14	162									128	64	32	16	8	4	2	1
	163	128	64	32	16	8	4	2	1								
F15	164									128	64	32	16	8	4	2	1
	165	128	64	32	16	8	4	2	1								
F16	166									128	64	32	16	8	4	2	1
	167	128	64	32	16	8	4	2	1								

Als Beispiel:

Sie möchten mit F2 AUX4 aktivieren: F2 wird über die CV41 (für AUX1-8) und CV42 (für AUX9-16) definiert

Bit3 wird verwendet um AUX4 zu aktivieren, es muss also der Wert 8 in CV41 und der Wert 0 in CV42 programmiert werden.

Soll nun F2 dazu benutzt werden um AUX3 und AUX4 zu aktivieren, muss zusätzlich zum Bit3 auch der Bit4 aktiviert werden. Es muss also in CV41 der Wert 4 addiert werden (CV41=4+8=12).

Soll mit F2 außerdem AUX13 und 14 aktiviert werden, müssen Bit4 und Bit5 in CV42 aktiviert werden, CV42 muss also auf 48 programmiert werden (CV42=16+32=48).

4.5 Analog - Modus

Beim Einschalten prüft der Decoder, ob ein DCC-Signal auf dem Gleis vorhanden ist, und führt die empfangenen Befehle aus. Liegt nun Gleichspannung länger als die definierte Zeitüberschreitung auf dem Gleis an, schaltet der Decoder in den Analog - Modus um und schaltet die in CV13 und CV14 konfigurierten Funktionen ein. Die Zeitüberschreitung ist in CV11 definiert und entspricht dem geschriebenen Wert mit 8 multipliziert in Millisekunden. Der Maximalwert liegt bei 2,048 s.

Die verwendeten Modi werden in CV12 und CV29 festgelegt

Die Bedeutungen der Bits von CV12 werden in dieser Tabelle dargestellt.

CV12 Bit	Wert	Betriebsart
0	0	DC Modus AUS
	1	DC Modus AN
2	0	DCC Modus AUS
	4	DCC Modus AN

Wenn in der CV29 der Bit2=0 gesetzt ist schaltet der Funktionsdekor nicht in den analogen Modus um, wenn die digitale Kommunikation ausgeschaltet ist. Aus Sicherheitsgründen kann CV12 auch dann geändert werden, wenn Bit 0 von CV12 auf 0 gesetzt ist (DCC-Modus = AUS). Die DCC-Programmierbefehle werden auch dann ausgeführt, wenn der DCC-Digitalmodus ausgeschaltet ist.

Standardmäßig sind die DC-Analog- und DCC-Operationen aktiviert.

Standardmäßig ist F0 für beide Fahrrichtungen konfiguriert (Bit 2 in CV29 und CV14 aktiviert für die Verwendung der F0-Funktion in beide Richtungen. CV14 = 1 + 2 = 3). Wenn die Platine an eine DC-Analogspannung angeschlossen ist, werden alle Ausgänge des Funktionsdecoders eingeschaltet.

Um andere Funktionen im analogen Gleichspannungsbetrieb einzuschalten, müssen diese in CV13 und CV14 definiert werden. Das Mapping der Funktion muss vorab gemäß der Tabelle unterhalb definiert werden. Es können nur die Funktionen F0 und F1-F14 im DC-Betrieb verwendet werden.

	F0 vorw.	F0 rückw.	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
Bit	0	1	0	1	2	3	4	5	6	7	2	3	4	5	6	7
CV13			1	2	4	8	16	32	64	128						
CV14	1	2									4	8	16	32	64	128

4.6 OnBoard Pufferspeicher (SPP)

Der OnBoard Pufferspeicher (SPP) bietet bis zu 20 Sekunden Pufferung. Es benötigt ca. 30-40 Sekunden für eine vollständige Aufladung, aber nach 20 Sekunden Stromanschluss ist es bereits nutzbar.

Die Ausschaltzeit des SPP nach Verlust der Gleisspannung in CV168 eingestellt werden. Der werkseitige Standardwert ist 62. Eine Einheit entspricht 80 Millisekunden. Dies bedeutet, dass die werkseitige Standard-Ausschaltzeit $62 * 0,080 = 4,96 \sim 5$ Sekunden beträgt. Nachdem die Elektronik von der Schiene getrennt wurde, leuchten die Lichter 5 Sekunden lang nach. Nach dieser Zeit werden sie automatisch ausgeschaltet.

Der Maximalwert für CV168 liegt bei 255, was in etwa 20 Sekunden entspricht. Das bedeutet, dass die Lichter 20 Sekunden nach dem Entfernen der Gleisspannung weiter leuchten. Dies funktioniert auch mit analogem Gleichstrom. Der im CV168 eingestellte Wert gilt also auch für den analogen DC-Betrieb.

4.7 Anwendererkennung

CV105 und CV106 sind zwei CV's, die zum Speichern von Benutzerkennungen (Seriennummer usw.) verwendet werden können. Die Besonderheit dieser beiden CV's besteht darin, dass ihr Inhalt nach einem Reset nicht gelöscht wird.

4.8 Andere Funktionen

Letzten Befehl speichern

Der On-Board-Decoder hat eine Funktion implementiert, um den zuletzt empfangenen Funktionsbefehl zu speichern. Diese Funktion kann aktiviert werden, indem der Wert 1 in CV152 programmiert wird. Wenn diese Funktion aktiviert ist, führt der Decoder die Funktionen aus, die vor der Stromunterbrechung aktiv waren, auch wenn keine DCC-Befehle empfangen wurden, um diese Funktionen zu aktivieren.

DCC Signalqualität

Der DCC-Signalqualitätsindikator (QoS = Quality of Signal) wird in CV108 in Prozent (im Bereich von 0-100%) gespeichert. Der niedrigste QoS-Wert, der vom Decoder aus dem letzten Messwert erkannt wurde, wird in CV109 geschrieben.

In CV169 wird die Funktionstaste definiert, deren Aktivierung das Speichern der aktuellen QoS-Werte im nichtflüchtigen Decoder Speicher (EEPROM) auslöst. Das Speichern der Momentan Werte erfolgt durch Aktivieren (Einschalten und Ausschalten) dieser Funktion über die Zentrale (oder den tOm Programmer). (z.B. muss bei der Funktion F28 der Wert 28 in CV169 eingetragen werden).

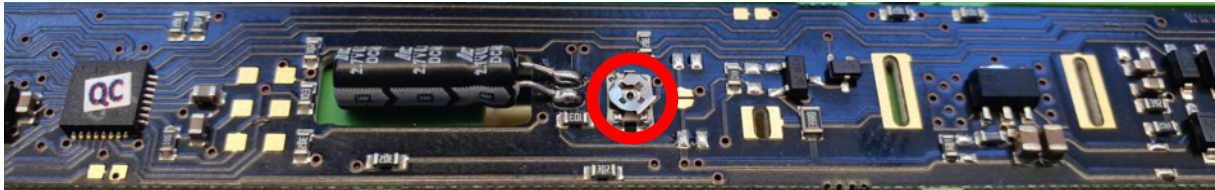
Ohne Ein- und Ausschalten der in CV169 angegebenen Funktionstaste werden die Werte in den entsprechenden CV's nicht aktualisiert!

5. Effekte für Funktionsausgänge

5.1 Lichtintensität

Die Lichtintensität der LED's, die an die Ausgänge des OnBoard-Decoders angeschlossen sind, kann individuell verändert werden, indem die Werte von CV120 bis CV135 geändert werden. Der werkseitige Standardwert für jeden Ausgang ist der Dezimalwert 255 (maximale Intensität). Der Wert von 255 in einem dieser CV's führt zu einer kontinuierlichen Ausgabe des entsprechenden Ausgangs bei maximaler Intensität.

Das Dimmer - Potentiometer arbeitet sowohl im analogen als auch im digitalen Modus als globale Lichtintensität. Im digitalen Modus kann jede Ausgangsintensität in den CV's gedimmt werden. Das Potentiometer hat die höchste Priorität und steht über den einzelnen CV-Einstellungen.



Es stehen verschiedene Effekte zur Verfügung (die mit weiteren Firmware-Upgrades erweitert werden können), deren Parameter in den CV's 112-117 definiert sind.

In CV170 wird die Frequenz des PWM-Signals eingestellt, das für alle Ausgänge verwendet wird. Der Standardwert ist 2 (500 Hz).

CV170 Wert	Output PWM Frequenz
0	125 Hz
1	250 Hz
2	500 Hz

5.2 Lichteffekte

Die Effekte können für jeden Ausgang separat in den CV's 136-152 konfiguriert. Diese CV's müssen mit folgenden Werten programmiert werden:

- 0 - Stetiger Ausgang
- 1 - Ein- und Ausblenden
- 2 - Neonröhren - Effekt
- 3 - Flackereffekt
- 4 - Defekte Neonröhre - Effekt
- 32 - Einschaltverzögerung
- 64 - Ausschaltverzögerung
- 128 - Zufälliges Ein- und Ausschalten

Standardmäßig ist das Ein- und Ausblenden aktiviert (CV136 bis CV152 sind auf Wert 1 eingestellt).

5.2.1 Ein- und Ausblenden

CV112 und CV113 definieren die Ein- und Ausblendzeit, sobald dieser Effekt verwendet wird.
Wert 1 = 8ms, 15 = 120ms, 125 = 1000ms

5.2.2 Neonröhren Effekt

In CV114 wird der Start-Effekt einer Neonröhre definiert. Es kann also definiert werden wie schnell sie „startet“, von schnell (Wert 0) zu langsam (Wert 7).

5.2.3 Flackereffekt

In CV116 wird die Flackerperiode für den flackernden Lampeneffekt angegeben. Es kann von schnellen (Wert 0) bis langsamen (Wert 7) Flackern eingestellt werden.

5.2.4 Defekte Neonröhre - Effekt

Die Wiederholungszeit der defekten Neonröhre (blinken/flackern) kann in CV117 eingestellt werden. Es kann von schnell (Wert 0) bis langsam (Wert 7) eingestellt werden.

5.2.5 Ein- und Ausschaltverzögerung

Alle Ausgänge können mit einer Verzögerung ein- und/oder ausgeschaltet werden, die in CV111 (Einschaltverzögerung) und CV110 (Ausschaltverzögerung) angegeben ist. Diese Verzögerungen sind für alle Ausgänge gleich und können in 8ms-Schritten eingestellt werden. Der maximal mögliche Verzögerungswert beträgt $8 \cdot 255 = 2040$ ms, ca. 2 Sekunden.

Das Ein- und Ausschalten für jeden der Effekte, die auf einen Ausgang angewendet werden, kann mit einer Verzögerung ausgeführt werden. Wenn Sie die Verzögerungsoption für einen bestimmten Ausgang verwenden, wird der Ausgang nach Erhalt des Aktivierungsbefehls (Funktion) mit einer in CV111 eingestellten Verzögerungszeit aktiviert und nach Erhalt des Deaktivierungsbefehls (Funktion) mit einer in CV110 eingestellten Verzögerungszeit deaktiviert.

5.2.6 Zufälliges Ein- und Ausschalten

Wenn Sie die Zufallsgeneratorfunktion für einen Ausgang aktivieren, schaltet er diese ein und aus, ohne dass eine Funktionstaste ein- und ausgeschaltet werden muss.

Um einen Ausgang für diese Funktion auszuwählen, muss der Wert 128 (Bit 7) der einzelnen Ausgangseffekt-CV's gesetzt werden.

Die Schaltperiode der Zufallssequenz ist in CV115 im Bereich von 1-255 Sekunden definiert. Infolgedessen ändert sich nach jedem Ablauf der Zufallsperiode der Zustand der Ausgänge, für die der Zufallseffekt aktiviert ist, in einen neuen Ein-/Aus-Zustand, der auf dem Zufallsprinzip basiert. Es verleiht einem Wagen einen sehr realistischen Effekt, wenn er auf der Strecke rollt, und einige der Abteilleuchten werden ein- oder ausgeschaltet.

Die Anzahl der zufälligen Zustände nimmt mit der Zunahme der für diese Funktion ausgewählten Ausgänge dramatisch zu. Wir empfehlen die Verwendung von 2-4 der Ausgänge mit der zufälligen Reihenfolge, um einen realistischen visuellen Effekt zu erzielen.

6. CV Tabelle

In der Tabelle auf den folgenden Seiten sind alle CV's des Decoders aufgelistet. Wir empfehlen Ihnen, die CV-Werte nur dann zu ändern, wenn Sie sich ihrer Funktion und der Auswirkungen Ihrer Maßnahme sicher sind. Falsche CV-Einstellungen können sich negativ auf die Leistung des Decoders auswirken oder zu falschen Reaktionen auf die von der Zentrale übertragenen Befehle führen.

Die Spalte "Werkseinstellung" enthält den "Standard"-Wert der CV's (nach einem Decoder-Reset haben alle CV's den entsprechenden Wert in dieser Spalte), die Spalte "Werte - Bereich" enthält den Bereich der verwendbaren Werte für jeden CV und die Spalte "Beschreibung" enthält den Namen (falls es einen etablierten Namen gibt) und Informationen über die CV-Funktion.

CV	Werkseinstellung	Wertebereich	Beschreibung
1	3	0-127	kurze Decoder Adresse
7	4	-	Software Version (nur Lesbar)
8	78	-	Hersteller ID/RESET (lesbar = 78 = train-O-matic.) Das schreiben jedes Wertes führt zum Reset
11	25	0-255	Maximalzeit ohne Datenempfang
12	5	0-5	Zulässige Betriebsarten
13	0	0-255	Funktionstasten F1-F8 im Analogbetrieb aktivieren Bit 0 = 0(0): F1 not active in Analog mode = 1(1): F1 active in Analog mode Bit 1 = 0(0): F2 not active in Analog mode = 1(2): F2 active in Analog mode Bit 2 = 0(0): F3 not active in Analog mode = 1(4): F3 active in Analog mode Bit 3 = 0(0): F4 not active in Analog mode = 1(8): F4 active in Analog mode Bit 4 = 0(0): F5 not active in Analog mode = 1(16): F5 active in Analog mode Bit 5 = 0(0): F6 not active in Analog mode = 1(32): F6 active in Analog mode Bit 6 = 0(0): F7 not active in Analog mode = 1(64): F7 active in Analog mode Bit 7 = 0(0): F8 not active in Analog mode = 1(255): F8 active in Analog mode
14	3	0-255	Funktionstasten F0f, F0r, F9-F14 im Analogbetrieb aktivieren Bit 0 = 0(0): F0 forward not active in Analog mode = 1(1): F0 forward active in Analog mode Bit 1 = 0(0): F0 revers not active in Analog mode = 1(2): F0 revers active in Analog mode Bit 2 = 0(0): F9 not active in Analog mode = 1(4): F9 active in Analog mode Bit 3 = 0(0): F10 not active in Analog mode = 1(8): F10 active in Analog mode Bit 4 = 0(0): F11 not active in Analog mode = 1(16): F11 active in Analog mode Bit 5 = 0(0): F12 not active in Analog mode = 1(32): F12 active in Analog mode Bit 6 = 0(0): F13 not active in Analog mode = 1(64) F13 active in Analog mode Bit 7 = 0(0): F14 not active in Analog mode = 1(255): F14 active in Analog mode
15	0	0-7	LockCV (sekundäre Adresse): Die Programmierung des Decoders (CV-Änderung) ist nur zulässig, wenn CV15 = CV16 ist. CV15 kann in jeder Situation geschrieben werden
16	0	0-7	LockID: Verwenden Sie sekundäre Adressen für jeden Decoder, um zu verhindern, dass mehrere Decoder versehentlich gleichzeitig programmiert werden
17	192	192-255	Erweiterte (Lange) Lokadresse, Höherwertiges Byte

18	3	0-255	Erweiterte (Lange) Lokadresse, Niederwertiges Byte
19	0	0-127	Mehrfachtraktions - Adresse Wenn CV19> 0: Geschwindigkeit und Richtung werden von dieser Adresse bestimmt (nicht von der Haupt- oder erweiterten individuellen Adresse). Funktionen werden entweder von der Konsist (Mehrfachtraktions) Adresse oder der individuellen Adresse gesteuert, siehe auch CV's 21 + 22.
21	0	0-255	Hier definierte Funktionen sind von der Consist-Adresse gesteuert. Bit 0 = 0(0): F1 gesteuert durch Einzeladresse = 1(1): durch Consist-Adresse Bit 1 = 0(0): F2 gesteuert durch Einzeladresse = 1(2): durch Consist-Adresse Bit 2 = 0(0): F3 gesteuert durch Einzeladresse = 1(4): durch Consist-Adresse Bit 3 = 0(0): F4 gesteuert durch Einzeladresse = 1(8): durch Consist-Adresse Bit 4 = 0(0): F5 gesteuert durch Einzeladresse = 1(16): durch Consist-Adresse Bit 5 = 0(0): F6 gesteuert durch Einzeladresse = 1(32): durch Consist-Adresse Bit 6 = 0(0): F7 gesteuert durch Einzeladresse = 1(64): durch Consist-Adresse Bit 7 = 0(0): F8 gesteuert durch Einzeladresse = 1(255): durch Consist-Adresse
22	3= 1+ 2	0-63	Hier definierte Funktionen sind von der Consist-Adresse gesteuert. Bit 0 = 0(0): F0 (Vorwärts- gesteuert durch Einzeladresse = 1(1): durch Consist-Adresse Bit 1 = 0 (0): F0 (Rückwärts) gesteuert durch Einzeladresse = 1(2): durch Consist-Adresse Bit 2 = 0(0): F9 gesteuert durch Einzeladresse = 1(4): durch Consist-Adresse Bit 3 = 0(0): F10 gesteuert durch Einzeladresse = 1(8): durch Consist-Adresse Bit 4 = 0(0): F11 gesteuert durch Einzeladresse = 1(16): durch Consist-Adresse Bit 5 = 0(0): F12 gesteuert durch Einzeladresse = 1(32): durch Consist-Adresse
29	6	0-63	Konfigurationsdaten Bit 0 = 0(0): Fahrtrichtung normal = 1(1): Fahrtrichtung rückwärts Bit 1 - nicht benutzt Bit 2 = 0(0): Speisequellenkonvertierung NMRA Nur Digital (nur DCC) = 1(4): Speisequellenkonvertierung freigegeben (DC + DCC) Bit 3 - nicht benutzt Bit 4 - nicht benutzt Bit 5 = 0(0): Ein-Byte-Adressierung (Kurzadressierung) = 1(32): Zwei-Byte-Adressierung (Lang-/ erweiterte Adressierung) Bit 6 - nicht benutzt Bit 7 - nicht benutzt
30	0	0/1	CV-Fehler. Falls der Auslesewert "1" ist, gab es einen Überstrom seit dem letzten Reset. Der Reset erfolgt indem der Wert 0 geschrieben wird.

33	255= 1+ 2+ 4+ 8+ 16+ 32+ 64+ 128	0-255	F0, Vorwärts Mapping, niedrige Bitwertigkeit Bit 0 = 0(0): Out1 nicht betätigt in F0 vorwärts = 1(1): Out1 betätigt in F0 vorwärts Bit 1 = 0(0): Out2 nicht betätigt in F0 vorwärts = 1(2): Out2 betätigt in F0 vorwärts Bit 2 = 0(0): Out3 nicht betätigt in F0 vorwärts = 1(4): Out3 betätigt in F0 vorwärts Bit 3 = 0(0): Out4 nicht betätigt in F0 vorwärts = 1(8): Out4 betätigt in F0 vorwärts Bit 4 = 0(0): Out5 nicht betätigt in F0 vorwärts = 1(16): Out5 betätigt in F0 vorwärts Bit 5 = 0(0): Out6 nicht betätigt in F0 vorwärts = 1(32): Out6 betätigt in F0 vorwärts Bit 6 = 0(0): Out7 nicht betätigt in F0 vorwärts = 1(64): Out7 betätigt in F0 vorwärts Bit 7 = 0(0): Out8 nicht betätigt in F0 vorwärts = 1(128): Out8 betätigt in F0 vorwärts
34	255= 1+ 2+ 4+ 8+ 16+ 32+ 64+ 128	0-255	F0, Vorwärts Mapping, hohe Bitwertigkeit Bit 0 = 0(0): Out9 nicht betätigt in F0 vorwärts = 1(1): Out9 betätigt in F0 vorwärts Bit 1 = 0(0): Out10 nicht betätigt in F0 vorwärts = 1(2): Out10 betätigt in F0 vorwärts Bit 2 = 0(0): Out11 nicht betätigt in F0 vorwärts = 1(4): Out11 betätigt in F0 vorwärts Bit 3 = 0(0): Out12 nicht betätigt in F0 vorwärts = 1(8): Out12 betätigt in F0 vorwärts Bit 4 = 0(0): Out13 nicht betätigt in F0 vorwärts = 1(16): Out13 betätigt in F0 vorwärts Bit 5 = 0(0): Out14 nicht betätigt in F0 vorwärts = 1(32): Out14 betätigt in F0 vorwärts Bit 6 = 0(0): Out15 nicht betätigt in F0 vorwärts = 1(64): Out15 betätigt in F0 vorwärts Bit 7 = 0(0): Out16 nicht betätigt in F0 vorwärts = 1(128): Out16 betätigt in F0 vorwärts
35	255= 1+ 2+ 4+ 8+ 16+ 32+ 64+ 128	0-255	F0, Rückwärts Mapping, niedrige Bitwertigkeit Bit 0 = 0(0): Out1 nicht betätigt in F0 Rückwärts = 1(1): Out1 betätigt in F0 Rückwärts Bit 1 = 0(0): Out2 nicht betätigt in F0 Rückwärts = 1(2): Out2 betätigt in F0 Rückwärts Bit 2 = 0(0): Out3 nicht betätigt in F0 Rückwärts = 1(4): Out3 betätigt in F0 Rückwärts Bit 3 = 0(0): Out4 nicht betätigt in F0 Rückwärts = 1(8): Out4 betätigt in F0 Rückwärts Bit 4 = 0(0): Out5 nicht betätigt in F0 Rückwärts = 1(16): Out5 betätigt in F0 Rückwärts Bit 5 = 0(0): Out6 nicht betätigt in F0 Rückwärts = 1(32): Out6 betätigt in F0 Rückwärts Bit 6 = 0(0): Out7 nicht betätigt in F0 Rückwärts = 1(64): Out7 betätigt in F0 Rückwärts Bit 7 = 0(0): Out8 nicht betätigt in F0 Rückwärts = 1(128): Out8 betätigt in F0 Rückwärts

36	255= 1+ 2+ 4+ 8+ 16+ 32+ 64+ 128	0-255	<p>F0, Rückwärts Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F0 Rückwärts = 1(1): Out9 betätigt in F0 Rückwärts</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F0 Rückwärts = 1(2): Out10 betätigt in F0 Rückwärts</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F0 Rückwärts = 1(4): Out11 betätigt in F0 Rückwärts</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F0 Rückwärts = 1(8): Out12 betätigt in F0 Rückwärts</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F0 Rückwärts = 1(16): Out13 betätigt in F0 Rückwärts</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F0 Rückwärts = 1(32): Out14 betätigt in F0 Rückwärts</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F0 Rückwärts = 1(64): Out15 betätigt in F0 Rückwärts</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F0 Rückwärts = 1(128): Out16 betätigt in F0 Rückwärts</p>
37	1= 1	0-255	<p>F1, Vorwärts Mapping, niedrige Bitwertigkeit</p> <p>Bit 0 = 0(0): Out1 nicht betätigt in F1 vorwärts = 1(1): Out1 betätigt in F1 vorwärts</p> <p>Bit 1 = 0(0): Out2 nicht betätigt in F1 vorwärts = 1(2): Out2 betätigt in F1 vorwärts</p> <p>Bit 2 = 0(0): Out3 nicht betätigt in F1 vorwärts = 1(4): Out3 betätigt in F1 vorwärts</p> <p>Bit 3 = 0(0): Out4 nicht betätigt in F1 vorwärts = 1(8): Out4 betätigt in F1 vorwärts</p> <p>Bit 4 = 0(0): Out5 nicht betätigt in F1 vorwärts = 1(16): Out5 betätigt in F1 vorwärts</p> <p>Bit 5 = 0(0): Out6 nicht betätigt in F1 vorwärts = 1(32): Out6 betätigt in F1 vorwärts</p> <p>Bit 6 = 0(0): Out7 nicht betätigt in F1 vorwärts = 1(64): Out7 betätigt in F1 vorwärts</p> <p>Bit 7 = 0(0): Out8 nicht betätigt in F1 vorwärts = 1(128): Out8 betätigt in F1 vorwärts</p>
38	0	0-255	<p>F1, Vorwärts Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F1 vorwärts = 1(1): Out9 betätigt in F1 vorwärts</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F1 vorwärts = 1(2): Out10 betätigt in F1 vorwärts</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F1 vorwärts = 1(4): Out11 betätigt in F1 vorwärts</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F1 vorwärts = 1(8): Out12 betätigt in F1 vorwärts</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F1 vorwärts = 1(16): Out13 betätigt in F1 vorwärts</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F1 vorwärts = 1(32): Out14 betätigt in F1 vorwärts</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F1 vorwärts = 1(64): Out15 betätigt in F1 vorwärts</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F1 vorwärts = 1(128): Out16 betätigt in F1 vorwärts</p>

39	1= 1	0-255	<p>F1, Rückwärts Mapping, niedrige Bitwertigkeit</p> <p>Bit 0 = 0(0): Out1 nicht betätigt in F1 Rückwärts = 1(1): Out1 betätigt in F1 Rückwärts</p> <p>Bit 1 = 0(0): Out2 nicht betätigt in F1 Rückwärts = 1(2): Out2 betätigt in F1 Rückwärts</p> <p>Bit 2 = 0(0): Out3 nicht betätigt in F1 Rückwärts = 1(4): Out3 betätigt in F1 Rückwärts</p> <p>Bit 3 = 0(0): Out4 nicht betätigt in F1 Rückwärts = 1(8): Out4 betätigt in F1 Rückwärts</p> <p>Bit 4 = 0(0): Out5 nicht betätigt in F1 Rückwärts = 1(16): Out5 betätigt in F1 Rückwärts</p> <p>Bit 5 = 0(0): Out6 nicht betätigt in F1 Rückwärts = 1(32): Out6 betätigt in F1 Rückwärts</p> <p>Bit 6 = 0(0): Out7 nicht betätigt in F1 Rückwärts = 1(64): Out7 betätigt in F1 Rückwärts</p> <p>Bit 7 = 0(0): Out8 nicht betätigt in F1 Rückwärts = 1(128): Out8 betätigt in F1 Rückwärts</p>
40	0	0-255	<p>F1, Rückwärts Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F1 Rückwärts = 1(1): Out9 betätigt in F1 Rückwärts</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F1 Rückwärts = 1(2): Out10 betätigt in F1 Rückwärts</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F1 Rückwärts = 1(4): Out11 betätigt in F1 Rückwärts</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F1 Rückwärts = 1(8): Out12 betätigt in F1 Rückwärts</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F1 Rückwärts = 1(16): Out13 betätigt in F1 Rückwärts</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F1 Rückwärts = 1(32): Out14 betätigt in F1 Rückwärts</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F1 Rückwärts = 1(64): Out15 betätigt in F1 Rückwärts</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F1 Rückwärts = 1(128): Out16 betätigt in F1 Rückwärts</p>
41	2= 2	0-255	<p>F2 Mapping, niedrige Bitwertigkeit</p> <p>Bit 0 = 0(0): Out1 nicht betätigt in F2 = 1(1): Out1 betätigt in F2</p> <p>Bit 1 = 0(0): Out2 nicht betätigt in F2 = 1(2): Out2 betätigt in F2</p> <p>Bit 2 = 0(0): Out3 nicht betätigt in F2 = 1(4): Out3 betätigt in F2</p> <p>Bit 3 = 0(0): Out4 nicht betätigt in F = 1(8): Out4 betätigt in F2</p> <p>Bit 4 = 0(0): Out5 nicht betätigt in F2 = 1(16): Out5 betätigt in F2</p> <p>Bit 5 = 0(0): Out6 nicht betätigt in F2 = 1(32): Out6 betätigt in F2</p> <p>Bit 6 = 0(0): Out7 nicht betätigt in F2 = 1(64): Out7 betätigt in F2</p> <p>Bit 7 = 0(0): Out8 nicht betätigt in F2 = 1(128): Out8 betätigt in F2</p>

42	0	0-255	<p>F2 Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F2 = 1(1): Out9 betätigt in F2</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F2 = 1(2): Out10 betätigt in F2</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F2 = 1(4): Out11 betätigt in F2</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F2 = 1(8): Out12 betätigt in F2</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F2 = 1(16): Out13 betätigt in F2</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F2 = 1(32): Out14 betätigt in F2</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F2 = 1(64): Out15 betätigt in F2</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F2 = 1(128): Out16 betätigt in F2</p>
43	4= 4	0-255	<p>F3 Mapping, niedrige Bitwertigkeit</p> <p>Bit 0 = 0(0): Out1 nicht betätigt in F3 = 1(1): Out1 betätigt in F3</p> <p>Bit 1 = 0(0): Out2 nicht betätigt in F3 = 1(2): Out2 betätigt in F3</p> <p>Bit 2 = 0(0): Out3 nicht betätigt in F3 = 1(4): Out3 betätigt in F3</p> <p>Bit 3 = 0(0): Out4 nicht betätigt in F3 = 1(8): Out4 betätigt in F3</p> <p>Bit 4 = 0(0): Out5 nicht betätigt in F3 = 1(16): Out5 betätigt in F3</p> <p>Bit 5 = 0(0): Out6 nicht betätigt in F3 = 1(32): Out6 betätigt in F3</p> <p>Bit 6 = 0(0): Out7 nicht betätigt in F3 = 1(64): Out7 betätigt in F3</p> <p>Bit 7 = 0(0): Out8 nicht betätigt in F3 = 1(128): Out8 betätigt in F3</p>
44	0	0-255	<p>F3 Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F3 = 1(1): Out9 betätigt in F3</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F3 = 1(2): Out10 betätigt in F3</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F3 = 1(4): Out11 betätigt in F3</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F3 = 1(8): Out12 betätigt in F3</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F3 = 1(16): Out13 betätigt in F3</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F3 = 1(32): Out14 betätigt in F3</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F3 = 1(64): Out15 betätigt in F3</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F3 = 1(128): Out16 betätigt in F3</p>

48	0	0-255	<p>F5 Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F5 = 1(1): Out9 betätigt in F5</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F5 = 1(2): Out10 betätigt in F5</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F5 = 1(4): Out11 betätigt in F5</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F5 = 1(8): Out12 betätigt in F5</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F5 = 1(16): Out13 betätigt in F5</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F5 = 1(32): Out14 betätigt in F5</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F5 = 1(64): Out15 betätigt in F5</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F5 = 1(128): Out16 betätigt in F5</p>
49	32= 32	0-255	<p>F6 Mapping, niedrige Bitwertigkeit</p> <p>Bit 0 = 0(0): Out1 nicht betätigt in F6 = 1(1): Out1 betätigt in F6</p> <p>Bit 1 = 0(0): Out2 nicht betätigt in F6 = 1(2): Out2 betätigt in F6</p> <p>Bit 2 = 0(0): Out3 nicht betätigt in F6 = 1(4): Out3 betätigt in F6</p> <p>Bit 3 = 0(0): Out4 nicht betätigt in F6 = 1(8): Out4 betätigt in F6</p> <p>Bit 4 = 0(0): Out5 nicht betätigt in F6 = 1(16): Out5 betätigt in F6</p> <p>Bit 5 = 0(0): Out6 nicht betätigt in F6 = 1(32): Out6 betätigt in F6</p> <p>Bit 6 = 0(0): Out7 nicht betätigt in F6 = 1(64): Out7 betätigt in F6</p> <p>Bit 7 = 0(0): Out8 nicht betätigt in F6 = 1(128): Out8 betätigt in F6</p>
50	0	0-255	<p>F6 Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F6 = 1(1): Out9 betätigt in F6</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F6 = 1(2): Out10 betätigt in F6</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F6 = 1(4): Out11 betätigt in F6</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F6 = 1(8): Out12 betätigt in F6</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F6 = 1(16): Out13 betätigt in F6</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F6 = 1(32): Out14 betätigt in F6</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F6 = 1(64): Out15 betätigt in F6</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F6 = 1(128): Out16 betätigt in F6</p>

54	0	0-255	<p>F8 Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F8 = 1(1): Out9 betätigt in F8</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F8 = 1(2): Out10 betätigt in F8</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F8 = 1(4): Out11 betätigt in F8</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F8 = 1(8): Out12 betätigt in F8</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F8 = 1(16): Out13 betätigt in F8</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F8 = 1(32): Out14 betätigt in F8</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F8 = 1(64): Out15 betätigt in F8</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F8 = 1(128): Out16 betätigt in F8</p>
55	0	0-255	<p>F9 Mapping, niedrige Bitwertigkeit</p> <p>Bit 0 = 0(0): Out1 nicht betätigt in F9 = 1(1): Out1 betätigt in F9</p> <p>Bit 1 = 0(0): Out2 nicht betätigt in F9 = 1(2): Out2 betätigt in F9</p> <p>Bit 2 = 0(0): Out3 nicht betätigt in F9 = 1(4): Out3 betätigt in F9</p> <p>Bit 3 = 0(0): Out4 nicht betätigt in F = 1(8): Out4 betätigt in F9</p> <p>Bit 4 = 0(0): Out5 nicht betätigt in F9 = 1(16): Out5 betätigt in F9</p> <p>Bit 5 = 0(0): Out6 nicht betätigt in F9 = 1(32): Out6 betätigt in F9</p> <p>Bit 6 = 0(0): Out7 nicht betätigt in F9 = 1(64): Out7 betätigt in F9</p> <p>Bit 7 = 0(0): Out8 nicht betätigt in F9 = 1(128): Out8 betätigt in F9</p>
56	1= 1	0-255	<p>F9 Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F9 = 1(1): Out9 betätigt in F9</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F9 = 1(2): Out10 betätigt in F9</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F9 = 1(4): Out11 betätigt in F9</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F9 = 1(8): Out12 betätigt in F9</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F9 = 1(16): Out13 betätigt in F9</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F9 = 1(32): Out14 betätigt in F9</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F9 = 1(64): Out15 betätigt in F9</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F2 = 1(128): Out16 betätigt in F2</p>

57	0	0-255	<p>F10 Mapping, niedrige Bitwertigkeit</p> <p>Bit 0 = 0(0): Out1 nicht betätigt in F10 = 1(1): Out1 betätigt in F10</p> <p>Bit 1 = 0(0): Out2 nicht betätigt in F10 = 1(2): Out2 betätigt in F10</p> <p>Bit 2 = 0(0): Out3 nicht betätigt in F10 = 1(4): Out3 betätigt in F10</p> <p>Bit 3 = 0(0): Out4 nicht betätigt in F10 = 1(8): Out4 betätigt in F10</p> <p>Bit 4 = 0(0): Out5 nicht betätigt in F10 = 1(16): Out5 betätigt in F10</p> <p>Bit 5 = 0(0): Out6 nicht betätigt in F10 = 1(32): Out6 betätigt in F10</p> <p>Bit 6 = 0(0): Out7 nicht betätigt in F10 = 1(64): Out7 betätigt in F10</p> <p>Bit 7 = 0(0): Out8 nicht betätigt in F10 = 1(128): Out8 betätigt in F10</p>
58	2= 2	0-255	<p>F10 Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F10 = 1(1): Out9 betätigt in F10</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F10 = 1(2): Out10 betätigt in F10</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F10 = 1(4): Out11 betätigt in F10</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F10 = 1(8): Out12 betätigt in F10</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F10 = 1(16): Out13 betätigt in F10</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F2 = 1(32): Out14 betätigt in F10</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F210 = 1(64): Out15 betätigt in F10</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F10 = 1(128): Out16 betätigt in F10</p>
59	0	0-255	<p>F11 Mapping, niedrige Bitwertigkeit</p> <p>Bit 0 = 0(0): Out1 nicht betätigt in F11 = 1(1): Out1 betätigt in F11</p> <p>Bit 1 = 0(0): Out2 nicht betätigt in F11 = 1(2): Out2 betätigt in F11</p> <p>Bit 2 = 0(0): Out3 nicht betätigt in F11 = 1(4): Out3 betätigt in F11</p> <p>Bit 3 = 0(0): Out4 nicht betätigt in F11 = 1(8): Out4 betätigt in F11</p> <p>Bit 4 = 0(0): Out5 nicht betätigt in F11 = 1(16): Out5 betätigt in F11</p> <p>Bit 5 = 0(0): Out6 nicht betätigt in F11 = 1(32): Out6 betätigt in F11</p> <p>Bit 6 = 0(0): Out7 nicht betätigt in F11 = 1(64): Out7 betätigt in F11</p> <p>Bit 7 = 0(0): Out8 nicht betätigt in F11 = 1(128): Out8 betätigt in F11</p>

60	4= 4	0-255	<p>F11 Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F11 = 1(1): Out9 betätigt in F11</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F11 = 1(2): Out10 betätigt in F11</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F11 = 1(4): Out11 betätigt in F11</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F11 = 1(8): Out12 betätigt in F11</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F11 = 1(16): Out13 betätigt in F11</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F11 = 1(32): Out14 betätigt in F11</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F11 = 1(64): Out15 betätigt in F11</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F11 = 1(128): Out16 betätigt in F11</p>
61	0	0-255	<p>F12 Mapping, niedrige Bitwertigkeit</p> <p>Bit 0 = 0(0): Out1 nicht betätigt in F12 = 1(1): Out1 betätigt in F12</p> <p>Bit 1 = 0(0): Out2 nicht betätigt in F12 = 1(2): Out2 betätigt in F12</p> <p>Bit 2 = 0(0): Out3 nicht betätigt in F12 = 1(4): Out3 betätigt in F12</p> <p>Bit 3 = 0(0): Out4 nicht betätigt in F12 = 1(8): Out4 betätigt in F12</p> <p>Bit 4 = 0(0): Out5 nicht betätigt in F12 = 1(16): Out5 betätigt in F12</p> <p>Bit 5 = 0(0): Out6 nicht betätigt in F12 = 1(32): Out6 betätigt in F12</p> <p>Bit 6 = 0(0): Out7 nicht betätigt in F12 = 1(64): Out7 betätigt in F12</p> <p>Bit 7 = 0(0): Out8 nicht betätigt in F12 = 1(128): Out8 betätigt in F12</p>
62	8= 8	0-255	<p>F12 Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F12 = 1(1): Out9 betätigt in F12</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F12 = 1(2): Out10 betätigt in F12</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F12 = 1(4): Out11 betätigt in F12</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F12 = 1(8): Out12 betätigt in F12</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F12 = 1(16): Out13 betätigt in F12</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F12 = 1(32): Out14 betätigt in F12</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F12 = 1(64): Out15 betätigt in F12</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F12 = 1(128): Out16 betätigt in F12</p>

105	0	0-255	Benutzerdaten
106	0	0-255	Benutzerdaten
108	0	0-100	Actual QoS (Quality of Service)
109	100	0-100	QoS minimal value
110	0	0-255	Output switching off delay time
111	0	0-255	Output switching on delay time
112	25	1-127	Aufblendung AUX Lichteffect Fading EIN, ex.: 1 = 8ms, 15 = 120ms, 125 = 1000ms
113	15	1-127	Ausblendung AUX Lichteffect Fading AUS
114	3	0-7	Verzögerung, Leuchtröhre Start, Blinkverzögerung 1-8 Verzögerungsschritt [0..7]
115	10	1-255	Zufallszeit, 1s-255s
116	3	0-7	Flackerhäufigkeit: schnell-langsam 0..7 werte
117	3	0-7	Wiederholrate des fehlerhaften Leuchtröhreneffekts, 0 schnelle Wiederholrate, 7 langsame Wiederholrate
120	255	0-255	Out 1 Lichtstärke, [1-255]
121	255	0-255	Out 2 Lichtstärke, [1-255]
122	255	0-255	Out 3 Lichtstärke, [1-255]
123	255	0-255	Out 4 Lichtstärke, [1-255]
124	255	0-255	Out 5 Lichtstärke, [1-255]
125	255	0-255	Out 6 Lichtstärke, [1-255]
126	255	0-255	Out 7 Lichtstärke, [1-255]
127	255	0-255	Out 8 Lichtstärke, [1-255]
128	255	0-255	Out 9 Lichtstärke, [1-255]
129	255	0-255	Out 10 Lichtstärke, [1-255]
130	255	0-255	Out 11 Lichtstärke, [1-255]
131	255	0-255	Out 12 Lichtstärke, [1-255]
132	255	0-255	Out 13 Lichtstärke, [1-255]
133	255	0-255	Out 14 Lichtstärke, [1-255]
134	255	0-255	Out 15 Lichtstärke, [1-255]
135	255	0-255	Out 16 Lichtstärke, [1-255]
136	1	0-255	Out 1, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeleuchte, 4-Effekt fehlerhafter Leuchtröhre
137	1	0-255	Out 2, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeleuchte, 4-Effekt fehlerhafter Leuchtröhre

138	1	0-255	Out 3, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeuchte, 4-Effekt fehlerhafter Leuchtröhre
139	1	0-255	Out 4, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeuchte, 4-Effekt fehlerhafter Leuchtröhre
140	1	0-255	Out 5, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeuchte, 4-Effekt fehlerhafter Leuchtröhre
141	1	0-255	Out 6, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeuchte, 4-Effekt fehlerhafter Leuchtröhre
142	1	0-255	Out 7, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeuchte, 4-Effekt fehlerhafter Leuchtröhre
143	1	0-255	Out 8, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeuchte, 4-Effekt fehlerhafter Leuchtröhre

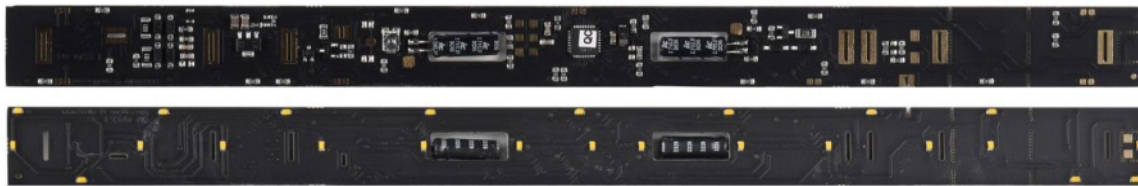
144	1	0-255	Out 9, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeuchte, 4-Effekt fehlerhafter Leuchtröhre
145	1	0-255	Out 10, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeuchte, 4-Effekt fehlerhafter Leuchtröhre
146	1	0-255	Out 11, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeuchte, 4-Effekt fehlerhafter Leuchtröhre
147	1	0-255	Out 12, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeuchte, 4-Effekt fehlerhafter Leuchtröhre
148	1	0-255	Out 13, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeuchte, 4-Effekt fehlerhafter Leuchtröhre
149	1	0-255	Out 14, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeuchte, 4-Effekt fehlerhafter Leuchtröhre

150	1	0-255	Out 15, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeleuchte, 4-Effekt fehlerhafter Leuchtröhre
151	1	0-255	Out 16, Effekt: Bit7 = 128 Zufallsbetrieb / 0 Normalbetrieb + Bit0, 1, 3 = 0-Kontinuierlich, 1-Blendeleuchte, 2-Leuchtröhre, 3-Blinkeleuchte, 4-Effekt fehlerhafter Leuchtröhre
152	0	0-1	Letzten Zustand speichern 1-speichern 0-nicht speichern
160	0	0-255	F13 Mapping, niedrige Bitwertigkeit Bit 0 = 0(0): Out1 nicht betätigt in F13 = 1(1): Out1 betätigt in F13 Bit 1 = 0(0): Out2 nicht betätigt in F13 = 1(2): Out2 betätigt in F13 Bit 2 = 0(0): Out3 nicht betätigt in F13 = 1(4): Out3 betätigt in F13 Bit 3 = 0(0): Out4 nicht betätigt in F13 = 1(8): Out4 betätigt in F13 Bit 4 = 0(0): Out5 nicht betätigt in F13 = 1(16): Out5 betätigt in F13 Bit 5 = 0(0): Out6 nicht betätigt in F13 = 1(32): Out6 betätigt in F13 Bit 6 = 0(0): Out7 nicht betätigt in F13 = 1(64): Out7 betätigt in F13 Bit 7 = 0(0): Out8 nicht betätigt in F13 = 1(128): Out8 betätigt in F13
161	16= 16	0-255	F13 Mapping, hohe Bitwertigkeit Bit 0 = 0(0): Out9 nicht betätigt in F13 = 1(1): Out9 betätigt in F13 Bit 1 = 0(0): Out10 nicht betätigt in F13 = 1(2): Out10 betätigt in F13 Bit 2 = 0(0): Out11 nicht betätigt in F13 = 1(4): Out11 betätigt in F13 Bit 3 = 0(0): Out12 nicht betätigt in F13 = 1(8): Out12 betätigt in F13 Bit 4 = 0(0): Out13 nicht betätigt in F13 = 1(16): Out13 betätigt in F13 Bit 5 = 0(0): Out14 nicht betätigt in F13 = 1(32): Out14 betätigt in F13 Bit 6 = 0(0): Out15 nicht betätigt in F13 = 1(64): Out15 betätigt in F13 Bit 7 = 0(0): Out16 nicht betätigt in F13 = 1(128): Out16 betätigt in F13

162	0	0-255	<p>F14 Mapping, niedrige Bitwertigkeit</p> <p>Bit 0 = 0(0): Out1 nicht betätigt in F14 = 1(1): Out1 betätigt in F14</p> <p>Bit 1 = 0(0): Out2 nicht betätigt in F14 = 1(2): Out2 betätigt in F14</p> <p>Bit 2 = 0(0): Out3 nicht betätigt in F14 = 1(4): Out3 betätigt in F14</p> <p>Bit 3 = 0(0): Out4 nicht betätigt in F14 = 1(8): Out4 betätigt in F14</p> <p>Bit 4 = 0(0): Out5 nicht betätigt in F14 = 1(16): Out5 betätigt in F14</p> <p>Bit 5 = 0(0): Out6 nicht betätigt in F14 = 1(32): Out6 betätigt in F14</p> <p>Bit 6 = 0(0): Out7 nicht betätigt in F14 = 1(64): Out7 betätigt in F14</p> <p>Bit 7 = 0(0): Out8 nicht betätigt in F14 = 1(128): Out8 betätigt in F14</p>
163	32= 32	0-255	<p>F14 Mapping, hohe Bitwertigkeit</p> <p>Bit 0 = 0(0): Out9 nicht betätigt in F14 = 1(1): Out9 betätigt in F14</p> <p>Bit 1 = 0(0): Out10 nicht betätigt in F14 = 1(2): Out10 betätigt in F14</p> <p>Bit 2 = 0(0): Out11 nicht betätigt in F14 = 1(4): Out11 betätigt in F14</p> <p>Bit 3 = 0(0): Out12 nicht betätigt in F14 = 1(8): Out12 betätigt in F14</p> <p>Bit 4 = 0(0): Out13 nicht betätigt in F14 = 1(16): Out13 betätigt in F14</p> <p>Bit 5 = 0(0): Out14 nicht betätigt in F14 = 1(32): Out14 betätigt in F14</p> <p>Bit 6 = 0(0): Out15 nicht betätigt in F14 = 1(64): Out15 betätigt in F14</p> <p>Bit 7 = 0(0): Out16 nicht betätigt in F14 = 1(128): Out16 betätigt in F14</p>
164	0	0-255	<p>F15 Mapping, niedrige Bitwertigkeit</p> <p>Bit 0 = 0(0): Out1 nicht betätigt in F15 = 1(1): Out1 betätigt in F15</p> <p>Bit 1 = 0(0): Out2 nicht betätigt in F15 = 1(2): Out2 betätigt in F15</p> <p>Bit 2 = 0(0): Out3 nicht betätigt in F15 = 1(4): Out3 betätigt in F15</p> <p>Bit 3 = 0(0): Out4 nicht betätigt in F15 = 1(8): Out4 betätigt in F15</p> <p>Bit 4 = 0(0): Out5 nicht betätigt in F15 = 1(16): Out5 betätigt in F15</p> <p>Bit 5 = 0(0): Out6 nicht betätigt in F15 = 1(32): Out6 betätigt in F15</p> <p>Bit 6 = 0(0): Out7 nicht betätigt in F15 = 1(64): Out7 betätigt in F15</p> <p>Bit 7 = 0(0): Out8 nicht betätigt in F15 = 1(128): Out8 betätigt in F15</p>

168	62	0-255	SPP-Timeout. Eine Einheit sind 80 Millisekunden. Das werkseitig eingestellte Timeout ist $62 \cdot 0,080 = 4,96 \sim 5$ Sekunden. Nachdem die Elektronik von der Schiene getrennt wurde, bleiben die Lichter noch 5 Sekunden lang an.
169	28	0-28	Special function number for QoS saving (F28)
170	2	0-2	PWM Output frequency (0 = 125 Hz, 1 = 250 Hz, 2 = 500 Hz)

Operating instructions



Index

1. Introduction	page 2
2. Function output mapping	
2.1 Function outputs	page 3
2.2 Function button mapping	page 4
3. Installation	page 4
3.1 Driving cab coach Bybdzf 482, type Halberstadt	page 5
3.2 Driving cab coach Bnrdfz 483, type Silberling	page 6
3.3 Driving cab coach BDnf 738	page 7
3.4 Driving cab coach BDnf 740	page 8
3.5 Coach type Halberstadt “middle entry coach”	page 9
3.6 Coach type Halberstadt “side corridor coach”	page 10
3.7 Coach ABn/Bn, type Silberling	page 11
4. Standard CV – programming	
4.1 Adress programming	page 12
4.2 Secondary address (decoder lock)	page 13
4.3 Decoder-Reset	page 13
4.4 Function-mapping	page 14
4.5 DC operation	page 15
4.6 Onboard power pack (SPP)	page 16
4.8 User Data	page 16
4.9 Other functions	page 16
5. Effects for function outputs	
5.1 Light intensity	page 17
5.2 Light effects	page 17
5.2.1 Fade effect	page 17
5.2.2 Fluorescent effect	page 17
5.2.3 Flickering effect	page 18
5.2.4 Defective neon effect	page 18
5.2.5 Turn On/Off Delay	page 18
5.2.6 Random sequence generator function	page 18
6. CV table	page 19

1. Introduction

Congratulations for the purchase of a TILLIG quality product. We are giving you all the information you need to install the interior lighting and customize all the features, as you wish.

This type of interior lightning is matching to follow coaches:

- Driving cab coach Bybdzf 482, type Halberstadt



- Driving cab coach Bnrdfz 483, type Silberling



- Driving cab coach BDnf 738



- Driving cab coach BDnf 740



- Coach Bmh/ABy/By, type Halberstadt "middle entry coach" (new construction)



- Coach type Halberstadt "side corridor coach"



- Coach ABn/Bn, type Silberling



2. Function output mapping

The integrated decoder has 16 outputs, so that single or groups of LED's can be switched on or off separately.

2.1 Function outputs

At the following drawing you can see how the LED's are connected.

For this special type of interior lighting we use cutting points, so you can specify the lighting to the coach - type and interior. For this cut through the connection between both solder pads with a sharp knife. If you cut through the wrong one, you can reconnect it by soldering.

Single LED's:

AUX1=LED1; AUX2=LED2; AUX3=LED3; AUX4=LED4; AUX5=LED5; AUX6=LED6; AUX7=LED7;
AUX8=LED8; AUX9=LED9; AUX10=LED10; AUX11=LED11; AUX12=LED12; AUX13=LED13

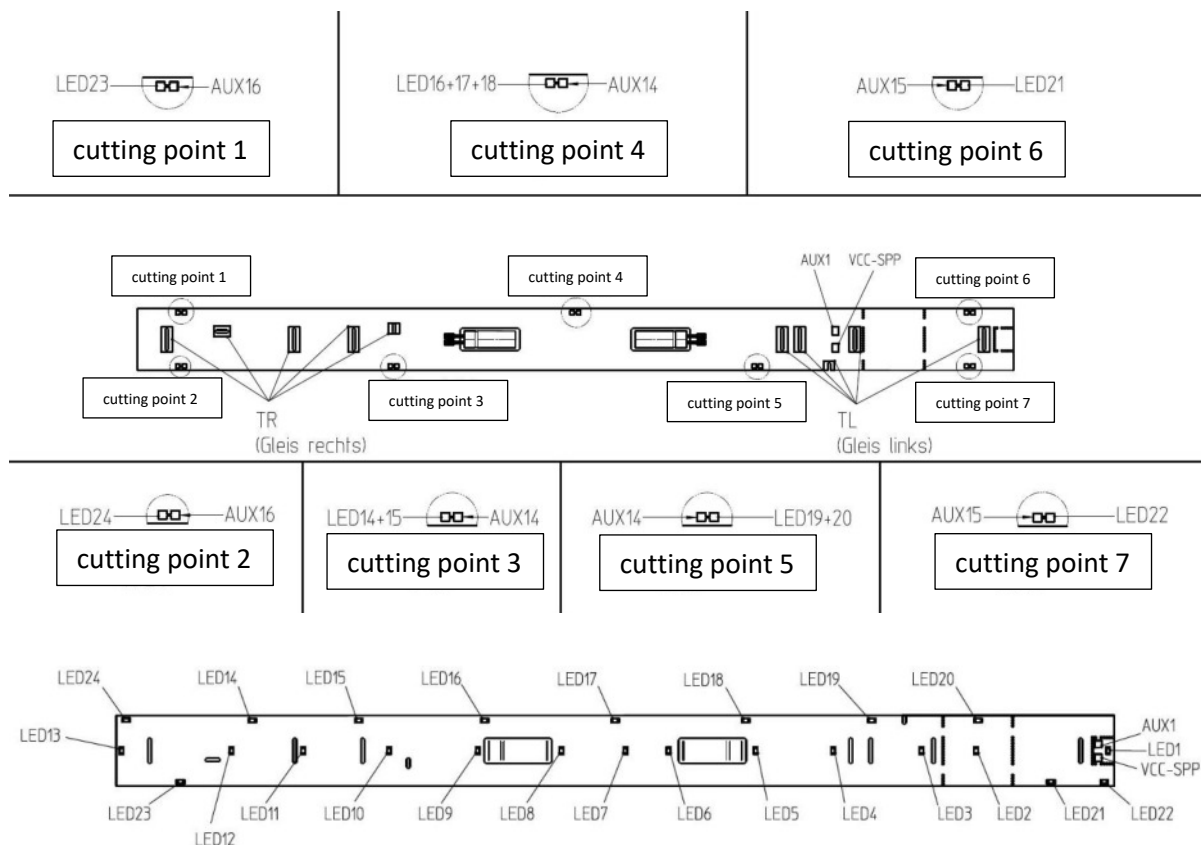
Groups:

AUX14= corridor = (group 1=LED14+15)+(group 2=LED16+17+18)+(group 3=LED19+20)

AUX15= front toilet = (LED21)+(LED22)

AUX16= back toilet = (LED23)+(LED24)

LED's in brackets could be separated while cutting points



2.2 Function button mapping

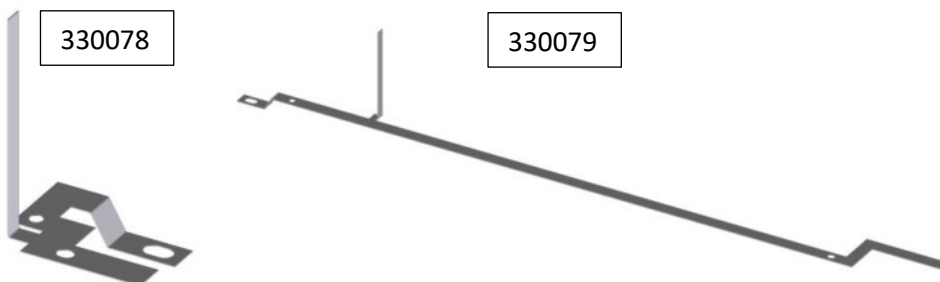
F0	All function outputs on
F1	AUX1
F2	AUX2
F3	AUX3
F4	AUX4
F5	AUX5
F6	AUX6
F7	AUX7
F8	AUX8
F9	AUX9
F10	AUX10
F11	AUX11
F12	AUX12
F13	AUX13
F14	AUX14 (side corridor)
F15	AUX15 (toilet front)
F16	AUX16 (toilet back)

3. Installation

For Plug and Play you must look for this symbol: 

Older wagons can be used also, but you must to install the needed contact strips or using wires.
-WARNING- : For that it is needable to demount the whole choach. This can lead to loss or damage to parts.

For driving cab coaches BDnf 738 and BDnf 740 you need:



For coaches ABn/Bn, type "Silberling" you need 333435:



NOTE FOR DCC USERS: If you have a driving cab coach and intend to use it in DCC mode make sure you program a secondary address (see Fehler! Verweisquelle konnte nicht gefunden werden.) before installing it to the coach. This is to prevent programing issues if other decoders are installed in the same coach.

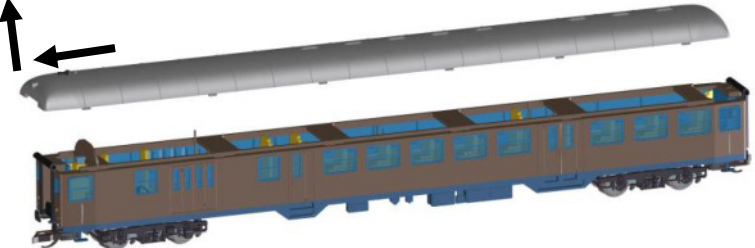
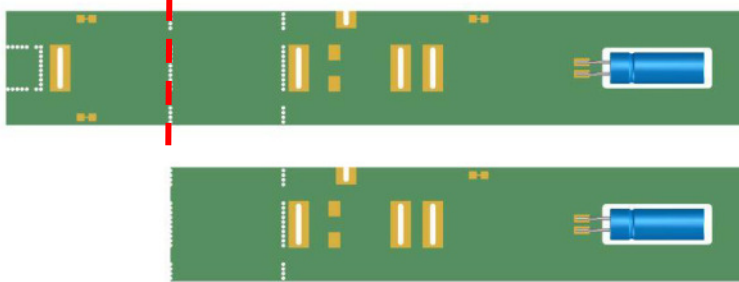
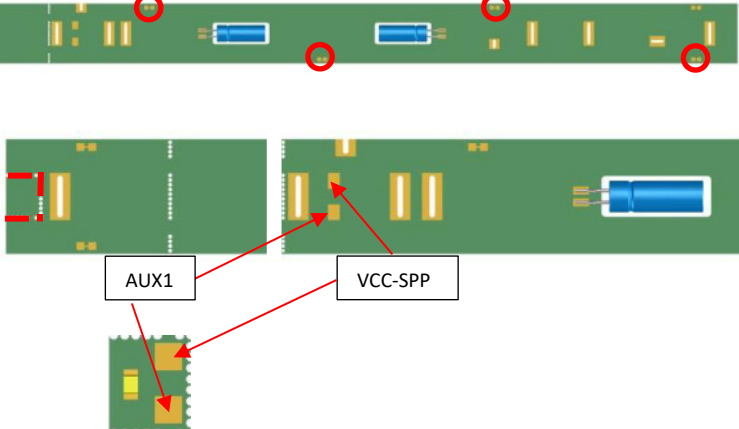
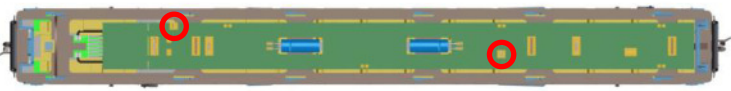
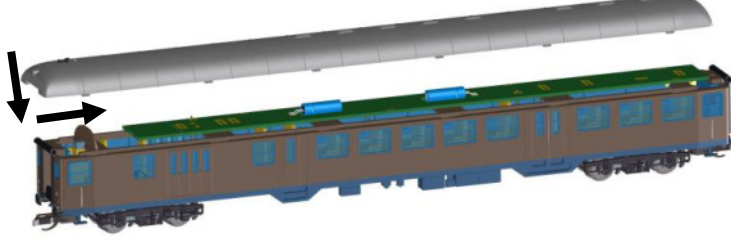
3.1 Driving cab coach Bybdzf 482, type Halberstadt

<p>The interior lighting will be installed on the top of the interior.</p> <ol style="list-style-type: none">1. Remove the body, for that spread the body at the doors position.	
<ol style="list-style-type: none">2. Trim the length to the first mark. Use a sharp knife or a cutting disc for that.	
<ol style="list-style-type: none">3. Cut through at the cutting points 1; 3; 4 and 5 <p>Note: AUX1; AUX14 and AUX15 will not be available anymore. AUX1 can be used as a free function output. The LED from the cut off PCB part can be used as AUX1 by wires.</p>	
<ol style="list-style-type: none">4. Install the lighting at the contact rails by soldering	
<ol style="list-style-type: none">5. Reinstall the body	

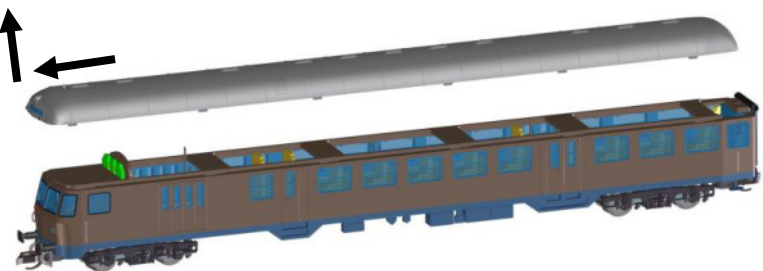
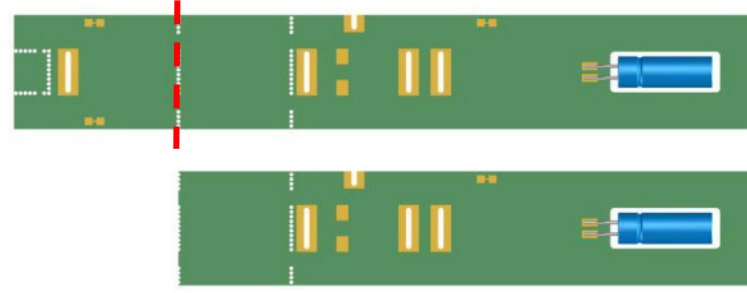
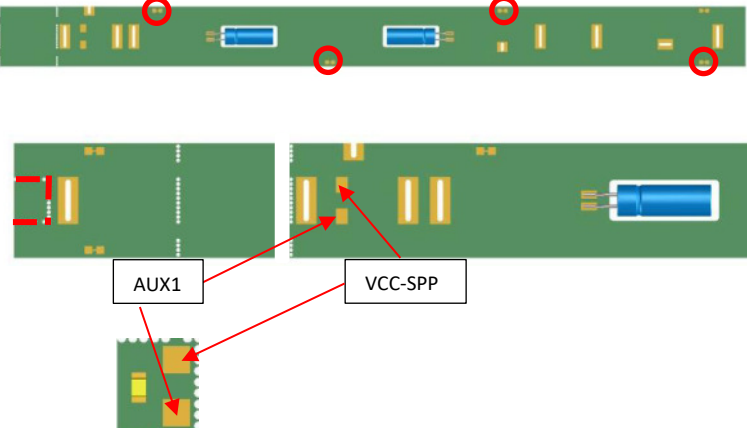
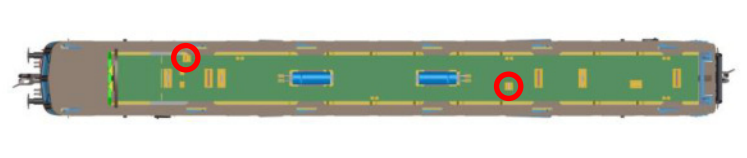
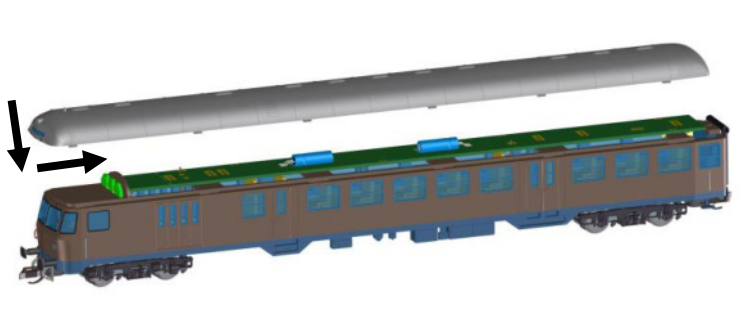
3.2 Driving cab coach Bnrdfz 483, type Silberling

<p>The interior lightning will be installed on the top of the interior.</p> <ol style="list-style-type: none">1. Remove the body, for that spread the body at the doors	
<ol style="list-style-type: none">2. Trim the length to the first mark. Use a sharp knife or a cutting disc for that.	
<ol style="list-style-type: none">3. Cut through at the cutting points 1; 2; 3; 4 and 5 <p>Note: AUX1; AUX14; AUX15 and AUX16 will not be available anymore. AUX1 can be used as a free function output. The LED from the cut off PCB part can be used as AUX1 by wires.</p>	
<ol style="list-style-type: none">4. Install the lighting at the contact rails by soldering	
<ol style="list-style-type: none">5. Reinstall the body	

3.3 Driving cab coach B Dnf 738

<p>The interior lighting will be installed on the top of the body.</p> <ol style="list-style-type: none">1. Remove the roof. For that slide the roof to the front.	
<ol style="list-style-type: none">2. Trim the length to the first mark. Use a sharp knife or a cutting disc for that.	
<ol style="list-style-type: none">3. Cut through at the cutting points 1; 3; 4 and 5 <p>Note: AUX1; AUX14 and AUX15 won't be anymore. AUX1 can be used as a free function output. The LED from the cut off PCB part can be used for for the function of AUX1 by wires.</p>	
<ol style="list-style-type: none">4. Install the lighting at the contact rails by soldering	
<ol style="list-style-type: none">5. Reinstall the roof	

3.4 Driving cab coach B Dnf 740

<p>The interior lighting will be installed on the top of the body.</p> <ol style="list-style-type: none">1. Remove the roof. For that slide the roof to the front.	
<ol style="list-style-type: none">2. Trim the length to the first mark. Use for that a sharp knife or a cutting disc.	
<ol style="list-style-type: none">3. Cut through at the cutting points 1; 3; 4 and 5 <p>Note: AUX1; AUX14 and AUX15 will not be available anymore. AUX1 can be used as a free function output. The LED from the cut off PCB part can be used as AUX1 by wires.</p>	
<ol style="list-style-type: none">4. Install the lighting at the contact rails by soldering	
<ol style="list-style-type: none">5. Reinstall the roof	

3.5 Coach Bmh/ABy/By, Type "Halberstadt middle entry coach"

<p>The interior lighting will be installed on the top of the interior.</p> <ol style="list-style-type: none">1. Remove the body, for that spread the body at the doors	
<ol style="list-style-type: none">2. For ABy (see interior) cut through at the cutting points 1; 3; 5 and 6 <p>For all other interior versions use the cutting points 1; 3; 4; 5; and 6.</p> <p>Note: AUX14 will not be available anymore.</p>	
<ol style="list-style-type: none">3. Install the lighting at the contact rails by soldering	
<ol style="list-style-type: none">4. Reinstall the body	

3.6 Coach Type Halberstadt "side corridor coach"

<p>The interior lighting will be installed on the top of the body.</p> <ol style="list-style-type: none">1. Remove the roof. For that unlock the detents with a thin piece of plastic between the roof and the body	
<ol style="list-style-type: none">2. Cut through the cutting points 2 and 7	
<ol style="list-style-type: none">3. Install the lighting at the contact rails by soldering	
<ol style="list-style-type: none">4. Reinstall the roof	

3.7 Coach ABn/Bn, type Silberling

<p>The interior lighting will be installed on the top of the body.</p> <p>1. Remove the roof. For that remove the front rubber corridor and slide the roof to the front.</p>	
<p>5. For ABn (see interior) cut through at the cutting points 1; 3; 5 and 6</p> <p>For all other interior versions use the cutting points 1; 3; 4; 5; and 6.</p> <p>For note: AUX14 will not be available anymore.</p>	
<p>2. Install the lighting at the contact rails by soldering</p>	
<p>3. Reinstall the roof and after that the rubber corridor.</p>	

4. Standard CV – programming

-WARNING- : To make sure you are programming only the lighting decoder you must activate the secondary address (see Fehler! Verweisquelle konnte nicht gefunden werden.)

You can write CV's on Programming Track (PT) or Main Track (PoM), read only on PT.

4.1 Adress programming

The on board function decoder of the interior light can be used either with short addresses (1-127) or long addresses (1-9999). The factory default setting is short addressing (bit5 of CV29 is 0), with the address 3 (CV1 = 3).

The address can be changed by placing the decoder on the Programming Track (PT) or Main Track (PoM), and changing the CV1 value, according to the instructions of your Command Station.

If long addressing is needed, the addressing mode must be changed in the configuration CV of the decoder (bit5 of CV29). Changing the bit5 value of CV29 to 1 will activate the long addressing mode, and the decoder will respond to the long address specified in CV17 and CV18. Bit5 has a decimal value of 32, so changing bit5 to binary 1 is equivalent with adding 32 to the decimal value of CV29 (CV29 has the factory default value 6, activating bit5 means, to add 32 to this value, $6 + 32 = 38$, the new value for CV29 will be 38).

The long addresses will be calculated with the following algorithm (in our example we will consider the long address 2000):

- divide the desired long address with 256 (in our example $2000 / 256 = 7$, remainder = 208)
- add 192 to the result and program it in CV17 ($7 + 192 = 199$, program the value 199 in CV17)
- program the value of the remainder of the division in CV18 (program the value 208 in CV18)

After programming CV29, CV17 and CV18 with the mentioned values, the decoder can be accessed with the address 2000. To switch back to short addressing, the bit5 of CV29 must be deactivated.

4.2 Secondary address (decoder lock)

This will be necessary while using the interior lightning at driving cab coaches.

When using multiple decoders within the same housing, a secondary address is needed to be able to select each decoder. In this way all the decoders that are inside the same housing (carriage body) can be programmed on the Programming Track without removing them. The secondary addresses are programmed into CV16 before the decoders (in our case the interior lighting with the onboard decoder) is being assembled in the housing. The range of secondary addresses is between 1 and 7 (value of 0 means that secondary addressing is not used). This allows the use of maximum 7 decoders in the same carriage or locomotive housing, which is more than enough.

If the value of CV16 is not equal to zero, the decoders will accept programming commands only if the secondary decoder address that is about to be programmed is written prior in CV15, and it matches the value in CV16 (it should be the same as decoder'S CV16 value).

When using secondary address, it is important to know that the only CV that can be read and written without knowing the secondary address is CV15. For this reason the used values are limited to the range of 1 to 7. If the secondary decoder address is forgotten, within 7 try's it can be found.

Assigning secondary addresses to each decoder of the railcar or carriage sets, when placing them on the Programming Track, only the decoder for which the CV15 = CV16 will be programmed. In this way we can program all the decoders independently, even if they are all on the programming track at the same time.

For example: The driving cab coach Bybdzf 482, Halberstadt type, can be equipped with interior light. This model has an OnBoard function decoder. That means that before you install the interior lighting, the OnBoard function decoder must be programmed with a secondary address 1 (CV16=1).

Now you can install the interior light. It should get the secondary address 2. For that you need to write CV16=2. The the interior light will be available for programming using the secondary address 2.

During operation both decoders are accessed on short (primary) address 3, which is the standard (default) decoder address, but they can be programmed separately/individually selecting their secondary addresses in CV15.

This means if you want to program the OnBoard function decoder, you must write the secondary address to CV15 (CV15=1). Now only the OnBoard function decoder with this secondary address will be programmed, the interior light (and all others) is (are) locked.

If you want to program the interior light, prior doing this, you must program its secondary address to CV15 by writing CV15=2.

4.3 Decoder-Reset

The factory default CV values are specified in the CV table "Default value" column. The decoder can load the factory default values anytime by resetting it. Any value must be written in CV8 to reset the decoder to the factory default CV values.

4.4 Function-mapping

Each function (from F0 to F16) can be used to activate/deactivate one or more outputs (from the total of 16 outputs) of the on-board decoder. The relationship between functions and outputs is called Function Mapping. The mapping is performed by programming like the bit-mask shows.

For the **functions F0** (generally called light function) and **F1** the mapping can be defined separately for each **driving direction**, so 4 CV's are used for each function. The other functions (F2-F16) don't depend on the driving direction, so 2 CV's are used for each function, to map the function to each physical output of the on board decoder.

OUTPUT		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
bit		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
function	CV																
F0 fwd.	33									128	64	32	16	8	4	2	1
	34	128	64	32	16	8	4	2	1								
F0 rev	35									128	64	32	16	8	4	2	1
	36	128	64	32	16	8	4	2	1								
F1 fwd.	37									128	64	32	16	8	4	2	1
	38	128	64	32	16	8	4	2	1								
F1 rev	39									128	64	32	16	8	4	2	1
	40	128	64	32	16	8	4	2	1								
F2	41									128	64	32	16	8	4	2	1
	42	128	64	32	16	8	4	2	1								
F3	43									128	64	32	16	8	4	2	1
	44	128	64	32	16	8	4	2	1								
F4	45									128	64	32	16	8	4	2	1
	46	128	64	32	16	8	4	2	1								
F5	47									128	64	32	16	8	4	2	1
	48	128	64	32	16	8	4	2	1								
F6	49									128	64	32	16	8	4	2	1
	50	128	64	32	16	8	4	2	1								
F7	51									128	64	32	16	8	4	2	1
	52	128	64	32	16	8	4	2	1								
F8	53									128	64	32	16	8	4	2	1
	54	128	64	32	16	8	4	2	1								
F9	55									128	64	32	16	8	4	2	1
	56	128	64	32	16	8	4	2	1								
F10	57									128	64	32	16	8	4	2	1
	58	128	64	32	16	8	4	2	1								
F11	59									128	64	32	16	8	4	2	1
	60	128	64	32	16	8	4	2	1								
F12	61									128	64	32	16	8	4	2	1
	62	128	64	32	16	8	4	2	1								
F13	160									128	64	32	16	8	4	2	1
	161	128	64	32	16	8	4	2	1								
F14	162									128	64	32	16	8	4	2	1
	163	128	64	32	16	8	4	2	1								
F15	164									128	64	32	16	8	4	2	1
	165	128	64	32	16	8	4	2	1								
F16	166									128	64	32	16	8	4	2	1
	167	128	64	32	16	8	4	2	1								

For example:

If you want to use function F2 to activate output 4, CV41 and CV42 will be used for mapping (which configures / maps F2).

Bit3 is used to activate Output 4, its decimal value is 8, so CV41 will be programmed with the value 8, and CV42 with the value of 0.

If we want to use F2 to activate Output3 too, we will add to the previously determined value the corresponding value for Output3 (bit2, with a decimal value 4). CV41 will be programmed with the sum of $4 + 8 = 12$. If Out13 and 14 will be used with F2, the corresponding bits are bit4 and bit5 in CV42, with the values 16 and 32, thus CV42 will be programmed with the value $16 + 32 = 48$.

For functions F0 and F1, 4 CVs are used for mapping, two for each travel direction. F0 is factory configured in such a way that all outputs are activated for both directions. F1 is factory configured to activate Output1 for both directions.

4.5 DC operation

When powered on the decoder will check if DCC signal is present on the track and execute the received commands. If DC voltage is present on the track for more than a timeout period, the decoder will switch to analog mode and will turn on the functions configured in CV13 and CV14. The timeout period is set in CV11, and it is equal to CV11 decimal value * 8ms. The maximal value is 2,048s.

CV12 and CV29 are influencing the decoder behavior related to protocols.

The meaning of CV12 bits is described at the table.

CV12 bit	Value	Working mode (protocol)
0	0	DC mode OFF
	1	DC mode ON
2	0	DCC mode(protocol) OFF
	4	DCC mode(protocol) ON

CV29 bit 2 will turn off the analog conversion if it is set to 0 so the Function Decoder will not switch to analog mode when digital communication is off. For safety reasons, even if bit 0 of CV12 is set to 0 (DCC mode = OFF), CV12 can still be modified over DCC. The DCC programming commands are executed even if DCC digital mode is turned off.

By default the DC analog and DCC operations are activated.

F0 is configured for both travel directions (Bit 2 set in CV29 and CV14 activated for the use of the F0 function in both directions. $CV14 = 1 + 2 = 3$). When the circuit board is connected to a DC analog voltage, all outputs of the function decoder are switched on.

To switch on other functions in analog DC voltage operation, they must be defined in CV13 and CV14. The mapping of the function must be defined in advance according to the steps described in the table below. So only the functions F0, and F1-F14 can be used in DC operation.

	F0 fwd.	F0 rev	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
bit	0	1	0	1	2	3	4	5	6	7	2	3	4	5	6	7
CV13			1	2	4	8	16	32	64	128						
CV14	1	2									4	8	16	32	64	128

4.6 Onboard power pack (SPP)

The on board power pack (SPP) offers up to 20 seconds of energy storage. The power pack needs about 30-40 seconds for a full charge, but after 20 seconds of power connection it will already be usable.

The switching off time of the energy buffering is set in CV168 (SPP timeout). This CV contains the value of the light remanence time after the power is removed. The factory default value is 62. One unit is 80 milliseconds. This means, that the factory default timeout is $62 * 0.080 = 4.96 \sim 5$ seconds. After the electronics are disconnected from the track, the lights will be still on for about 5 seconds. After this time, they will be switched off automatically.

The maximal value for CV168 is 255, which is roughly equivalent to 20 seconds. This means, that the lights can be on for 20 seconds after the track voltage is removed. This interior lighting board is also working also in analog DC. The value set in CV168 is valid also for analog DC operation.

4.7 User Data

CV105 and CV106 are two CVs that can be used to save user identifiers (serial number, etc.). The particularity of these two CVs is that after a reset their contents will not be erased.

4.8 Other functions

Last command safe

The OnBoard decoder has a feature to save the last function command received. This feature can be activated programming the value 1 in CV152. With this feature activated, the decoder will start up activating the functions that were active before power interruption, even if DCC commands were not received to activate these functions.

DCC quality

The DCC Signal Quality Indicator (QoS = Quality of Signal) is saved in CV108 as percentage (in the range 0-100%). The lowest QoS value detected by the decoder from the last reading is written in CV109.

CV169 contains the function number which activation will trigger the saving of the current QoS values into the decoder non-volatile memory (EEPROM). Saving the instantaneous values is done by activating (turn ON and turn OFF) this function from the Command Station (or tOm Programmer). In CV169 the number of the function must be entered (for example for function F28, the value 28 must be entered in CV169).

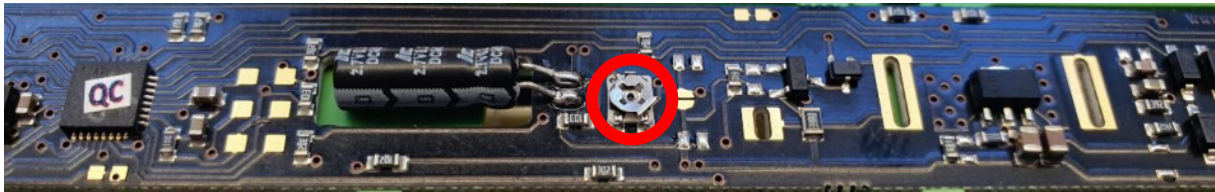
Without activating the function number given in CV169 (turn On, then Off), the values in the corresponding CVs will not be updated!

5. Effects for function outputs

5.1 Light intensity

The light intensity of the LEDs connected to the outputs of the onboard decoder can be changed individually by changing the values of CV120 to CV135. The factory default value for each of them is the decimal value 255 (maximal intensity). The 255 value in any of these CVs, result in a continuous output at maximum intensity level of the corresponding output.

The trimmer potentiometer works both in analogue and digital mode as a global light intensity. In digital mode, each output intensity can be trimmed in the CVs. The trimmer potentiometer has a global influence over the individual CV settings (it is a weighting function).



There are several effects available (which can be extended with further firmware upgrades) their parameters are globally defined in CVs 112-117.

In CV170 the PWM signal frequency used for all outputs is set. The default value is 2 (500 Hz).

CV170 value	Output PWM frequency
0	125 Hz
1	250 Hz
2	500 Hz

5.2 Light effects

The effects are configured for each of the outputs in CVs 136-152. The CVs must be programmed with the following values:

- 0 - Continuous outputs
- 1 - Fade effect
- 2 - Fluorescent (neon) effect
- 3 - Flickering effect
- 4 - Defective Neon effects
- 32 - Turn On Delay
- 64 - Turn Off Delay
- 128 - Random sequence generator function

The modules are delivered with the outputs configured for fade operation (CV136 till CV152 are set to value 1).

5.2.1 Fade effect

CV112 and CV113 define the progressive On and Off time if the Fade effect is used.

value 1 = 8ms, 15 = 120ms, 125 = 1000ms

5.2.2 Fluorescent (neon) effect

In CV114 the On delay is defined for the neon effect. It can be set from fast (value 0) to slow delay (value 7).

5.2.3 Flickering effect

In CV116 the flickering period is given for the flickering lamp effect. It can be set from fast (value 0) to slow flickering (value 7).

5.2.4 Defective Neon effects

The Defective Neon effects repetition time can be set in CV117. It can be set from fast (value 0) to slow repetition (value 7)

5.2.5 Turn On/Off Delay

All outputs can be turned on and/or off with a delay specified in CV111 (turn on delay) and CV110 (turn off delay). These delays are common for all outputs and can be set in 8ms steps. The maximal possible delay value is $8 \cdot 255 = 2040$ ms, approximately 2 second.

The switching on and off for each of the effects applied to an output can be executed with a delay. In the case of using the delay option on a certain output, the output will be activated after receiving the activation command (Function) with a delay time set in CV111, and will be deactivated after receiving the deactivation command (Function) with a delay time set in CV110.

5.2.6 Random sequence generator function

If you are activating the random sequence generator function for an output, it will switch on and off without the need of switching on and off a function.

To select an output for the random sequence, the value 128 (bit 7) of the individual output effect CVs must be set.

The switching period of the random sequence is defined in CV115 in the range of 1-255 seconds. As a result, after each elapse of the random period, the state of the outputs which have the random effect activated will change to a new on/off state, which is random based. It gives a very realistic effect to a carriage, when rolling on the track, and some of the compartment lights are switched on, or off.

The number of random states increases dramatically with the increase of the outputs selected for this function. We recommend the use of 2-4 of the outputs with the random sequence to get an efficient visual effect.

6. CV table

In the table on the following pages are listed all the CVs of the Decoder. We recommend that you change the CV values only if you are sure of their function and the impact of your action. Incorrect CV settings can negatively affect the performance of the ECU or cause incorrect responses to the commands transmitted from the command station.

The "Factory Default Values" column contains the "factory" value of the CVs (after a decoder reset, all CVs will have the value in this column), the column "Value Range" contains the range of usable values for each CV and the "Description" column contains the name (if there is an established name) and information about the CV function.

CV	Default value	Value Range	Description
1	3	0-127	Decoder Address Short, 7 bits
7	4	-	Software Version (only readable)
8	78	-	Manufactured ID/RESET (readable 78 = train-O-matic. Write any other value to reset the decoder to the factory default values
11	25	0-255	Packet Timeout
12	5	0-5	Working Mode
13	0	0-255	Analog Mode, Alternate Mode Function Status F1 ÷ F8 Bit 0 = 0(0): F1 not active in Analog mode = 1(1): F1 active in Analog mode Bit 1 = 0(0): F2 not active in Analog mode = 1(2): F2 active in Analog mode Bit 2 = 0(0): F3 not active in Analog mode = 1(4): F3 active in Analog mode Bit 3 = 0(0): F4 not active in Analog mode = 1(8): F4 active in Analog mode Bit 4 = 0(0): F5 not active in Analog mode = 1(16): F5 active in Analog mode Bit 5 = 0(0): F6 not active in Analog mode = 1(32): F6 active in Analog mode Bit 6 = 0(0): F7 not active in Analog mode = 1(64): F7 active in Analog mode Bit 7 = 0(0): F8 not active in Analog mode = 1(255): F8 active in Analog mode
14	3	0-255	Analog Mode, Alternate Mode Function. Status F0f, F0r, F9 ÷ F14, Bit 0 = 0(0): F0 forward not active in Analog mode = 1(1): F0 forward active in Analog mode Bit 1 = 0(0): F0 revers not active in Analog mode = 1(2): F0 revers active in Analog mode Bit 2 = 0(0): F9 not active in Analog mode = 1(4): F9 active in Analog mode Bit 3 = 0(0): F10 not active in Analog mode = 1(8): F10 active in Analog mode Bit 4 = 0(0): F11 not active in Analog mode = 1(16): F11 active in Analog mode Bit 5 = 0(0): F12 not active in Analog mode = 1(32): F12 active in Analog mode Bit 6 = 0(0): F13 not active in Analog mode = 1(64) F13 active in Analog mode Bit 7 = 0(0): F14 not active in Analog mode = 1(255): F14 active in Analog mode
15	0	0-7	Lock Value: Enter the value to match Lock ID in CV16 to unlock CV programming. No action and ACK will be performed by the decoder when Lock Value is different from LockID. In this situation only CV15 write is allowed.
16	0	0-7	LockID: To prevent accidental programming use unique ID number for decoders with same address (0..7) 1-loco decoder, 2-sound decoder, 3-function decoder, ...

17	192	192-255	Extended Address, Address High
18	3	0-255	Extended Address, Address Low
19	0	0-127	Consist Address If CV #19 > 0: Speed and direction is governed by this consist address (not the individual address in CV #1 or #17+18); functions are controlled by either the consist address or individual address, see CV's #21 + 22.
21	0	0-255	Functions defined here will be controlled by the consist address. Bit 0 = 0(0): F1 controlled by individual address = 1(1): by consist address Bit 1 = 0(0): F2 controlled by individual address = 1(2): by consist address Bit 2 = 0(0): F3 controlled by individual address = 1(4): by consist address Bit 3 = 0(0): F4 controlled by individual address = 1(8): by consist address Bit 4 = 0(0): F5 controlled by individual address = 1(16): by consist address Bit 5 = 0(0): F6 controlled by individual address = 1(32): by consist address Bit 6 = 0(0): F7 controlled by individual address = 1(64): by consist address Bit 7 = 0(0): F8 controlled by individual address = 1(255): by consist address
22	3= 1+ 2	0-63	Functions defined here will be controlled by the consist address. Bit 0 = 0(0): F0 (forw.) controlled by individual address = 1(1): by consist address Bit 1 = 0 (0): F0 (rev.) controlled by individual address = 1(2): by consist address Bit 2 = 0(0): F9 controlled by individual address = 1(4): by consist address Bit 3 = 0(0): F10 controlled by individual address = 1(8): by consist address Bit 4 = 0(0): F11 controlled by individual address = 1(16): by consist address Bit 5 = 0(0): F12 controlled by individual address = 1(32): by consist address
29	6= 2+ 4	0-63	Configuration Data Bit 0 = 0(0): Locomotive Direction normal = 1(1): Locomotive Direction reversed Bit 1 - Bit 2 = 0(0): Power Source Conversion NMRA Digital Only (only DCC) = 1(4): Power Source Conversion Enabled (DC + DCC) Bit 3 - Bit 4 - Bit 5 = 0(0): One byte addressing (short addressing) = 1(32): Two byte addressing (extended/long addressing) Bit 6 - Bit 7 -
30	0	0/1	Error CV. If the read out value is "1", an overcurrent event occurred since the last reset. The value can be cleared with programming "0" to CV30

33	255= 1+ 2+ 4+ 8+ 16+ 32+ 64+ 128	0-255	F0, Forward move mapping, low byte Bit 0 = 0(0): Out1 not active on F0 forward = 1(1): Out1 active on F0 forward Bit 1 = 0(0): Out2 not active on F0 forward = 1(2): Out2 active on F0 forward Bit 2 = 0(0): Out3 not active on F0 forward = 1(4): Out3 active on F0 forward Bit 3 = 0(0): Out4 not active on F0 forward = 1(8): Out4 active on F0 forward Bit 4 = 0(0): Out5 not active on F0 forward = 1(16): Out5 active on F0 forward Bit 5 = 0(0): Out6 not active on F0 forward = 1(32): Out6 active on F0 forward Bit 6 = 0(0): Out7 not active on F0 forward = 1(64): Out7 active on F0 forward Bit 7 = 0(0): Out8 not active on F0 forward = 1(128): Out8 active on F0 forward
34	255= 1+ 2+ 4+ 8+ 16+ 32+ 64+ 128	0-255	F0, Forward move mapping, high byte Bit 0 = 0(0): Out9 not active on F0 forward = 1(1): Out9 active on F0 forward Bit 1 = 0(0): Out10 not active on F0 forward = 1(2): Out10 active on F0 forward Bit 2 = 0(0): Out11 not active on F0 forward = 1(4): Out11 active on F0 forward Bit 3 = 0(0): Out12 not active on F0 forward = 1(8): Out12 active on F0 forward Bit 4 = 0(0): Out13 not active on F0 forward = 1(16): Out13 active on F0 forward Bit 5 = 0(0): Out14 not active on F0 forward = 1(32): Out14 active on F0 forward Bit 6 = 0(0): Out15 not active on F0 forward = 1(64): Out15 active on F0 forward Bit 7 = 0(0): Out16 not active on F0 forward = 1(128): Out16 active on F0 forward
35	255= 1+ 2+ 4+ 8+ 16+ 32+ 64+ 128	0-255	F0, Backward move mapping, low byte Bit 0 = 0(0): Out1 not active on F0 backward = 1(1): Out1 active on F0 backward Bit 1 = 0(0): Out2 not active on F0 backward = 1(2): Out2 active on F0 backward Bit 2 = 0(0): Out3 not active on F0 backward = 1(4): Out3 active on F0 backward Bit 3 = 0(0): Out4 not active on F0 backward = 1(8): Out4 active on F0 backward Bit 4 = 0(0): Out5 not active on F0 backward = 1(16): Out5 active on F0 backward Bit 5 = 0(0): Out6 not active on F0 backward = 1(32): Out6 active on F0 backward Bit 6 = 0(0): Out7 not active on F0 backward = 1(64): Out7 active on F0 backward Bit 7 = 0(0): Out8 not active on F0 backward = 1(128): Out8 active on F0 backward

36	255= 1+ 2+ 4+ 8+ 16+ 32+ 64+ 128	0-255	F0, Backward move mapping, high byte Bit 0 = 0(0): Out9 not active on F0 backward = 1(1): Out9 active on F0 backward Bit 1 = 0(0): Out10 not active on F0 backward = 1(2): Out10 active on F0 backward Bit 2 = 0(0): Out11 not active on F0 backward = 1(4): Out11 active on F0 backward Bit 3 = 0(0): Out12 not active on F0 backward = 1(8): Out12 active on F0 backward Bit 4 = 0(0): Out13 not active on F0 backward = 1(16): Out13 active on F0 backward Bit 5 = 0(0): Out14 not active on F0 backward = 1(32): Out14 active on F0 backward Bit 6 = 0(0): Out15 not active on F0 backward = 1(64): Out15 active on F0 backward Bit 7 = 0(0): Out16 not active on F0 backward = 1(128): Out16 active on F0 backward
37	1= 1	0-255	F1, Forward move mapping, low byte Bit 0 = 0(0): Out1 not active on F1 forward = 1(1): Out1 active on F1 forward Bit 1 = 0(0): Out2 not active on F1 forward = 1(2): Out2 active on F1 forward Bit 2 = 0(0): Out3 not active on F1 forward = 1(4): Out3 active on F1 forward Bit 3 = 0(0): Out4 not active on F1 forward = 1(8): Out4 active on F1 forward Bit 4 = 0(0): Out5 not active on F1 forward = 1(16): Out5 active on F1 forward Bit 5 = 0(0): Out6 not active on F1 forward = 1(32): Out6 active on F1 forward Bit 6 = 0(0): Out7 not active on F1 forward = 1(64): Out7 active on F1 forward Bit 7 = 0(0): Out8 not active on F1 forward = 1(128): Out8 active on F1 forward
38	0	0-255	F1, Forward move mapping, high byte Bit 0 = 0(0): Out9 not active on F1 forward = 1(1): Out9 active on F1 forward Bit 1 = 0(0): Out10 not active on F1 forward = 1(2): Out10 active on F1 forward Bit 2 = 0(0): Out11 not active on F1 forward = 1(4): Out11 active on F1 forward Bit 3 = 0(0): Out12 not active on F1 forward = 1(8): Out12 active on F1 forward Bit 4 = 0(0): Out13 not active on F1 forward = 1(16): Out13 active on F1 forward Bit 5 = 0(0): Out14 not active on F1 forward = 1(32): Out14 active on F1 forward Bit 6 = 0(0): Out15 not active on F1 forward = 1(64): Out15 active on F1 forward Bit 7 = 0(0): Out16 not active on F1 forward = 1(128): Out16 active on F1 forward

39	1= 1	0-255	F1, Backward move mapping, low byte Bit 0 = 0(0): Out1 not active on F1 backward = 1(1): Out1 active on F1 backward Bit 1 = 0(0): Out2 not active on F1 backward = 1(2): Out2 active on F1 backward Bit 2 = 0(0): Out3 not active on F1 backward = 1(4): Out3 active on F1 backward Bit 3 = 0(0): Out4 not active on F1 backward = 1(8): Out4 active on F1 backward Bit 4 = 0(0): Out5 not active on F1 backward = 1(16): Out5 active on F1 backward Bit 5 = 0(0): Out6 not active on F1 backward = 1(32): Out6 active on F1 backward Bit 6 = 0(0): Out7 not active on F1 backward = 1(64): Out7 active on F1 backward Bit 7 = 0(0): Out8 not active on F1 backward = 1(128): Out8 active on F1 backward
40	0	0-255	F1, Backward move mapping, high byte Bit 0 = 0(0): Out9 not active on F1 backward = 1(1): Out9 active on F1 backward Bit 1 = 0(0): Out10 not active on F1 backward = 1(2): Out10 active on F1 backward Bit 2 = 0(0): Out11 not active on F1 backward = 1(4): Out11 active on F1 backward Bit 3 = 0(0): Out12 not active on F1 backward = 1(8): Out12 active on F1 backward Bit 4 = 0(0): Out13 not active on F1 backward = 1(16): Out13 active on F1 backward Bit 5 = 0(0): Out14 not active on F1 backward = 1(32): Out14 active on F1 backward Bit 6 = 0(0): Out15 not active on F1 backward = 1(64): Out15 active on F1 backward Bit 7 = 0(0): Out16 not active on F1 backward = 1(128): Out16 active on F1 backward
41	2= 2	0-255	F2 mapping, low byte Bit 0 = 0(0): Out1 not active on F2 = 1(1): Out1 active on F2 Bit 1 = 0(0): Out2 not active on F2 = 1(2): Out2 active on F2 Bit 2 = 0(0): Out3 not active on F2 = 1(4): Out3 active on F2 Bit 3 = 0(0): Out4 not active on F2 = 1(8): Out4 active on F2 Bit 4 = 0(0): Out5 not active on F2 = 1(16): Out5 active on F2 Bit 5 = 0(0): Out6 not active on F2 = 1(32): Out6 active on F2 Bit 6 = 0(0): Out7 not active on F2 = 1(64): Out7 active on F2 Bit 7 = 0(0): Out8 not active on F2 = 1(128): Out8 active on F2

42	0	0-255	F2 mapping, high byte Bit 0 = 0(0): Out9 not active on F2 = 1(1): Out9 active on F2 Bit 1 = 0(0): Out10 not active on F2 = 1(2): Out10 active on F2 Bit 2 = 0(0): Out11 not active on F2 = 1(4): Out11 active on F2 Bit 3 = 0(0): Out12 not active on F2 = 1(8): Out12 active on F2 Bit 4 = 0(0): Out13 not active on F2 = 1(16): Out13 active on F2 Bit 5 = 0(0): Out14 not active on F2 = 1(32): Out14 active on F2 Bit 6 = 0(0): Out15 not active on F2 = 1(64): Out15 active on F2 Bit 7 = 0(0): Out16 not active on F2 = 1(128): Out16 active on F2
43	4= 4	0-255	F3 mapping, low byte Bit 0 = 0(0): Out1 not active on F3 = 1(1): Out1 active on F3 Bit 1 = 0(0): Out2 not active on F3 = 1(2): Out2 active on F3 Bit 2 = 0(0): Out3 not active on F3 = 1(4): Out3 active on F3 Bit 3 = 0(0): Out4 not active on F3 = 1(8): Out4 active on F3 Bit 4 = 0(0): Out5 not active on F3 = 1(16): Out5 active on F3 Bit 5 = 0(0): Out6 not active on F3 = 1(32): Out6 active on F3 Bit 6 = 0(0): Out7 not active on F3 = 1(64): Out7 active on F3 Bit 7 = 0(0): Out8 not active on F3 = 1(128): Out8 active on F3
44	0	0-255	F3 mapping, high byte Bit 0 = 0(0): Out9 not active on F3 = 1(1): Out9 active on F3 Bit 1 = 0(0): Out10 not active on F3 = 1(2): Out10 active on F3 Bit 2 = 0(0): Out11 not active on F3 = 1(4): Out11 active on F3 Bit 3 = 0(0): Out12 not active on F3 = 1(8): Out12 active on F3 Bit 4 = 0(0): Out13 not active on F3 = 1(16): Out13 active on F3 Bit 5 = 0(0): Out14 not active on F3 = 1(32): Out14 active on F3 Bit 6 = 0(0): Out15 not active on F3 = 1(64): Out15 active on F3 Bit 7 = 0(0): Out16 not active on F3 = 1(128): Out16 active on F3

45	8= 8	0-255	F4 mapping, low byte Bit 0 = 0(0): Out1 not active on F4 = 1(1): Out1 active on F4 Bit 1 = 0(0): Out2 not active on F4 = 1(2): Out2 active on F4 Bit 2 = 0(0): Out3 not active on F4 = 1(4): Out3 active on F4 Bit 3 = 0(0): Out4 not active on F4 = 1(8): Out4 active on F4 Bit 4 = 0(0): Out5 not active on F4 = 1(16): Out5 active on F4 Bit 5 = 0(0): Out6 not active on F4 = 1(32): Out6 active on F4 Bit 6 = 0(0): Out7 not active on F4 = 1(64): Out7 active on F4 Bit 7 = 0(0): Out8 not active on F4 = 1(128): Out8 active on F4
46	0	0-255	F4 mapping, high byte Bit 0 = 0(0): Out9 not active on F4 = 1(1): Out9 active on F4 Bit 1 = 0(0): Out10 not active on F4 = 1(2): Out10 active on F4 Bit 2 = 0(0): Out11 not active on F4 = 1(4): Out11 active on F4 Bit 3 = 0(0): Out12 not active on F4 = 1(8): Out12 active on F4 Bit 4 = 0(0): Out13 not active on F4 = 1(16): Out13 active on F4 Bit 5 = 0(0): Out14 not active on F4 = 1(32): Out14 active on F4 Bit 6 = 0(0): Out15 not active on F4 = 1(64): Out15 active on F4 Bit 7 = 0(0): Out16 not active on F4 = 1(128): Out16 active on F4
47	16= 16	0-255	F5 mapping, low byte Bit 0 = 0(0): Out1 not active on F5 = 1(1): Out1 active on F5 Bit 1 = 0(0): Out2 not active on F5 = 1(2): Out2 active on F5 Bit 2 = 0(0): Out3 not active on F5 = 1(4): Out3 active on F5 Bit 3 = 0(0): Out4 not active on F5 = 1(8): Out4 active on F5 Bit 4 = 0(0): Out5 not active on F5 = 1(16): Out5 active on F5 Bit 5 = 0(0): Out6 not active on F5 = 1(32): Out6 active on F5 Bit 6 = 0(0): Out7 not active on F5 = 1(64): Out7 active on F5 Bit 7 = 0(0): Out8 not active on F5 = 1(128): Out8 active on F5

48	0	0-255	F5 mapping, high byte Bit 0 = 0(0): Out9 not active on F5 = 1(1): Out9 active on F5 Bit 1 = 0(0): Out10 not active on F5 = 1(2): Out10 active on F5 Bit 2 = 0(0): Out11 not active on F5 = 1(4): Out11 active on F5 Bit 3 = 0(0): Out12 not active on F5 = 1(8): Out12 active on F5 Bit 4 = 0(0): Out13 not active on F5 = 1(16): Out13 active on F5 Bit 5 = 0(0): Out14 not active on F5 = 1(32): Out14 active on F5 Bit 6 = 0(0): Out15 not active on F5 = 1(64): Out15 active on F5 Bit 7 = 0(0): Out16 not active on F5 = 1(128): Out16 active on F5
49	32= 32	0-255	F6 mapping, low byte Bit 0 = 0(0): Out1 not active on F6 = 1(1): Out1 active on F6 Bit 1 = 0(0): Out2 not active on F6 = 1(2): Out2 active on F6 Bit 2 = 0(0): Out3 not active on F6 = 1(4): Out3 active on F6 Bit 3 = 0(0): Out4 not active on F6 = 1(8): Out4 active on F6 Bit 4 = 0(0): Out5 not active on F6 = 1(16): Out5 active on F6 Bit 5 = 0(0): Out6 not active on F6 = 1(32): Out6 active on F6 Bit 6 = 0(0): Out7 not active on F6 = 1(64): Out7 active on F6 Bit 7 = 0(0): Out8 not active on F6 = 1(128): Out8 active on F6
50	0	0-255	F6 mapping, high byte Bit 0 = 0(0): Out9 not active on F6 = 1(1): Out9 active on F6 Bit 1 = 0(0): Out10 not active on F6 = 1(2): Out10 active on F6 Bit 2 = 0(0): Out11 not active on F6 = 1(4): Out11 active on F6 Bit 3 = 0(0): Out12 not active on F6 = 1(8): Out12 active on F6 Bit 4 = 0(0): Out13 not active on F6 = 1(16): Out13 active on F6 Bit 5 = 0(0): Out14 not active on F6 = 1(32): Out14 active on F6 Bit 6 = 0(0): Out15 not active on F6 = 1(64): Out15 active on F6 Bit 7 = 0(0): Out16 not active on F6 = 1(128): Out16 active on F6

54	0	0-255	F8 mapping, high byte Bit 0 = 0(0): Out9 not active on F8 = 1(1): Out9 active on F8 Bit 1 = 0(0): Out10 not active on F8 = 1(2): Out10 active on F8 Bit 2 = 0(0): Out11 not active on F8 = 1(4): Out11 active on F8 Bit 3 = 0(0): Out12 not active on F8 = 1(8): Out12 active on F8 Bit 4 = 0(0): Out13 not active on F8 = 1(16): Out13 active on F8 Bit 5 = 0(0): Out14 not active on F8 = 1(32): Out14 active on F8 Bit 6 = 0(0): Out15 not active on F8 = 1(64): Out15 active on F8 Bit 7 = 0(0): Out16 not active on F8 = 1(128): Out16 active on F8
55	0	0-255	F9 mapping, low byte Bit 0 = 0(0): Out1 not active on F9 = 1(1): Out1 active on F9 Bit 1 = 0(0): Out2 not active on F9 = 1(2): Out2 active on F9 Bit 2 = 0(0): Out3 not active on F9 = 1(4): Out3 active on F9 Bit 3 = 0(0): Out4 not active on F9 = 1(8): Out4 active on F9 Bit 4 = 0(0): Out5 not active on F9 = 1(16): Out5 active on F9 Bit 5 = 0(0): Out6 not active on F9 = 1(32): Out6 active on F9 Bit 6 = 0(0): Out7 not active on F9 = 1(64): Out7 active on F9 Bit 7 = 0(0): Out8 not active on F9 = 1(128): Out8 active on F9
56	1= 1	0-255	F9 mapping, high byte Bit 0 = 0(0): Out9 not active on F9 = 1(1): Out9 active on F9 Bit 1 = 0(0): Out10 not active on F9 = 1(2): Out10 active on F9 Bit 2 = 0(0): Out11 not active on F9 = 1(4): Out11 active on F9 Bit 3 = 0(0): Out12 not active on F9 = 1(8): Out12 active on F9 Bit 4 = 0(0): Out13 not active on F9 = 1(16): Out13 active on F9 Bit 5 = 0(0): Out14 not active on F9 = 1(32): Out14 active on F9 Bit 6 = 0(0): Out15 not active on F9 = 1(64): Out15 active on F9 Bit 7 = 0(0): Out16 not active on F9 = 1(128): Out16 active on F9

57	0	0-255	F10 mapping, low byte Bit 0 = 0(0): Out1 not active on F10 = 1(1): Out1 active on F10 Bit 1 = 0(0): Out2 not active on F10 = 1(2): Out2 active on F10 Bit 2 = 0(0): Out3 not active on F10 = 1(4): Out3 active on F10 Bit 3 = 0(0): Out4 not active on F10 = 1(8): Out4 active on F10 Bit 4 = 0(0): Out5 not active on F10 = 1(16): Out5 active on F10 Bit 5 = 0(0): Out6 not active on F10 = 1(32): Out6 active on F10 Bit 6 = 0(0): Out7 not active on F10 = 1(64): Out7 active on F10 Bit 7 = 0(0): Out8 not active on F10 = 1(128): Out8 active on F10
58	2= 2	0-255	F10 mapping, high byte Bit 0 = 0(0): Out9 not active on F10 = 1(1): Out9 active on F10 Bit 1 = 0(0): Out10 not active on F10 = 1(2): Out10 active on F10 Bit 2 = 0(0): Out11 not active on F10 = 1(4): Out11 active on F10 Bit 3 = 0(0): Out12 not active on F10 = 1(8): Out12 active on F10 Bit 4 = 0(0): Out13 not active on F10 = 1(16): Out13 active on F10 Bit 5 = 0(0): Out14 not active on F10 = 1(32): Out14 active on F10 Bit 6 = 0(0): Out15 not active on F10 = 1(64): Out15 active on F10 Bit 7 = 0(0): Out16 not active on F10 = 1(128): Out16 active on F10
59	0	0-255	F11 mapping, low byte Bit 0 = 0(0): Out1 not active on F11 = 1(1): Out1 active on F11 Bit 1 = 0(0): Out2 not active on F11 = 1(2): Out2 active on F11 Bit 2 = 0(0): Out3 not active on F11 = 1(4): Out3 active on F11 Bit 3 = 0(0): Out4 not active on F11 = 1(8): Out4 active on F11 Bit 4 = 0(0): Out5 not active on F11 = 1(16): Out5 active on F11 Bit 5 = 0(0): Out6 not active on F11 = 1(32): Out6 active on F11 Bit 6 = 0(0): Out7 not active on F11 = 1(64): Out7 active on F11 Bit 7 = 0(0): Out8 not active on F11 = 1(128): Out8 active on F11

60	4= 4	0-255	F11 mapping, high byte Bit 0 = 0(0): Out9 not active on F11 = 1(1): Out9 active on F11 Bit 1 = 0(0): Out10 not active on F11 = 1(2): Out10 active on F11 Bit 2 = 0(0): Out11 not active on F11 = 1(4): Out11 active on F11 Bit 3 = 0(0): Out12 not active on F11 = 1(8): Out12 active on F11 Bit 4 = 0(0): Out13 not active on F11 = 1(16): Out13 active on F11 Bit 5 = 0(0): Out14 not active on F11 = 1(32): Out14 active on F11 Bit 6 = 0(0): Out15 not active on F11 = 1(64): Out15 active on F11 Bit 7 = 0(0): Out16 not active on F11 = 1(128): Out16 active on F11
61	0	0-255	F12 mapping, low byte Bit 0 = 0(0): Out1 not active on F12 = 1(1): Out1 active on F12 Bit 1 = 0(0): Out2 not active on F12 = 1(2): Out2 active on F12 Bit 2 = 0(0): Out3 not active on F12 = 1(4): Out3 active on F12 Bit 3 = 0(0): Out4 not active on F12 = 1(8): Out4 active on F12 Bit 4 = 0(0): Out5 not active on F12 = 1(16): Out5 active on F12 Bit 5 = 0(0): Out6 not active on F12 = 1(32): Out6 active on F12 Bit 6 = 0(0): Out7 not active on F12 = 1(64): Out7 active on F12 Bit 7 = 0(0): Out8 not active on F12 = 1(128): Out8 active on F12
62	8= 8	0-255	F12 mapping, high byte Bit 0 = 0(0): Out9 not active on F12 = 1(1): Out9 active on F12 Bit 1 = 0(0): Out10 not active on F12 = 1(2): Out10 active on F12 Bit 2 = 0(0): Out11 not active on F12 = 1(4): Out11 active on F12 Bit 3 = 0(0): Out12 not active on F12 = 1(8): Out12 active on F12 Bit 4 = 0(0): Out13 not active on F12 = 1(16): Out13 active on F12 Bit 5 = 0(0): Out14 not active on F12 = 1(32): Out14 active on F12 Bit 6 = 0(0): Out15 not active on F12 = 1(64): Out15 active on F12 Bit 7 = 0(0): Out16 not active on F12 = 1(128): Out16 active on F12

105	0	0-255	USER data
106	0	0-255	USER data
108	0	0-100	Actual QoS (Quality of Service)
109	100	0-100	QoS minimal value
110	0	0-255	Output switching off delay time
111	0	0-255	Output switching on delay time
112	25	1-127	FadeIN AUX Light Effect Fade ON, ex.: 1 = 8ms, 15 = 120ms, 125 = 1000ms
113	15	1-127	FadeOUT AUX Light Effect Fade OFF
114	3	0-7	Delay, Flourescent Tube Start, Blinking Delay 1-8 delay step [0..7]
115	10	1-255	Random Time Period, 1s-255s
116	3	0-7	Flicker Period: Fast-Slow 0..7 val
117	3	0-7	Defective Neon effects repetition time, 0 fast repetition, 7 slow repetition
120	255	0-255	Out 1 Light intensity, [1-255]
121	255	0-255	Out 2 Light intensity, [1-255]
122	255	0-255	Out 3 Light intensity, [1-255]
123	255	0-255	Out 4 Light intensity, [1-255]
124	255	0-255	Out 5 Light intensity, [1-255]
125	255	0-255	Out 6 Light intensity, [1-255]
126	255	0-255	Out 7 Light intensity, [1-255]
127	255	0-255	Out 8 Light intensity, [1-255]
128	255	0-255	Out 9 Light intensity, [1-255]
129	255	0-255	Out 10 Light intensity, [1-255]
130	255	0-255	Out 11 Light intensity, [1-255]
131	255	0-255	Out 12 Light intensity, [1-255]
132	255	0-255	Out 13 Light intensity, [1-255]
133	255	0-255	Out 14 Light intensity, [1-255]
134	255	0-255	Out 15 Light intensity, [1-255]
135	255	0-255	Out 16 Light intensity, [1-255]
136	1	0-255	Out 1, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
137	1	0-255	Out 2, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function

138	1	0-255	Out 3, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
139	1	0-255	Out 4, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
140	1	0-255	Out 5, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
141	1	0-255	Out 6, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
142	1	0-255	Out 7, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
143	1	0-255	Out 8, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function

144	1	0-255	Out 9, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
145	1	0-255	Out 10, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
146	1	0-255	Out 11, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
147	1	0-255	Out 12, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
148	1	0-255	Out 13, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
149	1	0-255	Out 14, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function

150	1	0-255	Out 15, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
151	1	0-255	Out 16, Effect: 0 - Continuous outputs 1 - Fade effect 2 - Fluorescent (neon) effect 3 - Flickering effect 4 - Defective Neon effects 32 - Turn On Delay 64 - Turn Off Delay 128 - Random sequence generator function
152	0	0-1	Save Last State 1-Save 0-Don't Save
160	0	0-255	F13 mapping, low byte Bit 0 = 0(0): Out1 not active on F13 = 1(1): Out1 active on F13 Bit 1 = 0(0): Out2 not active on F13 = 1(2): Out2 active on F13 Bit 2 = 0(0): Out3 not active on F13 = 1(4): Out3 active on F13 Bit 3 = 0(0): Out4 not active on F13 = 1(8): Out4 active on F13 Bit 4 = 0(0): Out5 not active on F13 = 1(16): Out5 active on F13 Bit 5 = 0(0): Out6 not active on F13 = 1(32): Out6 active on F13 Bit 6 = 0(0): Out7 not active on F13 = 1(64): Out7 active on F13 Bit 7 = 0(0): Out8 not active on F13 = 1(128): Out8 active on F13

161	16= 16	0-255	F13 mapping, high byte Bit 0 = 0(0): Out9 not active on F13 = 1(1): Out9 active on F13 Bit 1 = 0(0): Out10 not active on F13 = 1(2): Out10 active on F13 Bit 2 = 0(0): Out11 not active on F13 = 1(4): Out11 active on F13 Bit 3 = 0(0): Out12 not active on F13 = 1(8): Out12 active on F13 Bit 4 = 0(0): Out13 not active on F13 = 1(16): Out13 active on F13 Bit 5 = 0(0): Out14 not active on F13 = 1(32): Out14 active on F13 Bit 6 = 0(0): Out15 not active on F13 = 1(64): Out15 active on F13 Bit 7 = 0(0): Out16 not active on F13 = 1(128): Out16 active on F13
162	0	0-255	F14 mapping, low byte Bit 0 = 0(0): Out1 not active on F14 = 1(1): Out1 active on F14 Bit 1 = 0(0): Out2 not active on F14 = 1(2): Out2 active on F14 Bit 2 = 0(0): Out3 not active on F14 = 1(4): Out3 active on F14 Bit 3 = 0(0): Out4 not active on F14 = 1(8): Out4 active on F14 Bit 4 = 0(0): Out5 not active on F14 = 1(16): Out5 active on F14 Bit 5 = 0(0): Out6 not active on F14 = 1(32): Out6 active on F14 Bit 6 = 0(0): Out7 not active on F14 = 1(64): Out7 active on F14 Bit 7 = 0(0): Out8 not active on F14 = 1(128): Out8 active on F14
163	32= 32	0-255	F14 mapping, high byte Bit 0 = 0(0): Out9 not active on F14 = 1(1): Out9 active on F14 Bit 1 = 0(0): Out10 not active on F14 = 1(2): Out10 active on F14 Bit 2 = 0(0): Out11 not active on F14 = 1(4): Out11 active on F14 Bit 3 = 0(0): Out12 not active on F14 = 1(8): Out12 active on F14 Bit 4 = 0(0): Out13 not active on F14 = 1(16): Out13 active on F14 Bit 5 = 0(0): Out14 not active on F14 = 1(32): Out14 active on F14 Bit 6 = 0(0): Out15 not active on F14 = 1(64): Out15 active on F14 Bit 7 = 0(0): Out16 not active on F14 = 1(128): Out16 active on F14

164	0	0-255	F15 mapping, low byte Bit 0 = 0(0): Out1 not active on F15 = 1(1): Out1 active on F15 Bit 1 = 0(0): Out2 not active on F15 = 1(2): Out2 active on F15 Bit 2 = 0(0): Out3 not active on F15 = 1(4): Out3 active on F15 Bit 3 = 0(0): Out4 not active on F15 = 1(8): Out4 active on F15 Bit 4 = 0(0): Out5 not active on F15 = 1(16): Out5 active on F15 Bit 5 = 0(0): Out6 not active on F15 = 1(32): Out6 active on F15 Bit 6 = 0(0): Out7 not active on F15 = 1(64): Out7 active on F15 Bit 7 = 0(0): Out8 not active on F15 = 1(128): Out8 active on F15
165	64= 64	0-255	F15 mapping, high byte Bit 0 = 0(0): Out9 not active on F15 = 1(1): Out9 active on F15 Bit 1 = 0(0): Out10 not active on F15 = 1(2): Out10 active on F15 Bit 2 = 0(0): Out11 not active on F15 = 1(4): Out11 active on F15 Bit 3 = 0(0): Out12 not active on F15 = 1(8): Out12 active on F15 Bit 4 = 0(0): Out13 not active on F15 = 1(16): Out13 active on F15 Bit 5 = 0(0): Out14 not active on F15 = 1(32): Out14 active on F15 Bit 6 = 0(0): Out15 not active on F15 = 1(64): Out15 active on F15 Bit 7 = 0(0): Out16 not active on F15 = 1(128): Out16 active on F15
166	0	0-255	F16 mapping, low byte Bit 0 = 0(0): Out1 not active on F16 = 1(1): Out1 active on F16 Bit 1 = 0(0): Out2 not active on F16 = 1(2): Out2 active on F16 Bit 2 = 0(0): Out3 not active on F16 = 1(4): Out3 active on F16 Bit 3 = 0(0): Out4 not active on F16 = 1(8): Out4 active on F16 Bit 4 = 0(0): Out5 not active on F16 = 1(16): Out5 active on F16 Bit 5 = 0(0): Out6 not active on F16 = 1(32): Out6 active on F16 Bit 6 = 0(0): Out7 not active on F16 = 1(64): Out7 active on F16 Bit 7 = 0(0): Out8 not active on F16 = 1(128): Out8 active on F16

167	128= 128	0-255	F16 mapping, high byte Bit 0 = 0(0): Out9 not active on F16 = 1(1): Out9 active on F16 Bit 1 = 0(0): Out10 not active on F16 = 1(2): Out10 active on F16 Bit 2 = 0(0): Out11 not active on F16 = 1(4): Out11 active on F16 Bit 3 = 0(0): Out12 not active on F16 = 1(8): Out12 active on F16 Bit 4 = 0(0): Out13 not active on F16 = 1(16): Out13 active on F16 Bit 5 = 0(0): Out14 not active on F16 = 1(32): Out14 active on F16 Bit 6 = 0(0): Out15 not active on F16 = 1(64): Out15 active on F16 Bit 7 = 0(0): Out16 not active on F16 = 1(128): Out16 active on F16
168	62	0-255	SPP Timeout. One step is 80 milliseconds. The factory default timeout is $62 * 0.080 = 4.96 \sim 5$ seconds. After the electronics are disconnected from the track the lights will be still on for 5 seconds.
169	28	0-28	Special function number for QoS saving (F28)
170	2	0-2	PWM Output frequency (0 = 125 Hz, 1 = 250 Hz, 2 = 500 Hz)