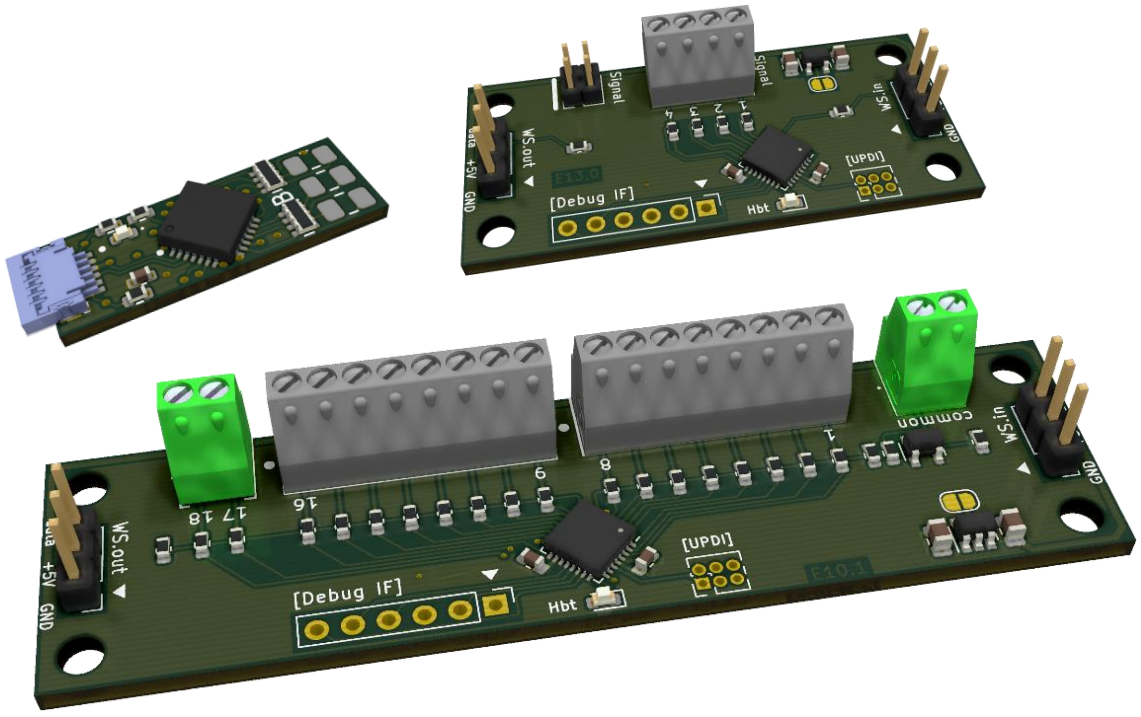
WS2811 Emulation

Anleitung

Frank Schumacher, 2025



Inhaltsverzeichnis

1 Revisionen 3
2 Einleitung 4
2.1 Allgemein 4
2.2 Eigenschaften 4
3 Einrichten 5
4 Module 6
4.1 E10 Modul (PWM) 7
4.2 E11 Modul (PWM) 8
4.3 E12 Modul (WS2811) 9
4.4 E13 Modul (Multiplex) 10
4.5 E14 Modul (MPX) 11
4.6 E15 Modul (Multiplex) 12
4.7 P18 Modul (Adapter) 13
5 Debug-IF 14
5.1 Konsole 14
5.2 Befehle 16
5.2.1 Break 16
5.2.2 Reset 16
5.2.3 Start Optionen auswählen 16
5.2.4 Modul starten 17
5.2.5 Modul in den Grundzustand setzen 18
5.2.6 LED-Erkennung durchführen 19
5.2.7 Feste LED-Spannung aktivieren 20
5.2.8 Variable LED-Spannung aktivieren 21
5.2.9 EEPROM Anzeigen 22
5.2.10 Firmware Update 23
5.2.11 Konfig Upload 24
6 Terminal Programme 25
6.1 TeraTerm 26
6.2 PuTTY 27
7 Bus Timing 28

1 Revisionen

2024/4/12	Initialer Draft
2024/4/14	Beschreibung P18 Adapter Modul
2024/12/06	Aktualisierung einiger Bilder
2025/03/22	Konfig upload Option

2 Einleitung

2.1 Allgemein

Die Exx-Module sind für den Betrieb an einem WS2811 Bus vorgesehen und können dabei native WS2811 Baugruppen ersetzen bzw. mit denen gemischt werden. Folgende Unterschiede sind zu beachten,

Exx Vorteile:

+ variable LED-Anzahl (optional)
+ dynamisch einstellbare LED-Spannung in den Grenzen [0…5V] (optional)
+ Multiplex Signal Anschluss (optional)
+ kurze LED-Funktionstest nach dem Einschalten
+ optische Fehlermeldung bei Störung der Datenleitung
+ Funktion anpassbar über Firmware

Exx Nachteile:

- maximal 5V LED-Spannung (keine LED-Serienschaltungen möglich)

2.2 Eigenschaften

Alle Module (Ausnahme E12) haben folgende gemeinsame Eigenschaften,

- Beim Einschalten blitzen die angeschlossenen LEDs einmal kurz auf. So kann einfach überprüft werden, ob alle angeschlossenen LEDs funktionsfähig sind.

- Wenn das Bus-Signal für länger als 5 Sekunden ausbleibt, blitzen alle LEDs im Sekunden Takt und signalisieren ein technisches Problem. Sollte das Bus-Signal innerhalb einer Minute nicht erneut anlaufen endet das Blitzen, um Modelbahnbetrieb nicht unnötig optisch zu stören.

- (optional) Automatische Erkennung angeschlossener LEDs.
  Stellt die Zahl der Bytes, die von dem Bus entnommen werden, auf das exakt notwendige Maß ein.

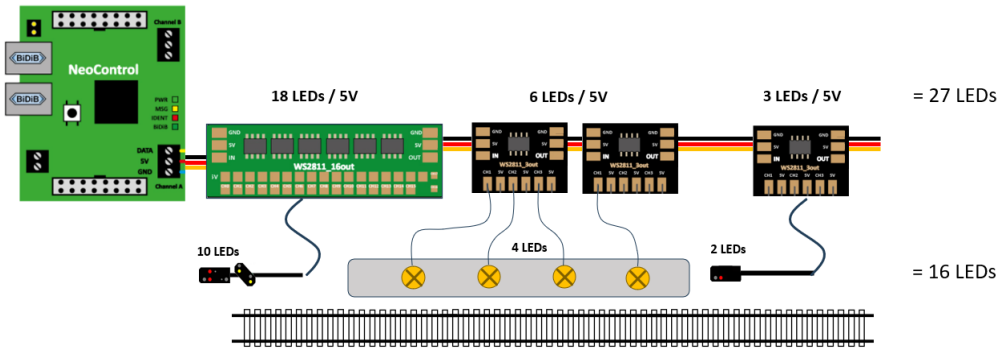
- (optional) Einstellung auf ein bestimmtes MPX-Signal mittels Konfig Datei.

- (optional) Die LED-Spannung ist dynamisch über Bus steuerbar um beispielsweise einen Tag/Nacht Betrieb der Modellbahn zu implementieren.

- Ein serielles Debug-IF ermöglicht
  - das Einstellen der oben genannten Optionen
  - Firmware update

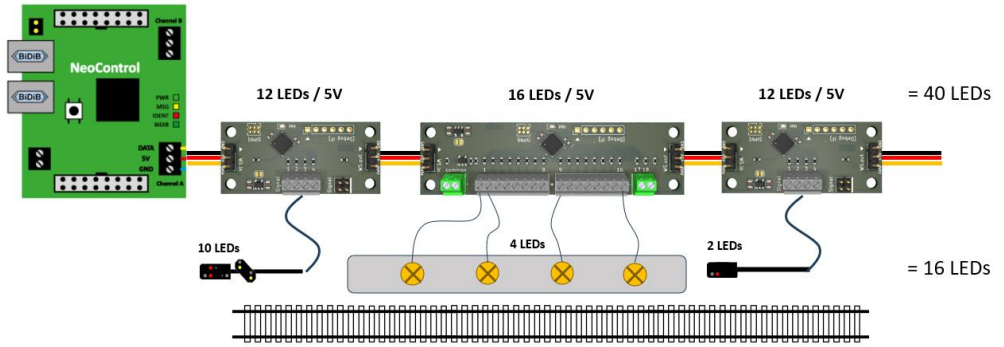
3 Einrichten

Initial: Aufbau mit nativen WS2811 Bausteinen. Diese Bausteine haben durch die verwendeten WS2811 Chips eine fest vorgegebene LED-Zahl.



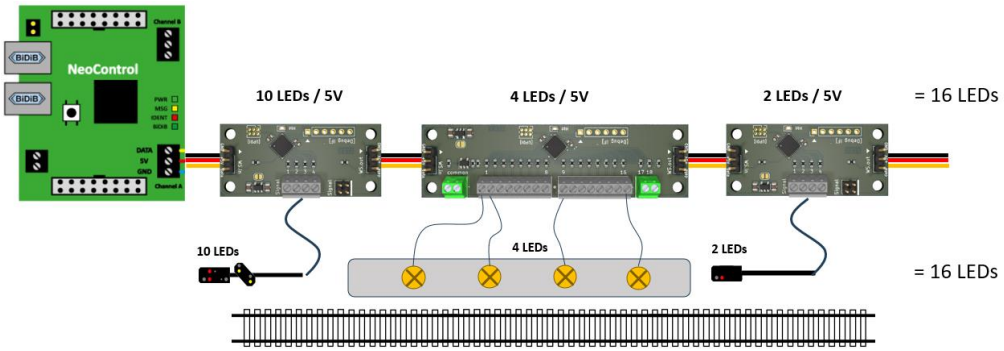
Leider entsteht oft ein Missverhältnis (27:16) zwischen den zu definierenden LEDs und den real angeschlossenen LEDs.

Umbau: Ersetzt man die nativen WS2811 Bausteine durch Exx-Module verschlechtert sich zunächst das Missverhältnis auf (40:16).



Anpassung: Die automatischen LED-Erkennung der Exx-Module kann das Missverhältnis beheben. Siehe dazu Kapitel „5.2.6 LED-Erkennung durchführen“.

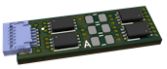
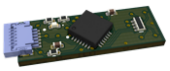
Ergebnis: Das Verhältnis ist nun stimmig (16:16).



Sollte sich das LED-Arrangement ändern, kann die automatischen LED-Erkennung (Kapitel 5.2.6) jederzeit erneut angestoßen werden.

4 Module

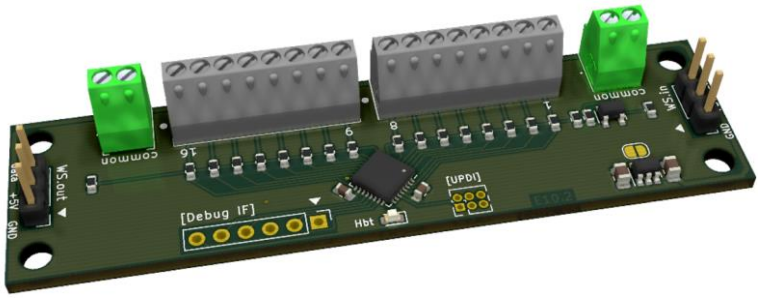
Es existieren sechs Modul Bauformen, optimiert für unterschiedliche Anwendungsbereiche.

 E10 – 16 LED PWM	 **E11 – 12 LED PWM (imbedding)**	 **E12 – 12 LED nativ WS2811 (imbedding)**
 E13 – 12 LED-Multiplex	 **E14 – 12 LED-Multiplex (imbedding)**	 **E15 – 12 LED-Multiplex (5V fix)**

Alle Module werden an dem WS2811 Bus betrieben. Die Module E11, E12, E14 sind zum Einbau direkt in den Signal Fuß vorgesehen. Das E12 Module verwendet native WS2811 Chips, alle anderen Module eine CPU mit Firmware.

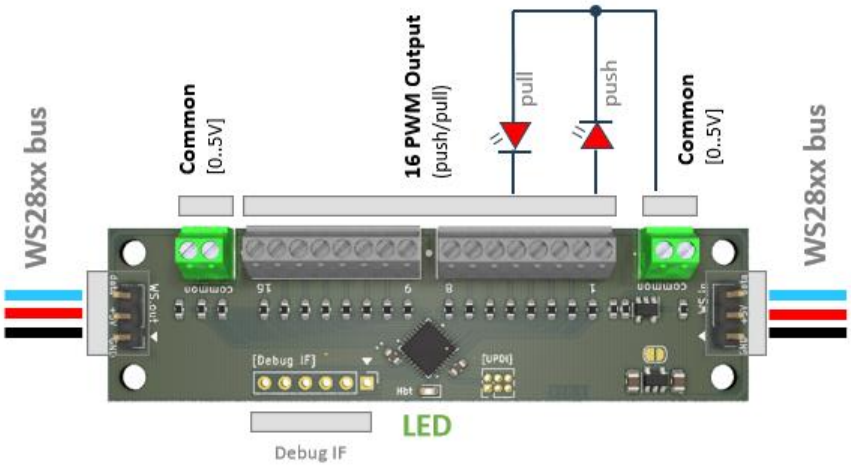
4.1 E10 Modul (PWM)

Es handelt sich um ein 16 Kanal LED-PWM Treiber mit WS2811 kompatiblen Bus Anschluss.



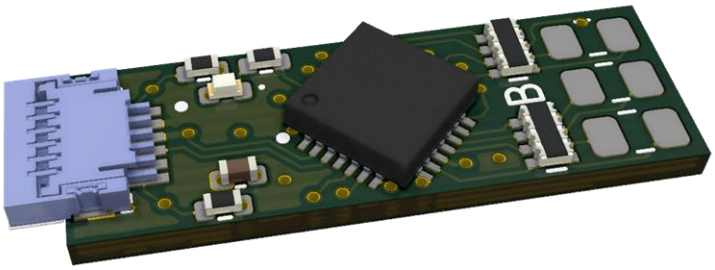
Modul Verbindungen

- **Bus-Anschluss**
  Die Verbinder links/rechts werden an den WS28xx Bus angeschlossen (rechts Eingang, links Ausgang)
- **LED-Anschluss**
  Die Schraubklemmen oben bilden die Anschlüsse der LEDs. **Wichtig**: Es darf keine externe Spannung verwendet werden. Alle LEDs müssen über *Common* gespeist werden.
  Hinweis: Die Hardware erlaubt alternativ den Push (Common Kathode) oder Pull (Common Anode) Betrieb. Derzeit ist nur der Pull-Modus implementiert.
- **Debug-IF**
  Dient der Konfiguration des Moduls mithilfe eines FTDI-Kabels (s. Kapitel 5).



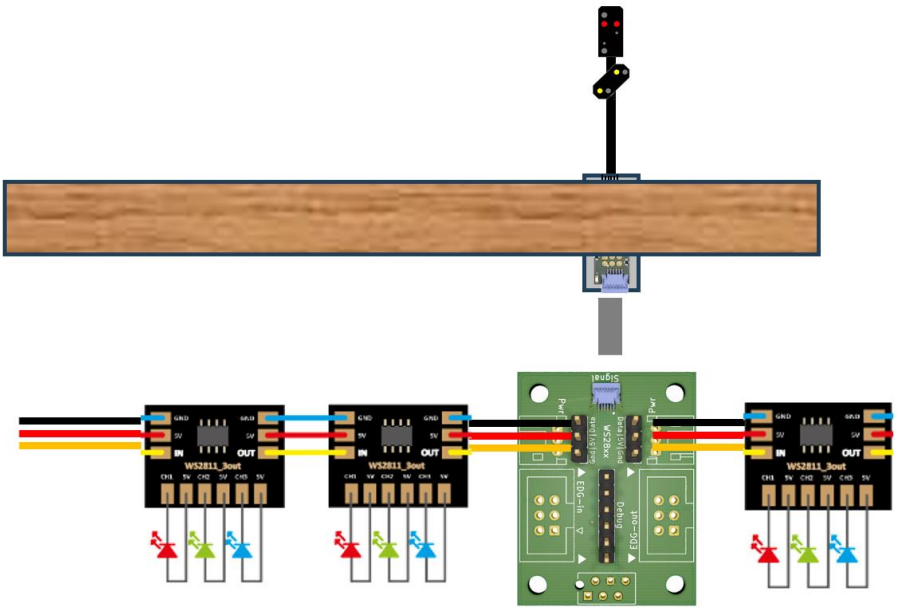
4.2 E11 Modul (PWM)

Es handelt sich um ein 12 Kanal LED-PWM Treiber mit WS2811 kompatiblen Bus Anschluss. Diese Bauform ist zum Einbau in den Signal Fuß vorgesehen.



Modul Verbindungen

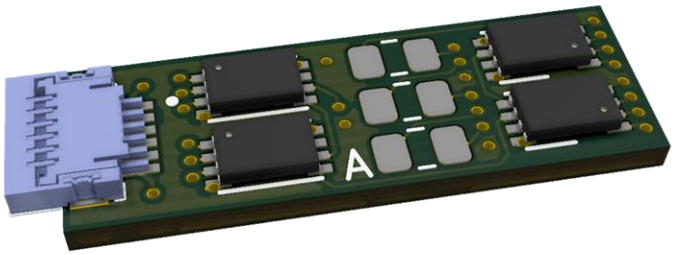
- **Bus-Anschluss**
  Nur über das FFC Adapter Modul (s. Bild).
- **LED-Anschluss**
  Die Löt-Pads bilden die Anschlüsse der LEDs.
- **Debug-IF**
  Dient der Konfiguration des Moduls mithilfe eines FTDI-Kabels (Kapitel 5).



> **Wichtig:** Das Modul darf nicht zeitgleich mit dem WS28xx Bus und dem Debug-IF verbunden sein. Bevor das FTDI-Kabel aufgesteckt wird, bitte die Verbindung zum WS28xx Bus trennen.

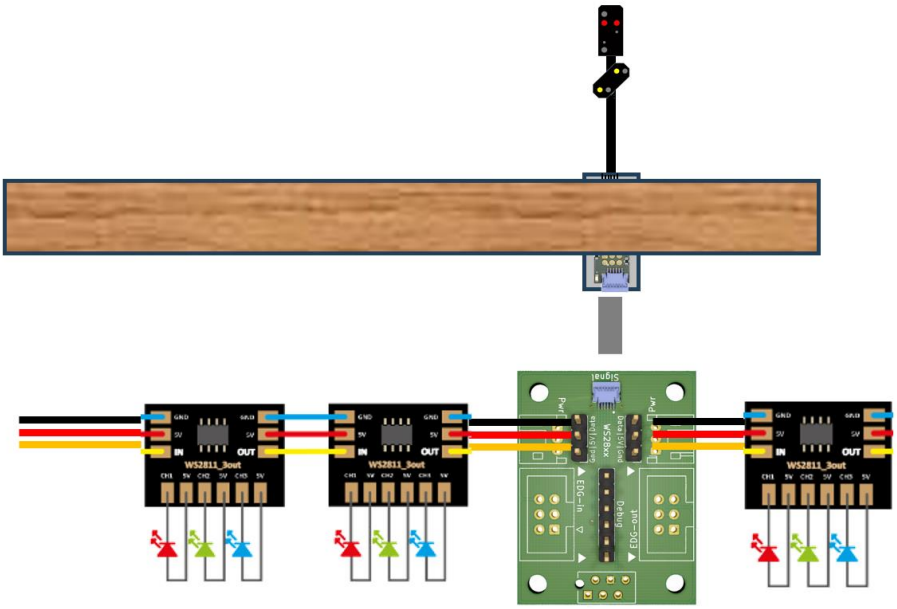
4.3 E12 Modul (WS2811)

Es handelt sich um ein 12 Kanal nativen WS2811 Baustein. Diese Bauform ist zum Einbau in den Signal Fuß vorgesehen.



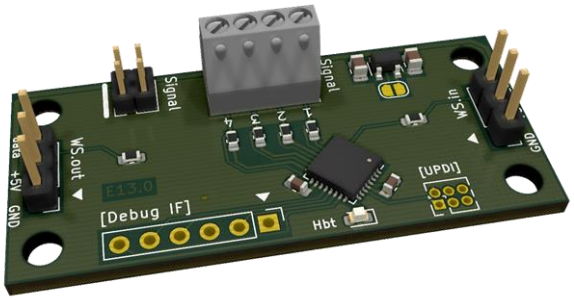
Modul Verbindungen

- **Bus-Anschluss**
  Nur über das FFC Adapter Modul (s. Bild).
- **LED-Anschluss**
  Die Löt-Pads bilden die Anschlüsse der LEDs.
- **Debug-IF**
  Nicht vorhanden (WS2811 native Chips).



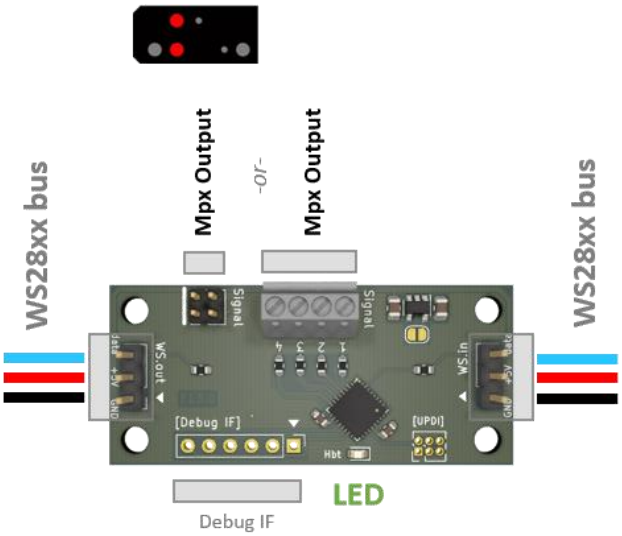
4.4 E13 Modul (Multiplex)

Es handelt sich um ein 12 Kanal Multiplex LED Treiber mit WS2811 kompatiblen Bus Anschluss.



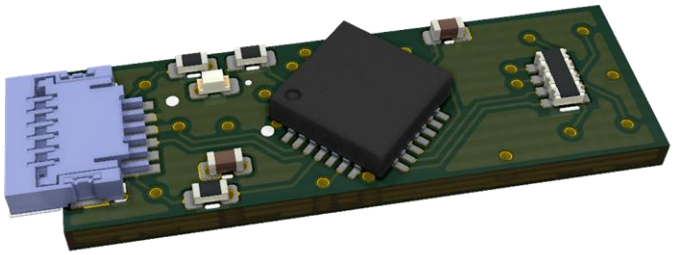
Modul Verbindungen

- **Bus-Anschluss**
  Die Verbinder links/rechts werden an den WS28xx Bus angeschlossen (rechts Eingang, links Ausgang)
- **MPX-Anschluss**
  Die Schraubklemmen bzw. 2x2 Stift-Verbinder können wahlweise verwendet werden. Es dürfen nur Signale mit Multiplex Technik angeschlossen werden. Die Markierung der Viessmann 2x2 Stecker muss nach links zu der Markierung | auf der Platine zeigen.
- **Debug-IF**
  Dient der Konfiguration des Moduls mithilfe eines FTDI-Kabels (Kapitel 5).



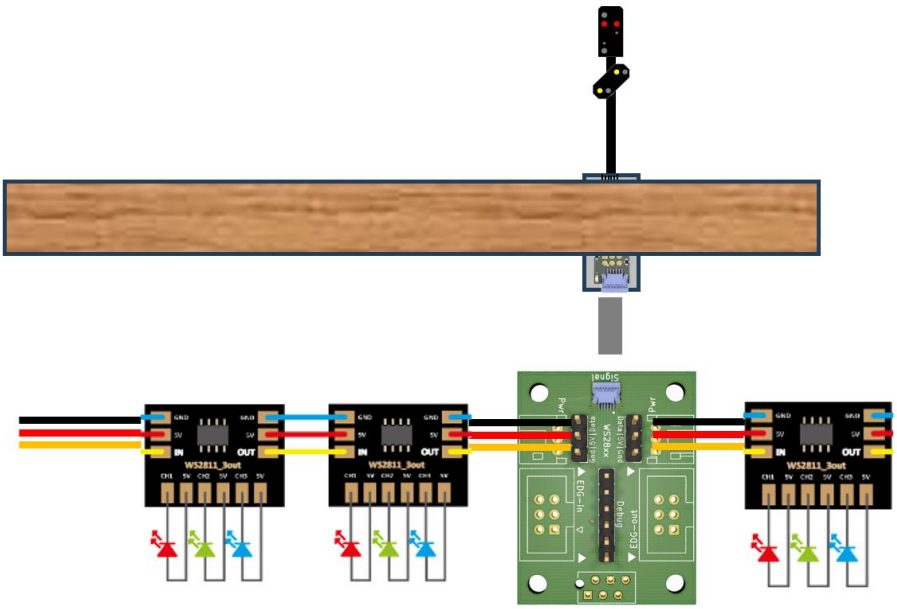
4.5 E14 Modul (MPX)

Es handelt sich um ein 12 Kanal Multiplex LED Treiber mit WS2811 kompatiblen Bus Anschluss. Diese Bauform ist zum Einbau in den Signal Fuß vorgesehen.



Modul Verbindungen

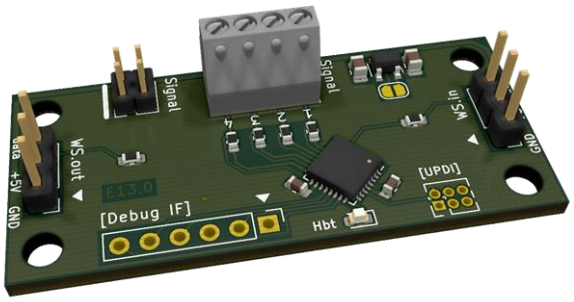
- **Bus-Anschluss**
  Nur über das FFC Adapter Modul (s. Bild).
- **MPX-Anschluss**
  Es können nur Signale mit Multiplex Technik angeschlossen werden.
- **Debug-IF**
  Dient der Konfiguration des Moduls mithilfe eines FTDI-Kabels (Kapitel 5).



> **Wichtig:** Das Modul darf nicht zeitgleich mit dem WS28xx Bus und dem Debug-IF verbunden sein. Bevor das FTDI-Kabel aufgesteckt wird, bitte die Verbindung zum WS28xx Bus trennen.

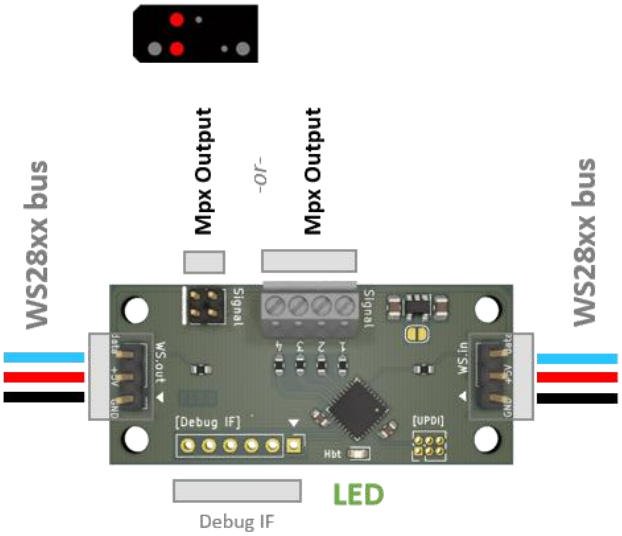
4.6 E15 Modul (Multiplex)

Es handelt sich um ein 12 Kanal Multiplex LED Treiber mit WS2811 kompatiblen Bus Anschluss.
(Identisch zum E13 Modul, allerdings erlaubt dieses Modul keine variable LED-Spannung.)



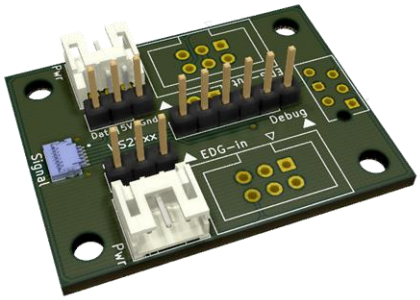
Modul Verbindungen

- **Bus-Anschluss**
  Die Verbinder links/rechts werden an den WS28xx Bus angeschlossen (rechts Eingang, links Ausgang)
- **MPX-Anschluss**
  Die Schraubklemmen bzw. 2x2 Stift-Verbinder können wahlweise verwendet werden. Es dürfen nur Signale mit Multiplex Technik angeschlossen werden. Die Markierung der Viessmann 2x2 Stecker muss nach links zu der Markierung | auf der Platine zeigen.
- **Debug-IF**
  Dient der Konfiguration des Moduls mithilfe eines FTDI-Kabels (Kapitel 5).



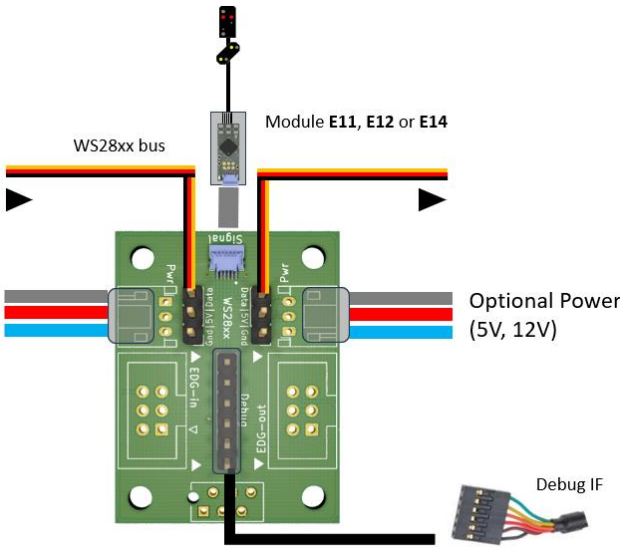
4.7 P18 Modul (Adapter)

Das Adapter Modul konvertiert die 3-polige WS2811 Steckverbindung auf ein 6-poliges FFC-Kabel worüber die E11/E12/E14 Module angeschlossen werden.



Modul Verbindungen

- **Bus-Anschluss**
  Die Verbinder links/rechts werden an den WS28xx Bus angeschlossen (rechts Eingang, links Ausgang)
- **Power**
  (optional) Einspeisung von 5V/GND *(die 12V wird nicht verwendet)*.
- **FFC-Anschluss**
  6-pol (0.5mm pitch) FFC (top/btm) zum Anschluss der Module E11/E12/E14.
- **Debug-IF**
  Dient der Konfiguration des Moduls mithilfe eines FTDI-Kabels. **Wichtig:** Wenn das Debug-Kabel ausgesteckt ist, muss die Verbindung zum WS28xx Bus getrennt sein.

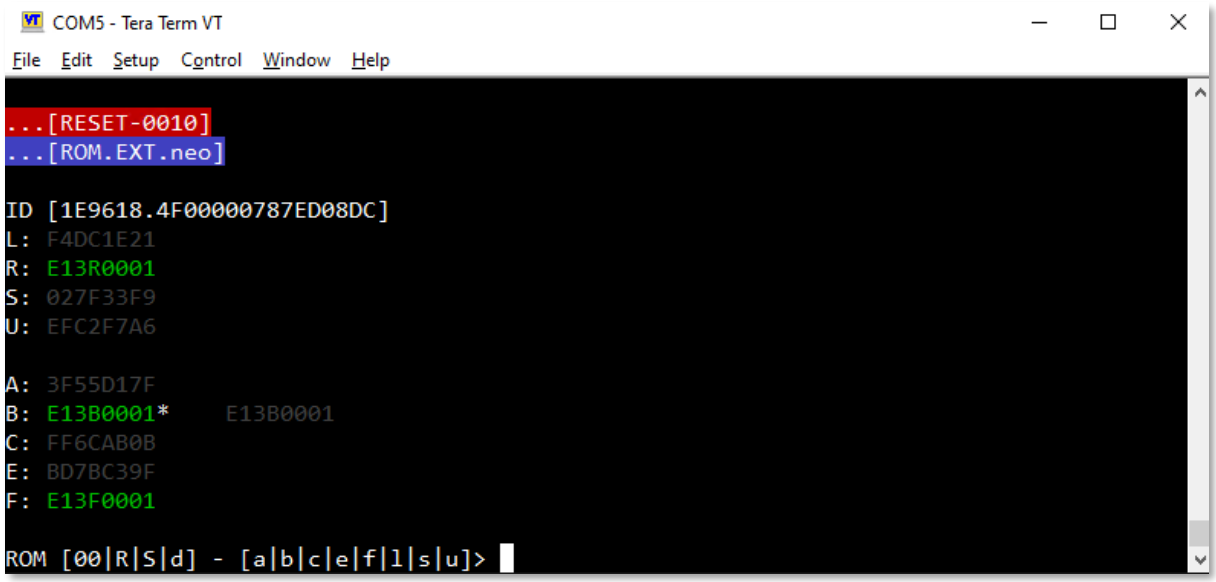


5 Debug-IF

Alle Module ermöglichen den Anschluss eines VT100-fähigen Terminalprogramms (z. B. TeraTerm, PuTTY) um weitergehende Konfigurationsanpassungen sowie Firmware update vorzunehmen. Die erforderlichen Terminal Einstellungen sind in Kapitel-6 beschrieben.

5.1 Konsole

Nach Verbinden des Terminalprogramms muss ein ‚Break' (Tastenkombination abhängig vom verwendeten Terminalprogramm, siehe Kapitel 6) gesendet werden, um das Debug-IF zu initialisieren.

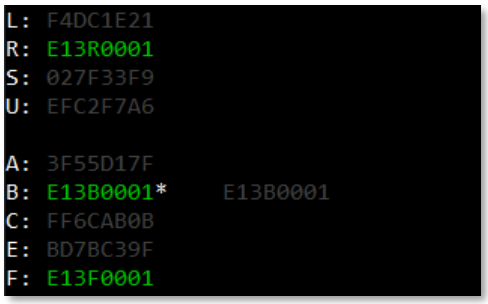


Die auf der Konsole angezeigten Informationen bedeuten,

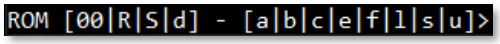
Reset Grund [RESET-####]
0001: Power-On Reset
0002: Brown-Out Reset
0004: external Reset via CPU Reset-Pin
0008: Watchdog Reset
0010: Software Reset
0020: UPDI Reset

Modul Type [ROM.EXT.neo]

ID [Cpu-Typ.64bit-UID]



Speicherbank X: <CRC32> [*]
L: LOCK
R: Rom
S: SIGROW
U: USERROW
A: SRAM
B: Bootloader
C: FUSE
E: EEPROM
F: Main Flash
Über jede Speicherbank wird ein CRC32 berechnet.
Liegt die CRC in einem Erwartungsintervall, wird diese grün angezeigt.
Das (*) zeigt die aktuell selektierte Speicherbank an.



Prompt ##=Selektierte Boot-Optionen
Mögliche Tastatureingaben,
R Software Reset (Modul neu starten)
S Ausführen der aktiven (*) Speicherbank (nur Speicherbank B oder F)
d Hex Dump der aktiven (*) Speicherbank
a|b|c|e|f|l|s|u Setzen der aktiven(*) Speicherbank
0|1|2|3|4|5|6|7|8 Setzen/Löschen der Boot-Optionen

5.2 Befehle

5.2.1 Break

Ein Break erzwingt immer einen Neustart des Moduls (0010: Software Reset). Die Tastenkombination ist vom verwendeten Terminalprogramm abhängig.

TeraTerm: ALT-B
PuTTY: CTRL-Break (bzw. STRG-Unterbrechen)

5.2.2 Reset

Tastatur Eingabe: **R**

5.2.3 Start Optionen auswählen

Startoptionen sind einfache ja/nein Informationen, die beim Start an das Hauptprogramm übergeben werden. Es sind bis zu 8 Optionen (Bits) möglich. Diese werden über die Tasten 0-7 ein- bzw. ausgeschaltet. Die Optionen bedeuten,

Taste-0 [01]: Modul auf Standardeinstellungen setzen (Auslieferungszustand)
Taste-1 [02]: LEDs beim nächsten Start automatisch erkennen und im EEPROM merken
Taste-2 [04]: feste LED-Spannung (=5V) im EEPROM aktivieren
Taste-3 [08]: variable LED-Spannung (=0...5V) im EEPROM aktivieren
Taste-4 [10]: <unbenutzt>
Taste-5 [20]: <unbenutzt>
Taste-6 [40]: <unbenutzt>
Taste-7 [80]: <unbenutzt>

Beispiel: Durch einmaliges drücken der Tasten ‚1' und ‚3' wird im Prompt die Hex-Zahl ‚0a' angezeigt.

ROM [**0a**|R|S|d] – [a|b|c|e|f|l|s|u]>

Nun kann das Modul durch die Eingabe von ‚b' (Kleinbuchstabe) und ‚S' (Großbuchstabe) gestartet werden und die angewählten Optionen werden aktiviert und gespeichert.

Beispiel: Ist keine Option aktiviert (Normalfall), werden die zuletzt gespeicherten Einstellungen aus dem EEPROM übernommen.

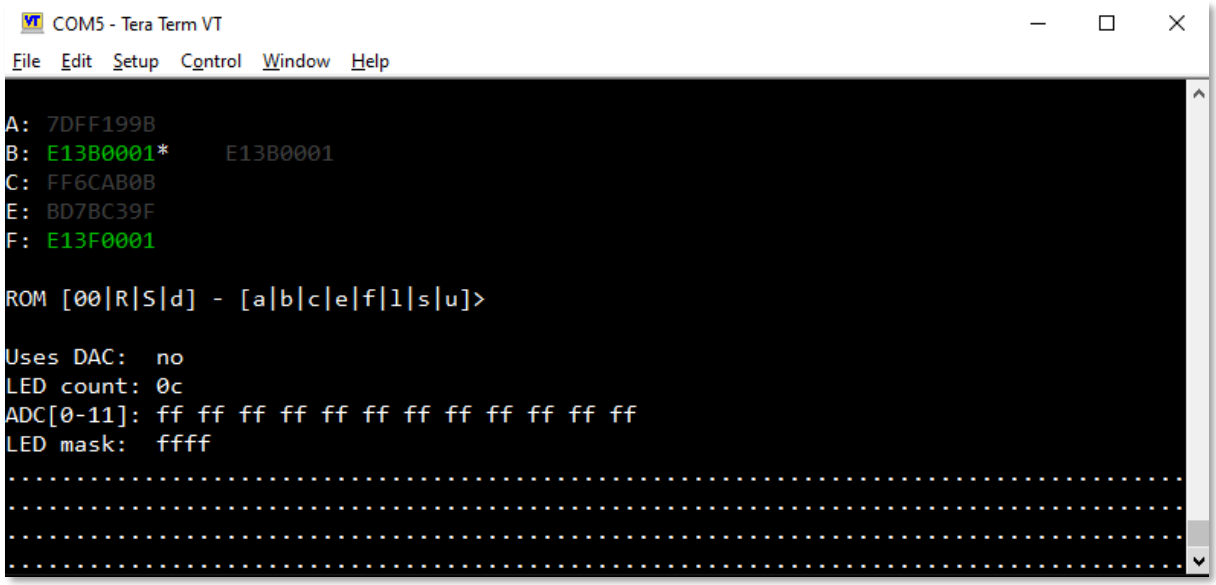
ROM [**00**|R|S|d] – [a|b|c|e|f|l|s|u]>

5.2.4 Modul starten

Tastatur Eingabe: **b S**

ROM [**00**|R|S|d] – [a|b|c|e|f|l|s|u]>

Ist keines der Boot-Optionen Bits gesetzt, bleiben alle Einstellungen im EEPROM unverändert. Die Baugruppe startet mit den zuletzt eingestellten Parametern. Dieses entspricht auch den normalen Start ohne Debug-IF.



Direkt nach dem Start wird immer die aktuelle Konfiguration ausgeben. So kann kontrolliert werden, ob die aktuellen Einstellungen dem Entsprechen was gewünscht ist. (Im obigen Beispiel: feste LED-Spannung, 12 LEDs).

Sobald ein vollständiges WS28xx Frame wurde, wird ein Punkt ‚.' im Debug-IF ausgegeben. Das ist ein Indikator, dass der WS28xx Bus ordnungsgemäß arbeitet. Die Ausgabe der Punkte läuft immer so weiter. Die ‚Break' Tastenkombination (s. 5.2.1) bringt das Modul zurück auf den interaktive Konsole Prompt.

5.2.5 Modul in den Grundzustand setzen

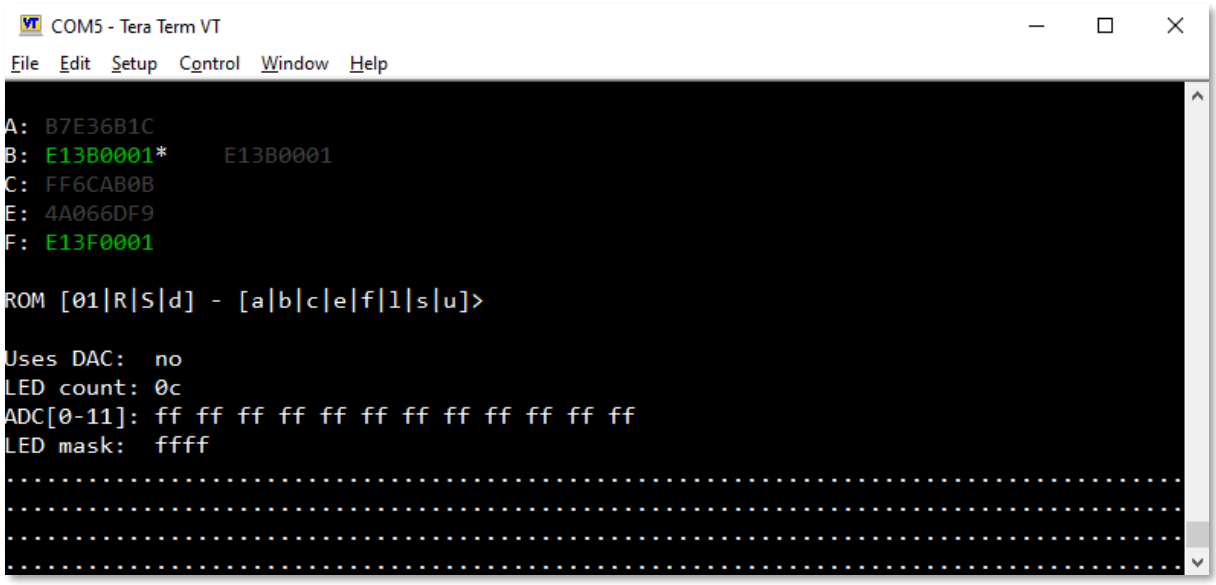
Tastatur Eingabe: **0 b S**

ROM [**01**|R|S|d] – [a|b|c|e|f|l|s|u]>

Bootoption Bit-0 ist gesetzt: Das EEPROM wird in Grundzustand gesetzt,

- feste LED-Spannung (5V)
- maximale LED-Zahl (abhängig vom Modul-Typ)

unabhängig davon, was zuvor konfiguriert war.



Alternativ kann dieses auch mit einem Jumper erreicht werden, sollte kein Debug Kabel verfügbar sein.

Vorgehensweise:

1. Jumper wie im Bild setzen.

Grundzustand

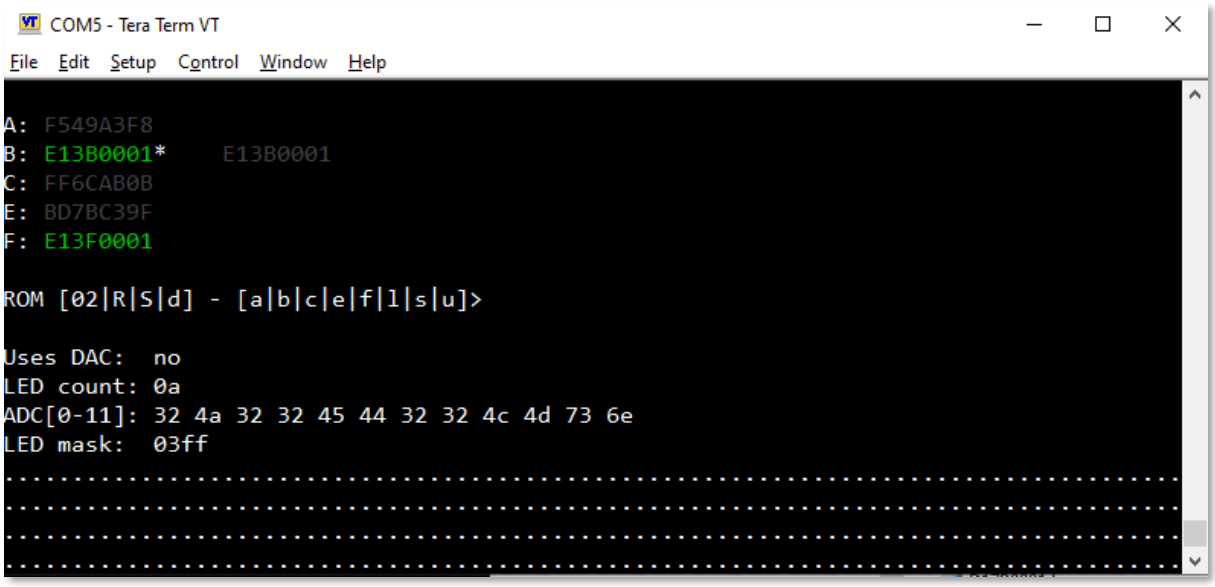
2. Versorgungsspannung einschalten (erzeugt den Grundzustand, Modul-LED flimmert)
3. Jumper entfernen und Versorgungsspannung abschalten.
   Fertig!

5.2.6 LED-Erkennung durchführen

Tastatur Eingabe: **1 b S**

ROM [**02**|R|S|d] – [a|b|c|e|f|l|s|u]>

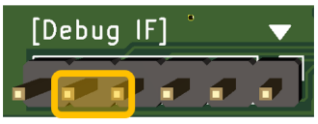
Bootoption Bit-1 ist gesetzt: Die angeschlossenen LEDs werden einmalig beim nächsten Start automatisch erkannt und dieses im EEPROM abgelegt. In dem Beispiel wurden zehn (=0x0a hex) LEDs erkannt. Die ADC-Werte sind die gemessenen Spannungen über den LEDs (nur bei Multiplex) und stellen einen Fingerabdruck des angeschlossenen LEDs dar. Die gesetzten Bits in der LED-Maske (16-bit) zeigen die Anschlusspositionen der erkannten LEDs an. Werden die angeschlossenen LEDs verändert, muss die Erkennung (Option [02]) erneut aktiviert werden.



Alternativ kann dieses auch mit einem Jumper erreicht werden, sollte kein Debug Kabel verfügbar sein.

Vorgehensweise:

1. Signale bzw. LEDs an die Baugruppe anschließen.
2. Jumper wie im Bild setzen.



Automatische LED Erkennung

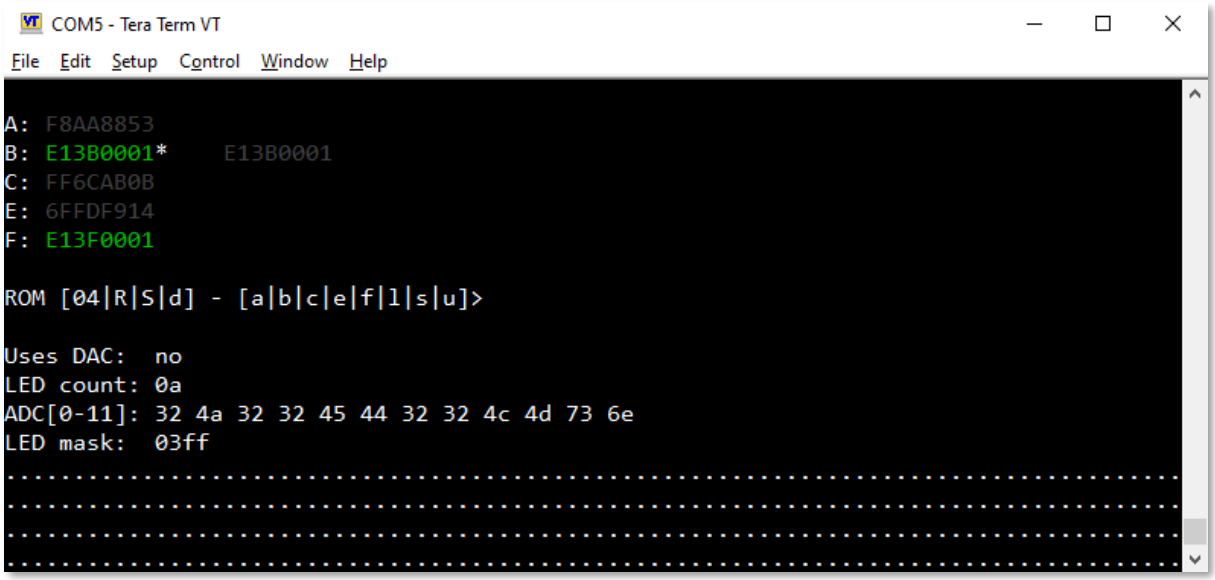
3. Versorgungsspannung einschalten (löst die Erkennung aus, Modul-LED flimmert)
4. Jumper entfernen und Versorgungsspannung abschalten.
   Fertig!

5.2.7 Feste LED-Spannung aktivieren

Tastatur Eingabe: **2 b S**

ROM [**04**|R|S|d] – [a|b|c|e|f|l|s|u]>

Bootoption Bit-2 ist gesetzt: Es wird die feste LED-Spannung aktiviert (Uses DAC=no) und im EEPROM hinterlegt.



Kontrolle: Die Startmeldung „Uses DAC: no“ bedeutet feste LED-Spannung (5V).

Wichtig:

Im Gegensatz zum Modus mit variablen LED-Spannung (s. 5.2.8) entfällt hier das extra Byte auf dem Bus.

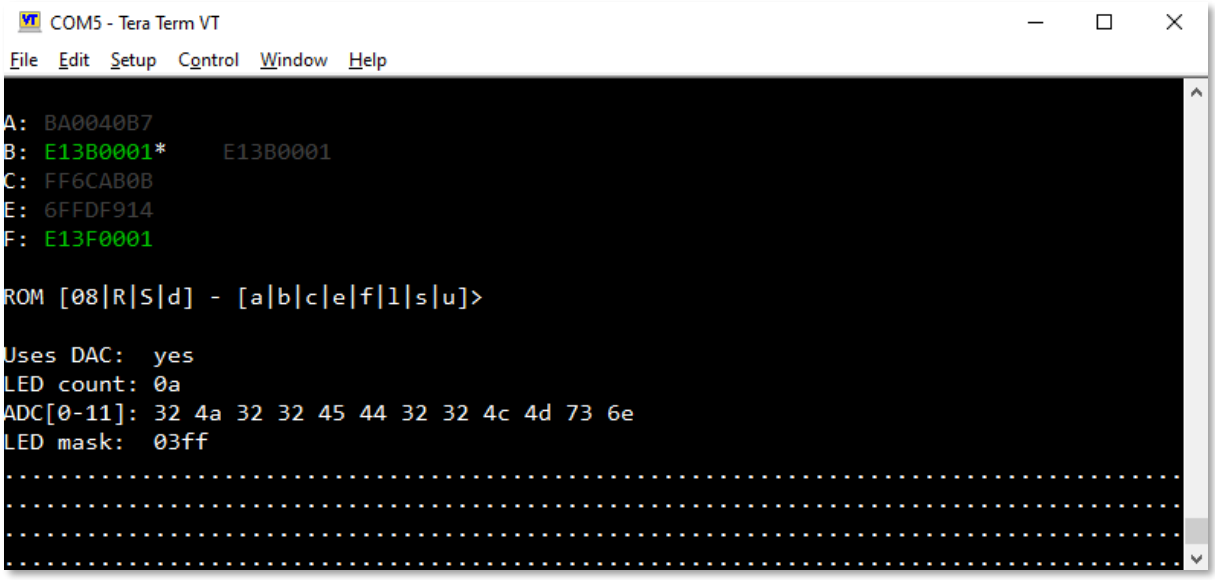
```
{Led-1} {Led-2} ... {Led-N}
```

5.2.8 Variable LED-Spannung aktivieren

Tastatur Eingabe: **3 b S**

ROM [**08**|R|S|d] – [a|b|c|e|f|l|s|u]>

Bootoption Bit-3 ist gesetzt: Es wird die variable LED-Spannung aktiviert (Uses DAC=yes) und im EEPROM hinterlegt.



Kontrolle: Die Startmeldung „Uses DAC: yes“ bedeutet variable LED Spannung (0..5V).

Wichtig:

In diesem Betriebsmodus wird das erste empfangene Byte als Spannungsteiler interpretiert.

```
{Common} {Led-1} {Led-2} ... {Led-N}
```

Die LED-Spannung berechnet sich wie folgt,

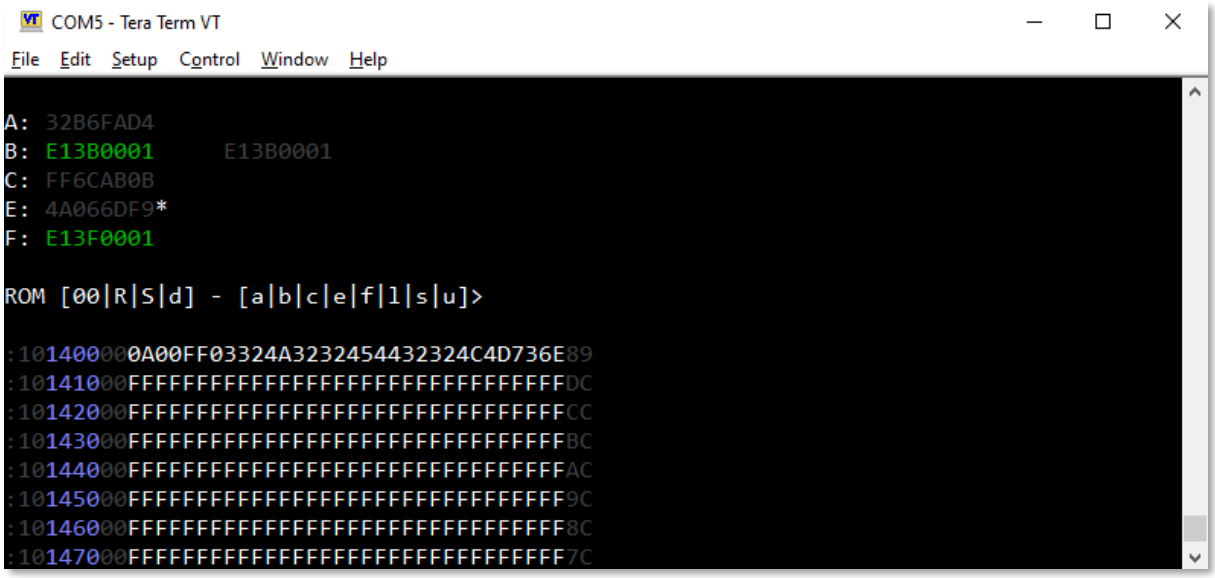
```
Led-Spannung[V] = 5V * {Common} / 255
```

Beispiel: Der Wert {128} reduziert die gemeinsame LED-Spannung auf 2.5V. Das erlaubt, in gewissen Grenzen, die maximale Helligkeit aller angeschlossenen LEDs zu reduzieren. Dabei ist zu beachten, eine LED benötigen immer eine Minimalspannung. Wird diese unterschritten, bleibt die LED dunkel. Der Wert unterscheidet sich in Abhängigkeit der LED-Farbe. Typische Werte sind: Rot >1.6V, Gelb/Grün > 1.9V, Blau/Weiß > 3V.

5.2.9 EEPROM Anzeigen

Tastatur Eingabe: **e d**

Um den aktuellen Inhalt des EEPROM auf dem Bildschirm auszugeben, muss zuerst die Speicherbank-E über die Eingabe von ‚e' selektiert werden. Anschließend kann über die Eingabe von ‚d' der Inhalt der Speicherbank im Intel-HEX Format ausgeben werden.



5.2.10 Firmware Update

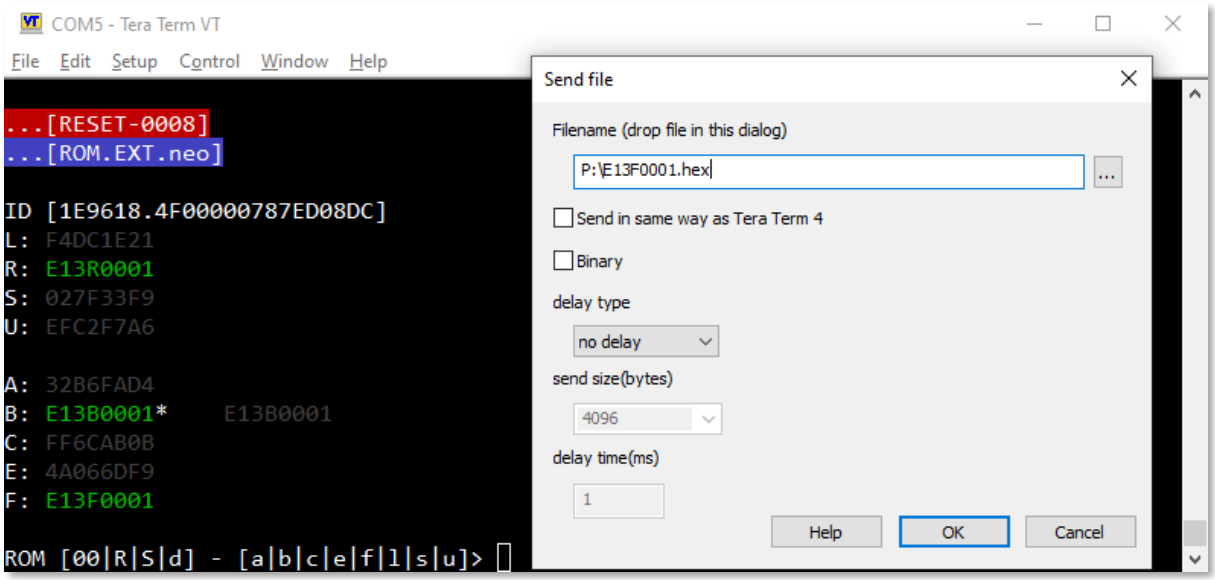
Die aktuelle Firmware wird im Terminal in der Zeile F: <Version> angezeigt. Beispiel,

F: E13F0000

Es wird einfach die entsprechende Firmware Datei übertragen (dauert ca. 1-2 Sekunden). Der Name der Firmware Datei muss dabei dem Modultyp entsprechen,

E10 Module	<-	E10Fxxxx.hex
E11 Module	<-	E11Fxxxx.hex
E12 Module		keine Firmware (native WS2811 Chips)
E13 Module	<-	E13Fxxxx.hex
E14 Module	<-	E14Fxxxx.hex
E15 Module	<-	E15Fxxxx.hex

Beispiel: Firmware update eines E13 Moduls.



Anschließend sollte die neue Firmware Version (xxxx) in grün angezeigt werden,

F: E13Fxxxx

5.2.11 Konfig Upload

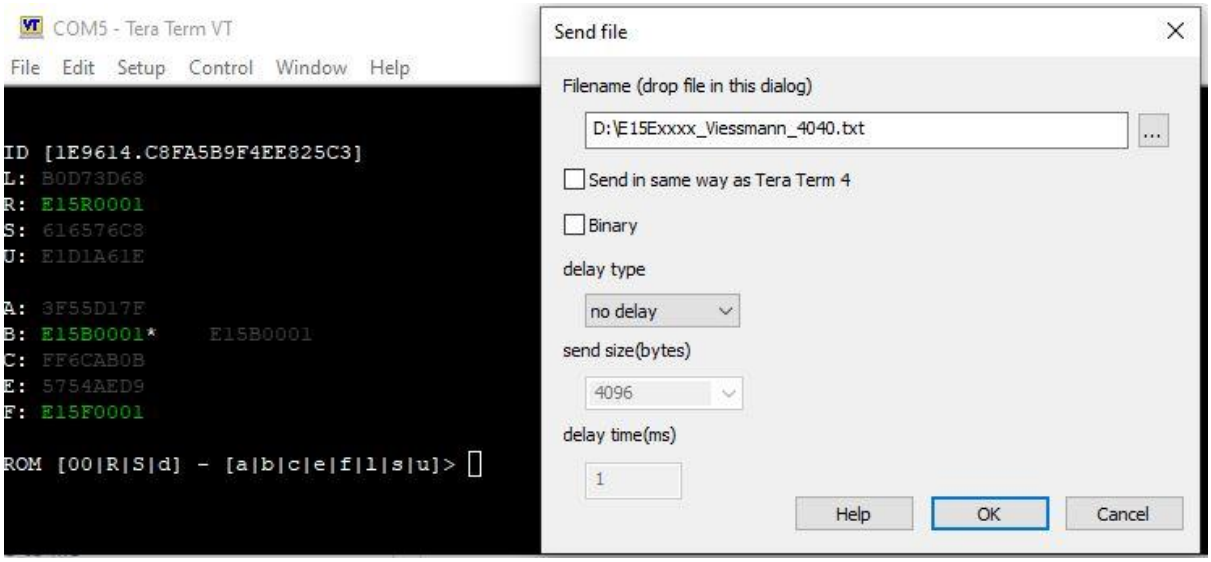
Das Modul kann bei Bedarf auch mittels einer vordefinierten Konfig Datei auf ein bestimmten Signaltyp vorkonfiguriert werden. Dazu muss das Modul nicht mit dem Signal verbunden sein.

Beispiel: Hier wird ein E15 Modul auf den Signaltyp „Viessmann 4040“ eingestellt.

Dazu wird das Modul in den Terminal Modus gebootet und anschließend über die Menü Option „File -> Send file …“ die Datei **E15Exxx_Viessmann_4040.txt** übertragen. Die Daten aus der Datei werden in das EEPROM geschrieben.

Wichtig: die Option ‚Binary‘ darf nicht selektiert sein, und ‚delay type‘ muss auf ‚no delay‘ stehen.

Der Transfer überschreibt alle bisherigen Modul Einstellungen und kann bei Bedarf beliebig oft wiederholt werden. Diese Methode ist lediglich eine Alternative zu der oben beschriebenen automatischen Erkennungsfunktion.



Die Konfig Dateien der verschiedenen Multiplex Signal sind via GitHub verfügbar,

https://github.com/git4dcc/RTB_E15/tree/main/supplemental

6 Terminal Programme

Folgende Einstellungen sind für die Benutzung des Debug-IF erforderlich,

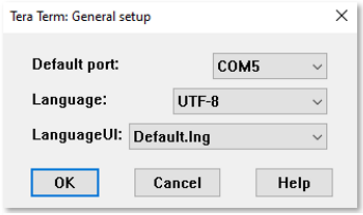
- Baudrate: 1 MBaud
- Datenbits: 8
- Parität: Keine
- Stoppbits: 1
- Flow Control: CTS/RTS Hardware

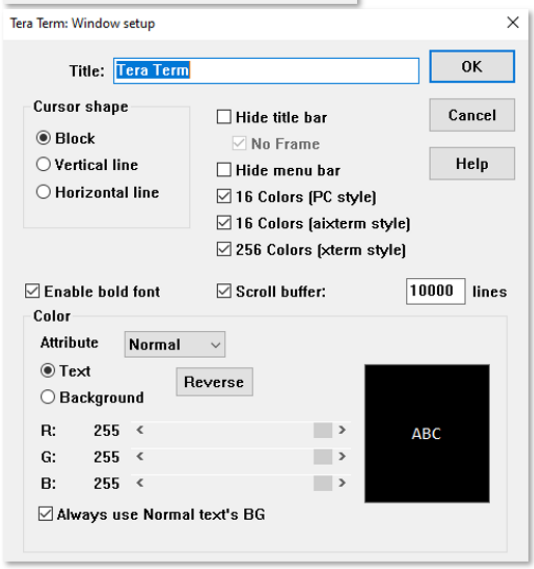
Die notwendigen Einstellungen der Terminal Emulation,

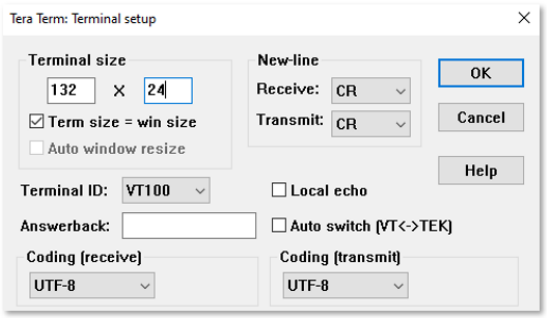
- Emulation: VT100 (oder höher)
- CR: CR
- LF: LF

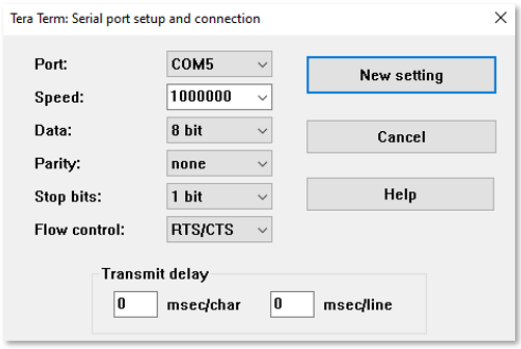
6.1 TeraTerm

TeraTerm ist ein „open source“ Terminal Emulator für Windows. Die z.Zt. aktuelle Version 5.1 wurde erfolgreich getestet. Folgende Einstellungen sind unter TeraTerm vorzunehmen.





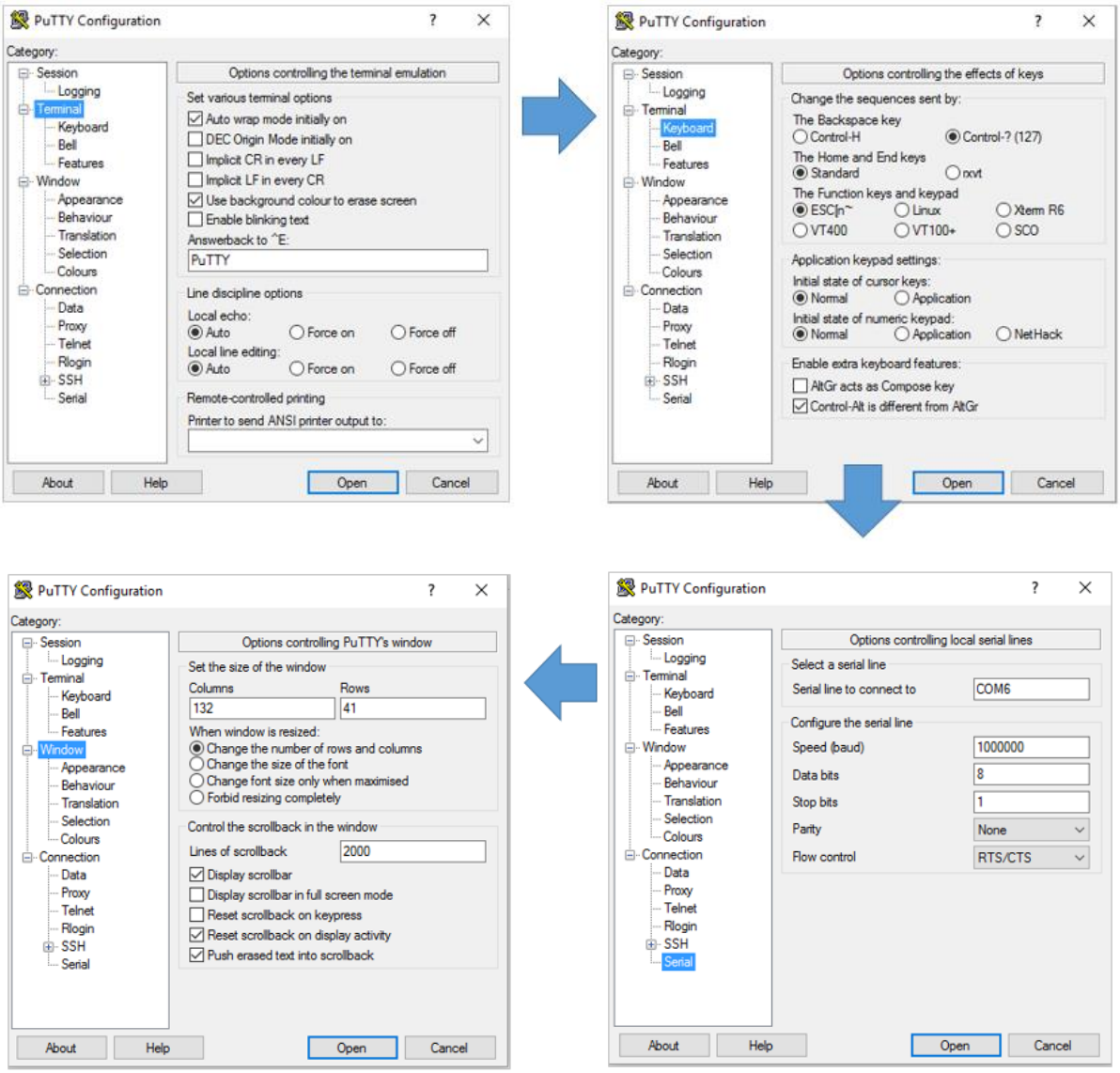




Bei TeraTerm muss die Baudrate „1000000“ über die Tastatur eingegeben werden.
Der ‚Break‘ wird bei TeraTerm über ‚ALT-B‘ ausgelöst.

6.2 PuTTY

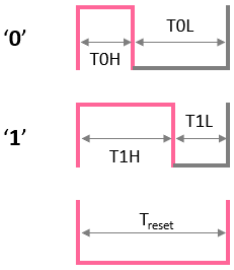
PuTTY ist ein „open source“ Terminal Emulator sowohl für Windows als auch für Linux. Beide Varianten wurden erfolgreich getestet. Folgende Einstellungen sind unter PuTTY vorzunehmen,



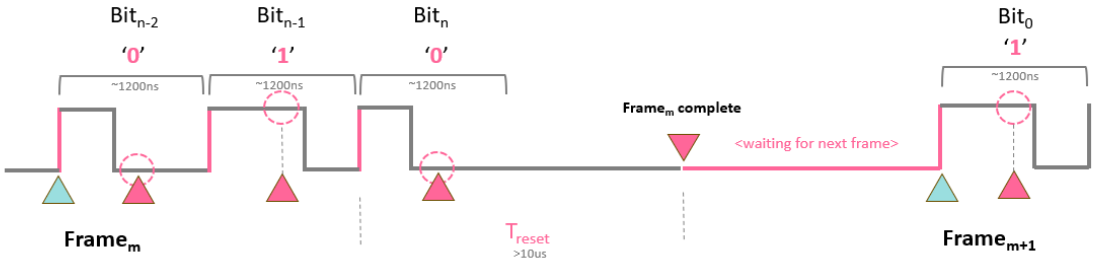
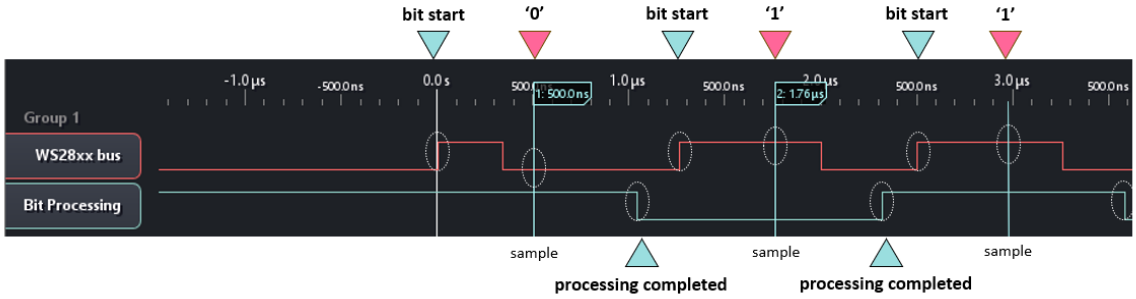
Der ‚Break‘ wird bei PuTTY über ‚CTRL-Break / Strg-Unterbrechen‘ ausgelöst.

7 Bus Timing

Das Bus Timing der Exx Module ist wie folgt spezifiziert,



	min	Max	TxH + TxL
T0H	>150ns	<450ns	≥ 1200ns
T0L			
T1H	>650ns	<900ns	≥ 1200ns
T1L			
T_{reset}	>10us	<1s	



Logik Analysator mit Debug Ausgabe (Bit Processing)