

Grundprinzip von CBUS-Ereignissen

CBUS ist **ereignisorientiert**. Es gibt keine festen „Adressen“ wie bei DCC-Zubehördecodern, sondern **Ereignisse**, die von beliebigen Modulen **erzeugt (Producer)** und von beliebigen Modulen **ausgewertet (Consumer)** werden.

Ein Ereignis ist eindeutig definiert durch:

- NN (Node Number) – die Knotennummer des Producers
- EN (Event Number) – die Ereignisnummer innerhalb dieses Knotens

NN und EN bilden den eindeutigen Ereignisschlüssel im gesamten CBUS. Ein Ereignis wird immer als **ON- oder OFF-Ereignis** gesendet.

Producer und Consumer

Ein Producer erzeugt Ereignisse, z. B.:

- Taster
- Gleiskontakte
- Rückmelder
- Software-Logik

Typisch:

- Taster gedrückt → `EVENT ON`
- Taster losgelassen → `EVENT OFF`

Ein Consumer reagiert auf Ereignisse, z. B.:

- Weichendecoder
- Signaldekoder
- Relaismodule
- Logikmodule

Ein Consumer entscheidet **lokal**, was bei einem bestimmten Ereignis passiert. Ein Node kann nicht nur eines von beiden sein, sondern auch Producer und Consumer.

Der Lernmodus (Learn Mode)

Der Lernmodus dient dazu, **Ereignisse mit Aktionen zu verknüpfen**, ohne feste Programmierung per Adresse.

- „Wenn Ereignis X empfangen wird, führe Aktion Y aus.“

Diese Zuordnung wird **im Consumer gespeichert**, nicht zentral.

1. Ablauf des Lernens

- Consumer in Lernmodus versetzen
- Per Taste am Modul
- Oder per CBUS-Befehl (`ENTER LEARN MODE`)

Der Consumer signalisiert den Lernmodus meist durch:

- blinkende LED
- Statusmeldung auf dem Bus

2. Producer sendet ein Ereignis

- Taster drücken
- Sensor auslösen
- Software sendet `EVENT ON`

Der Consumer empfängt:

- NN
- EN
- Event-Typ (ON oder OFF)

3. Ereignis wird gespeichert

Der Consumer speichert intern:

- `(NN, EN)`
- ob es ein ON- oder OFF-Ereignis ist
- welche Aktion damit verknüpft ist (z.B. Ausgang schalten, Servo fahren, Signalbild setzen)

Das Ereignis ist jetzt „angelernt“.

4. Lernmodus verlassen

- erneut Taste drücken
- Timeout
- expliziter CBUS-Befehl

Ab jetzt reagiert der Consumer **bei jedem passenden Ereignis**, egal von welchem Modul es kommt.

ON- und OFF-Ereignisse

CBUS unterscheidet immer zwei Ereignistypen:

- **ON-Ereignis** → Zustand *aktiv*
- **OFF-Ereignis** → Zustand *inaktiv*

Typische Nutzung:

- Weiche:
 - ON → gerade
 - OFF → abzweigend
- Relais:
 - ON → ein
 - OFF → aus
- Signal:
 - ON → Fahrt
 - OFF → Halt

Viele Consumer können:

- nur ON auswerten
- oder ON und OFF getrennt lernen

Mehrere Consumer, ein Ereignis

Ein zentrales Merkmal von CBUS:

→ **Ein Ereignis kann von beliebig vielen Consumern genutzt werden.**

Beispiel:

- Taster erzeugt Ereignis `(NN=10, EN=5)`
- Weichendecoder schaltet Weiche
- Signaldekoder setzt Signal
- Logikmodul startet eine Fahrstraße

Alles gleichzeitig, ohne zusätzliche Konfiguration im Producer.

Mehrere Ereignisse pro Consumer

Ein Consumer kann:

- viele Ereignisse lernen
- verschiedene Aktionen pro Ereignis ausführen
- ON und OFF getrennt behandeln

Damit lassen sich komplexe Abhängigkeiten rein ereignisbasiert abbilden.

Löschen von Ereignissen

Je nach Modul:

- kompletter Reset aller gelernten Ereignisse
- selektives Löschen einzelner Ereignisse

- Löschen per Lernmodus + spezielles Ereignis

Das Verhalten ist **modulspezifisch**, das Grundprinzip aber identisch.

Zusammenfassung

- CBUS arbeitet **ereignisorientiert**, nicht adressbasiert
- Ereignisse bestehen aus **Node Number + Event Number**
- **Producer senden, Consumer reagieren**
- Der **Lernmodus** verknüpft Ereignisse mit Aktionen
- Ein Ereignis kann **beliebig viele Consumer steuern**
- Die Intelligenz liegt **dezentral in den Modulen**
- Es ist keine zentrale Steuereinheit (Digitalzentrale, Computer mit Modellbahnsteuerungssoftware) erforderlich!

From:
<https://isnix.de/> - **It's boring when it works!**

Permanent link:
<https://isnix.de/doku.php?id=modellbahn:elektronik:steuerung:cbus:events>

Last update: **2026-01-24 12:25**

